SUPPORTING INFORMATION

Isotope-Induced Desymmetrization Can Mimic Isotopic Perturbation of Equilibria. On the Symmetry of Bromonium Ions and Hydrogen Bonds

Xavier S. Bogle and Daniel A. Singleton*

Department of Chemistry, Texas A&M University, PO Box 30012, College Station, Texas 77842

singleton@mail.chem.tamu.edu

Computational Procedures and Supporting Computational Results

General

Calculations of structures, energies, and frequencies employed standard procedures in Gaussian09 (S1) unless otherwise noted. Most structures were optimized with tight convergence criteria (opt=tight in Gaussian09) instead of the default criteria. Complete structures and energetics are provided in a section below.

The program suite PROGDYN used for dynamics is listed at the end of the Supporting Information as a series of component programs as either Unix shell scripts or awk programs. Gaussian09 was used to calculate the forces at each point in trajectories. The latest version of this program can be obtained by emailing Daniel Singleton at singleton@mail.chem.tamu.edu. The original version of this program was published in the Supporting Information of a previous paper (S2).

The awk program progaveragetraj, listed at the end of the Supporting Information, was used to average the trajectory geometries for **3** in curvilinear coordinates. The awk program progNMRt, listed at the end of the Supporting Information, was used to average NMR results from the trajectories for **9a** and **9b**.

Numerical solutions to Schroedinger's equation in one dimension were calculated using the program NUCFUN obtained from Robert Lucchese. The program uses as input an analytical function for the energy versus displacement of the normal mode and a reduced mass. The analytical function for the energy was E (in cm^{-1}) = $61.841x + 1860.5x^2 + 754.06x^3 + 62.719x^4 - 1632x^5$ where x is the distance in Angstroms. This function was obtained by fitting a fifth-order polynomial to calculated values for the potential energy + zero-point energy at a series of displacements of the *B2* mode. The reduced mass used was 3.3883, the value for the harmonic normal mode. The average displacement of the mode was then $\langle\psi|x|\psi\rangle$, which was integrated on an Excel spreadsheet.

Initialization of Trajectories and Additional Details on Trajectories

The trajectories employed in the manuscript are quasiclassical, i. e. including zero-point energy. For trajectories starting from **3a** (L = D), **9a**, **9b**, or formic acid, the desired energy in each of the normal modes was mapped from a random number generator to a Boltzmann distribution set at the desired temperature. The phase of each of the normal modes was mapped from Gaussian distribution of random numbers. This distribution is not correct for a classical oscillator but would be approximately correct for a quantum oscillator in its ground state (only ignoring mode displacements outside of the classical limit), and has the advantage that a larger portion of the trajectories start with an energy approximating the initially desired energy. After an energy/force calculation on the initial geometry, the total initial energy was calculated and the trajectory was thrown out if the energy did not agree satisfactorily (within 1 kcal/mol) with the desired energy. (This is a variation of the conventional practice of scaling energies. (S3))

The trajectories for **9a** and **9b** were followed forward and backward in time (this is just a matter of reversing all of the initial velocities) because preliminary results had suggested that this led to faster convergence of the NMR chemical shifts. Our hypothesis that led to trying this was that extreme trajectories leading to unusual

average chemical shifts when followed forward in time would be partially counterbalanced by the trajectory going backward in time. We do not have sufficient data to fully confirm or disaffirm this hypothesis.

Calculation of Uncertainties and Aggregated Raw Data

All of the trajectory studies described provide long series of results (either geometries or NMR chemical shifts). Uncertainties in the average of results were determined by dividing the results into at least 11 independent pieces, each made up of approximately the same number of raw results. For each piece, the average was calculated, and the standard deviation of the averages from the pieces was calculated. From this standard deviation and the number of pieces, 95% confidence ranges were calculated in a standard way using a t-distribution function, as defined by IUPAC. See: http://www.iupac.org/publications/analytical_compendium/Cha02sec3.pdf.

Carbon-Bromine Distances at 0 K

Each result below is taken from approximately 30,000 trajectory geometries. Distances are in Angstroms. BrToC1 refers to the distance 1 in Figure 1 in the main text, while BrToC2 refers to the distance 2 in Figure 1 in the main text.

BrToC1	BrToC2
2.13491	2.12
2.13497	2.12072
2.13263	2.12052
2.13398	2.12067
2.13196	2.12185
2.1329	2.12191
2.13607	2.12173
2.13301	2.12138
2.13313	2.12274
2.13672	2.12113
2.13746	2.11797
2.13444	2.12057
2.13577	2.11796
2.13657	2.12064
2.13551	2.11833
2.13644	2.1192
2.13671	2.12005
2.13773	2.11854
2.13502	2.11975
2.13598	2.11843
average of results for pieces	
2.1351	2.1202
real average of all results	
2.1351	2.1202
standard deviations	
0.0017	0.0014
95% confidence range	
0.0008	0.0007

Carbon-Bromine Distances at 193 K

Each result below is taken from approximately 45,000 trajectory geometries. Distances are in Angstroms. BrToC1 refers to the distance 1 in Figure 1 in the main text, while BrToC2 refers to the distance 2 in Figure 1 in the main text.

BrToC1	BrToC2
2.14007	2.12456

2.13917	2.12539
2.1417	2.12317
2.142	2.12483
2.13779	2.12722
2.13953	2.12537
2.13628	2.12778
2.13906	2.12728
2.1411	2.12245
2.1394	2.12739
2.13897	2.12586
2.1400	2.1264
2.1409	2.1247
2.1398	2.1261
2.1407	2.1241
2.1415	2.1239
2.1395	2.1250
2.1411	2.1254
2.1407	2.1238
2.1397	2.1262
2.1399	2.1254
2.1395	2.1258
2.13851	2.12618
average of results for pieces	
2.1399	2.1254
real average of all results	
2.1398	2.1254
standard deviations	
0.0013	0.0014
95% confidence range	
0.0006	0.0006

Carbon-Bromine Distances at 298 K

Each result below is taken from approximately 41,000 trajectory geometries. Distances are in Angstroms. BrToC1 refers to the distance 1 in Figure 1 in the main text, while BrToC2 refers to the distance 2 in Figure 1 in the main text.

BrToC1	BrToC2
2.14622	2.13222
2.14465	2.13038
2.14607	2.12908
2.14327	2.13249
2.14512	2.13147
2.14414	2.13297
2.14944	2.12775
2.14547	2.13196
2.14461	2.13145
2.14869	2.12968
2.1458	2.13146
average of results for pieces	
2.1458	2.1310
real average of all results	

2.14576	2.13099
standard deviations	
0.0019	0.0016
95% confidence range	
0.0012	0.0011

O-H Distances in **9a** at 298 K

Each result below is taken from approximately 18,000 trajectory geometries, and the numbers represent the negative of the quantity (a-b) illustrated in Figure 2 in the main text, in Angstroms.

0.00477	-0.02139	-0.00450	-0.01194	0.00373	-0.00203
-0.02191	-0.01288	-0.01029	0.00068	-0.00048	0.01057
0.01393	-0.00037	-0.00137	0.00000	0.00229	-0.01318
-0.01012	0.01847	-0.00227	0.00367	-0.00112	0.00780
-0.00109	0.00477	-0.00982	0.00195	0.01040	-0.01182
-0.00954	-0.01941	-0.01472	-0.00150	0.00264	-0.00130
-0.01445	-0.00592	-0.00486	-0.01326	0.00232	0.01236
-0.00673	-0.01222	-0.01534	0.00179	-0.00236	-0.01684
0.00136	-0.01013	-0.01193	0.01086	-0.00119	-0.00709
-0.02269	0.00705	0.01515	-0.01055	0.00727	-0.02024
0.00346	-0.00304	-0.01116	-0.00029	0.01246	0.00178
0.01100	0.00783	0.00964	-0.00938	0.00407	-0.00018
-0.00312	-0.01360	-0.00227	-0.01454	-0.00379	-0.00452
-0.00134	0.01606	0.00410	0.01430	0.01282	-0.01027
0.00903	-0.01661	-0.01167	-0.00471	0.00849	average of results for pieces
-0.01987	-0.02143	-0.01012	0.00359	-0.00342	-0.00174
-0.01480	-0.00557	-0.01740	-0.00207	-0.01703	real average of all results
-0.01879	0.00784	0.01798	0.00550	-0.00117	-0.00170
-0.00063	0.00705	-0.00356	-0.00548	-0.00571	standard deviation
0.00209	0.01645	-0.00170	-0.01853	-0.01085	0.00985
0.00475	-0.00060	0.00224	0.00277	0.01200	95% confidence range
0.01987	0.00227	-0.01655	-0.00265	-0.00267	0.00119
0.00045	-0.01289	-0.00499	-0.00921	0.00482	
0.00693	0.00361	-0.00558	0.01591	0.00718	
0.00567	-0.00848	0.00362	0.00618	0.00582	
-0.00412	0.01527	-0.00327	0.00529	-0.00276	
0.00640	0.00111	0.01138	-0.00026	0.01224	
-0.01191	-0.01198	0.00463	0.00967	-0.01737	
-0.00120	0.00317	0.00134	0.00592	0.00241	
0.00688	-0.01256	0.01006	0.00005	0.00995	
-0.01259	0.01186	0.00428	-0.00920	-0.00865	
-0.01516	-0.00308	-0.00584	0.00388	0.00381	
-0.00713	0.00972	0.00533	-0.00109	0.00748	
-0.00028	-0.00985	-0.00819	0.00147	0.01559	
-0.00490	0.00873	0.02514	0.00163	0.00143	
0.01243	-0.00994	0.00684	-0.02064	-0.00264	
-0.00473	0.01888	-0.01521	0.00156	0.00384	
-0.01806	-0.00242	-0.00056	-0.00114	0.00900	
0.00491	-0.01328	-0.01756	0.00777	-0.01700	
-0.01724	0.01589	-0.00610	-0.01284	-0.00757	
0.00661	-0.00625	-0.01995	0.00495	0.01706	

0.00271	-0.00805	0.01848	0.00606	-0.00785
0.01655	-0.02010	0.01029	-0.00107	-0.01188
-0.00123	0.00358	0.00047	-0.00139	-0.00887
-0.00883	-0.01377	0.00344	-0.00580	-0.01288
0.00411	-0.00600	0.00061	-0.01211	-0.01997
-0.00904	0.00082	-0.00969	-0.00751	-0.00168
0.01364	-0.00711	0.00137	-0.01479	-0.00921
0.00507	-0.01483	-0.00355	-0.01008	-0.00921
-0.00239	0.00167	0.00186	0.00093	0.00397

O-H Distances in **9b** at 298 K

Each result below is taken from approximately 9500 trajectory geometries, and the numbers represent the negative of the quantity (a-b) illustrated in Figure 2 in the main text, in Angstroms.

-0.00224	0.00357	-0.03360	0.00583	-0.04000
-0.00797	0.02627	0.00042	-0.02869	-0.01038
-0.01522	0.00775	0.00830	-0.00723	-0.01918
-0.00588	-0.00276	0.00430	0.00771	-0.01337
0.01375	0.00590	-0.00584	-0.02375	-0.02066
-0.00748	-0.00382	-0.00030	-0.00749	-0.01904
-0.01075	-0.00353	0.01360	-0.00236	-0.00519
-0.01193	-0.00484	-0.01136	0.00599	-0.00729
-0.02059	-0.02410	-0.02130	-0.02778	0.00332
0.02256	-0.01130	-0.00346	-0.00184	-0.00418
-0.00975	-0.00305	0.00891	0.00936	-0.00105
-0.02362	-0.01095	0.01261	0.00978	-0.01462
0.02890	-0.02260	0.00110	-0.02046	0.01214
-0.02708	-0.01580	0.02177	0.01029	-0.01560
-0.00399	0.00573	0.00554	-0.00194	0.00123
0.00246	-0.00946	0.00661	-0.01252	0.00588
-0.01121	-0.01764	-0.00869	-0.00501	-0.00181
-0.00807	-0.02906	-0.01363	0.00208	0.00295
-0.01972	-0.01196	0.00353	-0.01977	-0.00787
-0.01070	-0.02332	-0.00980	0.00718	-0.02449
-0.00794	-0.02210	0.02013	0.01237	0.00685
-0.01343	0.01003	-0.00957	0.01730	-0.01518
0.00665	0.01139	0.00398	-0.00662	-0.03697
-0.00247	-0.01050	-0.01673	-0.00291	0.00469
-0.00099	-0.00415	0.01240	-0.00542	0.00180
0.01032	-0.00801	-0.03277	0.00607	0.00333
0.01600	-0.02234	0.00719	0.00252	0.02298
0.01474	-0.02839	0.00737	-0.02538	0.00396
-0.00097	-0.00771	-0.02270	-0.00018	0.00394
-0.01354	-0.00111	0.00176	-0.02190	0.00009
-0.00630	0.00940	-0.00626	0.00895	-0.00382
-0.00729	-0.02054	-0.02397	-0.01199	0.00504
-0.00093	-0.01986	0.00856	-0.00863	-0.01931
0.01776	0.00094	-0.01296	-0.02740	0.00360
-0.01527	-0.01185	0.00925	-0.00397	-0.04375
0.01407	-0.00164	-0.00137	-0.00226	-0.00760
0.00787	0.00072	-0.02146	0.00208	0.00953

-0.00124	-0.01874	0.02189	0.00995	-0.01045
-0.03765	-0.01083	0.00470	0.01657	0.00907
-0.00084	-0.00764	-0.00518	-0.00971	0.01478
-0.02153	-0.03598	0.00161	-0.02161	-0.00783
0.01042	-0.00065	0.02497	0.00814	-0.00082
-0.01043	0.01805	0.00487	-0.01213	0.03165
0.02005	0.01347	-0.01612	0.01119	-0.01979
0.02301	0.01049	-0.00173	0.00211	
-0.00807	-0.00977	-0.00050	-0.01247	average of results for pieces
-0.00561	-0.01148	0.00527	-0.03261	-0.00385
0.00576	-0.00529	0.01876	-0.01092	real average of all results
-0.00295	0.00153	0.00306	0.00128	-0.00377
-0.01664	-0.00621	0.01442	0.00273	standard deviation
0.01935	-0.00454	-0.01295	0.00194	0.0139
0.02404	0.01063	-0.02172	-0.01508	95% confidence range
-0.02207	-0.01582	-0.01546	-0.00450	0.00170
-0.01872	-0.00153	0.02227	0.00461	

$-D_{obs}$(predicted) in **9a** at 298 K

Each result below is the average of the NMR calculations for approximately 19,000 trajectory points, and is $+\Delta_{obs}$(predicted) (see Figure 2 in the main text) in ppm. The average of these numbers is -0.0738 ppm, but the accurate average after allowing for the exact count leading to each number is -0.0768 ppm. The standard deviation is 0.295 ppm. The calculated 95% confidence range is 0.0365 ppm.

0.12715	0.10824	0.23088	0.18089	0.04443	0.15509	0.05675
-0.52805	-0.12780	0.17821	-0.43175	-0.40214	0.13247	-0.40350
0.54053	0.59437	0.20138	-0.23486	-0.38587	-0.18743	-0.01490
-0.24913	-0.22338	-0.14705	-0.33480	0.50014	-0.07135	-0.09986
-0.60510	-0.42415	0.41986	-0.18184	-0.08063	0.24515	-0.29121
-0.26592	0.20984	-0.05798	-0.02071	0.20483	0.15961	-0.11305
-0.25474	-0.41769	0.45908	0.17619	-0.15324	0.00961	0.00114
-0.14311	0.02710	-0.38431	-0.00025	-0.01875	-0.13216	-0.10305
0.21145	-0.30358	-0.27582	0.05305	-0.27071	-0.16085	-0.24689
-0.68375	-0.29195	0.35904	0.12252	-0.63252	0.14448	-0.07163
0.02522	-0.53860	-0.38319	-0.00558	0.23914	0.57513	-0.49356
0.21986	-0.35974	-0.11827	-0.13090	-0.09346	0.17793	-0.69040
0.15710	0.05306	-0.44240	0.09088	-0.15756	-0.24308	
-0.27324	0.23491	-0.04199	-0.17754	0.41670	0.05816	
0.48056	0.03671	-0.17681	0.54083	0.19190	0.17955	
-0.25319	-0.44255	-0.14550	-0.19844	0.17484	0.32556	
-0.49743	-0.11031	-0.25858	-0.69491	-0.27133	-0.36436	
-0.82645	-0.28545	-0.37152	0.34086	0.47894	-0.12125	
0.26082	-0.36912	-0.42014	-0.24130	0.22333	-0.22574	
0.04062	0.40668	0.07090	0.06120	-0.25803	-0.13425	
0.12251	-0.05160	-0.17226	-0.46322	-0.10468	0.34203	
0.11688	0.06866	-0.31981	0.24315	0.02348	-0.23068	
0.32430	-0.17461	-0.23981	0.28598	0.08333	0.22959	
0.03312	0.37087	-0.03343	-0.16665	-0.12410	-0.07593	
0.04757	-0.56283	-0.29937	0.08471	0.30540	0.26091	
-0.05984	-0.47580	-0.48911	0.00464	-0.52937	0.26900	
0.10367	-0.38701	-0.04191	-0.30020	-0.10912	0.30463	

-0.47285	0.38951	-0.36281	-0.12488	-0.31470	-0.53216
-0.06831	0.05264	-0.47985	-0.13767	-0.16801	0.05352
0.15152	0.05489	0.31542	-0.33773	-0.16375	0.51799
-0.33362	-0.15172	-0.36093	-0.22803	0.14224	-0.04080
-0.64711	-0.16309	0.31547	0.23394	0.41215	-0.01524
0.09209	-0.29623	-0.09745	-0.16516	-0.05363	0.15163
0.07413	-0.18888	-0.12435	0.20990	-0.09557	0.44055
-0.00897	-0.36646	-0.42262	0.18250	-0.31905	-0.01708
0.19196	0.38117	-0.32643	-0.28774	-0.31570	-0.23433
-0.24868	0.30577	-0.44160	-0.44186	-0.19896	0.11697
-0.52756	-0.14486	0.41278	-0.17884	-0.50023	0.34345
0.26437	0.14133	-0.16193	0.56458	-0.30280	-0.69221
-0.86301	-0.60880	0.12292	-0.16509	0.10654	-0.33410

$-D_{obs}$(predicted) in **9b** at 298 K

Each result below is the average of the NMR calculations for a single complete trajectory of 1000 points, and is $+\Delta_{obs}$(predicted) (see Figure 2 in the main text) in ppm. The average of these numbers is -0.1389 ppm. The standard deviation is 1.300 ppm. The calculated 95% confidence range is 0.052 ppm.

-1.9337	0.8284	-2.9980	-0.4695	-1.8596	-1.6928	0.6921	-0.4555
1.6344	-2.4621	-1.7181	-0.0078	0.9720	-0.9090	0.0178	0.4734
1.2398	0.9235	-0.7669	-0.5469	-3.1142	0.2912	0.6369	-0.9131
-0.9173	1.5188	-1.0497	-0.8483	-1.3729	0.5029	2.9505	1.1998
-2.6289	2.4136	-1.6710	-1.7279	-1.7911	-1.1033	0.4487	-1.0550
1.9784	1.0265	-2.6635	1.0528	0.9579	0.4284	1.6787	0.2033
-1.0241	-1.0220	-0.2582	0.7905	-0.4654	-2.6099	0.6329	-0.0364
-1.2892	-0.9666	1.6166	-1.1660	-0.8384	-2.1763	-0.3813	-1.1414
-0.3280	-0.9207	0.4048	-0.6816	-1.1603	0.9017	-0.0520	1.5435
0.4553	-0.9051	1.5261	-4.1336	-2.2414	1.4500	1.5595	-1.9544
0.7652	-1.9299	-0.3660	-0.6920	2.9327	-0.6329	-1.7620	-1.5402
0.0679	0.2765	-1.5816	-2.0224	0.2240	-1.2593	-2.2265	1.3624
0.3973	0.2949	2.1905	0.9973	1.6623	2.9065	1.0053	-0.5428
0.8521	-0.2235	1.9649	3.4413	0.5282	-1.4004	0.8250	-0.2551
-1.6581	-1.2076	-0.0937	1.3890	0.7252	-0.9940	-1.5916	-0.3250
-0.1222	-0.6175	-1.0934	-0.9350	0.0042	-0.2754	-0.0423	0.1801
0.2455	-0.3962	0.0542	-1.3753	-1.7790	-0.3583	0.8558	-1.3780
1.8127	-1.2693	-0.9614	0.0776	2.6754	0.6438	-1.0363	-1.3944
-0.9568	0.8144	1.5173	-0.6520	-0.5358	-0.1166	-1.7122	0.9954
-0.8460	2.9694	0.9157	0.2301	-1.2936	0.8033	-0.4377	-3.6114
0.2463	-0.0085	-2.2329	-0.5239	-0.9827	1.6871	0.9704	1.2062
-0.2596	1.2513	0.7717	-0.8820	1.7329	-1.6239	-0.4975	-1.6416
1.0073	-1.4494	0.9916	1.0550	0.8096	-3.2146	-0.0438	1.6924
0.8459	-0.8558	-1.2013	-0.2994	0.4353	-0.6167	-2.0322	2.0301
-0.8232	-2.4591	1.6576	0.8498	0.1786	-3.1268	-0.7858	-1.7646
0.1503	0.5310	-0.9457	0.4790	-2.7430	-0.6957	-1.4375	-1.5487
-0.9995	-0.2746	0.3544	-1.8225	-1.5648	1.4926	0.3564	1.0076
-0.6929	3.8288	-2.6649	-0.4819	-0.1305	1.5784	-1.1243	0.7268
0.8852	-0.5399	-1.6279	-0.9110	-1.7657	1.2149	-1.2859	-0.0754
-0.4849	0.6903	0.1120	1.3902	0.2730	-0.9745	-0.5245	-0.3049
-0.4182	-0.9062	1.5775	-1.1793	1.7087	-0.3585	0.9426	-1.1189
-0.3084	0.5748	-0.2561	-0.3635	0.0390	-0.3882	-0.6277	-1.3296
-0.9114	-0.1690	0.3232	-0.1884	-0.3445	-0.7144	-0.4661	-1.4500
1.7203	-0.6738	-1.6097	-2.5149	1.0118	-1.9885	1.9404	1.4601
-0.7954	-0.1030	-0.6282	1.7989	-0.2377	-0.8556	-0.2525	-0.2803
1.4767	1.2633	1.1447	-0.4646	1.2852	-2.0325	-0.1491	0.5889
0.1725	1.1451	-0.8456	-1.8401	-1.7859	-0.2131	-1.1995	-1.3983

-1.2052	-1.4490	-1.9344	-0.4457	1.1226	-1.6721	0.9978	0.6622
0.1735	0.2401	-1.1248	0.0943	-1.0408	-1.7054	-1.9692	-0.7707
0.1536	0.3299	-1.4137	0.7653	1.6617	-0.5470	0.2514	-1.8961
1.2262	-0.3423	0.6423	-1.1766	-2.0834	-0.1240	-1.2901	-3.1234
0.3637	-2.3936	-1.7654	0.8422	-1.1495	0.6357	-2.2910	1.3783
-0.2063	1.4256	-1.7598	-0.8852	-0.9286	1.4830	2.3789	0.1795
-1.1067	-2.7318	0.4194	0.4922	-0.5358	-1.6186	-1.0479	-3.6706
-0.0494	-0.4613	-0.4119	1.9742	-1.4026	0.2810	-0.5397	-0.5315
-0.8279	1.3050	0.4150	0.2553	1.4564	-0.8621	1.8874	-0.3878
-0.7869	-0.4544	0.2630	1.6624	-0.5346	-0.9393	-1.0923	-1.2023
-1.2992	-0.2389	-0.9396	0.0114	-1.3067	0.8122	0.9281	1.1024
0.3453	-2.0105	-0.3467	-2.0309	-1.2255	2.0068	0.4943	-0.7596
0.5403	-0.8225	0.7123	1.0873	2.8219	-1.4559	-0.6705	-0.0777
1.6578	-3.0450	-0.3740	1.3740	1.0124	1.2529	0.0779	1.5153
-2.4533	-1.1822	-1.7164	0.9619	-0.2159	1.1100	1.9386	1.5076
0.4084	-1.1087	-0.8268	-2.3189	-0.6220	0.6253	-0.4532	-0.8735
1.5986	0.3727	0.9732	-0.0003	-0.8493	-0.9296	0.7505	0.3508
-0.3263	0.3895	0.5318	-2.0969	1.9291	2.1077	0.3410	1.6016
-0.9166	-1.2655	0.1514	0.9928	-1.1280	-2.0198	0.1183	-1.7918
-0.9207	1.1897	0.5134	-2.2410	-2.0213	-1.2098	-1.6722	-0.0276
-0.2961	0.0285	-0.1873	-1.4579	-0.2688	1.6870	0.3528	-0.8003
-1.1738	-1.2768	-0.0794	-1.0659	-1.8091	0.3063	0.8436	-0.4470
-0.9004	0.5074	1.7299	0.5031	2.0419	-0.6804	-0.2291	1.2783
1.3646	-1.8194	0.7275	1.1566	2.8064	0.5752	1.2042	-1.2901
-0.2485	1.4007	-0.1980	-2.1769	2.8258	0.6091	-0.5669	-1.2730
0.8693	-0.3360	-1.5012	-0.5657	-1.4423	1.9112	-0.3319	0.3280
0.4197	-0.3965	-1.9703	-2.2074	-0.6894	1.4663	-1.4981	-0.6765
-0.6096	-1.1379	-1.1544	-0.4619	-1.4502	-0.9090	0.0878	2.0237
-0.9258	-2.4093	-1.0327	1.1640	-0.7161	-2.1204	1.7693	-0.2795
1.5153	-1.3970	1.8693	3.3637	-1.2858	-0.0605	-1.4693	1.5587
-2.0886	-1.6981	0.0825	-0.2501	1.6894	-0.1226	1.7536	-2.5544
0.7537	-0.2457	0.6048	-0.1186	2.0235	-1.0155	-0.2628	2.1540
-0.0161	0.3313	0.0997	0.8754	-0.3495	1.0711	-1.3647	-0.4174
-1.0180	3.2324	-1.2683	0.8378	-1.0600	-1.8473	-2.3544	1.9653
1.3080	0.6219	0.6183	0.3956	-0.5019	-0.2758	-1.1167	1.6287
-2.2695	0.1949	-0.7359	0.3384	-0.0859	-0.1998	1.6443	-0.0030
0.6866	0.4020	-0.4272	-1.1089	1.5902	-1.0235	-0.2260	-0.9202
-0.9760	-0.4382	-1.5613	-2.2384	0.9891	-1.3187	1.5925	-1.3987
0.6988	-1.8726	0.9758	-0.6102	-0.6113	0.5880	-1.4348	-1.0236
-1.0102	0.2421	0.2148	-1.1532	-0.3260	-0.3678	-0.8598	-1.5651
0.9078	-1.0984	0.9506	-2.6361	-0.9413	0.6992	-2.7090	1.8159
-3.0023	1.1204	1.0432	-0.7124	0.1121	2.2311	-0.3702	-0.3347
0.9522	-0.8157	0.4982	-0.8473	-0.0805	-0.3479	-1.0085	-0.5534
-2.8498	-1.4022	2.8038	0.7917	-0.3816	0.1691	-0.3969	1.3557
1.8935	0.9098	-1.3662	-0.6351	-1.9843	0.3494	-1.6134	1.4939
1.0762	-0.5330	-0.4278	-1.8026	-0.3192	1.6555	-0.3892	0.8119
-0.0469	-0.9505	-1.0390	1.3042	1.1266	0.7026	0.2203	-3.2006
-1.2063	-1.8788	-1.2744	1.4416	0.4570	2.1472	1.3064	-1.0148
-2.1777	-0.2664	1.6476	-0.8087	0.8993	-1.5300	-0.5446	-3.1557
3.6691	-0.3100	0.1970	0.3335	-0.8381	-1.8480	-1.4511	0.8405
1.1475	1.1977	0.8215	-1.6181	1.5168	-1.1278	2.4719	-0.0242
-0.3754	1.5135	-2.8176	-0.1810	1.5844	1.9423	0.6205	-2.3507
-0.5115	0.4826	0.5383	-0.1683	-2.3989	-2.1648	1.0546	-1.1336
-1.2909	1.8165	1.9026	-0.9376	0.4912	-0.6613	1.0634	-3.0813
-1.1113	-2.1607	-1.2799	0.6869	-0.7479	1.7257	-1.1629	1.0941
-0.2523	1.6582	0.8303	-0.3198	0.5515	-1.2151	-1.2087	-0.1098
0.2714	-0.8558	3.1044	1.7778	0.1759	-1.1889	0.6524	0.0575
1.7235	1.1078	-2.7714	0.7692	-0.9996	0.6415	-1.0224	-0.4528
0.4832	-0.1775	-1.4502	-1.6418	-0.0190	-1.6945	0.6120	2.1422

-0.2856	-1.9918	-0.5401	0.5536	-1.0788	-0.2664	0.8353	-0.3198
1.7242	-0.2913	0.2704	1.8853	0.3255	-0.7813	-0.2567	0.6408
-1.1531	2.0889	-0.0472	-1.4161	0.1583	-0.7561	-0.3889	-2.0085
-0.1860	3.7759	-0.6892	0.5167	-0.0401	-0.5623	0.9526	-0.0396
-0.0796	-0.0565	-1.0277	2.3281	-3.2981	0.3868	-1.0626	0.6726
-1.3638	-0.3216	0.9663	1.1653	-1.7350	0.4863	-2.1484	1.3242
-0.4182	-0.5544	-1.3078	-0.0942	-1.2761	-3.2976	0.0953	-0.2990
-2.0507	2.5008	-2.1747	0.2021	1.1465	3.7010	-1.2177	-0.1337
-1.2125	-0.8798	0.0338	-0.3168	0.6348	-0.1073	1.4636	-2.1791
-1.6424	-0.6757	-1.0204	-0.0250	-1.0344	0.7983	-0.6675	1.7302
-0.2698	-0.2999	1.1170	-0.9284	4.3454	-1.1822	-0.5482	-1.2076
-2.4898	-0.6642	0.3274	0.6135	0.0570	-0.0342	0.0707	0.3105
-1.6206	0.0393	0.5488	-0.8398	-0.5313	-1.1148	0.6735	1.5546
1.3563	0.3577	0.2371	-0.7105	2.3089	0.2336	1.0883	-0.4747
0.7123	-1.3297	-0.2810	1.0624	0.2779	1.4417	0.6108	0.4151
2.2680	0.2669	-0.4910	-1.0864	-0.9552	1.3564	-2.1094	-0.3558
3.3142	-2.2427	0.0364	-1.0612	1.9317	1.3210	1.5850	1.3648
-0.3549	0.5027	0.6048	0.8952	-2.0642	0.2770	-0.5997	-0.0402
1.5796	0.4247	0.9849	0.3492	2.9719	-0.3369	0.8237	0.9278
-0.2996	-0.2542	-3.9813	-0.0480	0.0173	-0.3486	-0.1047	0.4633
0.0264	-1.2957	0.6286	0.5295	-0.1999	-1.7905	-1.4264	0.6812
-0.4022	-0.6952	-0.1638	0.2782	-2.4342	1.0778	-1.2315	-1.4430
0.2893	-0.7977	0.4189	0.0740	0.7171	-1.7340	-1.9505	-1.2560
-0.1623	-0.4382	-1.1087	1.0047	1.0516	0.8664	-0.6937	1.3914
-0.3570	-1.6087	0.7507	1.9331	-0.8875	-1.4665	-1.0285	-1.0243
0.5658	1.0418	-2.0651	-0.8808	-0.4686	-0.9483	3.4373	0.7032
1.0493	1.2109	1.3583	2.3503	-1.1574	-1.2014	0.1907	-0.3583
0.6432	-1.0112	-0.5703	1.3448	-0.1797	0.8503	1.1391	0.3674
-1.2945	0.9443	-0.3302	-0.1022	-0.1282	-2.6019	-2.1204	0.2326
-2.6505	0.2500	-0.6009	-0.8210	1.4745	1.0213	1.6534	1.2774
0.0668	0.0997	0.6088	-0.6213	-0.2862	1.9025	2.1831	1.9393
0.6589	0.4600	-1.8843	-1.3413	0.2559	0.5318	-0.1069	1.1945
-0.5810	-0.3229	-1.0956	-0.4658	-0.1998	1.5272	-1.6706	-0.6775
-1.2885	-1.8577	-2.4178	0.4306	0.6242	-0.8142	1.7418	0.3547
-0.7071	-0.0224	-1.1505	0.0373	-1.2161	-1.9603	-0.3844	-0.4654
-0.2430	1.5828	-0.1912	1.9306	0.3432	-0.2965	-1.6852	1.1918
-0.4224	0.4058	-1.1000	0.3651	-0.3513	1.2868	0.0732	-0.8080
1.2318	0.9641	1.0437	0.3675	-2.7058	1.4302	1.4487	0.3666
-2.2719	0.8079	-3.3166	2.4705	0.0384	-0.2655	-1.2495	0.0842
-0.0405	-1.6155	1.5272	-0.2744	-1.1337	3.0776	1.5100	1.0657
-1.3881	-2.4912	-0.1051	-0.4682	1.6224	-0.3712	1.7098	3.0832
-1.3064	0.0243	-0.6213	-0.4085	-0.4690	2.4602	-0.6336	-0.0109
-1.5069	-0.9331	1.6780	1.3365	2.3538	0.5440	-0.5854	-3.0559
-1.4246	2.6085	-1.9212	-1.6771	-1.2503	1.9727	0.3466	0.8636
1.4358	-0.7186	-1.8108	-2.2310	1.8054	-0.9855	1.1693	-0.7552
0.2034	0.0545	0.7038	1.0893	-1.0221	1.4733	0.1713	-0.7286
2.2765	-0.4472	-0.2476	0.9586	0.1579	1.6279	-1.3768	0.4777
0.1472	0.0056	-2.4218	-0.0025	-0.0685	-0.5465	-0.6342	0.4568
-1.3277	-1.7608	-1.5597	-1.5120	1.4674	-0.9354	-1.2664	1.5296
0.8154	-2.8031	-1.2796	-0.4259	2.5210	1.3223	-2.5144	0.8668
-1.7889	0.2896	1.2245	-1.4510	0.8435	-0.6866	-1.0010	-2.7640
1.8750	1.2500	-0.7923	0.4478	2.2687	0.2904	-0.9419	-1.1268
-0.3206	-0.4314	-2.0064	-2.9561	-0.4191	1.2771	-0.6304	-1.5813
1.5948	-0.3940	0.0615	0.5610	1.2426	-0.6479	-1.5997	-2.1314
-0.8591	1.1709	-0.6607	0.3279	-0.3618	0.7168	-1.9982	-1.5136
-0.8615	-2.7156	-0.2058	-1.4056	3.2186	-0.1210	0.5492	-0.7253
-1.3797	2.2624	1.0771	-1.5074	0.1783	0.9662	1.0139	-0.5009
1.5705	1.7359	-0.2375	2.3087	1.7579	-0.7446	-0.1193	-2.1837
-1.8943	-1.0670	-0.6940	2.1179	0.1656	-1.5702	2.1753	1.6674

-1.1822	1.6686	2.2418	-0.7090	-3.5762	-0.4818	-4.6269	0.1320
1.0071	-0.0188	1.0889	-0.9340	0.2253	-0.6557	1.6100	-0.1981
-0.2478	0.6018	0.8780	-0.9744	-0.4523	0.0628	-1.3093	-1.3521
-0.6806	0.3420	1.1906	1.6656	-1.7261	-0.3203	-3.3361	0.6609
-1.0019	-1.0522	0.3559	-1.7989	-1.0012	1.6527	-0.4176	1.3562
0.3358	1.0954	-1.4389	-1.2018	0.0251	-1.0289	-0.5077	-0.3345
-0.5151	-0.6219	0.0949	-0.4387	-1.7361	0.3930	0.1320	-0.6614
0.6124	-0.4865	0.2669	0.6380	-0.1814	0.4244	-2.4157	0.7744
0.6654	0.3007	1.2369	0.3524	0.0300	1.4188	-1.6753	1.6231
-0.3700	1.1958	-1.8297	-0.3694	0.6524	-0.5266	0.6038	-0.1214
-1.9195	0.6159	1.2928	0.6565	-0.6186	-1.2499	-0.4845	-1.2370
-1.0170	1.1313	0.9022	1.3282	-1.1621	-0.6745	-0.1758	-1.9861
-0.5015	2.8939	-0.2145	0.2419	1.3951	1.5554	0.4411	0.5426
-1.6604	-1.8876	0.0413	-2.9932	1.1104	0.0009	0.0386	-0.8912
0.5215	0.2421	-1.2680	0.1951	0.6251	1.2496	-0.7725	2.1440
-1.0068	1.0122	1.1372	0.0936	0.6380	-2.0381	0.4135	-0.4559
0.5285	-1.4690	-1.7140	-0.1610	-0.3112	-1.1031	0.3072	-0.5444
-0.1980	-0.5038	-3.5973	0.3763	-0.0199	1.6980	-0.1073	-3.6089
0.8795	-1.6755	0.5936	1.2606	0.0325	0.1163	-1.7307	0.4127
-0.7970	-1.6326	-0.8765	0.5289	-1.8494	-0.0324	-0.3265	0.2571
-1.5971	-0.3834	-1.6367	-0.8248	1.7818	-1.6576	-1.2518	1.6916
-1.1820	1.4033	0.0863	0.2388	-1.3499	-0.4998	-1.3492	1.8566
0.2223	0.6759	0.5621	0.4481	-0.3050	-0.4560	-0.8272	0.2141
-2.1112	-1.1097	-1.0372	-0.8683	-0.6831	-0.7383	-1.4119	0.1629
-1.1795	-1.3009	-1.1658	-0.3820	0.2868	-0.9749	-0.1533	-0.5111
0.0031	-0.9937	0.7694	-1.5271	2.8676	-0.3257	0.6657	-0.5983
-2.2756	-1.3228	-1.5249	-1.4569	1.0378	-0.7623	-1.3841	0.0668
1.7046	-0.3095	1.3060	-0.1939	-2.1496	1.1565	-0.2942	-0.5923
0.4122	1.8887	0.4123	-1.1341	-0.5272	1.8987	-1.8727	-0.7901
0.9417	-1.2300	-0.5440	-0.4085	-0.1618	-0.5879	-1.1772	-0.5111
2.3470	0.8395	0.3903	0.9707	-1.1908	0.0222	1.4410	0.3205
-0.0608	-0.4510	0.1316	2.1491	-0.3718	2.6386	-1.0287	1.5103
-0.7652	0.5759	0.5255	1.5904	0.1623	0.3898	-0.2590	-2.0095
-0.8769	-0.1923	0.0906	2.2342	0.3482	-2.2772	-0.4524	-0.1149
-3.2297	-0.9237	-0.8807	2.7342	1.4314	0.7080	0.1729	-1.1126
-0.0741	2.6957	-1.6878	-0.8705	1.9409	1.4692	-1.0558	0.9923
-0.7729	-0.4085	-1.1377	0.4138	-0.5491	-2.0435	-0.6243	-0.3775
-1.5583	-0.9040	-1.0142	1.5402	-1.3566	0.7885	-0.9023	-3.2911
1.7941	-1.5666	0.1164	1.0877	-1.2526	1.7712	-1.3533	-1.5254
1.1023	-0.6706	-0.2105	0.5042	4.2121	1.3736	-1.0131	2.3442
0.3207	0.5683	-1.0867	-1.1806	0.1858	-0.3308	0.0972	-0.2681
-0.3696	1.3974	-0.8372	-0.9352	2.1442	-1.2118	-2.1610	-0.6399
-0.7061	1.5672	0.8444	0.7401	0.4016	-2.0381	-0.2933	-1.1292
-0.1171	0.3414	-0.6861	-1.7569	0.2513	-0.8001	0.1729	-2.0798
1.5533	-0.6003	1.1850	-0.5898	0.9953	-0.6565	-1.7891	-0.0215
-0.8033	2.3878	0.4032	-0.8859	0.3011	-1.6005	-0.0070	0.0026
-0.7137	0.6865	-3.5340	2.6506	-1.6866	-1.0406	3.2914	1.8263
0.9914	0.0023	1.2043	2.5093	-0.7652	0.8561	-0.3475	-2.8296
1.7762	0.0343	-2.0029	0.6176	-0.3301	-0.7339	0.4595	-0.8740
-0.5652	0.6286	-1.5586	0.7168	2.9544	-0.6019	-1.7481	1.1475
1.6688	-0.0072	1.1836	-0.5989	-0.7388	0.8958	-1.9842	-1.5949
-0.1182	-1.3178	0.4483	-1.6928	0.1230	-1.7910	-0.4192	0.9248
-0.0965	-0.3566	1.9447	2.7342	0.5159	0.0098	-0.8110	0.5493
-0.5516	2.3553	1.1594	-0.2630	-1.0622	0.2821	-0.9147	-0.1379
1.1718	1.4729	0.8485	2.7949	0.0889	-0.4629	1.3581	-0.5118
1.5057	1.1422	0.6544	-1.7287	-2.4626	-0.5592	-1.1816	-0.4086
-0.2760	-1.1952	-0.6054	-0.0605	0.6525	0.6071	-0.9623	-1.4251
-1.8975	-0.4764	0.1675	-1.9810	-1.8237	0.0881	0.7096	0.7079
-0.7899	1.5167	-0.2326	-1.5731	3.6192	-0.7152	0.5233	-2.5109

-0.8350	1.0700	-0.7869	-1.2917	-0.1733	-0.6407	-1.7261	-1.3946
3.0058	-1.1345	1.3124	0.5205	-1.9751	-0.4107	-0.3219	-0.0266
0.0304	0.2290	-0.8877	-0.1975	1.7612	0.7799	-0.2931	2.3286
0.5737	1.0904	2.3630	1.9708	0.0993	0.9336	0.5285	2.2611
-0.0714	-1.3635	-0.2012	-2.6255	-0.8238	-0.8565	-0.9640	-1.6405
0.7555	0.1264	-1.5171	0.2446	0.9016	1.1354	0.7736	-1.0161
-0.2713	-1.7257	-0.6641	-1.7812	0.1265	0.2909	0.2142	1.3903
0.1379	-0.2108	0.7351	1.5287	-0.7945	1.4101	-1.3887	0.2323
-1.5221	-0.1904	-1.5172	-1.2430	-0.7847	-0.8013	0.5767	1.4292
0.9373	-0.8357	-1.1351	0.6193	-2.6017	-2.8013	-1.8559	-1.7315
-2.2563	0.9675	1.3863	0.3487	-0.5925	-0.9050	-2.0114	1.0081
0.2622	-0.9330	1.3007	0.0381	0.0746	-1.6735	-0.2482	1.6970
0.2539	-0.7856	-2.0548	-2.7813	-1.8628	-1.2155	2.5213	0.8306
-0.2605	1.4541	-1.7655	-1.5536	0.3704	0.4661	-0.8856	1.4908
0.1356	0.5736	-0.5792	-2.0047	0.0664	0.3383	-0.0812	-0.7609
0.2208	-0.1627	-1.0289	0.7662	0.9013	-0.9449	1.6671	1.1227
1.4469	-0.0622	-1.7196	-1.7943	-1.4582	1.7485	-0.1475	-1.6576
-0.4713	0.6849	-0.2273	3.3378	-1.4417	0.0299	2.5943	1.1525
-0.3312	-1.0774	-0.0770	-1.5823	-2.0097	-1.7562	-0.1833	1.9720
0.1439	0.5104	-1.7932	-2.4562	-0.3953	1.1999	0.4752	0.7309
0.9103	1.5361	1.0242	1.3862	-0.5098	0.0706	-0.1193	-0.1298
1.1597	1.0397	-1.7137	1.6958	0.1150	-0.5711	-0.7758	-1.7180
-0.9508	-1.2339	0.5502	-0.6416	1.8138	-0.3904	0.2637	-0.0037
0.1079	-0.2249	-0.1588	1.2172	-1.4247	-0.2261	-0.6969	1.7975
0.9335	0.3753	-0.1578	-1.4992	-0.3689	-0.7896	-1.1300	0.5247
2.4697	-1.2899	-1.2050	-1.4977	-1.1974	-0.0097	-0.9254	-3.0770
-0.0418	0.4123	-0.4858	-0.1113	-1.6995	-0.1751	1.0968	-3.4156
0.3047	-0.2546	-0.2878	-0.1295	-0.3568	-0.9684	0.4188	2.1400
0.1570	0.3394	0.1110	0.7844	-0.1791	-1.6360	0.1775	1.0379
0.0631	-0.7493	-0.8183	-0.9641	1.1518	1.2567	1.9560	-1.0654
1.3707	2.6210	-0.4768	-0.8812	-0.4897	-1.0121	-1.7906	0.3525
-1.1481	-2.9019	-0.2715	-0.7957	-0.8815	-0.5502	0.1025	2.4271
-2.4151	-0.2295	-0.6511	0.3492	0.5904	1.1530	-0.6635	0.1766
1.2368	2.2797	-0.1168	-0.0295	0.8139	-3.1461	-0.0356	-1.5846
0.5616	-0.2951	-2.6255	-0.2784	-0.9806	-0.7052	-1.2265	0.1929
5.1445	1.2642	-0.8195	-1.4272	0.0392	-1.0226	-1.2991	0.0402
1.0539	-1.4813	-0.1636	-0.6858	-1.3379	1.6835	0.1448	0.0374
-0.3636	0.5143	-3.0295	0.2453	1.9463	-1.3687	-1.2743	1.2834
-0.8881	-0.1028	-1.7255	2.6797	0.4108	0.7696	-0.1149	0.2886
-0.3476	-1.1599	-1.0403	0.5620	1.0875	0.9093	0.5269	2.3704
2.3094	-0.6564	-1.6135	1.2562	0.7136	-2.4480	1.8248	-2.1811
-2.9212	-0.3237	2.7482	-0.0907	0.5787	-0.3213	1.7677	-3.3768
1.8862	1.3062	-0.9091	-1.4198	-1.0731	-2.5610	0.1606	-0.1066
-2.2248	-1.8760	0.5722	-0.4515	0.9922	1.2654	0.4893	-1.3424
-0.1428	-1.4154	-1.6182	0.4638	-1.0961	-0.5754	-0.1098	0.1573
0.5055	-1.8440	-2.1191	-0.0218	1.3464	0.7104	-1.4244	-0.8167
-0.7234	-0.6175	0.0673	0.0204	0.1837	-1.0151	-0.7765	1.0370
-0.4641	-1.0992	0.3990	-0.1801	0.6070	-1.3802	0.6171	-0.6417
-0.7652	-0.9496	0.4535	1.8315	1.6277	-2.0549	-0.5146	-1.4073
0.5327	-0.3432	-2.8275	0.9637	-0.8275	0.0707	-0.2784	1.5374
-0.4470	1.8663	0.6496	-1.8698	-0.6237	1.3955	1.3205	0.0569
1.2077	-0.1134	2.6474	-0.2650	0.1511	-0.5094	1.6122	-1.5756
1.2051	-0.3142	-0.7063	-2.9485	-1.0823	-0.4226	0.3409	-3.1684
0.3011	-2.0931	0.7014	-0.7726	-1.4417	1.2974	-1.6152	2.2132
0.7281	-1.4819	1.2930	-1.5061	1.6636	0.3021	-0.4129	1.1197
-2.2798	-0.2511	2.3425	-1.1651	-0.4298	-2.0651	1.5450	-0.2925
-0.8134	1.2360	-0.0407	1.2370	-0.6380	-2.5298	0.0520	0.6451
-1.3539	-0.4283	0.0183	-0.7582	-3.8090	0.5153	-1.6625	0.8396
0.4855	1.0277	-0.7616	-0.2748	-1.7354	1.2433	-0.2958	-4.0356

1.9918	-0.7944	-1.5868	2.8122	-0.7461	0.4305	-0.8456	-0.3667
-1.2677	0.0183	-0.0269	0.5168	2.3556	-0.8483	-0.0616	0.1384
0.9967	0.5266	0.3845	0.7721	-1.5685	-2.5906	2.0603	-1.2495
-0.4459	-0.7767	2.1254	0.2197	0.2742	-1.6364	0.7397	2.7902
-0.3583	-0.7698	1.1918	-1.4707	-1.4340	-0.1340	1.1162	0.5436
-1.4697	-1.5345	0.3807	-0.4191	-1.4956	-1.4107	-0.5333	-0.0920
1.4825	-0.8814	-0.8773	-3.7642	1.6749	0.9562	-2.0904	-0.6911
-0.5516	-0.0262	1.1382	0.9321	-0.6577	-0.7928	-0.3120	0.9012
-1.4863	-0.4924	-0.8856	-0.5711	-0.4621	1.4284	-0.4850	-1.7378
-0.1153	-1.8621	-0.3326	-0.3374	1.0799	-0.5375	1.3001	0.3494
-0.8614	-0.4028	-0.1371	-1.0487	0.3135	1.0984	2.1955	-0.4245
1.6652	0.1519	2.6328	2.7992	-0.1862	-0.2123	0.6029	1.5186
-0.7946	2.0780	-1.4317	-0.4495	0.8392	-0.2335	-1.7782	-1.0536
-2.5408	-1.6404	1.9557	0.1945	-0.0296	-1.5823	0.4334	-3.9459
0.1066	-2.2336	1.4572	-1.0649	1.8912	0.3597	2.3805	0.5682
0.9103	-0.1984	-0.3598	0.0571	-1.0186	-0.5187	-1.0431	0.2132
-1.0479	1.0085	0.7310	-0.6372	0.8266	1.5610	1.8218	-0.4093
-0.4472	3.2907	-2.9674	-0.0975	-0.0631	-0.4245	-0.1957	-0.7459
2.5979	-1.3939	0.2777	1.0365	0.9010	1.6726	-0.9583	0.0127
0.8818	-1.7836	0.0558	-0.3999	-1.6863	-0.1892	1.2923	-0.6111
-0.9612	-0.5321	-1.0056	2.1648	-1.5156	1.6960	-0.6866	0.5404
-1.7049	-2.0870	0.2671	1.6224	0.6113	-0.3125	-1.2478	-0.1202
-0.9075	0.5149	2.2560	0.0863	-2.6796	-0.9000	1.0439	1.7506
0.4084	-0.7197	-1.0449	0.6505	-0.3052	-2.0969	-0.1676	2.1860
-2.6339	-1.1645	-0.7888	0.3730	1.4501	-1.0338	0.2204	-1.2626
0.5310	1.7180	-0.0524	-1.3619	-0.7803	-2.7062	-3.2773	
1.8106	-1.6939	-0.9753	-0.7358	-0.6124	-1.8108	0.8589	

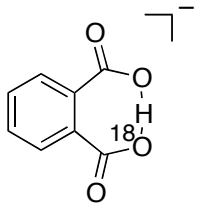
Convergence in Second-Order Perturbation Theory Calculations of Phthalates

As mentioned in the main text, we encountered convergence difficulties in approaching the geometry of hydrogen phthalate anions using the second-order perturbation theory approach implemented in Gaussian 09. The lore in this area is that relatively low-level calculations perform well in calculating the anharmonic force-field terms, and it might be expected from this that the geometry obtained from second-order perturbation theory would not vary with modest changes in the calculation. This is not what we observed. I have listed below the geometry obtained from various similar calculations, and the variation observed was far too large for the purpose of estimating the degree of asymmetry that results from an ^{18}O isotopic substitution in hydrogen phthalates or estimating the effect of isotopic substitution on the NMR spectrum. Second-order perturbation theory appeared to perform better for the large changes in geometry with **3**, and MPW1PW91, B3LYP, and M06 calculations gave roughly similar results.

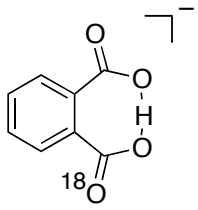
In each case the starting structure for the second-order perturbation theory was optimized with opt=tight or opt=verytight convergence criteria, and with all of the same options (such as integral(grid=ultrafine) or acc2e=11 or scf=(conver=10)) as used in the subsequent anharmonic frequency calculation. Less tightly optimized structures led to significant variations in the geometries, not recorded here.

The listings below show the options used in the Gaussian 09 calculation.

In each case, "18 in O2" refers to a calculation on:



and "18 in O6" refers to a calculation on:



mPW1PW91/6-311++G** freq=(anharmonic,readisotopes)
18 in O2

1	0.000000	0.000838	2.209362
8	0.000000	1.205003	2.187763
8	0.000000	-1.205701	2.187202
6	0.000000	1.686946	1.002197
6	0.000000	-1.687696	1.001968
8	0.000000	2.884603	0.760057
8	0.000000	-2.885322	0.759338
6	0.000000	0.709855	-0.198478
6	0.000000	-0.710568	-0.198712
6	0.000000	1.372515	-1.438012
6	0.000000	-1.372917	-1.438568
6	0.000000	0.696270	-2.648653
6	0.000000	-0.696568	-2.649220
1	0.000000	2.458403	-1.401144
1	0.000000	-2.458828	-1.402316
1	0.000000	1.252952	-3.583810
1	0.000000	-1.252794	-3.584652

mPW1PW91/6-311++G** freq=(anharmonic,readisotopes)
18 in O6

1	0.000000	0.001090	2.209314
8	0.000000	1.205175	2.187481
8	0.000000	-1.205427	2.187386
6	0.000000	1.687133	1.001960
6	0.000000	-1.687598	1.002346
8	0.000000	2.884917	0.759816
8	0.000000	-2.885146	0.759946
6	0.000000	0.709922	-0.198513
6	0.000000	-0.710495	-0.198436
6	0.000000	1.372719	-1.438050
6	0.000000	-1.372711	-1.438296
6	0.000000	0.696549	-2.648786
6	0.000000	-0.696271	-2.648871
1	0.000000	2.458608	-1.401262
1	0.000000	-2.458610	-1.401914
1	0.000000	1.253116	-3.584047
1	0.000000	-1.252598	-3.584245

mPW1PW91/6-31+G** freq=(restart,anharmonic,readisotopes) integral(acc2e=11,grid=99770)
scf=(conver=10) scrf=(pcm,solvent=dichloromethane)
  nosymm geom=check
18 in O2

1	-0.018779	0.000000	0.000629
8	0.006671	0.000000	1.205354
8	0.007125	0.000000	-1.206248
6	1.205501	0.000000	1.678886
6	1.205809	0.000000	-1.679666
8	1.428609	0.000000	2.890181
8	1.429371	0.000000	-2.890846
6	2.400672	0.000000	0.712614
6	2.400920	0.000000	-0.713205
6	3.642434	0.000000	1.375409
6	3.643020	0.000000	-1.375828
6	4.857028	0.000000	0.696963
6	4.857638	0.000000	-0.697393
1	3.620743	0.000000	2.461604
1	3.621486	0.000000	-2.462035
1	5.791789	0.000000	1.253909
1	5.792722	0.000000	-1.253812

```
mPW1PW91/6-31+G** freq=(restart,anharmonic,readisotopes) integral(acc2e=11,grid=99770)
scf=(conver=10) scrf=(pcm,solvent=dichloromethane)
  nosymm geom=check
18 in O6
1                   -0.018724    0.000000    0.001527
8                    0.006957    0.000000    1.205482
8                    0.006947    0.000000   -1.206027
6                    1.205855    0.000000    1.678979
6                    1.205282    0.000000   -1.679566
8                    1.428867    0.000000    2.890277
8                    1.428600    0.000000   -2.890682
6                    2.400702    0.000000    0.712621
6                    2.400625    0.000000   -0.713180
6                    3.642459    0.000000    1.375695
6                    3.642682    0.000000   -1.375544
6                    4.857187    0.000000    0.697450
6                    4.857173    0.000000   -0.696880
1                    3.620456    0.000000    2.461878
1                    3.621346    0.000000   -2.461743
1                    5.792212    0.000000    1.253971
1                    5.792045    0.000000   -1.253663

mPW1PW91/6-31+G** freq=(anharmonic,readisotopes)
18 in O2
1                    0.000000    0.001192    2.217189
8                    0.000000    1.203985    2.193688
8                    0.000000   -1.204359    2.193115
6                    0.000000    1.686742    1.002399
6                    0.000000   -1.687569    1.002294
8                    0.000000    2.892410    0.760360
8                    0.000000   -2.893223    0.759617
6                    0.000000    0.711248   -0.198566
6                    0.000000   -0.711946   -0.198674
6                    0.000000    1.375737   -1.440531
6                    0.000000   -1.376023   -1.440921
6                    0.000000    0.697990   -2.654312
6                    0.000000   -0.698208   -2.654664
1                    0.000000    2.462963   -1.402842
1                    0.000000   -2.463266   -1.403807
1                    0.000000    1.255365   -3.590649
1                    0.000000   -1.255364   -3.591136

mPW1PW91/6-31+G** freq=(anharmonic,readisotopes)
18 in O6
1                    0.000000   -0.000187    2.217185
8                    0.000000    1.203867    2.193398
8                    0.000000   -1.204355    2.193364
6                    0.000000    1.687044    1.002415
6                    0.000000   -1.687300    1.002417
8                    0.000000    2.892815    0.760054
8                    0.000000   -2.892890    0.760253
6                    0.000000    0.711426   -0.198479
6                    0.000000   -0.711774   -0.198504
6                    0.000000    1.375826   -1.440491
6                    0.000000   -1.375919   -1.440725
6                    0.000000    0.698114   -2.654301
6                    0.000000   -0.698051   -2.654414
1                    0.000000    2.463050   -1.402877
1                    0.000000   -2.463153   -1.403358
1                    0.000000    1.255476   -3.590678
1                    0.000000   -1.255228   -3.590866



mPW1PW91/6-31+G** freq=(anharmonic,readisotopes) scrf=(pcm,solvent=dichloromethane) geom=check
18 in O2
1                   -2.195351   -0.000184   -0.343441
8                   -2.173429   -1.207464   -0.269713
8                   -2.173315    1.208642   -0.268861
6                   -0.996920   -1.665911    0.058307
6                   -0.996532    1.666918    0.056361
8                   -0.795325   -2.853486    0.375200
8                   -0.794093    2.854630    0.370670
```

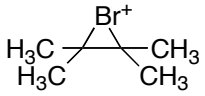
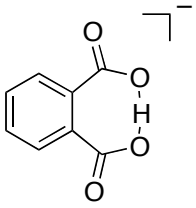
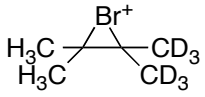
```
6                    0.203346   -0.711621    0.007747
6                    0.203487    0.712095    0.006161
6                    1.442730   -1.377520   -0.038874
6                    1.443156    1.377634   -0.042314
6                    2.655558   -0.697618   -0.106613
6                    2.655929    0.697554   -0.108546
1                    1.423192   -2.464136   -0.022363
1                    1.423960    2.464290   -0.028802
1                    3.590067   -1.253694   -0.154683
1                    3.590598    1.253300   -0.158565

mPW1PW91/6-31+G** freq=(anharmonic,readisotopes) scrf=(pcm,solvent=dichloromethane) geom=check
18 in O6
1                   -2.194591   -0.000731   -0.353803
8                   -2.172610   -1.208155   -0.282306
8                   -2.172732    1.207865   -0.276099
6                   -0.996303   -1.666580    0.047406
6                   -0.996775    1.666064    0.055644
8                   -0.794674   -2.854338    0.362788
8                   -0.795429    2.853217    0.376071
6                    0.203720   -0.711911    0.001165
6                    0.203618    0.711766    0.004991
6                    1.443222   -1.377649   -0.045959
6                    1.443142    1.377514   -0.038166
6                    2.656114   -0.697531   -0.108771
6                    2.656031    0.697634   -0.104759
1                    1.423826   -2.464305   -0.033630
1                    1.423904    2.464118   -0.020106
1                    3.590873   -1.253208   -0.157512
1                    3.590603    1.253694   -0.150456

mPW1PW91/6-31G* freq=(anharmonic,readisotopes) scrf=(pcm,solvent=water) geom=check
18 in O2
1                   -2.208993   -0.002499    0.000000
8                   -2.193024   -1.217801    0.000000
8                   -2.191978    1.218792    0.000000
6                   -0.994342   -1.682756    0.000000
6                   -0.994022    1.683824    0.000000
8                   -0.768101   -2.886728    0.000000
8                   -0.766933    2.887988    0.000000
6                    0.197898   -0.711909    0.000000
6                    0.198550    0.712864    0.000000
6                    1.439107   -1.373603    0.000000
6                    1.440451    1.374112    0.000000
6                    2.653273   -0.696590    0.000000
6                    2.654286    0.697280    0.000000
1                    1.413736   -2.460048    0.000000
1                    1.415327    2.460560    0.000000
1                    3.588566   -1.253573    0.000000
1                    3.589985    1.253574    0.000000


mPW1PW91/6-31G* freq=(anharmonic,readisotopes)
18 in O2
1                    0.000000    0.001980    2.211908
8                    0.000000    1.212945    2.195341
8                    0.000000   -1.213707    2.194657
6                    0.000000    1.690882    1.001291
6                    0.000000   -1.691942    1.001327
8                    0.000000    2.893663    0.752746
8                    0.000000   -2.894712    0.751803
6                    0.000000    0.710578   -0.196677
6                    0.000000   -0.711308   -0.196834
6                    0.000000    1.374110   -1.437828
6                    0.000000   -1.374448   -1.438256
6                    0.000000    0.697500   -2.650708
6                    0.000000   -0.697746   -2.651104
1                    0.000000    2.461409   -1.395327
1                    0.000000   -2.461764   -1.395961
1                    0.000000    1.254576   -3.588379
1                    0.000000   -1.254516   -3.588964

mPW1PW91/6-31G* freq=(anharmonic,readisotopes)
```

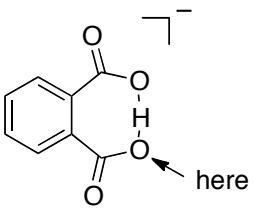
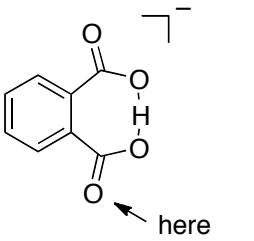
```
18 in O6
1                  0.000000    0.000355    2.212045
8                  0.000000    1.212937    2.195171
8                  0.000000   -1.213542    2.195058
6                  0.000000    1.691145    1.001444
6                  0.000000   -1.691605    1.001515
8                  0.000000    2.894018    0.752539
8                  0.000000   -2.894309    0.752408
6                  0.000000    0.710745   -0.196599
6                  0.000000   -0.711116   -0.196576
6                  0.000000    1.374298   -1.437763
6                  0.000000   -1.374324   -1.437975
6                  0.000000    0.697699   -2.650711
6                  0.000000   -0.697564   -2.650798
1                  0.000000    2.461596   -1.395193
1                  0.000000   -2.461628   -1.395607
1                  0.000000    1.254689   -3.588467
1                  0.000000   -1.254411   -3.588607
```

Key to Structure Titles

The structure titles in the next section have retained the original file names for the computational output files. This allows consistency in naming structures, energies in tables and our records. We will give here some general explanation of the titles along with some structural drawings.

The titles consist of a series of codes. The table below lists the codes used and their meaning. When this is not sufficiently descriptive, additional information is provided below the title.

Code in title	meaning
Bromonium	Refers to a calculation on  or an isotopologue
Phthalate	Refers to a calculation on  or an isotopologue
isopropylBr	Refers to a calculation on 2-bromopropane or an isotopologue
Formicacid	Refers to a calculation on formic acid or an isotopologues. All formic acid calculations were carried out with a 6-31G* basis set.
MPW	Calculated using the MPW1PW91 method
B3	Calculated using the B3LYP method
M06	Calculated using the M06 method
BP86	Calculated using the BP86 method
SB	Calculated using a 6-31G* basis set
PS	Calculated using a 6-31+G** basis set
BB	Calculated using a 6-311+G** basis set
C2V	A structure with exact C_{2v} symmetry
6D	A calculation carried out on 
anharm	The structure obtained from second-order perturbation theory based on the cubic and semi-diagonal quartic force constants. The is the "Vib.Av.Geom" structure obtained from Gaussian 09 using "freq=anharmonic".
mass24	A calculation using a ^{24}O isotopologue

O2	A calculation in which the isotopic substitution is: 
O6	A calculation in which the isotopic substitution is: 
Averagegeo0	Geometry from the average of trajectories carried out at 0 K, in curvilinear coordinates
Averagegeo193	Geometry from the average of trajectories carried out at 193 K, in curvilinear coordinates
Averagegeo298	Geometry from the average of trajectories carried out at 298 K, in curvilinear coordinates
B2displaced	Geometry obtained by displacing the 167 cm^{-1} *B2* mode by the average displacement value obtained by solving Schroedinger's for the energy + zero-point energy curve
Tight	This indicates specifically that tight convergence criteria (opt=tight in Gaussian09) were used, though most stationary point structures were obtained using tight convergence criteria without it being incorporated into the file name.

Calculated Structures and Energies

It should be noted that many of the structures below are the result of either second-order perturbation theory or the averaging of geometries from trajectories. Such structures are not stationary points on the potential energy surfaces, and the energies of these structures are not reported. Full energies are reported for calculations at stationary points.

bromoniumC2VMPWPS

mpw1pw91/6-31+G**
E(RmPW1PW91) = -2807.44363261

Zero-point correction= 0.167118 (Hartree/Particle)
Thermal correction to Energy= 0.176754
Thermal correction to Enthalpy= 0.177698
Thermal correction to Gibbs Free Energy= 0.133527
Sum of electronic and ZPE= -2807.276515
Sum of electronic and thermal Energies= -2807.266879
Sum of electronic and thermal Enthalpies= -2807.265934
Sum of electronic and thermal Free Energies= -2807.310106

	E	CV	S
	KCal/Mol	Cal/Mol-K	Cal/Mol-K
Total	110.915	34.730	92.966

C,0,0.9997397581,1.5055067515,1.2617059419
C,0,0.9997397581,1.5055067515,-1.2617059419
C,0,0.9997397581,-1.5055067515,-1.2617059419
C,0,0.9997397581,-1.5055067515,1.2617059419

C,0,0.7360095497,0.740637821,0.
C,0,0.7360095497,-0.740637821,0.
H,0,0.5494387589,2.4966384937,1.2032695064
H,0,0.5494387589,-2.4966384937,1.2032695064
H,0,0.5494387589,-2.4966384937,-1.2032695064
H,0,0.5494387589,2.4966384937,-1.2032695064
H,0,2.0859364171,1.6401751035,1.3370951466
H,0,2.0859364171,1.6401751035,-1.3370951466
H,0,2.0859364171,-1.6401751035,1.3370951466
H,0,2.0859364171,-1.6401751035,-1.3370951466
H,0,0.6533130637,1.013953161,2.16721167
H,0,0.6533130637,1.013953161,-2.16721167
H,0,0.6533130637,-1.013953161,2.16721167
H,0,0.6533130637,-1.013953161,-2.16721167
Br,0,-1.2393901257,0.,0.

bromonium6DanharmMPWPS

6	1.267187	1.509204	-0.979909
6	-1.267187	1.509204	-0.979909
6	-1.266837	-1.507266	-0.973468
6	1.266837	-1.507266	-0.973468
6	0.000000	0.745503	-0.711406
6	0.000000	-0.741395	-0.705178
1	1.199238	2.518696	-0.581373
1	1.200721	-2.510832	-0.557845
1	-1.200721	-2.510832	-0.557845
1	-1.199238	2.518696	-0.581374
1	1.368798	1.595998	-2.064440
1	-1.368798	1.595998	-2.064440
1	1.360509	-1.609632	-2.058582
1	-1.360509	-1.609632	-2.058582
1	2.169022	1.045161	-0.595694
1	-2.169022	1.045161	-0.595694
1	2.170258	-1.035292	-0.600189
1	-2.170258	-1.035292	-0.600189
35	0.000000	-0.001192	1.279209

bromonium6DB2displacedMPWPS

6	0	0.762332	-0.684661	0.000178
6	0	0.642608	0.791634	0.000268
6	0	0.845910	1.574274	1.262428
35	0	-1.267504	-0.097187	-0.000432
6	0	0.846850	1.574188	-1.261822
6	0	1.084802	-1.426021	1.262008
6	0	1.086344	-1.425858	-1.261262
1	0	0.700675	-0.964427	-2.166762
1	0	2.180007	-1.470655	-1.337112
1	0	0.719343	-2.450710	-1.202292
1	0	0.716325	-2.450347	1.203128
1	0	2.178370	-1.472400	1.338248
1	0	0.699437	-0.963873	2.167248
1	0	0.325123	2.529542	-1.201952
1	0	1.920051	1.787823	-1.340582

1	0	0.534437	1.058582	-2.166402
1	0	0.322788	2.528885	1.202688
1	0	0.534229	1.058295	2.167038
1	0	1.918795	1.789498	1.341088

bromoniumaveragegeo0MPWPS

```
 c
 c   1 cc2
 c    2 cc3        1 ccc3
br    1 brc4       2 brcc4       3 dih4
 c    2 cc5        1 ccc5        4 dih5
 c    1 cc6        2 ccc6        3 dih6
 c    1 cc7        2 ccc7        3 dih7
 h    7 hc8        1 hcc8        2 dih8
 h    7 hc9        1 hcc9        2 dih9
 h    7 hc10       1 hcc10       2 dih10
 h    6 hc11       1 hcc11       2 dih11
 h    6 hc12       1 hcc12       2 dih12
 h    6 hc13       1 hcc13       2 dih13
 h    5 hc14       2 hcc14       1 dih14
 h    5 hc15       2 hcc15       1 dih15
 h    5 hc16       2 hcc16       1 dih16
 h    3 hc17       2 hcc17       1 dih17
 h    3 hc18       2 hcc18       1 dih18
 h    3 hc19       2 hcc19       1 dih19

cc2 1.49155
cc3 1.50975
brc4 2.13508
cc5 1.5096
cc6 1.50833
cc7 1.50839
hc8 1.10886
hc9 1.1211
hc10 1.1125
hc11 1.11204
hc12 1.11994
hc13 1.10755
hc14 1.10487
hc15 1.11128
hc16 1.10149
hc17 1.105
hc18 1.10199
hc19 1.1116
ccc3 120.332
brcc4 68.9848
ccc5 120.332
ccc6 120.385
ccc7 120.392
hcc8 114.276
hcc9 106.936
hcc10 110.32
hcc11 110.316
hcc12 106.959
```

hcc13 114.262
hcc14 110.144
hcc15 107.054
hcc16 114.225
hcc17 110.157
hcc18 114.21
hcc19 107.059
dih4 102.065
dih5 -102.088
dih6 0.554975
dih7 -156.405
dih8 -32.8899
dih9 87.0471
dih10 -156.383
dih11 156.253
dih12 -87.1585
dih13 32.7739
dih14 156.5
dih15 -86.8121
dih16 33.2919
dih17 -156.627
dih18 -33.4348
dih19 86.6527

bromoniumaveragegeo193MPWPS

```
 c
 c   1 cc2
 c    2 cc3         1 ccc3
br    1 brc4        2 brcc4         3 dih4
 c    2 cc5         1 ccc5          4 dih5
 c    1 cc6         2 ccc6          3 dih6
 c    1 cc7         2 ccc7          3 dih7
 h    7 hc8         1 hcc8          2 dih8
 h    7 hc9         1 hcc9          2 dih9
 h    7 hc10        1 hcc10         2 dih10
 h    6 hc11        1 hcc11         2 dih11
 h    6 hc12        1 hcc12         2 dih12
 h    6 hc13        1 hcc13         2 dih13
 h    5 hc14        2 hcc14         1 dih14
 h    5 hc15        2 hcc15         1 dih15
 h    5 hc16        2 hcc16         1 dih16
 h    3 hc17        2 hcc17         1 dih17
 h    3 hc18        2 hcc18         1 dih18
 h    3 hc19        2 hcc19         1 dih19

cc2 1.49204
cc3 1.5102
brc4 2.13984
cc5 1.51023
cc6 1.50874
cc7 1.50875
hc8 1.10834
hc9 1.12051
hc10 1.11259
```

hc11 1.11202
hc12 1.12088
hc13 1.10844
hc14 1.10505
hc15 1.11141
hc16 1.10205
hc17 1.10515
hc18 1.10194
hc19 1.11157
ccc3 120.27
brcc4 69.0616
ccc5 120.264
ccc6 120.336
ccc7 120.332
hcc8 114.132
hcc9 107.082
hcc10 110.33
hcc11 110.324
hcc12 107.052
hcc13 114.16
hcc14 110.173
hcc15 107.143
hcc16 114.132
hcc17 110.17
hcc18 114.142
hcc19 107.124
dih4 102.08
dih5 -102.11
dih6 0.545971
dih7 -156.38
dih8 -32.7939
dih9 87.1744
dih10 -156.171
dih11 156.6
dih12 -86.7711
dih13 33.179
dih14 156.676
dih15 -86.6277
dih16 33.4908
dih17 -156.593
dih18 -33.4291
dih19 86.6816

bromoniumaveragegeo298MPWPS

```
 c
 c   1 cc2
 c    2 cc3         1 ccc3
br    1 brc4        2 brcc4         3 dih4
 c    2 cc5         1 ccc5          4 dih5
 c    1 cc6         2 ccc6          3 dih6
 c    1 cc7         2 ccc7          3 dih7
 h    7 hc8         1 hcc8          2 dih8
 h    7 hc9         1 hcc9          2 dih9
 h    7 hc10        1 hcc10         2 dih10
```

```
 h    6 hc11      1 hcc11      2 dih11
 h    6 hc12      1 hcc12      2 dih12
 h    6 hc13      1 hcc13      2 dih13
 h    5 hc14      2 hcc14      1 dih14
 h    5 hc15      2 hcc15      1 dih15
 h    5 hc16      2 hcc16      1 dih16
 h    3 hc17      2 hcc17      1 dih17
 h    3 hc18      2 hcc18      1 dih18
 h    3 hc19      2 hcc19      1 dih19

cc2 1.49289
cc3 1.51092
brc4 2.14576
cc5 1.51108
cc6 1.50951
cc7 1.50965
hc8 1.10876
hc9 1.11948
hc10 1.11232
hc11 1.11186
hc12 1.12051
hc13 1.10893
hc14 1.10522
hc15 1.11131
hc16 1.10224
hc17 1.10512
hc18 1.10216
hc19 1.11136
ccc3 120.219
brcc4 69.1149
ccc5 120.169
ccc6 120.273
ccc7 120.268
hcc8 113.929
hcc9 107.301
hcc10 110.309
hcc11 110.344
hcc12 107.245
hcc13 113.948
hcc14 110.178
hcc15 107.32
hcc16 113.976
hcc17 110.177
hcc18 114.008
hcc19 107.254
dih4 102.094
dih5 -102.081
dih6 0.529599
dih7 -156.416
dih8 -33.5472
dih9 86.4846
dih10 -156.801
dih11 155.789
dih12 -87.5032
dih13 32.4672
dih14 157.389
```

dih15 -85.8751
dih16 34.2523
dih17 -156.067
dih18 -33.0045
dih19 87.1381

phthalateMPWPStight

phthalate
mPW1PW91/6-31+G**
E(RmPW1PW91) = -608.762187781

Zero-point correction= 0.115440 (Hartree/Particle)
Thermal correction to Energy= 0.124677
Thermal correction to Enthalpy= 0.125621
Thermal correction to Gibbs Free Energy= 0.079954
Sum of electronic and ZPE= -608.646748
Sum of electronic and thermal Energies= -608.637511
Sum of electronic and thermal Enthalpies= -608.636567
Sum of electronic and thermal Free Energies= -608.682234

	E	CV	S
	KCal/Mol	Cal/Mol-K	Cal/Mol-K
Total	78.236	34.781	96.115

H,0,0.002487603,0.,0.
O,0,0.016061738,0.,1.1827678155
O,0,0.016061738,0.,-1.1827678155
C,0,1.2038311422,0.,1.6757979323
C,0,1.2038311422,0.,-1.6757979323
O,0,1.4416086775,0.,2.8844741488
O,0,1.4416086775,0.,-2.8844741488
C,0,2.4025782307,0.,0.7092881608
C,0,2.4025782307,0.,-0.7092881608
C,0,3.6398806013,0.,1.3724233108
C,0,3.6398806013,0.,-1.3724233108
C,0,4.8505592603,0.,0.696463939
C,0,4.8505592603,0.,-0.696463939
H,0,3.6001544922,0.,2.4556367347
H,0,3.6001544922,0.,-2.4556367347
H,0,5.7840820668,0.,1.2523892259
H,0,5.7840820668,0.,-1.2523892259

phthalateMPWPSmass24anharmO2

1	0.000000	0.002327	2.217732
8	0.000000	1.203336	2.194843
8	0.000000	-1.205267	2.192547
6	0.000000	1.685487	1.003076
6	0.000000	-1.688358	1.002042
8	0.000000	2.890926	0.760960
8	0.000000	-2.893863	0.758806
6	0.000000	0.710263	-0.197823
6	0.000000	-0.712765	-0.198646
6	0.000000	1.375111	-1.439604

6	0.000000	-1.376601	-1.440934
6	0.000000	0.697861	-2.653534
6	0.000000	-0.698333	-2.654293
1	0.000000	2.462251	-1.400600
1	0.000000	-2.463785	-1.402668
1	0.000000	1.255499	-3.589686
1	0.000000	-1.255099	-3.590977

phthalateMPWPSmass24anharmO6

1	0.000000	0.000900	2.217970
8	0.000000	1.203249	2.194070
8	0.000000	-1.204978	2.193718
6	0.000000	1.686273	1.002819
6	0.000000	-1.687517	1.003178
8	0.000000	2.892152	0.760359
8	0.000000	-2.892642	0.760593
6	0.000000	0.710897	-0.197776
6	0.000000	-0.712055	-0.197738
6	0.000000	1.375789	-1.439571
6	0.000000	-1.376007	-1.439986
6	0.000000	0.698272	-2.653469
6	0.000000	-0.697915	-2.653496
1	0.000000	2.462905	-1.400504
1	0.000000	-2.463150	-1.401632
1	0.000000	1.255513	-3.589956
1	0.000000	-1.254987	-3.590032

phthalateMPWSBanharmO2mass24

1	0.000000	0.001712	2.212045
8	0.000000	1.211825	2.196091
8	0.000000	-1.214987	2.193581
6	0.000000	1.689349	1.001679
6	0.000000	-1.692933	1.000289
8	0.000000	2.891993	0.752665
8	0.000000	-2.895583	0.750103
6	0.000000	0.709357	-0.196482
6	0.000000	-0.712305	-0.197476
6	0.000000	1.373486	-1.437319
6	0.000000	-1.375248	-1.438942
6	0.000000	0.697363	-2.650420
6	0.000000	-0.697948	-2.651357
1	0.000000	2.460732	-1.393237
1	0.000000	-2.462539	-1.395548
1	0.000000	1.254698	-3.587915
1	0.000000	-1.254156	-3.589534

phthalateMPWSBanharmO6mass24

1	0.000000	0.000875	2.212580
8	0.000000	1.212062	2.195646
8	0.000000	-1.214426	2.194974
6	0.000000	1.690281	1.001649

6	0.000000	-1.692040	1.001686
8	0.000000	2.893291	0.752427
8	0.000000	-2.894416	0.752043
6	0.000000	0.710131	-0.196149
6	0.000000	-0.711456	-0.196237
6	0.000000	1.374260	-1.437031
6	0.000000	-1.374349	-1.437710
6	0.000000	0.697987	-2.650122
6	0.000000	-0.697292	-2.650329
1	0.000000	2.461478	-1.392719
1	0.000000	-2.461612	-1.394324
1	0.000000	1.254998	-3.587890
1	0.000000	-1.253923	-3.588271

isopropylBrMPWPS

2-bromopropane
mpw1pw91/6-31+G**
E(RmPW1PW91) = -2690.41128287

Zero-point correction= 0.095189 (Hartree/Particle)
Thermal correction to Energy= 0.100485
Thermal correction to Enthalpy= 0.101430
Thermal correction to Gibbs Free Energy= 0.066046
Sum of electronic and ZPE= -2690.316094
Sum of electronic and thermal Energies= -2690.310797
Sum of electronic and thermal Enthalpies= -2690.309853
Sum of electronic and thermal Free Energies= -2690.345237

	E	CV	S
	KCal/Mol	Cal/Mol-K	Cal/Mol-K
Total	63.056	18.131	74.472

C,0,-0.0202968924,0.,0.0129844704
H,0,0.066198526,0.,1.1000674359
Br,0,1.8559252711,0.,-0.6032467526
C,0,-0.6910170065,-1.2625726718,-0.4859757102
C,0,-0.6910170065,1.2625726718,-0.4859757102
H,0,-0.701960796,1.2841335191,-1.5784081401
H,0,-1.7270901824,1.2888445794,-0.1303352444
H,0,-0.1826824662,2.1596107731,-0.128678482
H,0,-0.701960796,-1.2841335191,-1.5784081401
H,0,-0.1826824662,-2.1596107731,-0.128678482
H,0,-1.7270901824,-1.2888445794,-0.1303352444

isopropylBrD6anharmMPWPS

6	-0.463306	-0.902769	0.000000
1	-1.561064	-0.866357	0.000000
35	0.066187	1.007367	0.000000
6	0.067406	-1.552472	1.268937
6	0.067406	-1.552472	-1.268937
1	1.160374	-1.521978	-1.290367
1	-0.246711	-2.601989	-1.300155
1	-0.307407	-1.057786	-2.167222
1	1.160374	-1.521978	1.290367

1	-0.307407	-1.057786	2.167222
1	-0.246711	-2.601989	1.300155

isopropylBranharmMPWPS (unlabeled)

6	-0.462608	-0.903363	0.000000
1	-1.560421	-0.866246	0.000000
35	0.066289	1.007395	0.000000
6	0.067065	-1.553702	1.269550
6	0.067065	-1.553702	-1.269550
1	1.159804	-1.528141	-1.291504
1	-0.250243	-2.601665	-1.303186
1	-0.304688	-1.057913	-2.168022
1	1.159804	-1.528141	1.291504
1	-0.304688	-1.057913	2.168022
1	-0.250243	-2.601665	1.303186

bromoniumB3BBC2V

bromonium ion structure
B3LYP/6-311+G**
E(RB3LYP) = -2809.79994541

Zero-point correction= 0.164900 (Hartree/Particle)
Thermal correction to Energy= 0.174673
Thermal correction to Enthalpy= 0.175617
Thermal correction to Gibbs Free Energy= 0.131113
Sum of electronic and ZPE= -2809.635045
Sum of electronic and thermal Energies= -2809.625272
Sum of electronic and thermal Enthalpies= -2809.624328
Sum of electronic and thermal Free Energies= -2809.668833

	E	CV	S
	KCal/Mol	Cal/Mol-K	Cal/Mol-K
Total	109.609	35.059	93.668

C,0,1.00009,1.50651,1.264721
C,0,1.00009,1.50651,-1.264721
C,0,1.00009,-1.50651,-1.264721
C,0,1.00009,-1.50651,1.264721
C,0,0.73109,0.741039,0.
C,0,0.73109,-0.741039,0.
H,0,0.5824,2.511144,1.200904
H,0,0.5824,-2.511144,1.200904
H,0,0.5824,-2.511144,-1.200904
H,0,0.5824,2.511144,-1.200904
H,0,2.0904,1.610634,1.348115
H,0,2.0904,1.610634,-1.348115
H,0,2.0904,-1.610634,1.348115
H,0,2.0904,-1.610634,-1.348115
H,0,0.63682,1.024933,2.168121
H,0,0.63682,1.024933,-2.168121
H,0,0.63682,-1.024933,2.168121
H,0,0.63682,-1.024933,-2.168121
Br,0,-1.314679,0.,0.

bromoniumC2VB3PS

B3LYP/6-31+G**
E(RB3LYP) = -2807.34218254

Zero-point correction= 0.165745 (Hartree/Particle)
Thermal correction to Energy= 0.175502
Thermal correction to Enthalpy= 0.176446
Thermal correction to Gibbs Free Energy= 0.131934
Sum of electronic and ZPE= -2807.176438
Sum of electronic and thermal Energies= -2807.166681
Sum of electronic and thermal Enthalpies= -2807.165737
Sum of electronic and thermal Free Energies= -2807.210249

	E	CV	S
	KCal/Mol	Cal/Mol-K	Cal/Mol-K
Total	110.129	34.966	93.684

C,0,1.0016974263,1.5131632976,1.2666507146
C,0,1.0016974263,1.5131632976,-1.2666507146
C,0,1.0016974263,-1.5131632976,-1.2666507146
C,0,1.0016974263,-1.5131632976,1.2666507146
C,0,0.7421361843,0.7424946075,0.
C,0,0.7421361843,-0.7424946075,0.
H,0,0.5478142855,2.5041575839,1.2061936124
H,0,0.5478142855,-2.5041575839,1.2061936124
H,0,0.5478142855,-2.5041575839,-1.2061936124
H,0,0.5478142855,2.5041575839,-1.2061936124
H,0,2.0894036211,1.6527290401,1.3431077996
H,0,2.0894036211,1.6527290401,-1.3431077996
H,0,2.0894036211,-1.6527290401,1.3431077996
H,0,2.0894036211,-1.6527290401,-1.3431077996
H,0,0.6566269571,1.0206626041,2.1733571751
H,0,0.6566269571,1.0206626041,-2.1733571751
H,0,0.6566269571,-1.0206626041,2.1733571751
H,0,0.6566269571,-1.0206626041,-2.1733571751
Br,0,-1.2801004647,0.,0.

bromoniumC2VM06PS

M06/6-31+G**
E(RM06) = -2806.98256864

Zero-point correction= 0.164433 (Hartree/Particle)
Thermal correction to Energy= 0.174306
Thermal correction to Enthalpy= 0.175250
Thermal correction to Gibbs Free Energy= 0.130464
Sum of electronic and ZPE= -2806.818136
Sum of electronic and thermal Energies= -2806.808263
Sum of electronic and thermal Enthalpies= -2806.807319
Sum of electronic and thermal Free Energies= -2806.852105

	E	CV	S
	KCal/Mol	Cal/Mol-K	Cal/Mol-K
Total	109.379	35.266	94.261

C,0,0.9999902805,1.5027883093,1.2573106384
C,0,0.9999902805,1.5027883093,-1.2573106384
C,0,0.9999902805,-1.5027883093,-1.2573106384
C,0,0.9999902805,-1.5027883093,1.2573106384
C,0,0.7420943535,0.7368271596,0.
C,0,0.7420943535,-0.7368271596,0.
H,0,0.545903961,2.4953510937,1.1949570855
H,0,0.545903961,-2.4953510937,1.1949570855
H,0,0.545903961,-2.4953510937,-1.1949570855
H,0,0.545903961,2.4953510937,-1.1949570855
H,0,2.0874578945,1.6436975074,1.3379765368
H,0,2.0874578945,1.6436975074,-1.3379765368
H,0,2.0874578945,-1.6436975074,1.3379765368
H,0,2.0874578945,-1.6436975074,-1.3379765368
H,0,0.6513662093,1.0148095678,2.1678070592
H,0,0.6513662093,1.0148095678,-2.1678070592
H,0,0.6513662093,-1.0148095678,2.1678070592
H,0,0.6513662093,-1.0148095678,-2.1678070592
Br,0,-1.2367210174,0.,0.

bromoniumC2VBP86PS

BP86/6-31+G**
E(RB-P86) = -2807.58048967

Zero-point correction= 0.161057 (Hartree/Particle)
Thermal correction to Energy= 0.170971
Thermal correction to Enthalpy= 0.171915
Thermal correction to Gibbs Free Energy= 0.127132
Sum of electronic and ZPE= -2807.419433
Sum of electronic and thermal Energies= -2807.409518
Sum of electronic and thermal Enthalpies= -2807.408574
Sum of electronic and thermal Free Energies= -2807.453357

	E	CV	S
	KCal/Mol	Cal/Mol-K	Cal/Mol-K
Total	107.286	35.773	94.254

C,0,0.9999199855,1.5172566321,1.2708751244
C,0,0.9999199855,1.5172566321,-1.2708751244
C,0,0.9999199855,-1.5172566321,-1.2708751244
C,0,0.9999199855,-1.5172566321,1.2708751244
C,0,0.7381025231,0.7496486178,0.
C,0,0.7381025231,-0.7496486178,0.
H,0,0.5451508076,2.5177356758,1.2161759805
H,0,0.5451508076,-2.5177356758,1.2161759805
H,0,0.5451508076,-2.5177356758,-1.2161759805
H,0,0.5451508076,2.5177356758,-1.2161759805
H,0,2.0963516936,1.6580233323,1.34750531
H,0,2.0963516936,1.6580233323,-1.34750531
H,0,2.0963516936,-1.6580233323,1.34750531
H,0,2.0963516936,-1.6580233323,-1.34750531
H,0,0.6557928456,1.0177829425,2.185132765
H,0,0.6557928456,1.0177829425,-2.185132765
H,0,0.6557928456,-1.0177829425,2.185132765

H,0,0.6557928456,-1.0177829425,-2.185132765
Br,0,-1.2787255037,0.,0.

bromonium6DB3BBanharm

6	0.742462	0.747303	0.000000
6	0.732566	-0.740620	0.000000
35	-1.323140	-0.002328	0.000000
6	1.012131	1.513799	1.269948
1	0.628908	2.527691	1.200336
1	0.623628	1.055302	2.171173
1	2.097166	1.589782	1.374895
6	1.012131	1.513799	-1.269948
6	1.004110	-1.509391	-1.269652
6	1.004110	-1.509391	1.269652
1	0.628908	2.527691	-1.200336
1	2.097166	1.589782	-1.374895
1	0.623628	1.055302	-2.171173
1	0.605960	-2.518900	1.201250
1	2.090551	-1.597833	1.366477
1	0.624859	-1.043598	2.172471
1	0.605960	-2.518900	-1.201250
1	0.624859	-1.043598	-2.172471
1	2.090551	-1.597833	-1.366477

bromoniumB3BBanharm

6	1.008925	1.511741	1.269872
6	0.738432	0.744081	0.000000
1	0.621671	2.524002	1.200618
1	0.622969	1.052576	2.171796
1	2.093603	1.591718	1.373122
6	0.738431	-0.744080	0.000000
6	1.008925	1.511741	-1.269872
6	1.008925	-1.511740	-1.269872
6	1.008925	-1.511740	1.269872
1	0.621671	2.524002	-1.200618
1	2.093603	1.591718	-1.373122
1	0.622969	1.052576	-2.171796
1	0.621670	-2.524000	1.200619
1	2.093600	-1.591720	1.373119
1	0.622972	-1.052575	2.171796
1	0.621670	-2.524000	-1.200619
1	0.622972	-1.052575	-2.171796
1	2.093600	-1.591720	-1.373119
35	-1.323747	0.000000	0.000000

formicacidMPW

formic acid

RmPW1PW91/6-31G*
E(RmPW1PW91) = -189.705188937

Zero-point correction= 0.034428 (Hartree/Particle)
Thermal correction to Energy= 0.037572
Thermal correction to Enthalpy= 0.038517
Thermal correction to Gibbs Free Energy= 0.010362
Sum of electronic and ZPE= -189.670761
Sum of electronic and thermal Energies= -189.667616
Sum of electronic and thermal Enthalpies= -189.666672
Sum of electronic and thermal Free Energies= -189.694827

	E	CV	S
	KCal/Mol	Cal/Mol-K	Cal/Mol-K
Total	23.577	8.494	59.257

H,0,0.0922201568,0.,0.1541659907
O,0,-0.058184235,0.,1.115476464
C,0,1.1500969642,0.,1.6922684226
O,0,2.1970595421,0.,1.1039623386
H,0,1.0355371312,0.,2.7841267841

formicacidMPWanharm (no label)

1	-0.660875	-1.336333	0.000000
8	-1.034973	-0.437524	0.000000
6	0.001404	0.421497	0.000000
8	1.163231	0.106100	0.000000
1	-0.380261	1.461093	0.000000

formicacidMPWanharmCO24 (^{24}O in carbonyl oxygen of formic acid)

1	-0.661521	-1.336230	0.000000
8	-1.035461	-0.437558	0.000000
6	0.000891	0.421394	0.000000
8	1.162456	0.105894	0.000000
1	-0.380657	1.460936	0.000000

formicacidMPWanharmOH24 (^{24}O in OH oxygen of formic acid)

1	-0.660096	-1.335719	0.000000
8	-1.034108	-0.436960	0.000000
6	0.001900	0.421728	0.000000
8	1.163680	0.106276	0.000000
1	-0.379663	1.461313	0.000000

Miscellaneous Tables of Energies

Energies of structures obtained from stepping along the 167 cm^{-1} (169 in the labeled compound) *B2* mode in the bromoniumC2VMPWPS structure. These are all MPW1PW91/6-31+G**. The E + zpe listed is the raw calculated value including all modes.

Stepsize (Å)	potential E	E + zpe
-0.4	-2807.442487	-2807.295059
-0.3	-2807.443004	-2807.295536
-0.2	-2807.443339	-2807.295925
-0.1	-2807.443512	-2807.296124
0	-2807.443633	-2807.296111
0.1	-2807.443508	-2807.296076
0.2	-2807.443297	-2807.295747
0.3	-2807.442927	-2807.295224
0.4	-2807.442373	-2807.294545

Energies of structures obtained from stepping along the 133 cm^{-1} mode in the bromoniumB3BBC2V6D structure. These are all B3LYP/6-311+G**. The E + zpe listed is the raw calculated value including all modes.

Stepsize (Å)	potential E	E + zpe
-0.4	-2809.798934	-2809.653348
-0.3	-2809.799429	-2809.653835
-0.2	-2809.799734	-2809.654162
-0.1	-2809.799896	-2809.654354
0	-2809.799945	-2809.654418
0.1	-2809.799898	-2809.654348
0.2	-2809.799749	-2809.654148
0.3	-2809.799479	-2809.653821
0.4	-2809.799053	-2809.653358

Listing of Subprograms for Program Suite PROGDYN

Program progdynstarterHP

```
#!/bin/bash
# This is the master control program for dynamics, in the form of a Unix Shell Script.
#
# Necessary 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 contains explanations of many of the program options.
#
# Optional input:
# isomernumber      - A number in file isomernumber provides a start for numbering runs.
# 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
# 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
#
# Programs called:
#  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, used the 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 often 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.
#
# 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 and olderdynrun – files containing the last two 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.
# dyn  - A record of the Gaussian outputs.
# dynfollowfile – A short record of the runs and their results.
# NMRlist or NMRlistdis – output of NMR predictions at each point in a trajectory
# 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 list the velocities of each atom, decided by variable in progdyn.conf
# A number of files starting with 'temp' are created then later erased.

#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
#
#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 calculatons but the routine in prog2ndpoint would have to be changed for other calcs.
#
#                                   OUTLINE
# A. initilize to perform Gaussian jobs and know where we are
#    start loop
# B. 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
# C. loop over propagation steps
```

```
#
#  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
#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=/apps/lms/g09_A02_XEON
. $g09root/g09/bsd/g09.profile
origdir=`pwd`
cd $origdir
logfile=docslog
randdir=~/bin
cp /scratch/d-singleton/binall300/* /tmp/$PBS_JOBID
proggramdir=/tmp/$PBS_JOBID
freqfile=/scratch/d-singleton/binall300/freqinHP
echo ORIGDIR:
echo $origdir
echo SCRATCHDIR:
echo $scratchdir
echo PROGGRAMDIR:
echo $proggramdir

rm -f nogo    # assume that if someone is starting a job, they want it to go.
rm -f diagnostics # contains extra info from start of progFS

#### Triple 'while' loop - will have to break multiple times to get out, but advantage is ability to control starting over
while (true)
do

# As long as there is a file "goingwell" the program will not exit entirely by itself
rm -f goingwell
while (true)
do
#  BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
  if (test -f skipstart) then
    echo "skipping start and continuing from previous runs"
  else
#  B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1B1 generate geoPlusVel and
first input file
    if [ `cat runpointnumber` = "1" ]; then
      echo "XXXX did not complete first point so new isomer started" >> dynfollowfile
    fi
    if [ `cat runpointnumber` = "2" ]; then
      echo "XXXX did not complete second point so new isomer started" >> dynfollowfile
    fi
    if [ `cat runpointnumber` = "3" ]; then
      echo "XXXX did not complete third point so new isomer started" >> dynfollowfile
    fi
    cd $origdir
    $randdir/randgen > temp811
# the next 8 lines would have to be changed to use low-precision modes
    awk '/       1       2       3       4/,/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) print}' $freqfile > tempstangeos
    awk -f $proggramdir/proggenHP $freqfile > geoPlusVel
    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
#  B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2B2  if first part
successfule 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
      cat isomernumber >> geoRecord
      cat geoPlusVel >> geoRecord
      rm -f 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 > goingwell
      if (test -s goingwell) then
        cat $scratchdir/g09.log >> dyn
        cp $scratchdir/g09.log olderdynrun
      else
        cp $scratchdir/g09.log $origdir/g09.log
        break
      fi
    else
      break
    fi
#  B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3B3 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 tempdone
    tail -1 dynfollowfile | awk '/XXXX/ {print}' > tempdone
    if (test  -s tempdone) then
      rm -f dyn
      rm -f traj
      echo 0 > runpointnumber
      break
    fi
    if (test -s g09.com) then
      rm -f 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 > goingwell
      if (test -s goingwell) then
        cp $scratchdir/g09.log olddynrun
        cat $scratchdir/g09.log >> dyn
        awk -f $proggramdir/proganal $scratchdir/g09.log >> dynfollowfile
        awk '/Input orientation/,/Distance matrix/ {print}' olddynrun | awk '/   0   / {print}' > old
        awk '/Input orientation/,/Distance matrix/ {print}' olderdynrun | awk '/   0   / {print}' > older
        echo 3 > runpointnumber
        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
# 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 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 > goingwell
      if (test -s goingwell) then
        cp $scratchdir/g09.log olderdynrun
      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 tempdone
    if (test -s g09.com) then
      rm -f 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 > goingwell
      if (test -s goingwell) then
```

```
          cp $scratchdir/g09.log olddynrun
          cat $scratchdir/g09.log >> dyn
          awk -f $proggramdir/proganal $scratchdir/g09.log >> dynfollowfile
          awk '/Input orientation/,/Distance matrix/ {print}' olddynrun | awk '/   0   / {print}' > old
          awk '/Input orientation/,/Distance matrix/ {print}' olderdynrun | awk '/   0   / {print}' > older
          echo 3 > runpointnumber
          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___

#  CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
propagation loop
  while (true)
  do
#increment runpointnumber
      cp runpointnumber temp533
      awk 'BEGIN {getline;i=$1+1;print i}' temp533 > runpointnumber
      rm temp533
      rm -f 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 > goingwell
      if (test -s goingwell) then
        mv olddynrun olderdynrun
        cp $scratchdir/g09.log olddynrun
        awk '/Input orientation/,/Distance matrix/ {print}' olddynrun | awk '/   0   / {print}' > old
        awk '/Input orientation/,/Distance matrix/ {print}' olderdynrun | awk '/   0   / {print}' > older
        awk -f $proggramdir/progdynb olddynrun > g09.com
        rm -f old older
      else
        cp $scratchdir/g09.log $origdir/g09.log
        break
      fi

# here is a cool link that lets you interupt 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
      awk -f $proggramdir/proganal $scratchdir/g09.log >> dynfollowfile
      rm -f tempdone
      tail -2 dynfollowfile | awk '/XXXX/ {print}' > tempdone
      if (test  -s 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 olderdynrun
            a=`awk '{print $1}' isomernumber`
            mv traj traj$a
            mv dyn dyn$a
          fi
          if [ `cat skipstart` = "forward" ]; then
            echo reverserestart > skipstart
          fi
        else
          rm -f skipstart
          rm -f geoPlusVel
          rm -f olddynrun
          rm -f olderdynrun
          a=`awk '{print $1}' isomernumber`
          mv traj traj$a
          mv dyn dyn$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____E
ND_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 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 goingwell) then
    echo "starting a new point or a new direction2"
  else
    break
  fi
done
exit 0
```

Program proggenHP

```
BEGIN {
# Aut 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 incorportate 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

#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=="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=="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
  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
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(); 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
}
# 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 compatability 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
  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]==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 compatability
  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) {
  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)
  }

# 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
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])
  }
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
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
```

```
}
```

Program prog1stpoint

```
BEGIN {
# 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 - incorportates 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

# 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
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; linkatoms=0

#initialization
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

# read progdyn.conf for configuration info
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=="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=="method3") meth3=$2
  if ($1=="method4") meth4=$2
  if ($1=="method5") meth5=$2
  if ($1=="method6") meth6=$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=="methodfile") methodfilelines=$2
  if ($1=="killcheck") killcheck=$2
  if ($1=="title") {
    title1=$2
    title2=$3
    title3=$4
    title4=$5
    }
  blankLineTester=length($0)
  }

if (diag==1) print "***************** starting prog1stpoint *****************" >> "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" >> "diagnostics"
if (diag==1) print processors,checkpoint,title1,title2,title3,title4 >> "diagnostics"

getline < "isomernumber"
isomernum = $1
#read in number of atoms, geometry, masses, and velocity 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 not needed for 1st point
for (i=1;i<=numAtoms;i++) {
  getline < "geoPlusVel"
  for (j=1;j<=3;j++) {
    velArr[i,j]=$j
    }
  }

print "%nproc=" processors
print "%mem=" memory
if (killcheck!=1) print "%chk=" checkpoint
print "# " method " force scf=(tight,nosym) "
if (meth2=="unrestricted") print "guess=mix" #for unrestricted calculations
if (length(meth3)>2) print meth3
if (length(meth4)>2) print meth4
print ""
# make the title four words exactly, leaving out spaces if necessary
print title1,title2,title3,title4
print "runpoint 1"
print "runisomer ", isomernum
print ""
print charge,multiplicity
}

END {
```

```
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 ""
}
```

Program prog2ndpoint

```
BEGIN {
#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

# 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
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; linkatoms=0
etolerance=1

#initialization
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

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=="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=="method3") meth3=$2
  if ($1=="method4") meth4=$2
  if ($1=="method5") meth5=$2
  if ($1=="method6") meth6=$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=="DRP") DRP=$2
  if ($1=="methodfile") methodfilelines=$2
  if ($1=="killcheck") killcheck=$2
  if ($1=="etolerance") etolerance=$2
  if ($1=="reversetraj") reversetraj=$2
  if ($1=="title") {
    title1=$2
    title2=$3
    title3=$4
    title4=$5
    }
  blankLineTester=length($0)
  }

if (diag>=1) print "***************** starting prog2ndpoint *****************" >> "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" >> "diagnostics"
if (diag>=1) print processors,checkpoint,title1,title2,title3,title4 >> "diagnostics"

#get the isomer number from file
getline < "isomernumber"
isomernum = $1

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

print "%nproc=" processors
print "%mem=" memory
if (killcheck!=1) print "%chk=" checkpoint
print "# " method " force scf=(tight,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
print ""
print title1,title2,title3,title4
print "runpoint 2"
print "runisomer ", isomernum
print ""
print charge,multiplicity

# ok, now we have to figure the second point.  this should be
```

```
# x(t) = x + v*t + 1/2*F*t^2/m
# so we need to set up arrays for position, velocity, and force

#read in number of atoms, geometry, masses, and velocity 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
    }
  }

#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"
  }

#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)
  }
#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"
  }
#added by Samae on 102910
scfcount=0
} # end of BEGIN

#pull out the potential energy
```

```
/SCF Done/ || /EUMP2 =/ || / Energy=/ {
if (($1=="Energy=") && ($3=="NIter="))  newPotentialE=$2
if (($1=="SCF") && (scfcount==0)) newPotentialE=$5
if ($1=="E2") {
  tempstring=$6
  split(tempstring, arr10, "D")
  newPotentialE=arr10[1]*(10^arr10[2])
  }
newPotentialEK=(newPotentialE-potentialE)*627.509
if ($1=="SCF") {
  if (scfcount==0) {
    pddga=$5
    }
  if (scfcount==1) {
    qm=$5
    }
  if (scfcount==2) {
    pddgb=$5
    pddgc=(pddga-pddgb)
    newPotentialE=(qm+pddgc)
    newPotentialEK=(newPotentialE-potentialE)*627.509
    }
  scfcount++
  }
}

# now we go ahead and translate the forces and add them
(/  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 {
#put out Echeck but only if not a DRP
if (DRP==0) {
  print "trajectory #",isomernum >> "Echeck"
  print "point 1 potential E=",newPotentialEK,"   point 1 kinetic E=",KEinittotal,"
Total=",newPotentialEK+KEinittotal >> "Echeck"
  print "desired total energy=", desiredModeEnK >> "Echeck"
  if ((newPotentialEK+KEinittotal)>(desiredModeEnK+etolerance)) print "XXXX bad total Energy" >> "Echeck"
  if ((newPotentialEK+KEinittotal)<(desiredModeEnK-etolerance)) print "XXXX bad total Energy" >> "Echeck"
  }
# 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"
  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
     }
  }
print ""
#get second geometry into file traj
print numAtoms >> "traj"
print newPotentialE,title1,title2,title3,title4,"runpoint 2 ","runisomer ",isomernum >> "traj"
for (i=1;i<=numAtoms;i++) {
  print atSym[i],arr[i,1],arr[i,2],arr[i,3] >> "traj"
  }
}
```

Program progdynb

```
BEGIN { #this is the main routine for generating new .com files by the Verlet algorithym
# 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

# 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
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; linkatoms=0
```

```
damping=1

#initialization
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
conver1=4.184E26 #dividing by this converts amu angs^2 /s^2 to kcal/mol
OFS="     "

# read progdyn.conf for configuration info
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=="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=="method3") meth3=$2
  if ($1=="method4") meth4=$2
  if ($1=="method5") meth5=$2
  if ($1=="method6") meth6=$2
  if ($1=="highlevel") highlevel=$2
  if ($1=="linkatoms") linkatoms=$2
  if ($1=="keepevery") keepevery=$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=="DRP") DRP=$2
  if ($1=="maxAtomMove") maxAtomMove=$2
  if ($1=="methodfile") methodfilelines=$2
  if ($1=="killcheck") killcheck=$2
  if ($1=="damping") damping=$2
  if ($1=="title") {
    title1=$2
    title2=$3
    title3=$4
    title4=$5
    }
  blankLineTester=length($0)
  }

if (diag>=1) print "***************** starting progdynb *****************" >> "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" >> "diagnostics"
if (diag>=1) print processors,checkpoint,title1,title2,title3,title4 >> "diagnostics"
```

```
# 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
  }

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
  }

#for DRPs read in oldAdjForces and maxAtomMove
if (DRP==1) {
  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
  }

# record atom velocities for IVR analysis.  This is actually the velocity in the previous run, which is the easiest to
calculate.
getline < "isomernumber"
isomernum = $1
getline < "runpointnumber"
runpointnum = $1
if (diag==3) print "runpoint ",runpointnum-1,"runisomer ",isomernum >> "vellist"
for (at=1;at<=numAtoms;at++) {
  atomVel=((oldarr[at,1]-olderarr[at,1])^2 + (oldarr[at,2]-olderarr[at,2])^2 +(oldarr[at,3]-olderarr[at,3])^2)^.5
  KEatomstotal=KEatomstotal+0.5*weight[at]*(atomVel^2)/((timestep^2)*conver1)
  if (diag==3) print atomVel >> "vellist"
  }
apparentTemp=KEatomstotal*2/(3*RgasK*numAtoms)
if (diag==4) print "KEatomstotal",KEatomstotal,"apparent Temperature",apparentTemp >> "vellist"
}

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

```
#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 {
#############routine for DRPs##############
if (DRP==1) {
  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]
      }
    }
  }
########

#############normal routine for Verlet ##############
if (DRP==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))
        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]
          }
        }
      }
    }
########

if ((runpointnum % keepevery)==0) system("cat g09.log >> dyn")
print "%nproc=" processors
print "%mem=" memory
if (killcheck!=1) print "%chk=" checkpoint
print "# " method " force scf=(maxcycle=200)"
if (meth2=="unrestricted") print "guess=mix" #for unrestricted calculations
if (meth2=="read") print "guess=tcheck" #for reading orbitals from check, sometimes faster, sometimes not
print "pop=none "
if (length(meth3)>2) print meth3
if (length(meth4)>2) print meth4
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 ""
print charge,multiplicity
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])
  printf("%s %.7f %.7f %.7f",atSym[i],newarr[i,1],newarr[i,2],newarr[i,3]) >> "traj"
  print "" >> "traj"
  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 ""
}
```

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<=10000)
  {
    d = drand48();
    printf ("%.20f\n", d);
    count++;
  }
  return 0;
}

/* Function returns the product of the two values provided
int product(int x, int y)
{
   return (x * y);
}
*/
```

Program proganal

```
BEGIN {
firsttitle=1
getline < "isomernumber"
isomer=$1
}
/ phthalate/ {
  if (firsttitle==1) {
    printf("%s %s %s %s %s %s %s  ",$1,$2,$3,$4,$6,$7,$8)
    print $1,$2,$3,$4,$6,$7,$8 >> "NMRlistdis"
    runpoint=$6
    }
  firsttitle++
  }
/Standard orientation/,/Rotational constants/ {
  if (($1>.5) && ($1<30)) {
    A[$1]=$4;B[$1]=$5;C[$1]=$6
    }
  }
/C    Isotropic/ {print >> "NMRlistdis"}
```

```
END {
  printf("%s %.5f  %s %.5f ","H1O2",Distance(1,2),"H1O3",Distance(1,3))
  if (runpoint>500) {
    print "Too many points.  XXXX"
    getline < "skipstart"
#    if ($1=="reverse") system("date > nogo")
    }
  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(Atom2,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

The progdyn.conf below is the one used for the most extensive phthalate ion calculations. Other calculations would differ in logical ways in the lines labeled title, method, temperature, and charge.

```
#This is the configuration file for PROGDYN.  This file is read by progdynstarterHP 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 mPW1PW91/6-31G*
#*** 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 restricted
charge -1
multiplicity 1
processors 2
#*** memory --The following "word" is copied exactly to the gaussian input file after %mem=.
memory 200mw
#*** 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 hastle 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 0
#*** title -- the title keyword must be followed by exactly four words
title phthalate dyn mpw 298dis2
#*** 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 2
#*** 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
#*** 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 scrf=(pcm,Solvent=water)
#add the line below with big structures to get it to put out the distance matrix and the input orientation
#method3 iop(2/9=2000)
#method4 scrf=(pcm,solvent=dmso,read)
#method4 IOp(3/76=1000001970)IOp(3/77=0800008000)IOp(3/78=0700010000)
#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 10
#*** 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 negative
#*** 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 0
#*** 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 99
#*** 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 999
```

```
#linkatoms 1
#*** 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 2
#fixedatom2 3
#fixedatom3 19
#*** 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 0
boxsize 7.5
#*** 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
#*** 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 1.0
#*** 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 3 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
#*** reversetraj --This keyword sets the trajectories so that both directions from a transition state are explored.
reversetraj true


#updated Aug 9, 2007 to include the possibility of classical dynamics by the keyword classical
#updated Jan 2008 to include fixed atoms, ONIOM jobs, keepevery, and box size
#update Feb 2008 to include methodfile parameter
# updated Nov 2008 to allow for start without an initial freq calc using classical = 2
# update Aug 2010 to include etolerance, damping controlphase and reversetraj
```

methodfile

The methodfile below is the one used for the most extensive phthalate ion calculations. Other calculations would differ in logical ways in the calculational method employed and the charge.

```
--link1--
%chk=g09.chk
%nproc=2
%mem=200mw
#t mPW1PW91/6-31G* geom=check guess=read  nmr=giao
pop=none

NMR calc

-1     1
```

Listings of Miscellaneous programs used

Program progNMRt

```
#  This is an awk program that takes the predicted isotropic shielding constants from files NMRlist or NMRlistdis
and calculates the average shielding constant for each complete trajectory.  It then outputs the difference in the
average shift for the ipso carbons in the file out89.
# It is used with the command awk –f progNMRt filelist
BEGIN {
countc1=0.00000000000000000001
countc9=0.00000000000000000001
totali1=0
totali2=0
totali8=0
totali9=0
print "" > "out45"
print "" > "out89"
print "" > "outother"
print "" > "out"
}

/phthalate/ {
oldpoint=point
point=$5
oldisomer=isomer
isomer=$7
if ((oldpoint>1) && ((isomer!=oldisomer) || (FILENAME!=OLDFILENAME))) {
  print OLDFILENAME,"isomer",oldisomer,"count",countc1,"isotropic difference C4C5  ",totali2/countc1-
totali1/countc1,"     isotropic difference C8C9      ",totali9/countc1-totali8/countc1
  print OLDFILENAME,"isomer",oldisomer,"count",countc1,"isotropic difference C4C5  ",totali2/countc1-
totali1/countc1,"     isotropic difference C8C9      ",totali9/countc1-totali8/countc1 >> "out"
  if (countc1!=countc9) print "royalfuckup"
  if (countc1!=countc9) print "royalfuckup" >> "out"
  if ((countc1>800) && (countc1<1001)) {
    print totali2/countc1-totali1/countc1 >> "out45"
    print totali9/countc1-totali8/countc1 >> "out89"
    }
  if ((countc1<801) || (countc1>1000)) {
    print OLDFILENAME,"isomer",oldisomer,"count",countc1,"isotropic difference C4C5  ",totali2/countc1-
```

```
totali1/countc1,"     isotropic difference C8C9      ",totali9/countc1-totali8/countc1 >> "outother"
     }
  countc1=0.00000000000000000001
  countc9=0.00000000000000000001
  totali1=0
  totali2=0
  totali8=0
  totali9=0
  }
OLDFILENAME=FILENAME

}
/4  C    Isotropic/ {
if (point>1) {
  countc1++
  totali1=totali1+$5
  }
}
/5  C    Isotropic/ {
if (point>1) {
  totali2=totali2+$5
  }
}
/8  C    Isotropic/ {
if (point>1) {
  totali8=totali8+$5
  }
}
/9  C    Isotropic/ {
if (point>1) {
  countc9++
  totali9=totali9+$5
  }
}


END {
  print FILENAME,"isomer",oldisomer,"count",countc1,"isotropic difference C4C5  ",totali2/countc1-
totali1/countc1,"     isotropic difference C8C9      ",totali9/countc1-totali8/countc1
  print FILENAME,"isomer",oldisomer,"count",countc1,"isotropic difference C4C5  ",totali2/countc1-
totali1/countc1,"     isotropic difference C8C9      ",totali9/countc1-totali8/countc1 >> "out"
}
```

Program progaveragetraj

```
#  This is an awk program that takes the bromonium ion trajectory geometries output by PROGDYN in files named
traj# or traj## and averages their geometries in curvilinear coordinates.
# It is used with the command awk –f progaveragetraj filelist
BEGIN {
count=0
for(i=1;i<=19;i++) {
  x[i]=0;y[i]=0;z[i]=0
  }
atomnum=0
}
```

```
{
if (($1=="C") || ($1=="Br") || ($1=="H")) {
  atomnum++
  x[atomnum]=$2;y[atomnum]=$3;z[atomnum]=$4
  if (atomnum==19) {
  atomnum=0
  count++
  cc2=cc2+Distance(1,2)
  cc3=cc3+Distance(2,3)
  brc4=brc4+Distance(1,4)
  cc5=cc5+Distance(2,5)
  cc6=cc6+Distance(1,6)
  cc7=cc7+Distance(1,7)
  hc8=hc8+Distance(8,7)
  hc9=hc9+Distance(9,7)
  hc10=hc10+Distance(10,7)
  hc11=hc11+Distance(11,6)
  hc12=hc12+Distance(12,6)
  hc13=hc13+Distance(13,6)
  hc14=hc14+Distance(14,5)
  hc15=hc15+Distance(15,5)
  hc16=hc16+Distance(16,5)
  hc17=hc17+Distance(17,3)
  hc18=hc18+Distance(18,3)
  hc19=hc19+Distance(19,3)
  ccc3=ccc3+Angle(3,2,1)
  brcc4=brcc4+Angle(4,1,2)
  ccc5=ccc5+Angle(5,2,1)
  ccc6=ccc6+Angle(6,1,2)
  ccc7=ccc7+Angle(7,1,2)
  hcc8=hcc8+Angle(8,7,1)
  hcc9=hcc9+Angle(9,7,1)
  hcc10=hcc10+Angle(10,7,1)
  hcc11=hcc11+Angle(11,6,1)
  hcc12=hcc12+Angle(12,6,1)
  hcc13=hcc13+Angle(13,6,1)
  hcc14=hcc14+Angle(14,5,2)
  hcc15=hcc15+Angle(15,5,2)
  hcc16=hcc16+Angle(16,5,2)
  hcc17=hcc17+Angle(17,3,2)
  hcc18=hcc18+Angle(18,3,2)
  hcc19=hcc19+Angle(19,3,2)
  dih4=dih4+Dihedral(4,1,2,3)
  dih5=dih5+Dihedral(5,2,1,4)
  dih6=dih6+Dihedral(6,1,2,3)
  dih7=dih7+Dihedral(7,1,2,3)
  if (Dihedral(7,1,2,3)>24) dih7=dih7-360.
  dih8=dih8+Dihedral(8,7,1,2)
  if (Dihedral(8,7,1,2)>148) dih8=dih8-360.
  dih9=dih9+Dihedral(9,7,1,2)
  if (Dihedral(9,7,1,2)<-92) dih9=dih9+360.
  dih10=dih10+Dihedral(10,7,1,2)
  if (Dihedral(10,7,1,2)>25) dih10=dih10-360.
  dih11=dih11+Dihedral(11,6,1,2)
  if (Dihedral(11,6,1,2)<-25) dih11=dih11+360.
```

```
  dih12=dih12+Dihedral(12,6,1,2)
  if (Dihedral(12,6,1,2)>92) dih12=dih12-360.
  dih13=dih13+Dihedral(13,6,1,2)
  if (Dihedral(13,6,1,2)<-148) dih13=dih13+360.
  dih14=dih14+Dihedral(14,5,2,1)
  if (Dihedral(14,5,2,1)<-25) dih14=dih14+360.
  dih15=dih15+Dihedral(15,5,2,1)
  if (Dihedral(15,5,2,1)>92) dih15=dih15-360
  dih16=dih16+Dihedral(16,5,2,1)
  if (Dihedral(16,5,2,1)<-148) dih16=dih16+360.
  dih17=dih17+Dihedral(17,3,2,1)
  if (Dihedral(17,3,2,1)>25) dih17=dih17-360.
  dih18=dih18+Dihedral(18,3,2,1)
if (Dihedral(18,3,2,1)>148) dih18=dih18-360.
  dih19=dih19+Dihedral(19,3,2,1)
  if (Dihedral(19,3,2,1)<-92) dih19=dih19+360.
#other important distances
  BrToC1=BrToC1+Distance(1,4)
  BrToC2=BrToC2+Distance(2,4)
  }
}
}

END {
#print "count",count
print "cc2",cc2/count
print "cc3",cc3/count
print "brc4",brc4/count
print "cc5",cc5/count
print "cc6",cc6/count
print "cc7",cc7/count
print "hc8",hc8/count
print "hc9",hc9/count
print "hc10",hc10/count
print "hc11",hc11/count
print "hc12",hc12/count
print "hc13",hc13/count
print "hc14",hc14/count
print "hc15",hc15/count
print "hc16",hc16/count
print "hc17",hc17/count
print "hc18",hc18/count
print "hc19",hc19/count
print "ccc3",ccc3/count
print "brcc4",brcc4/count
print "ccc5",ccc5/count
print "ccc6",ccc6/count
print "ccc7",ccc7/count
print "hcc8",hcc8/count
print "hcc9",hcc9/count
print "hcc10",hcc10/count
print "hcc11",hcc11/count
print "hcc12",hcc12/count
print "hcc13",hcc13/count
print "hcc14",hcc14/count
print "hcc15",hcc15/count
```

```
print "hcc16",hcc16/count
print "hcc17",hcc17/count
print "hcc18",hcc18/count
print "hcc19",hcc19/count
print "dih4",dih4/count
print "dih5",dih5/count
print "dih6",dih6/count
print "dih7",dih7/count
print "dih8",dih8/count
print "dih9",dih9/count
print "dih10",dih10/count
print "dih11",dih11/count
print "dih12",dih12/count
print "dih13",dih13/count
print "dih14",dih14/count
print "dih15",dih15/count
print "dih16",dih16/count
print "dih17",dih17/count
print "dih18",dih18/count
print "dih19",dih19/count
#test=Dihedral(4,1,2,3)
#test=Dihedral(1,2,3,4)
#test=Dihedral(1,2,5,19)
#other important distances
print ""
print "BrToC1",BrToC1/count
print "BrToC2",BrToC2/count
print "Count",count
}

function Distance(Atom1,Atom2) {
 return sqrt((x[Atom1]-x[Atom2])^2 + (y[Atom1]-y[Atom2])^2 + (z[Atom1]-z[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(Atom2,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) {
#  print at[Atom1],x[Atom1],y[Atom1],z[Atom1]
#  print at[Atom2],x[Atom2],y[Atom2],z[Atom2]
#  print at[Atom3],x[Atom3],y[Atom3],z[Atom3]
#  print at[Atom4],x[Atom4],y[Atom4],z[Atom4]
  B1x=x[Atom2]-x[Atom1]
  B1y=y[Atom2]-y[Atom1]
  B1z=z[Atom2]-z[Atom1]
  B2x=x[Atom3]-x[Atom2]
```

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

References:

S1. 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.

S2. Singleton, D. A.; Hang, C.; Szymanski, M. J.; Greenwald, E. E. *J. Am. Chem. Soc.* **2003**, 125, 1176-1177.

S3. Chapman, S.; Bunker, D.L. *J. Chem. Phys.* **1975,** *62,* 2890-2899.