SUPPORTING INFORMATION

Failure and Redemption of Statistical and Nonstatistical Rate Theories in the Hydroboration of Alkenes

Johnathan O. Bailey and Daniel A. Singleton*

Department of Chemistry, Texas A&M University, 3255 TAMU,
College Station, Texas 77843, United States
singleton@chem.tamu.edu

Experimental Procedures 4
  General Methods. 4
  Hydroboration of Alkenes 4
  Control Reactions on the Measurement of the Product Ratio 6
Computational Procedures and Supporting Computational Results 7
  General 7
  Statistical CCNM Calculations..8
  Nonstatistical CCNM Calculations 9
  Initialization of Trajectories in a sphere of 53 THF Molecules 10
  RRKM-ME Calculations 12
  The $\Delta E_{\text{"experimental"}}$ Calculation 14
  Approximate Tunneling Corrections for RRKM-ME Calculations 14
  The CLN Calculation 16
  The Parameters for Figure 4b, and Additional Example of Random Walks 17
  The Classical and Harmonic Quantum-Mechanical Variation in Total Molecular Energies....18
  Gas-Phase Analog of Figure 3..21
  Extended Version of Figure 5a..22
  1-Hexene Analog of Figure 5a 23
  Details of the Energy Analysis 23
  Conformation Analysis and the Role of Alternative Conformations of Alkenes......................24
Programs for Calculations, and Sample Input Files 26
  Program Suite PROGDYN..26
    progdynstarterHP..27
    proggenHP..33
    prog1stpoint..42
    prog2ndpoint..47
    progdynb 53
    Program progcfour..66
    Program randgen.c..66
    Program proganal used for 53-THF product-forming trajectories (using PROGDYN)........66
    Program progdynsam..68
    Program proglookstart 69
    Program progKECM 70
    progEchangelocal example..71
    ProgEchangetail example 72

Program progcomptime 73
Program Suite ProgdynONIOM 73
progdynONIOM 75
proggenHP 82
prog1stpoint 91
prog2ndpoint 95
progdynb 101
Program proganal used for 80-THF product-forming trajectories (using ProgdynONIOM) 114
progdyn.conf used for 80-THF product-forming trajectories (using ProgdynONIOM) 116
progdyn.conf used for 80-THF feeder trajectories (using ProgdynONIOM) 119
Sample Input Files for Gaussrate 122
poly.fu5 122
esp.fu70 125
esp.fu71 126
esp.fu73 127
esp.fu75 127
poly.fu51 128
Sample Input Files for Mesmer 128
Dodecene, Anti conformation, CCSD(T)/aug-cc-pvtz, full molecule 128
Dodecene, Anti conformation, CCSD(T)/aug-cc-pvtz, localized calculation 134
Calculated Structures and Complete Energies 139
Guide to Structures, Structure Titles and Their Organization 139
Tables of Energies 139
CCSD(T)/aug-cc-pvdz structures 144
BH_3 144
Butene Complex 144
Butene AMTS 144
Butene MTS 145
B3LYP/6-31G* structures 145
BH_3 145
Propene 145
Propene Complex 145
Propene AMTS 146
Propene MTS 146
Propene AM Product 146
Propene M Product 147
Butene Out of plane 147
Butene In Plane 147
Butene Complex out 148
Butene Complex anti 148
Butene Complex in 148
Butene AMTS out 149
Butene AMTS anti 149
Butene AMTS in 150
Butene MTS out 150

Butene MTS anti....150
Butene MTS in....151
Butene AM Product out....151
Butene AM Product out twisted....151
Butene AM Product out gauche....152
Butene AM Product anti....152
Butene AM Product anti twisted....152
Butene AM Product anti gauche....152
Butene AM Product in....153
Butene M Product out....153
Butene M Product anti....153
Butene M Product in....153
Butene VTS out....154
Butene VTS anti....154
Butene VTS in....154
Hexene out of plane....155
Hexene AM complex out....155
Hexene AM complex anti....155
Hexene M complex out....156
Hexene M complex anti....156
Hexene AMTS out....156
Hexene AMTS anti....157
Hexene MTS out....157
Hexene MTS anti....157
Hexene AM Product out....158
Hexene AM Product out twisted....158
Hexene AM Product out gauche....158
Hexene AM Product anti....159
Hexene AM Product anti twisted....159
Hexene M Product out....159
Hexene M Product anti....159
Octene....160
Octene AM Complex out....160
Octene AM Complex anti....161
Octene M Complex out....161
Octene M Complex anti....161
Octene AMTS out....162
Octene AMTS anti....162
Octene MTS out....162
Octene MTS anti....163
Octene AM Product out....163
Octene AM Product out twisted....164
Octene AM Product out gauche....164
Octene AM Product anti....164
Octene AM Product twisted....165
Octene M Product out....165

Octene M Product anti 165
Dodecene 166
Dodecene AM Complex out 166
Dodecene AM Complex anti 167
Dodecene M Complex out 167
Dodecene M Complex anti 168
Dodecene AMTS out 168
Dodecene AMTS anti 169
Dodecene MTS out 169
Dodecene MTS anti 170
Dodecene AM Product out 170
Dodecene AM Product out twisted 171
Dodecene AM Product out gauche 171
Dodecene AM Product anti 171
Dodecene AM Product twisted 172
Dodecene M Product out 172
Dodecene M Product anti 173
3,3-dimethyl-2-butene 173
3,3-dimethyl-2-butene AM complex 174
3,3-dimethyl-2-butene M complex 174
3,3-dimethyl-2-butene AMTS 174
3,3-dimethyl-2-butene MTS 175
3,3-dimethyl-2-butene AM product 175
3,3-dimethyl-2-butene AM product twisted 175
3,3-dimethyl-2-butene M Product 176
3,3-dimethyl-2-butene M product Twisted 176
References 177

Experimental Procedures

General Methods.

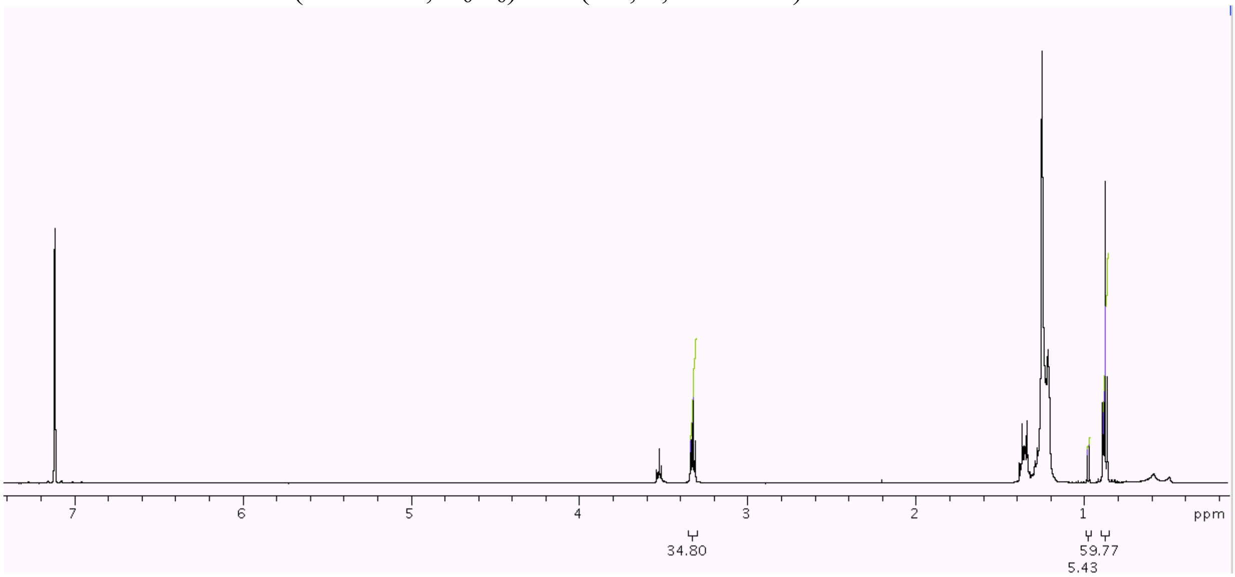
Oven dried glassware was dried in a desiccator and flushed with nitrogen prior to use, and standard syringe-and-septa techniques were employed in all cases. THF was freshly distilled from sodium-benzophenone ketyl. The alkenes were obtained from Sigma-Aldrich and in each case the nominally highest purity grade was employed, without further purification. ^{1}H NMR spectra were taken in C_6D_6 or $CDCl_3$ and the chemical shifts were assigned relative to residual C_6D_5H at δ 7.16 or $CHCl_3$ at 7.26.

Hydroboration of Alkenes

Example Procedure. A flask containing 2.08 g (55 mmol) of $NaBH_4$ was immersed in a 25 °C water bath. The flask was flushed with nitrogen, then 45 mL of THF was added and the solution was stirred rapidly. To this mixture was added dropwise 6.15 mL (50 mmol) of $BF_3O(CH_2CH_3)_2$ and the solution was stirred for 5 min. A solution of 1.1 mL (5 mmol) of 1-dodecene in 5 mL of THF was added dropwise. After 5 min the flask was cooled to 0 °C. The reaction was quenched by the slow addition of 25 mL of water (CAUTION gas evolution), followed by the dropwise addition of 16 mL of 3 M NaOH then 16.0 mL of 30% H_2O_2. The ice bath was removed, and the reaction was stirred at room temperature for 2 h. The aqueous layer was saturated with NaCl, and the organic layer was separated. The aqueous layer was then extracted with 25 mL of THF, and the two organic layers combined. The organic layer was washed twice with saturated NaCl solution, once with saturated NaS_2O_4 solution, and once with saturated NaCl. The organic layer was dried over $MgSO_4$ and concentrated under reduced pressure, though the solvent was not completely removed due to concerns about effects on the product ratio (unseen in control reactions). The residue was analyzed directly by ^1H NMR spectroscopy. The product ratios were measured using the doublet of the methyl group in the secondary alcohol (**M** product) and the triplet (sometimes a multiplet) of the methylene protons nearest the hydroxyl group in the primary alcohol (**A** product).

A sample spectrum for the 1-dodecene reaction is shown below. The secondary methine of the **M** product overlaps with residual THF. All of the alkene product mixtures have nearly identical spectra, but in many spectra the broad peak at d 0.6 interferes with the integration of the peak at 0.85.

1-dodecanol ^1H NMR (500 MHz, C_6D_6) 3.37 (2H, t, J=6.6 Hz), 1.37 (2H, pentet, J=6.8 Hz) overlap with THF, 1.34-1.21 (6H, m), 0.92 (3H, t, J=7.0)
2-dodecanol ^1H NMR (500 MHz, C_6D_6) 1.01(3H, d, J=6.1 Hz)



For *tert*-butylethylene, the product ratios were based on the *tert*-butyl peaks in the proton NMR for the two products. A spectrum of a mixture of 3,3-dimethyl-butan-1-ol and 3,3-dimethyl-butan-2-ol in $CDCl_3$ is shown below.

3,3-dimethyl-butan-1-ol ^1H NMR (500 MHz, $CDCl_3$) 3.71 (2H, AA'XX' m), 1.515(2H, AA'XX' m), 0.925 (9H, s)
3,3-dimethyl-butan-2-ol ^1H NMR (500 MHz, $CDCl_3$) 3.47 (1H, q, J=6.4 Hz), 1.112 (3H, d, J=6.4 Hz), 0.888 (9H, s)

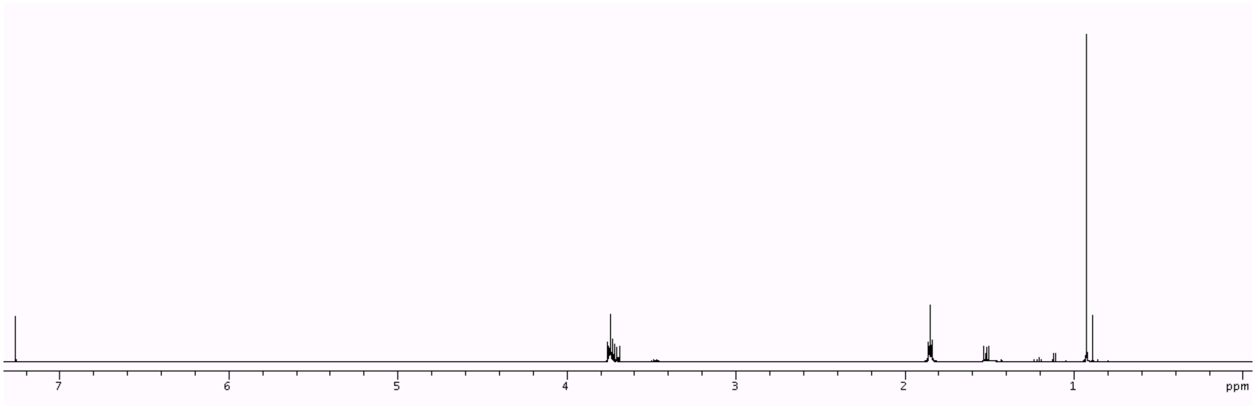


Table S1. Raw Integrals for Determination of Product Ratios

Alkene	**Sample**	**Major Product Peak**	**2 protons**	**Minor Product Peak**	**3 protons**	**% M**
		Raw	Normalized	Raw	Normalized	
1-hexene	1	30.62	15.315	5.32	1.773	10.5
	2	32.46	16.23	5.96	1.986	10.9
	3	33.9	16.95	5.75	1.917	10.2
	4	33.75	16.875	5.85	1.95	10.4
	5	83.92	41.96	16.08	5.36	11.3
	6	34.04	6.35	6.35	2.117	11.1
1-octene	1	34.77	17.385	5.64	1.88	9.8
	2	34.0	17.0	6.01	2.003	10.5
	3	35.34	17.67	5.38	1.793	9.2
	4	34.57	17.285	5.70	1.90	9.9
	5	34.66	17.33	5.66	1.887	9.8
	6	34.75	17.375	5.68	1.893	9.8
1-dodecene	1	34.75	17.375	5.29	1.763	9.2
	2	33.63	16.815	4.98	1.66	9.0
	3	33.01	16.505	4.88	1.627	9.0
	4	34.57	17.285	5.39	1.797	9.0
	5	34.89	17.445	5.33	1.777	8.9
	6	34.80	17.40	5.43	1.81	9.4
		Major Product Peak	9 protons	Minor Product Peak	9 protons	
3,3-dimethyl-2-butene	1	88.96	9.873	11.61	1.29	11.6
	2	85.65	9.517	11.07	1.23	11.4
	3	88.62	9.847	11.38	1.264	11.4
	4	89.40	9.933	10.60	1.117	10.6
	5	89.06	9.896	10.94	1.216	10.9
	6	86.07	9.563	10.83	1.203	11.2

Control Reactions on the Measurement of the Product Ratio

The product ratios for the various alkenes were measured as described above by NMR analysis after an aqueous workup. Because the variation in the product ratio with the size of the alkene is very important in the main text, it was necessary to determine whether the differences in the product ratios could result from differential loss of the primary versus secondary alcohol during the workup. This issue was studied by control reactions in which known mixtures of the alcohols were introduced to reactions matching the experimental conditions but lacking alkene. The reactions were then subjected to the normal workup and the ratio of alcohols was measured, in comparison with the ratio of the original alcohols. The results are summarized in Table S2. In the table, the “Standard” is the mixture of alcohols without being subjected to the reaction conditions and the samples labeled “1” and “2” are recovered after a workup of the control reactions. In each case the ratio of the alcohols was unchanged within the error of the measurement.

Table S2. Raw Integrals for the Determination of Ratios of a Standard Solution of Alcohols

Alkene	**Sample**	**Major Product Peak 1**	**2 protons**	**% M**	**Major Product Peak 2**	**3 protons**	**Minor Product Peak**	**3 protons**	**% M**
		Raw	Norm		Raw	Norm	Raw	Norm	
hexanol	Standard	34.18	17.09	11.7	59.04	19.68	6.77	2.26	11.5
	1	34.42	17.21	11.7	58.75	19.58	6.82	2.27	11.6
	2	34.84	17.72	11.5	58.36	19.45	6.80	2.27	11.7
octanol	Standard	34.28	17.14	12.3	58.53	19.51	7.19	2.40	12.3
	1	34.31	17.16	12.2	58.51	19.50	7.18	2.39	12.3
	2	34.06	17.03	12.2	58.81	19.61	7.13	2.38	12.1
dodecanol	Standard	37.47	18.74	6.0	58.92	19.64	3.61	1.20	6.1
	1	37.33	18.67	5.8	59.25	19.75	3.42	1.14	6.1
	2	37.43	18.72	6.0	58.96	19.65	3.61	1.20	5.8

Example Procedure. A flask containing 2.12 g (55 mmol) of $NaBH_4$ was immersed in a 25 °C water bath. The flask was flushed with nitrogen, then 45 mL of THF was added and the solution was stirred rapidly. To this mixture was added dropwise 6.15 mL (50 mmol) of $BF_3O(CH_2CH_3)_2$ and the solution was stirred for 5 min. A solution of 0.8 mL (5 mmol) of a known mixture of 1- and 2-octanol (standard in Table S2) in 5 mL of THF was added dropwise. After 5 min the flask was cooled to 0 °C. The reaction was quenched by the slow addition of 25 mL of water (CAUTION gas evolution), followed by the dropwise addition of 16 mL of 3 M NaOH then 16.0 mL of 30% H_2O_2. The ice bath was removed, and the reaction was stirred at room temperature for 2 h. The aqueous layer was saturated with NaCl, and the organic layer was separated. The aqueous layer was then extracted with 25 mL of THF, and the two organic layers combined. The organic layer was washed twice with saturated NaCl solution, once with saturated NaS_2O_4 solution, and once with saturated NaCl. The organic layer was dried over $MgSO_4$ and concentrated under reduced pressure. The residue was analyzed directly by ^1H NMR spectroscopy.

Computational Procedures and Supporting Computational Results

General

Calculations of structures, energies, and frequencies employed default procedures in Gaussian09[1,2,3] unless otherwise noted, with the exception that DFT calculations generally employed an ultrafine grid. Complete structures and energetics are provided in sections below. All absolute energies are in Hartrees. All relative energies are presented in kcal/mol.

Some of the dynamics calculations, including all of the 53-THF ONIOM model calculations, all of the single-trajectory calculations, and all of the steepest-descent paths in mass-weighted coordinates, employed the program suite PROGDYN. PROGDYN consists of a series of component programs written as either Unix shell scripts or awk programs. Gaussian09 was used to calculate the forces at each point in trajectories. A full description of PROGDYN including listings of the subprograms can be found in a later section. The latest version of this program can be obtained by emailing Daniel Singleton at singleton@chem.tamu.edu. The original version of this program was published in the Supporting Information of a previous paper.[4]

Other dynamics calculations employed the new program ProgdynONIOM. ProgdynONIOM is a new program that has adapted PROGDYN to the use of a combination of A. Gaussian09 calculations for the DFT calculations needed for an ONIOM and B. MOPAC2016®[5] for all semi-empirical calculations. A full description of ProgdynONIOM including listings of the subprograms can be found in a later section.

The TST and CVT/SCT calculations in the main text made use of the programs GAUSSRATE (reference 11 in the main text) and POLYRATE (reference 12 in the main text, see the last section for the complete reference).[6] These programs were modified in minor ways. In particular, the subroutine *mepout* was modified to output frequencies with higher precision, and the utility program *shuttle* was modified to save copies of all frequency calculations so that data could be extracted from them later. To show the complete set of options selected for these calculations, a sample set of input files for GAUSSRATE / POLYRATE are given in a later section, along with some additional details of the calculations.

Statistical CCNM Calculations

The key numbers that went into the statistical CCNM calculations were:

K The equilibrium ratio of **A** : **M**, based on their CCSD(T)/aug-cc-pvtz//B3LYP/6-31G* free energies for the initially formed products (ΔG_{AB} in the table below)

k_A The CVT/SCT rate constant (CCSD(T)/aug-cc-pvtz VTST-ISPE) for formation of **A** from **INT**, obtained from POLYRATE, in s^{-1}.

k_B The CVT/SCT rate constant (CCSD(T)/aug-cc-pvtz VTST-ISPE) for formation of **B** from **INT**, obtained from POLYRATE, in s^{-1}.

ΔG_{INT} The free energy for formation of **INT** from separate starting materials (but small changes in this value make no difference in the predicted product ratio in the current case).

$\Delta G^{\ddagger}_{VTS}$ The free energy barrier for the formation of **INT** from separate starting materials (small changes in this value also make no difference in the predicted product ratio in the current case).

The table below shows the numbers used and, in the case of K, their raw-number origin. For 1-octene and 1-dodecene, the same numbers were used as in the 1-hexene case, but the final ratios were adjusted slightly based on the difference seen along the series in corresponding CCSD(T)/aug-cc-pvdz calculations.

Table. Input Numbers for Statistical CCNM Calculations.

propene	propene 70 °C	butene anti	butene out	hexene anti	hexene out	*tert*-butylethylene
G **A** product	G **A** product	G **A** product	G **A** product	G **A** product	G **A** product	G **A** product
-144.196604	-144.201910	-183.412376	-183.413089	-261.845204	-261.845838	-261.850641
G **M** product	G **M** product	G **M** product	G **M** product	G **M** product	G **M** product	G **M** product
-144.194312	-144.199625	-183.4098758	-183.4108907	-261.842759	-261.843849	-261.846431

ΔG_{AB}	ΔG_{AB}	ΔG_{AB}	ΔG_{AB}	ΔG_{AB}	ΔG_{AB}	ΔG_{AB}
-1.44	-1.43	-1.57	-1.38	-1.53	-1.25	-2.64
K at 25 °C	K at 70 °C	K at 25 °C	K at 25 °C	K at 25 °C	K at 25 °C	K at 25 °C
11.33	8.19	14.13	10.2630	13.33	8.22	86.43
k_A	k_A	k_A	k_A	k_A	k_A	k_A
2.004E+12	2.051E+12	2.461E+12	3.936E+12	2.317E+12	3.749E+12	2.825E+12
k_B	k_B	k_B	k_B	k_B	k_B	k_B
4.602E+10	7.328E+10	8.909E+10	8.266E+10	8.112E+10	8.620E+10	7.484E+10
ΔG_{INT}	ΔG_{INT}	ΔG_{INT}	ΔG_{INT}	ΔG_{INT}	ΔG_{INT}	ΔG_{INT}
-0.6	0.9	-0.7	-0.9	-0.8	-0.9	-0.4
$\Delta G^{\ddagger}_{VTS}$	$\Delta G^{\ddagger}_{VTS}$	$\Delta G^{\ddagger}_{VTS}$	$\Delta G^{\ddagger}_{VTS}$	$\Delta G^{\ddagger}_{VTS}$	$\Delta G^{\ddagger}_{VTS}$	$\Delta G^{\ddagger}_{VTS}$
5.9	5.9	5.9	6.0	5.9	5.9	5.9

The numbers now needed for the calculation are k_{*1}, k_C, k_{*2}, k_{*2A}, k_{*2B}, and $R_{A/B}$; see main text reference 10 (Zheng, Papajak, and Truhlar) for a definition of these terms.

From $\Delta G^{\ddagger}_{VTS}$, the k_{*1} was calculated from $k_BT/h \exp(-\Delta G^{\ddagger}_{VTS}/RT)$ and multiplied by $1000*24.465/N_A$ to put k_{*1} in ordinary gas phase units.

The same thing was done with ΔG_{INT} to obtain k_C.

k_{*2A} was calculated as $k_A*\exp(-\Delta G_{INT}/RT)*1000*24.465/N_A$

k_{*2B} was calculated as $k_B*\exp(-\Delta G_{INT}/RT)*1000*24.465/N_A$

k_{*2} was calculated as the minimum of k_C versus $k_{*2A} + k_{*2B}$

$R_{A/B}$ is just K

The final results were then obtained straightforwardly (with no further unit conversions) using the equations in reference 10. These calculations were able to reproduce the results of Zheng, Papajak, and Truhlar. An example spreadsheet performing these calculations, saving much work, would be available on request from Dr. Singleton.

Nonstatistical CCNM Calculations

The nonstatistical CCNM calculation involves the calculation of a revised $R_{A/B}$, which is plugged into the calculation above in place of the K. The calculation is complex, and we leave is full description to the Zheng, Papajak, and Truhlar paper. An example spreadsheet performing these calculations, saving much work, would be available on request from Dr. Singleton. The key inputs for the calculation are the vibrational frequencies, energies for **TSA** and **TSM** (their CCSD(T)/aug-CC-pvtz energies and the B3LYP thermal correction to these energies), and a distance factor *l*. In the Zheng, Papajak, and Truhlar paper, *l* was taken as 10.2 Å for both the **A** and **M** pathways, but here *l* was measured based on the POLYRATE calculations for each pathway. The *l* values employed are listed in the table below. Following Zheng, Papajak, and Truhlar, we set the relaxation time constant $\tau_{relax} = 100$ fs.

The table also lists the $R_{A/B}$ values obtained. Plugging these values into the statistical CCNM calculation above in place of K leads to results in column 6 of Table 1 of the main text. It should be noted that $R_{A/B}$ is in each case greater than K due to the longer *l* values for the **A** pathway.

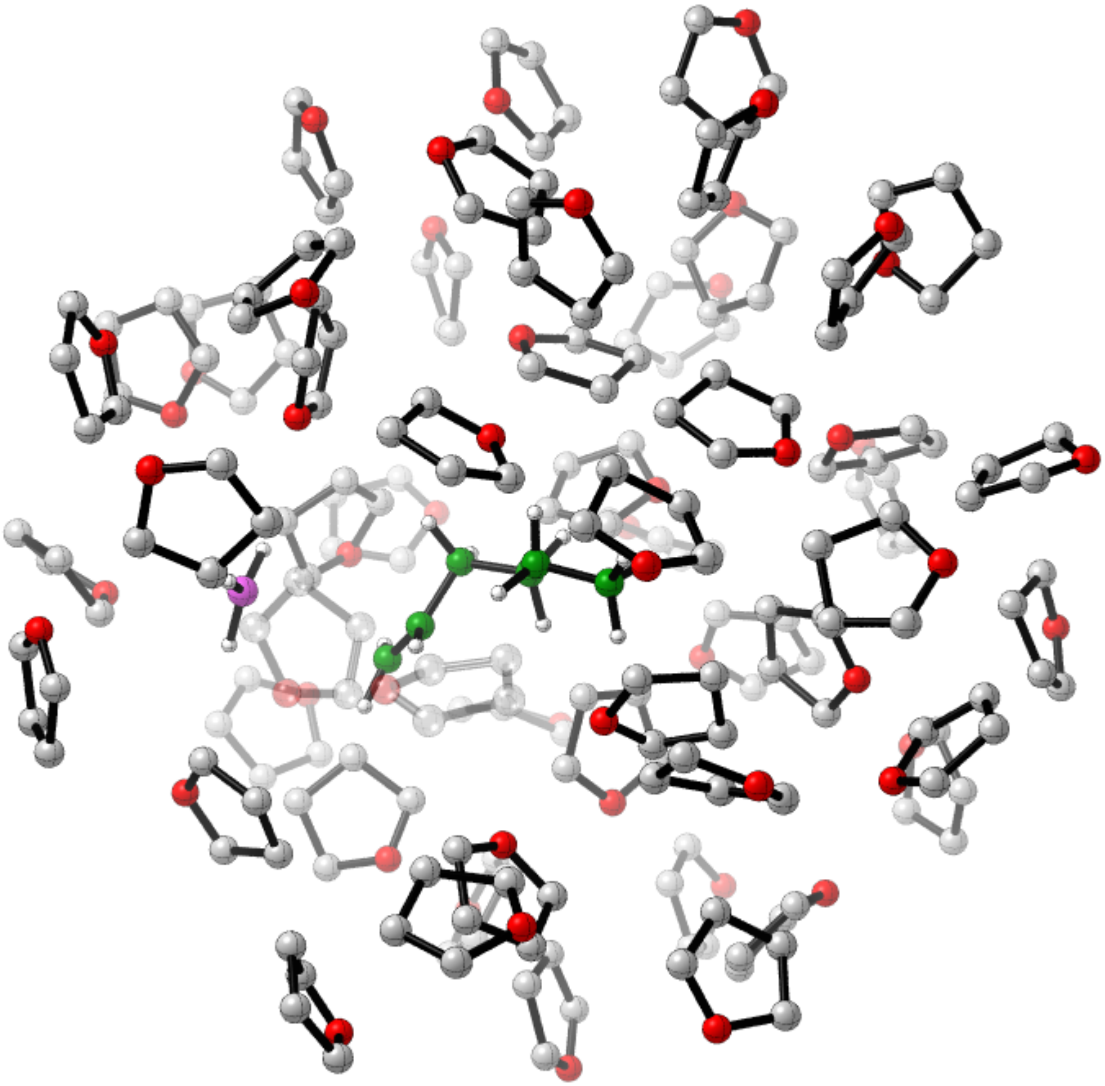
Table. POLYRATE-calculated distance factors *l* (in Å) from **TSA** or **TSM** to their corresponding products.

	A pathway	**M** pathway	$R_{A/B}$
propene	7.46	4.34	17.7
propene at 70 °C	7.46	4.34	12.25
butene out	7.68	4.4	16.215
butene anti	7.52	5.63	17.81
hexene out	10.3	4.72	15.0
hexene anti	12.98	4.76	27.44
octene out	11.68	5.3	10.9
octene anti	11.96	6.21	22.0
dodecene out	15.79	5.93	15.5
dodecene anti	16.25	7.41	19.5
tert-butylethylene	9.06	5.26	130

Initialization of Trajectories in a sphere of 53 THF Molecules

For classical dynamic trajectories inside a sphere of 53 THF molecules, the method for generating starting structures was a bit complicated. First a series of boxes of 53 THF molecules that were generated and equilibrated for a previous study were molded into a 12.3 Å radius sphere using the *sphereon* capability in PROGDYN, using a *sphereforce* parameter of 0.01 (11.8 kcal/mol/Å) during the molding process while the temperature was held at 1000 K. After the sphere formed, the pi complex of dodecene was pushed into the center of the sphere, using the *zeroatom* feature of PROGDYN, while using a harmonic potential to keep the reactive boron and olefinic atoms in place at an approximate location of the VTS (using the applyforce feature in PROGDYN.) When the reactants were in the center of the sphere, the whole system was cooled to 298.15 K using the *thermostat* feature with a *thermostatmult* of 0.999, which removes 0.1% of the energy per fs. After reaching 298.15 K, these "feeder trajectories" were continued up to 42,000 fs. At intervals of 250 fs, points and velocities were extracted and used as the starting point of new trajectories, with no constraints except retention of the sphereforce, integrated forward and backward in time until the products **A** or **M** were formed (defined by a C-B distance < 1.6 Å and a C-H distance < 1.1 Å) or the borane had dissociated (defined by all of the hydrogens of the BH_3 being greater than 4.5 Å from the olefinic carbons). Approximately 65% of the trajectories were extracted after > 30,000 fs after the temperature was equilibrated, but approximately 35% of the trajectories were extracted with <10,000 fs of equilibration. No difference was discerned between the short-equilibration and long-equilibration derived trajectories. A sample set of parameters for the trajectories is given in a later section.

A series of feeder trajectories for the reaction of 1-hexene were generated from the feeder trajectories for 1-dodecene by manually deleting the longer chain from geoPlusVel files (generated with the program progdynsam, listed in a later section) after 33,000 fs of equilibration. The feeder trajectories were continued with points extracted from 1000 to 16,000 fs after equilibration. Shown below is a typical starting point for the hydroboration of 1-hexene in 53 THF molecules (1-hexene in green and borane in pink the hydrogens on the solvent have been omitted for clarity and solvent molecules that obscured the alkene were made transparent.)

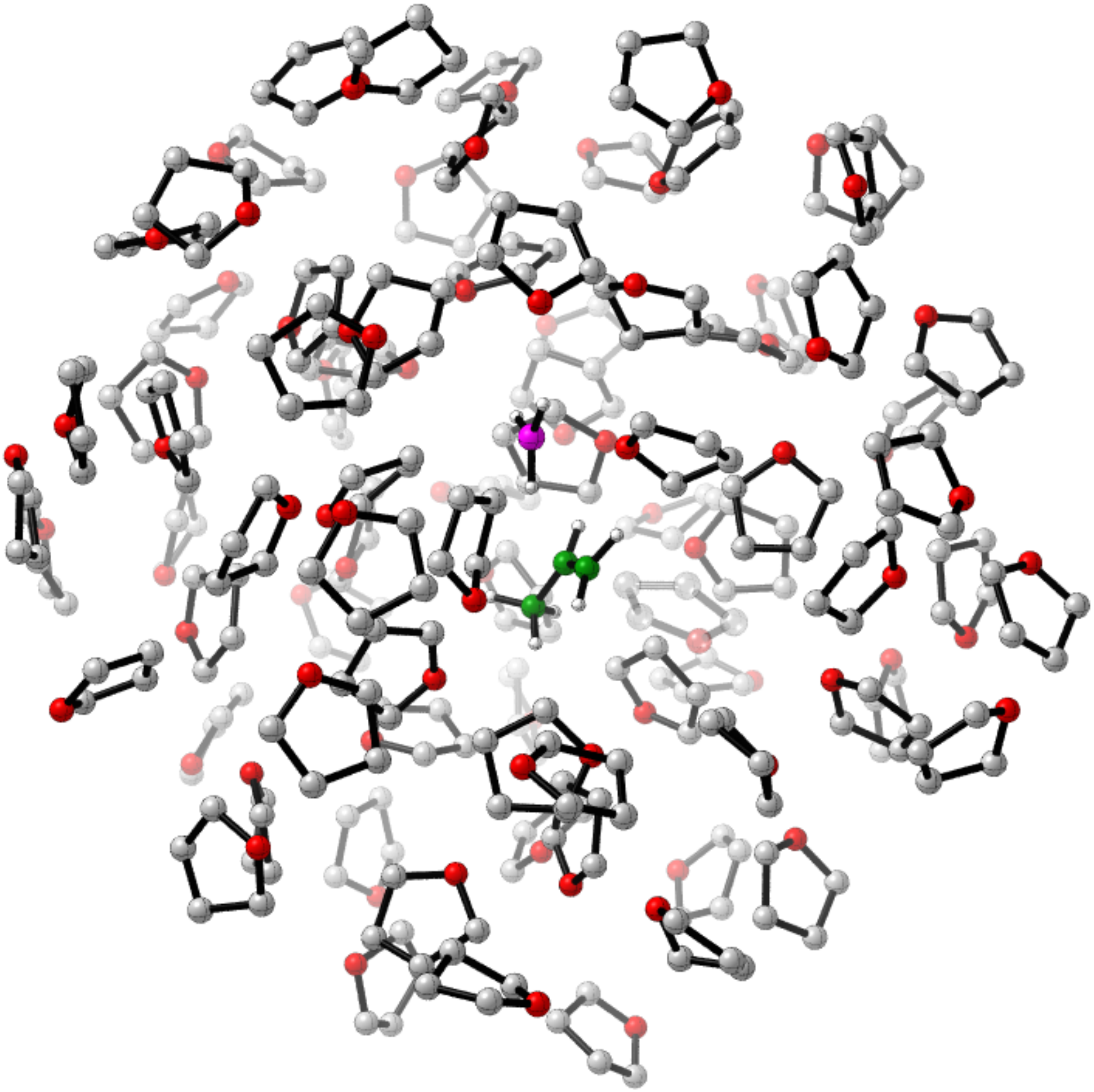


A series of feeder trajectories for the reaction of propene were generated from the feeder trajectories for 1-hexene by manually deleting the longer chain from geoPlusVel files (generated with the program progdynsam, listed in a later section) after 16,000 fs of equilibration. The feeder trajectories were continued with points extracted from 1500 to 15,000 fs after equilibration.

Initialization of Trajectories in a sphere of 80 THF Molecules

A series of starting points for the trajectories using 80 THF molecules were generated from scratch, using the program packmol[7] to place each approximate VTS for propene, 1-hexene, and 1-dodecene along with 80 THF molecule inside a sphere with a radius of 14.8 Å. Using ProgdynONIOM, these spheres were equilibrated at 298.15 K with radii 14.5 Å for propene, 14.8 Å for 1-hexene, and 15.0 Å for dodecene. Trajectory starting points were extracted at 250 fs intervals as above with a range from 2250 fs to 24,240 fs for propene, 3500 fs to 32,000 fs for 1-hexene, and 2750 fs to 15250 fs for 1-dodecene. The starting points were then integrated forward and backward in time as above using the ProgdynONIOM program, using B3LYP/6-

31G* for the model layer and PM6-D3H4 for the real layer, and including an additional small empirical dispersion correction as discussed in a later section. Shown below is a figure of a typical starting point for the hydroboration of propene in 80 THF molecules (propene in green and borane in pink the hydrogens on the solvent have been omitted for clarity and solvent molecules that obscured the alkene were made transparent.)



RRKM-ME Calculations

The program Mesmer (https://sourceforge.net/projects/mesmer/) was employed for the RRKM-ME calculations. Sample input files are given in a later section. Table S2 shows the

Lennard Jones parameters used for the alkenes and THF. The alkene parameters were used when available, but the alkane was substituted when the parameters for a particular alkene was unable to be located. As demonstrated in a sample input, the localized RRKM-ME calculations were carried out by replacing the frequencies of the larger molecules (alkene, complex, both TSs) with that of propene, while leaving the energetics and Lennard Jones parameters unchanged.

Table S2 also shows the energies used in Mesmer for **INT**, **TSA**, and **TSM** in the calculations, relative to zero for the starting BH_3 and alkenes. These energies are CCSD(T)/aug-cc-pvtz//B3LYP/6-31G* + zpe relative energies expressed in kJ/mol. The octene and dodecene numbers were taken from those for hexene. (The B3LYP relative energies for hexene versus octene versus dodecene differed by less than 0.1 kcal/mol.)

As a technical choice, the use of these energies relative to separate alkene + BH_3 is debatable versus the alternative of using energies relative to the canonical variational transition states. The former choice was previously employed by Glowacki *et al* in reference 6 in the main text, while as a reviewer of that paper we argued for the latter choice. Here we adopted the choice of Glowacki *et al*, though exploratory studies suggest that the choice usually makes no significant difference because the small difference in the initial energy in **INT** is countered in the last-step parameterization of the cooling rate. The place where the choice does make a difference is in calculating the upper bound on **M** product from dodecene (with an ordinary RRKM-ME calculation) if there is no cooling ; this goes down from 6.1% to 5.4%, further from experiment. The adoption of the same choice as Glowacki is then the most favorable possibility for the ordinary RRKM-ME calculations.

As another technical choice, the use of the same Lennard Jones parameters for the localized RRKM-ME calculations is debatable. On one side, if the excess energy is completely localized then collisions with the longer chain of the larger alkenes, exceeding those with shorter chains due to the differing σ values, cannot directly take away energy from the local area. On the other side, the trajectories clearly show that the larger systems have both their local areas and the overall molecule cooled more quickly. While the systems may act as if their heat is localized for the purpose of predicting selectivity in reactions, the trajectories provide no support for viewing the cooling of the molecules as a localized phenomenon. As we note in the main text, the qualitative choice of what constitutes the local system is a failing of the current localized RRKM-ME model that needs improvement.

Table S2. Lennard Jones Parameters and Relative Energies for RRKM-ME calculations

	σ[8,9,10]	ε/k[6,7,8]	**INT** (kJ/mol)	**TSA** (kJ/mol)	**TSM** (kJ/mol)
propene*	4.50	214.9	-40.931	-40.646	-30.504
Butene* out	4.89	240.8	-43.107	-43.925	-33.917
Butene* anti			-41.868	-42.215	-33.263
Hexene* out	5.5409	265.1	-43.034	-43.930	-34.079
Hexene* anti			-42.457	-42.688	-33.644
Octene** out	6.0263	264.3	-43.034	-43.930	-34.079
Octene** anti			-42.457	-42.688	-33.644
Dodecene** out	6.7697	272.1	-43.034	-43.930	-34.079
Dodecene** anti			-42.457	-42.688	-33.644
3,3-dimethyl-2-butene**	5.689	421	-40.766	-41.348	-32.403

THF	3.57	68.1	N/A	N/A	N/A

*LJ Parameters taken from alkene. **LJ parameters taken from alkane

The $\Delta E_{\text{"experimental"}}$ Calculation

The $\Delta E_{\text{"experimental"}}$ calculation was carried out by taking the input files used for the localized RRKM-ME calculations, turning off all cooling of **INT** by changing E_{down} to ~0, then decreasing the exothermicity of the reaction until the predicted selectivity matched the experimental selectivity. In other words, the calculation is a pure localized RRKM calculation, with no parameterization outside of the setting of $\Delta E_{\text{"experimental"}}$. This is why we note in the main text that there is no assumption of either collisions in solution or an exponential-down model in the $\Delta E_{\text{"experimental"}}$ calculation.

A subtle inaccuracy in this calculation is that the distribution of energies used for the calculation is not the same as the distribution of energies in the actual reaction. In the RRKM calculation for $\Delta E_{\text{"experimental"}}$, the distribution of energies is just that in the starting materials plus the exothermicity of the reaction. In the actual reaction, there were be some reactions where the localized portion of the molecule has cooled little, so that the energy is higher than any included in the RRKM calculation, and there will also be some reactions where the localized portion of the molecule has cooled almost completely, so that the energy is lower than any in the RRKM calculation. Our assumption in the $\Delta E_{\text{"experimental"}}$ calculation is that the average energy found is a reasonable approximation of the average that would give rise to the experimental observations.

Approximate Tunneling Corrections for RRKM-ME Calculations

Although one-dimensional tunneling corrections are implemented in some RRKM-ME programs, including *Mesmer*, there were four problems with applying such programs to the current problem. The first was that we desired the greater accuracy of a multidimensional tunneling correction. The second was that we wanted to also allow for variational dividing surface effects. The third was that we wanted to perform the calculations using CCSD(T)/aug-cc-pvtz energetics, and it was impractical to perform the tunneling calculation with sufficient directly obtained points. (With some degree of inventiveness, an interpolation scheme might have been effective.) The fourth is that we consistently encountered problems in applying tunneling corrections that appeared to result from numerical convergence issues, so that the tunneling predictions obtained were never reasonable. Because of these problems, we opted for an approximate process to obtain a variational transition state / small-curvature tunneling correction for the RRKM-ME rates.

The strategy employed was to model the non-canonical energy distributions in the various **INT** structures using canonical energy distributions having the same *average* energy. One can then use the POLYRATE results to calculate the CVT/SCT corrections for the canonical energy distributions and apply these same corrections to the corresponding non-canonical distributions. This is clearly an imperfect process, but error contributed to the final results should mainly arise from error in the variation of corrections from alkene to alkene, and that variation, below, is small.

We first calculated the average energy in the INT structures at a series of temperatures, by a process described in a later section. The table below shows these energies, which are relative to the 0 K energy and in kcal/mol.

Table. Average energy in **INT** versus 0 K energy for a canonical distribution (kcal/mol).

Temperature	propene	butene	hexene	octene	dodecene	t-butylethylene
5	0.0	0.0	0.0	0.0	0.0	0.0
298	2.0	2.7	4.4	6.2	9.6	4.4
350	2.9	4.1	6.0	8.6	12.9	6.3
375	3.4	4.7	7.1	9.5	14.5	7.0
400	3.9	5.3	8.0	10.8	16.9	8.3
425	4.6	6.0	9.2	12.2	18.7	9.5
450	5.1	6.6	10.6	13.7	20.7	10.4
475	5.6	7.5	11.3	15.6	23.3	11.9
500	6.4	8.5	12.9	16.5	25.3	12.9
525	7.0	9.5	13.9	18.9	27.6	14.4
550	7.6	10.4	15.7	20.1	29.9	15.7
575	8.4	11.3	17.1	22.5	33.1	17.0
600	9.1	12.6	18.4	24.1	36.1	18.9
625	10.0	13.5	20.1	26.2	39.2	19.8
650	10.8	14.5	20.7	28.3	41.9	21.8
675	12.0	15.6	22.9	29.9	44.5	23.5

We then calculated CVT/SCT corrections for the propene reaction at each temperature. These are shown in the next table. It should be noted that the corrections are based on the POLYRATE rate constant versus the conventional TST rate constant with CCSD(T)/aug-cc-pvtz energetics at the B3LYP saddle point (since this is what is used in the RRKM-ME calculations to be corrected).

Table. VTST-ISPE CVT/SCT corrections versus conventional TST (CCSD(T)/aug-cc-pvtz//B3LYP/6-31G*) at various temperatures.

Temperature	CVT/TST correction **A**	CVT/TST correction **M**	ratio of corrections
195.15	0.99	2.50	2.51
273.15	0.99	1.56	1.57
298.15	0.99	1.45	1.46
344.15	0.99	1.31	1.33
400	0.98	1.22	1.24
500	0.96	1.13	1.17
550	0.95	1.10	1.16
600	0.94	1.07	1.15
650	0.93	1.06	1.14
700	0.92	1.04	1.14
750	0.90	1.03	1.13

If there were no energy loss from the intermediate, then the average energy above the 0 K energy for propene, butene, hexene, octene, dodecene, and *tert*-butylethylene would be 12.4, 13.6, 15.3, 17.1, 20.5, and 14.7 kcal/mol, respectively. (These versus separate starting materials, not the VTS (see discussion above) and an extra RT for the molecules coming together). From the average energy lost in an initial set of RRKM-ME calculations and the two tables, the corrections in the following table were chosen. It should be noted that the corrections for the localized RRKM-ME calculations for larger alkenes were simply set off of those for propene. Once calculated, these corrections were used consistently; it might be argued that they should be changed with each change in, for example, E_{down}, but small changes should only impact the specific value of E_{down} in its parameterization and not significantly affect the important trends in the predictions.

Table. Variational Transition State / Tunneling Corrections Employed for RRKM-ME calculations

alkene	full RRKM-ME correction	localized RRKM-ME correction
propene	1.14	1.14
propene 70 °C	1.13	1.13
butene	1.15	1.14
hexene	1.16	1.14
octene	1.17	1.14
dodecene	1.21	1.14
t-butylethylene	1.17	1.14

The CLN Calculation

The table below shows the numerical details of the CLN calculation. The weighting of the direct versus indirect mechanisms was taken from the CCNM calculations. The % **M** for the direct mechanism was taken from a localized RRKM-ME calculation (see the discussion above and the example input file in a later section) with the E_{down} set negligibly low. The % **M** for the indirect mechanism was taken from a full RRKM-ME calculations (using all of the molecular frequencies) using an E_{down} of 726.1234 cm^{-1} and a solvent "pressure" of 2093950 Torr (1955440 for THF at 70 °C). These pressures are a factor of 10 greater than the normal pressure; the cooling was increased by increasing the pressure instead of the E_{down} due to an apparent numerical issue in Mesmer that limits immediate cooling of the solute by the simple increase in $E_{down.}$ (This is technically two parameters but adjustments in the two are normally interchangeable, and the same set of results can be obtained with a higher pressure and a lower E_{down}. It should be noted that the cooling rate is too high to be physical. In the CLN model the fast-reacting trajectories should correspond to the direct mechanism while the indirect mechanism should correspond to slow-reacting trajectories, but the nature of the RRKM-ME calculation is that it starts producing product immediately with no cooling. A more realistic calculation might take the tact of only counting RRKM product that was obtained after some arbitrary time period, such as 100 fs. Our goal at this stage was to maintain the simplicity of a single cooling rate with no other unnecessary other parameters.

Table. Numerical Details of the CLN calculation

	weighting		localized RRKM (direct)			RRKM-ME (indirect)			% **M**
	direct	indirect	raw	correction	% **M**	raw	correction	% **M**	weighted
propene	0.33	0.67	12.7%	1.14	14.2%	7.0%	1.14	7.9%	10.0%
propene 70°C	0.34	0.66	13.2%	1.13	14.7%	8.5%	1.13	9.5%	11.3%
butene out	0.65	0.35	13.4%	1.14	15.0%	6.3%	1.15	7.2%	12.3%

butene anti	0.41	0.59	15.5%	1.14	17.2%	7.9%	1.15	8.9%	12.3%
hexene out	0.62	0.38	14.1%	1.14	15.7%	5.0%	1.16	5.7%	11.9%
hexene anti	0.39	0.61	15.7%	1.14	17.5%	6.3%	1.16	7.2%	11.2%
octene out	0.62	0.38	14.0%	1.14	15.6%	3.2%	1.17	3.7%	11.1%
octene anti	0.39	0.61	15.5%	1.14	17.3%	5.2%	1.17	6.0%	10.4%
dodecene out	0.62	0.38	13.8%	1.14	15.5%	2.9%	1.21	3.5%	10.9%
dodecene anti	0.39	0.61	15.4%	1.14	17.2%	3.9%	1.21	4.7%	9.6%
tert-butylethylene	0.47	0.53	15.6%	1.14	17.4%	5.0%	1.17	5.8%	11.3%

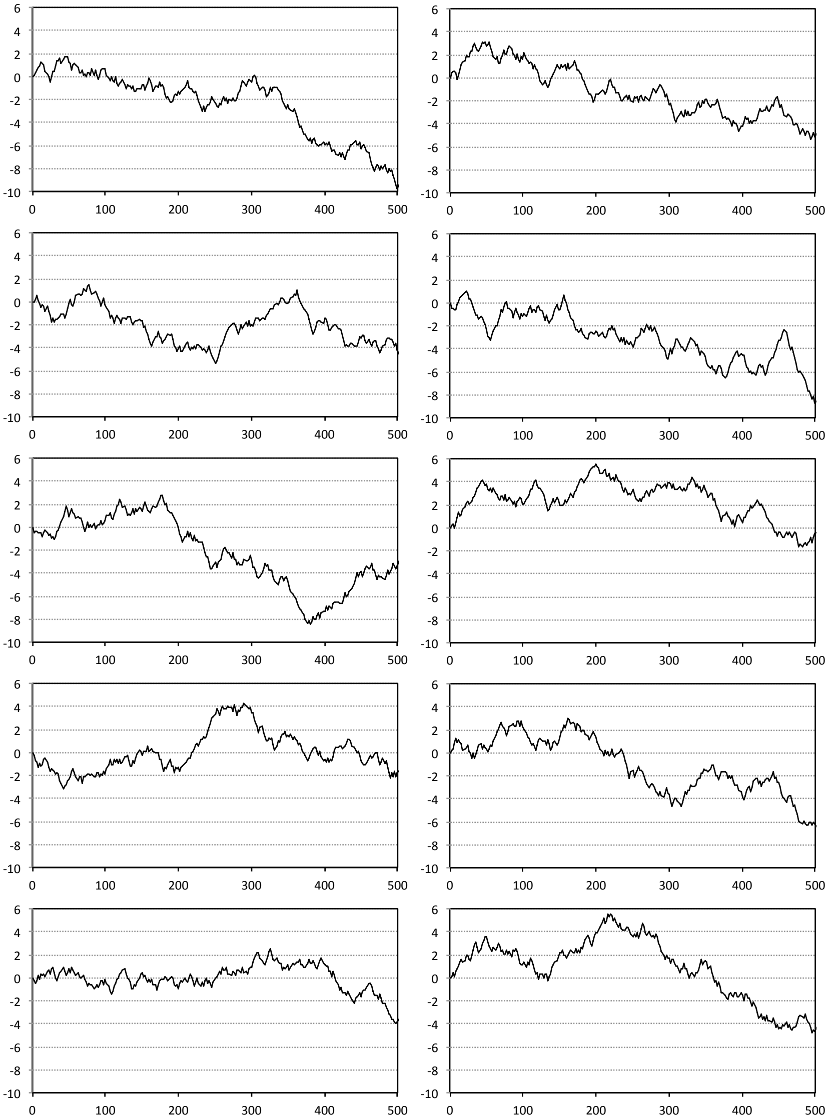
The Parameters for Figure 4b, and Additional Example of Random Walks

The comparison between observed data and a biased random walk only becomes meaningful when the random walk is parameterized to approximate the overall features of the observed system. As noted in the main text, the standard deviation on $\Delta E_{alkene+BH3}$ was 2.9 kcal/mol after 180 fs. In the equation:

$E(t+2fs)=E(t)+0.54*\textit{rand}-0.016$

the 0.54 value was chosen so as to produce a standard deviation after 180 fs of around 2.9 kcal/mol. For comparison with graphs of trajectories that last ~500 fs, the bias was set at -0.016 so as to produce the observed average energy loss of 4 kcal/mol.

The graphs below show additional examples of random walks produced from this formula. Visually, the graphs seem modestly less structured that the graphs of $E_{alkene+BH3}$ versus time from the trajectories given in the main text. The random walks of course have a zero autocorrelation time, while the autocorrelation times for the trajectories (obtained by calculating the fs by fs change in $E_{alkene+BH3}$ after subtracting out the overall change, then examining the correlation of the signs of the points at 1 fs, 2 fs, 3 fs, 4 fs, 5 fs, 6 fs, and 7 fs intervals) is greater than zero but less than 3 fs, allowing for a slightly greater structure in the trajectories.



The Classical and Harmonic Quantum-Mechanical Variation in Total Molecular Energies

We expect that these problems have been solved analytically previously in the literature, but it proved faster to solve the problems approximately from scratch than to find the literature analytical solutions.

For a classical degree of freedom, the distribution of energies in a single mode can be represented by E = -RT ln (random number from 0 to 1). We set up an Excel® spreadsheet with a table of 3N-6 normal modes with an energy defined by this formula, and replicated the set of modes 445 times, representing up to 50,730 randomized modes at once. For each set of modes, the total energy was found, and an average and standard deviation were found. This process was repeated 19 times, and the main text reports the average results from the 19 x 445 trials as 3.4, 4.6, and 6.6 kcal/mol, for the propene, hexene, and dodecene systems, respectively. The distribution of the results from the 19 trials allowed the calculation of a 95% confidence range, which is 0.06, 0.11, and 0.11 for the propene, hexene, and dodecene systems, respectively.

The harmonic quantum-mechanical distribution was calculated with the aid of the programs *proggenHP* and *randgen*, given in a later section, along with snippets of code taken from *progdynstarterHP*. Using a Gaussian09 frequency calculation (done with freq=hpmodes) for each of the **INT** structures for propene, hexene, and dodecene, this code was used to pull out appropriate data for proggenHP to run:

```
    awk '/       1         2/,/Harmonic frequencies/ {print}' $freqfile > temp401
    awk '/Frequencies --/ {print $3;print $4;print $5;print $6;print $7}' temp401 > tempfreqs
    awk '/Reduced masses/ {print $4;print $5;print $6;print $7;print $8}' temp401 > tempredmass
    awk '/Force constants/ {print $4;print $5;print $6;print $7;print $8}' temp401 > tempfrc
    awk '/0/ && ((length($1) < 2) && ($1 < 4)) {print}' temp401 > tempmodes
    awk '/has atomic number/ {print}' $freqfile > tempmasses
    awk '/Standard orientation:/,/tional const/ {if (($3=="0") || (substr($3,1,2)==10)) print}' $freqfile > tempstangeos
    awk '/Input orientation:/,/Stoichiometry/ {if (($3=="0") || (substr($3,1,2)==10)) print}' $freqfile > tempinputgeos
```

where $freqfile is the Gaussian09 output.

We then ran this code on an infinite loop:

```
randgen > temp811
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e298
sed -i 's/temperature 298/temperature 350/' progdyn.conf
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e350
sed -i 's/temperature 350/temperature 375/' progdyn.conf
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e375
sed -i 's/temperature 375/temperature 400/' progdyn.conf
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e400
sed -i 's/temperature 400/temperature 425/' progdyn.conf
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e425
sed -i 's/temperature 425/temperature 450/' progdyn.conf
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e450
sed -i 's/temperature 450/temperature 475/' progdyn.conf
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e475
sed -i 's/temperature 475/temperature 500/' progdyn.conf
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e500
sed -i 's/temperature 500/temperature 525/' progdyn.conf
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e525
sed -i 's/temperature 525/temperature 550/' progdyn.conf
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e550
sed -i 's/temperature 550/temperature 575/' progdyn.conf
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e575
sed -i 's/temperature 575/temperature 600/' progdyn.conf
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e600
sed -i 's/temperature 600/temperature 625/' progdyn.conf
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e625
sed -i 's/temperature 625/temperature 650/' progdyn.conf
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e650
sed -i 's/temperature 650/temperature 675/' progdyn.conf
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e675
sed -i 's/temperature 675/temperature 5/' progdyn.conf
awk -f proggenHP $freqfile | grep 'Total' | awk '{print $5-124.349066}' >> e005
sed -i 's/temperature 5/temperature 298/' progdyn.conf
sleep 1
```

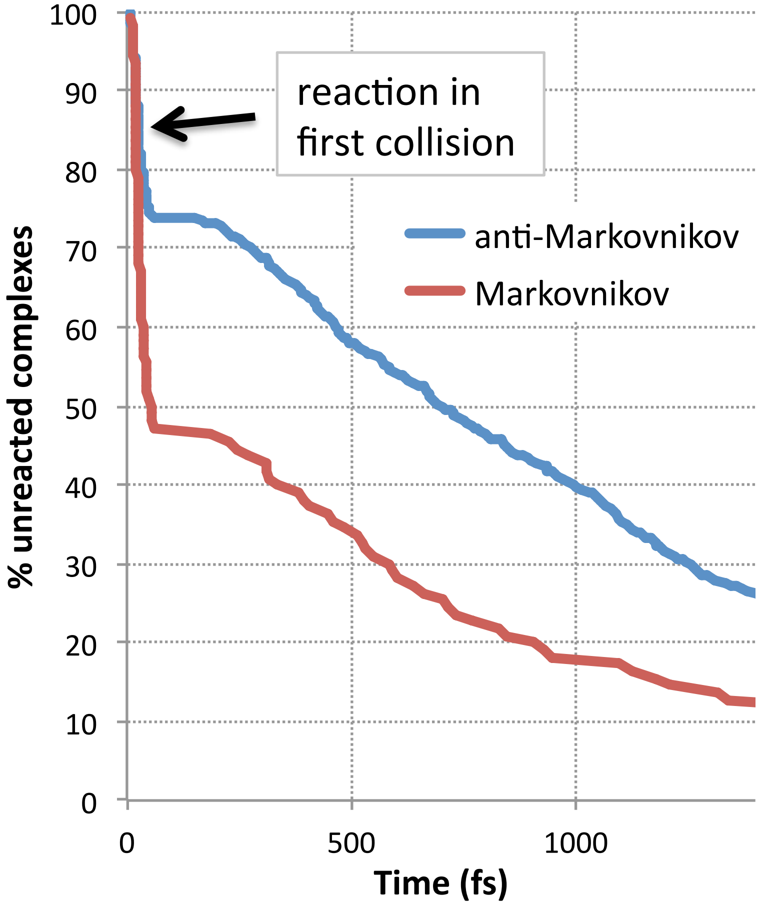
where progdyn.conf in an ordinary PROGDYN configuration file (an example is given in a later section) where the only important parameters set the calculation to be quasiclassical and the temperature to be initially 298.

The effect of this calculation is to give files named e??? with lists of total molecular energies at a series of temperatures. The "124.349006" is just the zero-point energy for the hexene system employed; this number was changed for each system. The code is written this way because the random numbers are seeded by the clock with the seed changing once per second.

From the lists of energies, averages and standard deviations were calculated. As noted in the main text, for the propene, 1-hexene, and 1-dodecene systems, the standard deviations at 25 °C would be 1.6, 2.3, and 3.2 kcal/mol, respectively The averages at each temperature were also used is the analysis of variational and tunneling effects on the rate constants in the RRKM calculations.

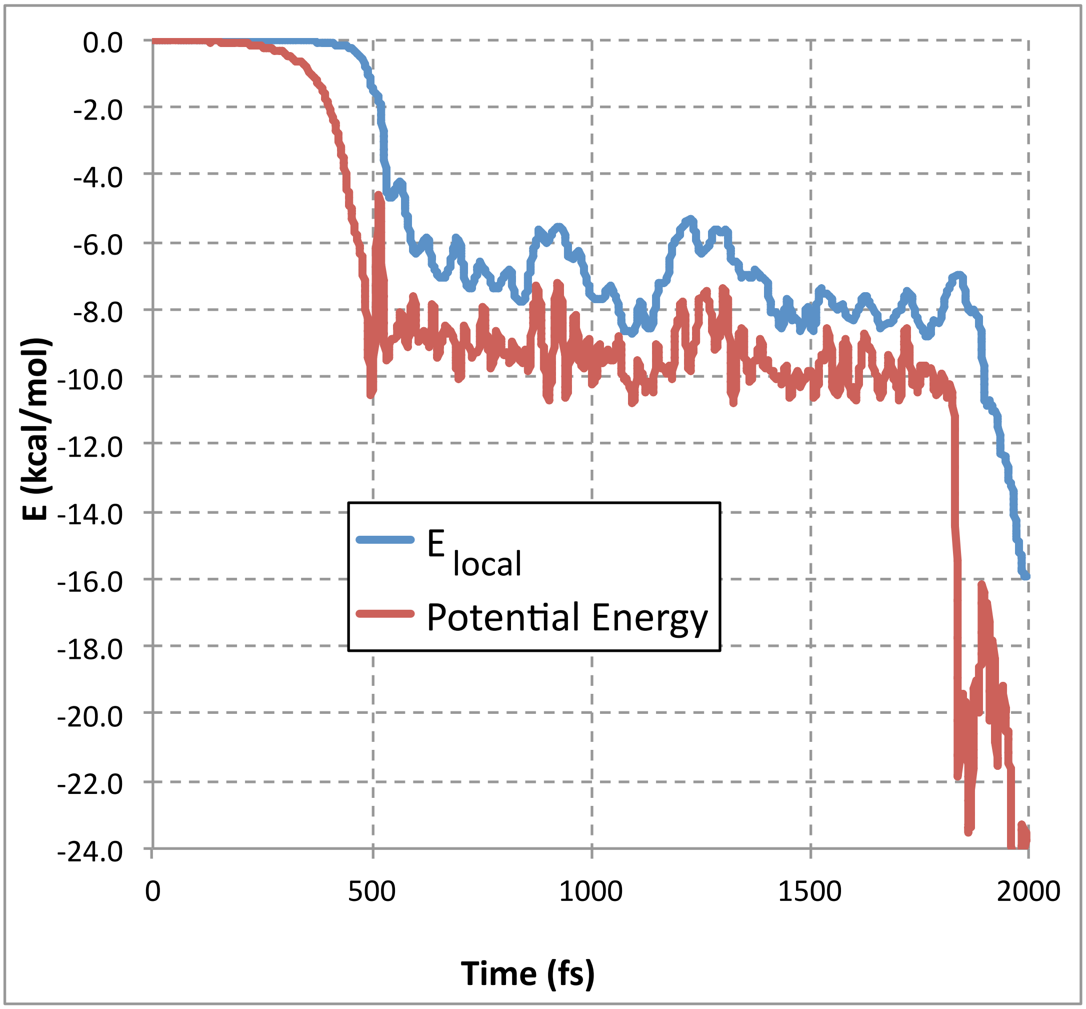
Gas-Phase Analog of Figure 3.

Reference 4 of the main text included extensive gas-phase trajectories started from the VTS for the reaction of BH_3 with propene. These trajectories were fully classical, and they were initiated by a normal rare-event process where the VTS normal modes were given a classical Boltzmann-random energy. In light of what was seen in Figure 3, we have reanalyzed those trajectories to explore their difference versus the solution trajectories. The results are shown in the figure below. As discussed in the main text, the gas phase trajectories show a similar burst to those seen in solution. There is a somewhat greater proportion of the initial collisions affording the Markovnikov product in the first collision, but the major difference is that many of the trajectories have the BH_3 rebound further from the alkene. As a result, fewer BH_3 molecules quickly reapproach the reactive area and 37% of the trajectories remain unreactive after 1000 fs.



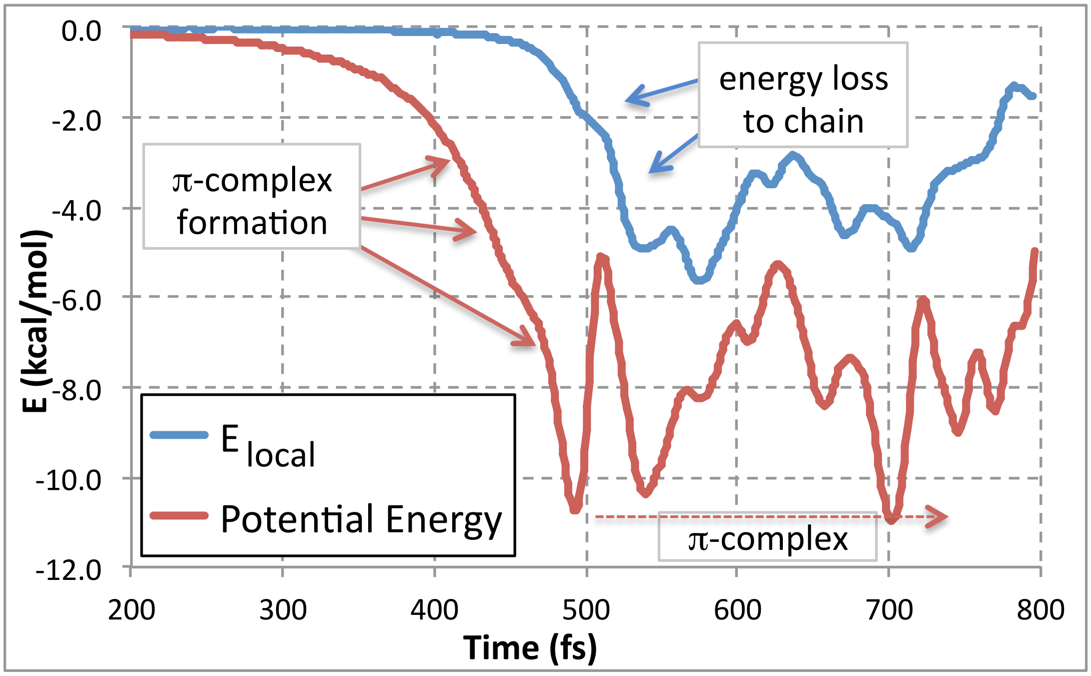
Extended Version of Figure 5a

The figure below shows a time-extended version of Figure 5a in the main text. The drop in energy at about 1800 fs corresponds to the formation of the **A** product. The equilibrium amount of energy in the chain, classically, is 8.3 kcal/mol, and it can be seen that this is reached briefly at ~1100 fs then consistently at ~1500 fs. The 8.3 value comes from the total energy available, 11.8 kcal/mol, and the number of atoms in the chain (28) versus the total atoms (40).



1-Hexene Analog of Figure 5a

The figure below shows the energy analysis of the single-trajectory study of the reaction of BH_3 with 1-hexene with the *anti* mode of approach, as an analog of Figure 5a (done with 1-dodecene) in the main text. With this smaller alkene, the E_{local} does not drop as far as with 1-dodecene, but it still drops to below -4 kcal/mol in less that 100 fs after π-complex formation.



Details of the Energy Analysis

The awk program *progKECM*, listed in a later section, was used to obtain kinetic energies for selected groups of atoms in trajectories based on the *traj* file output from either PROGDYN or ProgdynONIOM. The *progKECM* was modified from case to case depending on which alkene was used and whether $E_{alkene+BH3}$, E_{local}, or E_{chain} was desired.

For the potential energies for the $E_{alkene+BH3}$ calculations, the structures for the alkene / BH_3 system were extracted from the *traj* files and recalculated in B3LYP/6-31G* calculations in a straightforward manner.

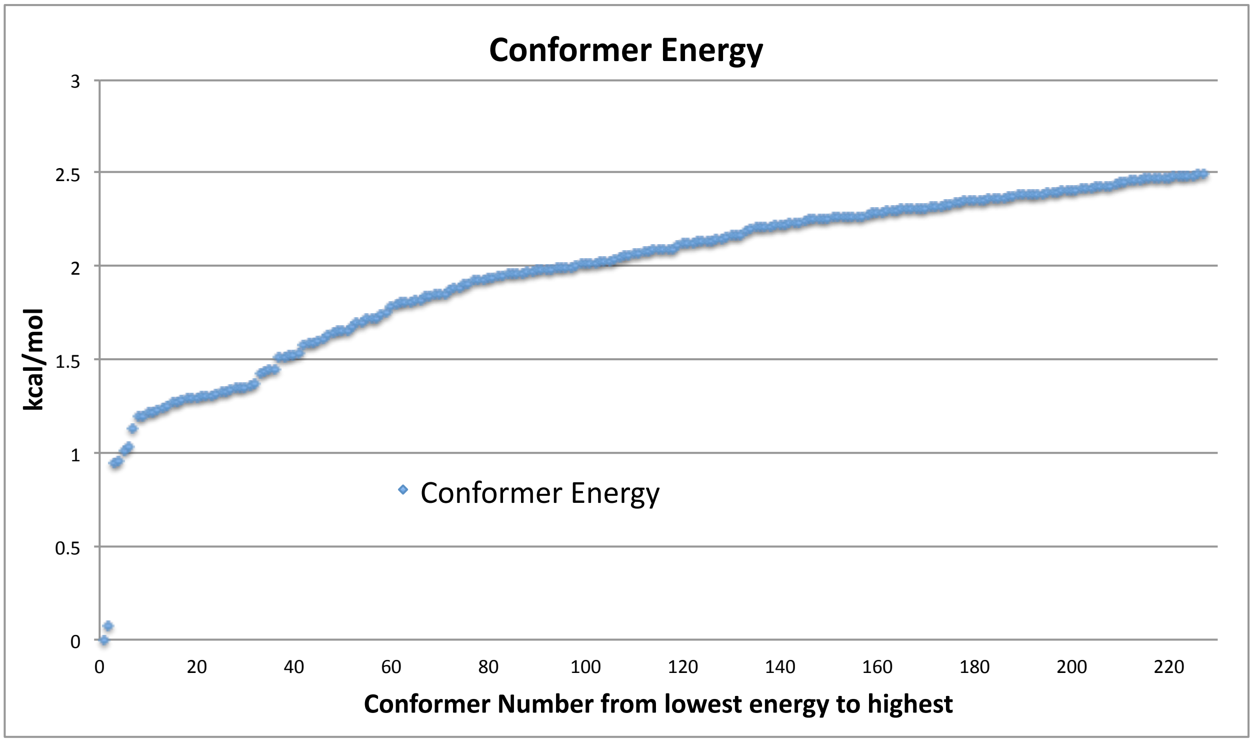
For the E_{local} and E_{chain} calculations, the complication was that a C-C bond was replaced by a C-H bond for the potential energy calculation. In these cases, the programs progEchangelocal and progEchangetail, given in a later section, were used to generate Gaussion09 input files for the calculation. As described in the main text, the unwanted chain was replaced by a hydrogen atom with a C-H distance of 1.09 Å. This has the effect of losing any potential energy associated with the C-C stretch but it retains bending strain.

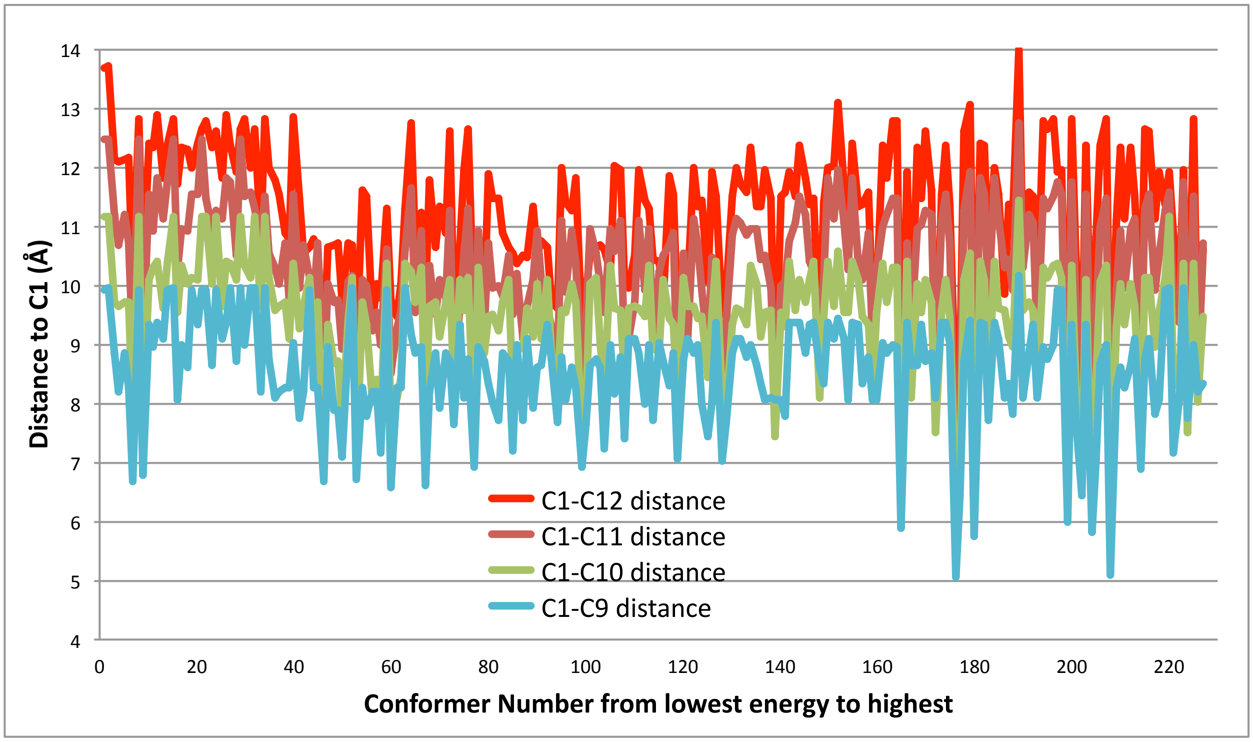
Conformation Analysis and the Role of Alternative Conformations of Alkenes

The analysis in the main text assumes that the alkyl chains are in an extended *anti* conformation. We consider here whether the contribution of reactions of alternative conformations could account for differences in the reactions of the alkenes.

Alternative conformations could in principle affect the product distribution in two ways, by a steric effect and by an electronic effect. A steric effect could come about if the end of the alkyl chain of dodecene were to be close enough to the olefinic carbons to have an influence on the borane addition sterically. To evaluate this issue, we carried out a conformational analysis of 1-dodecene. Using the *Materials Studio* program (Dassault Systèmes BIOVIA, Materials Studio 2017, San Diego: Dassault Systèmes, 2017) and the CompassII force field, we carried out a systematic search for conformations of 1-dodecene by varying the C3-C4, C4-C5, C5-C6, C6-C7, C7-C8, C8-C9, C9-C10, and C10-C11 dihedral angles. The C2-C3 dihedral was not varied since each of the conformations were already considered in the analysis. A total of 227 conformations with energies within 2.5 kcal/mol of the lowest energy were identified. A Boltzmann analysis found that approximately 16% of the 1-dodecene was in an extended *anti* conformation.

The energies of the conformers located were ordered from lowest to highest. The graphs below show how this energy varies versus this ordered conformation number, and the C1-C9, C1-C10, C1-C11, and C1-C12 distances along the ordered list. The key observation is that there are very few conformations in which the tail of the 1-dodecene is within 6 Å of C1, and these conformations are all over 2.2 kcal/mol above the lowest energy conformation. No conformations A Boltzmann analysis found that only 0.9% of the conformational population has a tail carbon within 6 Å. From this, it was considered very unlikely that a steric interaction of the tail carbons could significantly impact the reaction.





To explore the potential electronic effect of alkyl chain conformations, we systematically explored the anti-Markovnikov and Markovnikov transition structures for conformations of 1-hexene twisted about the C4-C5 bond. Within the *anti* series of structures, the calculated difference in free energy between the anti-Markovnikov and Markovnikov transition structures was 2.69, 2.68, and 2.75 kcal/mol for the three possibilities. For the *out* series of structures, the calculated difference in free energy between the anti-Markovnikov and Markovnikov transition structures was 2.14, 2.18, and 2.14 kcal/mol for the three possibilities. These results show that the C4-C5 bond conformation, common to all of the alkenes, has a negligible impact on the difference in energy between the competitive transition states. Given this, it seems very unlikely that more distant bonds differing between the alkenes will have a significant electronic effect on the reaction.

Programs for Calculations, and Sample Input Files

Program Suite PROGDYN

A full listing of the subprograms of PROGDYN is given below. To allow the reader to understand or make use of PROGDYN, we describe here first the overall structure of the program. We also list and describe in this section a number of helper programs that were used to analyze the data from the trajectory calculations.

The newest edition of PROGDYN includes the ability to include additional empirical dispersion using Grimme's D2 model (reference 22 in the main text). This was not employed for the 53-THF system but it was employed in the 80-THF system using ProdynONIOM, described in the next section.

The master control program for dynamics, in the form of a Unix Shell Script, is called *progdynstarterHP*. For a user to start to use *progdynstarterHP*, some early lines in it that assign the scratch space and the location of the program files and input files would have to be modified for the local environment. These lines are between lines 45 and 55 and should be apparent. The location of the scratch space is usually passed to *progdynstarterHP* as a parameter.

progdynstarterHP takes as input files:

freqinHP - This is the standard output from a Gaussian 98, 03, or 09 frequency calculation using freq=hpmodes. For isotopically labeled compounds, use freq=(hpmodes,readisotopes).

progdyn.conf – This is a file giving a variety of configuration options, called on by many of the subprograms. progdyn.conf examples are listed below and contains explanations of the program options.

isomernumber - A number in file isomernumber provides a start for numbering runs. The default is 1.

detour – A signal file that, by existing, signals the program to do a side calculations

nogo – A signal file that, by existing, signals the program to stop between points

bypassproggen - A signal file that, by existing, signals the program to use a supplied input file *geoPlusVel* instead of generating one for itself. This pathway for initialization is important here because it is used when the program progdynsam, described later, is used to generate the geoPlusVel file.

methodfile – A file that contains lines to be added to the end of each g09.com input file, such as lines that call for an NMR calculation

ZMAT – An input file for the CFOUR (http://www.cfour.de) suite of programs. When ZMAT is supplied, *progdynstarterHP* will automatically run call CFOUR (which must be set up independently by the user) by making use of the script *progcfour*.

cannontraj – A file containing a vector for each atom, used to fire an initial geometry in a particular direction.

progdynstarterHP calls the following programs:

proggenHP - An awk program that starts a trajectory, giving each mode its zero point energy (if a quasiclassical calculation) plus random additional excitations depending on the temperature.

prog1stpoint – Awk program that creates the first Gaussian input file for each run

prog2ndpoint – Awk program that creates the second Gaussian input file for each run. prog2ndpoint also checks the energy of the first point to see if it fits with the desired energy, and aborts the run if it does not by creating appropriate output in file Echeck

progdynb – Creates subsequent Gaussian input files until run is completed, written in awk

proganal – A program to analyze the latest point and see if a run is done. This program must be redone for each new system. Elaborate changes are sometimes programmed into proganal, such as the automatic changing of configuration variables. proganal creates the output to dynfollowfile and NMRlist or NMRlistdis

randgen – A program that generates random numbers between 0 and 1. These are generated all at once and stored in a file for use by proggenHP.

progcfour – A control script to run CFOUR calculations (not needed for most kinds of runs).

progdynstarterHP has the following output files:

isomernumber – A running tab of the trajectory number

runpointnumber – a running tab of the point in the trajectory

Echeck – output form where prog2ndpoint checks the energy of the trajectory to see if it fits with the desired energy

geoRecord – A record of all of the geoPlusVel files.

geoPlusVel – Created by proggen, this gives the starting positions, velocities, isotopic masses, excitations of the normal modes, and initial displacements of the normal modes for current run.

g09.com – Created by prog1stpoint, prog2ndpoint, and progdynb, this is the latest input file for Gaussian09 for current run and latest point.

olddynrun, *olddynrun2, olddynrun3* – files containing the last three outputs from Gaussian, for creation of the next point

traj, traj1, traj2, traj3, etc. – files containing the geometries and energies for each trajectory, numbered by the isomernumber, in a format suitable for reading by Molden.

dynfollowfile – A short record of the runs and their results. The data desired for *dynfollowfile* must be programed into the script *proganal* as needed for each system studied.

NMRlist or *NMRlistdis* – output of NMR predictions at each point in a trajectory, when desired

skipstart - A signal file that, by existing, tells progdynstarterHP that we are in the middle of a run. For trajectories that are propagated forward and backward in time, skipstart keeps track of whether one is in the forward or reverse part.

diagnostics – optional output that follows which subprograms are running and configuration variables, decided by variable in progdyn.conf

vellist – optional output that lists the velocities of each atom, decided by variable in progdyn.conf, or lists the total kinetic energy in the system and the classical temperature, often also keeps track of the density

A number of files starting with '*temp*' are created then later erased.

The following helper programs were used for the current study.

progdynsam – an awk program that generates a geoPlusVel file based on input from a *traj* file. Typically, the traj file is a trajectory performed with constraints, and the new geoPlusVel will be started without constraints. *progdynsam* must be modified for the desired temperature, and it takes as input the variable pt to decide which points in the traj file are used to define the geoPlusVel. For previously fully fixed atoms, *progdynsam* gives the atoms a Boltzmann-random velocity and direction of motion appropriate for the desired temperature. Usually any prior constraints were harmonic, in which case adjustment was made to the motions of the constrained atoms. The program is invoked with awk –v pt=## –f progdynsam traj where ## is the trajectory point used to start a new geoPlusVel, and traj is the output file from above containing the list of trajectory points.

proglookstart – an awk program used to generate the proper inputs for progdynsam.

progKECM – an awk program that calculates the kinetic energy of the alkene from the traj file while removing the motion of the center of mass. Other versions of the program were implemented to account for different alkenes as well as remove the center of mass calculation. Other kinetic energy programs calculated the energy for different portions of an alkene.

progcomptime – an awk program used to analyze data from the dynfollowfile. It calculated the time from the formation of the pi complex to the formation of the product.

progdynstarterHP

```
#!/bin/bash
#progdynstarterHP, made to use high-precision modes from Gaussian output with freq=hpmodes
#updated to create a random number file temp811 that is used by proggenHP
#version September 16, 2005, made for workstations
#version August 2007 to allow periodic copying of g09.log to dyn putting it under control of progdynb
#version Feb 2008 moves variables like the scratch directory and location of randgen to the beginning
#version March 2008 added proganal reporting to points 1 and 2
#version Jan 2009 fixed bug generator of having proganal run twice in checking for complete runs
#version May 2009 Echeck catches bad energies after only one point, other lines written simpler, triple while loop, revised comments
#version Aug 2010 isomernumber adds words to ease parsing, increased elements up to bromine, runpointnumber checked for more appropriate restarts
#version Aug 2011 runpointnumber starts better, restart better if died during first few points, awk bug fix
#version Aug 2012 freqinHP reads with only 3 freqs, goingwell and other temp files moved to $scratchdir
#version Aug 2013 adds ability to automatically run a CFOUR program if the file ZMAT exists
#version Nov 2013 adds ability to bypass generation of geoPlusVel using the signal file bypassproggen
#version Nov 2015 makes using guess=read easier, improves ability to restart after disk write failures, partially allows for MM runs in Gaussian
#LIMITATIONS - standard version only handles elements up to bromine, must change program to do higher atomic numbers
#   only handles up to 4000th excited state for modes - this could start to affect the initialization of classical modes or transition vectors at
#    extremely high temperatures
#   The routine that checks whether the actual energy approximately equals the desired energy checks for lines containing "SCF
Done" or "EUMP2 =" or " Energy="
#   This should handle ordinary calculations HF, DFT, ONIOM, and MP2 calculations but the routine in prog2ndpoint would
have to be changed for other calcs.
#
#                                OUTLINE
# A. initialize to perform Gaussian jobs, set the scratch, program, and other directories, remove errant control files
#    start outermost loop L1L1L1L1L1L1L1L1
#    start loop 2 L2L2L2L2L2L2L2L2L2
# B. branch on whether there is a file named "skipstart"
#    if there is, skip B1, B2, B3 entirely
#    if no file named "skipstart" then generate a new isomer.  Instructions: Get rid of skipstart to start new isomer.
#    the B loop generates geoPlusVel, adds it to geoRecord, generates and runs first and second points, and sets up for continuous
loop
#  B1. generates geoPlusVel, isomernumber, runpointnumber=1, then makes g09.com for point 1
#  B2. checks for success of B1 or exits, then sets geoRecord then runs g09.com.  Checks for its success or exits.
#  B3. runpointnumber=2, kills run if XXXX in dynfollowfile due to bad energy probably, makes g09.com for point 2, runs it
#      checks for its success or exits
#      runpointnumber=3, uses progdynb to make g09.com for point 3
#      sets skipstart=forward
#  B4. Only runs if skipstart=ireverserestart, performs reverserestart, ends with runpointnumber=3
#      sets skipstart=reverse
# C. loop over propagation steps
#
#
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
#
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
#
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
#origdir, randdir, scratchdir, g09root, logfile all may need varied from system to system and assigned here or by program calling
this one
export LC_ALL=C
echo $1
scratchdir=$1
export g09root=/software/lms/g09_D01
. $g09root/g09/bsd/g09.profile
origdir=`pwd`
cd $origdir
logfile=docslog
randdir=~/bin
proggramdir=/scratch/user/d-singleton/binall500
freqfile=/scratch/user/d-singleton/binall500/freqinHP
echo
echo ORIGDIR at the beginning of run:
echo $origdir
ls $origdir
echo
echo SCRATCHDIR at the beginning of run:
echo $scratchdir
ls $scratchdir
echo
echo PROGGRAMDIR at the beginning of run::
echo $proggramdir
ls $proggramdir
```

```
rm -f nogo    # assume that if someone is starting a job, they want it to go.
rm -f diagnostics goingwell tempdone # diagnostics contains extra info from previous runs, other two files are from older
versions of progdyn
if (test -s g09.com) then
  sed -i '/guess=tcheck/d' g09.com      # no chk file on first point
fi

#### Triple 'while' loop - will have to break multiple times to get out, but advantage is ability to control starting over
#L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1
L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1
while (true)
do
# As long as there is a file "goingwell" the program will not exit entirely by itself
rm -f $scratchdir/goingwell
#L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2
L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2
while (true)
do
#  BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
  if (test -f skipstart) then
    echo "skipping start and continuing from previous runs"
  else
#  B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1 generate geoPlusVel and first input file
    if [ `cat runpointnumber` = "1" ]; then
      echo "X did not complete first point so new isomer started" >> dynfollowfile
    fi
    if [ `cat runpointnumber` = "2" ]; then
      echo "X did not complete second point so new isomer started" >> dynfollowfile
    fi
    if [ `cat runpointnumber` = "3" ]; then
      echo "X did not complete third point so new isomer started" >> dynfollowfile
    fi
    cd $origdir
    if (test -f bypassproggen) then
      echo "taking starting conditions from pre-generated geoPlusVel"
    else
      $randdir/randgen > temp811
# the next 8 lines would have to be changed to use low-precision modes
      awk '/        1         2/,/Harmonic frequencies/ {print}' $freqfile > temp401
      awk '/Frequencies --/ {print $3;print $4;print $5;print $6;print $7}' temp401 > tempfreqs
      awk '/Reduced masses/ {print $4;print $5;print $6;print $7;print $8}' temp401 > tempredmass
      awk '/Force constants/ {print $4;print $5;print $6;print $7;print $8}' temp401 > tempfrc
      awk '/0/ && ((length($1) < 2) && ($1 < 4)) {print}' temp401 > tempmodes
      awk '/has atomic number/ {print}' $freqfile > tempmasses
      awk '/Standard orientation:/,/tional const/ {if (($3=="0") || (substr($3,1,2)==10)) print}' $freqfile > tempstangeos
      awk '/Input orientation:/,/Stoichiometry/ {if (($3=="0") || (substr($3,1,2)==10)) print}' $freqfile > tempinputgeos
      awk -f $proggramdir/proggenHP $freqfile > geoPlusVel
    fi
    if (test -f isomernumber) then
      cp isomernumber temp533
      awk 'BEGIN {getline;i=$1+1;print i,"----trajectory isomer number----"}' temp533 > isomernumber
      rm temp533
    else
      echo "1 ----trajectory isomer number----" > isomernumber
    fi
    echo 1 > runpointnumber
    rm g09.com
    awk -f $proggramdir/prog1stpoint isomernumber > g09.com
#  B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2  if first part successful then clean
up and run the first input file, otherwise die
    if (test -s g09.com) then
      rm tempfreqs tempredmass tempfrc tempmodes tempstangeos tempmasses temp401 temp811 tempinputgeos
      cat isomernumber >> geoRecord
      cat geoPlusVel >> geoRecord
      rm -f $scratchdir/goingwell
      cd $scratchdir
      cp $origdir/g09.com $scratchdir/g09.com
      $g09root/g09/g09 $scratchdir/g09.com > $scratchdir/g09.log
      cd $origdir
      grep 'Normal termination' $scratchdir/g09.log > $scratchdir/goingwell
      if (test -s $scratchdir/goingwell) then
        cp $scratchdir/g09.log olddynrun2
      else
        cp $scratchdir/g09.log $origdir/g09.log
        break
      fi
    else
      break
    fi
#  B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3 if B2 worked then you are here.
create 2nd point, run it, and set up for propagation loop
    rm g09.com
    echo 2 > runpointnumber
    awk -f $proggramdir/prog2ndpoint $scratchdir/g09.log > g09.com
# before we decide to run this, check the energy
    awk -f $proggramdir/proganal $scratchdir/g09.log >> dynfollowfile
    rm -f $scratchdir/tempdone
    tail -1 dynfollowfile | awk '/XXXX/ {print}' > $scratchdir/tempdone
    if (test  -s $scratchdir/tempdone) then
      rm -f dyn
      rm -f traj
      echo 0 > runpointnumber
      break
    fi
    if (test -s g09.com) then
      rm -f $scratchdir/goingwell
      cd $scratchdir
      cp $origdir/g09.com $scratchdir/g09.com
      $g09root/g09/g09 $scratchdir/g09.com > $scratchdir/g09.log
      cd $origdir
      grep 'Normal termination' $scratchdir/g09.log > $scratchdir/goingwell
      if (test -s $scratchdir/goingwell) then
        cp $scratchdir/g09.log olddynrun
        awk -f $proggramdir/proganal $scratchdir/g09.log >> dynfollowfile
        awk '/Input orientation/,/Distance matrix/ {print};/Matrix orientation/,/Stoichiometry/ {print}' olddynrun | awk '{if
(($2>.5) && ($2<100)) print}' > old
        awk '/Input orientation/,/Distance matrix/ {print};/Matrix orientation/,/Stoichiometry/ {print}' olddynrun2 | awk '{if
(($2>.5) && ($2<100)) print}' > older
        echo 3 > runpointnumber
        if (test -f bypassproggen) then
          cat bypassproggen > runpointnumber
          echo 3 > bypassproggen
        fi
        awk -f $proggramdir/progdynb olddynrun > g09.com
        rm -f old older
      else
        cp $scratchdir/g09.log $origdir/g09.log
        break
      fi
    else
      break
    fi
# we've just completed a start, so lets skipstart until instructed otherwise
    echo "forward" > skipstart
  fi
#
B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4
B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4
# Reverse trajectories starter routine
  if [ `cat skipstart` = "reverserestart" ]; then
    cd $origdir
    rm g09.com
    echo 1 > runpointnumber
    awk -f $proggramdir/prog1stpoint isomernumber > g09.com
    if (test -s g09.com) then
      rm -f $scratchdir/goingwell
      cd $scratchdir
      cp $origdir/g09.com $scratchdir/g09.com
      $g09root/g09/g09 $scratchdir/g09.com > $scratchdir/g09.log
      cd $origdir
      grep 'Normal termination' $scratchdir/g09.log > $scratchdir/goingwell
      if (test -s $scratchdir/goingwell) then
        cp $scratchdir/g09.log olddynrun2
      else
        cp $scratchdir/g09.log $origdir/g09.log
        break
      fi
    else
      break
    fi
    rm g09.com
    echo 2 > runpointnumber
    awk -f $proggramdir/prog2ndpoint $scratchdir/g09.log > g09.com
    awk -f $proggramdir/proganal $scratchdir/g09.log >> dynfollowfile
    rm -f $scratchdir/tempdone
    if (test -s g09.com) then
      rm -f $scratchdir/goingwell
      cd $scratchdir
      cp $origdir/g09.com $scratchdir/g09.com
      $g09root/g09/g09 $scratchdir/g09.com > $scratchdir/g09.log
      cd $origdir
      grep 'Normal termination' $scratchdir/g09.log > $scratchdir/goingwell
      if (test -s $scratchdir/goingwell) then
        cp $scratchdir/g09.log olddynrun
        awk -f $proggramdir/proganal $scratchdir/g09.log >> dynfollowfile
        awk '/Input orientation/,/Distance matrix/ {print};/Matrix orientation/,/Stoichiometry/ {print}' olddynrun | awk '{if
(($2>.5) && ($2<100)) print}' > old
        awk '/Input orientation/,/Distance matrix/ {print};/Matrix orientation/,/Stoichiometry/ {print}' olddynrun2 | awk '{if
(($2>.5) && ($2<100)) print}' > older
        echo 3 > runpointnumber
        if (test -f bypassproggen) then
          cat bypassproggen > runpointnumber
        fi
        awk -f $proggramdir/progdynb olddynrun > g09.com
        rm -f old older
      else
        cp $scratchdir/g09.log $origdir/g09.log
        break
      fi
    else
      break
    fi
# we've just completed a reversestart, so lets skipstart until instructed otherwise
    echo "reverse" > skipstart
  fi
```

```
#  END_of_B___END_of_B___END_of_B___END_of_B___END_of_B___END_of_B___END_of_B___END_of_B___

#  CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC  propagation loop
  while (true)
  do
    rm -f $scratchdir/goingwell
    cd $scratchdir
    cp $origdir/g09.com $scratchdir/g09.com
    $g09root/g09/g09 $scratchdir/g09.com > $scratchdir/g09.log
    cd $origdir
    grep 'Normal termination' $scratchdir/g09.log > $scratchdir/goingwell
    if (test -s $scratchdir/goingwell) then
      awk -f $proggramdir/proganal $scratchdir/g09.log >> $origdir/dynfollowfile
      mv olddynrun2 olddynrun3
      mv olddynrun olddynrun2
      awk '/Input orientation/,/Distance matrix/ {print};/Matrix orientation/,/Stoichiometry/ {print}' $scratchdir/g09.log | awk '{if
(($2>.5) && ($2<100)) print}' > old
      cp $scratchdir/g09.log olddynrun
      awk '/Input orientation/,/Distance matrix/ {print};/Matrix orientation/,/Stoichiometry/ {print}' olddynrun2 | awk '{if
(($2>.5) && ($2<100)) print}' > older
      #increment runpointnumber
      cp runpointnumber $scratchdir/temp533
      awk 'BEGIN {getline;i=$1+1;print i}' $scratchdir/temp533 > runpointnumber
      rm $scratchdir/temp533
      awk -f $proggramdir/progdynb $scratchdir/g09.log > g09.com
      rm -f old older
    else
      cp $scratchdir/g09.log $origdir/g09.log
      break
    fi
# kludge to do a side calculation of NMR using progcfour.  If ZMAT is there then it gets ran and renamed.
# creation of ZMAT is under the control of progdynb, which is controlled by keyword NMRcc in progdyn.conf
# decisions to be made: erase ZMAT at beginning?  what to do if cfour calc dies?
    if (test  -f ZMAT) then
      cp ZMAT $scratchdir
      cd $scratchdir
      $scratchdir/progcfour $origdir $scratchdir
      cd $origdir
      mv ZMAT temp.ZMAT
      echo "generic one two three" `cat runpointnumber` "runisomer" `cat isomernumber` >> NMRlistcc
      awk '/Nuclear Magnetic Resonance/,/HF-SCF/ {if ($2=="C") print $1,$2,"Isotropic =",$3; if ($2=="H") print
$1,$2,"Isotropic =",$3}' x.log >> NMRlistcc
    fi

# here is a cool link that lets you interrupt the dynamics with a short job, then
# it automatically goes back to the dynamics  just make the file 'detour' and it
# will delete detour, run run.com, then go back to dynamics
    if (test  -f detour) then
      rm detour
      date >> $logfile
      cat run.com >> $logfile
      cp run.log temp.log
      cd $scratchdir
      $g09root/g09/g09 $origdir/run.com > $origdir/run.log
      cd $origdir
    fi

#stop it all nicely by creating a nogo file
    if (test  -f nogo) then
      break
    fi
```

```
#figure out if this isomer is done - change in April 2013 is to move proganal call up from here
    rm -f $scratchdir/tempdone
    tail -2 dynfollowfile | awk '/XXXX/ {print}' > $scratchdir/tempdone
    if (test  -s $scratchdir/tempdone) then
      if [ `awk '/reversetraj/ {if ($1=="reversetraj") print $2}' progdyn.conf` = "true" ]; then
        if [ `cat skipstart` = "reverse" ]; then
          rm -f skipstart
          rm -f geoPlusVel
          rm -f olddynrun
          rm -f olddynrun2
          rm -f olddynrun3
          a=`awk '{print $1}' isomernumber`
          mv traj traj$a
        fi
        if [ `cat skipstart` = "forward" ]; then
          echo reverserestart > skipstart
        fi
      else
        rm -f skipstart
        rm -f geoPlusVel
        rm -f olddynrun
        rm -f olddynrun2
        rm -f olddynrun3
        a=`awk '{print $1}' isomernumber`
        mv traj traj$a
      fi
      break
    fi
  done
#
END_of_C_Loop____END_of_C_Loop____END_of_C_Loop____END_of_C_Loop____END_of_C_Loop____END_of_C_Loop____

# We've got to break a second time to get out of this loop
# if we really want to quit.  Otherwise, it will start over
# at the top
  if (test  -f nogo) then
    break
  fi
  if (test  -s $scratchdir/goingwell) then
    echo "starting a new point or a new direction"
  else
    break
  fi
done

  if (test  -f nogo) then
    break
  fi
  if (test  -s $scratchdir/goingwell) then
    echo "starting a new point or a new direction2"
  else
    break
  fi
done
exit 0
```

proggenHP

```
BEGIN {
# 2014 - avoids bug with a box on, so that starts without modes use input geometry, not standard orientation
# aug 2013 summary of changes
#includes molecular rotation, ability to do multiple NMR calculations, ONIOM with link atoms,
#nonstandard routes, handling of linear molecules using geometry linear, fixed but with atoms over 99 but
#bug varies with version of Gaussian, randomization based on PROCINFO (solved many problems), added initialDiss 3 for
random
#phase of normal modes
# Aug 2010 changes classicalSpacing to 2 and upped possible excited states to 4000
# Jan 2009 - a number of little changes to improve reporting, precision, etc, specification of displacement on particular modes
# Jan 2009 cannonball trajectories.  adds desired energy to initial velocities based on file cannontraj, so one can shoot toward a ts
# updated Nov 2008 to incorporate running DRPs
# updated Nov 2008 to allow for start without an initial freq calc using classical = 2
# updated Aug 2008 added to atom list to handle a large number of atoms without changes needed
# updated June 2008 to incorporate new method for choosing displacements with initialdis 2
# updated Jan 17 2008 - bug fix for > 99 atoms, 300 excitations of low modes possible
# version August 2007 - incorporates classical trajectory calculation option
#also allows listing of number of imaginary frequencies
# version Sept 16, 2005 - incorporates searchdir but not yet rotation
# now reads random numbers from temp811, starting at a random place
# The input files are generated before this and are tempfreqs, tempredmass,
# tempfrc, tempmodes, and tempstangeos.
# It will count the number of atoms.

# default parameters, including quassiclassical, no displacements, transition state, not a DRP
# do not change these - rather, change progdyn.conf to set the parameters
initialDis=0; timestep=1E-15; scaling=1.0; temp=298.15
classical=0; numimag=1; DRP=0; cannonball=0
charge=0; multiplicity=1; method="HF/3-21G"; memory=20000000
diag=1; checkpoint="g09.chk"; searchdir="positive"; boxon=0
boxsize=10; maxAtomMove=0.1; title1="you"; title2="need"
title3="a"; title4="progdyn.conf"; processors=1; highlevel=999
conver1=4.184E26 #dividing by this converts amu angs^2 /s^2 to kcal/mol
geometry="nonlinear";rotationmode=0

#initialization and constants
for (i=1;i<=10000;i++) {disMode[i]=-1}
i=1;j=1;k=1
c=29979245800; h=6.626075E-34; avNum=6.0221415E23
RgasK=0.00198588; RgasJ=8.31447
numAtoms=0; atomnumber=0; classicalSpacing=2
zpeGauss=0; zpeGaussK=0; zpePlusE=0; potentialE=0

# read progdyn.conf for configuration info
blankLineTester=10
while (blankLineTester>1) {
  getline < "progdyn.conf"
  if ($1=="method") method=$2
  if ($1=="charge") charge=$2
  if ($1=="multiplicity") multiplicity=$2
  if ($1=="memory") memory=$2
  if ($1=="processors") processors=$2
  if ($1=="checkpoint") checkpoint=$2
  if ($1=="diagnostics") diag=$2
  if ($1=="initialdis") initialDis=$2
  if ($1=="timestep") timestep=$2
  if ($1=="scaling") scaling=$2
  if ($1=="temperature") temp=$2
  if ($1=="searchdir") searchdir=$2
  if ($1=="classical") classical=$2
  if ($1=="numimag") numimag=$2
  if ($1=="geometry") geometry=$2
  if ($1=="highlevel") highlevel=$2
  if ($1=="boxon") boxon=$2
  if ($1=="boxsize") boxsize=$2
  if ($1=="DRP") DRP=$2; if (DRP==1) classical=2 #this lets one start a DRP from a point that is not a freq calc
  if ($1=="maxAtomMove") maxAtomMove=$2
  if ($1=="cannonball") cannonball=$2
  if ($1=="displacements") disMode[$2]=$3
  if ($1=="controlphase") controlPhase[$2]=$3
  if ($1=="rotationmode") rotationmode=$2
  if ($1=="title") {
    title1=$2
    title2=$3
    title3=$4
    title4=$5
    }
  blankLineTester=length($0)
  }

if (diag>=1) print "***************** starting proggen *****************" >> "diagnostics"
if (diag>=1) print "method,charge,multiplicity,memory" >> "diagnostics"
if (diag>=1) print method,charge,multiplicity,memory >> "diagnostics"
if (diag>=1) print "processors,checkpoint,title,initialdis,timestep,scaling,temperature" >> "diagnostics"
if (diag>=1) print processors,checkpoint,title1,title2,title3,title4,initialDis,timestep,scaling,temp >> "diagnostics"
if (diag>=1) print "classical,numimag,highlevel,boxon,boxsize,DRP,maxAtomMove,cannonball" >> "diagnostics"
if (diag>=1) print classical,numimag,highlevel,boxon,boxsize,DRP,maxAtomMove,cannonball >> "diagnostics"

# put geometries into array, also figure out number of atoms
# note that this picks out the last geometry in a file, assuming
# that if there is an optimization followed by a freq, nothing else follows
# kludgy - repeats last line twice - must be a better way
do {
  getline < "tempstangeos"
  if (oldline==$0) $0=""
  oldline=$0
  atom = $1
  if (atom>numAtoms) numAtoms=atom
  atNum[atom]=$2
  geoArr[atom,1]=$4; geoArr[atom,2]=$5; geoArr[atom,3]=$6
  geoArrOrig[atom,1]=$4; geoArrOrig[atom,2]=$5; geoArrOrig[atom,3]=$6
  velArr[atom,1]=0; velArr[atom,2]=0; velArr[atom,3]=0
  }
while (length($0) > 0)

#output the number of atoms, used in many routines
print numAtoms

# put in atomic symbols and atomic weights - assigns a default mass but then reads it from tempmasses when possible
for (i=1;i<=numAtoms;i++) {
  getline < "tempmasses"
  if (atNum[i]==1) {atSym[i]="H";atWeight[i]=1.00783}
  if (atNum[i]==2) {atSym[i]="He";atWeight[i]=4.0026}
  if (atNum[i]==3) {atSym[i]="Li";atWeight[i]=6.941}
  if (atNum[i]==4) {atSym[i]="Be";atWeight[i]=9.012}
  if (atNum[i]==5) {atSym[i]="B";atWeight[i]=10.811}
  if (atNum[i]==6) {atSym[i]="C";atWeight[i]=12.}
  if (atNum[i]==7) {atSym[i]="N";atWeight[i]=14.007}
  if (atNum[i]==8) {atSym[i]="O";atWeight[i]=15.9994}
  if (atNum[i]==9) {atSym[i]="F";atWeight[i]=18.9984}
  if (atNum[i]==10) {atSym[i]="Ne";atWeight[i]=20.1797}
  if (atNum[i]==11) {atSym[i]="Na";atWeight[i]=22.989}
  if (atNum[i]==12) {atSym[i]="Mg";atWeight[i]=24.305}
  if (atNum[i]==13) {atSym[i]="Al";atWeight[i]=26.98154}
  if (atNum[i]==14) {atSym[i]="Si";atWeight[i]=28.0855}
  if (atNum[i]==15) {atSym[i]="P";atWeight[i]=30.9738}
  if (atNum[i]==16) {atSym[i]="S";atWeight[i]=32.066}
  if (atNum[i]==17) {atSym[i]="Cl";atWeight[i]=35.4527}
  if (atNum[i]==18) {atSym[i]="Ar";atWeight[i]=39.948}
  if (atNum[i]==19) {atSym[i]="K";atWeight[i]=39.0983}
  if (atNum[i]==20) {atSym[i]="Ca";atWeight[i]=40.078}
  if (atNum[i]==21) {atSym[i]="Sc";atWeight[i]=44.96}
  if (atNum[i]==22) {atSym[i]="Ti";atWeight[i]=47.867}
  if (atNum[i]==23) {atSym[i]="V";atWeight[i]=50.94}
  if (atNum[i]==24) {atSym[i]="Cr";atWeight[i]=51.9961}
  if (atNum[i]==25) {atSym[i]="Mn";atWeight[i]=54.938}
  if (atNum[i]==26) {atSym[i]="Fe";atWeight[i]=55.845}
  if (atNum[i]==27) {atSym[i]="Co";atWeight[i]=58.933}
  if (atNum[i]==28) {atSym[i]="Ni";atWeight[i]=58.693}
  if (atNum[i]==29) {atSym[i]="Cu";atWeight[i]=63.546}
  if (atNum[i]==30) {atSym[i]="Zn";atWeight[i]=65.38}
  if (atNum[i]==31) {atSym[i]="Ga";atWeight[i]=69.723}
  if (atNum[i]==32) {atSym[i]="Ge";atWeight[i]=72.64}
  if (atNum[i]==33) {atSym[i]="As";atWeight[i]=74.9216}
  if (atNum[i]==34) {atSym[i]="Se";atWeight[i]=78.96}
  if (atNum[i]==35) {atSym[i]="Br";atWeight[i]=79.904}
  if (atNum[i]==46) {atSym[i]="Pd";atWeight[i]=106.42}
  if (atNum[i]==53) {atSym[i]="I";atWeight[i]=126.90447}
# gets actual weight from freqinHP when possible so a prior calc with readisotopes gets you isotopic substitution
  if ((i<100) && ($9>0)) atWeight[i]=$9
#  if ((i>99) && ($8>0)) atWeight[i]=$8

  if ((diag>1) && (i==1)) print "atNum[i],atSym[i],atWeight[i],geoArr[i,1],geoArr[i,2],geoArr[i,3]" >> "diagnostics"
  if (diag>1) print atNum[i],atSym[i],atWeight[i],geoArr[i,1],geoArr[i,2],geoArr[i,3] >> "diagnostics"
  }

# read in frequencies, scale them, read in Reduced masses, read in force
#constants, replace negative frequencies by 2 wavenumbers
numFreq=3*numAtoms-6
if (geometry=="linear") numFreq=3*numAtoms-5
for (i=1;i<=numFreq;i++) {
  $0=""
  getline < "tempfreqs"
  freq[i]=$0*scaling
  if (freq[i]<0) freq[i]=2
  }
for (i=1;i<=numFreq;i++) {
  $0=""
  getline < "tempredmass"
  redMass[i]=$0
  if (redMass[i]=="") redMass[i]=1.
  }
for (i=1;i<=numFreq;i++) {
  $0=""
  getline < "tempfrc"
  frc[i]=$0
  if (frc[i]=="") frc[i]=0.0001
  if (frc[i]==0) frc[i]=0.0001
  if ((diag>1) && (i==1)) print "freq[i],redMass[i],frc[i]" >> "diagnostics"
  if (diag>1) print freq[i],redMass[i],frc[i] >> "diagnostics"
  }

# read in the modes - note that trajectories always need a freq calc with freq=hpmodes unless classical=2
if (classical!=2) {
  for (i=1;i<=numFreq;i+=5) {
    for (j=1;j<=(3*numAtoms);j++) {
      getline < "tempmodes"
      mode[i,$2,$1]=$4; mode[i+1,$2,$1]=$5; mode[i+2,$2,$1]=$6; mode[i+3,$2,$1]=$7; mode[i+4,$2,$1]=$8
      }
    }
  }
if (diag>2) {for (i=1;i<=numFreq;i++) {print mode[i,1,1],mode[i,1,2],mode[i,1,3] >> "modesread"}}

# if doing a cannonball trajectory, read in the vector
if (cannonball>0) {
  for (i=1;i<=numAtoms;i++) {
    getline < "cannontraj"
    cannonArr[i,1]=$1; cannonArr[i,2]=$2; cannonArr[i,3]=$3
    }
  }

# collect a series of random numbers from file temp811, generated from an outside random number generator called by
prodynstarterHP
# read from temp811, starting at a random place
srand(PROCINFO["pid"]); tester=rand()*1000
for (i=1;i<=tester;i++) getline < "temp811"
for (i=1;i<=numFreq;i++) {
  getline < "temp811"; randArr[i]=$1
  getline < "temp811"; randArrB[i]=$1
  getline < "temp811"; randArrC[i]=$1
}
if (rotationmode>0) {
  for (i=1;i<=6;i++) {
    getline < "temp811"; randArrR[i]=$1
  }
}

# for a QM distribution for a harmonic oscillator in its ground state, we want to generate a set of random numbers
#between -1 and 1 weighted such that numbers toward the center are properly more common
i=1
while (i<=numFreq) {
  if ((initialDis==2) || (disMode[i]==2)) {
    getline < "temp811"
    tempNum=2*($1-.5)
    prob=exp(-(tempNum^2))
    getline < "temp811"
    if ($1<prob) {
      randArrD[i]=tempNum
      i++
      }
    }
  if ((initialDis!=2) && (disMode[i]!=2)) i++
  }

# to start without normal modes or frequencies we need to just pick a random direction for the motion of each atom, requiring 3N
random numbers
for (i=1;i<=numAtoms;i++) {
  for (j=1;j<=3;j++) {
  getline < "temp811"
  if ($1>0.5) randArrE[i,j]=1
  if ($1<.5) randArrE[i,j]=-1
    }
  }

# determine energy in each normal mode
for (i=1;i<=numFreq;i++) {
  zpeJ[i]=0.5*h*c*freq[i]     #units J per molecule
#if classical, treat as modes spaced by classicalSpacing wavenumbers
  if (classical==1) zpeJ[i]=0.5*h*c*classicalSpacing  # the zpe is not used when classical but the spacing is used to calculate the
E in mode
```

```
  zpeK[i]=zpeJ[i]*avNum/4184    #units kcal/mol
  if (temp<10) vibN[i]=0        # avoids working with very small temperatures - if the temp is too low, it just acts like 0 K
  if (temp>=10) {
    zpeRat[i]=exp((-2*zpeK[i])/(RgasK*temp))
    if (zpeRat[i]==1) zpeRat[i]=.99999999999
    Q[i]=1/(1-zpeRat[i])
    newRand=randArr[i]
    vibN[i]=0
    tester=1/Q[i]
#     get up to 4000 excitations of low modes
    for (j=1;j<=(4000*zpeRat[i]+2);j++) {
      if (newRand>tester) vibN[i]++
      tester=tester+((zpeRat[i]^j)/Q[i])
      }
    }
  }

# figure out mode energies and maximum classical shift and then actual shift
# also calculated total energy desired for molecule
desiredModeEnK=0
for (i=1;i<=numFreq;i++) {
  modeEn[i]=(zpeJ[i]*1E18)*(2*vibN[i]+1) # units here are mDyne Angstroms for compatibility with Gaussian force constants
  if (classical==1) modeEn[i]=(zpeJ[i]*1E18)*2*vibN[i]    #no zpe when classical
  modeEnK[i]=zpeK[i]*(2*vibN[i]+1)
  if (classical==1) modeEnK[i]=zpeK[i]*2*vibN[i]          #no zpe when classical
  if ((classical==4) && (i>1)) {
    modeEnK[i]=0                          #this lets you turn off the energy in modes outside of 1 for single-trajectory study
    modeEn[i]=0
    }
  desiredModeEnK=desiredModeEnK + modeEnK[i]
# no 1/2 hv for imaginary frequencies
# treating modes with frequencies <10 as translations, ignoring their zero point energies
  if (freq[i]<8) modeEn[i]=(zpeJ[i]*1E18)*(2*vibN[i])
  maxShift[i]=(2*modeEn[i]/frc[i])^0.5
# new 2012 initialDis 3 means random phase of normal mode
  if (initialDis==3) shift[i]=maxShift[i]*sin(randArrC[i]*3.141592*2)
  if (initialDis==2) shift[i]=maxShift[i]*randArrD[i]
  if (initialDis==1) shift[i]=maxShift[i]*(2*(randArrC[i]-0.5))
  if (initialDis==0) shift[i]=0
# lines below allow for setting of displacement mode for individual modes
# It used to be necessary to use disMode 10 to turn off displacements for a mode, but hopefully that bug is killed and you can use
disMode 0
  if (disMode[i]==3) shift[i]=maxShift[i]*sin(randArrC[i]*3.141592*2)
  if (disMode[i]==2) shift[i]=maxShift[i]*randArrD[i]
  if (disMode[i]==1) shift[i]=maxShift[i]*(2*(randArrC[i]-0.5))
  if (disMode[i]==10) shift[i]=0 #kept for backward compatibility
  if (disMode[i]==0) shift[i]=0
# no displacements along imaginary frequencies and very low ones - it is better to treat these
# as translations - employing a shift can give you initial weird geometries
  if (freq[i]<10) shift[i]=0
  if (numimag==1) shift[1]=0
  if (numimag==2) shift[2]=0
  }
for (i=1;i<=numFreq;i++) {
  if ((diag>1) && (i==1)) print "zpeJ[i],zpeK[i],zpeRat[i],Q[i],vibN[i],modeEn[i],maxShift[i],shift[i]" >> "diagnostics"
  if (diag>1) print zpeJ[i],zpeK[i],zpeRat[i],Q[i],vibN[i],modeEn[i],maxShift[i],shift[i] >> "diagnostics"
  }

# multiply each of the modes by its shift and add them up
# Do not do this if classical=2
if (classical!=2) {
  for (i=1;i<=numFreq;i++) {
    for (j=1;j<=numAtoms;j++) {
      for (k=1;k<=3;k++) {
        shiftMode[i,j,k]=mode[i,j,k]*shift[i]
        geoArr[j,k]=geoArr[j,k]+shiftMode[i,j,k]
        }
      }
    }
  }

#now start toward velocities
for (i=1;i<=numFreq;i++) {
  kinEn[i]=100000*(modeEn[i]-0.5*frc[i]*shift[i]^2)  # the 100000 converts to g angstrom^2 s^2
  vel[i]=(2*kinEn[i]/(redMass[i]/avNum))^0.5        # in angstrom / s
#use searchdir in progdyn.conf to control the direction for trajectories started from a saddle point
  if (numimag>1) numimag=1  #only the first freq can be sent in the searchdir direction, the rest go in a random direction
  if (i>numimag) {
    if (randArrB[i]<0.5) vel[i]=-vel[i]
    }
  if (i==numimag) {
    if (searchdir=="negative") vel[i]=-vel[i]
    }
  if ((diag>1) && (i==1)) print "vel[i]" >> "diagnostics"
  if (diag>1) print vel[i] >> "diagnostics"
  }

# if controlphase is being used, set the velocity on particular modes as positive or negative as requested
for (i=1;i<=numFreq;i++) {
  if ((controlPhase[i]=="positive") && (vel[i]<0)) vel[i]=-vel[i]
  if ((controlPhase[i]=="negative") && (vel[i]>0)) vel[i]=-vel[i]
  }

# multiply each of the modes by its velocity and add them up
# Do not do this if classical=2
if (classical!=2) {
  for (i=1;i<=numFreq;i++) {
    for (j=1;j<=numAtoms;j++) {
      for (k=1;k<=3;k++) {
        velMode[i,j,k]=mode[i,j,k]*vel[i]*timestep
        velArr[j,k]=velArr[j,k]+velMode[i,j,k]
        }
      }
    }
  }

# to start without normal modes or frequencies we figure out the energy per atom based on 1/2RT in degree of freedom
if (classical==2) {
# to avoid a bug with a box on, starts without modes should use the input geometry, not the standard
  do {
    getline < "tempinputgeos"
    if (oldline==$0) $0=""
    oldline=$0
    atom = $1
    geoArr[atom,1]=$4; geoArr[atom,2]=$5; geoArr[atom,3]=$6
    geoArrOrig[atom,1]=$4; geoArrOrig[atom,2]=$5; geoArrOrig[atom,3]=$6
    }
  while (length($0) > 0)
  degFreedomEnK=temp*RgasK
  degFreedomEnJ=degFreedomEnK/(avNum/4184)
  cartEn=degFreedomEnJ*1E18
  kinEnCart=100000*cartEn
#print degFreedomEnK, degFreedomEnJ, cartEn, kinEnCart
  for (i=1;i<=numAtoms;i++) {
    for (j=1;j<=3;j++) {
      velArr[i,j]=randArrE[i,j]*timestep*(2*kinEnCart/(atWeight[i]/avNum))^0.5
      if (DRP==1) velArr[i,j]=0
      }
    }
  }

# calculate the KE in the modes at this point
KEinitmodes=0
for (j=1;j<=numAtoms;j++) {
  KEinitmodes=KEinitmodes + 0.5*atWeight[j]*(velArr[j,1]^2 + velArr[j,2]^2 + velArr[j,3]^2)/((timestep^2)*conver1)
  }

# add molecular rotation if requested
if (rotationmode>0) {
#establish three rotation vectors
  for (j=1;j<=numAtoms;j++) {
    rotateX[j,1]=0
    rotateX[j,2]=-geoArrOrig[j,3]
    rotateX[j,3]=geoArrOrig[j,2]
    rotateY[j,1]=-geoArrOrig[j,3]
    rotateY[j,2]=0
    rotateY[j,3]=geoArrOrig[j,1]
    rotateZ[j,1]=-geoArrOrig[j,2]
    rotateZ[j,2]=geoArrOrig[j,1]
    rotateZ[j,3]=0
    }
#figure out how much energy is in the raw vectors
  eRotX=0;eRotY=0;eRotZ=0
  for (j=1;j<=numAtoms;j++) {
    for (k=1;k<=3;k++) {
      eRotX=eRotX + 0.5*atWeight[j]*(rotateX[j,k]^2)/((timestep^2)*conver1)
      eRotY=eRotY + 0.5*atWeight[j]*(rotateY[j,k]^2)/((timestep^2)*conver1)
      eRotZ=eRotZ + 0.5*atWeight[j]*(rotateZ[j,k]^2)/((timestep^2)*conver1)
      }
    }
#  print "rotation energies if raw vector used",eRotX,eRotY,eRotZ
#now decide how much energy we want in each rotation
  keRx=-0.5*0.001987*temp*log(1-randArrR[1])
  keRy=-0.5*0.001987*temp*log(1-randArrR[2])
  keRz=-0.5*0.001987*temp*log(1-randArrR[3])
  if (eRotX<1) keRx=0;if (eRotY<1) keRy=0;if (eRotZ<1) keRz=0
  rotEdesired=keRx+keRy+keRz
  signX=1;signY=1;signZ=1
  if (randArrR[4]<.5) signX=-1
  if (randArrR[5]<.5) signY=-1
  if (randArrR[6]<.5) signZ=-1

#  print "desired energies",keRx,keRy,keRz,"and random numbers",randArrR[1],randArrR[2],randArrR[3]
#protect against zero rotations
  if (eRotX<1) eRotX=1;if (eRotY<1) eRotY=1;if (eRotZ<1) eRotZ=1
#now scale the rotational vectors
  scaleX=(keRx/eRotX)^.5
  scaleY=(keRy/eRotY)^.5
  scaleZ=(keRz/eRotZ)^.5
#  print "scaling factors" scaleX,scaleY,scaleZ
  for (j=1;j<=numAtoms;j++) {
    for (k=1;k<=3;k++) {
      rotateX[j,k]=rotateX[j,k]*scaleX*signX
      rotateY[j,k]=rotateY[j,k]*scaleY*signY
      rotateZ[j,k]=rotateZ[j,k]*scaleZ*signZ
      }
      }
   for (j=1;j<=numAtoms;j++) {
#     print rotateX[j,1],"   ",rotateX[j,2],"   ",rotateX[j,3]
      }
#  print ""
   for (j=1;j<=numAtoms;j++) {
#     print rotateY[j,1],"   ",rotateY[j,2],"   ",rotateY[j,3]
      }
#  print ""
   for (j=1;j<=numAtoms;j++) {
#     print rotateZ[j,1],"   ",rotateZ[j,2],"   ",rotateZ[j,3]
      }
# now add the rotational vectors  to velArr
   for (j=1;j<=numAtoms;j++) {
      for (k=1;k<=3;k++) {
         velArr[j,k]=velArr[j,k]+rotateX[j,k]+rotateY[j,k]+rotateZ[j,k]
         }
      }
   }

# if doing a cannonball, adjust multiplier until extra energy is correct
if (cannonball>0) {
   multiplier=1; tester=0; tolerance=.1
   while (tester==0) {
      KEinittotal=0
      for (j=1;j<=numAtoms;j++) {
         cannonvelArr[j,1]=velArr[j,1]+multiplier*cannonArr[j,1]; cannonvelArr[j,2]=velArr[j,2]+multiplier*cannonArr[j,2];
cannonvelArr[j,3]=velArr[j,3]+multiplier*cannonArr[j,3]
         KEinittotal=KEinittotal + 0.5*atWeight[j]*(cannonvelArr[j,1]^2 + cannonvelArr[j,2]^2 +
cannonvelArr[j,3]^2)/((timestep^2)*conver1)
         }
      if (KEinittotal>(KEinitmodes+cannonball+tolerance)) multiplier=multiplier*0.98901364
      if (KEinittotal<(KEinitmodes+cannonball-tolerance)) multiplier=multiplier*1.01
      if ((KEinittotal<(KEinitmodes+cannonball+tolerance)) && (KEinittotal>(KEinitmodes+cannonball-tolerance))) tester=1
      }
   for (j=1;j<=numAtoms;j++) {
      velArr[j,1]=velArr[j,1]+multiplier*cannonArr[j,1]; velArr[j,2]=velArr[j,2]+multiplier*cannonArr[j,2];
velArr[j,3]=velArr[j,3]+multiplier*cannonArr[j,3]
      }
   }

#output the new geometry.
# ****** this section changed for special experiment for cyclopentadiene.  do not use this for other cases
# atWeight[4]=140.0001
# ****** line below added for special experiment switching mass from 12 to 140, keeping momenta the same
#velArr[4,1]=velArr[4,1]/11.66667; velArr[4,2]=velArr[4,2]/11.66667; velArr[4,3]=velArr[4,3]/11.66667
for (j=1;j<=numAtoms;j++) {
   printf("%2s % .7f % .7f % .7f %9.5f \n",atSym[j],geoArr[j,1],geoArr[j,2],geoArr[j,3],atWeight[j])
   }

#output the velocities and calculate the total kinetic energy overall
KEinittotal=0
for (j=1;j<=numAtoms;j++) {
   KEinittotal=KEinittotal + 0.5*atWeight[j]*(velArr[j,1]^2 + velArr[j,2]^2 + velArr[j,3]^2)/((timestep^2)*conver1)
   printf("% .8f % .8f % .8f \n",velArr[j,1],velArr[j,2],velArr[j,3])
   }

#anything else I add to the file will not affect the trajectories but will keep a record and be good for analysis
if (classical!=2) {
   for (i=1;i<=numFreq;i++) {
      if (initialDis==0) printf("%.6f   % .6f    %4i    % 1.4e       % .6f %1i\n", randArr[i], randArrB[i], vibN[i], vel[i], shift[i],
disMode[i])
```

```
      if (initialDis==1) printf("%.6f   % .6f    %4i    % 1.4e       % .6f %1i\n", randArr[i], randArrC[i], vibN[i], vel[i], shift[i],
disMode[i])
      if (initialDis==2) printf("%.6f   % .6f    %4i    % 1.4e       % .6f %1i\n", randArr[i], randArrD[i], vibN[i], vel[i], shift[i],
disMode[i])
      if (initialDis==3) printf("%.6f   % .6f    %4i    % 1.4e       % .6f %1i % .6f\n", randArr[i], randArrC[i], vibN[i], vel[i], shift[i],
disMode[i], sin(randArrC[i]*3.141592*2))
      }
   }
print "temp ",temp
print "initialDis",initialDis
print "classical",classical
print "timestep",timestep
print "numimag",numimag
OFMT = "%.3f"
print "Total mode energy desired=",desiredModeEnK
print "KE initial from modes=",KEinitmodes,"   KE initial total=",KEinittotal,"   Rotational Energy desired=",rotEdesired
if (cannonball>0) print "cannonball",cannonball,"  cannon Energy=",KEinittotal-KEinitmodes
if (boxon>0) print "boxsize",boxsize
if (DRP>0) print "DRP",DRP,"   maxAtomMove",maxAtomMove
if (DRP>0) print maxAtomMove > "maxMove"
}  # End of BEGIN

/Zero-point correction/ {zpeGauss=$3}
/zero-point Energies/ {zpePlusE=$7}
END {
zpeGaussK=zpeGauss*627.509
potentialE=zpePlusE - zpeGauss
OFMT = "%.6f"
print "Gaussian zpe=",zpeGauss,"or",zpeGaussK,"kcal/mol  E + zpe=",zpePlusE,"  potential E=",potentialE
print "" #will use blank line to mark end of geoPlusVel file
}
```

prog1stpoint

```
BEGIN {
# Nov 2016 organized by functions
# 2014 added ONIOMcharge, more reliable convergence commands
# 2013 added multiple NMR calculations, molecular rotations
# 2012 added NMR calculations
# 2011 added linkatoms in ONIOM
# aug 2010 changed so that it is more careful in reading in from geoPlusVel
#     removed some default parameters that should always be defined
# Jan 2009 - a number of little changes to improve reporting, precision, etc
# aug 2008 added to atom list so handles H to Cl without change needed
# version Feb 2008 incorporates methodfile, boxon and boxsize, though this point unaffected by box
# version Jan 2008 - allows for ONIOM jobs, fixed atoms
# version Sept 2005 - incorporates meth3, meth4, meth5, meth6, but not yet rotation
# this program creates the first input file for g09
# the title should be changed as appropriate
# the isomer number comes from a file isomernumber

initializeparameters()
initializeconstants()
readprogdynconf()

getline < "isomernumber"
isomernum = $1
getline < "runpointnumber"
runpointnum = $1

if (diag>=1) diagnosticsA()
```

```
readgeoPlusVel() #this is different from prog2ndpoint and progdynb because we are just getting the geometry, no velocities or
old geos

writecomfile()
}

END {
}

############################################################ FUNCTIONS
################################################
function Distance(Atom1,Atom2) {
  return sqrt((oldarr[Atom1,1]-oldarr[Atom2,1])^2+(oldarr[Atom1,2]-oldarr[Atom2,2])^2+(oldarr[Atom1,3]-
oldarr[Atom2,3])^2)
  }

# default parameters, including quassiclassical, no displacements, transition state, not a DRP
# do not change these - rather, change progdyn.conf to set the parameters
function initializeparameters() {
  initialDis=0; timestep=1E-15; scaling=1.0; temp=298.15
  classical=0; numimag=1; DRP=0; cannonball=0
  memory=20000000
  diag=1; checkpoint="g09.chk"; searchdir="positive"; boxon=0
  boxsize=10; maxAtomMove=0.1; title1="you"; title2="need"
  title3="a"; title4="progdyn.conf"; processors=1; highlevel=99999; linkatoms=0
  damping=1;nonstandard=0;geometry="nonlinear";nonstandard=0
  nmrtype=0;nmrevery=9999999;nmrcc=0;nmrrand=0;nmrdo=0
  thermostat=0;thermostatemult=1.00
  oniomcharge=0; oniommult=0
  applyforce=0; applyforceB=0; applyforceC=0; zeroatomon=0
  sphereon=0; spheresize=999; sphereforceK=0.01
  empiricaldispersion=0; radiusmultiplier=1.25
  }

function initializeconstants() {
  srand(PROCINFO["pid"])
  i=1;j=1;k=1
  c=29979245800; h=6.626075E-34; avNum=6.0221415E23
  RgasK=0.00198588; RgasJ=8.31447; pi=3.14159265359
  numAtoms=0; atomnumber=0
  conver1=4.184E26 #dividing by this converts amu angs^2 /s^2 to kcal/mol
  OFS="   "
  }

# read progdyn.conf for configuration info
function readprogdynconf() {
  blankLineTester=10
  while (blankLineTester>1) {
    getline < "progdyn.conf"
    if ($1=="method") method=$2
    if ($1=="method2") meth2=$2
    if ($1=="charge") charge=$2
    if ($1=="multiplicity") multiplicity=$2
    if ($1=="oniomchargemult") {
      oniomcharge=$2
      oniommult=$3
      }
    if ($1=="memory") memory=$2
    if ($1=="processors") processors=$2
    if ($1=="checkpoint") checkpoint=$2
    if ($1=="timestep") timestep=$2
    if ($1=="diagnostics") diag=$2
    if ($1=="temperature") temp=$2
    if ($1=="thermostat") thermostat=$2
    if ($1=="thermostatmult") thermostatmult=$2
    if (thermostatmult>1) thermostatmult=1/thermostatmult
    if ($1=="method3") meth3=$2
    if ($1=="method4") meth4=$2
    if ($1=="method5") meth5=$2
    if ($1=="method6") meth6=$2
    if ($1=="method7") meth7=$2
    if ($1=="highlevel") highlevel=$2
    if ($1=="linkatoms") linkatoms=$2
    if ($1=="fixedatom1") fixedatom1=$2
    if ($1=="fixedatom2") fixedatom2=$2
    if ($1=="fixedatom3") fixedatom3=$2
    if ($1=="fixedatom4") fixedatom4=$2
    if ($1=="boxon") boxon=$2
    if ($1=="boxsize") boxsize=$2
    if ($1=="sphereon") sphereon=$2
    if ($1=="spheresize") spheresize=$2
    if ($1=="sphereforce") sphereforceK=$2
    if ($1=="DRP") DRP=$2
    if ($1=="maxAtomMove") maxAtomMove=$2
    if ($1=="methodfile") methodfilelines=$2
    if ($1=="killcheck") killcheck=$2
    if ($1=="empiricaldispersion") empiricaldispersion=$2
    if ($1=="radiusmultiplier") radiusmultiplier=$2
    if ($1=="damping") damping=$2
    if ($1=="NMRmethod") nmrmethod=$2
    if ($1=="NMRmethod2") nmrmethod2=$2
    if ($1=="NMRmethod3") nmrmethod3=$2
    if ($1=="NMRtype") nmrtype=$2
    if ($1=="NMRevery") nmrevery=$2
    if ($1=="NMRrand") nmrrand=$2
    if ($1=="loadlimit") loadlimit=$2
    if ($1=="NMRcc") nmrcc=$2
    if ($1=="nonstandard") nonstandard=$2
    if ($1=="applyforce") {
      applyforce=$2; apforce=$3; apforceX0=$4; apforce2=$5; apforce3=$6
      }
    if ($1=="applyforceB") {
      applyforceB=$2; apforceB=$3; apforceX0B=$4; apforce2B=$5; apforce3B=$6
      }
    if ($1=="applyforceC") {
      applyforceC=$2; apforceC=$3; apforceX0C=$4; apforce2C=$5; apforce3C=$6
      }
    if ($1=="afatoms") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afatom[i]=$(i+1)
        }
      }
    if ($1=="afatomsB") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afatomB[i]=$(i+1)
        }
      }
    if ($1=="afatomsC") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afatomC[i]=$(i+1)
        }
      }
    if ($1=="applyforceplane") {
      applyforceplane=$2; apforceplane=$3; apforceplaneX0=$4
```

```
         }
      if ($1=="afplaneatoms") {
         for (i=1;i<8;i++) {
            if ($(i+1)>0) afplaneatoms[i]=$(i+1)
            }
         }
      if ($1=="zeroatom") {
         zeroatomon=1
         zeroatom=$2
         }
      if ($1=="title") {
         title1=$2
         title2=$3
         title3=$4
         title4=$5
         }
      blankLineTester=length($0)
      }
   }

function diagnosticsA() {
   print "***************** starting progdynb *****************" >> "diagnostics"
   print "method,charge,multiplicity,memory" >> "diagnostics"
   print method,charge,multiplicity,memory >> "diagnostics"
   print "processors,checkpoint,title" >> "diagnostics"
   print processors,checkpoint,title1,title2,title3,title4 >> "diagnostics"
   }

function readgeoPlusVel() {
#this is different from prog2ndpoint and progdynb because we are just getting the geometry, no velocities or old geos
#read in number of atoms, geometry, masses from geoPlusVel
   getline < "geoPlusVel"
   numAtoms=$1
# geometry
   for (i=1;i<=numAtoms;i++) {
      getline < "geoPlusVel"
      weight[i]=$5
      atSym[i]=$1
      for (j=1;j<=3;j++) {
         geoArr[i,j]=$(1+j)
         }
      }
   blankLineTester=10
   while (blankLineTester>1) {
      getline < "geoPlusVel"
      if ($11=="potential") potentialE=$13
      blankLineTester=length($0)
      }
   }

function writecomfile () {
   print "%nproc=" processors
   print "%mem=" memory
   if (killcheck!=1) print "%chk=" checkpoint
   if (nonstandard==0) {
      print "#p " method " force scf=(xqc,maxconven=155,fulllinear,nosym) "
      if (meth2=="unrestricted") print "guess=mix" #for unrestricted calculations
      if (length(meth3)>2) print meth3
      if (length(meth4)>2) print meth4
      }
   if (nonstandard==1) {
      print "# "
   print "nonstd"
   system("cat nonstandard")
   }
print ""
print  title1,title2,title3,title4
print "runpoint ",runpointnum
print "runisomer ",isomernum
if (DRP==1) {
   print "maxForce and forceMult and maxAtomMove",maxForce,forceMult,maxAtomMove
   print "maxForce and forceMult and maxAtomMove",maxForce,forceMult,maxAtomMove >> "movelist"
   }
print ""
if (oniommult==0) print charge,multiplicity
if (oniommult>0) print charge,multiplicity,oniomcharge,oniommult
for (i=1;i<=numAtoms;i++) {
   printf("%s %.7f %.7f %.7f",atSym[i],geoArr[i,1],geoArr[i,2],geoArr[i,3])
   if ((i>highlevel) && (i<=highlevel+linkatoms)) printf(" %s","M H")
   if (i>(highlevel+linkatoms)) printf(" %s","M")
   print ""
   }
print ""
if (length(meth5)>2) print meth5
if (length(meth6)>2) print meth6
if (methodfilelines>=1) {
   for (i=1;i<=methodfilelines;i++) {
      getline < "methodfile"
      print $0
      }
   }
print ""
if ((nmrtype>0) && ((runpointnum % nmrevery)==0)) {
   print "--link1--"
   print "%nproc=" processors
   print "%mem=" memory
   print "%chk=" checkpoint
   print "# " nmrmethod " nmr=giao geom=check"
   if (nmrmethod==method) print "guess=tcheck"
   if (length(meth7)>2) print meth7
   print ""
   print title1,title2,title3,title4
   print "runpoint ",runpointnum
   print "runisomer ",isomernum
   print ""
   print charge,multiplicity
   }
print ""
if ((nmrtype>1) && ((runpointnum % nmrevery)==0)) {
   print "--link1--"
   print "%nproc=" processors
   print "%mem=" memory
   print "%chk=" checkpoint
   print "# " nmrmethod2 " nmr=giao geom=check"
   if (length(meth7)>2) print meth7
   print ""
   print title1,title2,title3,title4
   print "runpoint ",runpointnum
   print "runisomer ",isomernum
   print ""
   print charge,multiplicity
   }
print ""
if ((nmrtype>2) && ((runpointnum % nmrevery)==0)) {
   print "--link1--"
   print "%nproc=" processors
   print "%mem=" memory
   print "%chk=" checkpoint
   print "# " nmrmethod3 " nmr=giao geom=check"
   if (length(meth7)>2) print meth7
   print ""
   print title1,title2,title3,title4
   print "runpoint ",runpointnum
   print "runisomer ",isomernum
   print ""
   print charge,multiplicity
   }
print ""
}
```

prog2ndpoint

```
BEGIN {
# Nov 2016 organized into functions
# 2014 added ONIOMcharge, more reliable convergence commands
# aug 2013 includes molecular rotation, ability to do multiple NMR calculations, ONIOM with link atoms,
#nonstandard routes, checks more kinds of energies at point 2
#Aug 2010 added etolerance to make it controllable from progdyn.conf, made it so that DRP does not check energy
# aug 2008 added to atom list so handles 1 to 17 without change needed
# version Feb 2008 incorporates methodfile, boxon and boxsize, though this point unaffected by box
# version Jan 2008 - allows for ONIOM jobs, fixed atoms
# version Sept 9, 2005 - incorportates meth3, meth4, meth5, meth6, but not yet rotation
# read progdyn.conf for configuration info

initializeparameters()
initializeconstants()
readprogdynconf()

getline < "isomernumber"
isomernum = $1
getline < "runpointnumber"
runpointnum = $1

if (diag>=1) diagnosticsA()

#get forward or reverse from skipstart if it exists
getline < "skipstart"
trajdirection = $1

readgeoPlusVel()
putPoint1Traj()
addVelocities()
} # end of BEGIN

#pull out the potential energy
/SCF Done/ || /EUMP2 =/ || / Energy=/ || /ONIOM:/ {
if (($1=="Energy=") && ($3=="NIter="))  newPotentialE=$2
if ($1=="SCF") newPotentialE=$5  #This was subject to a bug at one time - grep Samae in old files
if ($2=="extrapolated")  newPotentialE=$5
if ($1=="E2") {
  tempstring=$6
  split(tempstring, arr10, "D")
  newPotentialE=arr10[1]*(10^arr10[2])
  }
newPotEK=(newPotentialE-potentialE)*627.509
```

```
newPotentialEK=(newPotentialE-potentialE)*627.509
}

# now we go ahead and collect the forces from the point 1 file
(/     1   / || /     2   / || /     3   / || /     4   / || /     5   / || /     6   / || /     7   / || /     8   / || /     9   / || /    10   / || /
11   / || /    12   / || /    13   / || /    14   / || /    15   / || /    16   / || /    17   / || /    18   / || /    19   / || /    20   / || /
21   / || /    22   / || /    23   / || /    24   / || /    25   / || /    26   / || /    27   / || /    28   / || /    29   / || /    30   / || /
31   / || /    32   / || /    33   / || /    34   / || /    35   /) && length($3) > 9 {
i=$1
for (j=1;j<=3;j++) {
  forceArr[i,j]=$(2+j)   #the raw units of the forces are Hartree/Bohr
  }
if ((diag>1) && (i==1)) print "i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3]" >> "diagnostics"
if (diag>1) print i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3] >> "diagnostics"
}

END {
if (DRP==0) doEcheck()
addForceEffect()
writecomfile()
writetraj()
}

############################################################ FUNCTIONS
################################################
function Distance(Atom1,Atom2) {
  return sqrt((oldarr[Atom1,1]-oldarr[Atom2,1])^2+(oldarr[Atom1,2]-oldarr[Atom2,2])^2+(oldarr[Atom1,3]-
oldarr[Atom2,3])^2)
  }

# default parameters, including quassiclassical, no displacements, transition state, not a DRP
# do not change these - rather, change progdyn.conf to set the parameters
function initializeparameters() {
  initialDis=0; timestep=1E-15; scaling=1.0; temp=298.15
  classical=0; numimag=1; DRP=0; cannonball=0
  memory=20000000
  diag=1; checkpoint="g09.chk"; searchdir="positive"; boxon=0
  boxsize=10; maxAtomMove=0.1; title1="you"; title2="need"
  title3="a"; title4="progdyn.conf"; processors=1; highlevel=99999; linkatoms=0
  damping=1;nonstandard=0;geometry="nonlinear";nonstandard=0
  nmrtype=0;nmrevery=9999999;nmrcc=0;nmrrand=0;nmrdo=0
  thermostat=0;thermostatemult=1.00
  oniomcharge=0; oniommult=0
  applyforce=0; applyforceB=0; applyforceC=0; zeroatomon=0
  sphereon=0; spheresize=999; sphereforceK=0.01
  empiricaldispersion=0; radiusmultiplier=1.25
  etolerance=1
  }

function initializeconstants() {
  srand(PROCINFO["pid"])
  i=1;j=1;k=1
  c=29979245800; h=6.626075E-34; avNum=6.0221415E23
  RgasK=0.00198588; RgasJ=8.31447; pi=3.14159265359
  numAtoms=0; atomnumber=0
  conver1=4.184E26 #dividing by this converts amu angs^2 /s^2 to kcal/mol
  OFS="   "
  }

# read progdyn.conf for configuration info
function readprogdynconf() {
  blankLineTester=10
```

```
while (blankLineTester>1) {
   getline < "progdyn.conf"
   if ($1=="method") method=$2
   if ($1=="method2") meth2=$2
   if ($1=="charge") charge=$2
   if ($1=="multiplicity") multiplicity=$2
   if ($1=="oniomchargemult") {
      oniomcharge=$2
      oniommult=$3
      }
   if ($1=="memory") memory=$2
   if ($1=="processors") processors=$2
   if ($1=="checkpoint") checkpoint=$2
   if ($1=="timestep") timestep=$2
   if ($1=="diagnostics") diag=$2
   if ($1=="temperature") temp=$2
   if ($1=="thermostat") thermostat=$2
   if ($1=="thermostatmult") thermostatmult=$2
   if (thermostatmult>1) thermostatmult=1/thermostatmult
   if ($1=="method3") meth3=$2
   if ($1=="method4") meth4=$2
   if ($1=="method5") meth5=$2
   if ($1=="method6") meth6=$2
   if ($1=="method7") meth7=$2
   if ($1=="highlevel") highlevel=$2
   if ($1=="linkatoms") linkatoms=$2
   if ($1=="fixedatom1") fixedatom1=$2
   if ($1=="fixedatom2") fixedatom2=$2
   if ($1=="fixedatom3") fixedatom3=$2
   if ($1=="fixedatom4") fixedatom4=$2
   if ($1=="boxon") boxon=$2
   if ($1=="boxsize") boxsize=$2
   if ($1=="sphereon") sphereon=$2
   if ($1=="spheresize") spheresize=$2
   if ($1=="sphereforce") sphereforceK=$2
   if ($1=="DRP") DRP=$2
   if ($1=="maxAtomMove") maxAtomMove=$2
   if ($1=="methodfile") methodfilelines=$2
   if ($1=="killcheck") killcheck=$2
   if ($1=="etolerance") etolerance=$2
   if ($1=="reversetraj") reversetraj=$2
   if ($1=="empiricaldispersion") empiricaldispersion=$2
   if ($1=="radiusmultiplier") radiusmultiplier=$2
   if ($1=="damping") damping=$2
   if ($1=="NMRmethod") nmrmethod=$2
   if ($1=="NMRmethod2") nmrmethod2=$2
   if ($1=="NMRmethod3") nmrmethod3=$2
   if ($1=="NMRtype") nmrtype=$2
   if ($1=="NMRevery") nmrevery=$2
   if ($1=="NMRrand") nmrrand=$2
   if ($1=="loadlimit") loadlimit=$2
   if ($1=="NMRcc") nmrcc=$2
   if ($1=="nonstandard") nonstandard=$2
   if ($1=="applyforce") {
      applyforce=$2; apforce=$3; apforceX0=$4; apforce2=$5; apforce3=$6
      }
   if ($1=="applyforceB") {
      applyforceB=$2; apforceB=$3; apforceX0B=$4; apforce2B=$5; apforce3B=$6
      }
   if ($1=="applyforceC") {
      applyforceC=$2; apforceC=$3; apforceX0C=$4; apforce2C=$5; apforce3C=$6
      }
      if ($1=="afatoms") {
         for (i=1;i<8;i++) {
            if ($(i+1)>0) afatom[i]=$(i+1)
            }
         }
      if ($1=="afatomsB") {
         for (i=1;i<8;i++) {
            if ($(i+1)>0) afatomB[i]=$(i+1)
            }
         }
      if ($1=="afatomsC") {
         for (i=1;i<8;i++) {
            if ($(i+1)>0) afatomC[i]=$(i+1)
            }
         }
      if ($1=="applyforceplane") {
         applyforceplane=$2; apforceplane=$3; apforceplaneX0=$4
         }
      if ($1=="afplaneatoms") {
         for (i=1;i<8;i++) {
            if ($(i+1)>0) afplaneatoms[i]=$(i+1)
            }
         }
      if ($1=="zeroatom") {
         zeroatomon=1
         zeroatom=$2
         }
      if ($1=="title") {
         title1=$2
         title2=$3
         title3=$4
         title4=$5
         }
      blankLineTester=length($0)
      }
   }

function diagnosticsA() {
   print "***************** starting progdynb *****************" >> "diagnostics"
   print "method,charge,multiplicity,memory" >> "diagnostics"
   print method,charge,multiplicity,memory >> "diagnostics"
   print "processors,checkpoint,title" >> "diagnostics"
   print processors,checkpoint,title1,title2,title3,title4 >> "diagnostics"
   }

function readgeoPlusVel() {
#this is different from prog1stpoint and progdynb because we are just getting velocities
#read in number of atoms, geometry, masses from geoPlusVel
   getline < "geoPlusVel"
   numAtoms=$1
# geometry
   for (i=1;i<=numAtoms;i++) {
      getline < "geoPlusVel"
      weight[i]=$5
      atSym[i]=$1
      for (j=1;j<=3;j++) {
         geoArr[i,j]=$(1+j)
         }
      }
#velocities
   for (i=1;i<=numAtoms;i++) {
      getline < "geoPlusVel"
```

```
    for (j=1;j<=3;j++) {
      velArr[i,j]=$j
      }
    }
#pull out other information useful for testing whether total energy is right or bad
  blankLineTester=10
  while (blankLineTester>1) {
    getline < "geoPlusVel"
    if ($4=="desired=") desiredModeEnK=$5
    if ($4=="modes=") {
      KEinitmodes=$5
      KEinittotal=$9
      }
    if ($11=="potential") potentialE=$13
    blankLineTester=length($0)
    }
  }

function addVelocities() {
# ok, now we have to figure the second point.  this should be
# x(t) = x + v*t + 1/2*F*t^2/m
#now we go ahead and add the velocities
  for (i=1;i<=numAtoms;i++) {
    for (j=1;j<=3;j++) {
      arr[i,j]=velArr[i,j]+geoArr[i,j]
      if (trajdirection=="reverserestart") arr[i,j]=geoArr[i,j]-velArr[i,j]
      }
    if ((diag>1) && (i==1)) print "geometry after adding velocities" >> "diagnostics"
    if (diag>1) print arr[i,1],arr[i,2],arr[i,3] >> "diagnostics"
    }
  }

function putPoint1Traj() {
#get initial geometry into file traj
  print numAtoms >> "traj"
  print potentialE,title1,title2,title3,title4,"runpoint 1 ","runisomer ",isomernum >> "traj"
  for (i=1;i<=numAtoms;i++) {
    print atSym[i],geoArr[i,1],geoArr[i,2],geoArr[i,3] >> "traj"
    }
  }

function doEcheck() {
  print "trajectory #",isomernum >> "Echeck"
  print "point 1 potential E=",newPotentialEK,"   point 1 kinetic E=",KEinitmodes,"  Total=",newPotentialEK+KEinitmodes >>
"Echeck"
  print "desired total energy=", desiredModeEnK >> "Echeck"
  if ((newPotentialEK+KEinitmodes)>(desiredModeEnK+etolerance)) print "XXXX bad total Energy" >> "Echeck"
  if ((newPotentialEK+KEinitmodes)<(desiredModeEnK-etolerance)) print "XXXX bad total Energy" >> "Echeck"
  }

function addForceEffect() {
# x(t) = x + v*t + 1/2*F*t^2/m
# turn the forces into motion
  for (i=1;i<=numAtoms;i++) {
    for (j=1;j<=3;j++) {
# conversions here take force to J/angstrom, 1E20 converts to kg angstroms / s^2, then mult time (s^s) and divide by weight in kg
to get angstroms
      forceArr[i,j]=0.5*1E20*forceArr[i,j]*627.509*(4184/(0.529177*avNum))*(timestep^2)/(weight[i]/(avNum*1000))
# for simplicity, DRPs will throw away the forces at the second pont.  This means that if we are not at a saddlepoint, point 2 =
point 1 but this is a minor waste
      if (DRP==1) forceArr[i,j]=0
      arr[i,j]=arr[i,j]+forceArr[i,j]
# if atoms are fixed, replace calcd new position by original position
      if ((i==fixedatom1) || (i==fixedatom2) || (i==fixedatom3) || (i==fixedatom4)) arr[i,j]=geoArr[i,j]
      }
    if ((diag>1) && (i==1)) print "i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3]" >> "diagnostics"
    if (diag>1) print i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3] >> "diagnostics"
    }
  }

function writecomfile() {
  print "%nproc=" processors
  print "%mem=" memory
  if (killcheck!=1) print "%chk=" checkpoint
  if (nonstandard==0) {
    print "# " method " force scf=(xqc,maxconven=55,fulllinear,nosym) "
    if (meth2=="unrestricted") print "guess=mix" #for unrestricted calculations
    if (meth2=="read") print "guess=tcheck" #for reading orbitals from check, sometimes faster, sometimes not
    if (length(meth3)>2) print meth3
    if (length(meth4)>2) print meth4
    }
  if (nonstandard==1) {
    print "# "
    print "nonstd"
    system("cat nonstandard")
    }
  print ""
  print title1,title2,title3,title4
  print "runpoint ",runpointnum
  print "runisomer ", isomernum
  print ""
  if (oniommult==0) print charge,multiplicity
  if (oniommult>0) print charge,multiplicity,oniomcharge,oniommult
  for (i=1;i<=numAtoms;i++) {
    printf("%s %.7f %.7f %.7f",atSym[i],arr[i,1],arr[i,2],arr[i,3])
    if ((i>highlevel) && (i<=highlevel+linkatoms)) printf(" %s","M H")
    if (i>(highlevel+linkatoms)) printf(" %s","M")
    print ""
    }
  print ""
  if (length(meth5)>2) print meth5
  if (length(meth6)>2) print meth6
  if (methodfilelines>=1) {
    for (i=1;i<=methodfilelines;i++) {
      getline < "methodfile"
      print $0
      }
    }
  if ((nmrtype>0) && ((runpointnum % nmrevery)==0)) {
    print "--link1--"
    print "%nproc=" processors
    print "%mem=" memory
    print "%chk=" checkpoint
    print "# " nmrmethod " nmr=giao geom=check"
    if (nmrmethod==method) print "guess=tcheck"
    if (length(meth7)>2) print meth7
    print ""
    print title1,title2,title3,title4
    print "runpoint ",runpointnum
    print "runisomer ",isomernum
    print ""
    print charge,multiplicity
    }
  print ""
  if ((nmrtype>1) && ((runpointnum % nmrevery)==0)) {
    print "--link1--"
    print "%nproc=" processors
    print "%mem=" memory
    print "%chk=" checkpoint
    print "# " nmrmethod2 " nmr=giao geom=check"
    if (length(meth7)>2) print meth7
    print ""
    print title1,title2,title3,title4
    print "runpoint ",runpointnum
    print "runisomer ",isomernum
    print ""
    print charge,multiplicity
    }
  print ""
  if ((nmrtype>2) && ((runpointnum % nmrevery)==0)) {
    print "--link1--"
    print "%nproc=" processors
    print "%mem=" memory
    print "%chk=" checkpoint
    print "# " nmrmethod3 " nmr=giao geom=check"
    if (length(meth7)>2) print meth7
    print ""
    print title1,title2,title3,title4
    print "runpoint ",runpointnum
    print "runisomer ",isomernum
    print ""
    print charge,multiplicity
    }
  print ""
  }

function writetraj() {
#get second geometry into file traj
  print numAtoms >> "traj"
  print newPotentialE,title1,title2,title3,title4,"runpoint ",runpointnum,"runisomer ",isomernum >> "traj"
  for (i=1;i<=numAtoms;i++) {
    print atSym[i],arr[i,1],arr[i,2],arr[i,3] >> "traj"
    }
  }
```

progdynb

```
BEGIN { #this is the main routine for generating new .com files by the Verlet algorithym
# Nov 2016 organized into functions, added empirical dispersion
# Oct 2015 added sanity check, eliminated dyn files and keepevery
# May 2015 added ability to force solvent into a sphere
# Feb 2015 added zeroatom ability to make solute centered in solvent
# Jan 2015 added applyforce to apply forces on motion of atoms and allow PMF calculations
# Dec 2014 added oniom charge specifications
# 2013 added multiple NMR calculations, molecular rotations, thermostat commands
# 2012 added NMR calculations
# 2011 added linkatoms in ONIOM
# Aug 2010 increased elements handled automatically but only up to bromine!
# Jan 2009 - a number of little changes to improve reporting, precision, etc
# Nov 2008 added ability to handle DRPs
# Aug 2008 added long list of atoms to handle 1-17 without change
# May 2008 added option to put out velocities in vellist - make diag=3
# version Feb 2008 incorporates methodfile, boxon and boxsize
# version Jan 2008 incorporates fixed atoms, oniom, and velocity damping
# version August 2007 incorporates keepevery to decrease size of dyn file
# version Sept 11, 2005 - incorportates meth3, meth4, meth5, meth6, but not yet rotation

initializeparameters()
initializeconstants()
readprogdynconf()

getline < "isomernumber"
isomernum = $1
getline < "runpointnumber"
runpointnum = $1

if (diag>=1) diagnosticsA()
readgeoPlusVelAndoldAndolder() #readgeoPlusVelAndoldAndolder sets numAtoms, weight[i], atSym[i], potentialE, oldarr[at,1to3], olderarr[at,1to3]
        # also sets apparentTemp,  KEold, does sanitycheck
if (DRP==1) readoldAdjForcesAndmaxMove() #for DRPs read in oldForce[at,123] and maxAtomMove
nmrstuffA() # sets nmrdo to 0 or 1, controls where NMR calcs are done
}
########################  END OF BEGIN  ##################################

#pull out the potential energy
/SCF Done/ || /EUMP2 =/ || / Energy=/ || /ONIOM:/ {
if (($1=="Energy=") && ($3=="NIter="))  newPotentialE=$2
if ($1=="SCF") newPotentialE=$5
if ($2=="extrapolated")  newPotentialE=$5
if ($1=="E2") {
  tempstring=$6
  split(tempstring, arr10, "D")
  newPotentialE=arr10[1]*(10^arr10[2])
  }
newPotEK=(newPotentialE-potentialE)*627.509
}

#must adjust next line for weird atoms
(/    1   / || /    2   / || /    3   / || /    4   / || /    5   / || /    6   / || /    7   / || /    8   / || /    9   / || /    10   / || /
11   / || /    12   / || /    13   / || /    14   / || /    15   / || /    16   / || /    17   / || /    18   / || /    19   / || /    20   / || /
21   / || /    22   / || /    23   / || /    24   / || /    25   / || /    26   / || /    27   / || /    28   / || /    29   / || /    30   / || /
31   / || /    32   / || /    33   / || /    34   / || /    35   /) && length($3) > 9 {
i=$1
for (j=1;j<=3;j++) {
  forceArr[i,j]=$(2+j)   #the raw units of the forces are Hartree/Bohr
  }
#if ((diag>1) && (i==1)) print "i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3]" >> "diagnostics"
#if (diag>1) print i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3] >> "diagnostics"
}

END {
if (sphereon==1) applysphereforce() # apply a force to bring atoms within a sphere.  This also figures out the density at 0.9*spheresize
# routines to apply forces between atoms, used for umbrella sampling
if (applyforceplane>0) doapplyforceplane() #used to apply a force on an atom versus its distance from a plane
setinterlockingsphereatom() #apply force to one of a series of atoms, whichever is closest to afatom[1]
if (applyforce>0) doapplyforce()
if (applyforceB>0) doapplyforceB()
if (applyforceC>0) doapplyforceC()
if (empiricaldispersion>0) doempiricaldispersion()
if (zeroatomon==1) dozeroatom() #routine to slowly move an atom toward the origin
if (thermostat==1) dothermostat()
if (DRP==1) doDRP() #routine for steepest descent path in mass weighted coordinates
if (DRP==0) doVerlet() #normal routine for Verlet
writetraj()
writecomfile()
```

```
}

###################################################################### FUNCTIONS
##########################################################
function Distance(Atom1,Atom2) {
  return sqrt((oldarr[Atom1,1]-oldarr[Atom2,1])^2+(oldarr[Atom1,2]-oldarr[Atom2,2])^2+(oldarr[Atom1,3]-
oldarr[Atom2,3])^2)
  }

# default parameters, including quassiclassical, no displacements, transition state, not a DRP
# do not change these - rather, change progdyn.conf to set the parameters
function initializeparameters() {
  initialDis=0; timestep=1E-15; scaling=1.0; temp=298.15
  classical=0; numimag=1; DRP=0; cannonball=0
  memory=20000000
  diag=1; checkpoint="g09.chk"; searchdir="positive"; boxon=0
  boxsize=10; maxAtomMove=0.1; title1="you"; title2="need"
  title3="a"; title4="progdyn.conf"; processors=1; highlevel=99999; linkatoms=0
  damping=1;nonstandard=0;geometry="nonlinear";nonstandard=0
  nmrtype=0;nmrevery=9999999;nmrcc=0;nmrrand=0;nmrdo=0
  thermostat=0;thermostatemult=1.00
  oniomcharge=0; oniommult=0
  applyforce=0; applyforceB=0; applyforceC=0; zeroatomon=0
  sphereon=0; spheresize=999; sphereforceK=0.01
  empiricaldispersion=0; radiusmultiplier=1.25
  }

function initializeconstants() {
  srand(PROCINFO["pid"])
  i=1;j=1;k=1
  c=29979245800; h=6.626075E-34; avNum=6.0221415E23
  RgasK=0.00198588; RgasJ=8.31447; pi=3.14159265359
  numAtoms=0; atomnumber=0
  conver1=4.184E26 #dividing by this converts amu angs^2 /s^2 to kcal/mol
  OFS="   "
  }

# read progdyn.conf for configuration info
function readprogdynconf() {
  blankLineTester=10
  while (blankLineTester>1) {
    getline < "progdyn.conf"
    if ($1=="method") method=$2
    if ($1=="method2") meth2=$2
    if ($1=="charge") charge=$2
    if ($1=="multiplicity") multiplicity=$2
    if ($1=="oniomchargemult") {
      oniomcharge=$2
      oniommult=$3
      }
    if ($1=="memory") memory=$2
    if ($1=="processors") processors=$2
    if ($1=="checkpoint") checkpoint=$2
    if ($1=="timestep") timestep=$2
    if ($1=="diagnostics") diag=$2
    if ($1=="temperature") temp=$2
    if ($1=="thermostat") thermostat=$2
    if ($1=="thermostatmult") thermostatmult=$2
    if (thermostatmult>1) thermostatmult=1/thermostatmult
    if ($1=="method3") meth3=$2
    if ($1=="method4") meth4=$2
    if ($1=="method5") meth5=$2
    if ($1=="method6") meth6=$2
    if ($1=="method7") meth7=$2
    if ($1=="highlevel") highlevel=$2
    if ($1=="linkatoms") linkatoms=$2
    if ($1=="fixedatom1") fixedatom1=$2
    if ($1=="fixedatom2") fixedatom2=$2
    if ($1=="fixedatom3") fixedatom3=$2
    if ($1=="fixedatom4") fixedatom4=$2
    if ($1=="boxon") boxon=$2
    if ($1=="boxsize") boxsize=$2
    if ($1=="sphereon") sphereon=$2
    if ($1=="spheresize") spheresize=$2
    if ($1=="sphereforce") sphereforceK=$2
    if ($1=="DRP") DRP=$2
    if ($1=="maxAtomMove") maxAtomMove=$2
    if ($1=="methodfile") methodfilelines=$2
    if ($1=="killcheck") killcheck=$2
    if ($1=="empiricaldispersion") empiricaldispersion=$2
    if ($1=="radiusmultiplier") radiusmultiplier=$2
    if ($1=="damping") damping=$2
    if ($1=="NMRmethod") nmrmethod=$2
    if ($1=="NMRmethod2") nmrmethod2=$2
    if ($1=="NMRmethod3") nmrmethod3=$2
    if ($1=="NMRtype") nmrtype=$2
    if ($1=="NMRevery") nmrevery=$2
    if ($1=="NMRrand") nmrrand=$2
    if ($1=="loadlimit") loadlimit=$2
    if ($1=="NMRcc") nmrcc=$2
    if ($1=="nonstandard") nonstandard=$2
    if ($1=="applyforce") {
      applyforce=$2; apforce=$3; apforceX0=$4; apforce2=$5; apforce3=$6
      }
    if ($1=="applyforceB") {
      applyforceB=$2; apforceB=$3; apforceX0B=$4; apforce2B=$5; apforce3B=$6
      }
    if ($1=="applyforceC") {
      applyforceC=$2; apforceC=$3; apforceX0C=$4; apforce2C=$5; apforce3C=$6
      }
    if ($1=="afatoms") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afatom[i]=$(i+1)
        }
      }
    if ($1=="afatomsB") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afatomB[i]=$(i+1)
        }
      }
    if ($1=="afatomsC") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afatomC[i]=$(i+1)
        }
      }
    if ($1=="applyforceplane") {
      applyforceplane=$2; apforceplane=$3; apforceplaneX0=$4
      }
    if ($1=="afplaneatoms") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afplaneatoms[i]=$(i+1)
        }
      }
    if ($1=="zeroatom") {
      zeroatomon=1
      zeroatom=$2
      }
    if ($1=="title") {
      title1=$2
      title2=$3
      title3=$4
      title4=$5
      }
    blankLineTester=length($0)
    }
  }

function diagnosticsA() {
  print "***************** starting progdynb *****************" >> "diagnostics"
  print "method,charge,multiplicity,memory" >> "diagnostics"
  print method,charge,multiplicity,memory >> "diagnostics"
  print "processors,checkpoint,title" >> "diagnostics"
  print processors,checkpoint,title1,title2,title3,title4 >> "diagnostics"
  }

function readgeoPlusVelAndoldAndolder() {
# get number of atoms and weights from geoPlusVel, and previous geometries from old and older
  getline < "geoPlusVel"
  numAtoms=$1
  for (i=1;i<=numAtoms;i++) {
    getline < "geoPlusVel"
    weight[i]=$5; atSym[i]=$1
    }
  blankLineTester=10
  while (blankLineTester>1) {
    getline < "geoPlusVel"
    if ($11=="potential") potentialE=$13
    blankLineTester=length($0)
    }

  for (at=1;at<=numAtoms;at++) {
    getline < "old"
    oldarr[at,1]=$4; oldarr[at,2]=$5; oldarr[at,3]=$6
    }

  for (at=1;at<=numAtoms;at++) {
    getline < "older"
    olderarr[at,1]=$4; olderarr[at,2]=$5; olderarr[at,3]=$6
    atomVel=((oldarr[at,1]-olderarr[at,1])^2 + (oldarr[at,2]-olderarr[at,2])^2 +(oldarr[at,3]-olderarr[at,3])^2)^.5
    KEold=KEold+0.5*weight[at]*(atomVel^2)/((timestep^2)*conver1)
# sanity check - avoids trajectory blow up
    if (atomVel>1) exit
    }
  apparentTemp=KEold*2/(3*RgasK*numAtoms) # this is not the temperature at the previous point but rather 1.5 points back.
# sanity check - avoids trajectory blow up
  for (at=1;at<=numAtoms;at++) {
    if (((oldarr[at,1]-olderarr[at,1])^2)>1) exit
    }
  }

function readoldAdjForcesAndmaxMove() {
#for DRPs read in oldAdjForces and maxAtomMove
  for (at=1;at<=numAtoms;at++) {
    getline < "oldAdjForces"
    oldForce[at,1]=$1; oldForce[at,2]=$2; oldForce[at,3]=$3
    }
  getline < "maxMove"
  if (($1<maxAtomMove) && ($1>0)) maxAtomMove=$1
  if (maxAtomMove<0.000001) maxAtomMove=0.000001
  }

function nmrstuffA() {
# sets nmrdo to 0 or 1, controls where NMR calcs are done
  if ((nmrrand==0) && ((runpointnum % nmrevery)==0)) nmrdo=1
  if ((nmrrand==1) && (rand()<(1/nmrevery))) nmrdo=1
  getline < "uptimelist"
  x=1.0001*substr($10,1,3);if (x<8) x=8
# turn of nmrs if load is too high - this is under control of loadlimit parameter in progdyn.conf and requires proganal to make
uptimelist
  if ((nmrrand==1) && (x>loadlimit)) nmrdo=0
  }

function applysphereforce() {
# apply a force to bring atoms within a sphere.  This also figures out the density at 0.9*spheresize
# We are going through the loop twice, the first time figure outs the total pressure, then scales the forces to limit the pressure to
maxpressure
# the second time actually applies the pressure
  maxpressure=1000 #atmospheres.  Later we may parameterize this.
  sphereforcetotal=0
  for (i=1;i<=numAtoms;i++) {
    distToOrig=((oldarr[i,1]^2+oldarr[i,2]^2+oldarr[i,3]^2)^.5)
    if (distToOrig>spheresize) {
      sphereforce=sphereforceK*(distToOrig-spheresize)
      if (sphereforce>0.01) sphereforce=0.01 #important limit on force for atoms far outside of the sphere, not sure if value
chosen is best
      sphereforcetotal=sphereforcetotal+sphereforce
      }
    }
  sphereforcetotalNewtons=sphereforcetotal*627.509*4184*1E10/(0.529177*avNum)
  surfaceareaSqMeters=4*pi*spheresize^2/1E20
  pressurePascal=sphereforcetotalNewtons/surfaceareaSqMeters
  pressureAtm=pressurePascal/101325
  if (pressureAtm>maxpressure) sphereforceK=sphereforceK*maxpressure/pressureAtm
# go through the loop again
  sphereforcetotal=0; totalweight=0
  for (i=1;i<=numAtoms;i++) {
    distToOrig=((oldarr[i,1]^2+oldarr[i,2]^2+oldarr[i,3]^2)^.5)
    if (distToOrig>spheresize) {
      sphereforce=sphereforceK*(distToOrig-spheresize)
      if (sphereforce>0.01) sphereforce=0.01 #important limit on force for atoms far outside of the sphere, not sure if value
chosen is best
      sphereforcetotal=sphereforcetotal+sphereforce

unitX=sphereforce*oldarr[i,1]/distToOrig;unitY=sphereforce*oldarr[i,2]/distToOrig;unitZ=sphereforce*oldarr[i,3]/distToOrig
      forceArr[i,1]=forceArr[i,1]-unitX;forceArr[i,2]=forceArr[i,2]-unitY;forceArr[i,3]=forceArr[i,3]-unitZ
      }
# calculate the density at 0.9*spheresize
    if (distToOrig<0.9*spheresize) {
      totalweight=totalweight+weight[i]
      }
    }
#recalculate pressure
  sphereforcetotalNewtons=sphereforcetotal*627.509*4184*1E10/(0.529177*avNum)
  surfaceareaSqMeters=4*pi*spheresize^2/1E20
  pressurePascal=sphereforcetotalNewtons/surfaceareaSqMeters
  pressureAtm=pressurePascal/101325
  density=(totalweight/avNum)/((4/3)*pi*(0.9*spheresize*1E-8)^3)
  }
```

```
function doapplyforceplane() {
#used to apply a force on an atom versus its distance from a plane
  if (afplaneatoms[4]<.5) {
    print "you need more atoms to define a plane"
    exit
    }
  for (i=2;i<8;i++) {
    if (afplaneatoms[i]>.5) {
      k=afplaneatoms[i]
      A[1,1]=A[1,1]+oldarr[k,1]^2;A[1,2]=A[1,2]+oldarr[k,1]*oldarr[k,2];A[1,3]=A[1,3]+oldarr[k,1]
      A[2,1]=A[2,1]+oldarr[k,1]*oldarr[k,2];A[2,2]=A[2,2]+oldarr[k,2]^2;A[2,3]=A[2,3]+oldarr[k,2]
      A[3,1]=A[3,1]+oldarr[k,1];A[3,2]=A[3,2]+oldarr[k,2];A[3,3]++
      b[1]=b[1]+oldarr[k,1]*oldarr[k,3];b[2]=b[2]+oldarr[k,2]*oldarr[k,3];b[3]=b[3]+oldarr[k,3]
# find center of mass assuming all atoms same weight
      cmass[1]=cmass[1]+oldarr[k,1];cmass[2]=cmass[2]+oldarr[k,2];cmass[3]=cmass[3]+oldarr[k,3];
      }
    }
  numplaneatoms=A[3,3]
  cmass[1]=cmass[1]/A[3,3];cmass[2]=cmass[2]/A[3,3];cmass[3]=cmass[3]/A[3,3]
#   print "matrix A"
#   for (i=1;i<=3;i++) {
#     print A[i,1],A[i,2],A[i,3]
#     }
#   print "matrix b"
#   print b[1],b[2],b[3]
  Det=A[1,1]*A[2,2]*A[3,3]+A[1,2]*A[2,3]*A[3,1]+A[1,3]*A[2,1]*A[3,2]-A[1,3]*A[2,2]*A[3,1]-A[1,2]*A[2,1]*A[3,3]-
A[1,1]*A[2,3]*A[3,2]
  E=(b[1]*A[2,2]*A[3,3]+A[1,2]*A[2,3]*b[3]+A[1,3]*b[2]*A[3,2]-A[1,3]*A[2,2]*b[3]-A[1,2]*b[2]*A[3,3]-
b[1]*A[2,3]*A[3,2])/Det
  F=(A[1,1]*b[2]*A[3,3]+b[1]*A[2,3]*A[3,1]+A[1,3]*A[2,1]*b[3]-A[1,3]*b[2]*A[3,1]-b[1]*A[2,1]*A[3,3]-
A[1,1]*A[2,3]*b[3])/Det
  G=(A[1,1]*A[2,2]*b[3]+A[1,2]*b[2]*A[3,1]+b[1]*A[2,1]*A[3,2]-b[1]*A[2,2]*A[3,1]-A[1,2]*A[2,1]*b[3]-
A[1,1]*b[2]*A[3,2])/Det
#  E, F, and G are the coefficients in the plane z=Ex+Fy+G"
# make a function that tests the fit
  for (i=2;i<8;i++) {
    if (afplaneatoms[i]>.5) {
      k=afplaneatoms[i]
      distplane=(-oldarr[k,1]*E-oldarr[k,2]*F+oldarr[k,3]-G)/(E^2+F^2+1)^.5;if (distplane<0) distplane=-distplane
      aberror=aberror+distplane
      }
    }
# if the fit is bad, as can happen, decrease to 3 atoms in plane, using only the first three atoms in list after the conatom
  if (aberror>2) {
    print "original aberror",aberror >> "diagnostics"
    A[1,1]=0;A[1,2]=0;A[1,3]=0;A[2,1]=0;A[2,2]=0;A[2,3]=0;A[3,1]=0;A[3,2]=0;A[3,3]=0
    for (i=2;i<5;i++) {
      k=afplaneatoms[i]
      A[1,1]=A[1,1]+oldarr[k,1]^2;A[1,2]=A[1,2]+oldarr[k,1]*oldarr[k,2];A[1,3]=A[1,3]+oldarr[k,1]
      A[2,1]=A[2,1]+oldarr[k,1]*oldarr[k,2];A[2,2]=A[2,2]+oldarr[k,2]^2;A[2,3]=A[2,3]+oldarr[k,2]
      A[3,1]=A[3,1]+oldarr[k,1];A[3,2]=A[3,2]+oldarr[k,2];A[3,3]++
      b[1]=b[1]+oldarr[k,1]*oldarr[k,3];b[2]=b[2]+oldarr[k,2]*oldarr[k,3];b[3]=b[3]+oldarr[k,3]
      }
    Det=A[1,1]*A[2,2]*A[3,3]+A[1,2]*A[2,3]*A[3,1]+A[1,3]*A[2,1]*A[3,2]-A[1,3]*A[2,2]*A[3,1]-A[1,2]*A[2,1]*A[3,3]-
A[1,1]*A[2,3]*A[3,2]
    E=(b[1]*A[2,2]*A[3,3]+A[1,2]*A[2,3]*b[3]+A[1,3]*b[2]*A[3,2]-A[1,3]*A[2,2]*b[3]-A[1,2]*b[2]*A[3,3]-
b[1]*A[2,3]*A[3,2])/Det
    F=(A[1,1]*b[2]*A[3,3]+b[1]*A[2,3]*A[3,1]+A[1,3]*A[2,1]*b[3]-A[1,3]*b[2]*A[3,1]-b[1]*A[2,1]*A[3,3]-
A[1,1]*A[2,3]*b[3])/Det
    G=(A[1,1]*A[2,2]*b[3]+A[1,2]*b[2]*A[3,1]+b[1]*A[2,1]*A[3,2]-b[1]*A[2,2]*A[3,1]-A[1,2]*A[2,1]*b[3]-
A[1,1]*b[2]*A[3,2])/Det
```

```
    aberror=0
    for (i=2;i<5;i++) {
      k=afplaneatoms[i]
      distplane=(-oldarr[k,1]*E-oldarr[k,2]*F+oldarr[k,3]-G)/(E^2+F^2+1)^.5;if (distplane<0) distplane=-distplane
      aberror=aberror+distplane
      }
    }
  conatom=afplaneatoms[1]
  distplane=(-oldarr[conatom,1]*E-oldarr[conatom,2]*F+oldarr[conatom,3]-G)/(E^2+F^2+1)^.5
  if (distplane<0) distplane=-distplane
  t=(oldarr[conatom,1]*E+oldarr[conatom,2]*F-oldarr[conatom,3]+G)/(E^2+F^2+1)
  planepoint[1]=oldarr[conatom,1]-t*E;planepoint[2]=oldarr[conatom,2]-t*F;planepoint[3]=oldarr[conatom,3]+t
  distcmass=((planepoint[1]-cmass[1])^2+(planepoint[2]-cmass[2])^2+(planepoint[3]-cmass[3])^2)^.5
  print "distplane",distplane,"t",t,"planepoint",planepoint[1],planepoint[2],planepoint[3],"distcmass",distcmass,"aberror",aberror
>> "diagnostics"
  if (distcmass>1.4) {
    vector[1]=planepoint[1]-cmass[1];vector[2]=planepoint[2]-cmass[2];vector[3]=planepoint[3]-cmass[3]
    vector[1]=vector[1]*1.4/distcmass;vector[2]=vector[2]*1.4/distcmass;vector[3]=vector[3]*1.4/distcmass
    planepoint[1]=cmass[1]+vector[1];planepoint[2]=cmass[2]+vector[2];planepoint[3]=cmass[3]+vector[3];
    distcmass=((planepoint[1]-cmass[1])^2+(planepoint[2]-cmass[2])^2+(planepoint[3]-cmass[3])^2)^.5
    distplane=((planepoint[1]-oldarr[conatom,1])^2+(planepoint[2]-oldarr[conatom,2])^2+(planepoint[3]-
oldarr[conatom,3])^2)^.5
    print "new distplane",distplane,"newplanepoint",planepoint[1],planepoint[2],planepoint[3],"newdistcmass",distcmass >>
"diagnostics"
    }
  delX=oldarr[conatom,1]-planepoint[1];delY=oldarr[conatom,2]-planepoint[2];delZ=oldarr[conatom,3]-planepoint[3]
  if (applyforceplane==2) apforceplane=apforceplane*(distplane-apforceplaneX0)
  unitX=apforceplane*delX/distplane;unitY=apforceplane*delY/distplane;unitZ=apforceplane*delZ/distplane
  forceArr[conatom,1]=forceArr[conatom,1]-unitX;forceArr[conatom,2]=forceArr[conatom,2]-
unitY;forceArr[conatom,3]=forceArr[conatom,3]-unitZ
  for (i=2;i<8;i++) {
    if (afplaneatoms[i]>.5) {
      platom=afplaneatoms[i]

forceArr[platom,1]=forceArr[platom,1]+unitX/numplaneatoms;forceArr[platom,2]=forceArr[platom,2]+unitY/numplaneatoms;fo
rceArr[platom,3]=forceArr[platom,3]+unitZ/numplaneatoms
      }
    }
  }

function setinterlockingsphereatom() {
#apply force to one of a series of atoms, whichever is closest to afatom[1]
  for (i=3;i<8;i++) {
    if (afatom[i]>0) {
      if(Distance(afatom[1],afatom[i])<Distance(afatom[1],afatom[2])) afatom[2]=afatom[i]
      }
    if (afatomB[i]>0) {
      if(Distance(afatomB[1],afatomB[i])<Distance(afatomB[1],afatomB[2])) afatomB[2]=afatomB[i]
      }
    if (afatomC[i]>0) {
      if(Distance(afatomC[1],afatomC[i])<Distance(afatomC[1],afatomC[2])) afatomC[2]=afatomC[i]
      }
    }
  }

function doapplyforce() {
# applyforce 1 puts a linear constant force.
# applyforce 2 puts on a harmonic restoring force to apforceX0
  delX=oldarr[afatom[1],1]-oldarr[afatom[2],1];delY=oldarr[afatom[1],2]-oldarr[afatom[2],2];delZ=oldarr[afatom[1],3]-
oldarr[afatom[2],3];
  distatoms=(delX^2+delY^2+delZ^2)^.5
  if (applyforce==2) apforce=apforce*(distatoms-apforceX0)
  if (applyforce==3) apforce=apforce*(distatoms-apforceX0) + apforce2*(distatoms-apforceX0)^2
  if (applyforce==4) apforce=apforce*(distatoms-apforceX0) + apforce2*(distatoms-apforceX0)^2 + apforce3*(distatoms-
apforceX0)^3
  unitX=apforce*delX/distatoms;unitY=apforce*delY/distatoms;;unitZ=apforce*delZ/distatoms;
  forceArr[afatom[1],1]=forceArr[afatom[1],1]-unitX;forceArr[afatom[1],2]=forceArr[afatom[1],2]-
unitY;forceArr[afatom[1],3]=forceArr[afatom[1],3]-unitZ

forceArr[afatom[2],1]=forceArr[afatom[2],1]+unitX;forceArr[afatom[2],2]=forceArr[afatom[2],2]+unitY;forceArr[afatom[2],3]=
forceArr[afatom[2],3]+unitZ
  }

function doapplyforceB() {
  delX=oldarr[afatomB[1],1]-oldarr[afatomB[2],1];delY=oldarr[afatomB[1],2]-oldarr[afatomB[2],2];delZ=oldarr[afatomB[1],3]-
oldarr[afatomB[2],3];
  distatoms=(delX^2+delY^2+delZ^2)^.5
  if (applyforceB==2) apforceB=apforceB*(distatoms-apforceX0B)
  if (applyforceB==3) apforceB=apforceB*(distatoms-apforceX0B) + apforce2B*(distatoms-apforceX0B)^2
  if (applyforceB==4) apforceB=apforceB*(distatoms-apforceX0B) + apforce2B*(distatoms-apforceX0B)^2 +
apforce3B*(distatoms-apforceX0B)^3
  unitX=apforceB*delX/distatoms;unitY=apforceB*delY/distatoms;;unitZ=apforceB*delZ/distatoms;
  forceArr[afatomB[1],1]=forceArr[afatomB[1],1]-unitX;forceArr[afatomB[1],2]=forceArr[afatomB[1],2]-
unitY;forceArr[afatomB[1],3]=forceArr[afatomB[1],3]-unitZ

forceArr[afatomB[2],1]=forceArr[afatomB[2],1]+unitX;forceArr[afatomB[2],2]=forceArr[afatomB[2],2]+unitY;forceArr[afatom
B[2],3]=forceArr[afatomB[2],3]+unitZ
  }

function doapplyforceC() {
  delX=oldarr[afatomC[1],1]-oldarr[afatomC[2],1];delY=oldarr[afatomC[1],2]-oldarr[afatomC[2],2];delZ=oldarr[afatomC[1],3]-
oldarr[afatomC[2],3];
  distatoms=(delX^2+delY^2+delZ^2)^.5
  if (applyforceC==2) apforceC=apforceC*(distatoms-apforceX0C)
  if (applyforceC==3) apforceC=apforceC*(distatoms-apforceX0C) + apforce2C*(distatoms-apforceX0C)^2
  if (applyforceC==4) apforceC=apforceC*(distatoms-apforceX0C) + apforce2C*(distatoms-apforceX0C)^2 +
apforce3C*(distatoms-apforceX0C)^3
  unitX=apforceC*delX/distatoms;unitY=apforceC*delY/distatoms;;unitZ=apforceC*delZ/distatoms;
  forceArr[afatomC[1],1]=forceArr[afatomC[1],1]-unitX;forceArr[afatomC[1],2]=forceArr[afatomC[1],2]-
unitY;forceArr[afatomC[1],3]=forceArr[afatomC[1],3]-unitZ

forceArr[afatomC[2],1]=forceArr[afatomC[2],1]+unitX;forceArr[afatomC[2],2]=forceArr[afatomC[2],2]+unitY;forceArr[afatom
C[2],3]=forceArr[afatomC[2],3]+unitZ
  }

function doempiricaldispersion() {
  EdispHtotal=0
  s6=empiricaldispersion
  for (i=1;i<=numAtoms;i++) {
    if (atSym[i]=="H") {c6[i]=0.14;r0[i]=1.001}
#    if (atSym[i]=="H") {c6[i]=0.16;r0[i]=1.11}
    if (atSym[i]=="He") {c6[i]=0.08;r0[i]=1.012}
    if (atSym[i]=="Li") {c6[i]=1.61;r0[i]=0.825}
    if (atSym[i]=="Be") {c6[i]=1.61;r0[i]=1.408}
    if (atSym[i]=="B") {c6[i]=3.13;r0[i]=1.485}
    if (atSym[i]=="C") {c6[i]=1.75;r0[i]=1.452}
#    if (atSym[i]=="C") {c6[i]=1.65;r0[i]=1.61}
    if (atSym[i]=="N") {c6[i]=1.23;r0[i]=1.397}
#    if (atSym[i]=="N") {c6[i]=1.11;r0[i]=1.55}
    if (atSym[i]=="O") {c6[i]=0.70;r0[i]=1.342}
#    if (atSym[i]=="O") {c6[i]=0.70;r0[i]=1.49}
    if (atSym[i]=="F") {c6[i]=0.75;r0[i]=1.287}
    if (atSym[i]=="Ne") {c6[i]=0.63;r0[i]=1.243}
    if (atSym[i]=="Na") {c6[i]=5.71;r0[i]=1.144}
    if (atSym[i]=="Mg") {c6[i]=5.71;r0[i]=1.364}
    if (atSym[i]=="Al") {c6[i]=10.79;r0[i]=1.639}
    if (atSym[i]=="Si") {c6[i]=9.23;r0[i]=1.716}
    if (atSym[i]=="P") {c6[i]=7.84;r0[i]=1.705}
    if (atSym[i]=="S") {c6[i]=5.57;r0[i]=1.683}
    if (atSym[i]=="Cl") {c6[i]=5.07;r0[i]=1.639}
#     if (atSym[i]=="Cl") {c6[i]=8.00;r0[i]=1.82}
    if (atSym[i]=="Ar") {c6[i]=4.61;r0[i]=1.595}
    if (atSym[i]=="K") {c6[i]=10.8;r0[i]=1.485}
    if (atSym[i]=="Ca") {c6[i]=10.8;r0[i]=1.474}
    if (atSym[i]=="Sc") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Ti") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="V") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Cr") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Mn") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Fe") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Co") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Ni") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Cu") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Zn") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Ga") {c6[i]=16.99;r0[i]=1.65}
    if (atSym[i]=="Ge") {c6[i]=17.10;r0[i]=1.727}
    if (atSym[i]=="As") {c6[i]=16.37;r0[i]=1.76}
    if (atSym[i]=="Se") {c6[i]=12.64;r0[i]=1.771}
    if (atSym[i]=="Br") {c6[i]=12.47;r0[i]=1.749}
    if (atSym[i]=="Pd") {c6[i]=24.67;r0[i]=1.639}
    if (atSym[i]=="I") {c6[i]=31.5;r0[i]=1.892}
    r0[i]=r0[i]*radiusmultiplier
    }
  for (i=1;i<numAtoms;i++) {
    for (j=i+1;j<=numAtoms;j++) {
      Rij=Distance(i,j)
      if (Rij<8) { #cutoff of 8 angstroms to save time - later this can be parameterized
#need to get units of force to Hartrees/Bohr, since that is the units of forceArr
        EdispK=Edisp(c6[i],c6[j],Rij,r0[i],r0[j])*1E6/4184 #in kcal/mol
        FdispK=Fdisp(c6[i],c6[j],Rij,r0[i],r0[j])*1E6/4184 #in kcal/mol per angstrom
        EdispH=EdispK/627.509
        EdispHtotal=EdispHtotal+EdispH
        FdispHB=FdispK*0.52917725/627.509
#         print i,j,Cij,Rij,r0[i],r0[j],fdmp,"     ",EdispK,FdispK,"     ",EdispH,FdispHB,EdispHtotal
        delX=oldarr[i,1]-oldarr[j,1];delY=oldarr[i,2]-oldarr[j,2];delZ=oldarr[i,3]-oldarr[j,3];
        unitX=FdispHB*delX/Rij;unitY=FdispHB*delY/Rij;;unitZ=FdispHB*delZ/Rij;
        forceArr[i,1]=forceArr[i,1]+unitX;forceArr[i,2]=forceArr[i,2]+unitY;forceArr[i,3]=forceArr[i,3]+unitZ
        forceArr[j,1]=forceArr[j,1]-unitX;forceArr[j,2]=forceArr[j,2]-unitY;forceArr[j,3]=forceArr[j,3]-unitZ
        }
      }
    }
  newPotentialE=newPotentialE+EdispHtotal
  }

function Edisp(c6i,c6j,rij,r0i,r0j) {
  cij=(c6i*c6j)^.5
  fdmp=1/(1+exp(-20*((rij/(r0i+r0j)-1))))
  return -s6*cij*fdmp/(rij^6)
  }

function Fdisp(c6i,c6j,rij,r0i,r0j) {
  delta=0.001
  return (Edisp(c6i,c6j,rij-delta,r0i,r0j)-Edisp(c6i,c6j,rij+delta,r0i,r0j))/(2*delta)
  }

function dozeroatom() {
```

```
#routine to slowly move an atom toward the origin as set by a harmonic potential
  multiple=0.99996
  oldarr[zeroatom,1]=multiple*oldarr[zeroatom,1]
  oldarr[zeroatom,2]=multiple*oldarr[zeroatom,2]
  oldarr[zeroatom,3]=multiple*oldarr[zeroatom,3]
  }

function dothermostat() {
#print out some things to vellist and do thermostat
# the damping in the thermostat is based on temperature based on old geo vs older geo
  if (diag<4) print "KEold",KEold,"desired temperature",temp,"apparent Temperature",apparentTemp >> "vellist"
  if (apparentTemp>temp) damping=thermostatmult
  if (apparentTemp<temp) damping=1/thermostatmult
  }

function doDRP() {
#routine for DRPs
  maxForce=0;oscillTest=0
  for (i=1;i<=numAtoms;i++) {
    for (j=1;j<=3;j++) {
# conversions here take force to J/angstrom, 1E20 converts to kg angstroms / s^2, then mult time (s^s) and divide by weight in kg
to get angstroms
      forceArr[i,j]=1E20*forceArr[i,j]*627.509*(4184/(0.529177*avNum))*(timestep^2)/(weight[i]/(avNum*1000))
      oscillTest=oscillTest+forceArr[i,j]*oldForce[i,j]
      if (forceArr[i,j]>maxForce) maxForce=forceArr[i,j]
      if ((0-forceArr[i,j])>maxForce) maxForce=-forceArr[i,j]
      }
    if (i==1) printf("% .8f % .8f % .8f \n",forceArr[1,1],forceArr[1,2],forceArr[1,3])  > "oldAdjForces"
    if (i>1) printf("% .8f % .8f % .8f \n",forceArr[i,1],forceArr[i,2],forceArr[i,3])  >> "oldAdjForces"
    }
  print "oscillTest ",oscillTest >> "oldAdjForces"
  if (oscillTest<0) {
    maxAtomMove = maxAtomMove*0.5
    print maxAtomMove > "maxMove"
    }
  if (oscillTest>0) {
    maxAtomMove = maxAtomMove*1.2
    print maxAtomMove > "maxMove"
    }
  print "maxAtomMove ",maxAtomMove >> "oldAdjForces"
  forceMult=maxAtomMove/maxForce
  for (i=1;i<=numAtoms;i++) {
    for (j=1;j<=3;j++) {
      newarr[i,j]=oldarr[i,j]+forceMult*forceArr[i,j]
      }
    }
  }

function doVerlet() {
#normal routine for Verlet
  for (i=1;i<=numAtoms;i++) {
    for (j=1;j<=3;j++) {
# conversions here take force to J/angstrom, 1E20 converts to kg angstroms / s^2, then mult time (s^s) and divide by weight in kg
to get angstroms
      forceArr[i,j]=1E20*forceArr[i,j]*627.509*(4184/(0.529177*avNum))*(timestep^2)/(weight[i]/(avNum*1000))
#       if ((diag>1) && (i==1)) print "i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3]" >> "diagnostics"
#       if (diag>1) print i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3] >> "diagnostics"
      newarr[i,j]=oldarr[i,j]+damping*(oldarr[i,j]-olderarr[i,j])+forceArr[i,j]
      if ((i==fixedatom1) || (i==fixedatom2) || (i==fixedatom3) || (i==fixedatom4)) newarr[i,j]=oldarr[i,j]
#turn around atoms outside the box
      if (boxon==1) {
        if (newarr[i,j]>boxsize) if (oldarr[i,j]>olderarr[i,j]) newarr[i,j]=oldarr[i,j]+damping*(olderarr[i,j]-oldarr[i,j])+forceArr[i,j]
        if (newarr[i,j]<-1*boxsize) if (oldarr[i,j]<olderarr[i,j]) newarr[i,j]=oldarr[i,j]+damping*(olderarr[i,j]-
oldarr[i,j])+forceArr[i,j]
        }
      }
    }
#calculate the kinetic energy.  This is for the point prior to the current point.  Corrected calculation
  for (at=1;at<=numAtoms;at++) {
    atomVel=(((olderarr[at,1]-newarr[at,1])^2 + (olderarr[at,2]-newarr[at,2])^2 +(olderarr[at,3]-newarr[at,3])^2)^.5)/2
    KEnew=KEnew+0.5*weight[at]*(atomVel^2)/((timestep^2)*conver1)
    }
    KEave=KEnew # we used to average with KEold but that is incorrect.  This is better.
    Etotal=newPotEK+KEave
#still basing apparent Temperature on velocities from old vs older, even though the KE now represents an average of old and new
    if (diag==4) print runpointnum,"KEave",KEave,"apparent
Temperature",apparentTemp,"newPotEK",newPotEK,"Etotal",Etotal,"pressure in Atm",pressureAtm,"density in 0.9r",density >>
"vellist"
  }

function writetraj() {
  print numAtoms >> "traj"
  print newPotentialE,title1,title2,title3,title4,"runpoint ",runpointnum,"runisomer ",isomernum >> "traj"
  for (i=1;i<=numAtoms;i++) {
    printf("%s %.7f %.7f %.7f",atSym[i],newarr[i,1],newarr[i,2],newarr[i,3]) >> "traj"
    print "" >> "traj"
    }
  }

function writecomfile() {
  print "%nproc=" processors
  print "%mem=" memory
  if (killcheck!=1) print "%chk=" checkpoint
  if (nonstandard==0) {
    print "#p " method " force scf=(xqc,maxconven=155,fulllinear,nosym) "
    if (meth2=="unrestricted") print "guess=mix" #for unrestricted calculations
    if (meth2=="read") print "guess=tcheck" #for reading orbitals from check, sometimes faster, sometimes not
    if (length(meth3)>2) print meth3
    if (length(meth4)>2) print meth4
    }
  if (nonstandard==1) {
    print "# "
    print "nonstd"
    system("cat nonstandard")
    }
  print ""
  print  title1,title2,title3,title4
  print "runpoint ",runpointnum
  print "runisomer ",isomernum
  if (DRP==1) {
    print "maxForce and forceMult and maxAtomMove",maxForce,forceMult,maxAtomMove
    print "maxForce and forceMult and maxAtomMove",maxForce,forceMult,maxAtomMove >> "movelist"
    }
  print ""
  if (oniommult==0) print charge,multiplicity
  if (oniommult>0) print charge,multiplicity,oniomcharge,oniommult
  for (i=1;i<=numAtoms;i++) {
    printf("%s %.7f %.7f %.7f",atSym[i],newarr[i,1],newarr[i,2],newarr[i,3])
    if ((i>highlevel) && (i<=highlevel+linkatoms)) printf(" %s","M H")
    if (i>(highlevel+linkatoms)) printf(" %s","M")
    print ""
    }
  print ""
  if (length(meth5)>2) print meth5
  if (length(meth6)>2) print meth6
  if (methodfilelines>=1) {
    for (i=1;i<=methodfilelines;i++) {
       getline < "methodfile"
       print $0
       }
    }
  if ((nmrtype>0) && (nmrdo==1)) {
    print "--link1--"
    print "%nproc=" processors
    print "%mem=" memory
    print "%chk=" checkpoint
    print "# " nmrmethod " nmr=giao geom=check"
    if (nmrmethod==method) print "guess=tcheck"
    if (length(meth7)>2) print meth7
    print ""
    print title1,title2,title3,title4
    print "runpoint ",runpointnum
    print "runisomer ",isomernum
    print ""
    print charge,multiplicity
    }
  print ""
  if ((nmrtype>1) && (nmrdo==1)) {
    print "--link1--"
    print "%nproc=" processors
    print "%mem=" memory
    print "%chk=" checkpoint
    print "# " nmrmethod2 " nmr=giao geom=check"
    if (length(meth7)>2) print meth7
    print ""
    print title1,title2,title3,title4
    print "runpoint ",runpointnum
    print "runisomer ",isomernum
    print ""
    print charge,multiplicity
    }
  print ""
  if ((nmrtype>2) && (nmrdo==1)) {
    print "--link1--"
    print "%nproc=" processors
    print "%mem=" memory
    print "%chk=" checkpoint
    print "# " nmrmethod3 " nmr=giao geom=check"
    if (length(meth7)>2) print meth7
    print ""
    print title1,title2,title3,title4
    print "runpoint ",runpointnum
    print "runisomer ",isomernum
    print ""
    print charge,multiplicity
    }
  print ""
  if ((nmrcc==1) && (nmrdo==1)) {
    print "CCSD(T) NMR calculation" > "ZMAT"
    for (i=1;i<=numAtoms;i++) {
       printf("%s %.7f %.7f %.7f",atSym[i],newarr[i,1],newarr[i,2],newarr[i,3]) >> "ZMAT"
       print "" >> "ZMAT"
       }
    print "" >> "ZMAT"
    print "*ACES2(CALC=CCSD[T],PROP=NMR,BASIS=dzp" >> "ZMAT"
    print "ABCDTYPE=AOBASIS,TREAT_PERT=SEQUENTIAL,CC_PROG=ECC" >> "ZMAT"
```

```
    print "COORD=CARTESIAN" >> "ZMAT"
    print "MEM_UNIT=GB,MEMORY=2)" >> "ZMAT"
    print "" >> "ZMAT"
    }
  }
```

Program progcfour

This module calls an outside calculation with CFOUR but was not used for the current paper and its presence is not needed to reproduce the current work. Its listing was published in the SI for a previous paper: Biswas, B.; Singleton, D. A. *J. Am. Chem. Soc.* **2015**, *137*, 14244-14247.

Program randgen.c

This is compiled before use to give the service program *randgen*

```
#include <stdio.h>
#include <stdlib.h>

int a,b,c;
double d;

int product(int x, int y);

int main(void)
{
  int count=1;
  srand48(time (0));
  while (count<=100000)
  {
    d = drand48();
    printf ("%.20f\n", d);
    count++;
  }
  return 0;
}
```

Program proganal used for 53-THF product-forming trajectories (using PROGDYN)

```
BEGIN {
firsttitle=1
getline < "isomernumber"
isomer=$1
}
/ hydroboration/ {
  if (firsttitle==1) {
    printf("%s %s %s %s %s %s %s  ",$1,$2,$3,$4,$6,$7,$8)
    runpoint=$6
    }
  firsttitle++
  }
/Standard orientation/,/Rotational constants/ {
  if (($1>.5) && ($1<99)) {
    A[$1]=$4;B[$1]=$5;C[$1]=$6
    }
  }
#/before annihilation/ {
#   printf("%s %.5f ",$1,$6)
#   }

END {
BC1=Distance(1,2)
BC2=Distance(1,3)
```

```
HC1=Distance(14,2)
if (Distance(15,2)<HC1) HC1=Distance(15,2)
if (Distance(16,2)<HC1) HC1=Distance(16,2)
HC2=Distance(14,3)
if (Distance(15,3)<HC2) HC2=Distance(15,3)
if (Distance(16,3)<HC2) HC2=Distance(16,3)
  printf("%s %.3f %s %.3f %s %.3f %s %.3f ","BC1",BC1,"BC2",BC2,"HC1",HC1,"HC2",HC2)
  if (runpoint>500000) {
    print "Too many points.  XXXXN"
#     system("date > nogo")
    }
  if ((HC1<1.1) && (BC2<1.6)) {
    print "Mark product formed XXXX"
    }
  if ((HC2<1.1) && (BC1<1.6)) {
    print "Antimark product formed XXXX"
    }
  if ((HC1>4.5) && (HC2>4.5)) {
    print "Returning to separate SMs XXXX"
    }

  system("date '+%b:%d:%Y %T'")
  system("tail -1 Echeck | grep XXXX")
  }

function Distance(Atom1,Atom2) {
 return sqrt((A[Atom1]-A[Atom2])^2 + (B[Atom1]-B[Atom2])^2 + (C[Atom1]-C[Atom2])^2)
}

function Angle(Atom1,Atom2,Atom3) {
  value=((-
Distance(Atom1,Atom3)^2+Distance(Atom1,Atom2)^2+Distance(Atom2,Atom3)^2)/(2*Distance(Atom1,Atom2)*Distance(Ato
m2,Atom3)))
  return acos(value)
}

function asin(x) { return (180/3.141592)*atan2(x, sqrt(1-x*x)) }

function acos(x) { return (180/3.141592)*atan2(sqrt(1-x*x), x) }

function atan(x) { return (180/3.141592)*atan2(x,1) }

function Dihedral(Atom1,Atom2,Atom3,Atom4) {
  B1x=A[Atom2]-A[Atom1]
  B1y=B[Atom2]-B[Atom1]
  B1z=C[Atom2]-C[Atom1]
  B2x=A[Atom3]-A[Atom2]
  B2y=B[Atom3]-B[Atom2]
  B2z=C[Atom3]-C[Atom2]
  B3x=A[Atom4]-A[Atom3]
  B3y=B[Atom4]-B[Atom3]
  B3z=C[Atom4]-C[Atom3]
  modB2=sqrt((B2x^2)+(B2y^2)+(B2z^2))
# yAx is x-coord. etc of modulus of B2 times B1
  yAx=modB2*(B1x)
  yAy=modB2*(B1y)
  yAz=modB2*(B1z)
# CP2 is the crossproduct of B2 and B3
  CP2x=(B2y*B3z)-(B2z*B3y)
  CP2y=(B2z*B3x)-(B2x*B3z)
  CP2z=(B2x*B3y)-(B2y*B3x)
  termY=((yAx*CP2x)+(yAy*CP2y)+(yAz*CP2z))
```

```
# CP is the crossproduct of B1 and B2
   CPx=(B1y*B2z)-(B1z*B2y)
   CPy=(B1z*B2x)-(B1x*B2z)
   CPz=(B1x*B2y)-(B1y*B2x)
   termX=((CPx*CP2x)+(CPy*CP2y)+(CPz*CP2z))
  dihed4=(180/3.141592)*atan2(termY,termX)
  return dihed4
}

function killdyn(isomer) {
   system("rm -f dyn")
}
```

Program progdynsam

```
BEGIN {
temp=298.15
if (pt<1) startpoint=6363
if (pt>1) startpoint=pt
line=0
printon=0
pointline=0
secondpoint=0
}

{
line++
if (line==1) numAtoms=$1
pointline++
if (printon==1) {
   if ($1=="H") atWeight[pointline]=1.00783
   if ($1=="C") atWeight[pointline]=12.0000
   if ($1=="B") atWeight[pointline]=10.81
   if ($1=="O") atWeight[pointline]=15.99940
   if ($1=="F") atWeight[pointline]=18.9984
   if ($1=="P") atWeight[pointline]=30.9738
   if ($1=="S") atWeight[pointline]=31.972
   if ($1=="N") atWeight[pointline]=14.0030740
   if ($1=="Al") atWeight[pointline]=26.981
   if ($1=="Cl") atWeight[pointline]=35.4527
   if (($1=="C") || ($1=="H") || ($1=="O") || ($1=="P") || ($1=="N") || ($1=="Cl")  || ($1=="B") || ($1=="F") || ($1=="S") ||
($1=="Al")) {
      Arr0[pointline,0]=$1
      Arr0[pointline,1]=$2
      Arr0[pointline,2]=$3
      Arr0[pointline,3]=$4
      Arr1[pointline,0]=$1
      Arr1[pointline,1]=$2
      Arr1[pointline,2]=$3
      Arr1[pointline,3]=$4
      }
   }
if (secondpoint==1) {
   if (($1=="C") || ($1=="H") || ($1=="O") || ($1=="P") || ($1=="N") || ($1=="Cl")  || ($1=="B") || ($1=="F") || ($1=="S") ||
($1=="Al")) {
      Arr1[pointline,1]=$2-Arr1[pointline,1]
      Arr1[pointline,2]=$3-Arr1[pointline,2]
      Arr1[pointline,3]=$4-Arr1[pointline,3]
      }
   }
if ($8=="runisomer") {
   pointline=0
   }
if ($7==startpoint) {
   pointline=0
   printon=1
   }
if ($7==startpoint+1) {
   secondpoint=1
   pointline=0
   printon=0
   }
if ($7>startpoint+1) {
   secondpoint=0
   printon=0
   }

}

END {
print numAtoms
for (i=1;i<=numAtoms;i++) {
   print Arr0[i,0],Arr0[i,1],Arr0[i,2],Arr0[i,3],atWeight[i]
   }
conver1=4.184E26
srand()
timestep=1E-15
for (i=1;i<=100;i++) {
   newRand=rand()
   newRand2=rand()
   randArr[i]=newRand
   sign[i]=1
   if (newRand2<0.5) sign[i]=-1
}
for (i=1;i<=numAtoms;i++) {
   if ((Arr1[i,1]==0) && (Arr1[i,2]==0) && (Arr1[i,3]==0)) {
      for (j=1;j<=3;j++) {
        KE=-0.001987*temp*log(1-randArr[3*i+j])
        Vel=sign[3*i+j]*timestep*(2*KE*conver1/atWeight[i])^0.5
        Arr1[i,j]=Vel
        }
      }
   print Arr1[i,1],Arr1[i,2],Arr1[i,3]
   }
printf("%s %i %i %s ","generated from points",startpoint,startpoint+1,"in a trajectory, so no modes to print out")
system("pwd")
print "Total mode energy desired=",0
for (i=1;i<=700;i++) {
   velsq= Arr1[i,1]^2+Arr1[i,2]^2+Arr1[i,3]^2
   KE=1E30*0.5*atWeight[i]*velsq/conver1
   KE=2.388E3*0.5*atWeight[i]*velsq
   #print i,KE
   }
print ""
}
```

Program proglookstart

```
BEGIN {
dirnum=666
}
{
```

```
if (($5 % 250)==0) {
        print "cd /scratch/user/johnathanbailey13/"substr(FILENAME,1,4)
        print "mkdir /scratch/user/johnathanbailey13/n"dirnum
        print "cp /scratch/user/johnathanbailey13/progdyn.conf /scratch/user/johnathanbailey13/n"dirnum
        print "echo 3 > /scratch/user/johnathanbailey13/n"dirnum"/bypassproggen"
        print "awk -v pt="$5,"-f ~/progdynsam traj > /scratch/user/johnathanbailey13/n"dirnum"/geoPlusVel"
        dirnum++
  }
}
```

Program progKECM

```
BEGIN {
structure=0
point=0
}
/runpoint/ {
  oldpoint=point
  point=$7
  if ((point<2) || (point>99999999)) point=1
  linenumber=-1
# array A is three points back, B is two points back, and C is latest point
# calculation is done at the beginning of the new point, and once at the end
# D is the array of distances traveled by each atom between points A and B
  KE1=0;
  for (i=1;i<=22;i++) {
    D[i]=.5*((C[i,1]-A[i,1])^2+(C[i,2]-A[i,2])^2+(C[i,3]-A[i,3])^2)^.5
    KE1=KE1+.5*atWeight[i]*D[i]^2
    }
  KE1=KE1/6.02E26
  KE1=KE1/1E20
  KE1=KE1*1E30
  KE1=KE1*6.02E23
  KE1=KE1/4184
  if (point==2) print "                              "
  if (point>=4) print oldpoint-1,"          ",KE1
  if (point<oldpoint) {
    print oldpoint-1,"    ",KE1
    print "                                        "
    }
  for (i=1;i<=22;i++) {
    for (j=1;j<=3;j++) {
      A[i,j]=B[i,j]
      B[i,j]=C[i,j]
      }
    }
  }
{
linenumber++
if ((linenumber>0) && (linenumber < 23)) {
  i=linenumber
  if ($1=="H") atWeight[i]=1.00783
  if ($1=="B") atWeight[i]=10.811
  if ($1=="C") atWeight[i]=12.
  if ($1=="O") atWeight[i]=15.9994
  C[i,1]=$2;C[i,2]=$3;C[i,3]=$4
  }
molmass=0
CMtot[1]=0;CMtot[2]=0;CMtot[3]=0
if (linenumber==23) {
  for (i=1;i<=23;i++) {
    for (j=1;j<=3;j++) {
      CM[i,j]=C[i,j]*atWeight[i]
      CMtot[j]=CM[i,j]+CMtot[j]
      }
    molmass=molmass+atWeight[i]
    }
  for (j=1;j<=3;j++) {
    CMtot[j]=CMtot[j]/molmass
    }
  for (i=1;i<=23;i++) {
    for (j=1;j<=3;j++) {
      C[i,j]=C[i,j]-CMtot[j]
      }
    }
  }
}
END {
  KE1=0;
  for (i=1;i<=23;i++) {
    D[i]=.5*((C[i,1]-A[i,1])^2+(C[i,2]-A[i,2])^2+(C[i,3]-A[i,3])^2)^.5
    KE1=KE1+.5*atWeight[i]*D[i]^2
    }
  KE1=KE1/6.02E26
  KE1=KE1/1E20
  KE1=KE1*1E30
  KE1=KE1*6.02E23
  KE1=KE1/4184
  if (point>=4) print point-1,"  ",KE1

}
```

progEchangelocal example

```
awk 'BEGIN {
section=0
}
/XXX/ {
section++
}
/product/ {
sectionchoice=section;finalpt=$5
}
END {
print sectionchoice,2,21,finalpt-40,finalpt-21
}' dynfollowfile > tempptlist
pwd >> tempptlist
#
awk 'BEGIN {
getline < "tempptlist";goodsection=$1;ptA=$2;ptB=$3;ptC=$4;ptD=$5
getline < "tempptlist";dir=$1
point=0;oldpoint=0;section=1
}
/runpoint/ {
linenumber=-1
if ($5=="runpoint") point=$6
if ($6=="runpoint") point=$7
if (point<oldpoint) section++
oldpoint=point
if (section==goodsection) {
  if (((point>=ptA) && (point<=ptB)) || ((point>=ptC) && (point<=ptD))) {
    if (point>ptA) print ""
    print "--link1--"
    print "%chk=g09.chk"
    print "#p SP B3LYP/6-31G*"
    print "  "
    print dir,"energy point",point
    print ""
    print "0 1"
    }
  }
}
{
linenumber++
if (section==goodsection) {
  if (((point>=ptA) && (point<=ptB)) || ((point>=ptC) && (point<=ptD))) {
    if ((linenumber>=1) && (linenumber<=4)) print
    if ((linenumber>=9) && (linenumber<=16)) print
    if (linenumber==4) {
      X1=$2;Y1=$3;Z1=$4
      }
    if (linenumber==5) {
      X2=$2;Y2=$3;Z2=$4
      CCdist=((X2-X1)^2+(Y2-Y1)^2+(Z2-Z1)^2)^.5
      ratio=1.09/CCdist
      X3=ratio*(X2-X1)+X1;Y3=ratio*(Y2-Y1)+Y1;Z3=ratio*(Z2-Z1)+Z1;
      print "H",X3,Y3,Z3
      }
    }
  }
}
END {
print ""
#print goodsection,ptA,ptB,ptC,ptD
}' traj?
```

ProgEchangetail example

```
awk 'BEGIN {
getline < "tempptlist";goodsection=$1;ptA=$2;ptB=$3;ptC=$4;ptD=$5
getline < "tempptlist";dir=$1
point=0;oldpoint=0;section=1
}
/runpoint/ {
linenumber=-1
if ($5=="runpoint") point=$6
if ($6=="runpoint") point=$7
if (point<oldpoint) section++
oldpoint=point
if (section==goodsection) {
  if (((point>=ptA) && (point<=ptB)) || ((point>=ptC) && (point<=ptD))) {
    if (point>ptA) print ""
    print "--link1--"
    print "%chk=g09.chk"
    print "#p SP B3LYP/6-31G*"
    print "  "
    print dir,"energy point",point
    print ""
    print "0 1"
    }
  }
}
{
linenumber++
if (section==goodsection) {
  if (((point>=ptA) && (point<=ptB)) || ((point>=ptC) && (point<=ptD))) {
    if ((linenumber>=5) && (linenumber<=8)) print
    if ((linenumber>=17) && (linenumber<=22)) print
    if (linenumber==4) {
      X2=$2;Y2=$3;Z2=$4
      }
    if (linenumber==5) {
      X1=$2;Y1=$3;Z1=$4
      CCdist=((X2-X1)^2+(Y2-Y1)^2+(Z2-Z1)^2)^.5
      ratio=1.09/CCdist
      X3=ratio*(X2-X1)+X1;Y3=ratio*(Y2-Y1)+Y1;Z3=ratio*(Z2-Z1)+Z1;
      print "H",X3,Y3,Z3
      }
    }
  }
}
END {
print ""
#print goodsection,ptA,ptB,ptC,ptD
}' traj?
```

Program progcomptime

```
BEGIN {
compformed=0
}
{
if ($5<5) compformed=0
if ((compformed==0) && (($9<2) && ($11<2))) {
    compformed=1
    formedat=$5
    }
}
/Antimark/ {
print "Antimark:   complex formed at  ",formedat,"   and finished at  ",$5,"   difference:    ",$5-formedat
compformed=0
formedat=0
}
/Mark/ {
print "Mark:       complex formed at  ",formedat,"   and finished at  ",$5,"   difference:    ",$5-formedat
compformed=0
formedat=0
}
```

Program Suite ProgdynONIOM

ProgdynONIOM is a new program adapted from PROGDYN but designed to combine forces from Gaussian09 and MOPAC2016 in ONIOM calculations. Its ONIOM capabilities are currently limited to systems where there is no covalent bond between the model (small) system and the real (full) system. The combination of the two programs extends the practical size of the

QM/QM calculation, aside from allowing the extended semi-empirical methods in MOPAC2016.

PROGDYN and ProgdynONIOM are similar, and some of the description below is identical to that above, but there are differences to note. In particular, ProgdynONIOM runs three separate QM calculations for each trajectory point, with a separate input and output file for each.

ProgdynONIOM includes the ability to include additional empirical dispersion using Grimme's D2 model (reference 22 in the main text). The progdyn.conf employed for the 80-THF model includes additional empirical dispersion by the use of the parameters empiricaldispersion .7 and radiusmultiplier 1.6. The PM6-D3H4 method already includes dispersion, but this additional small corrections was added to bring the solution to an accurate density at pressures less than 1000 atm.

A full listing of the subprograms of ProgdynONIOM is given below. To allow the reader to understand or make use of ProgdynONIOM , we first describe the overall structure of the program.

The master control program for dynamics, in the form of a Unix Shell Script, is called progdynONIOM. For a user to start to use progdynONIOM, some early lines in it that assign the scratch space and the location of the program files and input files would have to be modified for the local environment. These lines are between lines 45 and 55 and should be apparent. The location of the scratch space is usually passed to progdynONIOM as a parameter.

progdynONIOM takes as input files:
freqinHP - This is the standard output from a Gaussian 98, 03, or 09 frequency calculation using freq=hpmodes. For isotopically labeled compounds, use freq=(hpmodes,readisotopes).
progdyn.conf – This is a file giving a variety of configuration options, called on by many of the subprograms. progdyn.conf examples are listed below and contains explanations of the program options.
isomernumber - A number in file isomernumber provides a start for numbering runs. The default is 1.
detour – A signal file that, by existing, signals the program to do a side calculations
nogo – A signal file that, by existing, signals the program to stop between points
bypassproggen - A signal file that, by existing, signals the program to use a supplied input file *geoPlusVel* instead of generating one for itself. This pathway for initialization is important here because it is used when the program progdynsam, described later, is used to generate the geoPlusVel file.
methodfile – A file that contains lines to be added to the end of each g09.com input file, such as lines that call for an NMR calculation
cannontraj – A file containing a vector for each atom, used to fire an initial geometry in a particular direction.

progdynONIOM calls the following programs:
proggenHP - An awk program that starts a trajectory, giving each mode its zero point energy (if a quasiclassical calculation) plus random additional excitations depending on the temperature.
prog1stpoint – Awk program that creates the first Gaussian input file for each run
prog2ndpoint – Awk program that creates the second Gaussian input file for each run. prog2ndpoint also checks the energy of the first point to see if it fits with the desired energy, and aborts the run if it does not by creating appropriate output in file Echeck
progdynb – Creates subsequent Gaussian input files until run is completed, written in awk
proganal – A program to analyze the latest point and see if a run is done. This program must be redone for each new system. Elaborate changes are sometimes programmed into proganal, such as the automatic changing of configuration variables. proganal creates the output to dynfollowfile and NMRlist or NMRlistdis
randgen – A program that generates random numbers between 0 and 1. These are generated all at once and stored in a file for use by proggenHP.

progdynONIOM has the following output files:
isomernumber – A running tab of the trajectory number
runpointnumber – a running tab of the point in the trajectory
Echeck – output form where prog2ndpoint checks the energy of the trajectory to see if it fits with the desired energy
geoRecord – A record of all of the geoPlusVel files.

geoPlusVel – Created by proggen, this gives the starting positions, velocities, isotopic masses, excitations of the normal modes, and initial displacements of the normal modes for current run.

g09.com – Created by prog1stpoint, prog2ndpoint, and progdynb, this is the latest input file for Gaussian09 for current run and latest point.

model.mop - Created by prog1stpoint, prog2ndpoint, and progdynb, this is the latest input file for the model system for MOPAC2016, for current run and latest point.

real.mop - Created by prog1stpoint, prog2ndpoint, and progdynb, this is the latest input file for the real (full) system for MOPAC2016, for current run and latest point.

g09.log, real.out, model.out – These are the output files for the latest Gaussian09 and MOPAC 2016 calculations, used for extracting forces in the creation of the next point.

olddynrun, *olddynrun2, olddynrun3* – files containing the last three outputs from MOPAC2016, for creation of the next point

traj, traj1, traj2, traj3, etc. – files containing the geometries and energies for each trajectory, numbered by the isomernumber, in a format suitable for reading by Molden.

dynfollowfile – A short record of the runs and their results. The data desired for *dynfollowfile* must be programed into the script *proganal* as needed for each system studied.

skipstart - A signal file that, by existing, tells progdynstarterHP that we are in the middle of a run. For trajectories that are propagated forward and backward in time, skipstart keeps track of whether one is in the forward or reverse part.

diagnostics – optional output that follows which subprograms are running and configuration variables, decided by variable in progdyn.conf

vellist – optional output that lists the velocities of each atom, decided by variable in progdyn.conf, or lists the total kinetic energy in the system and the classical temperature, often also keeps track of the density

A number of files starting with '*temp*' are created then later erased.

The helper programs *progdynsam*, *proglookstart*, *progKECM*, and *progcomptime* listed above for PROGDYN were also employed with ProgdynONIOM.

progdynONIOM

```
#!/bin/bash
#progdynMOPAC started November 6, 2016
#LIMITATIONS -
#
#                               OUTLINE
# A. initialize
#    start outermost loop L1L1L1L1L1L1L1L1
#    start loop 2 L2L2L2L2L2L2L2L2L2
# B. branch on whether there is a file named "skipstart"
#    if there is, skip B1, B2, B3 entirely
#    if no file named "skipstart" then generate a new isomer.
#  B1. generates geoPlusVel, isomernumber, runpointnumber=1, then makes real.mop, model.mop, g09.com for point 1
#  B2. checks for success of B1 or exits, then sets geoRecord then runs real.mop, model.mop, g09.com.  Checks for their success
or exits.
#  B3. runpointnumber=2, kills run if XXXX in dynfollowfile due to bad energy probably, makes real.mop, model.mop, g09.com
for point 2, runs them
#      checks for their success or exits
#      runpointnumber=3, uses progdynb to make real.mop, model.mop, g09.com for point 3
#      sets skipstart=forward
#  B4. Only runs if skipstart=reverserestart, performs reverserestart, ends with runpointnumber=3
#      sets skipstart=reverse
# C. loop over propagation steps
#
#
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
#
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
#
```

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
#origdir, randdir, scratchdir, g09root, logfile all may need varied from system to system and assigned here or by program calling
this one
export LC_ALL=C
export MOPAC_LICENSE='/home/d-singleton/mopac'
export LD_LIBRARY_PATH='/home/d-singleton/mopac'
echo $1
scratchdir=$1
export g09root=/software/lms/g09_D01
. $g09root/g09/bsd/g09.profile
cp /home/d-singleton/mopac/MOPAC2016.exe $scratchdir
cp /home/d-singleton/mopac/lib* $scratchdir
export LD_LIBRARY_PATH=$scratchdir
origdir=`pwd`
cd $origdir
logfile=docslog
randdir=~/bin
proggramdir=/scratch/user/d-singleton/binoniom1100
freqfile=/scratch/user/d-singleton/binoniom1100/freqinHP
echo
echo ORIGDIR at the beginning of run:
echo $origdir
ls $origdir
echo
echo SCRATCHDIR at the beginning of run:
echo $scratchdir
ls $scratchdir
echo
echo PROGGRAMDIR at the beginning of run::
echo $proggramdir
ls $proggramdir

rm -f nogo    # assume that if someone is starting a job, they want it to go.
rm -f diagnostics goingwell tempdone # diagnostics contains extra info from previous runs, other two files are from older
versions of progdyn
if (test -s g09.com) then
  sed -i '/guess=tcheck/d' g09.com      # no chk file on first point
fi

#### Triple 'while' loop - will have to break multiple times to get out, but advantage is ability to control starting over
#L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1
L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1L1
while (true)
do
# As long as there is a file "goingwell" the program will not exit entirely by itself
rm -f $scratchdir/goingwell
#L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2
L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2L2
while (true)
do
#  BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
  if (test -f skipstart) then
    echo "skipping start and continuing from previous runs"
  else
#  B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1 generate geoPlusVel and first input file
    if [ `cat runpointnumber` = "1" ]; then
      echo "X did not complete first point so new isomer started" >> dynfollowfile
    fi
    if [ `cat runpointnumber` = "2" ]; then
      echo "X did not complete second point so new isomer started" >> dynfollowfile
    fi
    if [ `cat runpointnumber` = "3" ]; then
      echo "X did not complete third point so new isomer started" >> dynfollowfile
    fi
    cd $origdir
    if (test -f bypassproggen) then
      echo "taking starting conditions from pre-generated geoPlusVel"
    else
      $randdir/randgen > temp811
# the next 8 lines would have to be changed to use low-precision modes
      awk '/        1         2/,/Harmonic frequencies/ {print}' $freqfile > temp401
      awk '/Frequencies --/ {print $3;print $4;print $5;print $6;print $7}' temp401 > tempfreqs
      awk '/Reduced masses/ {print $4;print $5;print $6;print $7;print $8}' temp401 > tempredmass
      awk '/Force constants/ {print $4;print $5;print $6;print $7;print $8}' temp401 > tempfrc
      awk '/0/ && ((length($1) < 2) && ($1 < 4)) {print}' temp401 > tempmodes
      awk '/has atomic number/ {print}' $freqfile > tempmasses
      awk '/Standard orientation:/,/tional const/ {if (($3=="0") || (substr($3,1,2)==10)) print}' $freqfile > tempstangeos
      awk '/Input orientation:/,/Stoichiometry/ {if (($3=="0") || (substr($3,1,2)==10)) print}' $freqfile > tempinputgeos
      awk -f $proggramdir/proggenHP $freqfile > geoPlusVel
    fi
    if ! (test -s geoPlusVel) then
      echo "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX No geoPlusVel XXXXXXXXXXXXXXXXXXXXXXXXXXXX"
      exit 0
    fi
    if (test -f isomernumber) then
      cp isomernumber temp533
      awk 'BEGIN {getline;i=$1+1;print i,"----trajectory isomer number----"}' temp533 > isomernumber
      rm temp533
    else
      echo "1 ----trajectory isomer number----" > isomernumber
    fi
    echo 1 > runpointnumber
    rm real.mop model.mop g09.com
    awk -f $proggramdir/prog1stpoint isomernumber > real.mop     #prog1stpoint now also makes model.mop and g09.com
#  B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2  if first part successful then clean up and run the first input file, otherwise die
    if ((test -s real.mop) && (test -s model.mop) && (test -s g09.com)) then
      rm tempfreqs tempredmass tempfrc tempmodes tempstangeos tempmasses temp401 temp811 tempinputgeos
      cat isomernumber >> geoRecord
      cat geoPlusVel >> geoRecord
      rm -f $scratchdir/goingwell
      cd $scratchdir
      cp $origdir/real.mop $scratchdir/real.mop
      cp $origdir/model.mop $scratchdir/model.mop
      cp $origdir/g09.com $scratchdir/g09.com
      $g09root/g09/g09 $scratchdir/g09.com > $scratchdir/g09.log
      grep 'Normal termination' $scratchdir/g09.log > $scratchdir/goingwell
      if ! (test -s $scratchdir/goingwell) then
        cp $scratchdir/g09.log $origdir/g09.log
        break
      fi
      $scratchdir/MOPAC2016.exe $scratchdir/model.mop
      grep 'JOB ENDED NORMALLY' $scratchdir/model.out > $scratchdir/goingwell
      if ! (test -s $scratchdir/goingwell) then
        cp $scratchdir/real.out $origdir/real.out
        break
      fi
      $scratchdir/MOPAC2016.exe $scratchdir/real.mop
      cd $origdir
      grep 'JOB ENDED NORMALLY' $scratchdir/real.out > $scratchdir/goingwell
      if (test -s $scratchdir/goingwell) then
        cp $scratchdir/real.out olddynrun2
      else
        cp $scratchdir/real.out $origdir/real.out
        break
      fi
    else
      break
    fi
#  B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3 if B2 worked then you are here.
create 2nd point, run it, and set up for propagation loop
    rm real.mop model.mop g09.com
    echo 2 > runpointnumber
    awk -f $proggramdir/prog2ndpoint $scratchdir/real.out $scratchdir/model.out $scratchdir/g09.log > real.mop #prot2ndpoint
now also makes model.mop and g09.com
# before we decide to run this, check the energy
    awk -f $proggramdir/proganal $scratchdir/real.out >> dynfollowfile
    rm -f $scratchdir/tempdone
    tail -1 dynfollowfile | awk '/XXXX/ {print}' > $scratchdir/tempdone
    if (test  -s $scratchdir/tempdone) then
      rm -f traj
      echo 0 > runpointnumber
      break
    fi
    if ((test -s real.mop) && (test -s model.mop) && (test -s g09.com)) then
      rm -f $scratchdir/goingwell
      cd $scratchdir
      cp $origdir/real.mop $scratchdir/real.mop
      cp $origdir/model.mop $scratchdir/model.mop
      cp $origdir/g09.com $scratchdir/g09.com
      $g09root/g09/g09 $scratchdir/g09.com > $scratchdir/g09.log
      grep 'Normal termination' $scratchdir/g09.log > $scratchdir/goingwell
      if ! (test -s $scratchdir/goingwell) then
        cp $scratchdir/g09.log $origdir/g09.log
        break
      fi
      $scratchdir/MOPAC2016.exe $scratchdir/model.mop
      grep 'JOB ENDED NORMALLY' $scratchdir/model.out > $scratchdir/goingwell
      if ! (test -s $scratchdir/goingwell) then
        cp $scratchdir/real.out $origdir/real.out
        break
      fi
      $scratchdir/MOPAC2016.exe $scratchdir/real.mop
      cd $origdir
      grep 'JOB ENDED NORMALLY' $scratchdir/real.out > $scratchdir/goingwell
      if (test -s $scratchdir/goingwell) then
        cp $scratchdir/real.out olddynrun
        awk -f $proggramdir/proganal $scratchdir/real.out >> dynfollowfile
        awk '/ATOM   CHEMICAL         X/,/CARTESIAN COORDINATES/ {print}' olddynrun | awk '{if (($1>.5) &&
($1<10000)) print}' > old
        awk '/ATOM   CHEMICAL         X/,/CARTESIAN COORDINATES/ {print}' olddynrun2 | awk '{if (($1>.5) &&
($1<10000)) print}' > older
        echo 3 > runpointnumber
        if (test -f bypassproggen) then
          cat bypassproggen > runpointnumber
          echo 3 > bypassproggen
        fi
        awk -f $proggramdir/progdynb $scratchdir/real.out $scratchdir/model.out $scratchdir/g09.log > real.mop
        rm -f old older
      else
        cp $scratchdir/real.out $origdir/real.out
        break
      fi
    else
```

```
         break
      fi
# we've just completed a start, so lets skipstart until instructed otherwise
      echo "forward" > skipstart
   fi
#
B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4
B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4B4
# Reverse trajectories starter routine
   if [ `cat skipstart` = "reverserestart" ]; then
      cd $origdir
      rm real.mop model.mop g09.com
      echo 1 > runpointnumber
      awk -f $proggramdir/prog1stpoint isomernumber > real.mop     #prog1stpoint now also makes model.mop and g09.com
      if ((test -s real.mop) && (test -s model.mop) && (test -s g09.com)) then
         rm -f $scratchdir/goingwell
         cd $scratchdir
         cp $origdir/real.mop $scratchdir/real.mop
         cp $origdir/model.mop $scratchdir/model.mop
         cp $origdir/g09.com $scratchdir/g09.com
         $g09root/g09/g09 $scratchdir/g09.com > $scratchdir/g09.log
         grep 'Normal termination' $scratchdir/g09.log > $scratchdir/goingwell
         if ! (test -s $scratchdir/goingwell) then
            cp $scratchdir/g09.log $origdir/g09.log
            break
         fi
         $scratchdir/MOPAC2016.exe $scratchdir/model.mop
         grep 'JOB ENDED NORMALLY' $scratchdir/model.out > $scratchdir/goingwell
         if ! (test -s $scratchdir/goingwell) then
            cp $scratchdir/real.out $origdir/real.out
            break
         fi
         $scratchdir/MOPAC2016.exe $scratchdir/real.mop
         cd $origdir
         grep 'JOB ENDED NORMALLY' $scratchdir/real.out > $scratchdir/goingwell
         if (test -s $scratchdir/goingwell) then
            cp $scratchdir/real.out olddynrun2
         else
            cp $scratchdir/real.out $origdir/real.out
            break
         fi
      else
         break
      fi
      rm real.mop model.mop g09.com
      echo 2 > runpointnumber
      awk -f $proggramdir/prog2ndpoint $scratchdir/real.out $scratchdir/model.out $scratchdir/g09.log > real.mop #prot2ndpoint
now also makes model.mop and g09.com
      awk -f $proggramdir/proganal $scratchdir/real.out >> dynfollowfile
      rm -f $scratchdir/tempdone
      if ((test -s real.mop) && (test -s model.mop) && (test -s g09.com)) then
         rm -f $scratchdir/goingwell
         cd $scratchdir
         cp $origdir/real.mop $scratchdir/real.mop
         cp $origdir/model.mop $scratchdir/model.mop
         cp $origdir/g09.com $scratchdir/g09.com
         $g09root/g09/g09 $scratchdir/g09.com > $scratchdir/g09.log
         grep 'Normal termination' $scratchdir/g09.log > $scratchdir/goingwell
         if ! (test -s $scratchdir/goingwell) then
            cp $scratchdir/g09.log $origdir/g09.log
            break
         fi
      $scratchdir/MOPAC2016.exe $scratchdir/model.mop
      grep 'JOB ENDED NORMALLY' $scratchdir/model.out > $scratchdir/goingwell
      if ! (test -s $scratchdir/goingwell) then
        cp $scratchdir/real.out $origdir/real.out
        break
      fi
      $scratchdir/MOPAC2016.exe $scratchdir/real.mop
      cd $origdir
      grep 'JOB ENDED NORMALLY' $scratchdir/real.out > $scratchdir/goingwell
      if (test -s $scratchdir/goingwell) then
        cp $scratchdir/real.out olddynrun
        awk -f $proggramdir/proganal $scratchdir/real.out >> dynfollowfile
        awk '/ATOM   CHEMICAL          X/,/CARTESIAN COORDINATES/ {print}' olddynrun | awk '{if (($1>.5) &&
($1<10000)) print}' > old
        awk '/ATOM   CHEMICAL          X/,/CARTESIAN COORDINATES/ {print}' olddynrun2 | awk '{if (($1>.5) &&
($1<10000)) print}' > older
        echo 3 > runpointnumber
        if (test -f bypassproggen) then
          cat bypassproggen > runpointnumber
          echo 3 > bypassproggen
        fi
        awk -f $proggramdir/progdynb $scratchdir/real.out $scratchdir/model.out $scratchdir/g09.log > real.mop
        rm -f old older
      else
        cp $scratchdir/real.out $origdir/real.out
        break
      fi
    else
      break
    fi
# we've just completed a reversestart, so lets skipstart until instructed otherwise
    echo "reverse" > skipstart
  fi

#  END_of_B___END_of_B___END_of_B___END_of_B___END_of_B___END_of_B___END_of_B___END_of_B___

#  CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC  propagation loop
  while (true)
  do
    rm -f $scratchdir/goingwell
    cd $scratchdir
    cp $origdir/real.mop $scratchdir/real.mop
    cp $origdir/model.mop $scratchdir/model.mop
    cp $origdir/g09.com $scratchdir/g09.com
    $g09root/g09/g09 $scratchdir/g09.com > $scratchdir/g09.log
cp $scratchdir/g09.log $origdir/g09.log
    grep 'Normal termination' $scratchdir/g09.log > $scratchdir/goingwell
    if ! (test -s $scratchdir/goingwell) then
      cp $scratchdir/g09.log $origdir/g09.log
      break
    fi
    $scratchdir/MOPAC2016.exe $scratchdir/model.mop
    grep 'JOB ENDED NORMALLY' $scratchdir/model.out > $scratchdir/goingwell
cp $scratchdir/model.out $origdir/model.out
    if ! (test -s $scratchdir/goingwell) then
      cp $scratchdir/model.out $origdir/model.out
      break
    fi
    $scratchdir/MOPAC2016.exe $scratchdir/real.mop
    cd $origdir
    grep 'JOB ENDED NORMALLY' $scratchdir/real.out > $scratchdir/goingwell
cp $scratchdir/real.out $origdir/real.out
```

```
   if (test -s $scratchdir/goingwell) then
     awk -f $proggramdir/proganal $scratchdir/real.out >> $origdir/dynfollowfile
     mv olddynrun2 olddynrun3
     mv olddynrun olddynrun2
     cp $scratchdir/real.out olddynrun
     awk '/ATOM   CHEMICAL          X/,/CARTESIAN COORDINATES/ {print}' olddynrun | awk '{if (($1>.5) &&
($1<10000)) print}' > old
     awk '/ATOM   CHEMICAL          X/,/CARTESIAN COORDINATES/ {print}' olddynrun2 | awk '{if (($1>.5) &&
($1<10000)) print}' > older
     #increment runpointnumber
     cp runpointnumber $scratchdir/temp533
     awk 'BEGIN {getline;i=$1+1;print i}' $scratchdir/temp533 > runpointnumber
     rm $scratchdir/temp533
     awk -f $proggramdir/progdynb $scratchdir/real.out $scratchdir/model.out $scratchdir/g09.log > real.mop
     rm -f old older
   else
     cp $scratchdir/real.out $origdir/real.out
     break
   fi

# here is a cool link that lets you interrupt the dynamics with a short job, then
# it automatically goes back to the dynamics  just make the file 'detour' and it
# will delete detour, run run.com, then go back to dynamics
   if (test  -f detour) then
     rm detour
     date >> $logfile
     cat run.com >> $logfile
     cp run.log temp.log
     export g09root=/software/lms/g09_D01
     . $g09root/g09/bsd/g09.profile
     cd $scratchdir
     $g09root/g09/g09 $origdir/run.com > $origdir/run.log
     cd $origdir
   fi

#stop it all nicely by creating a nogo file
   if (test  -f nogo) then
     break
   fi

#figure out if this isomer is done - change in April 2013 is to move proganal call up from here
   rm -f $scratchdir/tempdone
   tail -2 dynfollowfile | awk '/XXXX/ {print}' > $scratchdir/tempdone
   if (test  -s $scratchdir/tempdone) then
     if [ `awk '/reversetraj/ {if ($1=="reversetraj") print $2}' progdyn.conf` = "true" ]; then
       if [ `cat skipstart` = "reverse" ]; then
         rm -f skipstart
         rm -f geoPlusVel
         rm -f olddynrun
         rm -f olddynrun2
         rm -f olddynrun3
         a=`awk '{print $1}' isomernumber`
         mv traj traj$a
       fi
       if [ `cat skipstart` = "forward" ]; then
         echo reverserestart > skipstart
       fi
     else
       rm -f skipstart
       rm -f geoPlusVel
       rm -f olddynrun
       rm -f olddynrun2
```

```
          rm -f olddynrun3
          a=`awk '{print $1}' isomernumber`
          mv traj traj$a
        fi
        break
      fi
   done
#
END_of_C_Loop____END_of_C_Loop____END_of_C_Loop____END_of_C_Loop____END_of_C_Loop____END_of_C_Loop____

# We've got to break a second time to get out of this loop
# if we really want to quit.  Otherwise, it will start over
# at the top
   if (test  -f nogo) then
      break
   fi
   if (test  -s $scratchdir/goingwell) then
      echo "starting a new point or a new direction"
   else
      break
   fi
done

   if (test  -f nogo) then
      break
   fi
   if (test  -s $scratchdir/goingwell) then
      echo "starting a new point or a new direction2"
   else
      break
   fi
done
exit 0
```

proggenHP

```
BEGIN {
# 2014 - avoids bug with a box on, so that starts without modes use input geometry, not standard orientation
# aug 2013 summary of changes
#includes molecular rotation, ability to do multiple NMR calculations, ONIOM with link atoms,
#nonstandard routes, handling of linear molecules using geometry linear, fixed but with atoms over 99 but
#bug varies with version of Gaussian, randomization based on PROCINFO (solved many problems), added initialDiss 3 for random
#phase of normal modes
# Aug 2010 changes classicalSpacing to 2 and upped possible excited states to 4000
# Jan 2009 - a number of little changes to improve reporting, precision, etc, specification of displacement on particular modes
# Jan 2009 cannonball trajectories.  adds desired energy to initial velocities based on file cannontraj, so one can shoot toward a ts
# updated Nov 2008 to incorporate running DRPs
# updated Nov 2008 to allow for start without an initial freq calc using classical = 2
# updated Aug 2008 added to atom list to handle a large number of atoms without changes needed
# updated June 2008 to incorportates new method for choosing displacements with initialdis 2
# updated Jan 17 2008 - bug fix for > 99 atoms, 300 excitations of low modes possible
# version August 2007 - incorporates classical trajectory calculation option
#also allows listing of number of imaginary frequencies
# version Sept 16, 2005 - incorportates searchdir but not yet rotation
# now reads random numbers from temp811, starting at a random place
# The input files are generated before this and are tempfreqs, tempredmass,
# tempfrc, tempmodes, and tempstangeos.
# It will count the number of atoms.
# default parameters, including quassiclassical, no displacements, transition state, not a DRP
# do not change these - rather, change progdyn.conf to set the parameters
initialDis=0; timestep=1E-15; scaling=1.0; temp=298.15
classical=0; numimag=1; DRP=0; cannonball=0
charge=0; multiplicity=1; method="HF/3-21G"; memory=20000000
diag=1; checkpoint="g09.chk"; searchdir="positive"; boxon=0
boxsize=10; maxAtomMove=0.1; title1="you"; title2="need"
title3="a"; title4="progdyn.conf"; processors=1; highlevel=999
conver1=4.184E26 #dividing by this converts amu angs^2 /s^2 to kcal/mol
geometry="nonlinear";rotationmode=0

#initialization and constants
for (i=1;i<=10000;i++) {disMode[i]=-1}
i=1;j=1;k=1
c=29979245800; h=6.626075E-34; avNum=6.0221415E23
RgasK=0.00198588; RgasJ=8.31447
numAtoms=0; atomnumber=0; classicalSpacing=2
zpeGauss=0; zpeGaussK=0; zpePlusE=0; potentialE=0

# read progdyn.conf for configuration info
blankLineTester=10
while (blankLineTester>1) {
  getline < "progdyn.conf"
  if ($1=="method") method=$2
  if ($1=="charge") charge=$2
  if ($1=="multiplicity") multiplicity=$2
  if ($1=="memory") memory=$2
  if ($1=="processors") processors=$2
  if ($1=="checkpoint") checkpoint=$2
  if ($1=="diagnostics") diag=$2
  if ($1=="initialdis") initialDis=$2
  if ($1=="timestep") timestep=$2
  if ($1=="scaling") scaling=$2
  if ($1=="temperature") temp=$2
  if ($1=="searchdir") searchdir=$2
  if ($1=="classical") classical=$2
  if ($1=="numimag") numimag=$2
  if ($1=="geometry") geometry=$2
  if ($1=="highlevel") highlevel=$2
  if ($1=="boxon") boxon=$2
  if ($1=="boxsize") boxsize=$2
  if ($1=="DRP") DRP=$2; if (DRP==1) classical=2 #this lets one start a DRP from a point that is not a freq calc
  if ($1=="maxAtomMove") maxAtomMove=$2
  if ($1=="cannonball") cannonball=$2
  if ($1=="displacements") disMode[$2]=$3
  if ($1=="controlphase") controlPhase[$2]=$3
  if ($1=="rotationmode") rotationmode=$2
  if ($1=="title") {
    title1=$2
    title2=$3
    title3=$4
    title4=$5
    }
  blankLineTester=length($0)
  }

if (diag>=1) print "***************** starting proggen *****************" >> "diagnostics"
if (diag>=1) print "method,charge,multiplicity,memory" >> "diagnostics"
if (diag>=1) print method,charge,multiplicity,memory >> "diagnostics"
if (diag>=1) print "processors,checkpoint,title,initialdis,timestep,scaling,temperature" >> "diagnostics"
if (diag>=1) print processors,checkpoint,title1,title2,title3,title4,initialDis,timestep,scaling,temp >> "diagnostics"
if (diag>=1) print "classical,numimag,highlevel,boxon,boxsize,DRP,maxAtomMove,cannonball" >> "diagnostics"
if (diag>=1) print classical,numimag,highlevel,boxon,boxsize,DRP,maxAtomMove,cannonball >> "diagnostics"

# put geometries into array, also figure out number of atoms
# note that this picks out the last geometry in a file, assuming
# that if there is an optimization followed by a freq, nothing else follows
# kludgy - repeats last line twice - must be a better way
do {
  getline < "tempstangeos"
  if (oldline==$0) $0=""
  oldline=$0
  atom = $1
  if (atom>numAtoms) numAtoms=atom
  atNum[atom]=$2
  geoArr[atom,1]=$4; geoArr[atom,2]=$5; geoArr[atom,3]=$6
  geoArrOrig[atom,1]=$4; geoArrOrig[atom,2]=$5; geoArrOrig[atom,3]=$6
  velArr[atom,1]=0; velArr[atom,2]=0; velArr[atom,3]=0
  }
while (length($0) > 0)

#output the number of atoms, used in many routines
print numAtoms

# put in atomic symbols and atomic weights - assigns a default mass but then reads it from tempmasses when possible
for (i=1;i<=numAtoms;i++) {
  getline < "tempmasses"
  if (atNum[i]==1) {atSym[i]="H";atWeight[i]=1.00783}
  if (atNum[i]==2) {atSym[i]="He";atWeight[i]=4.0026}
  if (atNum[i]==3) {atSym[i]="Li";atWeight[i]=6.941}
  if (atNum[i]==4) {atSym[i]="Be";atWeight[i]=9.012}
  if (atNum[i]==5) {atSym[i]="B";atWeight[i]=10.811}
  if (atNum[i]==6) {atSym[i]="C";atWeight[i]=12.}
  if (atNum[i]==7) {atSym[i]="N";atWeight[i]=14.007}
  if (atNum[i]==8) {atSym[i]="O";atWeight[i]=15.9994}
  if (atNum[i]==9) {atSym[i]="F";atWeight[i]=18.9984}
  if (atNum[i]==10) {atSym[i]="Ne";atWeight[i]=20.1797}
  if (atNum[i]==11) {atSym[i]="Na";atWeight[i]=22.989}
  if (atNum[i]==12) {atSym[i]="Mg";atWeight[i]=24.305}
  if (atNum[i]==13) {atSym[i]="Al";atWeight[i]=26.98154}
  if (atNum[i]==14) {atSym[i]="Si";atWeight[i]=28.0855}
  if (atNum[i]==15) {atSym[i]="P";atWeight[i]=30.9738}
  if (atNum[i]==16) {atSym[i]="S";atWeight[i]=32.066}
  if (atNum[i]==17) {atSym[i]="Cl";atWeight[i]=35.4527}
  if (atNum[i]==18) {atSym[i]="Ar";atWeight[i]=39.948}
  if (atNum[i]==19) {atSym[i]="K";atWeight[i]=39.0983}
  if (atNum[i]==20) {atSym[i]="Ca";atWeight[i]=40.078}
  if (atNum[i]==21) {atSym[i]="Sc";atWeight[i]=44.96}
  if (atNum[i]==22) {atSym[i]="Ti";atWeight[i]=47.867}
  if (atNum[i]==23) {atSym[i]="V";atWeight[i]=50.94}
  if (atNum[i]==24) {atSym[i]="Cr";atWeight[i]=51.9961}
  if (atNum[i]==25) {atSym[i]="Mn";atWeight[i]=54.938}
  if (atNum[i]==26) {atSym[i]="Fe";atWeight[i]=55.845}
  if (atNum[i]==27) {atSym[i]="Co";atWeight[i]=58.933}
  if (atNum[i]==28) {atSym[i]="Ni";atWeight[i]=58.693}
  if (atNum[i]==29) {atSym[i]="Cu";atWeight[i]=63.546}
  if (atNum[i]==30) {atSym[i]="Zn";atWeight[i]=65.38}
  if (atNum[i]==31) {atSym[i]="Ga";atWeight[i]=69.723}
  if (atNum[i]==32) {atSym[i]="Ge";atWeight[i]=72.64}
  if (atNum[i]==33) {atSym[i]="As";atWeight[i]=74.9216}
  if (atNum[i]==34) {atSym[i]="Se";atWeight[i]=78.96}
  if (atNum[i]==35) {atSym[i]="Br";atWeight[i]=79.904}
  if (atNum[i]==46) {atSym[i]="Pd";atWeight[i]=106.42}
  if (atNum[i]==53) {atSym[i]="I";atWeight[i]=126.90447}
```

```
# gets actual weight from freqinHP when possible so a prior calc with readisotopes gets you isotopic substitution
  if ((i<100) && ($9>0)) atWeight[i]=$9
#  if ((i>99) && ($8>0)) atWeight[i]=$8

  if ((diag>1) && (i==1)) print "atNum[i],atSym[i],atWeight[i],geoArr[i,1],geoArr[i,2],geoArr[i,3]" >> "diagnostics"
  if (diag>1) print atNum[i],atSym[i],atWeight[i],geoArr[i,1],geoArr[i,2],geoArr[i,3] >> "diagnostics"
  }

# read in frequencies, scale them, read in Reduced masses, read in force
#constants, replace negative frequencies by 2 wavenumbers
numFreq=3*numAtoms-6
if (geometry=="linear") numFreq=3*numAtoms-5
for (i=1;i<=numFreq;i++) {
  $0=""
  getline < "tempfreqs"
  freq[i]=$0*scaling
  if (freq[i]<0) freq[i]=2
  }
for (i=1;i<=numFreq;i++) {
  $0=""
  getline < "tempredmass"
  redMass[i]=$0
  if (redMass[i]=="") redMass[i]=1.
  }
for (i=1;i<=numFreq;i++) {
  $0=""
  getline < "tempfrc"
  frc[i]=$0
  if (frc[i]=="") frc[i]=0.0001
  if (frc[i]==0) frc[i]=0.0001
  if ((diag>1) && (i==1)) print "freq[i],redMass[i],frc[i]" >> "diagnostics"
  if (diag>1) print freq[i],redMass[i],frc[i] >> "diagnostics"
  }

# read in the modes - note that trajectories always need a freq calc with freq=hpmodes unless classical=2
if (classical!=2) {
  for (i=1;i<=numFreq;i+=5) {
    for (j=1;j<=(3*numAtoms);j++) {
      getline < "tempmodes"
      mode[i,$2,$1]=$4; mode[i+1,$2,$1]=$5; mode[i+2,$2,$1]=$6; mode[i+3,$2,$1]=$7; mode[i+4,$2,$1]=$8
      }
    }
  }
if (diag>2) {for (i=1;i<=numFreq;i++) {print mode[i,1,1],mode[i,1,2],mode[i,1,3] >> "modesread"}}

# if doing a cannonball trajectory, read in the vector
if (cannonball>0) {
  for (i=1;i<=numAtoms;i++) {
    getline < "cannontraj"
    cannonArr[i,1]=$1; cannonArr[i,2]=$2; cannonArr[i,3]=$3
    }
  }

# collect a series of random numbers from file temp811, generated from an outside random number generator called by
prodynstarterHP
# read from temp811, starting at a random place
srand(PROCINFO["pid"]); tester=rand()*1000
for (i=1;i<=tester;i++) getline < "temp811"
for (i=1;i<=numFreq;i++) {
  getline < "temp811"; randArr[i]=$1
  getline < "temp811"; randArrB[i]=$1
  getline < "temp811"; randArrC[i]=$1
```

```
}
if (rotationmode>0) {
  for (i=1;i<=6;i++) {
    getline < "temp811"; randArrR[i]=$1
  }
}

# for a QM distribution for a harmonic oscillator in its ground state, we want to generate a set of random numbers
#between -1 and 1 weighted such that numbers toward the center are properly more common
i=1
while (i<=numFreq) {
  if ((initialDis==2) || (disMode[i]==2)) {
    getline < "temp811"
    tempNum=2*($1-.5)
    prob=exp(-(tempNum^2))
    getline < "temp811"
    if ($1<prob) {
      randArrD[i]=tempNum
      i++
      }
    }
  if ((initialDis!=2) && (disMode[i]!=2)) i++
  }

# to start without normal modes or frequencies we need to just pick a random direction for the motion of each atom, requiring 3N random numbers
for (i=1;i<=numAtoms;i++) {
  for (j=1;j<=3;j++) {
  getline < "temp811"
  if ($1>0.5) randArrE[i,j]=1
  if ($1<.5) randArrE[i,j]=-1
    }
  }

# determine energy in each normal mode
for (i=1;i<=numFreq;i++) {
  zpeJ[i]=0.5*h*c*freq[i]      #units J per molecule
#if classical, treat as modes spaced by classicalSpacing wavenumbers
  if (classical==1) zpeJ[i]=0.5*h*c*classicalSpacing  # the zpe is not used when classical but the spacing is used to calculate the E in mode
  zpeK[i]=zpeJ[i]*avNum/4184    #units kcal/mol
  if (temp<10) vibN[i]=0        # avoids working with very small temperatures - if the temp is too low, it just acts like 0 K
  if (temp>=10) {
    zpeRat[i]=exp((-2*zpeK[i])/(RgasK*temp))
    if (zpeRat[i]==1) zpeRat[i]=.99999999999
    Q[i]=1/(1-zpeRat[i])
    newRand=randArr[i]
    vibN[i]=0
    tester=1/Q[i]
#     get up to 4000 excitations of low modes
    for (j=1;j<=(4000*zpeRat[i]+2);j++) {
      if (newRand>tester) vibN[i]++
      tester=tester+((zpeRat[i]^j)/Q[i])
      }
    }
  }

# figure out mode energies and maximum classical shift and then actual shift
# also calculated total energy desired for molecule
desiredModeEnK=0
for (i=1;i<=numFreq;i++) {
  modeEn[i]=(zpeJ[i]*1E18)*(2*vibN[i]+1) # units here are mDyne Ansgroms for compatibility with Gaussian force constants
  if (classical==1) modeEn[i]=(zpeJ[i]*1E18)*2*vibN[i]    #no zpe when classical
  modeEnK[i]=zpeK[i]*(2*vibN[i]+1)
  if (classical==1) modeEnK[i]=zpeK[i]*2*vibN[i]          #no zpe when classical
  desiredModeEnK=desiredModeEnK + modeEnK[i]
# no 1/2 hv for imaginary frequencies
# treating modes with frequencies <10 as translations, ignoring their zero point energies
  if (freq[i]<10) modeEn[i]=(zpeJ[i]*1E18)*(2*vibN[i])
  maxShift[i]=(2*modeEn[i]/frc[i])^0.5
# new 2012 initialDis 3 means random phase of normal mode
  if (initialDis==3) shift[i]=maxShift[i]*sin(randArrC[i]*3.141592*2)
  if (initialDis==2) shift[i]=maxShift[i]*randArrD[i]
  if (initialDis==1) shift[i]=maxShift[i]*(2*(randArrC[i]-0.5))
  if (initialDis==0) shift[i]=0
# lines below allow for setting of displacement mode for individual modes
# It used to be necessary to use disMode 10 to turn off displacements for a mode, but hopefully that bug is killed and you can use
disMode 0
  if (disMode[i]==3) shift[i]=maxShift[i]*sin(randArrC[i]*3.141592*2)
  if (disMode[i]==2) shift[i]=maxShift[i]*randArrD[i]
  if (disMode[i]==1) shift[i]=maxShift[i]*(2*(randArrC[i]-0.5))
  if (disMode[i]==10) shift[i]=0 #kept for backward compatibility
  if (disMode[i]==0) shift[i]=0
# no displacements along imaginary frequencies and very low ones - it is better to treat these
# as translations - employing a shift can give you initial weird geometries
  if (freq[i]<10) shift[i]=0
  if (numimag==1) shift[1]=0
  if (numimag==2) shift[2]=0
  }
for (i=1;i<=numFreq;i++) {
  if ((diag>1) && (i==1)) print "zpeJ[i],zpeK[i],zpeRat[i],Q[i],vibN[i],modeEn[i],maxShift[i],shift[i]" >> "diagnostics"
  if (diag>1) print zpeJ[i],zpeK[i],zpeRat[i],Q[i],vibN[i],modeEn[i],maxShift[i],shift[i] >> "diagnostics"
  }

# multiply each of the modes by its shift and add them up
# Do not do this if classical=2
if (classical!=2) {
  for (i=1;i<=numFreq;i++) {
    for (j=1;j<=numAtoms;j++) {
      for (k=1;k<=3;k++) {
        shiftMode[i,j,k]=mode[i,j,k]*shift[i]
        geoArr[j,k]=geoArr[j,k]+shiftMode[i,j,k]
        }
      }
    }
  }

#now start toward velocities
for (i=1;i<=numFreq;i++) {
  kinEn[i]=100000*(modeEn[i]-0.5*frc[i]*shift[i]^2)  # the 100000 converts to g angstrom^2 s^2
  vel[i]=(2*kinEn[i]/(redMass[i]/avNum))^0.5        # in angstrom / s
#use searchdir in progdyn.conf to control the direction for trajectories started from a saddle point
  if (numimag>1) numimag=1  #only the first freq can be sent in the searchdir direction, the rest go in a random direction
  if (i>numimag) {
    if (randArrB[i]<0.5) vel[i]=-vel[i]
    }
  if (i==numimag) {
    if (searchdir=="negative") vel[i]=-vel[i]
    }
  if ((diag>1) && (i==1)) print "vel[i]" >> "diagnostics"
  if (diag>1) print vel[i] >> "diagnostics"
  }

# if controlphase is being used, set the velocity on particular modes as positive or negative as requested
```

```
for (i=1;i<=numFreq;i++) {
  if ((controlPhase[i]=="positive") && (vel[i]<0)) vel[i]=-vel[i]
  if ((controlPhase[i]=="negative") && (vel[i]>0)) vel[i]=-vel[i]
  }

# multiply each of the modes by its velocity and add them up
# Do not do this if classical=2
if (classical!=2) {
  for (i=1;i<=numFreq;i++) {
    for (j=1;j<=numAtoms;j++) {
      for (k=1;k<=3;k++) {
        velMode[i,j,k]=mode[i,j,k]*vel[i]*timestep
        velArr[j,k]=velArr[j,k]+velMode[i,j,k]
        }
      }
    }
  }

# to start without normal modes or frequencies we figure out the energy per atom based on 1/2RT in degree of freedom
if (classical==2) {
# to avoid a bug with a box on, starts without modes should use the input geometry, not the standard
  do {
    getline < "tempinputgeos"
    if (oldline==$0) $0=""
    oldline=$0
    atom = $1
    geoArr[atom,1]=$4; geoArr[atom,2]=$5; geoArr[atom,3]=$6
    geoArrOrig[atom,1]=$4; geoArrOrig[atom,2]=$5; geoArrOrig[atom,3]=$6
    }
  while (length($0) > 0)
  degFreedomEnK=temp*RgasK
  degFreedomEnJ=degFreedomEnK/(avNum/4184)
  cartEn=degFreedomEnJ*1E18
  kinEnCart=100000*cartEn
#print degFreedomEnK, degFreedomEnJ, cartEn, kinEnCart
  for (i=1;i<=numAtoms;i++) {
    for (j=1;j<=3;j++) {
      velArr[i,j]=randArrE[i,j]*timestep*(2*kinEnCart/(atWeight[i]/avNum))^0.5
      if (DRP==1) velArr[i,j]=0
      }
    }
  }

# calculate the KE in the modes at this point
KEinitmodes=0
for (j=1;j<=numAtoms;j++) {
  KEinitmodes=KEinitmodes + 0.5*atWeight[j]*(velArr[j,1]^2 + velArr[j,2]^2 + velArr[j,3]^2)/((timestep^2)*conver1)
  }

# add molecular rotation if requested
if (rotationmode>0) {
#establish three rotation vectors
  for (j=1;j<=numAtoms;j++) {
    rotateX[j,1]=0
    rotateX[j,2]=-geoArrOrig[j,3]
    rotateX[j,3]=geoArrOrig[j,2]
    rotateY[j,1]=-geoArrOrig[j,3]
    rotateY[j,2]=0
    rotateY[j,3]=geoArrOrig[j,1]
    rotateZ[j,1]=-geoArrOrig[j,2]
    rotateZ[j,2]=geoArrOrig[j,1]
    rotateZ[j,3]=0
    }
#figure out how much energy is in the raw vectors
  eRotX=0;eRotY=0;eRotZ=0
  for (j=1;j<=numAtoms;j++) {
    for (k=1;k<=3;k++) {
      eRotX=eRotX + 0.5*atWeight[j]*(rotateX[j,k]^2)/((timestep^2)*conver1)
      eRotY=eRotY + 0.5*atWeight[j]*(rotateY[j,k]^2)/((timestep^2)*conver1)
      eRotZ=eRotZ + 0.5*atWeight[j]*(rotateZ[j,k]^2)/((timestep^2)*conver1)
      }
    }
#  print "rotation energies if raw vector used",eRotX,eRotY,eRotZ
#now decide how much energy we want in each rotation
  keRx=-0.5*0.001987*temp*log(1-randArrR[1])
  keRy=-0.5*0.001987*temp*log(1-randArrR[2])
  keRz=-0.5*0.001987*temp*log(1-randArrR[3])
  if (eRotX<1) keRx=0;if (eRotY<1) keRy=0;if (eRotZ<1) keRz=0
  rotEdesired=keRx+keRy+keRz
  signX=1;signY=1;signZ=1
  if (randArrR[4]<.5) signX=-1
  if (randArrR[5]<.5) signY=-1
  if (randArrR[6]<.5) signZ=-1

#  print "desired energies",keRx,keRy,keRz,"and random numbers",randArrR[1],randArrR[2],randArrR[3]
#protect against zero rotations
  if (eRotX<1) eRotX=1;if (eRotY<1) eRotY=1;if (eRotZ<1) eRotZ=1
#now scale the rotational vectors
  scaleX=(keRx/eRotX)^.5
  scaleY=(keRy/eRotY)^.5
  scaleZ=(keRz/eRotZ)^.5
#  print "scaling factors" scaleX,scaleY,scaleZ
  for (j=1;j<=numAtoms;j++) {
    for (k=1;k<=3;k++) {
      rotateX[j,k]=rotateX[j,k]*scaleX*signX
      rotateY[j,k]=rotateY[j,k]*scaleY*signY
      rotateZ[j,k]=rotateZ[j,k]*scaleZ*signZ
      }
    }
  for (j=1;j<=numAtoms;j++) {
#     print rotateX[j,1],"   ",rotateX[j,2],"   ",rotateX[j,3]
    }
#  print ""
  for (j=1;j<=numAtoms;j++) {
#     print rotateY[j,1],"   ",rotateY[j,2],"   ",rotateY[j,3]
    }
#  print ""
  for (j=1;j<=numAtoms;j++) {
#     print rotateZ[j,1],"   ",rotateZ[j,2],"   ",rotateZ[j,3]
    }
# now add the rotational vectors  to velArr
  for (j=1;j<=numAtoms;j++) {
    for (k=1;k<=3;k++) {
      velArr[j,k]=velArr[j,k]+rotateX[j,k]+rotateY[j,k]+rotateZ[j,k]
      }
    }
  }

# if doing a cannonball, adjust multiplier until extra energy is correct
if (cannonball>0) {
  multiplier=1; tester=0; tolerance=.1
  while (tester==0) {
    KEinittotal=0
    for (j=1;j<=numAtoms;j++) {
        cannonvelArr[j,1]=velArr[j,1]+multiplier*cannonArr[j,1]; cannonvelArr[j,2]=velArr[j,2]+multiplier*cannonArr[j,2];
cannonvelArr[j,3]=velArr[j,3]+multiplier*cannonArr[j,3]
        KEinittotal=KEinittotal + 0.5*atWeight[j]*(cannonvelArr[j,1]^2 + cannonvelArr[j,2]^2 +
cannonvelArr[j,3]^2)/((timestep^2)*conver1)
        }
    if (KEinittotal>(KEinitmodes+cannonball+tolerance)) multiplier=multiplier*0.98901364
    if (KEinittotal<(KEinitmodes+cannonball-tolerance)) multiplier=multiplier*1.01
    if ((KEinittotal<(KEinitmodes+cannonball+tolerance)) && (KEinittotal>(KEinitmodes+cannonball-tolerance))) tester=1
    }
  for (j=1;j<=numAtoms;j++) {
    velArr[j,1]=velArr[j,1]+multiplier*cannonArr[j,1]; velArr[j,2]=velArr[j,2]+multiplier*cannonArr[j,2];
velArr[j,3]=velArr[j,3]+multiplier*cannonArr[j,3]
    }
  }

#output the new geometry.
# ****** this section changed for special experiment for cyclopentadiene.  do not use this for other cases
# atWeight[4]=140.0001
# ****** line below added for special experiment switching mass from 12 to 140, keeping momenta the same
#velArr[4,1]=velArr[4,1]/11.66667; velArr[4,2]=velArr[4,2]/11.66667; velArr[4,3]=velArr[4,3]/11.66667
for (j=1;j<=numAtoms;j++) {
  printf("%2s % .7f % .7f % .7f %9.5f \n",atSym[j],geoArr[j,1],geoArr[j,2],geoArr[j,3],atWeight[j])
  }

#output the velocities and calculate the total kinetic energy overall
KEinittotal=0
for (j=1;j<=numAtoms;j++) {
  KEinittotal=KEinittotal + 0.5*atWeight[j]*(velArr[j,1]^2 + velArr[j,2]^2 + velArr[j,3]^2)/((timestep^2)*conver1)
  printf("% .8f % .8f % .8f \n",velArr[j,1],velArr[j,2],velArr[j,3])
  }

#anything else I add to the file will not affect the trajectories but will keep a record and be good for analysis
if (classical!=2) {
  for (i=1;i<=numFreq;i++) {
    if (initialDis==0) printf("%.6f   % .6f    %4i    % 1.4e       % .6f %1i\n", randArr[i], randArrB[i], vibN[i], vel[i], shift[i],
disMode[i])
    if (initialDis==1) printf("%.6f   % .6f    %4i    % 1.4e       % .6f %1i\n", randArr[i], randArrC[i], vibN[i], vel[i], shift[i],
disMode[i])
    if (initialDis==2) printf("%.6f   % .6f    %4i    % 1.4e       % .6f %1i\n", randArr[i], randArrD[i], vibN[i], vel[i], shift[i],
disMode[i])
    if (initialDis==3) printf("%.6f   % .6f    %4i    % 1.4e       % .6f %1i % .6f\n", randArr[i], randArrC[i], vibN[i], vel[i], shift[i],
disMode[i], sin(randArrC[i]*3.141592*2))
    }
  }
print "temp ",temp
print "initialDis",initialDis
print "classical",classical
print "timestep",timestep
print "numimag",numimag
OFMT = "%.3f"
print "Total mode energy desired=",desiredModeEnK
print "KE initial from modes=",KEinitmodes,"   KE initial total=",KEinittotal,"   Rotational Energy desired=",rotEdesired
if (cannonball>0) print "cannonball",cannonball,"  cannon Energy=",KEinittotal-KEinitmodes
if (boxon>0) print "boxsize",boxsize
if (DRP>0) print "DRP",DRP,"   maxAtomMove",maxAtomMove
if (DRP>0) print maxAtomMove > "maxMove"
}  # End of BEGIN

/Zero-point correction/ {zpeGauss=$3}
/zero-point Energies/ {zpePlusE=$7}
END {
zpeGaussK=zpeGauss*627.509
```

```
potentialE=zpePlusE - zpeGauss
OFMT = "%.6f"
print "Gaussian zpe=",zpeGauss,"or",zpeGaussK,"kcal/mol  E + zpe=",zpePlusE,"  potential E=",potentialE
print "" #will use blank line to mark end of geoPlusVel file
}
```

prog1stpoint

```
BEGIN {
# Nov 2016 Created
# this program creates the first input files for our homegrown ONIOM's first point

initializeparameters()
initializeconstants()
readprogdynconf()

getline < "isomernumber"
isomernum = $1
getline < "runpointnumber"
runpointnum = $1

if (diag>=1) diagnosticsA()
readgeoPlusVel() #this is different from prog2ndpoint and progdynb because we are just getting the geometry, no velocities or
old geos

writereal()
writemodel()
writeg09()
}

END {
}

############################################################ FUNCTIONS
##################################################
function Distance(Atom1,Atom2) {
  return sqrt((oldarr[Atom1,1]-oldarr[Atom2,1])^2+(oldarr[Atom1,2]-oldarr[Atom2,2])^2+(oldarr[Atom1,3]-
oldarr[Atom2,3])^2)
  }

# default parameters, including quassiclassical, no displacements, transition state, not a DRP
# do not change these - rather, change progdyn.conf to set the parameters
function initializeparameters() {
  initialDis=0; timestep=1E-15; scaling=1.0; temp=298.15
  classical=0; numimag=1; DRP=0; cannonball=0
  memory=20000000
  diag=1; checkpoint="g09.chk"; searchdir="positive"; boxon=0
  boxsize=10; maxAtomMove=0.1; title1="you"; title2="need"
  title3="a"; title4="progdyn.conf"; processors=1; highlevel=99999; linkatoms=0
  damping=1;nonstandard=0;geometry="nonlinear";nonstandard=0
  nmrtype=0;nmrevery=9999999;nmrcc=0;nmrrand=0;nmrdo=0
  thermostat=0;thermostatemult=1.00
  oniomcharge=0; oniommult=1
  applyforce=0; applyforceB=0; applyforceC=0; zeroatomon=0
  sphereon=0; spheresize=999; sphereforceK=0.01
  empiricaldispersion=0; radiusmultiplier=1.25
  lowmethod="PM3";method="HF/3-21G"
  }

function initializeconstants() {
  srand(PROCINFO["pid"])
  i=1;j=1;k=1
  c=29979245800; h=6.626075E-34; avNum=6.0221415E23
  RgasK=0.00198588; RgasJ=8.31447; pi=3.14159265359
  numAtoms=0; atomnumber=0
  conver1=4.184E26 #dividing by this converts amu angs^2 /s^2 to kcal/mol
  OFS="   "
  }

# read progdyn.conf for configuration info
function readprogdynconf() {
  blankLineTester=10
  while (blankLineTester>1) {
    getline < "progdyn.conf"
    if ($1=="method") method=$2
    if ($1=="lowmethod") lowmethod=$2
    if ($1=="method2") meth2=$2
    if ($1=="charge") charge=$2
    if ($1=="multiplicity") multiplicity=$2
    if ($1=="oniomchargemult") { #this will be the charge and multiplicity for the small model system - or else it will be 0 and 1
      oniomcharge=$2
      oniommult=$3
      }
    if ($1=="memory") memory=$2
    if ($1=="processors") processors=$2
    if ($1=="checkpoint") checkpoint=$2
    if ($1=="timestep") timestep=$2
    if ($1=="diagnostics") diag=$2
    if ($1=="temperature") temp=$2
    if ($1=="thermostat") thermostat=$2
    if ($1=="thermostatmult") thermostatmult=$2
    if (thermostatmult>1) thermostatmult=1/thermostatmult
    if ($1=="method3") meth3=$2
    if ($1=="method4") meth4=$2
    if ($1=="method5") meth5=$2
    if ($1=="method6") meth6=$2
    if ($1=="method7") meth7=$2
    if ($1=="highlevel") highlevel=$2
    if ($1=="linkatoms") linkatoms=$2
    if ($1=="fixedatom1") fixedatom1=$2
    if ($1=="fixedatom2") fixedatom2=$2
    if ($1=="fixedatom3") fixedatom3=$2
    if ($1=="fixedatom4") fixedatom4=$2
    if ($1=="boxon") boxon=$2
    if ($1=="boxsize") boxsize=$2
    if ($1=="sphereon") sphereon=$2
    if ($1=="spheresize") spheresize=$2
    if ($1=="sphereforce") sphereforceK=$2
    if ($1=="DRP") DRP=$2
    if ($1=="maxAtomMove") maxAtomMove=$2
    if ($1=="methodfile") methodfilelines=$2
    if ($1=="killcheck") killcheck=$2
    if ($1=="empiricaldispersion") empiricaldispersion=$2
    if ($1=="radiusmultiplier") radiusmultiplier=$2
    if ($1=="damping") damping=$2
    if ($1=="NMRmethod") nmrmethod=$2
    if ($1=="NMRmethod2") nmrmethod2=$2
    if ($1=="NMRmethod3") nmrmethod3=$2
    if ($1=="NMRtype") nmrtype=$2
    if ($1=="NMRevery") nmrevery=$2
    if ($1=="NMRrand") nmrrand=$2
    if ($1=="loadlimit") loadlimit=$2
    if ($1=="NMRcc") nmrcc=$2
    if ($1=="nonstandard") nonstandard=$2
    if ($1=="applyforce") {
      applyforce=$2; apforce=$3; apforceX0=$4; apforce2=$5; apforce3=$6
      }
    if ($1=="applyforceB") {
      applyforceB=$2; apforceB=$3; apforceX0B=$4; apforce2B=$5; apforce3B=$6
      }
    if ($1=="applyforceC") {
      applyforceC=$2; apforceC=$3; apforceX0C=$4; apforce2C=$5; apforce3C=$6
      }
    if ($1=="afatoms") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afatom[i]=$(i+1)
        }
      }
    if ($1=="afatomsB") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afatomB[i]=$(i+1)
        }
      }
    if ($1=="afatomsC") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afatomC[i]=$(i+1)
        }
      }
    if ($1=="applyforceplane") {
      applyforceplane=$2; apforceplane=$3; apforceplaneX0=$4
      }
    if ($1=="afplaneatoms") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afplaneatoms[i]=$(i+1)
        }
      }
    if ($1=="zeroatom") {
      zeroatomon=1
      zeroatom=$2
      }
    if ($1=="title") {
      title1=$2
      title2=$3
      title3=$4
      title4=$5
      }
    blankLineTester=length($0)
    }
  }

function diagnosticsA() {
  print "****************** starting progdynb ******************" >> "diagnostics"
  print "method,charge,multiplicity,memory" >> "diagnostics"
  print method,charge,multiplicity,memory >> "diagnostics"
  print "processors,checkpoint,title" >> "diagnostics"
  print processors,checkpoint,title1,title2,title3,title4 >> "diagnostics"
  }

function readgeoPlusVel() {
#this is different from prog2ndpoint and progdynb because we are just getting the geometry, no velocities or old geos
#read in number of atoms, geometry, masses from geoPlusVel
  getline < "geoPlusVel"
  numAtoms=$1
# geometry
  for (i=1;i<=numAtoms;i++) {
    getline < "geoPlusVel"
    weight[i]=$5
    atSym[i]=$1
    for (j=1;j<=3;j++) {
      geoArr[i,j]=$(1+j)
      }
    }
  blankLineTester=10
  while (blankLineTester>1) {
    getline < "geoPlusVel"
    if ($11=="potential") potentialE=$13
    blankLineTester=length($0)
    }
  }

function writereal() {
  print "XYZ GRADIENTS 1SCF PRTXYZ",lowmethod,"THREADS=" processors,"CHARGE=" charge
  print title1,title2,title3,title4,"runpoint ",runpointnum,"runisomer ",isomernum
  print ""
  for (i=1;i<=numAtoms;i++) {
    printf("%s %.7f %s %.7f %s %.7f %s ",atSym[i],geoArr[i,1],"1",geoArr[i,2],"1",geoArr[i,3],"1")
    print ""
    }
  print ""
  }

function writemodel() {
  print "XYZ GRADIENTS 1SCF PRTXYZ",method,"THREADS=" processors,"CHARGE=" oniomcharge > "model.mop"
  print title1,title2,title3,title4,"runpoint ",runpointnum,"runisomer ",isomernum >> "model.mop"
  print "" >> "model.mop"
  if (highlevel>numAtoms) highlevel=numAtoms
  for (i=1;i<=highlevel;i++) {
    printf("%s %.7f %s %.7f %s %.7f %s ",atSym[i],geoArr[i,1],"1",geoArr[i,2],"1",geoArr[i,3],"1") >> "model.mop"
    print "" >> "model.mop"
    }
  print "" >> "model.mop"
  }

function writeg09() {
  print "%nproc=" processors > "g09.com"
  print "%mem=" memory >> "g09.com"
  if (killcheck!=1) print "%chk=" checkpoint >> "g09.com"
  if (nonstandard==0) {
    print "#p " method " force scf=(xqc,maxconven=155,fulllinear,nosym) " >> "g09.com"
    if (meth2=="unrestricted") print "guess=mix" >> "g09.com" #for unrestricted calculations
    if (length(meth3)>2) print meth3 >> "g09.com"
    if (length(meth4)>2) print meth4 >> "g09.com"
    }
  if (nonstandard==1) {
    print "# " >> "g09.com"
    print "nonstd"  >> "g09.com"
    system("cat nonstandard >> g09.com")
    }
  print "" >> "g09.com"
  print  title1,title2,title3,title4 >> "g09.com"
  print "runpoint ",runpointnum >> "g09.com"
  print "runisomer ",isomernum >> "g09.com"
  if (DRP==1) {
    print "maxForce and forceMult and maxAtomMove",maxForce,forceMult,maxAtomMove >> "g09.com"
    print "maxForce and forceMult and maxAtomMove",maxForce,forceMult,maxAtomMove >> "movelist"
    }
  print "" >> "g09.com"
  print oniomcharge,oniommult >> "g09.com"
  if (highlevel>numAtoms) highlevel=numAtoms
  for (i=1;i<=highlevel;i++) {
    printf("%s %.7f %.7f %.7f",atSym[i],geoArr[i,1],geoArr[i,2],geoArr[i,3]) >> "g09.com"
    print "" >> "g09.com"
    }
  print "" >> "g09.com"
  if (length(meth5)>2) print meth5 >> "g09.com"
  if (length(meth6)>2) print meth6 >> "g09.com"
  if (methodfilelines>=1) {
    for (i=1;i<=methodfilelines;i++) {
      getline < "methodfile"
      print $0 >> "g09.com"
      }
    }
  print "" >> "g09.com"
  }
```

prog2ndpoint

```
BEGIN {
# Nov 2016 Created
# this program creates the input files for our homegrown ONIOM's second point

initializeparameters()
initializeconstants()
readprogdynconf()

getline < "isomernumber"
isomernum = $1
getline < "runpointnumber"
runpointnum = $1

if (diag>=1) diagnosticsA()

#get forward or reverse from skipstart if it exists
getline < "skipstart"
trajdirection = $1

readgeoPlusVel()
putPoint1Traj()
addVelocities()
} # end of BEGIN

#pull out the potential energy
/FINAL HEAT OF FORMATION/ {
if (basename(FILENAME)=="real.out") {
  newPotEK=newPotEK+$6
  newPotentialE=newPotEK/627.509
  }
if (basename(FILENAME)=="model.out") {
  newPotEK=newPotEK-$6
  newPotentialE=newPotEK/627.509
  }
}

/KCAL/ {
if (basename(FILENAME)=="real.out") {
  if ($5=="X") forceArr[$2,1]=forceArr[$2,1]-$7*0.52917725/627.509 #must translate into Hartree/Bohr for other calcs to work
  if ($5=="Y") forceArr[$2,2]=forceArr[$2,2]-$7*0.52917725/627.509
  if ($5=="Z") forceArr[$2,3]=forceArr[$2,3]-$7*0.52917725/627.509
  }
if (basename(FILENAME)=="model.out") {
  if ($5=="X") forceArr[$2,1]=forceArr[$2,1]+$7*0.52917725/627.509 #must translate into Hartree/Bohr for other calcs to work
  if ($5=="Y") forceArr[$2,2]=forceArr[$2,2]+$7*0.52917725/627.509
  if ($5=="Z") forceArr[$2,3]=forceArr[$2,3]+$7*0.52917725/627.509
  }
}

/SCF Done/ || /EUMP2 =/ || / Energy=/ || /ONIOM:/ {
if (basename(FILENAME)=="g09.log") {
  if (($1=="Energy=") && ($3=="NIter="))  newPotentialE=newPotentialE+$2
  if ($1=="SCF") newPotentialE=newPotentialE+$5  #This was subject to a bug at one time - grep Samae in old files
  if ($2=="extrapolated")  newPotentialE=newPotentialE+$5
  if ($1=="E2") {
    tempstring=$6
    split(tempstring, arr10, "D")
    newPotentialE=newPotentialE+arr10[1]*(10^arr10[2])
    }
  newPotEK=(newPotentialE-potentialE)*627.509
  newPotentialEK=(newPotentialE-potentialE)*627.509
  }
}

# now we go ahead and collect the forces from the point 1 file
(/     1   / || /     2   / || /     3   / || /     4   / || /     5   / || /     6   / || /     7   / || /     8   / || /     9   / || /    10   / || /
11   / || /    12   / || /    13   / || /    14   / || /    15   / || /    16   / || /    17   / || /    18   / || /    19   / || /    20   / || /
21   / || /    22   / || /    23   / || /    24   / || /    25   / || /    26   / || /    27   / || /    28   / || /    29   / || /    30   / || /
31   / || /    32   / || /    33   / || /    34   / || /    35   /) && length($3) > 9 {
if (basename(FILENAME)=="g09.log") {
  i=$1
  for (j=1;j<=3;j++) {
    forceArr[i,j]=forceArr[i,j]+$(2+j)    #the raw units of the forces are Hartree/Bohr
    }
  if ((diag>1) && (i==1)) print "i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3]" >> "diagnostics"
  if (diag>1) print i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3] >> "diagnostics"
  }
}

END {
if (DRP==0) doEcheck()
addForceEffect()
writereal()
writemodel()
writeg09()
writetraj()
}

############################################################ FUNCTIONS
################################################
function Distance(Atom1,Atom2) {
  return sqrt((oldarr[Atom1,1]-oldarr[Atom2,1])^2+(oldarr[Atom1,2]-oldarr[Atom2,2])^2+(oldarr[Atom1,3]-
oldarr[Atom2,3])^2)
  }

function basename(file) {
  sub(".*/", "", file)
  return file
}

# default parameters, including quassiclassical, no displacements, transition state, not a DRP
# do not change these - rather, change progdyn.conf to set the parameters
function initializeparameters() {
  initialDis=0; timestep=1E-15; scaling=1.0; temp=298.15
  classical=0; numimag=1; DRP=0; cannonball=0
  memory=20000000
  diag=1; checkpoint="g09.chk"; searchdir="positive"; boxon=0
  boxsize=10; maxAtomMove=0.1; title1="you"; title2="need"
  title3="a"; title4="progdyn.conf"; processors=1; highlevel=99999; linkatoms=0
  damping=1;nonstandard=0;geometry="nonlinear";nonstandard=0
  nmrtype=0;nmrevery=9999999;nmrcc=0;nmrrand=0;nmrdo=0
  thermostat=0;thermostatemult=1.00
  oniomcharge=0; oniommult=0
  applyforce=0; applyforceB=0; applyforceC=0; zeroatomon=0
  sphereon=0; spheresize=999; sphereforceK=0.01
  empiricaldispersion=0; radiusmultiplier=1.25
  etolerance=1
  includeacceleration=1
  getline < "bypassproggen"
  if (($1>2) && ($1<999999999)) includeacceleration=0
  }

function initializeconstants() {
  srand(PROCINFO["pid"])
  i=1;j=1;k=1
  c=29979245800; h=6.626075E-34; avNum=6.0221415E23
  RgasK=0.00198588; RgasJ=8.31447; pi=3.14159265359
  numAtoms=0; atomnumber=0
  conver1=4.184E26 #dividing by this converts amu angs^2 /s^2 to kcal/mol
  OFS="   "
  }

# read progdyn.conf for configuration info
function readprogdynconf() {
  blankLineTester=10
  while (blankLineTester>1) {
    getline < "progdyn.conf"
    if ($1=="method") method=$2
    if ($1=="lowmethod") lowmethod=$2
    if ($1=="method2") meth2=$2
    if ($1=="charge") charge=$2
    if ($1=="multiplicity") multiplicity=$2
    if ($1=="oniomchargemult") {
      oniomcharge=$2
      oniommult=$3
      }
    if ($1=="memory") memory=$2
    if ($1=="processors") processors=$2
    if ($1=="checkpoint") checkpoint=$2
    if ($1=="timestep") timestep=$2
    if ($1=="diagnostics") diag=$2
    if ($1=="temperature") temp=$2
    if ($1=="thermostat") thermostat=$2
    if ($1=="thermostatmult") thermostatmult=$2
    if (thermostatmult>1) thermostatmult=1/thermostatmult
    if ($1=="method3") meth3=$2
    if ($1=="method4") meth4=$2
    if ($1=="method5") meth5=$2
    if ($1=="method6") meth6=$2
    if ($1=="method7") meth7=$2
    if ($1=="highlevel") highlevel=$2
    if ($1=="linkatoms") linkatoms=$2
    if ($1=="fixedatom1") fixedatom1=$2
    if ($1=="fixedatom2") fixedatom2=$2
    if ($1=="fixedatom3") fixedatom3=$2
    if ($1=="fixedatom4") fixedatom4=$2
    if ($1=="boxon") boxon=$2
    if ($1=="boxsize") boxsize=$2
    if ($1=="sphereon") sphereon=$2
    if ($1=="spheresize") spheresize=$2
    if ($1=="sphereforce") sphereforceK=$2
    if ($1=="DRP") DRP=$2
    if ($1=="maxAtomMove") maxAtomMove=$2
    if ($1=="methodfile") methodfilelines=$2
    if ($1=="killcheck") killcheck=$2
    if ($1=="etolerance") etolerance=$2
    if ($1=="reversetraj") reversetraj=$2
    if ($1=="empiricaldispersion") empiricaldispersion=$2
    if ($1=="radiusmultiplier") radiusmultiplier=$2
    if ($1=="includeacceleration") includeacceleration=$2
    if ($1=="damping") damping=$2
    if ($1=="NMRmethod") nmrmethod=$2
    if ($1=="NMRmethod2") nmrmethod2=$2
    if ($1=="NMRmethod3") nmrmethod3=$2
    if ($1=="NMRtype") nmrtype=$2
    if ($1=="NMRevery") nmrevery=$2
    if ($1=="NMRrand") nmrrand=$2
    if ($1=="loadlimit") loadlimit=$2
    if ($1=="NMRcc") nmrcc=$2
    if ($1=="nonstandard") nonstandard=$2
    if ($1=="applyforce") {
      applyforce=$2; apforce=$3; apforceX0=$4; apforce2=$5; apforce3=$6
      }
    if ($1=="applyforceB") {
      applyforceB=$2; apforceB=$3; apforceX0B=$4; apforce2B=$5; apforce3B=$6
      }
    if ($1=="applyforceC") {
      applyforceC=$2; apforceC=$3; apforceX0C=$4; apforce2C=$5; apforce3C=$6
      }
    if ($1=="afatoms") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afatom[i]=$(i+1)
        }
      }
    if ($1=="afatomsB") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afatomB[i]=$(i+1)
        }
      }
    if ($1=="afatomsC") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afatomC[i]=$(i+1)
        }
      }
    if ($1=="applyforceplane") {
      applyforceplane=$2; apforceplane=$3; apforceplaneX0=$4
      }
    if ($1=="afplaneatoms") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afplaneatoms[i]=$(i+1)
        }
      }
    if ($1=="zeroatom") {
      zeroatomon=1
      zeroatom=$2
      }
    if ($1=="title") {
      title1=$2
      title2=$3
      title3=$4
      title4=$5
      }
    blankLineTester=length($0)
    }
  }

function diagnosticsA() {
  print "***************** starting progdynb *****************" >> "diagnostics"
  print "method,charge,multiplicity,memory" >> "diagnostics"
  print method,charge,multiplicity,memory >> "diagnostics"
  print "processors,checkpoint,title" >> "diagnostics"
  print processors,checkpoint,title1,title2,title3,title4 >> "diagnostics"
  }

function readgeoPlusVel() {
#this is different from prog1stpoint and progdynb because we are just getting velocities
#read in number of atoms, geometry, masses from geoPlusVel
  getline < "geoPlusVel"
  numAtoms=$1
# geometry
  for (i=1;i<=numAtoms;i++) {
    getline < "geoPlusVel"
    weight[i]=$5
    atSym[i]=$1
    for (j=1;j<=3;j++) {
      geoArr[i,j]=$(1+j)
      }
    }
#velocities
  for (i=1;i<=numAtoms;i++) {
    getline < "geoPlusVel"
    for (j=1;j<=3;j++) {
      velArr[i,j]=$j
      }
    }
#pull out other information useful for testing whether total energy is right or bad
  blankLineTester=10
  while (blankLineTester>1) {
    getline < "geoPlusVel"
    if ($4=="desired=") desiredModeEnK=$5
    if ($4=="modes=") {
      KEinitmodes=$5
      KEinittotal=$9
      }
    if ($11=="potential") potentialE=$13
    blankLineTester=length($0)
    }
  }

function addVelocities() {
# ok, now we have to figure the second point.  this should be
# x(t) = x + v*t + 1/2*F*t^2/m
#now we go ahead and add the velocities
  for (i=1;i<=numAtoms;i++) {
    for (j=1;j<=3;j++) {
      arr[i,j]=velArr[i,j]+geoArr[i,j]
      if (trajdirection=="reverserestart") arr[i,j]=geoArr[i,j]-velArr[i,j]
      }
    if ((diag>1) && (i==1)) print "geometry after adding velocities" >> "diagnostics"
    if (diag>1) print arr[i,1],arr[i,2],arr[i,3] >> "diagnostics"
    }
  }

function putPoint1Traj() {
#get initial geometry into file traj
  print numAtoms >> "traj"
  print potentialE,title1,title2,title3,title4,"runpoint 1 ","runisomer ",isomernum >> "traj"
  for (i=1;i<=numAtoms;i++) {
    print atSym[i],geoArr[i,1],geoArr[i,2],geoArr[i,3] >> "traj"
    }
  }

function doEcheck() {
  print "trajectory #",isomernum >> "Echeck"
  print "point 1 potential E=",newPotentialEK,"   point 1 kinetic E=",KEinitmodes,"  Total=",newPotentialEK+KEinitmodes >>
"Echeck"
  print "desired total energy=", desiredModeEnK >> "Echeck"
  if ((newPotentialEK+KEinitmodes)>(desiredModeEnK+etolerance)) print "XXXX bad total Energy" >> "Echeck"
  if ((newPotentialEK+KEinitmodes)<(desiredModeEnK-etolerance)) print "XXXX bad total Energy" >> "Echeck"
  }

function addForceEffect() {
# x(t) = x + v*t + 1/2*F*t^2/m
# turn the forces into motion
  for (i=1;i<=numAtoms;i++) {
    for (j=1;j<=3;j++) {
# conversions here take force to J/angstrom, 1E20 converts to kg angstroms / s^2, then mult time (s^s) and divide by weight in kg
to get angstroms
      forceArr[i,j]=0.5*1E20*forceArr[i,j]*627.509*(4184/(0.529177*avNum))*(timestep^2)/(weight[i]/(avNum*1000))
# for simplicity, DRPs will throw away the forces at the second pont.  This means that if we are not at a saddlepoint, point 2 =
point 1 but this is a minor waste
# restarts using bypassproggen should also normally use no acceleration so that the second point is exactly what it was in the
starting structure
# but if this is a rerun then it should - set this by putting in parameter includeacceleration
      if (DRP==1) forceArr[i,j]=0
      arr[i,j]=arr[i,j]+includeacceleration*forceArr[i,j]
# if atoms are fixed, replace calcd new position by original position
      if ((i==fixedatom1) || (i==fixedatom2) || (i==fixedatom3) || (i==fixedatom4)) arr[i,j]=geoArr[i,j]
      }
    if ((diag>1) && (i==1)) print "i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3]" >> "diagnostics"
    if (diag>1) print i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3] >> "diagnostics"
    }
  }

function writereal() {
  print "XYZ GRADIENTS 1SCF PRTXYZ",lowmethod,"THREADS=" processors,"CHARGE=" charge
  print title1,title2,title3,title4,"runpoint ",runpointnum,"runisomer ",isomernum
  print ""
  for (i=1;i<=numAtoms;i++) {
    printf("%s %.7f %s %.7f %s %.7f %s ",atSym[i],arr[i,1],"1",arr[i,2],"1",arr[i,3],"1")
    print ""
    }
  print ""
  }

function writemodel() {
  print "XYZ GRADIENTS 1SCF PRTXYZ",method,"THREADS=" processors,"CHARGE=" oniomcharge > "model.mop"
  print title1,title2,title3,title4,"runpoint ",runpointnum,"runisomer ",isomernum >> "model.mop"
  print "" >> "model.mop"
  if (highlevel>numAtoms) highlevel=numAtoms
  for (i=1;i<=highlevel;i++) {
    printf("%s %.7f %s %.7f %s %.7f %s ",atSym[i],arr[i,1],"1",arr[i,2],"1",arr[i,3],"1") >> "model.mop"
    print "" >> "model.mop"
    }
  print "" >> "model.mop"
  }

function writeg09() {
  print "%nproc=" processors > "g09.com"
  print "%mem=" memory >> "g09.com"
  if (killcheck!=1) print "%chk=" checkpoint >> "g09.com"
  if (nonstandard==0) {
    print "#p " method " force scf=(xqc,maxconven=155,fulllinear,nosym) " >> "g09.com"
    if (meth2=="unrestricted") print "guess=mix" >> "g09.com" #for unrestricted calculations
    if (length(meth3)>2) print meth3 >> "g09.com"
    if (length(meth4)>2) print meth4 >> "g09.com"
    }
  if (nonstandard==1) {
    print "# " >> "g09.com"
    print "nonstd"  >> "g09.com"
    system("cat nonstandard >> g09.com")
    }
  print "" >> "g09.com"
  print  title1,title2,title3,title4 >> "g09.com"
  print "runpoint ",runpointnum >> "g09.com"
  print "runisomer ",isomernum >> "g09.com"
  if (DRP==1) {
    print "maxForce and forceMult and maxAtomMove",maxForce,forceMult,maxAtomMove >> "g09.com"
    print "maxForce and forceMult and maxAtomMove",maxForce,forceMult,maxAtomMove >> "movelist"
    }
  print "" >> "g09.com"
  print oniomcharge,oniommult >> "g09.com"
  if (highlevel>numAtoms) highlevel=numAtoms
  for (i=1;i<=highlevel;i++) {
    printf("%s %.7f %.7f %.7f",atSym[i],arr[i,1],arr[i,2],arr[i,3]) >> "g09.com"
    print "" >> "g09.com"
    }
  print "" >> "g09.com"
  if (length(meth5)>2) print meth5 >> "g09.com"
  if (length(meth6)>2) print meth6 >> "g09.com"
  if (methodfilelines>=1) {
    for (i=1;i<=methodfilelines;i++) {
      getline < "methodfile"
      print $0 >> "g09.com"
      }
    }
  print "" >> "g09.com"
  }

function writetraj() {
#get second geometry into file traj
  print numAtoms >> "traj"
  print newPotentialE,title1,title2,title3,title4,"runpoint ",runpointnum,"runisomer ",isomernum >> "traj"
  for (i=1;i<=numAtoms;i++) {
    print atSym[i],arr[i,1],arr[i,2],arr[i,3] >> "traj"
    }
  }
```

progdynb

```
BEGIN { #this is the main routine for generating new .com files by the Verlet algorithym
# Nov 2016 created for progdynONIOM
initializeparameters()
initializeconstants()
readprogdynconf()

getline < "isomernumber"
isomernum = $1
getline < "runpointnumber"
runpointnum = $1

if (diag>=1) diagnosticsA()
readgeoPlusVelAndoldAndolder() #readgeoPlusVelAndoldAndolder sets numAtoms, weight[i], atSym[i], potentialE,
oldarr[at,1to3], olderarr[at,1to3]
        # also sets apparentTemp,  KEold, does sanitycheck
if (DRP==1) readoldAdjForcesAndmaxMove() #for DRPs read in oldForce[at,123] and maxAtomMove
nmrstuffA() # sets nmrdo to 0 or 1, controls where NMR calcs are done
}
#######################  END OF BEGIN  #################################

#pull out the potential energy
/FINAL HEAT OF FORMATION/ {
if (basename(FILENAME)=="real.out") {
  newPotEK=newPotEK+$6
  newPotentialE=newPotEK/627.509
  }
if (basename(FILENAME)=="model.out") {
  newPotEK=newPotEK-$6
  newPotentialE=newPotEK/627.509
  }
}

/KCAL/ {
if (basename(FILENAME)=="real.out") {
#  if (($2==2) && ($5=="X")) print FILENAME,$0,forceArr[$2,1],forceArr[$2,2],forceArr[$2,1] >> "forcelist"
  if ($5=="X") forceArr[$2,1]=forceArr[$2,1]-$7*0.52917725/627.509 #must translate into Hartree/Bohr for other calcs to work
  if ($5=="Y") forceArr[$2,2]=forceArr[$2,2]-$7*0.52917725/627.509
  if ($5=="Z") forceArr[$2,3]=forceArr[$2,3]-$7*0.52917725/627.509
#  if (($2==2) && ($5=="Z")) print FILENAME,$0,forceArr[$2,1],forceArr[$2,2],forceArr[$2,1] >> "forcelist"
  }
if (basename(FILENAME)=="model.out") {
#  if (($2==2) && ($5=="X")) print FILENAME,$0,forceArr[$2,1],forceArr[$2,2],forceArr[$2,1] >> "forcelist"
  if ($5=="X") forceArr[$2,1]=forceArr[$2,1]+$7*0.52917725/627.509 #must translate into Hartree/Bohr for other calcs to work
  if ($5=="Y") forceArr[$2,2]=forceArr[$2,2]+$7*0.52917725/627.509
  if ($5=="Z") forceArr[$2,3]=forceArr[$2,3]+$7*0.52917725/627.509
#  if (($2==2) && ($5=="Z")) print FILENAME,$0,forceArr[$2,1],forceArr[$2,2],forceArr[$2,1] >> "forcelist"
  }
}

/SCF Done/ || /EUMP2 =/ || / Energy=/ || /ONIOM:/ {
if (basename(FILENAME)=="g09.log") {
  if (($1=="Energy=") && ($3=="NIter="))  newPotentialE=newPotentialE+$2
  if ($1=="SCF") newPotentialE=newPotentialE+$5  #This was subject to a bug at one time - grep Samae in old files
  if ($2=="extrapolated")  newPotentialE=newPotentialE+$5
  if ($1=="E2") {
    tempstring=$6
    split(tempstring, arr10, "D")
    newPotentialE=newPotentialE+arr10[1]*(10^arr10[2])
    }
  newPotEK=(newPotentialE-potentialE)*627.509
  newPotentialEK=(newPotentialE-potentialE)*627.509
  }
}
```

```
(/      1   / || /      2   / || /      3   / || /      4   / || /      5   / || /      6   / || /      7   / || /      8   / || /      9   / || /     10   / || /
11   / || /     12   / || /     13   / || /     14   / || /     15   / || /     16   / || /     17   / || /     18   / || /     19   / || /     20   / || /
21   / || /     22   / || /     23   / || /     24   / || /     25   / || /     26   / || /     27   / || /     28   / || /     29   / || /     30   / || /
31   / || /     32   / || /     33   / || /     34   / || /     35   /) && length($3) > 9 {
if (basename(FILENAME)=="g09.log") {
  i=$1
  for (j=1;j<=3;j++) {
    forceArr[i,j]=forceArr[i,j]+$(2+j)    #the raw units of the forces are Hartree/Bohr
    }
  if ((diag>1) && (i==1)) print "i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3]" >> "diagnostics"
  if (diag>1) print i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3] >> "diagnostics"
  }
}

END {
if (sphereon==1) applysphereforce() # apply a force to bring atoms within a sphere.   This also figures out the density at
0.9*spheresize
# routines to apply forces between atoms, used for umbrella sampling
if (applyforceplane>0) doapplyforceplane() #used to apply a force on an atom versus its distance from a plane
setinterlockingsphereatom() #apply force to one of a series of atoms, whichever is closest to afatom[1]
if (applyforce>0) doapplyforce()
if (applyforceB>0) doapplyforceB()
if (applyforceC>0) doapplyforceC()
if (s6!=0) doempiricaldispersion()
if (s6b!=0) {
  s6=s6b; radiusmultiplier=radiusmultiplierb; EdispHtotal=0
  doempiricaldispersion()
  }
if (zeroatomon==1) dozeroatom() #routine to slowly move an atom toward the origin
if (specialforce>0) dospecialforce()
if (thermostat==1) dothermostat()
if (DRP==1) doDRP() #routine for steepest descent path in mass weighted coordinates
if (DRP==0) doVerlet() #normal routine for Verlet
writereal()
writemodel()
writeg09()
writetraj()
}

############################################################ FUNCTIONS
################################################
function Distance(Atom1,Atom2) {
  return sqrt((oldarr[Atom1,1]-oldarr[Atom2,1])^2+(oldarr[Atom1,2]-oldarr[Atom2,2])^2+(oldarr[Atom1,3]-
oldarr[Atom2,3])^2)
  }

function basename(file) {
  sub(".*/", "", file)
  return file
}

# default parameters, including quassiclassical, no displacements, transition state, not a DRP
# do not change these - rather, change progdyn.conf to set the parameters
function initializeparameters() {
  initialDis=0; timestep=1E-15; scaling=1.0; temp=298.15
  classical=0; numimag=1; DRP=0; cannonball=0
  memory=20000000
  diag=1; checkpoint="g09.chk"; searchdir="positive"; boxon=0
  boxsize=10; maxAtomMove=0.1; title1="you"; title2="need"
  title3="a"; title4="progdyn.conf"; processors=1; highlevel=99999; linkatoms=0
  damping=1;nonstandard=0
  nmrtype=0;nmrevery=9999999;nmrcc=0;nmrrand=0;nmrdo=0
  thermostat=0;thermostatemult=1.00
  oniomcharge=0; oniommult=0
  applyforce=0; applyforceB=0; applyforceC=0; zeroatomon=0
  sphereon=0; spheresize=999; sphereforceK=0.01
  s6=0; radiusmultiplier=1.0; s6b=0; radiusmultiplierb=1.0
  EdispHtotal=0; specialforce=0
  }

function initializeconstants() {
  srand(PROCINFO["pid"])
  i=1;j=1;k=1
  c=29979245800; h=6.626075E-34; avNum=6.0221415E23
  RgasK=0.00198588; RgasJ=8.31447; pi=3.14159265359
  numAtoms=0; atomnumber=0
  conver1=4.184E26 #dividing by this converts amu angs^2 /s^2 to kcal/mol
  OFS="   "
  }

# read progdyn.conf for configuration info
function readprogdynconf() {
  blankLineTester=10
  while (blankLineTester>1) {
    getline < "progdyn.conf"
    if ($1=="method") method=$2
    if ($1=="lowmethod") lowmethod=$2
    if ($1=="method2") meth2=$2
    if ($1=="charge") charge=$2
    if ($1=="multiplicity") multiplicity=$2
    if ($1=="oniomchargemult") {
      oniomcharge=$2
      oniommult=$3
      }
    if ($1=="memory") memory=$2
    if ($1=="processors") processors=$2
    if ($1=="checkpoint") checkpoint=$2
    if ($1=="timestep") timestep=$2
    if ($1=="diagnostics") diag=$2
    if ($1=="temperature") temp=$2
    if ($1=="thermostat") thermostat=$2
    if ($1=="thermostatmult") thermostatmult=$2
    if (thermostatmult>1) thermostatmult=1/thermostatmult
    if ($1=="method3") meth3=$2
    if ($1=="method4") meth4=$2
    if ($1=="method5") meth5=$2
    if ($1=="method6") meth6=$2
    if ($1=="method7") meth7=$2
    if ($1=="highlevel") highlevel=$2
    if ($1=="linkatoms") linkatoms=$2
    if ($1=="fixedatom1") fixedatom1=$2
    if ($1=="fixedatom2") fixedatom2=$2
    if ($1=="fixedatom3") fixedatom3=$2
    if ($1=="fixedatom4") fixedatom4=$2
    if ($1=="boxon") boxon=$2
    if ($1=="boxsize") boxsize=$2
    if ($1=="sphereon") sphereon=$2
    if ($1=="spheresize") spheresize=$2
    if ($1=="sphereforce") sphereforceK=$2
    if ($1=="DRP") DRP=$2
    if ($1=="maxAtomMove") maxAtomMove=$2
    if ($1=="methodfile") methodfilelines=$2
    if ($1=="killcheck") killcheck=$2
    if ($1=="empiricaldispersion") s6=$2
    if ($1=="radiusmultiplier") radiusmultiplier=$2
    if ($1=="empiricaldispersionb") s6b=$2
    if ($1=="radiusmultiplierb") radiusmultiplierb=$2
    if ($1=="damping") damping=$2
    if ($1=="NMRmethod") nmrmethod=$2
    if ($1=="NMRmethod2") nmrmethod2=$2
    if ($1=="NMRmethod3") nmrmethod3=$2
    if ($1=="NMRtype") nmrtype=$2
    if ($1=="NMRevery") nmrevery=$2
    if ($1=="NMRrand") nmrrand=$2
    if ($1=="loadlimit") loadlimit=$2
    if ($1=="NMRcc") nmrcc=$2
    if ($1=="nonstandard") nonstandard=$2
    if ($1=="specialforce") specialforce=$2
    if ($1=="applyforce") {
      applyforce=$2; apforce=$3; apforceX0=$4; apforce2=$5; apforce3=$6
      }
    if ($1=="applyforceB") {
      applyforceB=$2; apforceB=$3; apforceX0B=$4; apforce2B=$5; apforce3B=$6
      }
    if ($1=="applyforceC") {
      applyforceC=$2; apforceC=$3; apforceX0C=$4; apforce2C=$5; apforce3C=$6
      }
    if ($1=="afatoms") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afatom[i]=$(i+1)
        }
      }
    if ($1=="afatomsB") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afatomB[i]=$(i+1)
        }
      }
    if ($1=="afatomsC") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afatomC[i]=$(i+1)
        }
      }
    if ($1=="applyforceplane") {
      applyforceplane=$2; apforceplane=$3; apforceplaneX0=$4
      }
    if ($1=="afplaneatoms") {
      for (i=1;i<8;i++) {
        if ($(i+1)>0) afplaneatoms[i]=$(i+1)
        }
      }
    if ($1=="zeroatom") {
      zeroatomon=1
      zeroatom=$2
      }
    if ($1=="title") {
      title1=$2
      title2=$3
      title3=$4
      title4=$5
      }
    blankLineTester=length($0)
    }
  }

function diagnosticsA() {
  print "**************** starting progdynb ****************" >> "diagnostics"
  print "method,charge,multiplicity,memory" >> "diagnostics"
  print method,charge,multiplicity,memory >> "diagnostics"
  print "processors,checkpoint,title" >> "diagnostics"
  print processors,checkpoint,title1,title2,title3,title4 >> "diagnostics"
  }

function readgeoPlusVelAndoldAndolder() {
# get number of atoms and weights from geoPlusVel, and previous geometries from old and older
  getline < "geoPlusVel"
  numAtoms=$1
  for (i=1;i<=numAtoms;i++) {
    getline < "geoPlusVel"
    weight[i]=$5; atSym[i]=$1
    }
  blankLineTester=10
  while (blankLineTester>1) {
    getline < "geoPlusVel"
    if ($11=="potential") potentialE=$13
    blankLineTester=length($0)
    }

  for (at=1;at<=numAtoms;at++) {
    getline < "old"
    oldarr[at,1]=$3; oldarr[at,2]=$5; oldarr[at,3]=$7
    }

  for (at=1;at<=numAtoms;at++) {
    getline < "older"
    olderarr[at,1]=$3; olderarr[at,2]=$5; olderarr[at,3]=$7
    atomVel=((oldarr[at,1]-olderarr[at,1])^2 + (oldarr[at,2]-olderarr[at,2])^2 +(oldarr[at,3]-olderarr[at,3])^2)^.5
    KEold=KEold+0.5*weight[at]*(atomVel^2)/((timestep^2)*conver1)
# sanity check - avoids trajectory blow up
    if (atomVel>0.5) exit
    }
  apparentTemp=KEold*2/(3*RgasK*numAtoms) # this is not the temperature at the previous point but rather 1.5 points back.
# sanity check - avoids trajectory blow up
  for (at=1;at<=numAtoms;at++) {
    if (((oldarr[at,1]-olderarr[at,1])^2)>1) exit
    }
  }

function readoldAdjForcesAndmaxMove() {
#for DRPs read in oldAdjForces and maxAtomMove
  for (at=1;at<=numAtoms;at++) {
    getline < "oldAdjForces"
    oldForce[at,1]=$1; oldForce[at,2]=$2; oldForce[at,3]=$3
    }
  getline < "maxMove"
  if (($1<maxAtomMove) && ($1>0)) maxAtomMove=$1
  if (maxAtomMove<0.000001) maxAtomMove=0.000001
  }

function nmrstuffA() {
# sets nmrdo to 0 or 1, controls where NMR calcs are done
  if ((nmrrand==0) && ((runpointnum % nmrevery)==0)) nmrdo=1
  if ((nmrrand==1) && (rand()<(1/nmrevery))) nmrdo=1
  getline < "uptimelist"
  x=1.0001*substr($10,1,3);if (x<8) x=8
# turn of nmrs if load is too high - this is under control of loadlimit parameter in progdyn.conf and requires proganal to make uptimelist
  if ((nmrrand==1) && (x>loadlimit)) nmrdo=0
```

```
 }

function applysphereforce() {
# apply a force to bring atoms within a sphere.   This also figures out the density at 0.9*spheresize
# We are going through the loop twice, the first time figure outs the total pressure, then scales the forces to limit the pressure to
maxpressure
# the second time actually applies the pressure
 maxpressure=100000 #atmospheres.  Later we may parameterize this. Setting this really high eliminates the first scaling
 sphereforcetotal=0
 for (i=1;i<=numAtoms;i++) {
   distToOrig=((oldarr[i,1]^2+oldarr[i,2]^2+oldarr[i,3]^2)^.5)
   if (distToOrig>spheresize) {
     sphereforce=sphereforceK*(distToOrig-spheresize)
     if (sphereforce>0.01) sphereforce=0.01 #important limit on force for atoms far outside of the sphere, not sure if value
chosen is best
     sphereforcetotal=sphereforcetotal+sphereforce
     }
   }
 sphereforcetotalNewtons=sphereforcetotal*627.509*4184*1E10/(0.529177*avNum)
 surfaceareaSqMeters=4*pi*spheresize^2/1E20
 pressurePascal=sphereforcetotalNewtons/surfaceareaSqMeters
 pressureAtm=pressurePascal/101325
 if (pressureAtm>maxpressure) sphereforceK=sphereforceK*maxpressure/pressureAtm
# go through the loop again
 sphereforcetotal=0; totalweight=0
 for (i=1;i<=numAtoms;i++) {
   distToOrig=((oldarr[i,1]^2+oldarr[i,2]^2+oldarr[i,3]^2)^.5)
   if (distToOrig>spheresize) {
     sphereforce=sphereforceK*(distToOrig-spheresize)
     if (sphereforce>0.01) sphereforce=0.01 #important limit on force for atoms far outside of the sphere, not sure if value
chosen is best
     sphereforcetotal=sphereforcetotal+sphereforce

unitX=sphereforce*oldarr[i,1]/distToOrig;unitY=sphereforce*oldarr[i,2]/distToOrig;unitZ=sphereforce*oldarr[i,3]/distToOrig
     forceArr[i,1]=forceArr[i,1]-unitX;forceArr[i,2]=forceArr[i,2]-unitY;forceArr[i,3]=forceArr[i,3]-unitZ
     }
# calculate the density at 0.9*spheresize
   if (distToOrig<0.9*spheresize) {
     totalweight=totalweight+weight[i]
     }
   }
#recalculate pressure
 sphereforcetotalNewtons=sphereforcetotal*627.509*4184*1E10/(0.529177*avNum)
 surfaceareaSqMeters=4*pi*spheresize^2/1E20
 pressurePascal=sphereforcetotalNewtons/surfaceareaSqMeters
 pressureAtm=pressurePascal/101325
 density=(totalweight/avNum)/((4/3)*pi*(0.9*spheresize*1E-8)^3)
 }

function doapplyforceplane() {
#used to apply a force on an atom versus its distance from a plane
 if (afplaneatoms[4]<.5) {
   print "you need more atoms to define a plane"
   exit
   }
 for (i=2;i<8;i++) {
   if (afplaneatoms[i]>.5) {
     k=afplaneatoms[i]
     A[1,1]=A[1,1]+oldarr[k,1]^2;A[1,2]=A[1,2]+oldarr[k,1]*oldarr[k,2];A[1,3]=A[1,3]+oldarr[k,1]
     A[2,1]=A[2,1]+oldarr[k,1]*oldarr[k,2];A[2,2]=A[2,2]+oldarr[k,2]^2;A[2,3]=A[2,3]+oldarr[k,2]
     A[3,1]=A[3,1]+oldarr[k,1];A[3,2]=A[3,2]+oldarr[k,2];A[3,3]++
     b[1]=b[1]+oldarr[k,1]*oldarr[k,3];b[2]=b[2]+oldarr[k,2]*oldarr[k,3];b[3]=b[3]+oldarr[k,3]
# find center of mass assuming all atoms same weight
      cmass[1]=cmass[1]+oldarr[k,1];cmass[2]=cmass[2]+oldarr[k,2];cmass[3]=cmass[3]+oldarr[k,3];
      }
    }
  numplaneatoms=A[3,3]
  cmass[1]=cmass[1]/A[3,3];cmass[2]=cmass[2]/A[3,3];cmass[3]=cmass[3]/A[3,3]
#   print "matrix A"
#   for (i=1;i<=3;i++) {
#     print A[i,1],A[i,2],A[i,3]
#     }
#   print "matrix b"
#   print b[1],b[2],b[3]
  Det=A[1,1]*A[2,2]*A[3,3]+A[1,2]*A[2,3]*A[3,1]+A[1,3]*A[2,1]*A[3,2]-A[1,3]*A[2,2]*A[3,1]-A[1,2]*A[2,1]*A[3,3]-
A[1,1]*A[2,3]*A[3,2]
  E=(b[1]*A[2,2]*A[3,3]+A[1,2]*A[2,3]*b[3]+A[1,3]*b[2]*A[3,2]-A[1,3]*A[2,2]*b[3]-A[1,2]*b[2]*A[3,3]-
b[1]*A[2,3]*A[3,2])/Det
  F=(A[1,1]*b[2]*A[3,3]+b[1]*A[2,3]*A[3,1]+A[1,3]*A[2,1]*b[3]-A[1,3]*b[2]*A[3,1]-b[1]*A[2,1]*A[3,3]-
A[1,1]*A[2,3]*b[3])/Det
  G=(A[1,1]*A[2,2]*b[3]+A[1,2]*b[2]*A[3,1]+b[1]*A[2,1]*A[3,2]-b[1]*A[2,2]*A[3,1]-A[1,2]*A[2,1]*b[3]-
A[1,1]*b[2]*A[3,2])/Det
#  E, F, and G are the coefficients in the plane z=Ex+Fy+G"
# make a function that tests the fit
  for (i=2;i<8;i++) {
    if (afplaneatoms[i]>.5) {
      k=afplaneatoms[i]
      distplane=(-oldarr[k,1]*E-oldarr[k,2]*F+oldarr[k,3]-G)/(E^2+F^2+1)^.5;if (distplane<0) distplane=-distplane
      aberror=aberror+distplane
      }
    }
# if the fit is bad, as can happen, decrease to 3 atoms in plane, using only the first three atoms in list after the conatom
  if (aberror>2) {
    print "original aberror",aberror >> "diagnostics"
    A[1,1]=0;A[1,2]=0;A[1,3]=0;A[2,1]=0;A[2,2]=0;A[2,3]=0;A[3,1]=0;A[3,2]=0;A[3,3]=0
    for (i=2;i<5;i++) {
      k=afplaneatoms[i]
      A[1,1]=A[1,1]+oldarr[k,1]^2;A[1,2]=A[1,2]+oldarr[k,1]*oldarr[k,2];A[1,3]=A[1,3]+oldarr[k,1]
      A[2,1]=A[2,1]+oldarr[k,1]*oldarr[k,2];A[2,2]=A[2,2]+oldarr[k,2]^2;A[2,3]=A[2,3]+oldarr[k,2]
      A[3,1]=A[3,1]+oldarr[k,1];A[3,2]=A[3,2]+oldarr[k,2];A[3,3]++
      b[1]=b[1]+oldarr[k,1]*oldarr[k,3];b[2]=b[2]+oldarr[k,2]*oldarr[k,3];b[3]=b[3]+oldarr[k,3]
      }
    Det=A[1,1]*A[2,2]*A[3,3]+A[1,2]*A[2,3]*A[3,1]+A[1,3]*A[2,1]*A[3,2]-A[1,3]*A[2,2]*A[3,1]-A[1,2]*A[2,1]*A[3,3]-
A[1,1]*A[2,3]*A[3,2]
    E=(b[1]*A[2,2]*A[3,3]+A[1,2]*A[2,3]*b[3]+A[1,3]*b[2]*A[3,2]-A[1,3]*A[2,2]*b[3]-A[1,2]*b[2]*A[3,3]-
b[1]*A[2,3]*A[3,2])/Det
    F=(A[1,1]*b[2]*A[3,3]+b[1]*A[2,3]*A[3,1]+A[1,3]*A[2,1]*b[3]-A[1,3]*b[2]*A[3,1]-b[1]*A[2,1]*A[3,3]-
A[1,1]*A[2,3]*b[3])/Det
    G=(A[1,1]*A[2,2]*b[3]+A[1,2]*b[2]*A[3,1]+b[1]*A[2,1]*A[3,2]-b[1]*A[2,2]*A[3,1]-A[1,2]*A[2,1]*b[3]-
A[1,1]*b[2]*A[3,2])/Det
    aberror=0
    for (i=2;i<5;i++) {
      k=afplaneatoms[i]
      distplane=(-oldarr[k,1]*E-oldarr[k,2]*F+oldarr[k,3]-G)/(E^2+F^2+1)^.5;if (distplane<0) distplane=-distplane
      aberror=aberror+distplane
      }
    }
  conatom=afplaneatoms[1]
  distplane=(-oldarr[conatom,1]*E-oldarr[conatom,2]*F+oldarr[conatom,3]-G)/(E^2+F^2+1)^.5
  if (distplane<0) distplane=-distplane
  t=(oldarr[conatom,1]*E+oldarr[conatom,2]*F-oldarr[conatom,3]+G)/(E^2+F^2+1)
  planepoint[1]=oldarr[conatom,1]-t*E;planepoint[2]=oldarr[conatom,2]-t*F;planepoint[3]=oldarr[conatom,3]+t
  distcmass=((planepoint[1]-cmass[1])^2+(planepoint[2]-cmass[2])^2+(planepoint[3]-cmass[3])^2)^.5
  print "distplane",distplane,"t",t,"planepoint",planepoint[1],planepoint[2],planepoint[3],"distcmass",distcmass,"aberror",aberror
```

```
>> "diagnostics"
  if (distcmass>1.4) {
    vector[1]=planepoint[1]-cmass[1];vector[2]=planepoint[2]-cmass[2];vector[3]=planepoint[3]-cmass[3]
    vector[1]=vector[1]*1.4/distcmass;vector[2]=vector[2]*1.4/distcmass;vector[3]=vector[3]*1.4/distcmass
    planepoint[1]=cmass[1]+vector[1];planepoint[2]=cmass[2]+vector[2];planepoint[3]=cmass[3]+vector[3];
    distcmass=((planepoint[1]-cmass[1])^2+(planepoint[2]-cmass[2])^2+(planepoint[3]-cmass[3])^2)^.5
    distplane=((planepoint[1]-oldarr[conatom,1])^2+(planepoint[2]-oldarr[conatom,2])^2+(planepoint[3]-
oldarr[conatom,3])^2)^.5
    print "new distplane",distplane,"newplanepoint",planepoint[1],planepoint[2],planepoint[3],"newdistcmass",distcmass >>
"diagnostics"
    }
  delX=oldarr[conatom,1]-planepoint[1];delY=oldarr[conatom,2]-planepoint[2];delZ=oldarr[conatom,3]-planepoint[3]
  if (applyforceplane==2) apforceplane=apforceplane*(distplane-apforceplaneX0)
  unitX=apforceplane*delX/distplane;unitY=apforceplane*delY/distplane;unitZ=apforceplane*delZ/distplane
  forceArr[conatom,1]=forceArr[conatom,1]-unitX;forceArr[conatom,2]=forceArr[conatom,2]-
unitY;forceArr[conatom,3]=forceArr[conatom,3]-unitZ
  for (i=2;i<8;i++) {
    if (afplaneatoms[i]>.5) {
      platom=afplaneatoms[i]

forceArr[platom,1]=forceArr[platom,1]+unitX/numplaneatoms;forceArr[platom,2]=forceArr[platom,2]+unitY/numplaneatoms;fo
rceArr[platom,3]=forceArr[platom,3]+unitZ/numplaneatoms
      }
    }
  }

function setinterlockingsphereatom() {
#apply force to one of a series of atoms, whichever is closest to afatom[1]
  for (i=3;i<8;i++) {
    if (afatom[i]>0) {
      if(Distance(afatom[1],afatom[i])<Distance(afatom[1],afatom[2])) afatom[2]=afatom[i]
      }
    if (afatomB[i]>0) {
      if(Distance(afatomB[1],afatomB[i])<Distance(afatomB[1],afatomB[2])) afatomB[2]=afatomB[i]
      }
    if (afatomC[i]>0) {
      if(Distance(afatomC[1],afatomC[i])<Distance(afatomC[1],afatomC[2])) afatomC[2]=afatomC[i]
      }
    }
  }

function doapplyforce() {
# applyforce 1 puts a linear constant force.
# applyforce 2 puts on a harmonic restoring force to apforceX0
  delX=oldarr[afatom[1],1]-oldarr[afatom[2],1];delY=oldarr[afatom[1],2]-oldarr[afatom[2],2];delZ=oldarr[afatom[1],3]-
oldarr[afatom[2],3];
  distatoms=(delX^2+delY^2+delZ^2)^.5
  if (applyforce==2) apforce=apforce*(distatoms-apforceX0)
  if (applyforce==3) apforce=apforce*(distatoms-apforceX0) + apforce2*(distatoms-apforceX0)^2
  if (applyforce==4) apforce=apforce*(distatoms-apforceX0) + apforce2*(distatoms-apforceX0)^2 + apforce3*(distatoms-
apforceX0)^3
  unitX=apforce*delX/distatoms;unitY=apforce*delY/distatoms;;unitZ=apforce*delZ/distatoms;
  forceArr[afatom[1],1]=forceArr[afatom[1],1]-unitX;forceArr[afatom[1],2]=forceArr[afatom[1],2]-
unitY;forceArr[afatom[1],3]=forceArr[afatom[1],3]-unitZ

forceArr[afatom[2],1]=forceArr[afatom[2],1]+unitX;forceArr[afatom[2],2]=forceArr[afatom[2],2]+unitY;forceArr[afatom[2],3]=
forceArr[afatom[2],3]+unitZ
  }

function doapplyforceB() {
  delX=oldarr[afatomB[1],1]-oldarr[afatomB[2],1];delY=oldarr[afatomB[1],2]-oldarr[afatomB[2],2];delZ=oldarr[afatomB[1],3]-
oldarr[afatomB[2],3];
```

```
  distatoms=(delX^2+delY^2+delZ^2)^.5
  if (applyforceB==2) apforceB=apforceB*(distatoms-apforceX0B)
  if (applyforceB==3) apforceB=apforceB*(distatoms-apforceX0B) + apforce2B*(distatoms-apforceX0B)^2
  if (applyforceB==4) apforceB=apforceB*(distatoms-apforceX0B) + apforce2B*(distatoms-apforceX0B)^2 +
apforce3B*(distatoms-apforceX0B)^3
  unitX=apforceB*delX/distatoms;unitY=apforceB*delY/distatoms;;unitZ=apforceB*delZ/distatoms;
  forceArr[afatomB[1],1]=forceArr[afatomB[1],1]-unitX;forceArr[afatomB[1],2]=forceArr[afatomB[1],2]-
unitY;forceArr[afatomB[1],3]=forceArr[afatomB[1],3]-unitZ

forceArr[afatomB[2],1]=forceArr[afatomB[2],1]+unitX;forceArr[afatomB[2],2]=forceArr[afatomB[2],2]+unitY;forceArr[afatom
B[2],3]=forceArr[afatomB[2],3]+unitZ
  }

function doapplyforceC() {
  delX=oldarr[afatomC[1],1]-oldarr[afatomC[2],1];delY=oldarr[afatomC[1],2]-oldarr[afatomC[2],2];delZ=oldarr[afatomC[1],3]-
oldarr[afatomC[2],3];
  distatoms=(delX^2+delY^2+delZ^2)^.5
  if (applyforceC==2) apforceC=apforceC*(distatoms-apforceX0C)
  if (applyforceC==3) apforceC=apforceC*(distatoms-apforceX0C) + apforce2C*(distatoms-apforceX0C)^2
  if (applyforceC==4) apforceC=apforceC*(distatoms-apforceX0C) + apforce2C*(distatoms-apforceX0C)^2 +
apforce3C*(distatoms-apforceX0C)^3
  unitX=apforceC*delX/distatoms;unitY=apforceC*delY/distatoms;;unitZ=apforceC*delZ/distatoms;
  forceArr[afatomC[1],1]=forceArr[afatomC[1],1]-unitX;forceArr[afatomC[1],2]=forceArr[afatomC[1],2]-
unitY;forceArr[afatomC[1],3]=forceArr[afatomC[1],3]-unitZ

forceArr[afatomC[2],1]=forceArr[afatomC[2],1]+unitX;forceArr[afatomC[2],2]=forceArr[afatomC[2],2]+unitY;forceArr[afatom
C[2],3]=forceArr[afatomC[2],3]+unitZ
  }

function doempiricaldispersion() {
  for (i=1;i<=numAtoms;i++) {
    if (atSym[i]=="H") {c6[i]=0.14;r0[i]=1.001}
#    if (atSym[i]=="H") {c6[i]=0.16;r0[i]=1.11}
    if (atSym[i]=="He") {c6[i]=0.08;r0[i]=1.012}
    if (atSym[i]=="Li") {c6[i]=1.61;r0[i]=0.825}
    if (atSym[i]=="Be") {c6[i]=1.61;r0[i]=1.408}
    if (atSym[i]=="B") {c6[i]=3.13;r0[i]=1.485}
    if (atSym[i]=="C") {c6[i]=1.75;r0[i]=1.452}
#    if (atSym[i]=="C") {c6[i]=1.65;r0[i]=1.61}
    if (atSym[i]=="N") {c6[i]=1.23;r0[i]=1.397}
#    if (atSym[i]=="N") {c6[i]=1.11;r0[i]=1.55}
    if (atSym[i]=="O") {c6[i]=0.70;r0[i]=1.342}
#    if (atSym[i]=="O") {c6[i]=0.70;r0[i]=1.49}
    if (atSym[i]=="F") {c6[i]=0.75;r0[i]=1.287}
    if (atSym[i]=="Ne") {c6[i]=0.63;r0[i]=1.243}
    if (atSym[i]=="Na") {c6[i]=5.71;r0[i]=1.144}
    if (atSym[i]=="Mg") {c6[i]=5.71;r0[i]=1.364}
    if (atSym[i]=="Al") {c6[i]=10.79;r0[i]=1.639}
    if (atSym[i]=="Si") {c6[i]=9.23;r0[i]=1.716}
    if (atSym[i]=="P") {c6[i]=7.84;r0[i]=1.705}
    if (atSym[i]=="S") {c6[i]=5.57;r0[i]=1.683}
    if (atSym[i]=="Cl") {c6[i]=5.07;r0[i]=1.639}
#    if (atSym[i]=="Cl") {c6[i]=8.00;r0[i]=1.82}
    if (atSym[i]=="Ar") {c6[i]=4.61;r0[i]=1.595}
    if (atSym[i]=="K") {c6[i]=10.8;r0[i]=1.485}
    if (atSym[i]=="Ca") {c6[i]=10.8;r0[i]=1.474}
    if (atSym[i]=="Sc") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Ti") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="V") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Cr") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Mn") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Fe") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Co") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Ni") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Cu") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Zn") {c6[i]=10.8;r0[i]=1.562}
    if (atSym[i]=="Ga") {c6[i]=16.99;r0[i]=1.65}
    if (atSym[i]=="Ge") {c6[i]=17.10;r0[i]=1.727}
    if (atSym[i]=="As") {c6[i]=16.37;r0[i]=1.76}
    if (atSym[i]=="Se") {c6[i]=12.64;r0[i]=1.771}
    if (atSym[i]=="Br") {c6[i]=12.47;r0[i]=1.749}
    if (atSym[i]=="Pd") {c6[i]=24.67;r0[i]=1.639}
    if (atSym[i]=="I") {c6[i]=31.5;r0[i]=1.892}
    r0[i]=r0[i]*radiusmultiplier
    }
  for (i=1;i<numAtoms;i++) {
#the line below turns off empirical dispersion for within the solute, since this is best handled with the high-level method
    jstart=i+1;if (jstart<highlevel+1) jstart=highlevel+1
    for (j=jstart;j<=numAtoms;j++) {
      Rij=Distance(i,j)
      if (Rij<8) { #cutoff of 8 angstroms to save time - later this can be parameterized
#need to get units of force to Hartrees/Bohr, since that is the units of forceArr
        EdispK=Edisp(c6[i],c6[j],Rij,r0[i],r0[j])*1E6/4184 #in kcal/mol
        FdispK=Fdisp(c6[i],c6[j],Rij,r0[i],r0[j])*1E6/4184 #in kcal/mol per angstrom
        EdispH=EdispK/627.509
        EdispHtotal=EdispHtotal+EdispH
        FdispHB=FdispK*0.52917725/627.509
#         print i,j,Cij,Rij,r0[i],r0[j],fdmp,"     ",EdispK,FdispK,"     ",EdispH,FdispHB,EdispHtotal
        delX=oldarr[i,1]-oldarr[j,1];delY=oldarr[i,2]-oldarr[j,2];delZ=oldarr[i,3]-oldarr[j,3];
        unitX=FdispHB*delX/Rij;unitY=FdispHB*delY/Rij;;unitZ=FdispHB*delZ/Rij;
        forceArr[i,1]=forceArr[i,1]+unitX;forceArr[i,2]=forceArr[i,2]+unitY;forceArr[i,3]=forceArr[i,3]+unitZ
        forceArr[j,1]=forceArr[j,1]-unitX;forceArr[j,2]=forceArr[j,2]-unitY;forceArr[j,3]=forceArr[j,3]-unitZ
        }
      }
    }
  print "Total empirical Dispersion Energy in Hartrees",EdispHtotal,"  in kcal/mol",EdispHtotal*627.509 >> "diagnostics"
  newPotentialE=newPotentialE+EdispHtotal
  }

function Edisp(c6i,c6j,rij,r0i,r0j) {
  cij=(c6i*c6j)^.5
  fdmp=1/(1+exp(-20*((rij/(r0i+r0j)-1))))
  return -s6*cij*fdmp/(rij^6)
  }

function Fdisp(c6i,c6j,rij,r0i,r0j) {
  delta=0.001
  return (Edisp(c6i,c6j,rij-delta,r0i,r0j)-Edisp(c6i,c6j,rij+delta,r0i,r0j))/(2*delta)
  }

function dozeroatom() {
#routine to slowly move an atom toward the origin as set by a harmonic potential
  multiple=0.99996
  oldarr[zeroatom,1]=multiple*oldarr[zeroatom,1]
  oldarr[zeroatom,2]=multiple*oldarr[zeroatom,2]
  oldarr[zeroatom,3]=multiple*oldarr[zeroatom,3]
  }

#reprogram as needed, but take care
function dospecialforce() {
  for (i=45;i<numAtoms;i++) {
    if (atSym[i]=="H") {
      j=9
      Rij=Distance(i,j)
      if (Rij<2.1) {
        delX=oldarr[i,1]-oldarr[j,1];delY=oldarr[i,2]-oldarr[j,2];delZ=oldarr[i,3]-oldarr[j,3];
        FdispHB=specialforce*(2.1-Rij)
        unitX=FdispHB*delX/Rij;unitY=FdispHB*delY/Rij;;unitZ=FdispHB*delZ/Rij;
        forceArr[i,1]=forceArr[i,1]+unitX;forceArr[i,2]=forceArr[i,2]+unitY;forceArr[i,3]=forceArr[i,3]+unitZ
        forceArr[j,1]=forceArr[j,1]-unitX;forceArr[j,2]=forceArr[j,2]-unitY;forceArr[j,3]=forceArr[j,3]-unitZ
#         print "unitX",unitX,"unitY",unitY,"unitZ",unitZ >> "diagnostics"
        }
      }
    }
  }

function dothermostat() {
#print out some things to vellist and do thermostat
# the damping in the thermostat is based on temperature based on old geo vs older geo
  if (diag<4) print "KEold",KEold,"desired temperature",temp,"apparent Temperature",apparentTemp >> "vellist"
  if (apparentTemp>temp) damping=thermostatmult
  if (apparentTemp<temp) damping=1/thermostatmult
  }

function doDRP() {
#routine for DRPs
  maxForce=0;oscillTest=0
  for (i=1;i<=numAtoms;i++) {
    for (j=1;j<=3;j++) {
# conversions here take force to J/angstrom, 1E20 converts to kg angstroms / s^2, then mult time (s^s) and divide by weight in kg
to get angstroms
      forceArr[i,j]=1E20*forceArr[i,j]*627.509*(4184/(0.529177*avNum))*(timestep^2)/(weight[i]/(avNum*1000))
      oscillTest=oscillTest+forceArr[i,j]*oldForce[i,j]
      if (forceArr[i,j]>maxForce) maxForce=forceArr[i,j]
      if ((0-forceArr[i,j])>maxForce) maxForce=-forceArr[i,j]
      }
    if (i==1) printf("% .8f % .8f % .8f \n",forceArr[1,1],forceArr[1,2],forceArr[1,3])  > "oldAdjForces"
    if (i>1) printf("% .8f % .8f % .8f \n",forceArr[i,1],forceArr[i,2],forceArr[i,3])  >> "oldAdjForces"
    }
  print "oscillTest ",oscillTest >> "oldAdjForces"
  if (oscillTest<0) {
    maxAtomMove = maxAtomMove*0.5
    print maxAtomMove > "maxMove"
    }
  if (oscillTest>0) {
    maxAtomMove = maxAtomMove*1.2
    print maxAtomMove > "maxMove"
    }
  print "maxAtomMove ",maxAtomMove >> "oldAdjForces"
  forceMult=maxAtomMove/maxForce
  for (i=1;i<=numAtoms;i++) {
    for (j=1;j<=3;j++) {
      newarr[i,j]=oldarr[i,j]+forceMult*forceArr[i,j]
      }
    }
  }

function doVerlet() {
#normal routine for Verlet
  for (i=1;i<=numAtoms;i++) {
    for (j=1;j<=3;j++) {
# conversions here take force to J/angstrom, 1E20 converts to kg angstroms / s^2, then mult time (s^s) and divide by weight in kg
to get angstroms
      forceArr[i,j]=1E20*forceArr[i,j]*627.509*(4184/(0.529177*avNum))*(timestep^2)/(weight[i]/(avNum*1000))
#       if ((diag>1) && (i==1)) print "i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3]" >> "diagnostics"
#       if (diag>1) print i,weight[i],forceArr[i,1],forceArr[i,2],forceArr[i,3] >> "diagnostics"
# examine old motion.  If an atom is moving in a crazy way then say so in diagnostics
      oldmotion=damping*(oldarr[i,j]-olderarr[i,j])
      motionlimit=0.04
      if ((oldmotion>motionlimit) || (oldmotion<-1*motionlimit)) {
        print "Atom",i,"is moving",oldmotion,"in dimension",j,"in old motion" >> "diagnostics"
#         oldmotion=oldmotion/2; print "**** Non-production motion" >> "diagnostics"   # only uncomment for non-production
to move atoms while limiting massive disruptions
        }
      newarr[i,j]=oldarr[i,j]+oldmotion+forceArr[i,j]
      if ((i==fixedatom1) || (i==fixedatom2) || (i==fixedatom3) || (i==fixedatom4)) newarr[i,j]=oldarr[i,j]
#turn around atoms outside the box
      if (boxon==1) {
        if (newarr[i,j]>boxsize) if (oldarr[i,j]>olderarr[i,j]) newarr[i,j]=oldarr[i,j]+damping*(olderarr[i,j]-oldarr[i,j])+forceArr[i,j]
        if (newarr[i,j]<-1*boxsize) if (oldarr[i,j]<olderarr[i,j]) newarr[i,j]=oldarr[i,j]+damping*(olderarr[i,j]-
oldarr[i,j])+forceArr[i,j]
        }
      }
    }
#calculate the kinetic energy.  This is for the point prior to the current point.  Corrected calculation
  for (at=1;at<=numAtoms;at++) {
    atomVel=(((olderarr[at,1]-newarr[at,1])^2 + (olderarr[at,2]-newarr[at,2])^2 +(olderarr[at,3]-newarr[at,3])^2)^.5)/2
    KEnew=KEnew+0.5*weight[at]*(atomVel^2)/((timestep^2)*conver1)
    }
    KEave=KEnew # we used to average with KEold but that is incorrect.  This is better.
    Etotal=newPotEK+KEave
#still basing apparent Temperature on velocities from old vs older, even though the KE now represents an average of old and new
    if (diag==4) print runpointnum,"KEave",KEave,"apparent
Temperature",apparentTemp,"newPotEK",newPotEK,"Etotal",Etotal,"pressure in Atm",pressureAtm,"density in 0.9r",density >>
"vellist"
  }

function writetraj() {
  print numAtoms >> "traj"
  print newPotentialE,title1,title2,title3,title4,"runpoint ",runpointnum,"runisomer ",isomernum >> "traj"
  for (i=1;i<=numAtoms;i++) {
    printf("%s %.7f %.7f %.7f",atSym[i],newarr[i,1],newarr[i,2],newarr[i,3]) >> "traj"
    print "" >> "traj"
    }
  }

function writereal() {
  print "XYZ GRADIENTS 1SCF PRTXYZ",lowmethod,"THREADS=" processors,"CHARGE=" charge
  print title1,title2,title3,title4,"runpoint ",runpointnum,"runisomer ",isomernum
  print ""
  for (i=1;i<=numAtoms;i++) {
    printf("%s %.7f %s %.7f %s %.7f %s ",atSym[i],newarr[i,1],"1",newarr[i,2],"1",newarr[i,3],"1")
    print ""
    }
  print ""
  }

function writemodel() {
  print "XYZ GRADIENTS 1SCF PRTXYZ",lowmethod,"THREADS=" processors,"CHARGE=" oniomcharge > "model.mop"
  print title1,title2,title3,title4,"runpoint ",runpointnum,"runisomer ",isomernum >> "model.mop"
  print "" >> "model.mop"
  if (highlevel>numAtoms) highlevel=numAtoms
  for (i=1;i<=highlevel;i++) {
    printf("%s %.7f %s %.7f %s %.7f %s ",atSym[i],newarr[i,1],"1",newarr[i,2],"1",newarr[i,3],"1") >> "model.mop"
    print "" >> "model.mop"
    }
  print "" >> "model.mop"
  }
```

```
function writeg09() {
  print "%nproc=" processors > "g09.com"
  print "%mem=" memory >> "g09.com"
  if (killcheck!=1) print "%chk=" checkpoint >> "g09.com"
  if (nonstandard==0) {
    print "#p " method " force scf=(xqc,maxconven=155,fulllinear,nosym) " >> "g09.com"
    if (meth2=="unrestricted") print "guess=mix" >> "g09.com" #for unrestricted calculations
    if (length(meth3)>2) print meth3 >> "g09.com"
    if (length(meth4)>2) print meth4 >> "g09.com"
    if (length(meth5)>2) print meth5 >> "g09.com"
    }
  if (nonstandard==1) {
    print "# " >> "g09.com"
    print "nonstd"  >> "g09.com"
    system("cat nonstandard >> g09.com")
    }
  print "" >> "g09.com"
  print  title1,title2,title3,title4 >> "g09.com"
  print "runpoint ",runpointnum >> "g09.com"
  print "runisomer ",isomernum >> "g09.com"
  if (DRP==1) {
    print "maxForce and forceMult and maxAtomMove",maxForce,forceMult,maxAtomMove >> "g09.com"
    print "maxForce and forceMult and maxAtomMove",maxForce,forceMult,maxAtomMove >> "movelist"
    }
  print "" >> "g09.com"
  print oniomcharge,oniommult >> "g09.com"
  if (highlevel>numAtoms) highlevel=numAtoms
  for (i=1;i<=highlevel;i++) {
    printf("%s %.7f %.7f %.7f",atSym[i],newarr[i,1],newarr[i,2],newarr[i,3]) >> "g09.com"
    print "" >> "g09.com"
    }
  print "" >> "g09.com"
  if (length(meth6)>2) print meth6 >> "g09.com"
  if (length(meth7)>2) print meth7 >> "g09.com"
  if (methodfilelines>=1) {
    for (i=1;i<=methodfilelines;i++) {
      getline < "methodfile"
      print $0 >> "g09.com"
      }
    }
  print "" >> "g09.com"
  }
```

Program proganal used for 80-THF product-forming trajectories (using ProgdynONIOM)

```
BEGIN {
firsttitle=1
getline < "isomernumber"
isomer=$1
}
/ hydroboration/ {
  if (firsttitle==1) {
    printf("%s %s %s %s %s %s %s  ",$1,$2,$3,$4,$6,$7,$8)
    runpoint=$6
    }
  firsttitle++
  }
/ATOM   CHEMICAL          X  /,/CARTESIAN COORDINATES/ {
  if (($1>.5) && ($1<99)) {
    A[$1]=$3;B[$1]=$5;C[$1]=$7
    }
  }

END {
BC1=Distance(1,2)
BC2=Distance(1,3)
HC1=Distance(14,2)
if (Distance(15,2)<HC1) HC1=Distance(15,2)
if (Distance(16,2)<HC1) HC1=Distance(16,2)
HC2=Distance(14,3)
if (Distance(15,3)<HC2) HC2=Distance(15,3)
if (Distance(16,3)<HC2) HC2=Distance(16,3)
  printf("%s %.3f %s %.3f %s %.3f %s %.3f ","BC1",BC1,"BC2",BC2,"HC1",HC1,"HC2",HC2)
  if (runpoint>500000) {
    print "Too many points.  XXXXN"
#     system("date > nogo")
    }
  if ((HC1<1.1) && (BC2<1.6)) {
    print "Mark product formed XXXX"
    }
  if ((HC2<1.1) && (BC1<1.6)) {
    print "Antimark product formed XXXX"
    }
  if ((HC1>4.5) && (HC2>4.5)) {
    print "Returning to separate SMs XXXX"
    }

  system("date '+%b:%d:%Y %T'")
  system("tail -1 Echeck | grep XXXX")
  }

function Distance(Atom1,Atom2) {
 return sqrt((A[Atom1]-A[Atom2])^2 + (B[Atom1]-B[Atom2])^2 + (C[Atom1]-C[Atom2])^2)
}

function Angle(Atom1,Atom2,Atom3) {
  value=((-
Distance(Atom1,Atom3)^2+Distance(Atom1,Atom2)^2+Distance(Atom2,Atom3)^2)/(2*Distance(Atom1,Atom2)*Distance(Ato
m2,Atom3)))
  return acos(value)
}

function asin(x) { return (180/3.141592)*atan2(x, sqrt(1-x*x)) }

function acos(x) { return (180/3.141592)*atan2(sqrt(1-x*x), x) }

function atan(x) { return (180/3.141592)*atan2(x,1) }

function Dihedral(Atom1,Atom2,Atom3,Atom4) {
  B1x=A[Atom2]-A[Atom1]
  B1y=B[Atom2]-B[Atom1]
  B1z=C[Atom2]-C[Atom1]
  B2x=A[Atom3]-A[Atom2]
  B2y=B[Atom3]-B[Atom2]
  B2z=C[Atom3]-C[Atom2]
  B3x=A[Atom4]-A[Atom3]
  B3y=B[Atom4]-B[Atom3]
  B3z=C[Atom4]-C[Atom3]
  modB2=sqrt((B2x^2)+(B2y^2)+(B2z^2))
# yAx is x-coord. etc of modulus of B2 times B1
  yAx=modB2*(B1x)
  yAy=modB2*(B1y)
  yAz=modB2*(B1z)
# CP2 is the crossproduct of B2 and B3
  CP2x=(B2y*B3z)-(B2z*B3y)
  CP2y=(B2z*B3x)-(B2x*B3z)
  CP2z=(B2x*B3y)-(B2y*B3x)
  termY=((yAx*CP2x)+(yAy*CP2y)+(yAz*CP2z))
# CP is the crossproduct of B1 and B2
  CPx=(B1y*B2z)-(B1z*B2y)
  CPy=(B1z*B2x)-(B1x*B2z)
  CPz=(B1x*B2y)-(B1y*B2x)
  termX=((CPx*CP2x)+(CPy*CP2y)+(CPz*CP2z))
 dihed4=(180/3.141592)*atan2(termY,termX)
 return dihed4
}

function killdyn(isomer) {
  system("rm -f dyn")
}
```

progdyn.conf used for 80-THF product-forming trajectories (using ProgdynONIOM)

```
#This is the configuration file for ProgdynONIOM.  This file is read by progdynONIOM and
# the awk programs proggenHP, prog1stpoint, prog2ndpoint, and progdynb.
#The programs won't read anything past the first blank line,
#and this file must end with a blank line.
#The program has a number of default values but they are unlikely to be what you want.
#Do not delete lines - rather, comment out lines for unwanted options.
#The values here are read repeatedly and most can be changed in the middle of running jobs
#***The keywords are case sensitive.  The following keywords should always be defined:***
#***method, charge, multiplicity, memory, processors, title
#*** method --The following word is copied exactly to the gaussian input file.
method B3LYP/6-31G*
lowmethod PM6-D3H4
#To do a nonstandard route, make nonstandard 1.  For normal calcs, use nonstandard 0 or else leave it out.
#Then make a file called "nonstandard" containing the nonstandard route with no extra lines.
nonstandard 0
# NMRoptions As is NMRtype=1 will add a section for an NMR calc at every NMRevery intervals.  If you want to combine the
two use nonstandard
#NMRtype 1
#NMRmethod2 B97D/6-31G*
#NMRmethod LC-wPBE/6-31G*
#NMRmethod3 B3LYP/cc-pvtz
#NMRevery 4
#NMRrand 1
#NMRcc 1
#loadlimit 10.0
#geometry linear
rotationmode 0
#*** method2 --The options here are restricted, unrestricted, and read. restricted is the default
#If the method is U..., put unrestricted here and the .com files will have in them guess=mix.
#If you put read here, the .com files will contain guess=tcheck, which sometimes makes things faster, sometimes not.
#The use of read requires a specifically defined checkpoint file name using the keyword checkpoint.
method2 read
charge 0
multiplicity 1
```

```
oniomchargemult 0 1
processors 3
#*** memory --The following "word" is copied exactly to the gaussian input file after %mem=.
memory 7gb
#*** killcheck and checkpoint -- You can use a specifically defined checkpoint file name by putting
#the name after the keyword checkpoint.  This is necessary if you use the read option with method2.
#Defined checkpoint names are an unnecessary modest hassle and if you do not want to bother, use killcheck 1
killcheck 0
checkpoint g09.chk
#*** diagnostics -- 0 prints out nothing extra, 1 (default) prints out extra stuff to a
#file "diagnostics", 2 adds more stuff, 3 adds velocities to a file "vellist"
#4 adds the apparent temperature to vellist, but this is meaningless with quasiclassical calculations
diagnostics 4
#*** title -- the title keyword must be followed by exactly four words
title hydroboration dodeceneVTS80THF ONIOM traj
#*** initialdis -- 0 (default) turns off displacement of the normal modes, so that all trajectories start from the same place
# and only the energies and signs of the motion in the modes are randomized
# 1 gives a flat distribution of displacements where all of the possible values are equally likely
# 2 (recommended) gives a QM-like gaussian distribution of displacements, so that displacements in the middle are more likely
that
# those at the end by 1/e
initialdis 0
#*** timestep -- this is the time between points in the trajectory.  Typical values would be 1E-15 or 0.5E-15 or 0.25E-15
timestep 1E-15
#*** scaling -- this lets you scale the gaussian frequencies by a constant
scaling 1.0
temperature 298.15
#*** thermostat 1 puts in a damping factor so as to bring the classical temperature toward the desired temperature.
#*** use a thermostatmult between 0.95 and 1, typically 0.995, so the damping happens slowly - otherwise there will be
#*** over adjustment in response to random variation
#*** the thermostat is not exact.  The second traj point ignores this, so it only applies to later points handled by progdynb.
thermostat 1
thermostatmult 0.999
#*** method3, method4, method5, and method6 -- These keywords let you add extra lines to the gaussian input file.
#method3 and method4 add lines at the top of the input after the lines defining the method, and
#this is useful to implement things like the iop for mPW1k
#method5 and method6 add lines after the geometry, after a blank line of course
#only a single term with no spaces can be added, one per method line.  Here are some examples to uncomment if needed
#method3 IOp(3/76=0572004280)
method3 scf=(conver=5)
#method3 scrf=(pcm,Solvent=dichloromethane)
#add the line below with big structures to get it to put out the distance matrix and the input orientation
method4 iop(2/9=2000)
#method4 iop(3/124=3)
#method4 scrf=(pcm,solvent=dmso,read)
#method5 radii=bondi
#method6
#*** methodfile -- This keyword lets you add more complicated endings to gaussian input files
#such as a gen basis set.  Put after the keyword the number of lines in a file you create called
#methodfile that contains the test you want to add to the end of the gaussian input
methodfile 0
#*** numimag --This tells the program the number of imaginary frequencies in the starting structure.
#if 0, treats as ground state and direction of all modes is random
#if 1, motion along the reaction coordinate will start out in the direction defined by searchdir
#if 2, only lowest freq will go direction of searchdir and other imag mode will go in random direction
numimag 0
#*** searchdir -- This keyword says what direction to follow the mode associated with the imaginary frequency.
#The choices are "negative" and "positive".  Positive moves in the direction defined in the gaussian frequency calculation
#for the imaginary frequency, while negative moves in the opposite direction.  The correct choice can be made either
#by a careful inspection of the normal modes and standard orientation geometry, or by trial and error.
searchdir positive
#*** classical --  for quassiclassical dynamics, the default, use 0.  for classical dynamics, use 1
```

```
#if there are no normal modes and the velocities are to be generated from scratch, use classical 2
classical 2
#*** DRP, saddlepoint, and maxAtomMove --to run a DRP use 'DRP 1' in the line below, otherwise leave it at 0 or comment it out
#the treatment of starting saddlepoints is not yet implemented so use saddlepoint no
#if DRP shows oscillations then decrease maxAtomMove
#DRP 1
#saddlepoint no
#maxAtomMove 0.01
#*** cannonball -- The program can "fire" a trajectory from a starting position toward a particular target, such as toward
#a ts.  To use this, make a file cannontraj with numAtom lines and three numbers per line that defines the vector
#for firing the trajectory, relative to the starting geometry's standard orientation.  The number following cannonball sets
#the extra energy being put into the structure in kcal/mol
#cannonball 10
#*** keepevery --This tells the program how often to write the gaussian output file to file dyn, after the first two points.
#Use 1 for most dynamics to start with, but use a higher number to save on disk space or molden loading time.
keepevery 999999
#*** highlevel --For ONIOM jobs, the following line states the number of highlevel atoms,
#which must come before the medium level atoms.  Use some high value such as 999 if not using ONIOM
highlevel 40
#*** fixedatom1, fixedatom2, fixedatom3, and fixedatom4 - These fix atoms in space.
#Fixing one atom serves no useful purpose and messes things up, while fixing two atoms
#fixes one distance and fixing three has the effect of fixing three distances, not just two
#in current form fixed atoms only are meant to work with no displacements, that is, initialdis=0
#fixedatom1 16
#fixedatom2 1
#fixedatom3 4
#fixedatom4 20
#applyforce 1 lets one push atoms together or apart - a positive force pushes them together
#format is applyforce force    - with the units on force the same as in the Gaussian output file
#applyforce 2 or 3 or 4 applies a polynomial force centered at dist0.  2 is just harmonic, 3 is second order, 4 is third order
#format is applyforce 4 forcecoefficient dist0 forcecoefficient2 forcecoefficient3
#then use afatoms to chose the atoms with format afatoms firstatom secondatom [additional atoms]
#applyforce 2 0.1 3.4
#afatoms 1 2
#applyforceB 2 0.1 3.8
#afatomsB 1 3
#applyforceC 2 0.01 5.2
#afatomsC 8 15
#zeroatom pushes the numbered atom toward the origin with a small harmonic potential  - good with boxon when you want to keep the reaction in the center
#zeroatom 3
#*** boxon and boxsize - With boxon 1, a cubic box is set such that atoms that reach the edge
#are reflected back toward the middle.  Useful for dynamics with solvent molecules.  This is a crude
#implementation that is ok for a few thousand femtoseconds but will not conserve energy long term.
#Set the box size so as to fit the entire initial molecule but not have too much extra room.
#The dimensions of the box are two times the boxsize, e.g. boxsize 7.5 leads to a box that is 15 x 15 x 15 angstroms
#boxon 1
#boxsize 11.2
#*** sphereon and spheresize and sphereforce - uses a force to push atoms within a sphere.  notice that if the atom is far outside of
#the sphere then the force is large unless sphereforce is set small
sphereon 1
spheresize 14.5
sphereforce .1
#setting a value for empiricaldispersion sets its s6 value with the Grimme 2006 algorithm.  Default is 0, with no empiricaldispersion
#setting a radiusmultiplier sets the size of atoms relative to the standard in 2006.
empiricaldispersion .7
radiusmultiplier 1.6
#*** displacements -- This keyword lets you set the initialdis of particular modes by using a series of lines of the format
# displacements NumberOfMode InitialDisForThatMode, as in the example below. You should be able to do as many of these as
```

```
you like
# you might consider this for rotations where a straight-line displacement goes wrong at large displacements
# The choices for InitialDisForThatMode are 0, 1, 2, and 10, where 10 does the same thing as 0 but is maintained for now
because
# a previous version of the program had a bug that made 0 not work.
#displacements 2 0
#displacements 3 0
#displacements 4 0
#displacements 5 0
#displacements 6 0
#displacements 7 0
#displacements 8 0
#displacements 9 0
#displacements 10 0
#*** etolerance --This sets the allowable difference between the desired energy in a trajectory and the actual
#energy, known after point 1 from the potential energy + the kinetic energy in the initial velocities.
#The unit is kcal/mol and 1 is a normal value for mid-sized organic systems.  For very large and floppy molecules, a larger value
#may be needed, but the value must stay way below the average thermal energy in the molecule (not counting zpe).
#If initialdis is not 0 and few trajectories are being rejected, decrease the value.
etolerance 9999999
#*** controlphase --It is sometimes useful to set the phase of particular modes in the initialization of trajectories.
#The format is controlphase numberOfModeToControl positive or controlphase numberOfModeToControl negative.
#controlphase 2 positive
#*** damping -- The damping keyword lets you add or subtract energy from the system at each point, by multiplying the
velocities
#by the damping factor.  A damping of 1 has no effect, and since you mostly want to change the energy slowly, normal values
range
#from 0.95 to 1.05.  The use of damping lets one do simulated annealing - you add energy until the structure is moving enough
#to sample the kinds of possibilities you are interested in, then you take away the energy slowly.
damping 1.000
#at a damping of .9995, the energy is cut in half in 693 points
#*** reversetraj --This keyword sets the trajectories so that both directions from a transition state are explored.
reversetraj true
```

progdyn.conf used for 80-THF feeder trajectories (using ProgdynONIOM)

```
#This is the configuration file for ProgdynONIOM.  This file is read by progdynONIOM and
# the awk programs proggenHP, prog1stpoint, prog2ndpoint, and progdynb.
#The programs won't read anything past the first blank line,
#and this file must end with a blank line.
#The program has a number of default values but they are unlikely to be what you want.
#Do not delete lines - rather, comment out lines for unwanted options.
#The values here are read repeatedly and most can be changed in the middle of running jobs
#***The keywords are case sensitive.  The following keywords should always be defined:***
#***method, charge, multiplicity, memory, processors, title
#*** method --The following word is copied exactly to the gaussian input file.
method B3LYP/6-31G*
lowmethod PM6-D3H4
#To do a nonstandard route, make nonstandard 1.  For normal calcs, use nonstandard 0 or else leave it out.
#Then make a file called "nonstandard" containing the nonstandard route with no extra lines.
nonstandard 0
# NMRoptions As is NMRtype=1 will add a section for an NMR calc at every NMRevery intervals.  If you want to combine the
two use nonstandard
#NMRtype 1
#NMRmethod2 B97D/6-31G*
#NMRmethod LC-wPBE/6-31G*
#NMRmethod3 B3LYP/cc-pvtz
#NMRevery 4
#NMRrand 1
#NMRcc 1
#loadlimit 10.0
```

```
#geometry linear
rotationmode 0
#*** method2 --The options here are restricted, unrestricted, and read. restricted is the default
#If the method is U..., put unrestricted here and the .com files will have in them guess=mix.
#If you put read here, the .com files will contain guess=tcheck, which sometimes makes things faster, sometimes not.
#The use of read requires a specifically defined checkpoint file name using the keyword checkpoint.
method2 read
charge 0
multiplicity 1
oniomchargemult 0 1
processors 3
#*** memory --The following "word" is copied exactly to the gaussian input file after %mem=.
memory 7gb
#*** killcheck and checkpoint -- You can use a specifically defined checkpoint file name by putting
#the name after the keyword checkpoint.  This is necessary if you use the read option with method2.
#Defined checkpoint names are an unnecessary modest hassle and if you do not want to bother, use killcheck 1
killcheck 0
checkpoint g09.chk
#*** diagnostics -- 0 prints out nothing extra, 1 (default) prints out extra stuff to a
#file "diagnostics", 2 adds more stuff, 3 adds velocities to a file "vellist"
#4 adds the apparent temperature to vellist, but this is meaningless with quasiclassical calculations
diagnostics 4
#*** title -- the title keyword must be followed by exactly four words
title hydroboration dodeceneVTS80THF ONIOM traj
#*** initialdis -- 0 (default) turns off displacement of the normal modes, so that all trajectories start from the same place
# and only the energies and signs of the motion in the modes are randomized
# 1 gives a flat distribution of displacements where all of the possible values are equally likely
# 2 (recommended) gives a QM-like gaussian distribution of displacements, so that displacements in the middle are more likely
that
# those at the end by 1/e
initialdis 0
#*** timestep -- this is the time between points in the trajectory.  Typical values would be 1E-15 or 0.5E-15 or 0.25E-15
timestep 1E-15
#*** scaling -- this lets you scale the gaussian frequencies by a constant
scaling 1.0
temperature 298.15
#*** thermostat 1 puts in a damping factor so as to bring the classical temperature toward the desired temperature.
#*** use a thermostatmult between 0.95 and 1, typically 0.995, so the damping happens slowly - otherwise there will be
#*** over adjustment in response to random variation
#*** the thermostat is not exact.  The second traj point ignores this, so it only applies to later points handled by progdynb.
thermostat 1
thermostatmult 0.999
#*** method3, method4, method5, and method6 -- These keywords let you add extra lines to the gaussian input file.
#method3 and method4 add lines at the top of the input after the lines defining the method, and
#this is useful to implement things like the iop for mPW1k
#method5 and method6 add lines after the geometry, after a blank line of course
#only a single term with no spaces can be added, one per method line.  Here are some examples to uncomment if needed
#method3 IOp(3/76=0572004280)
method3 scf=(conver=5)
#method3 scrf=(pcm,Solvent=dichloromethane)
#add the line below with big structures to get it to put out the distance matrix and the input orientation
method4 iop(2/9=2000)
#method4 iop(3/124=3)
#method4 scrf=(pcm,solvent=dmso,read)
#method5 radii=bondi
#method6
#*** methodfile -- This keyword lets you add more complicated endings to gaussian input files
#such as a gen basis set.  Put after the keyword the number of lines in a file you create called
#methodfile that contains the test you want to add to the end of the gaussian input
methodfile 0
#*** numimag --This tells the program the number of imaginary frequencies in the starting structure.
#if 0, treats as ground state and direction of all modes is random
```

```
#if 1, motion along the reaction coordinate will start out in the direction defined by searchdir
#if 2, only lowest freq will go direction of searchdir and other imag mode will go in random direction
numimag 0
#*** searchdir -- This keyword says what direction to follow the mode associated with the imaginary frequency.
#The choices are "negative" and "positive".  Positive moves in the direction defined in the gaussian frequency calculation
#for the imaginary frequency, while negative moves in the opposite direction.  The correct choice can be made either
#by a careful inspection of the normal modes and standard orientation geometry, or by trial and error.
searchdir positive
#*** classical --  for quassiclassical dynamics, the default, use 0.  for classical dynamics, use 1
#if there are no normal modes and the velocities are to be generated from scratch, use classical 2
classical 2
#*** DRP, saddlepoint, and maxAtomMove --to run a DRP use 'DRP 1' in the line below, otherwise leave it at 0 or comment it out
#the treatment of starting saddlepoints is not yet implemented so use saddlepoint no
#if DRP shows oscillations then decrease maxAtomMove
#DRP 1
#saddlepoint no
#maxAtomMove 0.01
#*** cannonball -- The program can "fire" a trajectory from a starting position toward a particular target, such as toward
#a ts.  To use this, make a file cannontraj with numAtom lines and three numbers per line that defines the vector
#for firing the trajectory, relative to the starting geometry's standard orientation.  The number following cannonball sets
#the extra energy being put into the structure in kcal/mol
#cannonball 10
#*** keepevery --This tells the program how often to write the gaussian output file to file dyn, after the first two points.
#Use 1 for most dynamics to start with, but use a higher number to save on disk space or molden loading time.
keepevery 999999
#*** highlevel --For ONIOM jobs, the following line states the number of highlevel atoms,
#which must come before the medium level atoms.  Use some high value such as 999 if not using ONIOM
highlevel 40
#*** fixedatom1, fixedatom2, fixedatom3, and fixedatom4 - These fix atoms in space.
#Fixing one atom serves no useful purpose and messes things up, while fixing two atoms
#fixes one distance and fixing three has the effect of fixing three distances, not just two
#in current form fixed atoms only are meant to work with no displacements, that is, initialdis=0
#fixedatom1 16
#fixedatom2 1
#fixedatom3 4
#fixedatom4 20
#applyforce 1 lets one push atoms together or apart - a positive force pushes them together
#format is applyforce force    - with the units on force the same as in the Gaussian output file
#applyforce 2 or 3 or 4 applies a polynomial force centered at dist0.  2 is just harmonic, 3 is second order, 4 is third order
#format is applyforce 4 forcecoefficient dist0 forcecoefficient2 forcecoefficient3
#then use afatoms to chose the atoms with format afatoms firstatom secondatom [additional atoms]
applyforce 2 0.1 3.4
afatoms 1 2
applyforceB 2 0.1 3.8
afatomsB 1 3
#applyforceC 2 0.01 5.2
#afatomsC 8 15
#zeroatom pushes the numbered atom toward the origin with a small harmonic potential  - good with boxon when you want to keep the reaction in the center
zeroatom 6
#*** boxon and boxsize - With boxon 1, a cubic box is set such that atoms that reach the edge
#are reflected back toward the middle.  Useful for dynamics with solvent molecules.  This is a crude
#implementation that is ok for a few thousand femtoseconds but will not conserve energy long term.
#Set the box size so as to fit the entire initial molecule but not have too much extra room.
#The dimensions of the box are two times the boxsize, e.g. boxsize 7.5 leads to a box that is 15 x 15 x 15 angstroms
#boxon 1
#boxsize 11.2
#*** sphereon and spheresize and sphereforce - uses a force to push atoms within a sphere.  notice that if the atom is far outside of
#the sphere then the force is large unless sphereforce is set small
sphereon 1
```

```
spheresize 15
sphereforce .1
#setting a value for empiricaldispersion sets its s6 value with the Grimme 2006 algorithm.  Default is 0, with no
empiricaldispersion
#setting a radiusmultiplier sets the size of atoms relative to the standard in 2006.
empiricaldispersion .7
radiusmultiplier 1.6
#*** displacements -- This keyword lets you set the initialdis of particular modes by using a series of lines of the format
# displacements NumberOfMode InitialDisForThatMode, as in the example below. You should be able to do as many of these as
you like
# you might consider this for rotations where a straight-line displacement goes wrong at large displacements
# The choices for InitialDisForThatMode are 0, 1, 2, and 10, where 10 does the same thing as 0 but is maintained for now
because
# a previous version of the program had a bug that made 0 not work.
#displacements 2 0
#displacements 3 0
#displacements 4 0
#displacements 5 0
#displacements 6 0
#displacements 7 0
#displacements 8 0
#displacements 9 0
#displacements 10 0
#*** etolerance --This sets the allowable difference between the desired energy in a trajectory and the actual
#energy, known after point 1 from the potential energy + the kinetic energy in the initial velocities.
#The unit is kcal/mol and 1 is a normal value for mid-sized organic systems.  For very large and floppy molecules, a larger value
#may be needed, but the value must stay way below the average thermal energy in the molecule (not counting zpe).
#If initialdis is not 0 and few trajectories are being rejected, decrease the value.
etolerance 9999999
#*** controlphase --It is sometimes useful to set the phase of particular modes in the initialization of trajectories.
#The format is controlphase numberOfModeToControl positive or controlphase numberOfModeToControl negative.
#controlphase 2 positive
#*** damping -- The damping keyword lets you add or subtract energy from the system at each point, by multiplying the
velocities
#by the damping factor.  A damping of 1 has no effect, and since you mostly want to change the energy slowly, normal values
range
#from 0.95 to 1.05.  The use of damping lets one do simulated annealing - you add energy until the structure is moving enough
#to sample the kinds of possibilities you are interested in, then you take away the energy slowly.
damping 1.000
#at a damping of .9995, the energy is cut in half in 693 points
#*** reversetraj --This keyword sets the trajectories so that both directions from a transition state are explored.
reversetraj false
```

Sample Input Files for Gaussrate

The sample input files below are the ones used for the reaction of **8**-out via the Markovnikov pathway to afford **10M**. (Although it makes no difference in this calculation, the conformation of **10M** formed initially differs depending on whether the intermediate is **8**-out or **8**-anti.) This is a VTST-ISPE calculation using CCSD(T)/aug-cc-pvtz single point energies (see file poly.fu51) along the B3LYP/6-31G* path.

For the purpose of obtaining distances between the transition state and the product for the nonstatistical CCNM calculations, the path in this case was extended to a maximum SLP of 8.31 (reaching the product). For comparison, the SLP along the corresponding anti-Markovnikov path was 14.51 (reaching the product).

poly.fu5

```
*General
```

```
  TITLE
hydroboration Markovnikov with butene A
END

 ATOMS
   1  C
   2  C
   3  C
   4  B
   5  C
   6  H
   7  H
   8  H
   9  H
  10  H
  11  H
  12  H
  13  H
  14  H
  15  H
  16  H
 END

 NOSUPERMOL
 DL ispe
 WRITEFU31 ON
 INPUNIT AU

*OPTIMIZATION

 OPTMIN  OHOOK
 OPTTS   OHOOK

*SECOND

 HESSCAL HHOOK

 *REACT1
 SPECIES   NONLINRP
 INITGEO HOOKS
 GEOM
  1
  2
  3
  4
  5
  6
  7
  8
  9
 10
 11
 12
 13
  14
  15
  16
 END

 *PROD1
 INITGEO HOOKS
 SPECIES   NONLINRP
```

```
  GEOM
   1
   2
   3
   4
   5
   6
   7
   8
   9
  10
  11
  12
  13
   14
   15
   16
  END

 *START
  SPECIES NONLINTS
  INITGEO HOOKS
  GEOM
   1
   2
   3
   4
   5
   6
   7
   8
   9
  10
  11
  12
  13
   14
   15
   16
  END

# end of start section

*PATH

 SCALEMASS  1.00

 RODS ON

 INTMU   3
 SSTEP   0.01
 INH     10

 SRANGE

   SLP   2.01
   SLM  -2.01
 END

 RPM       pagem

 SIGN    PRODUCT
 IDIRECT   1

 COORD  CART

# INTDEF
# 5-6 18-19 5-6-1 9-12-18
# 2-1 3-2 4-3 5-4 6-1 7-3 8-7 9-8 10-9 11-7 12-9 13-4
# 14-5 15-6 16-1 17-2 18-11 19-8 20-8 3-2-1 4-3-2 5-4-3
# 6-1-2 7-3-2 8-7-3 9-8-7 10-9-8 11-7-3 12-9-8 13-4-3
# 14-5-4 15-6-1 16-1-2 17-2-1 18-11-7 19-8-7 20-8-7
# 4-3-2-1 5-4-3-2 6-1-2-3 7-3-2-1 8-7-3-2 9-8-7-3
# 10-9-8-7 11-7-3-2 12-9-8-7 13-4-3-2 14-5-4-3 15-6-1-2
# 16-1-2-3 17-2-1-6 18-11-7-3 19-8-7-3 20-8-7-3
# END

 FREQSCALE 1.00

 PRPATH
   COORD 8 9
   INTERVAL 1
   XMOL
 END

*TUNNEL

QUAD
 NQE   40
 NQTH  40
END

  SCT

*RATE

  FORWARDK

  SIGMAF        1
  CVT
  PRDELG ON
# PRGIGT ON

  TEMP
    195.15
    273.15
    298.15
    350
  END
```

esp.fu70

```
*GRGENERAL

  GRRESTART
  RSTTOL 0.00001

*GRSTART

  CHARGE       0
  MULTIPLICITY  1
```

```
*GRREACT1

  CHARGE      0
  MULTIPLICITY  1

*GRPROD1

  CHARGE      0
  MULTIPLICITY  1

*GRCOMMON

  GRENER
      #p B3LYP/6-31G* FCHK NOSYMM UNITS=AU
      scf=tight
      int(grid=ultrafine)
  END

  GRFIRST
      #p B3LYP/6-31G* FORCE FCHK NOSYMM UNITS=AU
      scf=tight
      int(grid=ultrafine)
  END

  GRSEC
      #p B3LYP/6-31G* FREQ=NORAMAN FCHK NOSYMM UNITS=AU
      scf=tight
      int(grid=ultrafine)
  END

  GRLINK0
     %chk=g09.chk
     %nproc=20
     %mem=40gb
  END
```

esp.fu71

```
%nproc=20
%mem=40gb
%chk=g09.chk
#p B3LYP/6-31G* opt fchk NOSYMM
scf=tight int(grid=ultrafine)

borane complex with butene A

0 1
 C,0,-0.4238755288,0.3130760765,-1.459316222
 C,0,-0.3048244057,0.3918311965,0.0436234579
 C,0,0.9068370849,0.3809101003,0.7162495166
 B,0,0.1796359643,-1.2762947905,0.8637416931
 C,0,-1.748734541,-0.2517890738,-1.9787513104
 H,0,-0.2984347216,1.3424485951,-1.8290135165
 H,0,-1.1844152206,0.7498958752,0.575588969
 H,0,1.8334837559,0.2928518951,0.1582384213
 H,0,0.9884248208,0.807981692,1.7088719915
 H,0,0.9297413628,-1.9589981541,0.2218029215
 H,0,0.0665391139,-1.3914891293,2.0524787378
 H,0,-0.9407963501,-1.4158010587,0.3993799205
 H,0,0.4226671509,-0.2603793734,-1.8550976714
 H,0,-1.7879536161,-0.1973237558,-3.0717245739
 H,0,-1.8727117131,-1.298651908,-1.6851013448
 H,0,-2.6029621563,0.3129348129,-1.5861999899
```

esp.fu73

```
%nproc=20
%mem=40gb
%chk=g09.chk
#p B3LYP/6-31G* opt fchk NOSYMM
scf=tight int(grid=ultrafine)

product from m with butene A

0 1
 C,0,-0.2310274656,0.1465969499,-1.3961442695
 C,0,-0.3264349594,0.0966749406,0.1390299704
 C,0,1.0053339503,0.5745667248,0.7962104601
 B,0,-0.5429507281,-1.3236843524,0.7609215402
 C,0,-1.5421170727,-0.2325352442,-2.0935844196
 H,0,0.0696740984,1.1557422968,-1.7152039991
 H,0,-1.1088866491,0.795341485,0.4721132253
 H,0,1.2517543942,1.5841341019,0.4443905146
 H,0,0.9400728723,0.6106834632,1.8896205605
 H,0,-0.1244475808,-2.3040027392,0.2090262816
 H,0,1.8464541097,-0.0773304152,0.5290750932
 H,0,-1.0412702509,-1.4544350061,1.8432160056
 H,0,0.567635157,-0.5313909663,-1.7292101333
 H,0,-1.4482131825,-0.1736669269,-3.1839397134
 H,0,-1.8444093497,-1.2577470149,-1.8433832002
 H,0,-2.3587583434,0.4359997032,-1.7934079165
```

esp.fu75

```
%nproc=20
%mem=40gb
%chk=g09.chk
#p B3LYP/6-31G* opt=(ts,calcfc,noeigentest) fchk NOSYMM
int(grid=ultrafine)

ts for M with butene A

0 1
 C,0,-0.3524889635,0.3614869034,-1.4798311054
 C,0,-0.3528953454,0.3744478104,0.0397807696
 C,0,0.8354805657,0.4121303924,0.79191366
 B,0,-0.0845832688,-1.1706060711,0.8143605506
 C,0,-1.5996827297,-0.2893281835,-2.0854697222
 H,0,-0.2803105874,1.4006302049,-1.8355918196
 H,0,-1.2275564573,0.8330317706,0.4961195438
 H,0,1.7994824255,0.3278793954,0.2967140306
 H,0,0.8607941696,0.8688597126,1.7765256443
 H,0,0.1225614025,-1.9579589028,-0.063202534
 H,0,0.9826846208,-1.1631093501,1.4524099676
 H,0,-0.9297891411,-1.3121644732,1.6493474871
 H,0,0.5477460092,-0.1536653873,-1.8375088923
 H,0,-1.5756208876,-0.239576975,-3.1797917213
 H,0,-1.672868262,-1.3407773446,-1.7898538477
 H,0,-2.5124115506,0.2178614981,-1.7493870111
```

poly.fu51

```
*ISPEGEN          # unit fu51 input using CCSD(T) energies on B3 single points

 MEPTYPER  TWO    # reactant
 MEPTYPEP  TWO    # product

 ENESAD     2.1512
 ENERXN     -18.7710

#+/- 0.1
*POINT
  SMEP -0.6
  VMEP 1.7834
*POINT
  SMEP -0.5
  VMEP 1.9272
*POINT
  SMEP -0.4
  VMEP 2.0553
*POINT
  SMEP -0.3
  VMEP 2.1559
*POINT
  SMEP -0.2
  VMEP 2.2154
*POINT
  SMEP -0.1
  VMEP 2.2193
*POINT
  SMEP 0.1
  VMEP 1.9966
*POINT
  SMEP 0.2
  VMEP 1.7380
*POINT
  SMEP 0.3
  VMEP 1.3613
*POINT
  SMEP 0.4
  VMEP 0.8545
*POINT
  SMEP 0.5
  VMEP 0.2104
*POINT
  SMEP 0.6
  VMEP -0.5721
```

Sample Input Files for Mesmer

Dodecene, Anti conformation, CCSD(T)/aug-cc-pvtz, full molecule

```
<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet type='text/xsl' href='../../mesmer2.xsl' media='other'?>
<?xml-stylesheet type='text/xsl' href='../../mesmer1.xsl' media='screen'?>
<me:mesmer xmlns="http://www.xml-cml.org/schema"
xmlns:me="http://www.chem.leeds.ac.uk/mesmer">
<title>Hydroboration</title>
<moleculeList>
<molecule id="dodecene" description="C12H24">
<propertyList>
<property dictRef="me:ZPE">
```

```
<scalar units="kJ/mol">0.0</scalar>
</property>
<property dictRef="me:rotConsts">
<array units="cm-1">0.219122789      0.005113113 0.005075742</array>
</property>
<property dictRef="me:symmetryNumber">
<scalar>1</scalar>
</property>
<property dictRef="me:frequenciesScaleFactor">
<scalar>0.9854</scalar>
</property>
<property dictRef="me:vibFreqs">
<array units="cm-1">
23.5041 35.7412 44.8815 59.7786 83.8432 94.0883 110.0615
130.7408 135.7867 162.8996 167.1213 192.6601 204.0764 249.7760
738.0284 738.8503 745.8320 766.4055 805.7903 860.7445 904.5081
923.3394 936.1495 940.0673 986.4711 994.4303 1012.5666 1027.5267
1035.0604 1045.7853 1052.0310 1066.6703 1071.4391 1073.0296 1075.8668
1097.5664 1149.5793 1209.0595 1227.7961 1239.6764 1258.6194 1275.7805
1291.5500 1312.1138 1320.1438 1333.0651 1339.0511 1342.5087 1346.1977
1351.8318 1354.2915 1354.6932 1381.9085 1405.7940 1420.6191 1425.8292
1426.8270 1441.3214 1475.6296 1509.6615 1512.8373 1513.4917 1516.3006
1517.0354 1522.3622 1529.2167 1529.8861 1535.9827 1541.5302 1545.1655
1733.1039 3008.0738 3009.0898 3009.8857 3011.9297 3013.0869 3016.4373
3022.0005 3026.5350 3028.6701 3029.8640 3031.5224 3035.6815 3040.6336
3042.8948 3051.1363 3057.1975 3065.3001 3073.5066 3079.8468 3105.1982
3109.8409 3133.2683 3154.1959 3232.0859
</array>
</property>
<property dictRef="me:MW">
<scalar units="amu">168.319</scalar>
</property>
<property dictRef="me:spinMultiplicity">
<scalar>1</scalar>
</property>
</propertyList>
<me:DOSCMethod>Classical rotors</me:DOSCMethod>
</molecule>

<molecule id="bh3" description="bh3">
<propertyList>
<property dictRef="me:ZPE">
<scalar units="kJ/mol">0.0</scalar>
</property>
<property dictRef="me:rotConsts">
<array units="cm-1">7.824596263      7.824596263 3.912297965</array>
</property>
<property dictRef="me:symmetryNumber">
<scalar>6</scalar>
</property>
<property dictRef="me:frequenciesScaleFactor">
<scalar>0.9854</scalar>
</property>
<property dictRef="me:vibFreqs">
<array units="cm-1">
1160.2127 1213.4739 1213.4744 2593.8604 2726.1153 2726.1166
</array>
S9
</property>
<property dictRef="me:MW">
<scalar units="amu">13.83</scalar>
</property>
```

```
<property dictRef="me:spinMultiplicity">
<scalar>1</scalar>
</property>
</propertyList>
<me:DOSCMethod>ClassicalRotors</me:DOSCMethod>
</molecule>

<molecule id="complex3" description="complex3">
<propertyList>
<property dictRef="me:ZPE">
<scalar units="kJ/mol">-42.457</scalar>
</property>
<property dictRef="me:rotConsts">
<array units="cm-1">0.168538205      0.004247915 0.004210544</array>
</property>
<property dictRef="me:symmetryNumber">
<scalar>1</scalar>
</property>
<property dictRef="me:frequenciesScaleFactor">
<scalar>0.9854</scalar>
</property>
<property dictRef="me:epsilon">
<scalar>272.01</scalar>
</property>
<property dictRef="me:sigma">
<scalar>6.7697</scalar>
</property>
<property dictRef="me:vibFreqs">
<array units="cm-1">
21.0698 31.7876 38.1089 52.1561 76.7525 85.6690 91.8916
119.5449 127.0780 134.0575 162.7833 165.2803 170.8137 191.0036
249.9306 256.0158 280.3986 323.2575 356.8956 412.5383 424.9810
484.0934 500.6757 567.0413 708.1558 737.9541 739.1944 746.6405
767.3003 805.2292 817.6024 857.0909 868.6660 904.8682 924.7710
941.9684 984.5293 991.7264 1008.4904 1014.1664 1024.3885 1038.0670
1049.7925 1063.5014 1066.9047 1073.1518 1073.8921 1081.9925 1096.4566
1116.5043 1149.5331 1171.4746 1183.3677 1213.6796 1229.6065 1240.3765
1259.6838 1276.1546 1290.5863 1303.7212 1315.6165 1324.0562 1338.2075
1342.3447 1346.7538 1352.5892 1355.2723 1355.5623 1382.7208 1405.7826
1420.6822 1426.3776 1427.2211 1441.6392 1471.8852 1512.7692 1513.0042
1515.6433 1516.5391 1518.6102 1523.6915 1529.9741 1530.3247 1536.7517
1541.9020 1545.2562 1609.9873 2481.0479 2557.0594 2643.7082 3009.7424
3010.3695 3013.1656 3014.6174 3018.6578 3023.2525 3027.6480 3030.3082
3030.4653 3032.8847 3037.6916 3040.9590 3045.2447 3053.7540 3057.0048
3062.9856 3071.3403 3077.9371 3102.3547 3105.4282 3110.4577 3179.8993
3182.9846 3268.3122
</array>
</property>
<property dictRef="me:MW">
<scalar units="amu">182.149</scalar>
</property>
<property dictRef="me:spinMultiplicity">
<scalar>1</scalar>
</property>
<property dictRef="me:deltaEDown">
<!--deltaEDown is the average energy transferred upon collision in the
downward direction-->
<scalar>860.1234</scalar>
</property>
</propertyList>
<me:DOSCMethod>ClassicalRotors</me:DOSCMethod>
</molecule>
```

<molecule id="AMTS" description="AMTS">
<propertyList>
<property dictRef="me:ZPE">
<scalar units="kJ/mol">-42.688</scalar>
</property>
<property dictRef="me:rotConsts">
<array units="cm-1">0.165355355   0.004271939 0.004234902</array>
</property>
<property dictRef="me:symmetryNumber">
<scalar>1</scalar>
</property>
S10
<property dictRef="me:frequenciesScaleFactor">
<scalar>0.9854</scalar>
</property>
<property dictRef="me:vibFreqs">
<array units="cm-1">
21.4906 32.8223 39.6208 54.5767 81.5338 92.2038 95.1861
126.6999 136.0528 161.1041 163.1062 166.6379 180.2035 233.4436
250.0555 284.7596 346.3512 370.0445 427.6035 444.9497 490.9093
506.3572 635.4032 716.6013 737.9775 739.2514 746.8062 767.5918
803.6578 807.9061 861.0684 896.3989 904.8135 924.1012 939.3775
984.0782 991.8206 1011.2221 1016.3118 1025.3052 1039.5081 1050.0437
1065.0597 1066.7031 1073.0394 1074.1097 1081.8647 1093.0336 1138.6141
1149.9751 1170.0054 1188.7612 1215.3873 1229.6503 1241.0175 1259.4356
1273.5719 1286.1794 1294.5836 1315.3554 1323.1918 1338.1038 1342.3541
1346.6361 1353.0707 1355.1839 1356.1410 1384.1393 1406.9039 1420.9807
1426.2856 1427.1506 1441.5813 1458.9472 1512.7229 1512.9883 1515.5362
1516.5438 1518.4581 1523.5808 1530.0062 1530.2557 1536.7244 1541.8978
1545.2431 1568.5568 2386.9414 2558.4074 2658.4231 3009.7103 3010.3856
3013.1135 3014.5629 3018.5750 3023.2950 3027.6370 3030.1266 3030.4923
3032.9101 3037.7905 3040.9335 3045.3799 3052.6351 3054.6730 3063.3371
3071.4051 3077.8029 3098.7466 3105.4274 3110.4243 3169.9522 3176.1247
3255.2348
</array>
</property>
<property dictRef="me:MW">
<scalar units="amu">182.149</scalar>
</property>
<property dictRef="me:spinMultiplicity">
<scalar>1</scalar>
</property>
<property dictRef="me:imFreqs">
<scalar units="cm-1">337.2266</scalar>
</property>
</propertyList>
<me:DOSCMethod>ClassicalRotors</me:DOSCMethod>
</molecule>

<molecule id="MTS" description="MTS">
<propertyList>
<property dictRef="me:ZPE">
<scalar units="kJ/mol">-33.644</scalar>
</property>
<property dictRef="me:rotConsts">
<array units="cm-1">
0.171014681   0.004253253   0.00421655
</array>
</property>
<property dictRef="me:symmetryNumber">
<scalar>1</scalar>

```
</property>
<property dictRef="me:frequenciesScaleFactor">
<scalar>0.9854</scalar>
</property>
<property dictRef="me:vibFreqs">
<array units="cm-1">
21.3819 33.6482 38.1231 54.8961 83.0534 91.2539 97.5590
128.6128 137.3051 164.0969 165.3259 171.1004 180.5804 250.3015
255.7835 283.1959 327.1411 359.7441 429.2812 434.3232 490.0040
505.1515 632.5675 737.9585 738.3783 743.2565 747.9672 765.8439
799.0249 805.5537 858.1430 904.5032 912.7519 928.6027 963.4384
990.0020 998.5021 1012.6040 1028.6545 1035.9554 1051.3515 1057.1956
1060.9848 1067.8084 1073.0420 1074.2663 1088.9764 1101.6510 1136.7832
1150.0787 1170.4541 1196.4960 1217.9783 1230.5519 1240.6890 1259.8527
1275.1692 1287.9725 1295.5357 1315.0062 1323.3623 1337.7705 1342.3659
1346.0830 1352.7005 1354.8413 1355.5385 1383.7697 1406.9828 1421.2708
1426.0841 1427.0787 1441.5305 1458.8487 1511.1602 1512.8258 1513.8121
1516.2442 1517.3534 1522.6668 1529.4046 1530.0401 1536.0483 1541.5307
1545.1699 1554.1247 2331.9922 2573.5304 2676.5607 3009.2819 3009.6339
3011.9618 3013.0588 3015.2519 3019.8539 3025.2349 3028.5043 3030.0727
3031.5454 3035.0312 3040.7364 3040.8870 3042.8409 3049.4248 3058.9350
3067.8747 3075.5445 3085.2311 3105.2551 3110.0758 3161.7617 3170.1155
3247.7746
</array>
</property>
<property dictRef="me:MW">
<scalar units="amu">182.149</scalar>
</property>
S11
<property dictRef="me:spinMultiplicity">
<scalar>1</scalar>
</property>
<property dictRef="me:imFreqs">
<scalar units="cm-1">525.9578</scalar>
</property>
</propertyList>
<me:DOSCMethod>ClassicalRotors</me:DOSCMethod>
</molecule>

<molecule id="THF">
<!--THF collision parameters used in this work; however  this choice is
    arbitrary  since the results are
    reported in terms of collision frequencies-->
<propertyList>
<property dictRef="me:epsilon">
<scalar>68.1</scalar>
</property>
<property dictRef="me:sigma">
<scalar>3.57</scalar>
</property>
<property dictRef="me:MW">
<scalar units="amu">72.11</scalar>
</property>
</propertyList>
</molecule>

<molecule id="antimarkov">
<propertyList>
<property dictRef="me:ZPE">
<scalar units="kJ/mol">-60.0</scalar>
</property>
</propertyList>
```

```
</molecule>

<molecule id="markov">
<propertyList>
<property dictRef="me:ZPE">
<scalar units="kJ/mol">-60.0</scalar>
</property>
</propertyList>
</molecule>

</moleculeList>
<reactionList>
<reaction id="R1">
<reactant>
<molecule ref="dodecene" me:type="deficientReactant" />
</reactant>
<reactant>
<molecule ref="bh3" me:type="excessReactant" />
</reactant>
<product>
<molecule ref="complex3" me:type="modelled" />
</product>
<me:MCRCMethod>Mesmer ILT</me:MCRCMethod>
<me:preExponential>1.75e-11</me:preExponential>
<me:excessReactantConc>2.25E16</me:excessReactantConc>
<me:activationEnergy>0.0</me:activationEnergy>
<me:TInfinity>298.0</me:TInfinity>
<me:nInfinity>-0.01</me:nInfinity>
</reaction>
<reaction id="R2">
<reactant>
<molecule ref="complex3" me:type="modelled" />
</reactant>
<product>
<molecule ref="antimarkov" me:type="sink" />
S12
</product>
<me:transitionState>
<molecule ref="AMTS" me:type="transitionState" />
</me:transitionState>
<me:MCRCMethod>SimpleRRKM</me:MCRCMethod>
</reaction>
<reaction id="R3">
<reactant>
<molecule ref="complex3" me:type="modelled" />
</reactant>
<product>
<molecule ref="markov" me:type="sink" />
</product>
<me:transitionState>
<molecule ref="MTS" me:type="transitionState" />
</me:transitionState>
<me:MCRCMethod>SimpleRRKM</me:MCRCMethod>
</reaction>
</reactionList>
<me:conditions>
<me:bathGas>THF</me:bathGas>
<me:PTs>
<!--the pressure/temp pairs given below are to cover the
entire 368 K range of collision frequencies reported in the paper-->
<me:PTpair me:units="Torr" me:P="209395." me:T="298." />
</me:PTs>
```

```
</me:conditions>
<me:modelParameters>
<!--Specify grain size directly...-->
<me:grainSize units="cm-1">25.</me:grainSize>
<!--...or by the total number of grains
<me:numberOfGrains> 500 </me:numberOfGrains>-->
<!--Specify increased energy range
<me:maxTemperature>6000</me:maxTemperature>-->
<me:energyAboveTheTopHill>25.0</me:energyAboveTheTopHill>
</me:modelParameters>
<me:control>
<me:testDOS />
<me:printSpeciesProfile />
S13
<!--<me:testMicroRates />-->
<me:testRateConstants />
<me:printGrainDOS />
<!--<me:printCellDOS />-->
<!--<me:printReactionOperatorColumnSums />-->
<!--<me:printTunnellingCoefficients />-->
<me:printGrainkfE />
<!--<me:printGrainBoltzmann />-->
<me:printGrainkbE />
<me:eigenvalues>0</me:eigenvalues>
</me:control>
</me:mesmer>
```

Dodecene, Anti conformation, CCSD(T)/aug-cc-pvtz, localized calculation

```
<?xml version="1.0" encoding="utf-8" ?>
<?xml-stylesheet type='text/xsl' href='../../mesmer2.xsl' media='other'?>
<?xml-stylesheet type='text/xsl' href='../../mesmer1.xsl' media='screen'?>
<me:mesmer xmlns="http://www.xml-cml.org/schema"
xmlns:me="http://www.chem.leeds.ac.uk/mesmer">
<title>Hydroboration</title>
<moleculeList>
<molecule id="dodecene" description="C12H24">
<propertyList>
<property dictRef="me:ZPE">
<scalar units="kJ/mol">0.0</scalar>
</property>
<property dictRef="me:rotConsts">
<array units="cm-1">0.219122789      0.005113113 0.005075742</array>
</property>
<property dictRef="me:symmetryNumber">
<scalar>1</scalar>
</property>
<property dictRef="me:frequenciesScaleFactor">
<scalar>0.9854</scalar>
</property>
<property dictRef="me:vibFreqs">
<array units="cm-1">
210.9293 425.3639 589.0547 933.3222 937.1024 956.6329 1036.0412
1081.6911 1203.3971 1339.6015 1434.8730 1474.2160 1509.9348 1524.0905
1739.1122 3032.3826 3079.0754 3116.7134 3144.6133 3156.6634
</array>
</property>
<property dictRef="me:MW">
<scalar units="amu">168.319</scalar>
</property>
<property dictRef="me:spinMultiplicity">
```

```
<scalar>1</scalar>
</property>
</propertyList>
<me:DOSCMethod>Classical rotors</me:DOSCMethod>
</molecule>

<molecule id="bh3" description="bh3">
<propertyList>
<property dictRef="me:ZPE">
<scalar units="kJ/mol">0.0</scalar>
</property>
<property dictRef="me:rotConsts">
<array units="cm-1">7.824596263      7.824596263 3.912297965</array>
</property>
<property dictRef="me:symmetryNumber">
<scalar>6</scalar>
</property>
<property dictRef="me:frequenciesScaleFactor">
<scalar>0.9854</scalar>
</property>
<property dictRef="me:vibFreqs">
<array units="cm-1">
1160.2127 1213.4739 1213.4744 2593.8604 2726.1153 2726.1166
</array>
S9
</property>
<property dictRef="me:MW">
<scalar units="amu">13.83</scalar>
</property>
<property dictRef="me:spinMultiplicity">
<scalar>1</scalar>
</property>
</propertyList>
<me:DOSCMethod>ClassicalRotors</me:DOSCMethod>
</molecule>

<molecule id="complex3" description="complex3">
<propertyList>
<property dictRef="me:ZPE">
<scalar units="kJ/mol">-42.457</scalar>
</property>
<property dictRef="me:rotConsts">
<array units="cm-1">0.168538205      0.004247915 0.004210544</array>
</property>
<property dictRef="me:symmetryNumber">
<scalar>1</scalar>
</property>
<property dictRef="me:frequenciesScaleFactor">
<scalar>0.9854</scalar>
</property>
<property dictRef="me:epsilon">
<scalar>272.01</scalar>
</property>
<property dictRef="me:sigma">
<scalar>6.7697</scalar>
</property>
<property dictRef="me:vibFreqs">
<array units="cm-1">
110.3949 193.9645 276.7160 339.2536 401.5664 494.2255 711.3290
802.5044 851.7722 929.6655 961.5245 1009.7585 1047.6309 1097.8983
1116.0805 1172.2780 1183.1837 1211.5323 1314.3351 1432.6004 1472.3195
1508.8553 1522.5427 1617.8505 2484.8461 2558.3458 2644.2066 3050.0175
```

```
3118.7710 3148.4316 3183.4473 3191.2662 3271.7531
</array>
</property>
<property dictRef="me:MW">
<scalar units="amu">182.149</scalar>
</property>
<property dictRef="me:spinMultiplicity">
<scalar>1</scalar>
</property>
<property dictRef="me:deltaEDown">
<!--deltaEDown is the average energy transferred upon collision in the
downward direction-->
<scalar>134.1234</scalar>
</property>
</propertyList>
<me:DOSCMethod>ClassicalRotors</me:DOSCMethod>
</molecule>

<molecule id="AMTS" description="AMTS">
<propertyList>
<property dictRef="me:ZPE">
<scalar units="kJ/mol">-42.688</scalar>
</property>
<property dictRef="me:rotConsts">
<array units="cm-1">0.165355355      0.004271939 0.004234902</array>
</property>
<property dictRef="me:symmetryNumber">
<scalar>1</scalar>
</property>
S10
<property dictRef="me:frequenciesScaleFactor">
<scalar>0.9854</scalar>
</property>
<property dictRef="me:vibFreqs">
<array units="cm-1">
213.4803 233.4630 388.7657 442.0287 576.9672 716.1992 797.2462
885.6810 929.4999 962.4460 1016.0393 1050.9048 1089.0271 1142.7874
1170.7303 1187.0138 1221.4235 1293.5833 1429.3509 1462.5154 1511.4203
1522.4767 1572.7520 2384.8265 2559.0615 2659.8540 3052.6305 3119.4382
3144.5452 3172.4152 3184.8931 3257.8205
</array>
</property>
<property dictRef="me:MW">
<scalar units="amu">182.149</scalar>
</property>
<property dictRef="me:spinMultiplicity">
<scalar>1</scalar>
</property>
<property dictRef="me:imFreqs">
<scalar units="cm-1">337.2266</scalar>
</property>
</propertyList>
<me:DOSCMethod>ClassicalRotors</me:DOSCMethod>
</molecule>

<molecule id="MTS" description="MTS">
<propertyList>
<property dictRef="me:ZPE">
<scalar units="kJ/mol">-33.644</scalar>
</property>
<property dictRef="me:rotConsts">
<array units="cm-1">
```

```
0.171014681    0.004253253    0.00421655
</array>
</property>
<property dictRef="me:symmetryNumber">
<scalar>1</scalar>
</property>
<property dictRef="me:frequenciesScaleFactor">
<scalar>0.9854</scalar>
</property>
<property dictRef="me:vibFreqs">
<array units="cm-1">
224.9188 268.2195 340.2778 420.6411 604.6537 720.4579 798.1757
890.0871 947.1378 987.8065 1058.3066 1069.1978 1103.5775 1128.5230
1171.6823 1199.2287 1225.1173 1296.3394 1434.8862 1461.8585 1514.2697
1529.1506 1560.8426 2329.9120 2574.5222 2677.6771 3041.0599 3105.7604
3129.5834 3163.5039 3179.7944 3249.6061
</array>
</property>
<property dictRef="me:MW">
<scalar units="amu">182.149</scalar>
</property>
S11
<property dictRef="me:spinMultiplicity">
<scalar>1</scalar>
</property>
<property dictRef="me:imFreqs">
<scalar units="cm-1">525.9578</scalar>
</property>
</propertyList>
<me:DOSCMethod>ClassicalRotors</me:DOSCMethod>
</molecule>

<molecule id="THF">
<!--THF collision parameters used in this work; however  this choice is
    arbitrary  since the results are
    reported in terms of collision frequencies-->
<propertyList>
<property dictRef="me:epsilon">
<scalar>68.1</scalar>
</property>
<property dictRef="me:sigma">
<scalar>3.57</scalar>
</property>
<property dictRef="me:MW">
<scalar units="amu">72.11</scalar>
</property>
</propertyList>
</molecule>

<molecule id="antimarkov">
<propertyList>
<property dictRef="me:ZPE">
<scalar units="kJ/mol">-60.0</scalar>
</property>
</propertyList>
</molecule>

<molecule id="markov">
<propertyList>
<property dictRef="me:ZPE">
<scalar units="kJ/mol">-60.0</scalar>
</property>
```

```
</propertyList>
</molecule>

</moleculeList>
<reactionList>
<reaction id="R1">
<reactant>
<molecule ref="dodecene" me:type="deficientReactant" />
</reactant>
<reactant>
<molecule ref="bh3" me:type="excessReactant" />
</reactant>
<product>
<molecule ref="complex3" me:type="modelled" />
</product>
<me:MCRCMethod>Mesmer ILT</me:MCRCMethod>
<me:preExponential>1.75e-11</me:preExponential>
<me:excessReactantConc>2.25E16</me:excessReactantConc>
<me:activationEnergy>0.0</me:activationEnergy>
<me:TInfinity>298.0</me:TInfinity>
<me:nInfinity>-0.01</me:nInfinity>
</reaction>
<reaction id="R2">
<reactant>
<molecule ref="complex3" me:type="modelled" />
</reactant>
<product>
<molecule ref="antimarkov" me:type="sink" />
S12
</product>
<me:transitionState>
<molecule ref="AMTS" me:type="transitionState" />
</me:transitionState>
<me:MCRCMethod>SimpleRRKM</me:MCRCMethod>
</reaction>
<reaction id="R3">
<reactant>
<molecule ref="complex3" me:type="modelled" />
</reactant>
<product>
<molecule ref="markov" me:type="sink" />
</product>
<me:transitionState>
<molecule ref="MTS" me:type="transitionState" />
</me:transitionState>
<me:MCRCMethod>SimpleRRKM</me:MCRCMethod>
</reaction>
</reactionList>
<me:conditions>
<me:bathGas>THF</me:bathGas>
<me:PTs>
<!--the pressure/temp pairs given below are to cover the
entire 368 K range of collision frequencies reported in the paper-->
<me:PTpair me:units="Torr" me:P="20939500." me:T="298." />
</me:PTs>
</me:conditions>
<me:modelParameters>
<!--Specify grain size directly...-->
<me:grainSize units="cm-1">25.</me:grainSize>
<!--...or by the total number of grains
<me:numberOfGrains> 500 </me:numberOfGrains>-->
<!--Specify increased energy range
```

```
<me:maxTemperature>6000</me:maxTemperature>-->
<me:energyAboveTheTopHill>25.0</me:energyAboveTheTopHill>
</me:modelParameters>
<me:control>
<me:testDOS />
<me:printSpeciesProfile />
S13
<!--<me:testMicroRates />-->
<me:testRateConstants />
<me:printGrainDOS />
<!--<me:printCellDOS />-->
<!--<me:printReactionOperatorColumnSums />-->
<!--<me:printTunnellingCoefficients />-->
<me:printGrainkfE />
<!--<me:printGrainBoltzmann />-->
<me:printGrainkbE />
<me:eigenvalues>0</me:eigenvalues>
</me:control>
</me:mesmer>
```

Calculated Structures and Complete Energies

Guide to Structures, Structure Titles and Their Organization

The structure titles below are in some cases self-explanatory, but in other cases require some explanation. Anti-Markovnikov and Markovnikov structure refer to the proximity of the boron atom to the external (anti-Markovnikov) or internal (Markovnikov) alkene carbon. These are labeled AM and M, respectively. The π-complexes (**INT** in the main text) are referred to as "complex" while transition structures are labeled as "TS" respectively. The alkenes that are labeled in plane and out of plane are referring to IUPAC carbon 4 in or out of the plane with respect to the alkene. The structures named "anti", "out", or "in" follow the same format as in the text. For products, the label "twisted" refers to the boron rotating so its hydrogens are in the plane with the carbons that were the alkene and "gauche" refers to the boron rotating gauche to carbon 3 of the chain. The label VTS is the variational transition structure. The structures given are approximate variational transition structures based on the nearest saved point to the calculated variational transition state in the POLYRATE calculation.

Tables of Energies

B3LYP/ 6-31G*	Potential E	E+ZPE	Enthalpy	Free Energy	Relative G	Relative E+ZPE
BH_3	-26.61300009	-26.586498	-26.582669	-26.604053	0	0
PROPENE						
propene	-117.9075586	-117.827483	-117.822462	-117.852487	0	0
INT	-144.53935	-144.426742	-144.419777	-144.45473	1.1	-8.0
TSA	-144.53899	-144.426563	-144.420438	-144.453598	1.8	-7.9
TSM	-144.53533	-144.422776	-144.416681	-144.449756	4.3	-5.5
A	-144.57158	-144.457548	-144.450586	-144.485717	-18.3	-27.3
M	-144.56896	-144.454787	-144.447672	-144.482901	-16.5	-25.6
BUTENE						
Butene Out of Plane	-157.2210724	-157.112059	-157.105863	-157.139545	0	0
Butene In Plane	-157.2203614	-157.111217	-157.105141	-157.138434	0	0

INT out	-183.85311	-183.711843	-183.703689	-183.741921	1.05	-8.3
INT anti	-183.85287	-183.7114	-183.703176	-183.741667	1.21	-8.1
INT in	-183.85149	-183.709993	-183.701764	-183.740292	2.07	-7.2
TSA out	-183.85308	-183.7121	-183.70466	-183.741437	1.36	-8.5
TSA anti	-183.85260	-183.711383	-183.703955	-183.740715	1.81	-8.0
TSA in	-183.85066	-183.709226	-183.701924	-183.73837	3.28	-6.7
TSM out	-183.8493367	-183.708031	-183.700708	-183.737134	4.06	-5.9
TSM anti	-183.8493441	-183.708093	-183.700718	-183.737325	3.94	-6.0
TSM in	-183.8472415	-183.705599	-183.698396	-183.734534	5.69	-4.4
A out	-183.88406	-183.740877	-183.732379	-183.772023	-17.84	-26.6
A out twisted	-183.88576	-183.743336	-183.734882	-183.774265	-19.24	-28.1
A out gauche	-183.88549	-183.742805	-183.73456	-183.773238	-18.60	-27.8
A anti	-183.88284	-183.739608	-183.731106	-183.771304	-17.39	-25.8
A anti twisted	-183.88456	-183.741897	-183.733584	-183.772578	-18.19	-27.2
A anti gauche	-183.88400	-183.741145	-183.732969	-183.771563	-17.55	-26.7
A in	-183.88456	-183.741897	-183.733584	-183.77258	-18.19	-27.2
M out	-183.88271	-183.739811	-183.731463	-183.770028	-16.59	-25.9
M anti	-183.88137	-183.738524	-183.730089	-183.769003	-15.94	-25.1
M in	-183.88212	-183.739522	-183.73131	-183.769566	-16.30	-25.7
HEXENE						
Hexene out of plane	-235.8486977	-235.682402	-235.673596	-235.714493	0	0
A INT out	-262.4805717	-262.282064	-262.27124	-262.316723	1.14	-8.3
A INT anti	-262.5	-262.281821	-262.270938	-262.316644	1.19	-8.1
M INT out	-262.4805728	-262.28205	-262.271231	-262.316698	1.16	-8.3
M INT anti	-262.5	-262.281821	-262.270938	-262.316644	1.19	-8.1
TSA out	-262.4805349	-262.282311	-262.272201	-262.316208	1.47	-8.4
TSA anti	-262.48023	-262.281746	-262.271666	-262.315596	1.85	-8.1
TSM out	-262.4768059	-262.278285	-262.268288	-262.311926	4.15	-5.9
TSM anti	-262.5	-262.278427	-262.268387	-262.312185	3.99	-6.0
A out	-262.5114918	-262.311064	-262.299898	-262.346744	-17.69	-26.5
A out twisted	-262.5131746	-262.313522	-262.302395	-262.349004	-19.11	-28.0
A out gauche	-262.5129355	-262.313008	-262.302101	-262.347996	-18.48	-27.7
A anti	-262.5	-262.309742	-262.298574	-262.346139	-17.31	-25.6
A anti twisted	-262.51195	-262.31206	-262.301083	-262.347342	-18.07	-27.1
M out	-262.5102871	-262.31015	-262.299134	-262.344885	-16.53	-25.9
M anti	-262.5	-262.308661	-262.297546	-262.343725	-15.80	-25.0
OCTENE						
Octene out of plane	-314.4761382	-314.252627	-314.241129	-314.289238	0	0
A INT out	-341.1080296	-340.852332	-340.838793	-340.89153	1.11	-8.3
A INT anti	-341.1	-340.852094	-340.838498	-340.891443	1.16	-8.1
M INT out	-341.1080307	-340.852319	-340.838786	-340.891502	1.12	-8.3
M INT anti	-341.1	-340.852094	-340.838498	-340.891443	1.16	-8.1
TSA out	-341.1079941	-340.852565	-340.839747	-340.89098	1.45	-8.4
TSA anti	-341.1	-340.852036	-340.839239	-340.89043	1.80	-8.1
TSM out	-341.104247	-340.848532	-340.835825	-340.886692	4.14	-5.9
TSM anti	-341.1	-340.848689	-340.835938	-340.88695	-341.1	-6.0
A out	-341.1389307	-340.881308	-340.867424	-340.92152	-17.71	-26.5
A out twisted	-341.1406052	-340.883763	-340.869925	-340.923752	-19.11	-28.0
A out gauche	-341.1403679	-340.88324	-340.869626	-340.922748	-18.48	-27.7
A anti	-341.1	-340.87995	-340.866074	-340.920891	-17.32	-25.6
A anti	-341.1	-340.882298	-340.868617	-340.922102	-18.08	-27.1

twisted						
M out	-341.1377233	-340.880393	-340.86667	-340.919627	-16.53	-25.9
M anti	-341.1	-340.878894	-340.865074	-340.918443	-15.78	-25.0
DODECENE						
Dodecene out of plane	-471.7309986	-471.393138	-471.37619	-471.438744	0	0
A INT out	-498.3629016	-497.992825	-497.973846	-498.040953	1.16	-8.3
A INT anti	-498.4	-497.99261	-497.973558	-498.040919	1.18	-8.1
M INT out	-498.3629015	-497.992834	-497.973851	-498.040971	1.15	-8.3
M INT anti	-498.4	-497.99261	-497.973558	-498.040919	1.18	-8.1
TSA out	-498.362865	-497.993098	-497.974822	-498.04048	1.45	-8.4
TSA anti	-498.4	-497.992532	-497.974292	-498.039845	1.85	-8.1
TSM out	-498.3591071	-497.989064	-497.970895	-498.036223	4.13	-5.9
TSM anti	-498.4	-497.989189	-497.970986	-498.036417	4.00	-6.0
A out	-498.3937911	-498.021826	-498.002482	-498.071035	-17.72	-26.5
A out twisted	-498.3954629	-498.02428	-498.004984	-498.073251	-19.11	-28.0
A out gauche	-498.3952267	-498.023761	-498.00469	-498.072233	-18.47	-27.7
A anti	-498.4	-498.020595	-498.001179	-498.071325	-17.90	-25.7
A anti twisted	-498.4	-498.022782	-498.00366	-498.071449	-17.98	-27.1
M out	-498.3925812	-498.020913	-498.001726	-498.069165	-16.55	-25.9
M anti	-498.4	-498.019412	-498.000132	-498.06793	-15.77	-25.0
3,3-DIMETHYL-2-BUTENE						
3,3-dimethyl-2-butene	-235.8482117	-235.682768	-235.674152	-235.713381	0	0
A INT	-262.47843	-262.280737	-262.270009	-262.313954	2.18	-7.2
M INT	-262.47843	-262.280756	-262.270022	-262.313982	2.17	-7.2
TSA	-262.5	-262.280894	-262.270956	-262.313165	2.68	-7.3
TSM	-262.5	-262.277176	-262.267391	-262.309136	5.21	-5.0
A	-262.5	-262.310411	-262.29943	-262.344773	-262.5	-25.8
A twisted	-262.5114249	-262.312685	-262.301768	-262.346625	-18.32	-27.2
M	-262.50600	-262.307069	-262.296222	-262.340676	-14.58	-23.7
M twisted	-262.50642	-262.307206	-262.296348	-262.34038	-14.40	-23.8

CCSD(T)	aug-cc-pvdz E	aug-cc-pvdz E+ZPE	aug-cc-pvtz E	aug-cc-pvtz E+ZPE	Relative E aug-cc-pvtz	Relative E+ZPE aug-cc-pvtz
BH_3	-26.51255	-26.48605	-26.539	-26.513	0	0
PROPENE						
propene	-117.57779	-117.49772	-117.69021	-117.61013	0	0
INT	-144.11042	-143.99781	-144.250904	-144.13829	-8.8	-9.5
TSA	-144.10975	-143.99732	-144.2506139	-144.13818	-8.5	-9.4
TSM	-144.10975	-143.99321	-144.2506139	-144.13432	-5.9	-7.0
A	-144.14328	-144.02925	-144.282465	-144.16843	-28.5	-28.4
M	-144.14108	-144.02691	-144.280373	-144.16620	-27.1	-27.0
BUTENE						
Butene Out of Plane	-156.78338	-156.67436	-156.93245	-156.82344	0	0
Butene In Plane	-156.78307	-156.67393	-156.93237	-156.82323	0	0
INT out	-183.31640	-183.17513	-183.49366	-183.35240	-9.2	-10.0
INT anti	-183.31639	-183.17492	-183.49343	-183.35196	-9.1	-9.7
TSA out	-183.31632	-183.17534	-183.49372	-183.35274	-9.4	-10.2
TSA anti	-183.31588	-183.17466	-183.49330	-183.35209	-8.9	-9.8
TSM out	-183.31257	-183.17126	-183.49024	-183.34893	-6.8	-7.8

TSM anti	-183.31227	-183.17102	-183.48993	-183.34868	-6.7	-7.7
A out	-183.34764	-183.20445	-183.52365	-183.38046	-27.6	-27.6
A out twisted	-183.34919	-183.20677	-183.52511	-183.38269	-29.1	-29.0
A out gauche	-183.34954	-183.20686	-183.52534	-183.38266	-29.1	-29.0
A anti	-183.34683	-183.20361	-183.52286	-183.37963	-27.1	-27.1
A anti twisted	-183.34839	-183.20573	-183.52435	-183.38170	-28.4	-28.4
A anti gauche	-183.34844	-183.20559	-183.52428	-183.38143	-28.4	-28.2
M out	-183.34770	-183.20479	-183.52358	-183.38067	-27.8	-27.8
M anti	-183.34623	-183.20339	-183.52225	-183.37940	-27.0	-27.0
HEXENE						
Hexene out of plane	-235.19561	-235.02932	-235.41801	-235.25171	0	0
A INT out	-261.72874	-261.53024	-261.97918	-261.78068	-9.3	-10.0
A INT anti	-261.72895	-261.53022	-261.97919	-261.78046	-9.3	-9.9
M INT out	-261.72874	-261.53021	-	-	-9.3	-
M INT anti	-261.72895	-261.53022	-	-	-9.3	-
TSA out	-261.72862	-261.53040	-261.97924	-261.78102	-9.4	-10.2
TSA anti	-261.72839	-261.52990	-261.97903	-261.78054	-9.1	-9.9
TSM out	-261.72493	-261.52640	-261.97579	-261.77726	-6.9	-7.9
TSM anti	-261.72474	-261.52623	-261.975601	-261.77710	-6.8	-7.8
A out	-261.75980	-261.55937	-	-	-27.6	-
A out twisted	-261.76137	-261.56172	-262.01051	-261.81086	-29.1	-29.0
A out gauche	-261.76173	-261.56181	-262.01078	-261.81085	-29.1	-29.0
A anti	-261.75913	-261.55866	-262.00831	-261.80784	-27.2	-27.1
A anti twisted	-261.76067	-261.56078	-262.00981	-261.80992	-28.5	-28.4
M out	-261.76019	-261.56005	-262.00925	-261.80911	-28.0	-27.9
M anti	-261.75862	-261.55854	-262.00777	-261.80770	-27.1	-27.0
OCTENE						
Octene out of plane	-313.60786	-313.38435	-	-	0	-
A INT out	-340.14103	-339.88534	-	-	-9.4	-
A INT anti	-340.14126	-339.88534	-	-	-9.4	-
M INT out	-340.13703	-339.88531	-	-	-9.4	-
M INT anti	-340.14127	-339.88535	-	-	-9.4	-
TSA out	-340.14091	-339.88549	-	-	-9.5	-
TSA anti	-340.14070	-339.88504	-	-	-9.2	-
TSM out	-340.13720	-339.88149	-	-	-7.0	-
TSM anti	-340.13703	-339.88135	-	-	-6.9	-
A out	-340.17397	-339.91443	-	-	-27.6	-
A out twisted	-340.17361	-339.91677	-	-	-29.1	-
A out gauche	-340.17205	-339.91684	-	-	-29.1	-
A anti	-340.17139	-339.91369	-	-	-27.2	-
A anti twisted	-340.17293	-339.91585	-	-	-28.5	-
M out	-340.17244	-339.91511	-	-	-28.1	-
M anti	-340.17091	-339.91363	-	-	-27.1	-
DODECENE						
Dodecene out of plane	-470.43237	-470.09451	-	-	0	-

A INT out	-496.96556	-496.59549	-	-	-9.4	-
A INT anti	-496.96582	-496.59554	-	-	-9.4	-
M INT out	-496.96556	-496.59549	-	-	-9.4	-
M INT anti	-496.96582	-496.59554	-	-	-9.4	-
TSA out	-496.96544	-496.59568	-	-	-9.5	-
TSA anti	-496.96525	-496.59521	-	-	-9.2	-
TSM out	-496.96172	-496.59168	-	-	-7.0	-
TSM anti	-496.96157	-496.59152	-	-	-6.9	-
A out	-496.99657	-496.62460	-	-	-27.6	-
A out twisted	-496.99813	-496.62695	-	-	-29.1	-
A out gauche	-496.99848	-496.62702	-	-	-29.2	-
A anti	-496.99591	-496.62404	-	-	-27.3	-
A anti twisted	-496.99745	-496.62599	-	-	-28.5	-
M out	-496.99696	-496.62529	-	-	-28.1	-
M anti	-496.99543	-496.62382	-	-	-27.1	-
3,3-DIMETHYL-2-BUTENE						
3,3-dimethyl-2-butene	-235.20211	-235.03667	-235.42319	-235.25775	0	0
A INT	-261.73508	-261.53739	-261.98354	-261.78585	-9.2	-9.5
M INT	-261.73507	-261.53740	-	-	-9.2	-
TSA	-261.73479	-261.53733	-261.98353	-261.78607	-9.2	-9.6
TSM	-261.73120	-261.53351	-261.98035	-261.78266	-6.8	-7.5
A	-261.76658	-261.56710	-	-	-27.8	-
A twisted	-261.76788	-261.56914	-262.01544	-261.81670	-29.1	-28.8
M	-261.76473	-261.56580	-262.01208	-261.81314	-27.0	-26.6
M twisted	-261.76499	-261.56578	-262.01247	-261.81326	-27.0	-26.7

CCSD(T)/aug-cc-pvqz	Potential E	Relative E
BH_3	-26.545123	0
propene	-117.718677	0
INT	-144.285478	-13.6
TSA	-144.285580	-13.7
TSM	-144.281762	-11.3
Butene	-156.9702597	0
INT	-183.5373791	-13.8
TSA	-183.5375532	-13.9
TSM	-183.5341261	-11.8

The table below shows absolute and relative potential energies for **TSA** versus **TSM** for selected structures including a PCM solvent model for THF (at left) and gas phase (at right). The difference in energy increases with the solvent model in each case. The B3LYP energy difference is approximately constant versus the size of the alkene.

Table. Effect of a PCM Solvent Model on Relative Energies.

	B3LYP potential energies with pcm			B3LYP potential energies without pcm	
anti ts B butene	-183.8563554		anti ts B butene	-183.8525964	
Mark ts B butene	-183.8526792	2.31	Mark ts B butene	-183.8493441	2.04
anti ts B hexene	-262.4840226		anti ts B hexene	-262.480232	
Mark ts B hexene	-262.4803218	2.32	Mark ts B hexene	-262.4769287	2.07
anti ts B dodecene	-498.3664782		anti ts B dodecene	-498.3625728	
Mark ts B dodecene	-498.3627517	2.34	Mark ts B dodecene	-498.3592408	2.09
	CCSD(T)/aug-cc-pvdz		CCSD(T)/aug-cc-pvdz		
anti ts B butene	-1.83E+02		anti ts B butene	-1.83E+02	
Mark ts B butene	-1.83E+02	2.47	Mark ts B butene	-1.83E+02	2.26
	CCSD(T)/aug-cc-pvtz		CCSD(T)/aug-cc-pvtz		
anti ts B butene	-1.83E+02		anti ts B butene	-1.83E+02	
Mark ts B butene	-1.83E+02	2.33	Mark ts B butene	-1.83E+02	2.12

CCSD(T)/aug-cc-pvdz structures

For comparison with the B3LYP structures, some structures were obtained that were fully optimized in CCSD(T)/aug-cc-pvdz calculations. Frequency calculations were not performed on these structures.

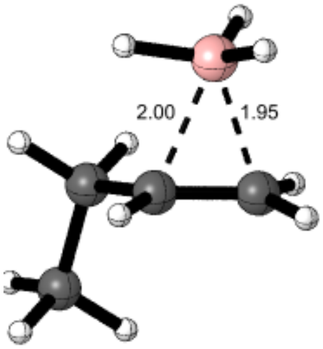
BH_3

CCSD(T)/aug-cc-pvdz
CCSD(T)= -26.5127597

B	0.000000	0.000000	0.000000
H	-1.190245	0.195977	0.000000
H	0.425402	-1.128771	0.000000
H	0.764843	0.932793	0.000000

Butene Complex

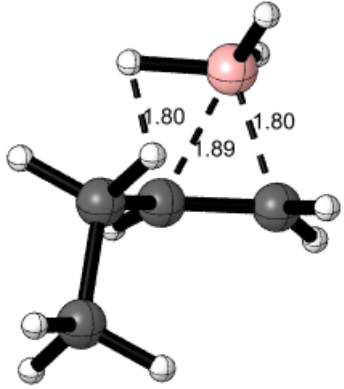
CCSD(T)/aug-cc-pvdz
CCSD(T)= -183.3178685



C	0.872374	-0.611238	-0.179042
C	-0.270509	0.129755	0.488708
C	-1.075283	1.033386	-0.184362
B	-2.008908	-0.679742	-0.070006
H	0.969926	-1.619172	0.261853
H	-0.309555	0.106563	1.585449
H	-0.936511	1.199645	-1.257686
H	-1.712910	1.736695	0.356468
H	-1.474090	-1.568357	0.579609
H	-2.025710	-0.935890	-1.255426
H	-2.998715	-0.261038	0.490412
H	0.638526	-0.738640	-1.252216
C	2.191832	0.174788	-0.004906
H	2.429742	0.317957	1.065916
H	3.031803	-0.371202	-0.470312
H	2.121552	1.172001	-0.476424

Butene AMTS

CCSD(T)/aug-cc-pvdz
CCSD(T)= -183.3171005

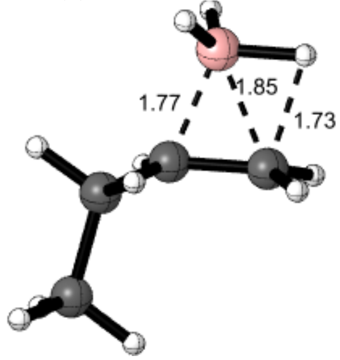


C	0.886965	-0.638706	-0.171891
C	-0.298413	0.036387	0.498317

C	-1.091854	0.987342	-0.176867
B	-1.979864	-0.575683	-0.108532
H	1.050276	-1.638631	0.269166
H	-0.314203	0.031297	1.595934
H	-0.863866	1.218152	-1.222312
H	-1.651584	1.736421	0.387989
H	-2.955867	-0.341464	0.566378
H	-1.325839	-1.435790	0.510851
H	-2.025128	-0.945248	-1.260568
H	0.651932	-0.781817	-1.242921
C	2.153758	0.229599	-0.009466
H	3.024929	-0.262568	-0.477489
H	2.017483	1.218174	-0.484308
H	2.388456	0.392153	1.059376

Butene MTS

CCSD(T)/aug-cc-pvdz
CCSD(T)= -183.3133853



H	0.952097	1.240983	-1.235950
C	1.133575	0.987328	-0.184525
C	0.317651	0.022117	0.459907
B	1.873858	-0.699600	0.005244
H	1.709454	1.717098	0.393581
C	-0.891542	-0.588434	-0.240219
H	0.279039	0.059579	1.555054
H	2.468227	-0.918525	1.033590
H	2.572232	0.139092	-0.618853
H	1.591111	-1.538462	-0.816951
H	-1.016576	-1.636621	0.089033
H	-0.697227	-0.619288	-1.329462
C	-2.177902	0.212264	0.051398
H	-3.053605	-0.249634	-0.440407
H	-2.379326	0.251551	1.138686
H	-2.085409	1.252576	-0.313910

B3LYP/6-31G* structures

The structures associated with propene have been previously reported but are included here to simplify the reproduction of calculations based on these structures.

BH_3

B3LYP/6-31G*
E(RB3LYP) = -26.6130000928

Zero-point correction= 0.026503 (Hartree/Particle)
Thermal correction to Energy= 0.029387
Thermal correction to Enthalpy= 0.030331
Thermal correction to Gibbs Free Energy= 0.008948
Sum of electronic and ZPE= -26.586498
Sum of electronic and thermal Energies= -26.583614
Sum of electronic and thermal Enthalpies= -26.582669
Sum of electronic and thermal Free Energies= -26.604053

E CV S
KCal/Mol Cal/Mol-K Cal/Mol-K
Total 18.440 6.589 45.005

B,0,0.,0.,0.
H,0,0.,0.,1.1938172328
H,0,1.0338760511,0.,-0.5969086164
H,0,-1.0338760511,0.,-0.5969086164

Propene

E(RB3LYP) = -117.907562163

Zero-point correction= 0.080099 (Hartree/Particle)
Thermal correction to Energy= 0.084173
Thermal correction to Enthalpy= 0.085117
Thermal correction to Gibbs Free Energy= 0.055099
Sum of electronic and ZPE= -117.827463
Sum of electronic and thermal Energies= -117.823389
Sum of electronic and thermal Enthalpies= -117.822445
Sum of electronic and thermal Free Energies= -117.852463

E CV S
KCal/Mol Cal/Mol-K Cal/Mol-K
Total 52.819 12.908 63.178

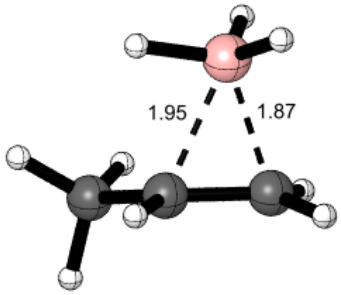
C,0,-0.0316860103,0.,-0.0387837613
C,0,0.0318476871,0.,1.2929722911
C,0,1.2982738493,0.,2.1006617837
H,0,-0.8976095917,0.,1.864679706
H,0,-0.9813004684,0.,-0.5672647441
H,0,0.8664849317,0.,-0.6537040143
H,0,2.18509124,0.,1.4579709445
H,0,1.3514294028,0.8804498739,2.7553276244
H,0,1.3514294028,-0.8804498739,2.7553276244

Propene Complex

B3LYP/6-31G*
E(RB3LYP) = -144.539354036

Zero-point correction= 0.112612 (Hartree/Particle)
Thermal correction to Energy= 0.118633
Thermal correction to Enthalpy= 0.119577
Thermal correction to Gibbs Free Energy= 0.084624
Sum of electronic and ZPE= -144.426742
Sum of electronic and thermal Energies= -144.420721
Sum of electronic and thermal Enthalpies= -144.419777
Sum of electronic and thermal Free Energies= -144.454730

E CV S
KCal/Mol Cal/Mol-K Cal/Mol-K
Total 74.443 21.094 73.565



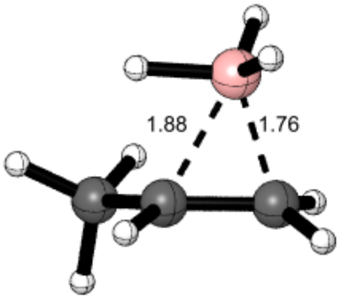
C	0.716835	0.145815	-0.039186
C	-0.056497	0.027531	1.245992
C	0.528086	-0.209957	2.470351
B	0.160630	1.617567	2.355429
H	0.246435	0.848421	-0.732447
H	0.752695	-0.838331	-0.525077
H	-1.130343	-0.115873	1.152808
H	1.605946	-0.303088	2.553882
H	-0.052781	-0.599894	3.297578
H	-0.482056	2.001277	1.397923
H	1.267385	2.079297	2.312445
H	-0.528173	1.731387	3.331479
H	1.744114	0.472159	0.146936

Propene AMTS

B3LYP/6-31G*
E(RB3LYP) = -144.538993703

Zero-point correction= 0.112431 (Hartree/Particle)
Thermal correction to Energy= 0.117611
Thermal correction to Enthalpy= 0.118555
Thermal correction to Gibbs Free Energy= 0.085396
Sum of electronic and ZPE= -144.426563
Sum of electronic and thermal Energies= -144.421382
Sum of electronic and thermal Enthalpies= -144.420438
Sum of electronic and thermal Free Energies= -144.453598

```
      E     CV       S
  KCal/Mol Cal/Mol-K Cal/Mol-K
Total 73.802 18.766 69.791
```



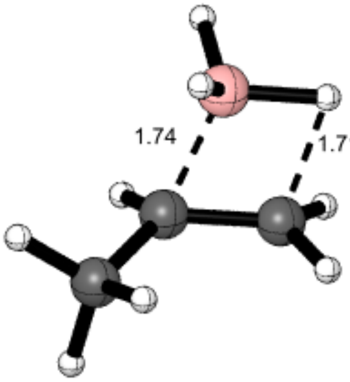
C	-0.382938	-0.400571	-1.498259
C	-0.409522	-0.408923	0.008951
C	0.756411	-0.409591	0.785972
B	0.073461	1.207558	0.831833
H	-1.286274	0.046214	-1.923654
H	-0.311532	-1.435961	-1.855533
H	-1.328328	-0.765745	0.468932
H	1.720840	-0.396967	0.287933
H	0.743158	-0.848613	1.777026
H	-0.244627	1.399319	1.971150
H	-0.987926	1.267311	0.204083
H	0.833715	1.910950	0.227519
H	0.487721	0.147763	-1.869757

Propene MTS

B3LYP/6-31G*
E(RB3LYP) = -144.535333246

Zero-point correction= 0.112557 (Hartree/Particle)
Thermal correction to Energy= 0.117708
Thermal correction to Enthalpy= 0.118652
Thermal correction to Gibbs Free Energy= 0.085577
Sum of electronic and ZPE= -144.422776
Sum of electronic and thermal Energies= -144.417625
Sum of electronic and thermal Enthalpies= -144.416681
Sum of electronic and thermal Free Energies= -144.449756

```
      E     CV       S
  KCal/Mol Cal/Mol-K Cal/Mol-K
Total 73.863 18.638 69.613
```



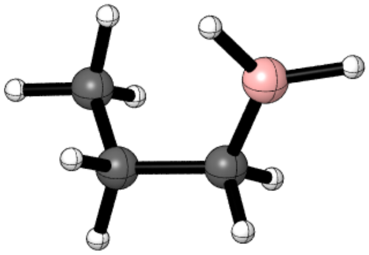
C	-0.365342	0.391568	-1.491184
C	-0.366020	0.391434	0.024379
C	0.827056	0.391911	0.772323
B	-0.135395	-1.165848	0.776756
H	-1.242451	-0.129908	-1.887130
H	-0.381662	1.418843	-1.879316
H	-1.225548	0.866818	0.489737
H	1.786682	0.295416	0.270888
H	0.866123	0.833289	1.763444
H	0.044953	-1.947797	-0.111035
H	0.940226	-1.187863	1.403788
H	-0.973569	-1.294841	1.620789
H	0.524873	-0.107926	-1.887239

Propene AM Product

B3LYP/6-31G*
E(RB3LYP) = -144.571577894

Zero-point correction= 0.114027 (Hartree/Particle)
Thermal correction to Energy= 0.120046
Thermal correction to Enthalpy= 0.120990
Thermal correction to Gibbs Free Energy= 0.085856
Sum of electronic and ZPE= -144.457550
Sum of electronic and thermal Energies= -144.451532
Sum of electronic and thermal Enthalpies= -144.450588
Sum of electronic and thermal Free Energies= -144.485722

E CV S
KCal/Mol Cal/Mol-K Cal/Mol-K
Total 75.330 20.520 73.946



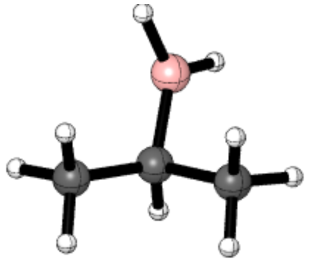
C	-0.167792	0.220722	-0.019849
C	-0.356496	-0.081912	1.470508
C	0.910226	0.202820	2.294993
B	1.348787	1.687735	2.489635
H	-1.079580	0.012484	-0.591237
H	0.639810	-0.386634	-0.447832
H	-0.657525	-1.131773	1.589412
H	1.749961	-0.422325	1.954136
H	0.756645	-0.127992	3.344388
H	0.591850	2.597837	2.293018
H	-1.185950	0.522967	1.860831
H	2.440271	1.944358	2.913593
H	0.091050	1.275360	-0.179061

Propene M Product

B3LYP/6-31G*
E(RB3LYP) = -144.568964309

Zero-point correction= 0.114180 (Hartree/Particle)
Thermal correction to Energy= 0.120349
Thermal correction to Enthalpy= 0.121293
Thermal correction to Gibbs Free Energy= 0.086068
Sum of electronic and ZPE= -144.454784
Sum of electronic and thermal Energies= -144.448615
Sum of electronic and thermal Enthalpies= -144.447671
Sum of electronic and thermal Free Energies= -144.482896

E CV S
KCal/Mol Cal/Mol-K Cal/Mol-K
Total 75.520 20.980 74.138



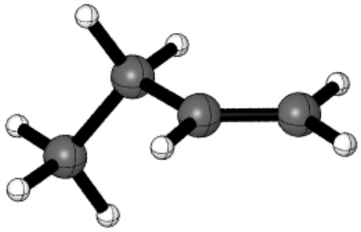
C	0.159981	-0.140557	0.073462
C	0.036717	-0.231314	1.602787
C	1.392401	0.127632	2.285087
B	-0.294887	-1.651000	2.172635
H	-0.793757	-0.359535	-0.421212
H	0.474860	0.861767	-0.246302
H	-0.695073	0.511010	1.952704
H	1.714092	1.127390	1.967061
H	1.320030	0.133932	3.378785
H	0.068138	-2.640813	1.599383
H	2.186205	-0.574959	2.002038
H	-0.837446	-1.778163	3.234045
H	0.897632	-0.858865	-0.305309

Butene Out of plane

B3LYP/6-31G*
E(RB3LYP) = -157.221072411

Zero-point correction= 0.109014 (Hartree/Particle)
Thermal correction to Energy= 0.114265
Thermal correction to Enthalpy= 0.115209
Thermal correction to Gibbs Free Energy= 0.081527
Sum of electronic and ZPE= -157.112059
Sum of electronic and thermal Energies= -157.106807
Sum of electronic and thermal Enthalpies= -157.105863
Sum of electronic and thermal Free Energies= -157.139545

E CV S
KCal/Mol Cal/Mol-K Cal/Mol-K
Total 71.702 17.643 70.890



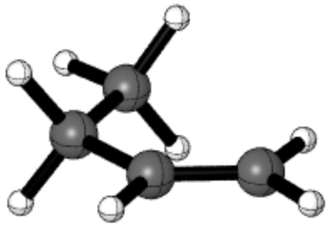
C,0,-0.247919402,-0.5070131258,-1.4362530184
C,0,-0.2598029168,-0.633739236,0.0632469602
C,0,0.7337860797,-0.2745459274,0.8767951623
H,0,-1.0671275963,0.1571428394,-1.7506304772
C,0,-0.4101652809,-1.8605287448,-2.1488352843
H,0,-1.1639950144,-1.0700242903,0.493227837
H,0,1.6551009816,0.1614885252,0.4946947797
H,0,0.6668220106,-0.3978732421,1.9545783381
H,0,0.6846216667,-0.0256903378,-1.7572461408
H,0,-0.4297163855,-1.7324710302,-3.2370516926
H,0,0.4157198966,-2.5357636157,-1.8992145162
H,0,-1.3441170393,-2.3539538145,-1.8536409478

Butene In Plane

B3LYP/6-31G*
E(RB3LYP) = -157.220361409

Zero-point correction= 0.109144 (Hartree/Particle)
Thermal correction to Energy= 0.114277
Thermal correction to Enthalpy= 0.115221
Thermal correction to Gibbs Free Energy= 0.081928
Sum of electronic and ZPE= -157.111217
Sum of electronic and thermal Energies= -157.106085
Sum of electronic and thermal Enthalpies= -157.105141
Sum of electronic and thermal Free Energies= -157.138434

E CV S
KCal/Mol Cal/Mol-K Cal/Mol-K
Total 71.710 17.565 70.071



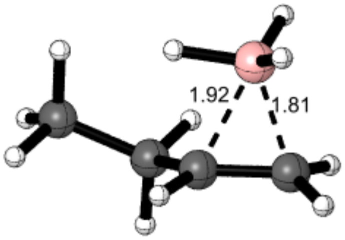
```
C,0,-0.2958085067,-0.4093276225,-1.4533058824
C,0,-0.3307354651,-0.4717965348,0.052946196
C,0,0.7164928677,-0.4356739083,0.878220354
H,0,-0.919667336,0.4372011606,-1.7795646229
C,0,1.0931781515,-0.2925261027,-2.0840670582
H,0,-1.3299982603,-0.5549312968,0.4840237784
H,0,1.7388635287,-0.3538638953,0.5188733164
H,0,0.5883373053,-0.4871728249,1.9560880365
H,0,-0.804983052,-1.3017435413,-1.8490237014
H,0,1.0192459411,-0.253785765,-3.1761350695
H,0,1.6072063914,0.6160643711,-1.7505083316
H,0,1.7236514345,-1.1494580401,-1.8210200153
```

Butene Complex out

B3LYP/6-31G*
E(RB3LYP) = -183.853110954

Zero-point correction= 0.141268 (Hartree/Particle)
Thermal correction to Energy= 0.148478
Thermal correction to Enthalpy= 0.149422
Thermal correction to Gibbs Free Energy= 0.111190
Sum of electronic and ZPE= -183.711843
Sum of electronic and thermal Energies= -183.704633
Sum of electronic and thermal Enthalpies= -183.703689
Sum of electronic and thermal Free Energies= -183.741921

```
      E      CV       S
  KCal/Mol Cal/Mol-K Cal/Mol-K
Total 93.171 25.831 80.466
```



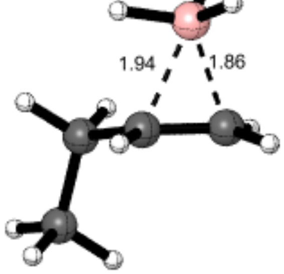
```
C,0,-0.3771801621,-0.3716153827,-1.484176563
C,0,-0.3945289441,-0.4092893283,0.0251078674
C,0,0.7531878912,-0.3907034679,0.8032921664
B,0,0.0317668119,1.2699971672,0.8456014853
C,0,-1.6453953667,0.1856706063,-2.136187435
H,0,-0.226753556,-1.4113853618,-1.8129822821
H,0,-1.3210316187,-0.7486427122,0.4848363472
H,0,1.7264057725,-0.3262764658,0.3273364681
H,0,0.7419413928,-0.7954877729,1.8084853677
H,0,-0.1985752676,1.4174750092,2.0135274377
H,0,-1.0372019898,1.4030229138,0.2696383391
H,0,0.8421520561,1.9348292665,0.2607316634
H,0,0.50600955,0.1862921006,-1.8172517381
H,0,-1.5863811091,0.1016604105,-3.2264008297
H,0,-2.5359146556,-0.3635563557,-1.8074787245
H,0,-1.7873738047,1.2408753732,-1.8831825699
```

Butene Complex anti

B3LYP/6-31G*
E(RB3LYP) = -183.852872662

Zero-point correction= 0.141473 (Hartree/Particle)
Thermal correction to Energy= 0.148752
Thermal correction to Enthalpy= 0.149697
Thermal correction to Gibbs Free Energy= 0.111205
Sum of electronic and ZPE= -183.711400
Sum of electronic and thermal Energies= -183.704120
Sum of electronic and thermal Enthalpies= -183.703176
Sum of electronic and thermal Free Energies= -183.741667

```
      E      CV       S
  KCal/Mol Cal/Mol-K Cal/Mol-K
Total 93.344 25.866 81.012
```



```
C,0,-0.3583501994,-0.4061427618,-1.4744896255
C,0,-0.3853099473,-0.424319802,0.0344197558
C,0,0.7467819809,-0.3572819781,0.8187368401
B,0,-0.0431004037,1.3260580925,0.8027075358
H,0,-1.2462816095,0.1101126378,-1.8552612651
C,0,-0.3060719472,-1.8362280813,-2.040664833
H,0,-1.3098639188,-0.767543844,0.4942658259
H,0,1.723917228,-0.2511723037,0.358402925
H,0,0.7364451245,-0.7051572412,1.8448238467
H,0,-0.2353612561,1.4713321193,1.9780730638
H,0,-1.1126150028,1.4361737255,0.2343352459
H,0,0.7838774629,1.9690031109,0.2173460353
H,0,0.5139997997,0.1643914657,-1.8129361338
H,0,-0.3065080615,-1.8122544619,-3.1356076743
H,0,0.5972950865,-2.3614960171,-1.7122716422
H,0,-1.1728463361,-2.4258796607,-1.7197469004
```

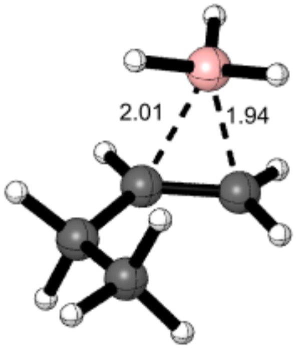
Butene Complex in

B3LYP/6-31G*
E(RB3LYP) = -183.851492972

Zero-point correction= 0.141499 (Hartree/Particle)
Thermal correction to Energy= 0.148785
Thermal correction to Enthalpy= 0.149729
Thermal correction to Gibbs Free Energy= 0.111201
Sum of electronic and ZPE= -183.709993
Sum of electronic and thermal Energies= -183.702708
Sum of electronic and thermal Enthalpies= -183.701764
Sum of electronic and thermal Free Energies= -183.740292

```
      E      CV       S
```

```
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 93.364 25.958 81.089
```



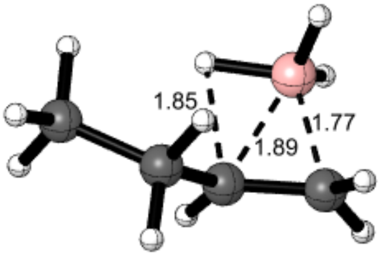
```
C,0,-0.3798773097,-0.4155536036,-1.4836275978
C,0,-0.3210936699,-0.4441023498,0.0266201996
C,0,0.8077345336,-0.3113056194,0.7874161584
B,0,-0.1955529704,1.3358045361,0.9510500174
H,0,-1.3079800764,0.0813840175,-1.7878419486
H,0,-0.4831736771,-1.4637506306,-1.8058751062
H,0,-1.2279055788,-0.7923104607,0.5166755877
H,0,1.7699665672,-0.1139620692,0.3286048825
H,0,0.8266906224,-0.6157452696,1.827270748
H,0,-0.7789372529,1.2129160301,1.9907744204
H,0,-0.8788637945,1.6822936352,0.0187030105
H,0,0.8338150153,1.9615111428,0.9453115039
C,0,0.8136776873,0.2260239438,-2.1938132157
H,0,0.6540977454,0.2213271724,-3.2769088346
H,0,0.9471781454,1.2652095012,-1.8761023816
H,0,1.745778013,-0.3153359763,-1.9970454441
```

Butene AMTS out

```
B3LYP/6-31G*
E(RB3LYP) = -183.853079894

Zero-point correction= 0.140980 (Hartree/Particle)
Thermal correction to Energy= 0.147476
Thermal correction to Enthalpy= 0.148420
Thermal correction to Gibbs Free Energy= 0.111643
Sum of electronic and ZPE= -183.712100
Sum of electronic and thermal Energies= -183.705604
Sum of electronic and thermal Enthalpies= -183.704660
Sum of electronic and thermal Free Energies= -183.741437

     E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 92.543 23.758 77.404
```



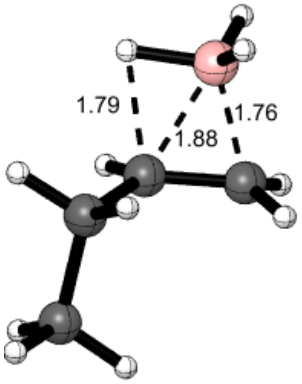
```
C,0,-0.3768166225,-0.3821672094,-1.4850707602
C,0,-0.3907847359,-0.4155956167,0.0250490594
C,0,0.7711049552,-0.4263551328,0.7986877484
B,0,0.0914724406,1.2089166838,0.8665260158
C,0,-1.6405864945,0.1878308729,-2.1345195559
H,0,-0.2365743355,-1.4240390443,-1.8110634357
H,0,-1.3149161355,-0.7618116758,0.4847825625
H,0,1.7375806519,-0.3976373925,0.3053926384
H,0,0.7593757175,-0.8686259739,1.7882595864
H,0,-0.1895892389,1.3751712341,2.0201298549
H,0,-0.9746933159,1.3192125066,0.2644276607
H,0,0.8754626894,1.8991294902,0.276017252
H,0,0.5111886599,0.1666123923,-1.8204235776
H,0,-1.5837526584,0.1071062611,-3.2250646576
H,0,-2.5359027879,-0.3538693498,-1.8064267162
H,0,-1.7728377894,1.2440909542,-1.8789046753
```

Butene AMTS anti

```
B3LYP/6-31G*
E(RB3LYP) = -183.852596446

Zero-point correction= 0.141213 (Hartree/Particle)
Thermal correction to Energy= 0.147697
Thermal correction to Enthalpy= 0.148641
Thermal correction to Gibbs Free Energy= 0.111882
Sum of electronic and ZPE= -183.711383
Sum of electronic and thermal Energies= -183.704900
Sum of electronic and thermal Enthalpies= -183.703955
Sum of electronic and thermal Free Energies= -183.740715

     E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 92.681 23.624 77.366
```



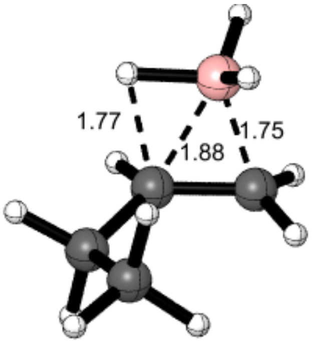
```
C,0,-0.377109718,-0.3951255229,-1.4828903688
C,0,-0.4036113944,-0.3874577382,0.0289757518
C,0,0.7603405146,-0.3848906009,0.8073650043
B,0,0.0701121363,1.2335128793,0.8479161122
H,0,-1.27270367,0.0991056864,-1.876417694
C,0,-0.296078341,-1.8326155857,-2.0220251904
H,0,-1.3232346219,-0.7442668715,0.4891192732
H,0,1.7258998254,-0.3657746592,0.3112492696
H,0,0.7496489945,-0.8185075976,1.8008446575
H,0,-0.2551777181,1.4216694729,1.9858539
H,0,-0.9832641658,1.298128566,0.2095012432
H,0,0.8408403509,1.9358393807,0.2559144132
H,0,0.4864960436,0.1850941814,-1.827639239
H,0,-0.2903195571,-1.8292399648,-3.1172199451
H,0,0.6143623494,-2.3355229552,-1.6788928733
H,0,-1.1540970285,-2.4319356709,-1.6953173143
```

Butene AMTS in

B3LYP/6-31G*
E(RB3LYP) = -183.850656787

Zero-point correction= 0.141431 (Hartree/Particle)
Thermal correction to Energy= 0.147788
Thermal correction to Enthalpy= 0.148732
Thermal correction to Gibbs Free Energy= 0.112287
Sum of electronic and ZPE= -183.709226
Sum of electronic and thermal Energies= -183.702869
Sum of electronic and thermal Enthalpies= -183.701924
Sum of electronic and thermal Free Energies= -183.738370

```
   E     CV       S
 KCal/Mol Cal/Mol-K Cal/Mol-K
Total 92.739 23.447 76.706
```



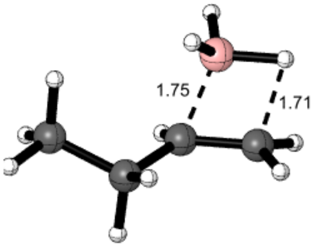
```
C,0,-0.3665513944,-0.3803225913,-1.4896073777
C,0,-0.3794443209,-0.3761224355,0.0272558693
C,0,0.756832547,-0.3718651386,0.8499080848
B,0,0.0154289969,1.2115041737,0.9582958413
H,0,-1.2974781468,0.0696935305,-1.8546645989
H,0,-0.4097063755,-1.441629525,-1.7791064642
H,0,-1.3010622387,-0.757067691,0.4631693889
H,0,1.7419640218,-0.3103866552,0.4008627116
H,0,0.7141762524,-0.8589902283,1.8173779784
H,0,-0.4340498623,1.3148414451,2.0642231779
H,0,-0.9660805179,1.2854906409,0.2089731296
H,0,0.8095230111,1.9842062288,0.5031607711
C,0,0.8330893857,0.2854533059,-2.1677042922
H,0,0.730060003,0.2178261736,-3.2557040559
H,0,0.9043577215,1.3428659622,-1.8974061315
H,0,1.7764789171,-0.1997441958,-1.8955730326
```

Butene MTS out

B3LYP/6-31G*
E(RB3LYP) = -183.849336691

Zero-point correction= 0.141306 (Hartree/Particle)
Thermal correction to Energy= 0.147684
Thermal correction to Enthalpy= 0.148628
Thermal correction to Gibbs Free Energy= 0.112203
Sum of electronic and ZPE= -183.708031
Sum of electronic and thermal Energies= -183.701653
Sum of electronic and thermal Enthalpies= -183.700708
Sum of electronic and thermal Free Energies= -183.737134

```
   E     CV       S
 KCal/Mol Cal/Mol-K Cal/Mol-K
Total 92.673 23.457 76.664
```



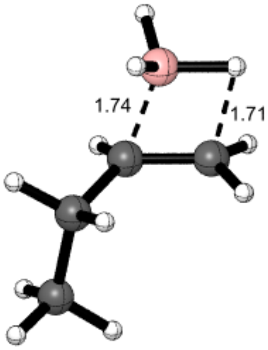
```
C,0,-0.3524889635,0.3614869034,-1.4798311054
C,0,-0.3528953454,0.3744478104,0.0397807696
C,0,0.8354805657,0.4121303924,0.79191366
B,0,-0.0845832688,-1.1706060711,0.8143605506
C,0,-1.5996827297,-0.2893281835,-2.0854697222
H,0,-0.2803105874,1.4006302049,-1.8355918196
H,0,-1.2275564573,0.8330317706,0.4961195438
H,0,1.7994824255,0.3278793954,0.2967140306
H,0,0.8607941696,0.8688597126,1.7765256443
H,0,0.1225614025,-1.9579589028,-0.063202534
H,0,0.9826846208,-1.1631093501,1.4524099676
H,0,-0.9297891411,-1.3121644732,1.6493474871
H,0,0.5477460092,-0.1536653873,-1.8375088923
H,0,-1.5756208876,-0.239576975,-3.1797917213
H,0,-1.672868262,-1.3407773446,-1.7898538477
H,0,-2.5124115506,0.2178614981,-1.7493870111
```

Butene MTS anti

B3LYP/6-31G*
E(RB3LYP) = -183.849344083

Zero-point correction= 0.141251 (Hartree/Particle)
Thermal correction to Energy= 0.147682
Thermal correction to Enthalpy= 0.148626
Thermal correction to Gibbs Free Energy= 0.112019
Sum of electronic and ZPE= -183.708093
Sum of electronic and thermal Energies= -183.701662
Sum of electronic and thermal Enthalpies= -183.700718
Sum of electronic and thermal Free Energies= -183.737325

```
   E     CV       S
 KCal/Mol Cal/Mol-K Cal/Mol-K
Total 92.672 23.491 77.046
```



```
C,0,-0.3632389466,0.3791947665,-1.4765163165
C,0,-0.3653350722,0.3692251437,0.0426127774
C,0,0.8269840137,0.3747598968,0.7911307757
B,0,-0.1287882746,-1.1884688491,0.7919436782
H,0,-1.2301603564,-0.1829950572,-1.8439270409
C,0,-0.3924586511,1.8026105263,-2.0517199589
```

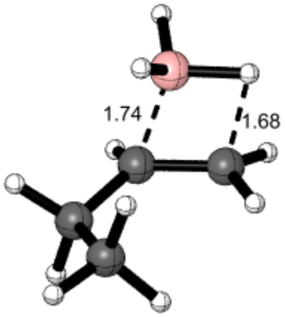
```
H,0,-1.2279814428,0.842931971,0.50603138
H,0,1.786960251,0.2788101567,0.2898904538
H,0,0.8658945529,0.8139506558,1.7832684671
H,0,0.0550859964,-1.9663380918,-0.0989309599
H,0,0.943419034,-1.2113515485,1.4229643422
H,0,-0.9713186855,-1.3210745782,1.6311090847
H,0,0.5231546322,-0.1548302158,-1.8408933258
H,0,-0.414017584,1.7832047495,-3.1473678739
H,0,-1.2797320652,2.3506463199,-1.711900613
H,0,0.4893825981,2.3759011544,-1.7414988701
```

Butene MTS in

```
B3LYP/6-31G*
E(RB3LYP) = -183.847241490

Zero-point correction= 0.141643 (Hartree/Particle)
Thermal correction to Energy= 0.147901
Thermal correction to Enthalpy= 0.148845
Thermal correction to Gibbs Free Energy= 0.112708
Sum of electronic and ZPE= -183.705599
Sum of electronic and thermal Energies= -183.699341
Sum of electronic and thermal Enthalpies= -183.698396
Sum of electronic and thermal Free Energies= -183.734534

      E     CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 92.809 23.219 76.058
```



```
C,0,-0.3528460413,0.3499183581,-1.4792010531
C,0,-0.3424777376,0.3701114812,0.0470738925
C,0,0.8328952095,0.3303798225,0.8278857264
B,0,-0.2252213088,-1.1574325504,0.8620880468
H,0,-1.2295293906,-0.2186973194,-1.8123187251
H,0,-0.5152861896,1.3822348365,-1.8233278776
H,0,-1.1780183665,0.9122549822,0.4827848366
H,0,1.8003392287,0.1647534528,0.3634574117
H,0,0.8684585458,0.8129788979,1.7998077389
H,0,-0.0613759659,-1.9963533118,0.024164575
H,0,0.8409024875,-1.2055554969,1.5091205568
H,0,-1.0799192174,-1.2072197922,1.6974936403
C,0,0.8993786987,-0.2061052197,-2.1615043483
H,0,0.7611035755,-0.223472322,-3.247859967
H,0,1.7839232918,0.4088969208,-1.9563471199
H,0,1.1124301802,-1.2301237395,-1.8365203339
```

Butene AM Product out

```
B3LYP/6-31G*
E(RB3LYP) = -183.884063239

Zero-point correction= 0.143186 (Hartree/Particle)
Thermal correction to Energy= 0.150741
Thermal correction to Enthalpy= 0.151685
Thermal correction to Gibbs Free Energy= 0.112041
Sum of electronic and ZPE= -183.740877
Sum of electronic and thermal Energies= -183.733323
Sum of electronic and thermal Enthalpies= -183.732379
Sum of electronic and thermal Free Energies= -183.772023

      E     CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 94.591 25.209 83.438

 C,0,-0.5441999612,-0.4559092773,-1.5451745274
 C,0,-0.4880095734,0.3316807647,-0.2271713403
 C,0,0.8149548338,0.056186284,0.5836276649
 B,0,0.7398733293,0.9144198065,1.889564238
 C,0,-1.8217199414,-0.1905193216,-2.347748795
 H,0,-0.4619663473,-1.5290923808,-1.3244091875
 H,0,-1.3677818514,0.0747288513,0.379674915
 H,0,1.672221946,0.3215903463,-0.0493072236
 H,0,0.8677425476,-1.0216546272,0.7871495942
 H,0,0.2084631551,0.4907239454,2.8776390664
 H,0,-0.5706836159,1.4050822758,-0.4491296544
 H,0,1.1389467932,2.0451563818,1.9095365302
 H,0,0.3346327697,-0.1996692378,-2.1528101926
 H,0,-1.8321287014,-0.7656200254,-3.2807268892
 H,0,-2.7155020403,-0.4682768703,-1.7756228015
 H,0,-1.9131883426,0.8707100845,-2.6099453973
```

Butene AM Product out twisted

```
B3LYP/6-31G*
E(RB3LYP) = -183.885759815

Zero-point correction= 0.142424 (Hartree/Particle)
Thermal correction to Energy= 0.149934
Thermal correction to Enthalpy= 0.150878
Thermal correction to Gibbs Free Energy= 0.111494
Sum of electronic and ZPE= -183.743336
Sum of electronic and thermal Energies= -183.735826
Sum of electronic and thermal Enthalpies= -183.734882
Sum of electronic and thermal Free Energies= -183.774265

      E     CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 94.085 25.503 82.890

 C,0,-0.5774236576,-0.4831810864,-1.5387033098
 C,0,-0.559433044,0.3496327181,-0.2505324163
 C,0,0.6489635133,0.0660487511,0.6561489397
 B,0,0.838118962,0.9373465796,1.9369341566
 C,0,-1.7915482117,-0.1987839294,-2.4285637307
 H,0,-0.5551940144,-1.5513940527,-1.2784932165
 H,0,-1.4863329623,0.162426846,0.3107907328
 H,0,1.588331762,0.2395299197,0.0897247399
 H,0,0.7098555842,-1.0051686466,0.9077844328
 H,0,1.6531595915,0.6245712342,2.7586191678
 H,0,-0.5780265297,1.4184255604,-0.5071329255
 H,0,0.2219495889,1.9518752392,2.1105533409
 H,0,0.3459874223,-0.2913344187,-2.104585018
```

H,0,-1.7734506421,-0.808462756,-3.3391921607
H,0,-2.7288171394,-0.4157727725,-1.90126531
H,0,-1.821457223,0.8546058139,-2.7335704229

Butene AM Product out gauche

B3LYP/6-31G*
E(RB3LYP) = -183.885488497

Zero-point correction= 0.142683 (Hartree/Particle)
Thermal correction to Energy= 0.149984
Thermal correction to Enthalpy= 0.150928
Thermal correction to Gibbs Free Energy= 0.112251
Sum of electronic and ZPE= -183.742805
Sum of electronic and thermal Energies= -183.735504
Sum of electronic and thermal Enthalpies= -183.734560
Sum of electronic and thermal Free Energies= -183.773238

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 94.116 25.347 81.404
```

C,0,-0.6187297248,-0.5335321512,-1.5624733584
C,0,-0.4860032978,0.3730560246,-0.3315539228
C,0,0.8144821747,0.1268699985,0.4511668812
B,0,2.163554007,0.5597213282,-0.2022732215
C,0,-1.9099779499,-0.2993471008,-2.3520525886
H,0,-0.5690420698,-1.584812086,-1.2434158405
H,0,-1.3582441841,0.2190499196,0.3201129696
H,0,0.8119007772,0.7374473168,1.379563681
H,0,0.8584835587,-0.9063956998,0.8275211949
H,0,3.2051744663,0.1588991328,0.2350192018
H,0,-0.5237398325,1.4232882266,-0.6538366885
H,0,2.1972344224,1.3558021321,-1.0995499505
H,0,0.2477825581,-0.3743710099,-2.2213063453
H,0,-1.9742226929,-0.9624852071,-3.2223204823
H,0,-2.7938453381,-0.4819475363,-1.7283049593
H,0,-1.9700298744,0.7340377118,-2.7151515707

Butene AM Product anti

B3LYP/6-31G*
E(RB3LYP) = -183.882835669

Zero-point correction= 0.143228 (Hartree/Particle)
Thermal correction to Energy= 0.150785
Thermal correction to Enthalpy= 0.151729
Thermal correction to Gibbs Free Energy= 0.111532
Sum of electronic and ZPE= -183.739608
Sum of electronic and thermal Energies= -183.732051
Sum of electronic and thermal Enthalpies= -183.731106
Sum of electronic and thermal Free Energies= -183.771304

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 94.619 25.126 84.602
```

C,0,-0.5151442973,-0.522726232,-1.7665388042
C,0,-0.5615821353,0.1959991121,-0.4053306332
C,0,0.7677709918,0.0989016512,0.4040375462
B,0,0.5886468121,0.9512613497,1.7036279642
H,0,-1.4476154792,-0.3005699716,-2.3037165082
C,0,-0.3403648527,-2.0417730336,-1.6765908597
H,0,-1.3870876385,-0.2163711634,0.1913536583
H,0,1.5890817045,0.4532404244,-0.2326546932
H,0,0.9544925419,-0.9576841438,0.6374318413
H,0,-0.0213036184,0.5113931875,2.6381420976
H,0,-0.8084286823,1.2514059801,-0.5838662555
H,0,0.984959151,2.0806748705,1.775502382
H,0,0.2964150829,-0.0934401955,-2.3704403241
H,0,-0.3665277121,-2.4982605952,-2.6726198226
H,0,0.6146793189,-2.3163251378,-1.2151781638
H,0,-1.1413691872,-2.4972031026,-1.0806664253

Butene AM Product anti twisted

B3LYP/6-31G*
E(RB3LYP) = -183.884555032

Zero-point correction= 0.142658 (Hartree/Particle)
Thermal correction to Energy= 0.150026
Thermal correction to Enthalpy= 0.150971
Thermal correction to Gibbs Free Energy= 0.111977
Sum of electronic and ZPE= -183.741897
Sum of electronic and thermal Energies= -183.734529
Sum of electronic and thermal Enthalpies= -183.733584
Sum of electronic and thermal Free Energies= -183.772578

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 94.143 25.392 82.068
```

C,0,-0.4432871852,-0.5308326826,-1.83555931
C,0,-0.5577520489,0.278687965,-0.5329254513
C,0,0.6564347021,0.1683586276,0.4057815359
B,0,0.5841137819,0.8820595778,1.790386295
H,0,-1.2904810226,-0.2733374458,-2.4857364986
C,0,-0.4150484661,-2.0503417034,-1.6348681246
H,0,-1.4629032274,-0.0302561453,0.009260808
H,0,1.588649503,0.418840399,-0.124401789
H,0,0.8110842568,-0.8902999203,0.703663807
H,0,1.5798864302,1.0448775887,2.4377012408
H,0,-0.7099817829,1.3357748676,-0.790271673
H,0,-0.461772926,1.2264936122,2.2665263386
H,0,0.461180829,-0.2169926701,-2.3768101447
H,0,-0.3707221875,-2.5747599403,-2.5962219673
H,0,0.4546991452,-2.3696710661,-1.0489394039
H,0,-1.3142518017,-2.395556064,-1.1089276628

Butene AM Product anti gauche

B3LYP/6-31G*
E(RB3LYP) = -183.883997234

Zero-point correction= 0.142852 (Hartree/Particle)
Thermal correction to Energy= 0.150084
Thermal correction to Enthalpy= 0.151028
Thermal correction to Gibbs Free Energy= 0.112434
Sum of electronic and ZPE= -183.741145
Sum of electronic and thermal Energies= -183.733914
Sum of electronic and thermal Enthalpies= -183.732969
Sum of electronic and thermal Free Energies= -183.771563

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 94.179 25.267 81.227
```

```
 C,0,-0.5623894626,-0.6348083861,-1.8169162967
 C,0,-0.5885594061,0.1890273784,-0.5180484541
 C,0,0.7442857303,0.1880848448,0.2519916581
 B,0,1.9648593461,0.9240975927,-0.3825714847
 H,0,-1.4988439582,-0.4591038623,-2.3633094619
 C,0,-0.3867781001,-2.1436171709,-1.6072339171
 H,0,-1.3974252345,-0.1884116333,0.1245557691
 H,0,0.618828939,0.7572211057,1.1984948599
 H,0,1.0117646199,-0.8187026951,0.6023979616
 H,0,3.0699692112,0.743339289,0.0453037422
 H,0,-0.8510356407,1.2259822375,-0.7668449643
 H,0,1.8232007244,1.7394512604,-1.2514018788
 H,0,0.2418512674,-0.2576963344,-2.4657761302
 H,0,-0.419028785,-2.6804663387,-2.5622165978
 H,0,0.5701618871,-2.3834980455,-1.1295400172
 H,0,-1.1842411381,-2.5483132423,-0.9713157883
```

Butene AM Product in

B3LYP/6-31G*
E(RB3LYP) = -183.884555027

Zero-point correction= 0.142658 (Hartree/Particle)
Thermal correction to Energy= 0.150026
Thermal correction to Enthalpy= 0.150971
Thermal correction to Gibbs Free Energy= 0.111975
Sum of electronic and ZPE= -183.741897
Sum of electronic and thermal Energies= -183.734529
Sum of electronic and thermal Enthalpies= -183.733584
Sum of electronic and thermal Free Energies= -183.772580

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 94.143 25.392 82.073
```

```
 C,0,-0.4540830725,-0.3519024606,-1.6416426004
 C,0,-0.5002604914,0.5256540927,-0.3795011185
 C,0,0.4115690366,0.0567597048,0.7677787291
 B,0,0.4965598386,0.9412038941,2.0492720212
 H,0,-1.2353043634,-0.0123923236,-2.3353298375
 H,0,-0.7158167228,-1.3854787533,-1.3719494105
 H,0,-1.5361150988,0.5603528628,-0.0153361457
 H,0,1.4696329048,0.0429540391,0.4305634525
 H,0,0.2152744252,-0.9966818498,1.0219601533
 H,0,0.1464636738,2.0886632825,2.0457846783
 H,0,-0.2423543981,1.5606086608,-0.6460912175
 H,0,0.97669639,0.5006947429,3.0555049627
 C,0,0.8950526293,-0.3416423281,-2.3694260673
 H,0,0.8577201208,-0.9523550068,-3.278720172
 H,0,1.1763522999,0.6772542697,-2.6645188066
 H,0,1.701393828,-0.7395788272,-1.7423626211
```

Butene M Product out

B3LYP/6-31G*
E(RB3LYP) = -183.882714970

Zero-point correction= 0.142904 (Hartree/Particle)
Thermal correction to Energy= 0.150308
Thermal correction to Enthalpy= 0.151252
Thermal correction to Gibbs Free Energy= 0.112687
Sum of electronic and ZPE= -183.739811
Sum of electronic and thermal Energies= -183.732407
Sum of electronic and thermal Enthalpies= -183.731463
Sum of electronic and thermal Free Energies= -183.770028

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 94.319 25.779 81.167
```

```
 C,0,-0.2310274656,0.1465969499,-1.3961442695
 C,0,-0.3264349594,0.0966749406,0.1390299704
 C,0,1.0053339503,0.5745667248,0.7962104601
 B,0,-0.5429507281,-1.3236843524,0.7609215402
 C,0,-1.5421170727,-0.2325352442,-2.0935844196
 H,0,0.0696740984,1.1557422968,-1.7152039991
 H,0,-1.1088866491,0.795341485,0.4721132253
 H,0,1.2517543942,1.5841341019,0.4443905146
 H,0,0.9400728723,0.6106834632,1.8896205605
 H,0,-0.1244475808,-2.3040027392,0.2090262816
 H,0,1.8464541097,-0.0773304152,0.5290750932
 H,0,-1.0412702509,-1.4544350061,1.8432160056
 H,0,0.567635157,-0.5313909663,-1.7292101333
 H,0,-1.4482131825,-0.1736669269,-3.1839397134
 H,0,-1.8444093497,-1.2577470149,-1.8433832002
 H,0,-2.3587583434,0.4359997032,-1.7934079165
```

Butene M Product anti

B3LYP/6-31G*
E(RB3LYP) = -183.881372674

Zero-point correction= 0.142849 (Hartree/Particle)
Thermal correction to Energy= 0.150340
Thermal correction to Enthalpy= 0.151284
Thermal correction to Gibbs Free Energy= 0.112369
Sum of electronic and ZPE= -183.738524
Sum of electronic and thermal Energies= -183.731033
Sum of electronic and thermal Enthalpies= -183.730089
Sum of electronic and thermal Free Energies= -183.769003

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 94.340 25.860 81.902
```

```
 C,0,-0.3571391405,0.3358751128,-1.4213799792
 C,0,-0.3499296133,0.0750353122,0.1000802985
 C,0,0.9818070097,0.5347344077,0.7717723769
 B,0,-0.4503769976,-1.4315775326,0.5125363827
 H,0,-1.2573458171,-0.1216535568,-1.8549259184
 C,0,-0.3235046316,1.8159715353,-1.8230141672
 H,0,-1.1558706549,0.6605689864,0.5685255924
 H,0,1.8478428544,0.0162082368,0.3415579992
 H,0,1.1377667467,1.6095278735,0.6273885484
 H,0,-0.0026893764,-2.2928136963,-0.193147084
 H,0,0.981616939,0.3498893776,1.8525675385
 H,0,-0.897327367,-1.7489822406,1.5786011126
 H,0,0.4931619561,-0.1908650075,-1.8776399438
 H,0,-0.3893554286,1.9285393194,-2.9114011274
 H,0,-1.1656873013,2.3640178076,-1.3817452956
 H,0,0.5993378226,2.3103030643,-1.4993903336
```

Butene M Product in

B3LYP/6-31G*

```
E(RB3LYP) = -183.882116733

Zero-point correction= 0.142595 (Hartree/Particle)
Thermal correction to Energy= 0.149862
Thermal correction to Enthalpy= 0.150807
Thermal correction to Gibbs Free Energy= 0.112551
Sum of electronic and ZPE= -183.739522
Sum of electronic and thermal Energies= -183.732254
Sum of electronic and thermal Enthalpies= -183.731310
Sum of electronic and thermal Free Energies= -183.769566

      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 94.040 25.743 80.515

 C,0,-0.154764738,0.2992143753,-1.3901527963
 C,0,-0.2162341265,0.080073613,0.1384460233
 C,0,0.9707717745,0.7077498905,0.8944548019
 B,0,-0.658036229,-1.3404876481,0.6075608126
 H,0,-1.1109070628,-0.0107865898,-1.8312355876
 H,0,-0.0470385637,1.3722886425,-1.605446948
 H,0,-1.1336195032,0.6130958663,0.4923657142
 H,0,1.0968293236,1.7676563453,0.6383782005
 H,0,0.8317270632,0.6356494175,1.9785409402
 H,0,-1.3223501812,-2.0411394835,-0.1040769475
 H,0,1.9122552713,0.2003835542,0.6552563878
 H,0,-0.4352731209,-1.7161560812,1.7242959654
 C,0,0.9800845869,-0.4666225807,-2.0817911686
 H,0,0.9682676699,-0.2951615093,-3.1642170534
 H,0,1.9650126426,-0.1595494093,-1.7121756332
 H,0,0.8836211934,-1.5475784028,-1.9165307111
```

Butene VTS out

```
B3LYP/6-31G*
E(RB3LYP) = -183.835431471

Zero-point correction= 0.136320 (Hartree/Particle)
Thermal correction to Energy= 0.144696
Thermal correction to Enthalpy= 0.145640
Thermal correction to Gibbs Free Energy= 0.102109
Sum of electronic and ZPE= -183.699111
Sum of electronic and thermal Energies= -183.690735
Sum of electronic and thermal Enthalpies= -183.689791
Sum of electronic and thermal Free Energies= -183.733322

      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 90.798 26.059 91.619

6 -1.480381 -0.220345 -0.332966
6 -0.783224 0.877401 0.423973
6 -0.042249 1.844357 -0.121225
5 3.362340 -0.557704 -0.031017
1 -2.565473 -0.138934 -0.168686
6 -1.020604 -1.623500 0.097809
1 -0.903938 0.852955 1.509108
1 0.104322 1.913052 -1.197836
1 0.435430 2.610907 0.482977
1 3.653757 -0.203596 1.070935
1 3.663915 0.109820 -0.973542
1 2.784443 -1.590120 -0.192332
1 -1.316513 -0.086713 -1.409747
1 -1.568672 -2.401014 -0.446138
1 0.049036 -1.761915 -0.094816
1 -1.189257 -1.783402 1.169618
```

Butene VTS anti

```
B3LYP/6-31G*
E(RB3LYP) = -183.835794638

Zero-point correction= 0.136313 (Hartree/Particle)
Thermal correction to Energy= 0.144690
Thermal correction to Enthalpy= 0.145634
Thermal correction to Gibbs Free Energy= 0.101929
Sum of electronic and ZPE= -183.699482
Sum of electronic and thermal Energies= -183.691105
Sum of electronic and thermal Enthalpies= -183.690160
Sum of electronic and thermal Free Energies= -183.733866

      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 90.794 26.062 91.985

6 1.140149 -0.545135 0.307986
6 0.250544 0.439637 -0.402059
6 -0.502654 1.369684 0.189874
5 -3.807913 -0.740581 -0.035571
6 2.619153 -0.409584 -0.093949
1 0.805053 -1.567183 0.076532
1 0.253265 0.378071 -1.492403
1 -0.530464 1.476826 1.272773
1 -1.111831 2.064444 -0.382359
1 -4.232245 0.217026 -0.608578
1 -3.819114 -0.775926 1.157648
1 -3.394265 -1.672563 -0.656538
1 1.035153 -0.420030 1.393145
1 3.233574 -1.161191 0.414351
1 2.750351 -0.545558 -1.174200
1 3.006939 0.581373 0.166372
```

Butene VTS in

```
B3LYP/6-31G*
E(RB3LYP) = -183.835603141

Zero-point correction= 0.136628 (Hartree/Particle)
Thermal correction to Energy= 0.145687
Thermal correction to Enthalpy= 0.146631
Thermal correction to Gibbs Free Energy= 0.100465
Sum of electronic and ZPE= -183.698975
Sum of electronic and thermal Energies= -183.689916
Sum of electronic and thermal Enthalpies= -183.688972
Sum of electronic and thermal Free Energies= -183.735138

      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 91.420 27.892 97.164

6 1.387279 -0.116589 -0.711085
6 0.316357 0.913295 -0.454237
6 -0.288962 1.166153 0.709566
5 -3.411007 -0.766963 -0.133942
```

```
1 2.268712 0.399848 -1.121674
1 1.045096 -0.775760 -1.523510
1 0.032222 1.504974 -1.326128
1 -0.050011 0.622519 1.620051
1 -1.041458 1.945023 0.796701
1 -3.570563 -0.709416 1.047866
1 -2.889927 -1.724925 -0.619875
1 -3.799847 0.119598 -0.832675
6 1.804831 -0.962735 0.493188
1 2.578856 -1.682873 0.207569
1 2.208636 -0.339134 1.298847
1 0.956282 -1.525777 0.897946
```

Hexene out of plane

B3LYP/6-31G*
E(RB3LYP) = -235.848697696

Zero-point correction= 0.166296 (Hartree/Particle)
Thermal correction to Energy= 0.174158
Thermal correction to Enthalpy= 0.175102
Thermal correction to Gibbs Free Energy= 0.134205
Sum of electronic and ZPE= -235.682402
Sum of electronic and thermal Energies= -235.674540
Sum of electronic and thermal Enthalpies= -235.673596
Sum of electronic and thermal Free Energies= -235.714493

```
     E     CV       S
  KCal/Mol Cal/Mol-K Cal/Mol-K
Total 109.286 27.305 86.075
```

```
 C,0,-0.2291275171,-0.5225229911,-1.373635487
 C,0,-0.2723273257,-0.5858593642,0.1288500625
 C,0,0.6824056838,-0.1438943636,0.9482881913
 H,0,-1.0782877447,0.0769430452,-1.7377937994
 C,0,-0.2921231791,-1.9114492072,-2.0372703673
 H,0,-1.1642069644,-1.0477443155,0.557481658
 H,0,1.5902729952,0.3215692551,0.5687505665
 H,0,0.5949944515,-0.2254375689,2.028582721
 H,0,0.6817906165,-0.000069795,-1.6952657098
 C,0,-0.2967460763,-1.8504852961,-3.5698664475
 H,0,0.5620644638,-2.5115226032,-1.6940522128
 H,0,-1.1936300792,-2.4387083177,-1.6916174352
 C,0,-0.3571900371,-3.2328046141,-4.227665558
 H,0,-1.1520811946,-1.2467054447,-3.9054454344
 H,0,0.6027136046,-1.3201963703,-3.913765073
 H,0,-0.3586519977,-3.1559873847,-5.3209825262
 H,0,0.503846725,-3.8475920003,-3.9379703915
 H,0,-1.2639644245,-3.7740366635,-3.9305357572
```

Hexene AM complex out

B3LYP/6-31G*
E(RB3LYP) = -262.480571718

Zero-point correction= 0.198507 (Hartree/Particle)
Thermal correction to Energy= 0.208388
Thermal correction to Enthalpy= 0.209332
Thermal correction to Gibbs Free Energy= 0.163849
Sum of electronic and ZPE= -262.282064
Sum of electronic and thermal Energies= -262.272184
Sum of electronic and thermal Enthalpies= -262.271240
Sum of electronic and thermal Free Energies= -262.316723

```
     E     CV       S
  KCal/Mol Cal/Mol-K Cal/Mol-K
Total 130.765 35.509 95.728
```

```
 C,0,-0.3647606651,-0.3761253023,-1.4686259173
 C,0,-0.3847771151,-0.4131110267,0.0399659306
 C,0,0.7612097022,-0.3911537928,0.820171694
 B,0,0.0408377639,1.2714542171,0.8597179304
 C,0,-1.6327408527,0.1744044345,-2.1328558033
 H,0,-0.2104635473,-1.4162747466,-1.7976874905
 H,0,-1.3118196301,-0.7511598445,0.4992770104
 H,0,1.7353630581,-0.3275325722,0.3460603869
 H,0,0.7479974304,-0.7934746033,1.8263379614
 H,0,-0.1811327874,1.4203369386,2.0292350823
 H,0,-1.0312950962,1.4048629903,0.2907507821
 H,0,0.8500831796,1.9339346021,0.270412934
 H,0,0.5178001497,0.1840917406,-1.8020516989
 C,0,-1.5941706939,0.0783010791,-3.6634448872
 H,0,-2.5091466273,-0.3736799457,-1.7562138909
 H,0,-1.7714899605,1.2211352899,-1.8340800438
 C,0,-2.8563363286,0.6349966704,-4.3295535245
 H,0,-0.7137811952,0.6197930322,-4.0373485198
 H,0,-1.4552675251,-0.9711883515,-3.9602977971
 H,0,-2.7994475335,0.5548906951,-5.4208843437
 H,0,-3.7508456903,0.0913284778,-4.0015769792
 H,0,-3.0021190356,1.692842018,-4.0796998161
```

Hexene AM complex anti

B3LYP/6-31G*
E(RB3LYP) = -262.480552373

Zero-point correction= 0.198733 (Hartree/Particle)
Thermal correction to Energy= 0.208671
Thermal correction to Enthalpy= 0.209615
Thermal correction to Gibbs Free Energy= 0.163918
Sum of electronic and ZPE= -262.281820
Sum of electronic and thermal Energies= -262.271882
Sum of electronic and thermal Enthalpies= -262.270938
Sum of electronic and thermal Free Energies= -262.316635

```
     E     CV       S
  KCal/Mol Cal/Mol-K Cal/Mol-K
Total 130.943 35.531 96.178
```

```
 C,0,-0.3602270069,-0.3780400177,-1.4319526845
 C,0,-0.3868546048,-0.3949145001,0.0759865279
 C,0,0.7452071344,-0.330469415,0.8604103896
 B,0,-0.036943336,1.357397751,0.8438765226
 H,0,-1.2491802659,0.1369154335,-1.8145971426
 C,0,-0.3068209057,-1.808267387,-2.0054654652
 H,0,-1.3129175784,-0.7315972159,0.5373874581
 H,0,1.7227862764,-0.2293686503,0.3998946422
 H,0,0.7334737668,-0.6768715266,1.8869912582
 H,0,-0.2260167602,1.5036842137,2.0197766107
 H,0,-1.1064551478,1.4732964195,0.276954442
 H,0,0.7933353551,1.9953547342,0.2575150259
 H,0,0.5120406975,0.1926815786,-1.7729262863
 C,0,-0.3113439577,-1.8304527491,-3.5397462984
 H,0,0.5926365742,-2.3168945297,-1.6321865891
 H,0,-1.1641456142,-2.3866315084,-1.6314121911
 C,0,-0.2571842888,-3.2480535392,-4.1178695403
 H,0,-1.2117183848,-1.3179105652,-3.9062097176
 H,0,0.5434442497,-1.2485733026,-3.9118438277
 H,0,-0.261307506,-3.2309225899,-5.2134422473
 H,0,0.6501946201,-3.773659266,-3.7962281326
 H,0,-1.1183313171,-3.8434723679,-3.7907027544
```

Hexene M complex out

```
B3LYP/6-31G*
E(RB3LYP) = -262.480572824

Zero-point correction= 0.198523 (Hartree/Particle)
Thermal correction to Energy= 0.208398
Thermal correction to Enthalpy= 0.209342
Thermal correction to Gibbs Free Energy= 0.163875
Sum of electronic and ZPE= -262.282050
Sum of electronic and thermal Energies= -262.272175
Sum of electronic and thermal Enthalpies= -262.271231
Sum of electronic and thermal Free Energies= -262.316698

     E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 130.772 35.510 95.693

 C,0,-0.3962998814,0.320691612,-1.4159351779
 C,0,-0.2763705859,0.3970555013,0.086458243
 C,0,0.935688073,0.3835027268,0.758215532
 B,0,0.2091740047,-1.2744239952,0.9053380683
 C,0,-1.7233340981,-0.2369520415,-1.9453736783
 H,0,-0.2677511373,1.3507153559,-1.7854860507
 H,0,-1.1550344874,0.7541499308,0.620381292
 H,0,1.8619574387,0.2966317797,0.1994196436
 H,0,1.0179837629,0.8085124266,1.7516668884
 H,0,0.9571105104,-1.9560023831,0.2595277835
 H,0,0.1026405581,-1.3904598039,2.0946898646
 H,0,-0.9132579397,-1.4138092938,0.4467453905
 H,0,0.4485151853,-0.2548471122,-1.8145982148
 C,0,-1.8262703953,-0.1799280278,-3.4749501566
 H,0,-1.8391416677,-1.2746280155,-1.6079538742
 H,0,-2.5581320152,0.3271689041,-1.5035719539
 C,0,-3.1485588401,-0.7416984596,-4.0066982148
 H,0,-1.7083431336,0.8603722989,-3.8106411132
 H,0,-0.9880906412,-0.7387862789,-3.9145005591
 H,0,-3.1925421843,-0.6898633669,-5.1003361348
 H,0,-3.2777096191,-1.7914476359,-3.7168208961
 H,0,-4.0049979069,-0.1815991219,-3.6116556816
```

Hexene M complex anti

```
B3LYP/6-31G*
E(RB3LYP) = -262.480552376

Zero-point correction= 0.198733 (Hartree/Particle)
Thermal correction to Energy= 0.208671
Thermal correction to Enthalpy= 0.209615
Thermal correction to Gibbs Free Energy= 0.163918
Sum of electronic and ZPE= -262.281820
Sum of electronic and thermal Energies= -262.271882
Sum of electronic and thermal Enthalpies= -262.270938
Sum of electronic and thermal Free Energies= -262.316635

     E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 130.943 35.531 96.178

 C,0,-0.2731431039,0.3127647331,-1.4356890497
 C,0,-0.4324472638,0.3696757617,0.0630628106
 C,0,0.6278426269,0.4430761013,0.9413783597
 B,0,0.0210417156,-1.3154213211,0.9218212566
 H,0,-1.0700643751,-0.300670242,-1.8720927897
 C,0,-0.3118686516,1.7241244982,-2.0552720898
 H,0,-1.423907333,0.6233499188,0.4325149155
 H,0,1.6469842892,0.4289467594,0.5684065745
 H,0,0.4925741001,0.8155081758,1.9498412353
 H,0,-0.9784708389,-1.5555653443,0.2722077558
 H,0,0.9581542325,-1.8819615323,0.4308845812
 H,0,-0.2540671941,-1.4467785374,2.0824049353
 H,0,0.6776838956,-0.1756553844,-1.6810657607
 C,0,-0.1850189521,1.7021330633,-3.5843097754
 H,0,-1.2515196135,2.2226048533,-1.7759872118
 H,0,0.4969059053,2.3320698272,-1.6270797582
 C,0,-0.2213455874,3.100882164,-4.2079548664
 H,0,0.7522955388,1.1999548995,-3.8617293821
 H,0,-0.9949078357,1.090394598,-4.0056469448
 H,0,-0.1283891552,3.0522996567,-5.2986376803
 H,0,-1.1623931076,3.6146006604,-3.9763108685
 H,0,0.5980947081,3.7250506907,-3.8311492472
```

Hexene AMTS out

```
B3LYP/6-31G*
E(RB3LYP) = -262.480534863

Zero-point correction= 0.198224 (Hartree/Particle)
Thermal correction to Energy= 0.207390
Thermal correction to Enthalpy= 0.208334
Thermal correction to Gibbs Free Energy= 0.164327
Sum of electronic and ZPE= -262.282311
Sum of electronic and thermal Energies= -262.273145
Sum of electronic and thermal Enthalpies= -262.272201
Sum of electronic and thermal Free Energies= -262.316208

     E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 130.139 33.424 92.621

 C,0,-0.3649953691,-0.3850131299,-1.4692803645
 C,0,-0.3813605319,-0.4158781411,0.040395255
 C,0,0.7800984408,-0.4264670858,0.8155284036
 B,0,0.1034863465,1.2082511002,0.8826407518
 C,0,-1.6287819183,0.1771031756,-2.1313664585
 H,0,-0.2205185226,-1.4275399304,-1.7942913342
 H,0,-1.3057054403,-0.7613689907,0.5000056433
 H,0,1.7469330419,-0.4011038135,0.3227688628
 H,0,0.7665107755,-0.8687685904,1.8050892806
 H,0,-0.1776472493,1.3761065551,2.0360619685
 H,0,-0.9631927052,1.3177603301,0.281247948
 H,0,0.8864551559,1.8985531182,0.2907520737
 H,0,0.5221638475,0.1661021495,-1.8057258777
 C,0,-1.5928463303,0.0854033808,-3.6623070008
 H,0,-2.5096250649,-0.3642126269,-1.7553195051
 H,0,-1.7587694233,1.224789078,-1.8302276765
 C,0,-2.8511701418,0.6543260526,-4.3253377829
 H,0,-0.7083683546,0.6206125851,-4.035527793
```

```
H,0,-1.4629387872,-0.9643377698,-3.9621377219
H,0,-2.7962028354,0.5766875996,-5.4169302021
H,0,-3.7497468286,0.1171586034,-3.9977684604
H,0,-2.9879201052,1.7126953504,-4.0725300099
```

Hexene AMTS anti

```
B3LYP/6-31G*
E(RB3LYP) = -262.480231959

Zero-point correction= 0.198486 (Hartree/Particle)
Thermal correction to Energy= 0.207622
Thermal correction to Enthalpy= 0.208566
Thermal correction to Gibbs Free Energy= 0.164637
Sum of electronic and ZPE= -262.281746
Sum of electronic and thermal Energies= -262.272610
Sum of electronic and thermal Enthalpies= -262.271666
Sum of electronic and thermal Free Energies= -262.315595

     E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 130.285 33.271 92.458

 C,0,-0.3800775629,-0.3644044607,-1.4411646563
 C,0,-0.4067225728,-0.3550189803,0.0698707765
 C,0,0.7576651977,-0.3582333449,0.8490056257
 B,0,0.0794883643,1.2625137215,0.8893708635
 H,0,-1.2768987181,0.1278008865,-1.8368421551
 C,0,-0.2972193849,-1.802444104,-1.9875117973
 H,0,-1.3281337864,-0.7058126073,0.5308474931
 H,0,1.7230489106,-0.3478694546,0.3522674935
 H,0,0.7438118155,-0.7934215433,1.8418062204
 H,0,-0.2454652673,1.4549225968,2.026806168
 H,0,-0.974927403,1.3301628047,0.2518378026
 H,0,0.8509600162,1.9616945246,0.2945065715
 H,0,0.4833661177,0.2161165017,-1.7883242474
 C,0,-0.2937761338,-1.8540427771,-3.5211251435
 H,0,0.609530701,-2.2870305288,-1.6006696733
 H,0,-1.1454086966,-2.3898565711,-1.6065430747
 C,0,-0.2096600859,-3.2812542096,-4.0712332894
 H,0,-1.2018032708,-1.3660214629,-3.9021149313
 H,0,0.5516357397,-1.2631564329,-3.9004780789
 H,0,-0.2084288755,-3.2857649965,-5.1669313825
 H,0,0.7058606889,-3.782968578,-3.7348620534
 H,0,-1.0609507936,-3.8865819837,-3.7364955316
```

Hexene MTS out

```
B3LYP/6-31G*
E(RB3LYP) = -262.476805871

Zero-point correction= 0.198521 (Hartree/Particle)
Thermal correction to Energy= 0.207574
Thermal correction to Enthalpy= 0.208518
Thermal correction to Gibbs Free Energy= 0.164880
Sum of electronic and ZPE= -262.278285
Sum of electronic and thermal Energies= -262.269232
Sum of electronic and thermal Enthalpies= -262.268288
Sum of electronic and thermal Free Energies= -262.311926

     E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 130.255 33.138 91.844

 C,0,-0.3411059745,0.3643463123,-1.4645292592
 C,0,-0.3444098569,0.3741765232,0.0547800297
 C,0,0.8423013054,0.4175468244,0.8091867716
 B,0,-0.0680911215,-1.1706074975,0.8289232189
 C,0,-1.5881267243,-0.2810747396,-2.0817703981
 H,0,-0.2662747001,1.4045104978,-1.8195205747
 H,0,-1.2218142294,0.8275093472,0.5108609032
 H,0,1.8077917138,0.3391181222,0.3159734545
 H,0,0.8628744854,0.8737349195,1.7941728457
 H,0,0.146610162,-1.9559495497,-0.0484979726
 H,0,0.9965796768,-1.1568198676,1.4710266361
 H,0,-0.9153518108,-1.3177522163,1.6608952388
 H,0,0.5589777129,-0.1520734407,-1.8232884273
 C,0,-1.5954969166,-0.2367285984,-3.6148784242
 H,0,-1.6576881583,-1.3221266619,-1.7408269298
 H,0,-2.4859754937,0.2279859895,-1.7002425282
 C,0,-2.8444213571,-0.8777723194,-4.2284562222
 H,0,-1.5175357872,0.807479171,-3.9506742508
 H,0,-0.6986916738,-0.7460006455,-3.9958592753
 H,0,-2.8204395382,-0.8336993605,-5.3233100412
 H,0,-2.9309904479,-1.932062509,-3.9386726622
 H,0,-3.7564602658,-0.3678693011,-3.8942691325
```

Hexene MTS anti

```
B3LYP/6-31G*
E(RB3LYP) = -262.476928679

Zero-point correction= 0.198501 (Hartree/Particle)
Thermal correction to Energy= 0.207598
Thermal correction to Enthalpy= 0.208542
Thermal correction to Gibbs Free Energy= 0.164745
Sum of electronic and ZPE= -262.278428
Sum of electronic and thermal Energies= -262.269331
Sum of electronic and thermal Enthalpies= -262.268387
Sum of electronic and thermal Free Energies= -262.312184

     E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 130.270 33.152 92.179

 C,0,-0.3617153221,0.3515808445,-1.434093677
 C,0,-0.364492235,0.3382705522,0.0844589755
 C,0,0.8275460684,0.3482393256,0.8337149139
 B,0,-0.120676204,-1.2194539585,0.8315511102
 H,0,-1.229854595,-0.2086053813,-1.8045029213
 C,0,-0.388797963,1.7757961155,-2.0142325663
 H,0,-1.2293916114,0.8063524771,0.5491136157
 H,0,1.7881904248,0.2580886603,0.3327077683
 H,0,0.8637870382,0.786491456,1.8263715082
 H,0,0.0680192874,-1.9959423265,-0.0594169185
 H,0,0.9514008975,-1.2367137837,1.4634349105
 H,0,-0.96258799,-1.3573480762,1.670515472
 H,0,0.5245237707,-0.1826258062,-1.8011004509
 C,0,-0.4277343523,1.8028086995,-3.5476032973
 H,0,-1.2637861876,2.3137344129,-1.6199261589
 H,0,0.4926098649,2.3303770873,-1.6598557196
 C,0,-0.4564121907,3.2215430411,-4.1251425021
 H,0,0.4457640737,1.2639824307,-3.941546551
 H,0,-1.3093798495,1.2466802563,-3.8963342154
 H,0,-0.4835840666,3.2068283722,-5.2206259376
```

```
H,0,-1.3384444636,3.7733809833,-3.777479322
H,0,0.4304036053,3.7915656179,-3.8216550364
```

Hexene AM Product out

B3LYP/6-31G*
E(RB3LYP) = -262.511491848

Zero-point correction= 0.200428 (Hartree/Particle)
Thermal correction to Energy= 0.210650
Thermal correction to Enthalpy= 0.211594
Thermal correction to Gibbs Free Energy= 0.164747
Sum of electronic and ZPE= -262.311064
Sum of electronic and thermal Energies= -262.300842
Sum of electronic and thermal Enthalpies= -262.299898
Sum of electronic and thermal Free Energies= -262.346744

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 132.185 34.856 98.597
```

```
C,0,-0.5334907687,-0.4527907969,-1.5298491473
C,0,-0.477446635,0.3347908225,-0.2117710334
C,0,0.8260237601,0.0589123939,0.5988591115
B,0,0.7508320031,0.9174299938,1.904628456
C,0,-1.8097301949,-0.1930332485,-2.3401774452
H,0,-0.4508747621,-1.527021817,-1.3097898441
H,0,-1.3564035237,0.0780831295,0.3960874985
H,0,1.6832220984,0.324059891,-0.0342103226
H,0,0.8787304642,-1.0188911467,0.8023919943
H,0,0.2199836206,0.4940029467,2.8931847421
H,0,-0.5594699907,1.4083115404,-0.4326307187
H,0,1.1504858393,2.048003903,1.9249993376
H,0,0.3459579655,-0.1969608772,-2.1384090611
C,0,-1.8682870886,-0.9783295393,-3.6569979734
H,0,-2.6880965278,-0.4467004589,-1.7278885842
H,0,-1.8915963452,0.8828008012,-2.5561658906
C,0,-3.1444756245,-0.7153572592,-4.462117814
H,0,-0.9904629918,-0.7249603428,-4.268651339
H,0,-1.7862645829,-2.0532953111,-3.4411003911
H,0,-3.1552507007,-1.2902921267,-5.3953310181
H,0,-4.0389164106,-0.9925011109,-3.8902999311
H,0,-3.2370026038,0.3460366134,-4.724206626
```

Hexene AM Product out twisted

B3LYP/6-31G*
E(RB3LYP) = -262.513174556

Zero-point correction= 0.199653 (Hartree/Particle)
Thermal correction to Energy= 0.209835
Thermal correction to Enthalpy= 0.210780
Thermal correction to Gibbs Free Energy= 0.164171
Sum of electronic and ZPE= -262.313522
Sum of electronic and thermal Energies= -262.303339
Sum of electronic and thermal Enthalpies= -262.302395
Sum of electronic and thermal Free Energies= -262.349004

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 131.674 35.161 98.096
```

```
C,0,-0.5960746333,-0.462559135,-1.5081953712
C,0,-0.5562304085,0.3560512731,-0.2113563777
C,0,0.6764916762,0.0746440567,0.6630430932
B,0,0.886970923,0.9331917633,1.9490667328
C,0,-1.832346348,-0.1868222189,-2.3731728903
H,0,-0.5565552721,-1.5343579678,-1.2613431229
H,0,-1.4668548934,0.1530641946,0.3704668242
H,0,1.6002864927,0.2643001567,0.0765971341
H,0,0.7543092146,-0.9986743592,0.9006006593
H,0,1.7245118222,0.6198048244,2.747594687
H,0,-0.5911238239,1.4275095376,-0.4541785172
H,0,0.2645298302,1.9390184837,2.1490929709
H,0,0.3122079675,-0.2551104768,-2.0944180043
C,0,-1.8704724602,-1.0051427191,-3.6703450323
H,0,-2.7400083372,-0.3954007804,-1.787045556
H,0,-1.8718860276,0.8847855074,-2.6200579268
C,0,-3.1087919225,-0.7232727196,-4.5273748215
H,0,-0.9642174905,-0.7962274423,-4.2566407407
H,0,-1.8318027478,-2.0758558557,-3.4238575327
H,0,-3.1061392804,-1.3222751782,-5.4452808294
H,0,-4.0308098671,-0.9561943842,-3.9803938705
H,0,-3.156763414,0.3331604397,-4.8192705081
```

Hexene AM Product out gauche

B3LYP/6-31G*
E(RB3LYP) = -262.512935487

Zero-point correction= 0.199927 (Hartree/Particle)
Thermal correction to Energy= 0.209891
Thermal correction to Enthalpy= 0.210835
Thermal correction to Gibbs Free Energy= 0.164940
Sum of electronic and ZPE= -262.313008
Sum of electronic and thermal Energies= -262.303045
Sum of electronic and thermal Enthalpies= -262.302101
Sum of electronic and thermal Free Energies= -262.347996

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 131.708 35.003 96.595
```

```
C,0,-0.1378092593,-0.0415812204,-1.0238276961
C,0,-0.3656469647,-0.5028429668,0.4219068208
C,0,0.8695434884,-0.2935463252,1.3138760663
B,0,1.2763522724,1.1669554191,1.6820544543
C,0,-1.3572134991,-0.2380183754,-1.9326274758
H,0,0.7216268794,-0.5845098377,-1.4457601813
H,0,-0.6563698789,-1.5631851948,0.4171841718
H,0,1.7279584213,-0.8724113481,0.9407773411
H,0,0.6824625475,-0.7315069604,2.3178632302
H,0,0.4989979909,2.0761712108,1.5869619472
H,0,-1.2170797039,0.0483012415,0.8453685164
H,0,2.3587715966,1.3968186959,2.1429597626
H,0,0.1463718775,1.0221011654,-1.0223409596
C,0,-1.1269627262,0.2212944149,-3.3782386778
H,0,-1.6434911453,-1.3005489679,-1.9316068388
H,0,-2.2153450486,0.3070268517,-1.5117168663
C,0,-2.350677452,0.0218669393,-4.2782822914
H,0,-0.8408900549,1.2827827916,-3.3792803937
H,0,-0.2701508999,-0.3241106868,-3.7990711704
H,0,-2.154292886,0.3589894184,-5.3025385831
H,0,-2.6401388476,-1.0352878711,-4.3253529213
H,0,-3.2147637075,0.5838166059,-3.9026632548
```

Hexene AM Product anti

B3LYP/6-31G*
E(RB3LYP) = -262.510212130

Zero-point correction= 0.200470 (Hartree/Particle)
Thermal correction to Energy= 0.210694
Thermal correction to Enthalpy= 0.211639
Thermal correction to Gibbs Free Energy= 0.164074
Sum of electronic and ZPE= -262.309742
Sum of electronic and thermal Energies= -262.299518
Sum of electronic and thermal Enthalpies= -262.298574
Sum of electronic and thermal Free Energies= -262.346139

```
     E      CV        S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 132.213 34.793 100.109
```

```
6 0.125155 1.011974 -0.108236
6 1.558147 0.670588 0.341220
6 2.186104 -0.542530 -0.411459
5 3.662955 -0.686723 0.084645
1 -0.177902 1.944089 0.390556
6 -0.919398 -0.070741 0.189641
1 1.559737 0.472886 1.422049
1 2.110678 -0.359325 -1.491351
1 1.593854 -1.437449 -0.179673
1 3.899442 -1.200677 1.142435
1 2.184420 1.560909 0.191239
1 4.582804 -0.227619 -0.532482
1 0.129029 1.229430 -1.186244
6 -2.344459 0.339566 -0.206076
1 -0.658245 -0.998971 -0.336840
1 -0.896735 -0.311567 1.263393
6 -3.387393 -0.743737 0.087207
1 -2.616198 1.263382 0.324126
1 -2.364819 0.587283 -1.276977
1 -4.393097 -0.421901 -0.206405
1 -3.161708 -1.669137 -0.457081
1 -3.414968 -0.988433 1.156250
```

Hexene AM Product anti twisted

B3LYP/6-31G*
E(RB3LYP) = -262.511949830

Zero-point correction= 0.199889 (Hartree/Particle)
Thermal correction to Energy= 0.209922
Thermal correction to Enthalpy= 0.210867
Thermal correction to Gibbs Free Energy= 0.164608
Sum of electronic and ZPE= -262.312061
Sum of electronic and thermal Energies= -262.302027
Sum of electronic and thermal Enthalpies= -262.301083
Sum of electronic and thermal Free Energies= -262.347342

```
     E      CV        S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 131.728 35.056 97.359
```

```
 C,0,-0.5958185086,-0.4798390738,-1.774306171
 C,0,-0.6595561644,0.2616536888,-0.4283145967
 C,0,0.6599679712,0.2845761226,0.3632029525
 B,0,0.6588384715,0.9410260896,1.7777157894
 H,0,-1.536419398,-0.3073258348,-2.3169533547
 C,0,-0.3631031768,-1.9926872805,-1.6630879433
 H,0,-1.447818579,-0.1825840822,0.1959961891
 H,0,1.4839429931,0.6731126471,-0.2557542515
 H,0,0.9837637544,-0.7525079866,0.5928023557
 H,0,-0.3599580273,1.1352435666,2.3807071355
 H,0,-0.975830079,1.2968906949,-0.6152266834
 H,0,1.6945143399,1.2098936387,2.3182949174
 H,0,0.1973039275,-0.0355671406,-2.394811239
 C,0,-0.3603430716,-2.708395219,-3.0204949444
 H,0,0.5923598084,-2.1901091893,-1.1563298898
 H,0,-1.1434569783,-2.4326539588,-1.02343612
 C,0,-0.1286733767,-4.2186692474,-2.9082352204
 H,0,-1.3164742901,-2.5212153116,-3.5297464239
 H,0,0.415815372,-2.2654889715,-3.6608665979
 H,0,-0.1337578417,-4.6993377063,-3.8932539147
 H,0,0.8370124367,-4.43792289,-2.4359678231
 H,0,-0.9082965831,-4.6953955557,-2.3011921658
```

Hexene M Product out

B3LYP/6-31G*
E(RB3LYP) = -262.510287083

Zero-point correction= 0.200137 (Hartree/Particle)
Thermal correction to Energy= 0.210209
Thermal correction to Enthalpy= 0.211153
Thermal correction to Gibbs Free Energy= 0.165402
Sum of electronic and ZPE= -262.310150
Sum of electronic and thermal Energies= -262.300078
Sum of electronic and thermal Enthalpies= -262.299134
Sum of electronic and thermal Free Energies= -262.344885

```
     E      CV        S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 131.908 35.437 96.291
```

```
 C,0,-0.2083390633,0.1546515203,-1.3511170921
 C,0,-0.3012414715,0.102711348,0.1840642456
 C,0,1.0328123194,0.5761826616,0.8397054425
 B,0,-0.5196955154,-1.3192375278,0.8013955981
 C,0,-1.5194912198,-0.2240813681,-2.0528109537
 H,0,0.0916226352,1.1646543063,-1.6713133369
 H,0,-1.0811713702,0.8023986626,0.5201827773
 H,0,1.2797584777,1.5868922655,0.49165452
 H,0,0.9699814107,0.6081105954,1.933480264
 H,0,-0.0867979725,-2.2955037593,0.2533826973
 H,0,1.8721677989,-0.0759742184,0.5681195291
 H,0,-1.0330827828,-1.4554918937,1.8759653252
 H,0,0.5900855405,-0.5233698973,-1.6874813364
 C,0,-1.4399762616,-0.1589150074,-3.5831638505
 H,0,-1.8139154348,-1.2415851788,-1.7511503322
 H,0,-2.3242783454,0.4403717536,-1.7033640955
 C,0,-2.7538757899,-0.5371512252,-4.2744940108
 H,0,-1.1455960385,0.8561935796,-3.8858749916
 H,0,-0.6374085055,-0.8242762111,-3.9318577953
 H,0,-2.6636965385,-0.479937124,-5.3652892675
 H,0,-3.0562316842,-1.5602193902,-4.0186991054
 H,0,-3.5682871883,0.1328091086,-3.9720682311
```

Hexene M Product anti

B3LYP/6-31G*

```
E(RB3LYP) = -262.508738171

Zero-point correction= 0.200076 (Hartree/Particle)
Thermal correction to Energy= 0.210248
Thermal correction to Enthalpy= 0.211192
Thermal correction to Gibbs Free Energy= 0.165011
Sum of electronic and ZPE= -262.308662
Sum of electronic and thermal Energies= -262.298490
Sum of electronic and thermal Enthalpies= -262.297546
Sum of electronic and thermal Free Energies= -262.343727

      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 131.932 35.530 97.195

 C,0,-0.3490356871,0.3070417162,-1.3802250331
 C,0,-0.3428622786,0.0450423044,0.141141667
 C,0,0.990450478,0.4962023586,0.8157680754
 B,0,-0.4504170363,-1.4619403894,0.5502331173
 H,0,-1.2471369925,-0.1546230037,-1.8164302154
 C,0,-0.3229573221,1.7868712206,-1.7902870388
 H,0,-1.1466160856,0.6323070639,0.6107788239
 H,0,1.854924673,-0.0210334887,0.3810043675
 H,0,1.1488591691,1.5717009235,0.680274076
 H,0,-0.0003925783,-2.3232557755,-0.1538516323
 H,0,0.9900745764,0.3024810616,1.8950181049
 H,0,-0.9056291997,-1.7800503635,1.6126598468
 H,0,0.5044834132,-0.2158585055,-1.8369039988
 C,0,-0.4160163316,1.9951689244,-3.307943572
 H,0,-1.1571263731,2.3131599385,-1.3016503882
 H,0,0.5953513673,2.2656905412,-1.422930833
 C,0,-0.3904144159,3.4709596896,-3.7188718693
 H,0,0.4130838986,1.4643926145,-3.7973590269
 H,0,-1.3377571143,1.5267627418,-3.6816658644
 H,0,-0.4589778612,3.5853529712,-4.8067836394
 H,0,-1.2280401852,4.0212682659,-3.2726601562
 H,0,0.536217886,3.9580301905,-3.3906728111
```

Octene

```
B3LYP/6-31G*
E(RB3LYP) = -314.476138238

Zero-point correction= 0.223511 (Hartree/Particle)
Thermal correction to Energy= 0.234065
Thermal correction to Enthalpy= 0.235009
Thermal correction to Gibbs Free Energy= 0.186900
Sum of electronic and ZPE= -314.252627
Sum of electronic and thermal Energies= -314.242073
Sum of electronic and thermal Enthalpies= -314.241129
Sum of electronic and thermal Free Energies= -314.289238

      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 146.878 36.963 101.254

 C,0,-0.2944469466,-0.6765622381,-1.491431893
 C,0,-0.3056153614,-0.6697562418,0.0129584671
 C,0,0.604107414,-0.0800169584,0.7894616744
 C,0,-1.5400923394,-0.0094626394,-2.1057967252
 H,0,-0.2421865316,-1.7162575503,-1.8509763859
 H,0,-1.1454758668,-1.1864496258,0.4822040688
 H,0,1.4574555639,0.4488285237,0.3686831862
 H,0,0.5359193032,-0.1042462522,1.8739422744
 H,0,0.6117045356,-0.1747429152,-1.8563820327
 C,0,-1.5564584304,-0.055120728,-3.6388726931
 H,0,-2.4437269788,-0.5013038439,-1.7166492928
 H,0,-1.5908342811,1.0341141067,-1.7657926863
 C,0,-2.7931916104,0.6090883697,-4.257322392
 H,0,-0.6500563129,0.4334837509,-4.0263645712
 H,0,-1.5016776407,-1.1025467839,-3.9714649366
 C,0,-2.81105702,0.5650765883,-5.7907868054
 H,0,-3.700183139,0.120665506,-3.870579331
 H,0,-2.8482639111,1.6563491407,-3.9245079072
 C,0,-4.0491753901,1.2305243092,-6.4003780249
 H,0,-1.9051526147,1.0535511983,-6.1772445535
 H,0,-2.756591726,-0.4813728185,-6.1234058644
 H,0,-4.0320162776,1.1825466171,-7.4952384491
 H,0,-4.9702634765,0.7413576081,-6.0599997636
 H,0,-4.1123409616,2.287732877,-6.114296363
```

Octene AM Complex out

```
B3LYP/6-31G*
E(RB3LYP) = -341.108029610

Zero-point correction= 0.255697 (Hartree/Particle)
Thermal correction to Energy= 0.268292
Thermal correction to Enthalpy= 0.269236
Thermal correction to Gibbs Free Energy= 0.216499
Sum of electronic and ZPE= -340.852332
Sum of electronic and thermal Energies= -340.839738
Sum of electronic and thermal Enthalpies= -340.838793
Sum of electronic and thermal Free Energies= -340.891530

      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 168.356 45.173 110.994

 C,0,-0.3409097785,-0.3812106956,-1.4251300466
 C,0,-0.3585485866,-0.4184004183,0.0834555439
 C,0,0.7886958276,-0.3954388892,0.8618713693
 B,0,0.0665631672,1.2660711219,0.9040796019
 C,0,-1.6110366806,0.1672945919,-2.0872565176
 H,0,-0.1853378487,-1.4210827529,-1.7544090817
 H,0,-1.2845138664,-0.7574602848,0.5442185545
 H,0,1.7619526508,-0.330714643,0.3860510858
 H,0,0.777625841,-0.7984774774,1.8677768938
 H,0,-0.1554423529,1.4137873715,2.0737462652
 H,0,-1.0057114752,1.3987587391,0.3351227293
 H,0,0.8747560516,1.9302727214,0.3152348041
 H,0,0.5402376212,0.1804571039,-1.759791316
 C,0,-1.5749208737,0.0694434731,-3.6177683542
 H,0,-2.4854267829,-0.3816221236,-1.7074962811
 H,0,-1.7499206011,1.2138736425,-1.7883593025
 C,0,-2.836921708,0.6232236657,-4.2914027179
 H,0,-0.6949710618,0.6111090587,-3.9950098508
 H,0,-1.4357205043,-0.9808768815,-3.9147927319
 C,0,-2.8049819447,0.5290541861,-5.8223662353
 H,0,-3.7179639544,0.0832052529,-3.9135619419
 H,0,-2.9753181947,1.6735660651,-3.9955000807
 C,0,-4.0685032286,1.0843662361,-6.4877985124
 H,0,-1.9249215169,1.0690367031,-6.1996194412
 H,0,-2.6665979269,-0.5208181832,-6.1178079998
```

```
H,0,-4.0147173002,1.0029173054,-7.579383421
H,0,-4.962307104,0.5412661299,-6.1567628509
H,0,-4.2145848682,2.142768981,-6.2394001642
```

Octene AM Complex anti

```
B3LYP/6-31G*
E(RB3LYP) = -341.108015153

Zero-point correction= 0.255923 (Hartree/Particle)
Thermal correction to Energy= 0.268573
Thermal correction to Enthalpy= 0.269517
Thermal correction to Gibbs Free Energy= 0.216575
Sum of electronic and ZPE= -340.852092
Sum of electronic and thermal Energies= -340.839443
Sum of electronic and thermal Enthalpies= -340.838498
Sum of electronic and thermal Free Energies= -340.891440

     E      CV       S
  KCal/Mol Cal/Mol-K Cal/Mol-K
Total 168.532 45.194 111.426

 C,0,-0.3566475059,-0.3367819557,-1.3734206055
 C,0,-0.3856904452,-0.3545344416,0.1344353966
 C,0,0.7448659225,-0.2909276199,0.9210986852
 B,0,-0.0363657235,1.3973606109,0.9040910086
 H,0,-1.2444589609,0.179334202,-1.7570719124
 C,0,-0.3042243709,-1.7670927061,-1.9472984608
 H,0,-1.3127220712,-0.6910680485,0.5940216725
 H,0,1.7233956555,-0.1899274025,0.4625681202
 H,0,0.7310317917,-0.6379955817,1.9474308494
 H,0,-0.2274294613,1.5429474539,2.0797873134
 H,0,-1.1048704466,1.5140051539,0.3354447839
 H,0,0.795217245,2.0354685941,0.3197455266
 H,0,0.5167551809,0.2330317295,-1.7129224248
 C,0,-0.3098094324,-1.7883865227,-3.4814123722
 H,0,0.5950746026,-2.2755726358,-1.5737746138
 H,0,-1.1616107186,-2.3443948935,-1.572051821
 C,0,-0.257275322,-3.2041274839,-4.069719448
 H,0,-1.2105396212,-1.2746555564,-3.848038899
 H,0,0.545411808,-1.2064715934,-3.8551180831
 C,0,-0.2630495402,-3.228992517,-5.6037548106
 H,0,0.6435980898,-3.7178898098,-3.7024213577
 H,0,-1.1124197064,-3.786073842,-3.6947624751
 C,0,-0.2105682513,-4.6455774515,-6.1845552416
 H,0,-1.1633544722,-2.7156945803,-5.9703356117
 H,0,0.5914464358,-2.6475274889,-5.9780886283
 H,0,-0.216096325,-4.6290884675,-7.2803140187
 H,0,0.6967462282,-5.1723663163,-5.8640762983
 H,0,-1.0713415852,-5.2411198294,-5.8561732737
```

Octene M Complex out

```
B3LYP/6-31G*
E(RB3LYP) = -341.108030745

Zero-point correction= 0.255712 (Hartree/Particle)
Thermal correction to Energy= 0.268300
Thermal correction to Enthalpy= 0.269244
Thermal correction to Gibbs Free Energy= 0.216529
Sum of electronic and ZPE= -340.852319
Sum of electronic and thermal Energies= -340.839731
Sum of electronic and thermal Enthalpies= -340.838786
Sum of electronic and thermal Free Energies= -340.891502

     E      CV       S
  KCal/Mol Cal/Mol-K Cal/Mol-K
Total 168.361 45.174 110.948

 C,0,-0.3951456916,0.3196005674,-1.4253373012
 C,0,-0.2788673509,0.3972254441,0.0772945911
 C,0,0.9315638366,0.3843006791,0.7520157135
 B,0,0.2048318284,-1.2734058948,0.8998985561
 C,0,-1.7215931493,-0.2370824992,-1.957543293
 H,0,-0.2645582607,1.3492223908,-1.795228646
 H,0,-1.1588391855,0.7546353888,0.6088635136
 H,0,1.8591108769,0.2969357145,0.1954078276
 H,0,1.0115770322,0.8104657749,1.7451649376
 H,0,0.9551212682,-1.9559449827,0.2578037172
 H,0,0.0943954061,-1.3877269028,2.089075696
 H,0,-0.9158871068,-1.4137802499,0.4374682036
 H,0,0.4501917268,-0.2569800247,-1.8213536557
 C,0,-1.8208733701,-0.1789779928,-3.4873626712
 H,0,-1.8382035928,-1.2745477954,-1.6200772831
 H,0,-2.5562839966,0.3275844361,-1.5165115781
 C,0,-3.1401918559,-0.741108003,-4.0317980733
 H,0,-1.7036458099,0.8623673831,-3.8229965039
 H,0,-0.9803626115,-0.7363811333,-3.9265424885
 C,0,-3.2438888313,-0.6868452123,-5.5614425416
 H,0,-3.2567100427,-1.7824636333,-3.6971368932
 H,0,-3.9817021097,-0.1853301614,-3.5917293639
 C,0,-4.5640228729,-1.2502761775,-6.0972228751
 H,0,-3.1273862827,0.3540481836,-5.8956263492
 H,0,-2.4033649802,-1.2426312919,-6.0008394489
 H,0,-4.6071960485,-1.1974776005,-7.1910568033
 H,0,-4.6925092793,-2.3008907202,-5.8091016724
 H,0,-5.4224305467,-0.6926406867,-5.7026803144
```

Octene M Complex anti

```
B3LYP/6-31G*
E(RB3LYP) = -341.108015150

Zero-point correction= 0.255923 (Hartree/Particle)
Thermal correction to Energy= 0.268573
Thermal correction to Enthalpy= 0.269517
Thermal correction to Gibbs Free Energy= 0.216575
Sum of electronic and ZPE= -340.852092
Sum of electronic and thermal Energies= -340.839443
Sum of electronic and thermal Enthalpies= -340.838498
Sum of electronic and thermal Free Energies= -340.891440

     E      CV       S
  KCal/Mol Cal/Mol-K Cal/Mol-K
Total 168.532 45.194 111.426

 C,0,-0.2705948779,0.2737625512,-1.3755712887
 C,0,-0.4323880307,0.3313072471,0.1228610255
 C,0,0.626123313,0.4054302527,1.0032612553
 B,0,0.020274724,-1.3533928962,0.9833261989
 H,0,-1.066189395,-0.3407244636,-1.8128455563
 C,0,-0.3102366512,1.6850932439,-1.9956086568
 H,0,-1.4246340874,0.5847063683,0.4904185626
 H,0,1.6460284862,0.3915291509,0.6323539839
 H,0,0.4887251403,0.7783094698,2.0112732935
```

```
H,0,-0.9780094836,-1.5942071611,0.3320913166
H,0,0.9585165899,-1.9199158214,0.4945309684
H,0,-0.256879014,-1.4842504933,2.1435112662
H,0,0.6811321607,-0.2136170587,-1.619409313
C,0,-0.1843516235,1.6621043955,-3.524534846
H,0,-1.2499402676,2.182565101,-1.7151125875
H,0,0.4983821092,2.2928683922,-1.5671747761
C,0,-0.2211857561,3.0585476307,-4.1583934965
H,0,0.7535284225,1.1598698865,-3.8035074745
H,0,-0.994469343,1.0491574123,-3.9460257969
C,0,-0.0958498075,3.0390926158,-5.6873894483
H,0,-1.1591013436,3.5608570543,-3.8781689766
H,0,0.589012871,3.671559221,-3.7362235738
C,0,-0.1334736869,4.4365831004,-6.3138089252
H,0,0.8413212849,2.5372138255,-5.9670655553
H,0,-0.9055931303,2.4266207658,-6.1088146071
H,0,-0.0419483314,4.388485039,-7.4048173441
H,0,-1.0742608844,4.9505156964,-6.0809380901
H,0,0.6858956125,5.061873474,-5.938204558
```

Octene AMTS out

B3LYP/6-31G*
E(RB3LYP) = -341.107994089

Zero-point correction= 0.255429 (Hartree/Particle)
Thermal correction to Energy= 0.267303
Thermal correction to Enthalpy= 0.268247
Thermal correction to Gibbs Free Energy= 0.217014
Sum of electronic and ZPE= -340.852565
Sum of electronic and thermal Energies= -340.840691
Sum of electronic and thermal Enthalpies= -340.839747
Sum of electronic and thermal Free Energies= -340.890980

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 167.735 43.083 107.829
```

```
C,0,-0.3512903699,-0.387725426,-1.4485208456
C,0,-0.3679076734,-0.419227746,0.0611131546
C,0,0.793237393,-0.4286266476,0.8366208399
B,0,0.1142680129,1.205314321,0.9049429316
C,0,-1.6158583271,0.1731952852,-2.1104809191
H,0,-0.20553011,-1.4299724384,-1.7737904679
H,0,-1.2920216098,-0.765726556,0.5204412899
H,0,1.7602021454,-0.4015527575,0.3441986887
H,0,0.7799761102,-0.8713786534,1.8259852474
H,0,-0.1681440054,1.371559564,2.0583020445
H,0,-0.9515747856,1.3145023461,0.3021134494
H,0,0.8974571647,1.89707968,0.3150010832
H,0,0.5352986228,0.1645069845,-1.7845617904
C,0,-1.5803384631,0.079230608,-3.6412825395
H,0,-2.495668244,-0.3681347202,-1.7324029876
H,0,-1.7454766089,1.2209139254,-1.8097282574
C,0,-2.8368974084,0.6464508166,-4.3138075025
H,0,-0.6951626008,0.6135322577,-4.0167308329
H,0,-1.4509219441,-0.9716195843,-3.940680179
C,0,-2.8057018068,0.5554147262,-5.8448950681
H,0,-3.7231032973,0.1140062048,-3.9372103431
H,0,-2.9653783399,1.6974500371,-4.0155768246
C,0,-4.0636431566,1.1243979858,-6.5089825897
H,0,-1.9203935625,1.0877082779,-6.2208449803
H,0,-2.6774684476,-0.4951101552,-6.1426294457
H,0,-4.0107545413,1.0448618366,-7.6007514157
H,0,-4.962786023,0.5893602272,-6.1791991121
H,0,-4.1995351234,2.1836626004,-6.258360628
```

Octene AMTS anti

B3LYP/6-31G*
E(RB3LYP) = -341.107694481

Zero-point correction= 0.255659 (Hartree/Particle)
Thermal correction to Energy= 0.267511
Thermal correction to Enthalpy= 0.268455
Thermal correction to Gibbs Free Energy= 0.217265
Sum of electronic and ZPE= -340.852036
Sum of electronic and thermal Energies= -340.840183
Sum of electronic and thermal Enthalpies= -340.839239
Sum of electronic and thermal Free Energies= -340.890429

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 167.866 42.939 107.739
```

```
C,0,-0.3784836878,-0.3222277427,-1.3830329686
C,0,-0.4059888601,-0.3142552621,0.1279726239
C,0,0.7579666393,-0.3181332336,0.9078947119
B,0,0.0799296256,1.3023949207,0.9495059306
H,0,-1.27450292,0.1713115836,-1.7787915557
C,0,-0.297048037,-1.7602093839,-1.9303017329
H,0,-1.3276115215,-0.6656854555,0.5880619995
H,0,1.72363377,-0.3074001751,0.4117059485
H,0,0.7435090452,-0.7545891355,1.9001385443
H,0,-0.2454979947,1.4939464107,2.0869763593
H,0,-0.9744640575,1.3701843699,0.3118269038
H,0,0.8512575119,2.0023729674,0.3553846982
H,0,0.485791929,0.2576284002,-1.7291464697
C,0,-0.2932274918,-1.8102567391,-3.4639147764
H,0,0.6088617712,-2.2454546294,-1.5426618984
H,0,-1.1460542549,-2.3461380553,-1.5492173425
C,0,-0.2116674206,-3.2356559103,-4.0246471993
H,0,-1.2006844467,-1.3198722496,-3.8457912408
H,0,0.5537093904,-1.2201256904,-3.8436042104
C,0,-0.2071347126,-3.2900185523,-5.5578315183
H,0,0.6957858921,-3.7260570451,-3.64192059
H,0,-1.0585267349,-3.8257808161,-3.643638515
C,0,-0.1256280161,-4.7164671169,-6.1107556815
H,0,-1.1140138869,-2.8000831964,-5.9399336793
H,0,0.6391934996,-2.7005885384,-5.938245771
H,0,-0.1244650222,-4.7212797535,-7.2066353538
H,0,0.7888937995,-5.2205132195,-5.7745917682
H,0,-0.9776778083,-5.3209427518,-5.7760014481
```

Octene MTS out

B3LYP/6-31G*
E(RB3LYP) = -341.104247031

Zero-point correction= 0.255715 (Hartree/Particle)
Thermal correction to Energy= 0.267478
Thermal correction to Enthalpy= 0.268422
Thermal correction to Gibbs Free Energy= 0.217555
Sum of electronic and ZPE= -340.848532
Sum of electronic and thermal Energies= -340.836769

```
Sum of electronic and thermal Enthalpies= -340.835825
Sum of electronic and thermal Free Energies= -340.886692

      E      CV        S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 167.845 42.801 107.059

 C,0,-0.3290877325,0.369830234,-1.4432260285
 C,0,-0.3319898216,0.378544129,0.0761102862
 C,0,0.85489748,0.4227702606,0.8301813984
 B,0,-0.0540801004,-1.1661802418,0.8497603872
 C,0,-1.5754560679,-0.2772433945,-2.0604422299
 H,0,-0.2560393556,1.4103993075,-1.7973045905
 H,0,-1.2097020171,0.830791902,0.5326862197
 H,0,1.8202995162,0.3455225213,0.3365979204
 H,0,0.8753461427,0.8784667079,1.8153971036
 H,0,0.1606027784,-1.9513877508,-0.0278364057
 H,0,1.0110140545,-1.1518547676,1.491151981
 H,0,-0.9007473077,-1.3141217567,1.6822057626
 H,0,0.5717548257,-0.1447284791,-1.8026688724
 C,0,-1.5832092627,-0.230908316,-3.5935332558
 H,0,-1.6423459229,-1.3184587685,-1.7198840786
 H,0,-2.4733781665,0.2299880871,-1.6770535685
 C,0,-2.8293915245,-0.8727876299,-4.2166025086
 H,0,-1.507534922,0.8144590218,-3.9291401104
 H,0,-0.6848736053,-0.7378988415,-3.9764922623
 C,0,-2.8393777494,-0.8321802105,-5.7501280353
 H,0,-2.9057601619,-1.9177294157,-3.8812057991
 H,0,-3.7286492805,-0.3662028983,-3.8351526051
 C,0,-4.0874889379,-1.4749331625,-6.3636916137
 H,0,-2.7629696489,0.2121848063,-6.0852376161
 H,0,-1.9415054026,-1.3393521705,-6.131243984
 H,0,-4.0644938824,-1.4304485806,-7.4586171087
 H,0,-4.1725891094,-2.5298249062,-6.0748821018
 H,0,-5.0003708177,-0.966922687,-6.0288082839
```

Octene MTS anti

```
B3LYP/6-31G*
E(RB3LYP) = -341.104376064

Zero-point correction= 0.255688 (Hartree/Particle)
Thermal correction to Energy= 0.267494
Thermal correction to Enthalpy= 0.268438
Thermal correction to Gibbs Free Energy= 0.217426
Sum of electronic and ZPE= -340.848689
Sum of electronic and thermal Energies= -340.836883
Sum of electronic and thermal Enthalpies= -340.835938
Sum of electronic and thermal Free Energies= -340.886950

      E      CV        S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 167.855 42.813 107.362

 C,0,-0.3569607338,0.311338275,-1.3750662513
 C,0,-0.3603695449,0.2978089387,0.143492401
 C,0,0.8314625812,0.3079653877,0.8931704089
 B,0,-0.1165406903,-1.2598004332,0.8904718877
 H,0,-1.2244695983,-0.2493993313,-1.7460327907
 C,0,-0.3848185497,1.7359451694,-1.9546777688
 H,0,-1.2254389791,0.7658522764,0.6078865121
 H,0,1.7922619312,0.2180269146,0.3924128769
 H,0,0.8673570938,0.7463132743,1.8858017645
 H,0,0.0729133603,-2.0363532867,-0.0002971088
 H,0,0.9553060105,-1.2765685646,1.5228873667
 H,0,-0.9586520714,-1.3980340216,1.7291934848
 H,0,0.529849028,-0.2221502687,-1.7416766654
 C,0,-0.422092335,1.7628977356,-3.4880777163
 H,0,-1.2607079464,2.2721114228,-1.5603142763
 H,0,0.4955788557,2.2905940915,-1.5983012661
 C,0,-0.4522428327,3.179927265,-4.0747310939
 H,0,0.4529581972,1.2248896469,-3.88218437
 H,0,-1.3030342733,1.2051856057,-3.8388677441
 C,0,-0.4895671063,3.2082547874,-5.6081308313
 H,0,-1.3276773611,3.7179653598,-3.6812021023
 H,0,0.4286162186,3.7382168605,-3.7236493394
 C,0,-0.5190912167,4.6269195482,-6.1859159873
 H,0,0.3850738156,2.6707049467,-6.0013824958
 H,0,-1.3701823033,2.6514608562,-5.9588592878
 H,0,-0.5456115936,4.6129570555,-7.2814984876
 H,0,-1.4016221493,5.178262203,-5.8385204518
 H,0,0.3670641931,5.197596285,-5.8815636674
```

Octene AM Product out

```
B3LYP/6-31G*
E(RB3LYP) = -341.138930733

Zero-point correction= 0.257623 (Hartree/Particle)
Thermal correction to Energy= 0.270562
Thermal correction to Enthalpy= 0.271506
Thermal correction to Gibbs Free Energy= 0.217411
Sum of electronic and ZPE= -340.881308
Sum of electronic and thermal Energies= -340.868369
Sum of electronic and thermal Enthalpies= -340.867424
Sum of electronic and thermal Free Energies= -340.921520

      E      CV        S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 169.780 44.529 113.853

 C,0,-0.5094610366,-0.4410400137,-1.4878382397
 C,0,-0.4500974142,0.3450507738,-0.1690198529
 C,0,0.8545437694,0.0667762275,0.6388854791
 B,0,0.7825635035,0.923983226,1.9455962377
 C,0,-1.7870733145,-0.1788174961,-2.2955320062
 H,0,-0.4277120285,-1.5155417896,-1.2690371628
 H,0,-1.3282554294,0.0886385653,0.4401369734
 H,0,1.7108498375,0.3316373308,0.0044883707
 H,0,0.9063414463,-1.011316786,0.8411200324
 H,0,0.2525979468,0.5000934793,2.9344194007
 H,0,-0.5312794576,1.4189014949,-0.3886384091
 H,0,1.183191055,2.0542020924,1.966138203
 H,0,0.3691266302,-0.1855135759,-2.0976753361
 C,0,-1.8478264691,-0.963355773,-3.613190513
 H,0,-2.6641578147,-0.4323708827,-1.6817892623
 H,0,-1.867619649,0.8971526658,-2.5101176828
 C,0,-3.1248862905,-0.7026938022,-4.4228576289
 H,0,-0.970758021,-0.7100975008,-4.2270554906
 H,0,-1.7671392058,-2.0393593988,-3.3988898512
 C,0,-3.1864408459,-1.4865471823,-5.7401881958
 H,0,-4.0023938798,-0.9560611413,-3.8094257694
 H,0,-3.2059426878,0.3733179641,-4.6376639763
 C,0,-4.464313684,-1.2210335588,-6.5428499192
```

```
H,0,-2.3098823855,-1.2331634283,-6.3532927416
H,0,-3.1058150007,-2.5616769323,-5.525593834
H,0,-4.4775145215,-1.7948766618,-7.4766018619
H,0,-5.3573495458,-1.4977985949,-5.9689565958
H,0,-4.555365507,-0.1591843013,-6.8029483673
```

Octene AM Product out twisted

B3LYP/6-31G*
E(RB3LYP) = -341.140605183

Zero-point correction= 0.256842 (Hartree/Particle)
Thermal correction to Energy= 0.269736
Thermal correction to Enthalpy= 0.270680
Thermal correction to Gibbs Free Energy= 0.216853
Sum of electronic and ZPE= -340.883763
Sum of electronic and thermal Energies= -340.870870
Sum of electronic and thermal Enthalpies= -340.869925
Sum of electronic and thermal Free Energies= -340.923752

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 169.262 44.822 113.287
```

```
C,0,-0.5367399874,-0.3764195518,-1.5040037653
C,0,-0.5026029357,0.4011592371,-0.181986384
C,0,0.7890651419,0.1958810755,0.62601815
B,0,0.889425501,0.8409814137,2.0433927936
C,0,-1.8226661329,-0.1610118848,-2.3121566627
H,0,-0.4133656829,-1.4500550262,-1.2949589176
H,0,-1.3653901595,0.1120079621,0.434618138
H,0,1.6755937453,0.4516471372,0.0234319921
H,0,0.933431302,-0.8864277459,0.8287981873
H,0,-0.0725173607,1.280353937,2.6097524124
H,0,-0.6323324125,1.4727594787,-0.3913103621
H,0,1.9413504887,0.8628727881,2.6181303059
H,0,0.3307066697,-0.0876127729,-2.1166370346
C,0,-1.8557685788,-0.9401798716,-3.6334155976
H,0,-2.6892996302,-0.4496690966,-1.6988217556
H,0,-1.9442874276,0.9124206817,-2.5208450755
C,0,-3.1417266522,-0.7243623842,-4.4414969269
H,0,-0.9891191353,-0.6514077196,-4.2467383852
H,0,-1.7343185758,-2.0136593447,-3.4247740048
C,0,-3.1753501841,-1.5034200292,-5.7626901863
H,0,-4.0090320849,-1.013179425,-3.828952177
H,0,-3.2638008088,0.348954819,-4.6511046084
C,0,-4.4628681822,-1.2820193072,-6.5631250942
H,0,-2.3092751726,-1.2145048546,-6.3751699274
H,0,-3.0542310948,-2.5759100281,-5.5534287994
H,0,-4.4558438353,-1.851262754,-7.4997648754
H,0,-5.3446061369,-1.5946304992,-5.9900732359
H,0,-4.5940116776,-0.2230542345,-6.8180182035
```

Octene AM Product out gauche

B3LYP/6-31G*
E(RB3LYP) = -341.140367888

Zero-point correction= 0.257128 (Hartree/Particle)
Thermal correction to Energy= 0.269798
Thermal correction to Enthalpy= 0.270742
Thermal correction to Gibbs Free Energy= 0.217620
Sum of electronic and ZPE= -340.883240
Sum of electronic and thermal Energies= -340.870570
Sum of electronic and thermal Enthalpies= -340.869626
Sum of electronic and thermal Free Energies= -340.922748

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 169.301 44.659 111.806
```

```
C,0,-0.1235532101,-0.0436638207,-0.9869425958
C,0,-0.3483469887,-0.5036301268,0.4596861497
C,0,0.8890071433,-0.2945842634,1.3486894709
B,0,1.2980607396,1.1658059138,1.7148251698
C,0,-1.3447281016,-0.2427829508,-1.8930961216
H,0,0.7356637748,-0.5861069506,-1.4098692978
H,0,-0.6397083741,-1.5638067763,0.4564976352
H,0,1.7460296989,-0.8746335166,0.9742381093
H,0,0.7037257899,-0.7314270051,2.3535156809
H,0,0.5212249535,2.0756146129,1.6211256653
H,0,-1.1984556619,0.0483735657,0.8846839851
H,0,2.3818451814,1.3950304202,2.1728442733
H,0,0.1594464663,1.0203044186,-0.9871551926
C,0,-1.1176614671,0.216052675,-3.3394144897
H,0,-1.628584949,-1.3058136506,-1.8897846181
H,0,-2.202385476,0.3009539815,-1.4698991253
C,0,-2.3390459466,0.015718268,-4.2456931986
H,0,-0.8341208128,1.2791292963,-3.3425431453
H,0,-0.2593091309,-0.3273269777,-3.7620375674
C,0,-2.112826493,0.4744971238,-5.6920448991
H,0,-2.623140631,-1.0473711897,-4.24349785
H,0,-3.197962688,0.5590002391,-3.823897531
C,0,-3.3373555261,0.270586749,-6.5900222342
H,0,-1.8298930501,1.5368455345,-5.6945184607
H,0,-1.2550166857,-0.0685629354,-6.1138693455
H,0,-3.1437761432,0.6075085266,-7.6149042422
H,0,-3.6235927868,-0.7875063564,-6.6357921164
H,0,-4.202654625,0.8300551951,-6.2134291081
```

Octene AM Product anti

B3LYP/6-31G*
E(RB3LYP) = -341.137655648

Zero-point correction= 0.257706 (Hartree/Particle)
Thermal correction to Energy= 0.270637
Thermal correction to Enthalpy= 0.271582
Thermal correction to Gibbs Free Energy= 0.216764
Sum of electronic and ZPE= -340.879950
Sum of electronic and thermal Energies= -340.867018
Sum of electronic and thermal Enthalpies= -340.866074
Sum of electronic and thermal Free Energies= -340.920891

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 169.827 44.433 115.372
```

```
6 1.462429 1.071545 -0.062977
6 2.859198 0.575832 0.355094
6 3.353009 -0.671785 -0.440354
5 4.813850 -0.978387 0.028408
1 1.261185 2.015325 0.464532
6 0.318160 0.091198 0.223523
1 2.854168 0.347534 1.429883
```

```
1 3.282677 -0.450204 -1.513375
1 2.676721 -1.509370 -0.224665
1 5.010536 -1.545213 1.067164
1 3.569912 1.402969 0.220505
1 5.767534 -0.596260 -0.589796
1 1.474067 1.317175 -1.134850
6 -1.063148 0.655495 -0.135881
1 0.477851 -0.842895 -0.332197
1 0.333193 -0.180969 1.289767
6 -2.215136 -0.317750 0.144990
1 -1.231380 1.588048 0.423160
1 -1.075178 0.934432 -1.200135
6 -3.596225 0.246405 -0.212748
1 -2.048772 -1.250035 -0.415327
1 -2.203291 -0.598160 1.208957
6 -4.741071 -0.731430 0.071045
1 -3.762744 1.177587 0.347444
1 -3.608477 0.525953 -1.275929
1 -5.712608 -0.299299 -0.194809
1 -4.621245 -1.659221 -0.502006
1 -4.776695 -1.002527 1.133486
```

Octene AM Product twisted

B3LYP/6-31G*
E(RB3LYP) = -341.139382863

Zero-point correction= 0.257084 (Hartree/Particle)
Thermal correction to Energy= 0.269822
Thermal correction to Enthalpy= 0.270766
Thermal correction to Gibbs Free Energy= 0.217280
Sum of electronic and ZPE= -340.882299
Sum of electronic and thermal Energies= -340.869561
Sum of electronic and thermal Enthalpies= -340.868617
Sum of electronic and thermal Free Energies= -340.922103

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 169.316 44.715 112.571
```

```
C,0,-0.606685888,-0.4058003521,-1.7308458184
C,0,-0.648780993,0.3240555412,-0.3776677028
C,0,0.6782835541,0.3240944427,0.4015611768
B,0,0.6980722545,0.9677184303,1.8218155361
H,0,-1.5499889943,-0.2166769814,-2.2631099367
C,0,-0.3918140954,-1.9225552797,-1.6352241151
H,0,-1.4364273406,-0.1163275541,0.2501431881
H,0,1.5010457988,0.7085263802,-0.2215665031
H,0,0.9917929407,-0.7188044333,0.6187067443
H,0,-0.3127541756,1.168591469,2.4359219774
H,0,-0.9542964275,1.36467409,-0.5522691282
H,0,1.7418487423,1.219486113,2.3550168187
H,0,0.1861425632,0.0342723296,-2.3546431128
C,0,-0.4102132808,-2.6253171055,-2.9992977141
H,0,0.565440384,-2.1354324878,-1.1385103966
H,0,-1.1719189965,-2.3576189819,-0.9921816433
C,0,-0.1972624478,-4.1419639116,-2.9121588596
H,0,-1.3688505422,-2.4214342706,-3.4992634607
H,0,0.3661373063,-2.1863401191,-3.6436488364
C,0,-0.2179739287,-4.8445240536,-4.2758437071
H,0,0.7624698666,-4.3467919528,-2.4142553713
H,0,-0.9725136091,-4.5817269037,-2.2668285698
C,0,-0.0043040711,-6.3586232263,-4.179829267
H,0,-1.1771646907,-4.6409913851,-4.773002044
H,0,0.5567177996,-4.4056005241,-4.9207206729
H,0,-0.024638564,-6.8300732478,-5.1691078565
H,0,0.9628622992,-6.5940801277,-3.7185951518
H,0,-0.7841014639,-6.8311238979,-3.5697225733
```

Octene M Product out

B3LYP/6-31G*
E(RB3LYP) = -341.137723334

Zero-point correction= 0.257330 (Hartree/Particle)
Thermal correction to Energy= 0.270109
Thermal correction to Enthalpy= 0.271054
Thermal correction to Gibbs Free Energy= 0.218096
Sum of electronic and ZPE= -340.880393
Sum of electronic and thermal Energies= -340.867614
Sum of electronic and thermal Enthalpies= -340.866670
Sum of electronic and thermal Free Energies= -340.919627

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 169.496 45.096 111.458
```

```
C,0,-0.1744725895,0.1627135332,-1.2883983245
C,0,-0.2666550058,0.1088842488,0.2467799586
C,0,1.0675596567,0.5818383666,0.9024752067
B,0,-0.4846452422,-1.3138886633,0.8623009417
C,0,-1.486273358,-0.2155215174,-1.9895821594
H,0,0.1251800881,1.1730777632,-1.6076172346
H,0,-1.0466147988,0.8079571983,0.5841359705
H,0,1.3142056224,1.5929935001,0.5555070099
H,0,1.005140221,0.6124530348,1.9963123266
H,0,-0.0520380967,-2.2894474861,0.3127934118
H,0,1.9069336922,-0.0698435197,0.6298004438
H,0,-0.9973551861,-1.4515615512,1.937026336
H,0,0.6238475328,-0.5148009589,-1.6259442669
C,0,-1.4066867303,-0.1501262854,-3.520015068
H,0,-1.7802861456,-1.2327256371,-1.6870660726
H,0,-2.2900143277,0.4492791133,-1.6388925535
C,0,-2.7185497702,-0.527002494,-4.2202212088
H,0,-1.1117657729,0.8654666457,-3.8237390923
H,0,-0.6038289634,-0.8159857999,-3.8698290814
C,0,-2.6384307514,-0.4612878504,-5.7508517601
H,0,-3.0142031615,-1.5425245914,-3.9167209161
H,0,-3.5218669925,0.1389941985,-3.8709321035
C,0,-3.9525208388,-0.8390806384,-6.4421697262
H,0,-2.3435407473,0.5533756545,-6.054464922
H,0,-1.8364335736,-1.1272848947,-6.0999441291
H,0,-3.8624317633,-0.7824857665,-7.5330473239
H,0,-4.2556009495,-1.8617567772,-6.1856445886
H,0,-4.766496048,-0.1684858254,-6.1399870739
```

Octene M Product anti

B3LYP/6-31G*
E(RB3LYP) = -341.136173158

Zero-point correction= 0.257278 (Hartree/Particle)
Thermal correction to Energy= 0.270155
Thermal correction to Enthalpy= 0.271099
Thermal correction to Gibbs Free Energy= 0.217726

```
Sum of electronic and ZPE= -340.878896
Sum of electronic and thermal Energies= -340.866018
Sum of electronic and thermal Enthalpies= -340.865074
Sum of electronic and thermal Free Energies= -340.918447

     E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 169.525 45.187 112.333

 C,0,-0.3467487331,0.2615548894,-1.3243644087
 C,0,-0.3388302568,-0.0012366398,0.1969464776
 C,0,0.9960493635,0.4476996943,0.8698707317
 B,0,-0.4476542901,-1.5083631388,0.6051812613
 H,0,-1.2457820251,-0.1989431,-1.7598082119
 C,0,-0.3195475662,1.7418933408,-1.7332300959
 H,0,-1.1411797586,0.5868185805,0.6679737197
 H,0,1.8591671991,-0.0696206398,0.4325710826
 H,0,1.155182906,1.523293973,0.7360101861
 H,0,0.0021776216,-2.3694812764,-0.0992904048
 H,0,0.9975849002,0.2521830236,1.9488170396
 H,0,-0.9036774237,-1.8269690559,1.6671246751
 H,0,0.5056844191,-0.2619343451,-1.7823383237
 C,0,-0.4098023312,1.9508544512,-3.2509514759
 H,0,-1.154340814,2.2671325372,-1.2448436697
 H,0,0.5982146748,2.2193060273,-1.3631100573
 C,0,-0.3839285064,3.4262300745,-3.6703944274
 H,0,0.4206341131,1.42012186,-3.7401512385
 H,0,-1.3313747887,1.4825293493,-3.6277900937
 C,0,-0.4741300297,3.6352738623,-5.1875679112
 H,0,-1.2142407039,3.9579716522,-3.1818151335
 H,0,0.5379873417,3.8952503899,-3.2948758683
 C,0,-0.4477729356,5.1112585773,-5.5976012397
 H,0,0.3558730428,3.1047576381,-5.6757294625
 H,0,-1.3953914849,3.1673841134,-5.5630678678
 H,0,-0.5135122472,5.2265356974,-6.6856005028
 H,0,-1.2866130346,5.6612082609,-5.1532089936
 H,0,0.477871348,5.5982312032,-5.2664977866
```

Dodecene

```
B3LYP/6-31G*
E(RB3LYP) = -471.730998629

Zero-point correction= 0.337861 (Hartree/Particle)
Thermal correction to Energy= 0.353864
Thermal correction to Enthalpy= 0.354808
Thermal correction to Gibbs Free Energy= 0.292254
Sum of electronic and ZPE= -471.393138
Sum of electronic and thermal Energies= -471.377134
Sum of electronic and thermal Enthalpies= -471.376190
Sum of electronic and thermal Free Energies= -471.438744

     E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 222.053 56.286 131.656

 C,0,-0.2212353662,-0.6549953277,-1.35033435
 C,0,-0.2516434576,-0.6148043944,0.153264092
 C,0,0.6304551222,0.0188570427,0.9270406498
 C,0,-1.4779325093,-0.0387646042,-1.994626376
 H,0,-0.1341757685,-1.7005002293,-1.6854573824
 H,0,-1.0819013589,-1.1454541689,0.6240259376
 H,0,1.4732180017,0.5631578704,0.5046377626
 H,0,0.5492051904,0.0169602289,2.0108922496
 H,0,0.6744638712,-0.1350602897,-1.7157285539
 C,0,-1.4738009022,-0.1195872887,-3.5263489717
 H,0,-2.3717101621,-0.5479488226,-1.6050011486
 H,0,-1.5632121995,1.010304952,-1.6793132173
 C,0,-2.7216761982,0.4942713739,-4.1741921499
 H,0,-0.5771496023,0.3863130115,-3.9142574974
 H,0,-1.3846541184,-1.1722058835,-3.8340972131
 C,0,-2.7189201675,0.4146266577,-5.7063842689
 H,0,-3.6182475019,-0.0116290806,-3.7859932787
 H,0,-2.8103427535,1.5464511357,-3.8654507358
 C,0,-3.966544716,1.0286274444,-6.3546451283
 H,0,-1.8221301275,0.9203220559,-6.0943941245
 H,0,-2.6301242634,-0.6377208791,-6.0149808894
 C,0,-3.964171705,0.9491823648,-7.8867912994
 H,0,-4.8633165816,0.5230273416,-5.9664357022
 H,0,-4.0552160687,2.0809380432,-6.0459573797
 C,0,-5.2114136404,1.5631723988,-8.5353916424
 H,0,-3.0672840419,1.4547008246,-8.2749470002
 H,0,-3.8755388785,-0.1032046516,-8.1954353496
 C,0,-5.2095237136,1.4840232841,-10.0674947087
 H,0,-6.1089676144,1.0579454859,-8.1480841213
 H,0,-5.3006681328,2.6158564072,-8.2275836
 C,0,-6.4585382859,2.099416461,-10.706772174
 H,0,-4.31325031,1.9894247633,-10.4547258795
 H,0,-5.1211108036,0.4322954469,-10.3752936441
 H,0,-6.4267422335,2.0271203653,-11.8000024505
 H,0,-7.3693992224,1.5917950323,-10.3658332556
 H,0,-6.5554517802,3.1606146283,-10.4459091985
```

Dodecene AM Complex out

```
B3LYP/6-31G*
E(RB3LYP) = -498.362901587

Zero-point correction= 0.370076 (Hartree/Particle)
Thermal correction to Energy= 0.388111
Thermal correction to Enthalpy= 0.389055
Thermal correction to Gibbs Free Energy= 0.321948
Sum of electronic and ZPE= -497.992825
Sum of electronic and thermal Energies= -497.974790
Sum of electronic and thermal Enthalpies= -497.973846
Sum of electronic and thermal Free Energies= -498.040953

     E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 243.543 64.490 141.239

 C,0,-0.2719020926,-0.4063114042,-1.2985564901
 C,0,-0.2876683708,-0.439568139,0.2100920718
 C,0,0.858746364,-0.4109829628,0.9874674223
 B,0,0.1336974969,1.2561055491,1.0185843254
 C,0,-1.5427082241,0.1408981681,-1.9606241639
 H,0,-0.1173546387,-1.4470771092,-1.6255368526
 H,0,-1.2133914283,-0.7761403567,0.6731366816
 H,0,1.8322360632,-0.3435448067,0.5125475318
 H,0,0.8484127383,-0.8051910041,1.9968420859
 H,0,-0.0719491773,1.4077354696,2.1909003696
 H,0,-0.944376829,1.3874801879,0.462481259
 H,0,0.9412861922,1.9128369058,0.4206312316
 H,0,0.608964143,0.1541240047,-1.6359480427
```

```
C,0,-1.5070887038,0.0412768279,-3.4910584783
H,0,-2.4166460893,-0.4079764655,-1.5797573523
H,0,-1.6816854151,1.1876856731,-1.6628451511
C,0,-2.7699136406,0.5936669447,-4.1646393854
H,0,-0.6276783756,0.5832467094,-3.8690642788
H,0,-1.3672197484,-1.00922049,-3.7870810162
C,0,-2.7368478839,0.5004728293,-5.6956331607
H,0,-3.649635292,0.0519362974,-3.7864668601
H,0,-2.9092870945,1.6433288238,-3.8671411504
C,0,-3.9994253921,1.0525083625,-6.3700542751
H,0,-1.8571020077,1.0426696741,-6.0730153057
H,0,-2.5959719216,-0.549497779,-5.9928284999
C,0,-3.9661882685,0.9618407367,-7.9012043738
H,0,-4.8791132767,0.509681126,-5.9932319068
H,0,-4.1407128719,2.1019085099,-6.0715328742
C,0,-5.2283440608,1.5138325183,-8.5761326914
H,0,-3.0864416014,1.5048236229,-8.2777303085
H,0,-3.8245473607,-0.0876716009,-8.1997573094
C,0,-5.1952865362,1.4245364717,-10.107304281
H,0,-6.1087203042,0.9707310028,-8.2008538581
H,0,-5.3708569782,2.5633433779,-8.2778221024
C,0,-6.4594258686,1.9779051959,-10.7730088759
H,0,-4.3162307437,1.9678501266,-10.4824337735
H,0,-5.0535055672,0.3759134429,-10.4056367375
H,0,-6.4046163099,1.8999790348,-11.8649180086
H,0,-7.3521511682,1.4313430927,-10.4445482574
H,0,-6.6093317559,3.0352214318,-10.5217331571
```

Dodecene AM Complex anti

B3LYP/6-31G*
E(RB3LYP) = -498.362892192

Zero-point correction= 0.370283 (Hartree/Particle)
Thermal correction to Energy= 0.388390
Thermal correction to Enthalpy= 0.389334
Thermal correction to Gibbs Free Energy= 0.321956
Sum of electronic and ZPE= -497.992609
Sum of electronic and thermal Energies= -497.974502
Sum of electronic and thermal Enthalpies= -497.973558
Sum of electronic and thermal Free Energies= -498.040936

```
      E      CV        S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 243.719 64.510 141.810
```

```
C,0,-0.3581887032,-0.2426525659,-1.2335044486
C,0,-0.3859305054,-0.2618102855,0.2744025359
C,0,0.7452495612,-0.1966275631,1.0600683607
B,0,-0.0393241439,1.4900526241,1.0458704941
H,0,-1.2466623811,0.2730233495,-1.6161814819
C,0,-0.3051744834,-1.6723735361,-1.8085514134
H,0,-1.311876348,-0.6005896386,0.7345272732
H,0,1.72307314,-0.0931935862,0.6005765467
H,0,0.7332204951,-0.5448699366,2.0860273792
H,0,-0.2295486935,1.6339546034,2.2219192915
H,0,-1.1086390382,1.6052616379,0.4784494923
H,0,0.7903127668,2.1306447213,0.4614608703
H,0,0.5145339717,0.3280272301,-1.5733083631
C,0,-0.3085169652,-1.692167653,-3.3427206969
H,0,0.5935549875,-2.1813237863,-1.4343091565
H,0,-1.1631799907,-2.2498481933,-1.4350056085
```

```
C,0,-0.2563612137,-3.1076463356,-3.932048462
H,0,-1.2081496392,-1.1772961908,-3.7103924327
H,0,0.5477760924,-1.1106679683,-3.714557639
C,0,-0.2585649332,-3.1304721153,-5.4661171308
H,0,0.6428968742,-3.6221801005,-3.5622413064
H,0,-1.1130841401,-3.6881019161,-3.558750663
C,0,-0.2076560348,-4.5451627709,-6.0574834986
H,0,-1.1573659496,-2.6146118776,-5.8353591503
H,0,0.598328004,-2.549865191,-5.8388085089
C,0,-0.2095121294,-4.5685357268,-7.5914687783
H,0,0.6909516015,-5.0610973499,-5.6876372299
H,0,-1.0647815455,-5.1254429105,-5.6845418006
C,0,-0.1594484787,-5.982705565,-8.183699854
H,0,-1.1078671228,-4.0519542329,-7.9611075064
H,0,0.6478460987,-3.9884267243,-7.9641999041
C,0,-0.1611847591,-6.0066433511,-9.7176487885
H,0,0.7388790759,-6.5001215576,-7.8147607501
H,0,-1.0169237637,-6.5634351409,-7.8117912299
C,0,-0.1113900864,-7.4224788833,-10.3005995819
H,0,-1.0589079545,-5.4901408402,-10.0863891523
H,0,0.6959596187,-5.427185426,-10.0893647509
H,0,-0.1138286153,-7.4050959789,-11.3964828764
H,0,0.7933385911,-7.9525261925,-9.9779970881
H,0,-0.9747762601,-8.0158950758,-9.9750639919
```

Dodecene M Complex out

B3LYP/6-31G*
E(RB3LYP) = -498.362901457

Zero-point correction= 0.370068 (Hartree/Particle)
Thermal correction to Energy= 0.388106
Thermal correction to Enthalpy= 0.389050
Thermal correction to Gibbs Free Energy= 0.321930
Sum of electronic and ZPE= -497.992834
Sum of electronic and thermal Energies= -497.974795
Sum of electronic and thermal Enthalpies= -497.973851
Sum of electronic and thermal Free Energies= -498.040971

```
      E      CV        S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 243.540 64.490 141.266
```

```
C,0,-0.363469385,0.331430797,-1.3771488312
C,0,-0.2452235386,0.4044257617,0.1255660452
C,0,0.9661096602,0.3893813944,0.7985324886
B,0,0.2399985182,-1.2691967128,0.941006233
C,0,-1.6906924855,-0.2234163143,-1.9093505855
H,0,-0.2330432861,1.362052256,-1.744283771
H,0,-1.1245304217,0.7598741865,0.6595445708
H,0,1.8929221108,0.3040561982,0.2403810595
H,0,1.0473992958,0.8119570827,1.7931138936
H,0,0.9889051744,-1.9488930892,0.2942752562
H,0,0.1328198851,-1.3881610511,2.1300458606
H,0,-0.8820023054,-1.4079297212,0.4812550021
H,0,0.4811734196,-0.2442715447,-1.7759172342
C,0,-1.7901170761,-0.1647606254,-3.4391025985
H,0,-1.8082910695,-1.2608793053,-1.5722326437
H,0,-2.5246389005,0.3419808138,-1.4678232229
C,0,-3.1104720543,-0.7251646533,-3.9833001286
H,0,-1.671447396,0.8764021503,-3.7747065103
H,0,-0.9504657143,-0.7232826117,-3.8784421173
```

```
C,0,-3.2125597348,-0.6728794232,-5.5131645218
H,0,-3.2284989938,-1.7654715166,-3.6463151653
H,0,-3.9504151888,-0.1668944189,-3.543723596
C,0,-4.5326273007,-1.2332947392,-6.0582474937
H,0,-3.0932983611,0.3677603842,-5.8499933467
H,0,-2.3724040578,-1.2313804026,-5.9519831266
C,0,-4.6347713514,-1.1830234217,-7.5882030908
H,0,-4.6519709895,-2.2734636331,-5.7204335351
H,0,-5.3728016429,-0.6744521235,-5.6197226079
C,0,-5.9543875041,-1.7435125416,-8.1338482979
H,0,-4.5151365067,-0.1427322998,-7.9260631174
H,0,-3.7944570691,-1.7419403689,-8.0264232343
C,0,-6.0568315297,-1.6943442469,-9.6637693527
H,0,-6.0748037427,-2.7838743572,-7.7963349317
H,0,-6.7954551586,-1.1846504539,-7.696663515
C,0,-7.3775698917,-2.2560102737,-10.2000213457
H,0,-5.9371819253,-0.6548565968,-10.001206779
H,0,-5.2170319395,-2.2533603295,-10.1006788558
H,0,-7.4194541827,-2.2068761259,-11.2941748168
H,0,-7.5094980882,-3.3053679969,-9.9085235401
H,0,-8.2351062724,-1.6948921258,-9.8084194962
```

Dodecene M Complex anti

B3LYP/6-31G*
E(RB3LYP) = -498.362892191

Zero-point correction= 0.370283 (Hartree/Particle)
Thermal correction to Energy= 0.388390
Thermal correction to Enthalpy= 0.389334
Thermal correction to Gibbs Free Energy= 0.321956
Sum of electronic and ZPE= -497.992609
Sum of electronic and thermal Energies= -497.974502
Sum of electronic and thermal Enthalpies= -497.973558
Sum of electronic and thermal Free Energies= -498.040937

```
     E      CV        S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 243.719 64.510 141.811
```

```
C,0,-0.2750027945,0.1839793561,-1.2328776393
C,0,-0.4356554107,0.2430644231,0.2656609991
C,0,0.6237135957,0.3156275097,1.1451847294
B,0,0.014105693,-1.4418434646,1.1277036875
H,0,-1.0713932835,-0.4300944069,-1.6692547151
C,0,-0.3138655765,1.5947504702,-1.8540123819
H,0,-1.4270887427,0.4988525679,0.6337563225
H,0,1.6432451446,0.2991774565,0.7733576493
H,0,0.4880253407,0.6898542343,2.1529310166
H,0,-0.985253131,-1.6812478565,0.4775929096
H,0,0.9506449586,-2.011061479,0.6387558774
H,0,-0.2623930133,-1.5709251545,2.2882534229
H,0,0.6761527084,-0.3043610693,-1.4770347868
C,0,-0.1855927895,1.5704906952,-3.3827484813
H,0,-1.2540443609,2.0923022548,-1.5752772395
H,0,0.4940883427,2.202941365,-1.4249239099
C,0,-0.2226565803,2.9666048067,-4.0176494537
H,0,0.7531388933,1.0688129421,-3.6597766199
H,0,-0.9944264044,0.9565048796,-3.8051507713
C,0,-0.0935621657,2.9454794446,-5.5463027982
H,0,-1.1618387097,3.467241322,-3.7391639252
H,0,0.5856524557,3.5802775006,-3.5931381656
C,0,-0.1316083805,4.3406251492,-6.1833087649
H,0,0.8457513719,2.4447282403,-5.824164579
H,0,-0.9013352535,2.3305493299,-5.9701641157
C,0,-0.0022299379,4.3200823594,-7.7118701496
H,0,-1.071142199,4.8410345054,-5.905205452
H,0,0.675912653,4.9556269031,-5.7588642597
C,0,-0.0409880475,5.7145997323,-8.3497883139
H,0,0.9374995824,3.8198733562,-7.9897506067
H,0,-0.8094542653,3.704468567,-8.1360629837
C,0,0.0884508035,5.6946319164,-9.878324382
H,0,-0.98082365,6.2153902853,-8.0727600169
H,0,0.7661870561,6.3310208744,-7.9263249982
C,0,0.0484528553,7.0910417897,-10.5070850564
H,0,1.0278159791,5.1950782021,-10.155235809
H,0,-0.71823273,5.0791862291,-10.3015732237
H,0,0.1431511868,7.0423632854,-11.597924727
H,0,-0.894461006,7.6025281736,-10.2771998304
H,0,0.8647568117,7.7193583038,-10.1296954578
```

Dodecene AMTS out

B3LYP/6-31G*
E(RB3LYP) = -498.362865029

Zero-point correction= 0.369767 (Hartree/Particle)
Thermal correction to Energy= 0.387099
Thermal correction to Enthalpy= 0.388043
Thermal correction to Gibbs Free Energy= 0.322385
Sum of electronic and ZPE= -497.993098
Sum of electronic and thermal Energies= -497.975767
Sum of electronic and thermal Enthalpies= -497.974822
Sum of electronic and thermal Free Energies= -498.040480

```
     E      CV        S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 242.908 62.408 138.188
```

```
C,0,-0.3236879024,-0.3980625653,-1.3978416192
C,0,-0.3386581574,-0.426474597,0.1118904565
C,0,0.8234274384,-0.4340282694,0.8859754288
B,0,0.1446982621,1.2002249515,0.9509902886
C,0,-1.5889910792,0.1612495783,-2.0597797252
H,0,-0.177968621,-1.4409060077,-1.7212134039
H,0,-1.2621993541,-0.7720338729,0.5730712875
H,0,1.7897684934,-0.4081049659,0.3922658516
H,0,0.8114127882,-0.8743892069,1.8764242596
H,0,-0.1349971103,1.3694754757,2.1045933979
H,0,-0.9226510806,1.3078299088,0.3506304886
H,0,0.9266280691,1.8903724026,0.3574914032
H,0,0.5624408703,0.1537094158,-1.7358496419
C,0,-1.5534341469,0.0671178749,-3.5905887207
H,0,-2.4681633142,-0.3809688872,-1.6814781828
H,0,-1.7197149842,1.2089012399,-1.7592639231
C,0,-2.8112229078,0.6324098519,-4.2628571338
H,0,-0.6691986956,0.6028829195,-3.966089904
H,0,-1.4222735424,-0.9834807056,-3.8900112566
C,0,-2.7786619469,0.543046249,-5.7940554295
H,0,-3.6957412935,0.097388304,-3.886329319
H,0,-2.9414884387,1.6824746699,-3.9625678784
C,0,-4.0358651816,1.1087929961,-6.4671921199
H,0,-1.8937754604,1.0779180529,-6.1698554656
H,0,-2.6476525137,-0.5074015597,-6.0940263164
```

```
C,0,-4.0036667236,1.0207996287,-7.9985062299
H,0,-4.920785788,0.5738762267,-6.0913187322
H,0,-4.1667892711,2.1589379554,-6.1665674386
C,0,-5.2602082549,1.5869481061,-8.672169268
H,0,-3.118481167,1.5555329549,-8.3740805123
H,0,-3.8727505335,-0.0295114126,-8.2991173915
C,0,-5.2284766794,1.4997092901,-10.2034898865
H,0,-6.1460838722,1.052439295,-8.2975000614
H,0,-5.3917136155,2.6374146064,-8.3721730789
C,0,-6.4868344224,2.0675575589,-10.8679074888
H,0,-4.3437807135,2.0342222281,-10.5780007812
H,0,-5.0979362611,0.4501034614,-10.5034841388
H,0,-6.4332719155,1.9906324371,-11.9599468569
H,0,-7.3852571639,1.5301311706,-10.5399014365
H,0,-6.6253048089,3.12605824,-10.615055521
```

Dodecene AMTS anti

```
B3LYP/6-31G*
E(RB3LYP) = -498.362572824

Zero-point correction= 0.370040 (Hartree/Particle)
Thermal correction to Energy= 0.387337
Thermal correction to Enthalpy= 0.388281
Thermal correction to Gibbs Free Energy= 0.322733
Sum of electronic and ZPE= -497.992532
Sum of electronic and thermal Energies= -497.975236
Sum of electronic and thermal Enthalpies= -497.974292
Sum of electronic and thermal Free Energies= -498.039840

     E      CV        S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 243.057 62.247 137.957

 C,0,-0.3815365944,-0.226083658,-1.2443896071
 C,0,-0.4085660196,-0.2196072494,0.2666352732
 C,0,0.7556388777,-0.223028978,1.0461906821
 B,0,0.0759217859,1.3967167639,1.0897756886
 H,0,-1.2779104873,0.2674144951,-1.6393829454
 C,0,-0.2995574835,-1.6634975805,-1.7931024702
 H,0,-1.329655855,-0.5725240921,0.7266639195
 H,0,1.7210793132,-0.21084705,0.549593753
 H,0,0.7420036262,-0.660592314,2.0379608337
 H,0,-0.2494032447,1.5867494293,2.2275386587
 H,0,-0.9787391515,1.4639942621,0.4524565405
 H,0,0.8463106064,2.0981620242,0.4961754932
 H,0,0.4823546344,0.3545285225,-1.5902094006
 C,0,-0.2944083652,-1.7116667044,-3.3268660176
 H,0,0.6061016145,-2.1490871157,-1.4053206872
 H,0,-1.1489255602,-2.2498270097,-1.4134562351
 C,0,-0.2128397839,-3.1367610067,-3.8891166376
 H,0,-1.2013889766,-1.2205992407,-3.7088807015
 H,0,0.5531381579,-1.1214575723,-3.7049576935
 C,0,-0.2060450735,-3.1885401123,-5.4225169008
 H,0,0.6937119146,-3.6273565194,-3.5049298899
 H,0,-1.0607278001,-3.7259797746,-3.5094148365
 C,0,-0.1256444858,-4.6130266998,-5.9865989171
 H,0,-1.1121756285,-2.6965726735,-5.806219215
 H,0,0.6420876723,-2.5993129722,-5.8016710242
 C,0,-0.1184442095,-4.6654073587,-7.5199059123
 H,0,0.7802932698,-5.1049970899,-5.602306334
 H,0,-0.9739732399,-5.2019977452,-5.6072304281
 C,0,-0.0390100219,-6.0894221258,-8.0848337657
 H,0,-1.0240881838,-4.1726415837,-7.9040073284
 H,0,0.7302074377,-4.0767181173,-7.8990554671
 C,0,-0.0317230466,-6.1423855985,-9.6180699509
 H,0,0.8666432805,-6.5829771349,-7.7014055537
 H,0,-0.887713951,-6.6787786959,-7.7064221946
 C,0,0.0473518957,-7.568051282,-10.1736393724
 H,0,-0.9367216948,-5.6497220451,-10.0013452583
 H,0,0.8167389506,-5.5544459895,-9.9963106539
 H,0,0.0508106561,-7.571663347,-11.2696583331
 H,0,0.9598220573,-8.0750390066,-9.8361174371
 H,0,-0.8068288935,-8.1709790537,-9.8413576735
```

Dodecene MTS out

```
B3LYP/6-31G*
E(RB3LYP) = -498.359107146

Zero-point correction= 0.370043 (Hartree/Particle)
Thermal correction to Energy= 0.387268
Thermal correction to Enthalpy= 0.388213
Thermal correction to Gibbs Free Energy= 0.322884
Sum of electronic and ZPE= -497.989064
Sum of electronic and thermal Energies= -497.971839
Sum of electronic and thermal Enthalpies= -497.970895
Sum of electronic and thermal Free Energies= -498.036223

     E      CV        S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 243.015 62.126 137.496

 C,0,-0.3012388767,0.3828980325,-1.3937950127
 C,0,-0.3021797345,0.3904788861,0.1255672143
 C,0,0.8857576455,0.4359080214,0.877991572
 B,0,-0.0206590166,-1.1544669785,0.8972969678
 C,0,-1.5476558605,-0.2650290969,-2.0100346267
 H,0,-0.229658595,1.4237673061,-1.7473031489
 H,0,-1.1799424383,0.8410125284,0.5837410364
 H,0,1.8505161356,0.3606005915,0.3828640888
 H,0,0.9069989363,0.890919332,1.8635135655
 H,0,0.1944463354,-1.9384059966,0.0186962887
 H,0,1.0452756397,-1.1386397496,1.537358088
 H,0,-0.8658791055,-1.3047998769,1.73077957
 H,0,0.5996639665,-0.1305494981,-1.7546880034
 C,0,-1.5565666183,-0.2183719787,-3.5431198237
 H,0,-1.6134201127,-1.3063699555,-1.6696088894
 H,0,-2.4456632508,0.2414183385,-1.625789763
 C,0,-2.8028029312,-0.8613838337,-4.1653641028
 H,0,-1.4820366499,0.8270907291,-3.8785954458
 H,0,-0.6580629356,-0.7244614543,-3.9267996216
 C,0,-2.8129696959,-0.8204087741,-5.6989410279
 H,0,-2.8774828727,-1.906032269,-3.8290996081
 H,0,-3.7016060421,-0.355281818,-3.78259217
 C,0,-4.0591032519,-1.4634237598,-6.3214046298
 H,0,-2.737609913,0.2246558735,-6.0349386919
 H,0,-1.9140828067,-1.3266587598,-6.0813753282
 C,0,-4.069346053,-1.4230843,-7.8550295514
 H,0,-4.1343492198,-2.5082992036,-5.9852784442
 H,0,-4.9580874079,-0.957310393,-5.9390098797
 C,0,-5.3148325157,-2.0665277235,-8.4779256797
 H,0,-3.9943250639,-0.3780447488,-8.1911580103
 H,0,-3.1701176909,-1.9289492228,-8.2372936017
```

```
C,0,-5.3255313026,-2.0264296936,-10.0114965217
H,0,-5.3903490356,-3.1118811659,-8.1427405865
H,0,-6.2147979215,-1.5611441341,-8.0964494555
C,0,-6.5724381491,-2.6716002977,-10.6250103599
H,0,-5.2512028299,-0.9819766238,-10.3466508854
H,0,-4.4266744303,-2.5317082014,-10.3928657806
H,0,-6.5499666772,-2.6267268972,-11.7199547553
H,0,-6.6553525676,-3.7268368878,-10.3366222704
H,0,-7.4863150861,-2.1656453463,-10.2896877153
```

Dodecene MTS anti

B3LYP/6-31G*
E(RB3LYP) = -498.359240752

Zero-point correction= 0.370052 (Hartree/Particle)
Thermal correction to Energy= 0.387311
Thermal correction to Enthalpy= 0.388255
Thermal correction to Gibbs Free Energy= 0.322829
Sum of electronic and ZPE= -497.989189
Sum of electronic and thermal Energies= -497.971930
Sum of electronic and thermal Enthalpies= -497.970986
Sum of electronic and thermal Free Energies= -498.036412

```
      E     CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 243.041 62.128 137.701
```

```
C,0,-0.349602522,0.2174704162,-1.235527872
C,0,-0.3540986941,0.2036421693,0.2830395188
C,0,0.8372028549,0.2105961525,1.0336372665
B,0,-0.1149857629,-1.3545580537,1.0301010733
H,0,-1.2174542898,-0.3422747406,-1.6071855308
C,0,-0.3756328719,1.6422052302,-1.8149145793
H,0,-1.2182455886,0.6739608829,0.7468581003
H,0,1.7981281054,0.1182293888,0.5335633119
H,0,0.8735206314,0.6488782096,2.0262882236
H,0,0.0732181607,-2.131771409,0.1396274387
H,0,0.9563728237,-1.3739985929,1.6633439406
H,0,-0.9580170481,-1.4905997699,1.8682596016
H,0,0.5368782487,-0.3168883316,-1.6016893804
C,0,-0.410763868,1.6691198055,-3.3484555316
H,0,-1.2518265447,2.1787984445,-1.4218103636
H,0,0.5046407208,2.1962502695,-1.4573216631
C,0,-0.4408233245,3.086445568,-3.9351034089
H,0,0.4651111948,1.1315628758,-3.7412305873
H,0,-1.2910286111,1.1110983769,-3.7003249193
C,0,-0.4756081519,3.1139065273,-5.4687315658
H,0,-1.31732121,3.6230353721,-3.5423821609
H,0,0.4391017993,3.6445686525,-3.5819068239
C,0,-0.5070484834,4.530885007,-6.0561221616
H,0,0.4011374667,2.577527962,-5.8612634141
H,0,-1.3550977505,2.5550292237,-5.8216171613
C,0,-0.5418547087,4.5585335884,-7.5897083903
H,0,-1.3839342626,5.0670304352,-5.6635207677
H,0,0.3723686154,5.089879991,-5.7029554819
C,0,-0.5742309529,5.9750686326,-8.1776515529
H,0,0.3352896051,4.0227136776,-7.9822739646
H,0,-1.4209784831,3.999008728,-7.9427261367
C,0,-0.6091320979,6.0032089924,-9.7111676382
H,0,-1.4514720661,6.5115515698,-7.785876828
H,0,0.304875371,6.5354166285,-7.8254238065
C,0,-0.6416197104,7.4215826346,-10.2896553194
H,0,0.267720984,5.46807567,-10.102861637
H,0,-1.4876956963,5.4439613802,-10.0633101144
H,0,-0.6659311803,7.4074429645,-11.385353304
H,0,-1.5263749051,7.9705052276,-9.9440149928
H,0,0.2422552029,7.994884243,-9.9835734172
```

Dodecene AM Product out

B3LYP/6-31G*
E(RB3LYP) = -498.393791063

Zero-point correction= 0.371965 (Hartree/Particle)
Thermal correction to Energy= 0.390365
Thermal correction to Enthalpy= 0.391309
Thermal correction to Gibbs Free Energy= 0.322757
Sum of electronic and ZPE= -498.021826
Sum of electronic and thermal Energies= -498.003427
Sum of electronic and thermal Enthalpies= -498.002482
Sum of electronic and thermal Free Energies= -498.071035

```
      E     CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 244.957 63.848 144.280
```

```
C,0,-0.4335878381,-0.411648145,-1.3676885469
C,0,-0.3725357582,0.3736905167,-0.0485016042
C,0,0.9326993833,0.0942391785,0.7580975723
B,0,0.8623458864,0.9508199276,2.0652906554
C,0,-1.7119066436,-0.1480303992,-2.1738261005
H,0,-0.3523337069,-1.4863457094,-1.1496521622
H,0,-1.2502586858,0.1174835251,0.5613709561
H,0,1.7885769022,0.3589425758,0.1230592241
H,0,0.9840458342,-0.9839878671,0.9597236435
H,0,0.3324083187,0.5269279007,3.054127776
H,0,-0.453274039,1.4477196104,-0.2674186378
H,0,1.2630363966,2.0810169114,2.0858053148
H,0,0.4445184166,-0.1563330628,-1.9783121343
C,0,-1.7744567225,-0.931529433,-3.4914934949
H,0,-2.5885483233,-0.4014128098,-1.5592558894
H,0,-1.7919351727,0.928196976,-2.3876711198
C,0,-3.0519234641,-0.6696184048,-4.2995511736
H,0,-0.8978027579,-0.678522153,-4.1061684735
H,0,-1.6942085603,-2.0077857169,-3.2779687005
C,0,-3.1152124114,-1.4526290846,-5.6171459362
H,0,-3.9285518734,-0.9225806362,-3.6847610544
H,0,-3.1321383811,0.4066959997,-4.5129686366
C,0,-4.3925448996,-1.1914113085,-6.4261874979
H,0,-2.2384451706,-1.1999435738,-6.2319012107
H,0,-3.0347112088,-2.5289738218,-5.4038471507
C,0,-4.4558290012,-1.9745737625,-7.7440215378
H,0,-5.2693006799,-1.4441023072,-5.8114464673
H,0,-4.4730417626,-0.1150837672,-6.6394930304
C,0,-5.73276141,-1.7137675831,-8.5533389993
H,0,-3.5789317513,-1.7218666984,-8.3586318955
H,0,-4.3752039476,-3.0509074526,-7.53057149
C,0,-5.7963624017,-2.496772805,-9.8712251749
H,0,-6.6103809784,-1.966642169,-7.9396418898
H,0,-5.8140872215,-0.6375656206,-8.7677247546
C,0,-7.0744541906,-2.2303996124,-10.672722697
H,0,-4.9200165115,-2.2436956573,-10.4849238954
H,0,-5.7159266944,-3.5721317014,-9.6572400846
```

```
H,0,-7.0889075795,-2.8037212932,-11.606805405
H,0,-7.9672001603,-2.5069031061,-10.0981913008
H,0,-7.1652212302,-1.1683374599,-10.9321859953
```

Dodecene AM Product out twisted

B3LYP/6-31G*
E(RB3LYP) = -498.395462935

Zero-point correction= 0.371183 (Hartree/Particle)
Thermal correction to Energy= 0.389535
Thermal correction to Enthalpy= 0.390479
Thermal correction to Gibbs Free Energy= 0.322212
Sum of electronic and ZPE= -498.024280
Sum of electronic and thermal Energies= -498.005928
Sum of electronic and thermal Enthalpies= -498.004984
Sum of electronic and thermal Free Energies= -498.073251

```
     E      CV       S
  KCal/Mol Cal/Mol-K Cal/Mol-K
Total 244.437 64.144 143.680
```

```
C,0,-0.4972833284,-0.4102167355,-1.3405759111
C,0,-0.4321650583,0.3942409127,-0.0359154405
C,0,0.8227644296,0.1116341623,0.8059376694
B,0,1.058580496,0.957946926,2.0956186059
C,0,-1.755765719,-0.1331635655,-2.172803429
H,0,-0.4448156838,-1.4843702601,-1.1068314506
H,0,-1.3273892605,0.1785315333,0.5649332321
H,0,1.73108442,0.3142144684,0.1998801356
H,0,0.9133404647,-0.9636644643,1.0294756632
H,0,1.9157421549,0.640798819,2.8715377506
H,0,-0.4799020632,1.4680725738,-0.2657426528
H,0,0.4362325124,1.9586906345,2.3200067364
H,0,0.395436191,-0.1900467153,-1.9457153834
C,0,-1.8188694823,-0.9376164723,-3.4777296546
H,0,-2.647155866,-0.3542121259,-1.566851915
H,0,-1.8075262346,0.9409100561,-2.4057377318
C,0,-3.077848771,-0.6601929719,-4.3094918157
H,0,-0.9275752194,-0.716546944,-4.0837863712
H,0,-1.7672403656,-2.0116813305,-3.2447107797
C,0,-3.141015117,-1.4646017773,-5.6143894145
H,0,-3.9691212901,-0.8812306562,-3.7034121632
H,0,-3.1294272602,0.4138794762,-4.5424322051
C,0,-4.3999883592,-1.1871405978,-6.4461452599
H,0,-2.2497047532,-1.2435730611,-6.2204438419
H,0,-3.0894472268,-2.53868231,-5.3814092936
C,0,-4.463242098,-1.9915161423,-7.7510511091
H,0,-5.2912815451,-1.4081502314,-5.8400844696
H,0,-4.4515439973,-0.1130710936,-6.6791153016
C,0,-5.7219693591,-1.7143645602,-8.5829585978
H,0,-3.5719175421,-1.7704626309,-8.357122653
H,0,-4.4116501699,-3.0656179309,-7.5180488438
C,0,-5.7855883782,-2.5184375869,-9.8879626768
H,0,-6.6139852485,-1.9356263589,-7.9777955478
H,0,-5.7742747216,-0.6403760411,-8.8168082267
C,0,-7.0457214562,-2.2355735397,-10.7122706507
H,0,-4.8948787774,-2.2968967482,-10.4931488471
H,0,-5.7340682145,-3.5915449795,-9.6545569027
H,0,-7.0610451514,-2.8243673161,-11.6366577184
H,0,-7.952969022,-2.4807284284,-10.1462387865
H,0,-7.1075119285,-1.1763179866,-10.991108748
```

Dodecene AM Product out gauche

B3LYP/6-31G*
E(RB3LYP) = -498.395226700

Zero-point correction= 0.371466 (Hartree/Particle)
Thermal correction to Energy= 0.389593
Thermal correction to Enthalpy= 0.390537
Thermal correction to Gibbs Free Energy= 0.322994
Sum of electronic and ZPE= -498.023761
Sum of electronic and thermal Energies= -498.005634
Sum of electronic and thermal Enthalpies= -498.004690
Sum of electronic and thermal Free Energies= -498.072233

```
     E      CV       S
  KCal/Mol Cal/Mol-K Cal/Mol-K
Total 244.473 63.982 142.156
```

```
C,0,-0.069353893,-0.0539527698,-0.8655846333
C,0,-0.2919939611,-0.5132096511,0.5816090609
C,0,0.9465524953,-0.3034425249,1.4687593614
B,0,1.3550939353,1.1571991332,1.8344469775
C,0,-1.2918969348,-0.2534721206,-1.7697875273
H,0,0.7892192011,-0.5966374313,-1.2895121755
H,0,-0.5831605437,-1.5734469427,0.5793341354
H,0,1.8033755288,-0.8829547633,1.0930497923
H,0,0.7629945199,-0.7403186213,2.4738996163
H,0,0.5772480709,2.0663553684,1.7426973937
H,0,-1.1416043773,0.0388414931,1.007540861
H,0,2.4395841546,1.3872791107,2.2903732867
H,0,0.2137279918,1.0100038479,-0.8667570209
C,0,-1.0668935362,0.2040767879,-3.2168493612
H,0,-1.5760706409,-1.3164263332,-1.7651815978
H,0,-2.1487869478,0.2908352297,-1.3457471048
C,0,-2.2901503189,0.0033158145,-4.1207910581
H,0,-0.7828856574,1.2670024407,-3.2213307539
H,0,-0.209464024,-0.3399965931,-3.6403887107
C,0,-2.065694086,0.4608157458,-5.5678950186
H,0,-2.5741735343,-1.0596533306,-4.1161669942
H,0,-3.1475439991,0.5472488997,-3.6970819626
C,0,-3.2888541167,0.2595421551,-6.4718598279
H,0,-1.7820114335,1.5238553374,-5.5725015471
H,0,-1.2080234067,-0.0828418284,-5.991482561
C,0,-3.0646662299,0.7170678195,-7.9189956276
H,0,-3.5724854706,-0.8035277396,-6.4672274208
H,0,-4.1465232129,0.8031147336,-6.0482046006
C,0,-4.287457372,0.515758051,-8.8231594361
H,0,-2.7811791009,1.7801981326,-7.9236104712
H,0,-2.2068637358,0.1735872164,-8.3426125809
C,0,-4.063799399,0.9731991827,-10.2703367355
H,0,-4.5714781704,-0.5473556225,-8.8195486479
H,0,-5.1458362556,1.0592647353,-8.4004824438
C,0,-5.2898807452,0.7686766762,-11.1660564555
H,0,-3.7807114241,2.0355226382,-10.2742722552
H,0,-3.2067672596,0.4296977174,-10.6931793878
H,0,-5.0980815789,1.1046532492,-12.1915992215
H,0,-5.576383966,-0.2894159087,-11.2103949658
H,0,-6.1544815654,1.3285616646,-10.78845038
```

Dodecene AM Product anti

B3LYP/6-31G*
E(RB3LYP) = -498.392460367

Zero-point correction= 0.371865 (Hartree/Particle)
Thermal correction to Energy= 0.390337
Thermal correction to Enthalpy= 0.391282
Thermal correction to Gibbs Free Energy= 0.321135
Sum of electronic and ZPE= -498.020595
Sum of electronic and thermal Energies= -498.002123
Sum of electronic and thermal Enthalpies= -498.001179
Sum of electronic and thermal Free Energies= -498.071325

E CV S
KCal/Mol Cal/Mol-K Cal/Mol-K
Total 244.940 63.825 147.636

6	4.060443	-1.018098	0.026312
6	5.418154	-0.394639	-0.347276
6	5.793427	0.859829	0.500398
5	7.230662	1.303779	0.070157
1	3.947310	-1.956686	-0.535574
6	2.843815	-0.126137	-0.250704
1	5.412040	-0.129011	-1.413450
1	5.725119	0.594175	1.563503
1	5.052863	1.645634	0.301674
1	7.395534	1.922957	-0.943961
1	6.192704	-1.164750	-0.227411
1	8.203166	0.979678	0.692191
1	4.074202	-1.298311	1.089639
6	1.508112	-0.811688	0.067965
1	2.917909	0.800187	0.335224
1	2.852386	0.179397	-1.307947
6	0.285037	0.073767	-0.203825
1	1.424630	-1.737200	-0.521029
1	1.503223	-1.123244	1.123167
6	-1.050549	-0.610161	0.115922
1	0.368927	0.999213	0.385282
1	0.289890	0.385415	-1.258983
6	-2.273619	0.275404	-0.155551
1	-1.134618	-1.535685	-0.473050
1	-1.055131	-0.921743	1.171122
6	-3.609363	-0.407719	0.165255
1	-2.189037	1.201194	0.432941
1	-2.269322	0.586535	-1.210879
6	-4.832455	0.477647	-0.105747
1	-3.694070	-1.333520	-0.423268
1	-3.613479	-0.718885	1.220620
6	-6.168432	-0.204814	0.215295
1	-4.748668	1.403751	0.482634
1	-4.829572	0.788913	-1.161195
6	-7.384346	0.686421	-0.057991
1	-6.252756	-1.129683	-0.373100
1	-6.171761	-0.515326	1.269933
1	-8.321537	0.170671	0.180871
1	-7.346588	1.603649	0.542719
1	-7.428012	0.984924	-1.112773

Dodecene AM Product twisted

B3LYP/6-31G*
E(RB3LYP) = -498.394245935

Zero-point correction= 0.371463 (Hartree/Particle)
Thermal correction to Energy= 0.389642
Thermal correction to Enthalpy= 0.390586
Thermal correction to Gibbs Free Energy= 0.322796
Sum of electronic and ZPE= -498.022783
Sum of electronic and thermal Energies= -498.004604
Sum of electronic and thermal Enthalpies= -498.003660
Sum of electronic and thermal Free Energies= -498.071450

E CV S
KCal/Mol Cal/Mol-K Cal/Mol-K
Total 244.504 64.027 142.676

C,0,-0.603390437,-0.2567646407,-1.6368105009
C,0,-0.6319326232,0.4656726555,-0.2792197519
C,0,0.6971193356,0.4432177927,0.4961908074
B,0,0.7294860967,1.0784128477,1.9199764267
H,0,-1.5452948054,-0.0515450853,-2.1655593857
C,0,-0.4091055001,-1.7768527234,-1.5505247242
H,0,-1.423764545,0.0323188895,0.3482213701
H,0,1.5233096988,0.8199126598,-0.1271112759
H,0,0.9969576637,-0.605071534,0.706562365
H,0,-0.2768476754,1.2893468717,2.5380870286
H,0,-0.923808005,1.5113320877,-0.4469340258
H,0,1.7780046259,1.313125539,2.4516638982
H,0,0.1938855966,0.1760364472,-2.2600489814
C,0,-0.4359453565,-2.4711114252,-2.9188372431
H,0,0.5449443461,-2.0057414731,-1.0547469522
H,0,-1.1952905139,-2.2053009629,-0.9104386719
C,0,-0.2435582706,-3.9911214527,-2.8398932316
H,0,-1.391151797,-2.2512419106,-3.418532876
H,0,0.346996401,-2.0389674155,-3.5597843976
C,0,-0.2693323528,-4.6849753189,-4.2080106659
H,0,0.7117173483,-4.2106314048,-2.3399600753
H,0,-1.0265537647,-4.4232511318,-2.1989955336
C,0,-0.0776094319,-6.2050975324,-4.1287939229
H,0,-1.2241750364,-4.4651627119,-4.7085021568
H,0,0.5141920049,-4.2532667235,-4.848543672
C,0,-0.1013047817,-6.8990259592,-5.4968893621
H,0,0.8766492318,-6.424806475,-3.6270886955
H,0,-0.8618618569,-6.6369023387,-3.4892256336
C,0,0.0897523986,-8.4190616895,-5.4178067832
H,0,-1.0552895157,-6.678902595,-5.9989759247
H,0,0.6834216952,-6.4675644013,-6.1361692252
C,0,0.0669631371,-9.1134418437,-6.7856152957
H,0,1.0436340973,-8.6401484049,-4.9156943357
H,0,-0.6951580475,-8.8516226418,-4.7793119931
C,0,0.2585433763,-10.6309749124,-6.6971542154
H,0,-0.8862842048,-8.8934045224,-7.2872027057
H,0,0.8516219979,-8.6819884107,-7.4234575618
H,0,0.2372545631,-11.0965626123,-7.6892069219
H,0,1.2195782909,-10.8827298322,-6.2317230255
H,0,-0.5313653846,-11.0957697048,-6.0941611723

Dodecene M Product out

B3LYP/6-31G*
E(RB3LYP) = -498.392581165

Zero-point correction= 0.371668 (Hartree/Particle)
Thermal correction to Energy= 0.389911
Thermal correction to Enthalpy= 0.390855
Thermal correction to Gibbs Free Energy= 0.323416
Sum of electronic and ZPE= -498.020913

Sum of electronic and thermal Energies= -498.002670
Sum of electronic and thermal Enthalpies= -498.001726
Sum of electronic and thermal Free Energies= -498.069165

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 244.673 64.420 141.938
```

```
 C,0,-0.0930899827,0.1777187104,-1.1420279436
 C,0,-0.1855330434,0.1290249745,0.3932879494
 C,0,1.149259828,0.6024493297,1.0474949948
 B,0,-0.4052222545,-1.2913118551,1.0138722445
 C,0,-1.4051000709,-0.2012233825,-1.8424500429
 H,0,0.2076332473,1.1868155323,-1.4642471341
 H,0,-0.9645839765,0.8302790348,0.7281859939
 H,0,1.3974324798,1.6119499915,0.6967960131
 H,0,1.0867061681,0.6372444584,2.1412038137
 H,0,0.0271775357,-2.2692232511,0.468363184
 H,0,1.9877405691,-0.0514523035,0.7774037275
 H,0,-0.9189749189,-1.4247138451,2.0886446421
 H,0,0.7046080299,-0.5016074978,-1.4773821661
 C,0,-1.3261762918,-0.1356633744,-3.3729448693
 H,0,-1.6985073706,-1.2184985759,-1.5395942323
 H,0,-2.2089224445,0.4633423134,-1.4914659995
 C,0,-2.6382187751,-0.5136109831,-4.0726546337
 H,0,-1.0323322248,0.880267726,-3.6764659914
 H,0,-0.522826388,-0.8006169665,-3.7232583814
 C,0,-2.5588603174,-0.4445380949,-5.6032694655
 H,0,-2.9312754257,-1.5299675273,-3.7699514911
 H,0,-3.4417127775,0.1505054052,-3.7206262743
 C,0,-3.8704978696,-0.822793894,-6.3035317218
 H,0,-2.2660213822,0.5719614836,-5.9058323569
 H,0,-1.755054021,-1.1082463661,-5.9552570782
 C,0,-3.7914657428,-0.7523798885,-7.8340980967
 H,0,-4.1628736869,-1.8395888596,-6.0016180999
 H,0,-4.6744945365,-0.1596424696,-5.9508513283
 C,0,-5.1026412773,-1.1307278871,-8.5347995241
 H,0,-3.4992434989,0.26455903,-8.1358872264
 H,0,-2.9872782024,-1.4153296479,-8.1867955118
 C,0,-5.0242063008,-1.059840591,-10.0653064331
 H,0,-5.3953129879,-2.1479974434,-8.2342217334
 H,0,-5.9075894869,-0.4681906948,-8.1826962326
 C,0,-6.3377505748,-1.4393352146,-10.7568059617
 H,0,-4.7326656883,-0.0433150839,-10.3659100052
 H,0,-4.2204980068,-1.7222153148,-10.4174241793
 H,0,-6.2490097027,-1.378698308,-11.8476065352
 H,0,-6.6374456347,-2.4638356742,-10.5035255253
 H,0,-7.1535399951,-0.7723629949,-10.4514803876
```

Dodecene M Product anti

B3LYP/6-31G*
E(RB3LYP) = -498.391030888

Zero-point correction= 0.371618 (Hartree/Particle)
Thermal correction to Energy= 0.389954
Thermal correction to Enthalpy= 0.390899
Thermal correction to Gibbs Free Energy= 0.323100
Sum of electronic and ZPE= -498.019412
Sum of electronic and thermal Energies= -498.001076
Sum of electronic and thermal Enthalpies= -498.000132
Sum of electronic and thermal Free Energies= -498.067931

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
Total 244.700 64.511 142.694
```

```
 C,0,-0.3354962949,0.1515342907,-1.1977726974
 C,0,-0.3386649082,-0.1140827407,0.3229756865
 C,0,0.9899244094,0.3367316753,1.0071793087
 B,0,-0.4464333493,-1.622216673,0.7275937622
 H,0,-1.2319100932,-0.3075489354,-1.6401090675
 C,0,-0.3044035077,1.6325069474,-1.604158461
 H,0,-1.1460018543,0.4712495563,0.7887928934
 H,0,1.8577986954,-0.1766219676,0.5746464629
 H,0,1.1466483875,1.4132452407,0.8778625686
 H,0,0.0136779915,-2.4803353688,0.0261016723
 H,0,0.9842421788,0.137898335,2.0855326585
 H,0,-0.9119071975,-1.9444402549,1.7843305703
 H,0,0.5196322598,-0.3718624514,-1.6507593929
 C,0,-0.3940193416,1.8436027238,-3.1217221842
 H,0,-1.1381157872,2.1591430761,-1.1153951576
 H,0,0.6144255725,2.1070357358,-1.2329342722
 C,0,-0.3637631919,3.3195108862,-3.5395337848
 H,0,0.4345212315,1.3107338114,-3.6117295391
 H,0,-1.3171825651,1.3785205196,-3.4985805448
 C,0,-0.4595574496,3.5295511376,-5.0563251391
 H,0,-1.1899481471,3.8532718138,-3.0464784012
 H,0,0.5614734414,3.7836731712,-3.1665863818
 C,0,-0.4287284556,5.0052079771,-5.475036274
 H,0,0.365982619,2.9950355962,-5.5495983953
 H,0,-1.3852332288,3.0659863846,-5.4288337107
 C,0,-0.5275015991,5.2149128777,-6.9916458068
 H,0,-1.2532354635,5.5400706885,-4.9803856904
 H,0,0.4977946498,5.4684266679,-5.1041615498
 C,0,-0.4961593219,6.6901637003,-7.4111021796
 H,0,0.296601059,4.6795190811,-7.4864509603
 H,0,-1.4543725228,4.7520657916,-7.3621865434
 C,0,-0.5963408599,6.9001747322,-8.9274978249
 H,0,-1.3197821233,7.2266011087,-6.9163718699
 H,0,0.4311347499,7.1536623372,-7.0420984843
 C,0,-0.5638273349,8.376115233,-9.3375014313
 H,0,0.2266604838,6.3646841468,-9.4220619837
 H,0,-1.5233090876,6.4381760401,-9.2962179075
 H,0,-0.6373227152,8.4921536569,-10.4249435669
 H,0,-1.3956554323,8.9312495195,-8.8864111606
 H,0,0.3674411031,8.8570919317,-9.0134842207
```

3,3-dimethyl-2-butene

B3LYP/6-31G*
E(RB3LYP) = -235.848211652

Zero-point correction= 0.165444 (Hartree/Particle)
Thermal correction to Energy= 0.173116
Thermal correction to Enthalpy= 0.174060
Thermal correction to Gibbs Free Energy= 0.134831
Sum of electronic and ZPE= -235.682768
Sum of electronic and thermal Energies= -235.675096
Sum of electronic and thermal Enthalpies= -235.674152
Sum of electronic and thermal Free Energies= -235.713381

```
      E      CV       S
   KCal/Mol Cal/Mol-K Cal/Mol-K
```

Total 108.632 28.798 82.565

C,0,-0.4118512738,-0.4450293739,0.0317746888
C,0,0.8112195042,0.3311164217,-0.4221043473
C,0,1.8568817256,0.714319284,0.3115216974
H,0,0.8015506325,0.5911387846,-1.4833395294
H,0,1.9415536164,0.4950576989,1.372267287
H,0,2.6796500852,1.2705265106,-0.1301827223
C,0,-1.6650530226,0.4137943536,-0.2569544839
H,0,-1.7271419869,0.6815963649,-1.3188038321
H,0,-1.6468081741,1.3434104137,0.3230708367
C,0,-0.365277198,-0.7937895693,1.528022037
H,0,-0.3199045294,0.1086112568,2.1484062218
H,0,-1.2633862472,-1.3524371007,1.8161761064
H,0,0.5054993758,-1.4147548984,1.7676318315
C,0,-0.493796756,-1.7478779604,-0.7972761013
H,0,-0.5270955633,-1.5332126005,-1.8724072527
H,0,0.375285038,-2.3885669142,-0.6097586188
H,0,-1.397827556,-2.31409523,-0.5409194372
H,0,-2.5789216706,-0.1342674414,0.0039386183

3,3-dimethyl-2-butene AM complex

B3LYP/6-31G*
E(RB3LYP) = -262.478426404

Zero-point correction= 0.197689 (Hartree/Particle)
Thermal correction to Energy= 0.207474
Thermal correction to Enthalpy= 0.208418
Thermal correction to Gibbs Free Energy= 0.164472
Sum of electronic and ZPE= -262.280737
Sum of electronic and thermal Energies= -262.270953
Sum of electronic and thermal Enthalpies= -262.270009
Sum of electronic and thermal Free Energies= -262.313954

C	-0.341726	-0.564176	0.029673
C	0.939852	0.125302	-0.431458
C	1.749246	0.943784	0.331116
B	2.668963	-0.629369	0.052094
H	1.039454	0.198573	-1.513543
H	1.562475	1.074855	1.390883
H	2.361066	1.705001	-0.138226
H	2.045066	-1.559537	-0.421074
H	2.907120	-0.905311	1.196316
H	3.545572	-0.310655	-0.701265
C	-1.470032	0.470741	-0.217988
H	-1.521866	0.766208	-1.272850
H	-1.317397	1.375388	0.380566
C	-0.315334	-0.947939	1.518790
H	-0.204467	-0.071869	2.167315
H	-1.257728	-1.436909	1.790127
H	0.503563	-1.640347	1.735155
C	-0.620814	-1.814826	-0.825796
H	-0.618093	-1.574177	-1.896051
H	0.130263	-2.591621	-0.652409
H	-1.604380	-2.230941	-0.579866
H	-2.439506	0.038493	0.055625

3,3-dimethyl-2-butene M complex

B3LYP/6-31G*
E(RB3LYP) = -262.478425589

Zero-point correction= 0.197670 (Hartree/Particle)
Thermal correction to Energy= 0.207459
Thermal correction to Enthalpy= 0.208404
Thermal correction to Gibbs Free Energy= 0.164444
Sum of electronic and ZPE= -262.280756
Sum of electronic and thermal Energies= -262.270966
Sum of electronic and thermal Enthalpies= -262.270022
Sum of electronic and thermal Free Energies= -262.313982

```
     E       CV        S
  KCal/Mol Cal/Mol-K Cal/Mol-K
Total 130.183 37.113 92.522
```

H	-0.133388	-0.021587	0.163865
C	0.000763	-0.036562	1.239487
C	1.259507	0.085687	1.794613
B	0.156208	1.687657	1.868020
H	-0.813365	-0.466437	1.810850
C	2.584416	0.114770	1.036577
H	1.359496	-0.150771	2.853056
H	-0.582987	1.707940	2.811833
H	-0.112261	2.279086	0.858110
H	1.263469	2.051151	2.215803
C	3.661189	0.850061	1.857388
H	3.747179	0.431326	2.867651
H	4.639192	0.754388	1.372199
H	3.431190	1.915773	1.951949
C	2.465788	0.754451	-0.357024
H	1.771095	0.206425	-1.003124
H	2.125122	1.791823	-0.289321
H	3.444246	0.746887	-0.850343
C	2.999223	-1.372402	0.889549
H	3.968107	-1.441085	0.381478
H	3.098724	-1.861743	1.865954
H	2.264253	-1.933500	0.302248

3,3-dimethyl-2-butene AMTS

B3LYP/6-31G*
E(RB3LYP) = -262.478350849

Zero-point correction= 0.197457 (Hartree/Particle)
Thermal correction to Energy= 0.206451
Thermal correction to Enthalpy= 0.207395
Thermal correction to Gibbs Free Energy= 0.165186
Sum of electronic and ZPE= -262.280894
Sum of electronic and thermal Energies= -262.271900
Sum of electronic and thermal Enthalpies= -262.270956
Sum of electronic and thermal Free Energies= -262.313165

```
     E       CV        S
  KCal/Mol Cal/Mol-K Cal/Mol-K
Total 129.550 34.954 88.837
```

C	-0.39371	-0.3548	-1.24716
C	-0.37206	-0.41788	1.64054
C	1.76587	-0.49423	3.18391
5	0.37322	2.53332	3.5421
1	-2.12699	-1.11572	2.4592
1	3.63413	-0.33215	2.35815
1	1.67915	-1.46015	4.98825
1	-0.46066	2.58971	5.64758
1	-1.43614	2.87505	2.12665
1	1.97682	3.96792	2.83343
C	-0.25576	-3.16187	-2.0728

1	1.50014	-4.04808	-1.42806
1	-1.84364	-4.25795	-1.31614
C	1.8578	1.09793	-2.37268
1	1.85774	3.0733	-1.76398
1	3.67103	0.25595	-1.8358
1	1.74645	1.05837	-4.44019
C	-2.88661	0.79225	-2.21896
1	-4.52967	-0.18317	-1.415
1	-3.03167	2.79779	-1.7306
1	-3.00064	0.62581	-4.27997
1	-0.32598	-3.29358	-4.13876

3,3-dimethyl-2-butene MTS

B3LYP/6-31G*
E(RB3LYP) = -262.474866098

Zero-point correction= 0.197690 (Hartree/Particle)
Thermal correction to Energy= 0.206531
Thermal correction to Enthalpy= 0.207476
Thermal correction to Gibbs Free Energy= 0.165730
Sum of electronic and ZPE= -262.277176
Sum of electronic and thermal Energies= -262.268335
Sum of electronic and thermal Enthalpies= -262.267391
Sum of electronic and thermal Free Energies= -262.309136

E CV S
KCal/Mol Cal/Mol-K Cal/Mol-K
Total 129.600 34.617 87.861

1	-2.033963	-0.064144	-1.136644
C	-1.958249	-0.201199	-0.062333
C	-0.705233	-0.187853	0.581555
B	-1.523150	1.331853	0.834836
H	-2.813391	-0.686123	0.398522
C	0.652810	-0.205156	-0.141469
H	-0.671570	-0.692879	1.545011
H	-1.828534	1.397597	1.989630
H	-2.603155	1.355258	0.218273
H	-0.887566	2.173905	0.270853
C	1.695389	0.534048	0.720030
H	1.740610	0.112552	1.731828
H	2.695043	0.448058	0.277121
H	1.448315	1.596247	0.810398
C	0.604249	0.425671	-1.543490
H	-0.049895	-0.134260	-2.222942
H	0.257777	1.463390	-1.508358
H	1.606654	0.422015	-1.986687
C	1.073878	-1.687916	-0.274806
H	2.057416	-1.770086	-0.753125
H	1.139231	-2.173359	0.706621
H	0.354519	-2.251451	-0.881078

3,3-dimethyl-2-butene AM product

B3LYP/6-31G*
E(RB3LYP) = -262.509891549

Zero-point correction= 0.199480 (Hartree/Particle)
Thermal correction to Energy= 0.209518
Thermal correction to Enthalpy= 0.210462
Thermal correction to Gibbs Free Energy= 0.165119
Sum of electronic and ZPE= -262.310411
Sum of electronic and thermal Energies= -262.300374
Sum of electronic and thermal Enthalpies= -262.299430
Sum of electronic and thermal Free Energies= -262.344773

E CV S
KCal/Mol Cal/Mol-K Cal/Mol-K
Total 131.474 36.376 95.432

C	-0.036435	0.137150	-0.178928
C	-0.032264	0.697700	1.273493
C	1.238084	0.401988	2.128363
B	0.986896	1.066069	3.522662
H	-0.916406	0.296252	1.789171
H	2.115122	0.821353	1.620954
H	1.372508	-0.683723	2.203957
H	0.422132	0.471064	4.397482
H	-0.181605	1.785253	1.212483
H	1.280944	2.211293	3.723405
C	0.068410	-1.399048	-0.175496
H	1.024406	-1.742412	0.234704
H	-0.735316	-1.850466	0.419741
C	1.119674	0.734511	-1.002011
H	1.078082	1.831006	-1.007332
H	2.097746	0.436815	-0.608518
H	1.067717	0.394483	-2.043492
C	-1.374875	0.545378	-0.827372
H	-2.227355	0.133926	-0.272779
H	-1.487496	1.636334	-0.854035
H	-1.439142	0.177906	-1.858847
H	-0.011412	-1.793472	-1.195820

3,3-dimethyl-2-butene AM product twisted

B3LYP/6-31G*
E(RB3LYP) = -262.511424891

Zero-point correction= 0.198740 (Hartree/Particle)
Thermal correction to Energy= 0.208713
Thermal correction to Enthalpy= 0.209657
Thermal correction to Gibbs Free Energy= 0.164800
Sum of electronic and ZPE= -262.312685
Sum of electronic and thermal Energies= -262.302712
Sum of electronic and thermal Enthalpies= -262.301768
Sum of electronic and thermal Free Energies= -262.346625

E CV S
KCal/Mol Cal/Mol-K Cal/Mol-K
Total 130.969 36.679 94.410

C,0,-0.0632225984,0.1315710258,-0.1939656446
C,0,-0.1222358206,0.6807507604,1.2570829676
C,0,1.087068551,0.4079087244,2.1684142277
B,0,1.099790024,1.0864480866,3.5730834756
H,0,-1.021739135,0.2694609083,1.7378507555
H,0,2.0132957542,0.8144626352,1.7109745806
H,0,1.3005149779,-0.6670422946,2.2576933978
H,0,0.3764209575,2.0062484656,3.8374548867
H,0,-0.2839215381,1.7670555984,1.2076801681
H,0,1.8877951613,0.7310029238,4.4037570072
C,0,0.0598752139,-1.40465383,-0.1958454527
H,0,0.9995907642,-1.7407005949,0.2568500954
H,0,-0.7653726597,-1.8688570566,0.3589570855
C,0,1.1225820011,0.7442775741,-0.9637549307

H,0,1.0682238534,1.840365093,-0.9687496738
H,0,2.0863601607,0.4573900539,-0.5274668438
H,0,1.1241792819,0.4068456371,-2.0074679977
C,0,-1.3738867509,0.526014123,-0.9035558607
H,0,-2.2459950264,0.103986256,-0.3886405985
H,0,-1.4977115658,1.6158276817,-0.9311949125
H,0,-1.3884313029,0.161712288,-1.9381906206
H,0,0.0323276968,-1.7950440593,-1.220610112

3,3-dimethyl-2-butene M Product

B3LYP/6-31G*
E(RB3LYP) = -262.506415301

Zero-point correction= 0.199209 (Hartree/Particle)
Thermal correction to Energy= 0.209123
Thermal correction to Enthalpy= 0.210068
Thermal correction to Gibbs Free Energy= 0.166035
Sum of electronic and ZPE= -262.307206
Sum of electronic and thermal Energies= -262.297292
Sum of electronic and thermal Enthalpies= -262.296348
Sum of electronic and thermal Free Energies= -262.340380

```
   E     CV       S
 KCal/Mol Cal/Mol-K Cal/Mol-K
Total 131.227 37.015 92.675
```

H	-0.242538	0.482038	0.128128
C	-0.065528	-0.071344	1.057390
C	1.210412	0.420409	1.812566
B	0.863528	1.916851	2.123504
H	0.020854	-1.131664	0.795365
C	2.556949	0.132298	1.073262
H	1.233832	-0.147187	2.756072
H	0.131112	2.179125	3.035730
H	-0.959443	0.036135	1.682131
H	1.223787	2.822197	1.425950
C	3.695987	0.763468	1.901162
H	3.708967	0.366380	2.924272
H	4.675037	0.556083	1.451732
H	3.586795	1.853943	1.963976
C	2.582077	0.724567	-0.349175
H	1.832713	0.256150	-0.997545
H	2.396349	1.803948	-0.343170
H	3.562505	0.559271	-0.812488
C	2.808024	-1.387375	0.987012
H	3.786890	-1.593703	0.536681
H	2.797071	-1.848001	1.982998
H	2.055605	-1.895845	0.373548

3,3-dimethyl-2-butene M product Twisted further than the structure above

B3LYP/6-31G*
E(RB3LYP) = -262.506000861

Zero-point correction= 0.198932 (Hartree/Particle)
Thermal correction to Energy= 0.208835
Thermal correction to Enthalpy= 0.209779
Thermal correction to Gibbs Free Energy= 0.165325
Sum of electronic and ZPE= -262.307069
Sum of electronic and thermal Energies= -262.297166
Sum of electronic and thermal Enthalpies= -262.296222
Sum of electronic and thermal Free Energies= -262.340676

```
   E     CV       S
 KCal/Mol Cal/Mol-K Cal/Mol-K
Total 131.046 36.943 93.563
```

H,0,-0.190210443,0.0502965361,0.0546302282
C,0,-0.0489086835,-0.2362545166,1.1033596582
C,0,1.167541137,0.4883756227,1.734750188
B,0,0.8480067987,1.9960735746,2.02140955
H,0,0.0549616751,-1.3254025032,1.1341532294
C,0,2.5476000413,0.1630864315,1.0684472512
H,0,1.2271014582,0.1056058746,2.7790693717
H,0,-0.2409905762,2.2843339576,2.4331554661
H,0,-0.9694187848,0.0245280811,1.6360032452
H,0,1.6414765013,2.8846507253,1.9030182444
C,0,3.6860695712,0.6889799236,1.9664488093
H,0,3.665072958,0.2073594335,2.9528275063
H,0,4.6634319517,0.4769489815,1.5160928153
H,0,3.62008022,1.7711031631,2.1195069752
C,0,2.6509488942,0.8302833388,-0.3182935254
H,0,1.8645435634,0.4791963731,-0.9968811531
H,0,2.5713095053,1.9212996263,-0.2430173484
H,0,3.6147590708,0.6004509505,-0.7890256434
C,0,2.7406678612,-1.3599152379,0.9065360973
H,0,3.7527501983,-1.5813340324,0.5462251762
H,0,2.6079178092,-1.8822349675,1.8628439079
H,0,2.0374442727,-1.7921733364,0.1868949506

References

1. Gaussian 09, Revision A.02, Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G. A.; Nakatsuji, H.; Caricato, M.; Li, X.; Hratchian, H. P.; Izmaylov, A. F.; Bloino, J.; Zheng, G.; Sonnenberg, J. L.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Montgomery, Jr., J. A.; Peralta, J. E.; Ogliaro, F.; Bearpark, M.; Heyd, J. J.; Brothers, E.; Kudin, K. N.; Staroverov, V. N.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Rega, N.; Millam, N. J.; Klene, M.; Knox, J. E.; Cross, J. B.; Bakken, V.; Adamo, C.; Jaramillo, J.; Gomperts, R.; Stratmann, R. E.; Yazyev, O.; Austin, A. J.; Cammi, R.; Pomelli, C.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Zakrzewski, V. G.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Dapprich, S.; Daniels, A. D.; Farkas, Ö.; Foresman, J. B.; Ortiz, J. V.; Cioslowski, J.; Fox, D. J. Gaussian, Inc., Wallingford CT, 2009.

2. Gaussian 09, Revision B.01, Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G. A.; Nakatsuji, H.; Caricato, M.; Li, X.; Hratchian, H. P.; Izmaylov, A. F.; Bloino, J.; Zheng, G.; Sonnenberg, J. L.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Montgomery, Jr., J. A.; Peralta, J. E.; Ogliaro, F.; Bearpark, M.; Heyd, J. J.; Brothers, E.; Kudin, K. N.; Staroverov, V. N.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Rega, N.; Millam, N. J.; Klene, M.; Knox, J. E.; Cross, J. B.; Bakken, V.; Adamo, C.; Jaramillo, J.; Gomperts, R.; Stratmann, R. E.; Yazyev, O.; Austin, A. J.; Cammi, R.; Pomelli, C.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Zakrzewski, V. G.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Dapprich, S.; Daniels, A. D.; Farkas, Ö.; Foresman, J. B.; Ortiz, J. V.; Cioslowski, J.; Fox, D. J. Gaussian, Inc., Wallingford CT, 2009.

3. Gaussian 09, Revision D.01, Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G. A.; Nakatsuji, H.; Caricato, M.; Li, X.; Hratchian, H. P.; Izmaylov, A. F.; Bloino, J.; Zheng, G.; Sonnenberg, J. L.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Montgomery, J. A., Jr.; Peralta, J. E.; Ogliaro, F.; Bearpark, M.; Heyd, J. J.; Brothers, E.; Kudin, K. N.; Staroverov, V. N.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Rega, N.; Millam, N. J.; Klene, M.; Knox, J. E.; Cross, J. B.; Bakken, V.; Adamo, C.; Jaramillo, J.; Gomperts, R.; Stratmann, R. E.; Yazyev, O.; Austin, A. J.; Cammi, R.; Pomelli, C.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Zakrzewski, V. G.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Dapprich, S.; Daniels, A. D.; Farkas, Ö.; Foresman, J. B.; Ortiz, J. V.; Cioslowski, J.; Fox, D. J. Gaussian, Inc., Wallingford CT, 2009.

4. Singleton, D. A.; Hang, C.; Szymanski, M. J.; Greenwald, E. E. *J. Am. Chem. Soc.* **2003**, 125, 1176-1177.

5. MOPAC2016, James J. P. Stewart, Stewart Computational Chemistry, Colorado Springs, CO, USA, HTTP://OpenMOPAC.net (2016).

6. J. Zheng, S. Zhang, B. J. Lynch, J. C. Corchado, Y.-Y. Chuang, P. L. Fast, W.-P. Hu, Y.-P. Liu, G. C. Lynch, K. A. Nguyen, C. F. Jackels, A. Fernandez Ramos, B. A. Ellingson, V. S.

Melissas, J. Villà, I. Rossi, E. L. Coitiño, J. Pu, T. V. Albu, R. Steckler, B. C. Garrett, A. D. Isaacson, and D. G. Truhlar, POLYRATE–version 2010, University of Minnesota, Minneapolis, 2010.

7. Martinez, L.; Andrade, R.; Birgin, E. G.; Martinez, J. M. *J. Comput. Chem.* **2009,** 2157-2164.

8. Chen, C.; Banaszak, M.; Radosz, M. *J. Phys. Chem. B*, **1998**, *13*, 2427-2431

9.Goldman, S. *J. Phys. Chem.*, **1976**, *15*, 1697-1700

10. Richardi, J.; Fries, P. H.; Kriebke, H. *J. Phys. Chem. B*, **1998**, *102*, 5196-5201