Computer Simulations on Mahler's Conjecture in 3 dimensions

Adam Summerville

August 27, 2005

1 Introduction

In this paper I will discuss the findings of my computer program which dealt with simulating Mahler's conjecture. I will discuss the conjecture, the fundamental algorithms that are used in the program, and some interesting results of my program.

2 Overview of Mahler's conjecture

Given a body K that is centrally symmetric about the origin in $\mathbb{R}^n$ its dual is defined as

$K^* = \{y \in \mathbb{R}^n : (x, y) \leq 1 \forall x \in K\}$

where (x,y) is the standard inner product in $\mathbb{R}^n$. The volume product of K is defined as

$\mathrm{vol}_n(K) \cdot \mathrm{vol}_n(K^*)$

Mahler's conjecture states that for any centrally symmetric body K, $\frac{4^n}{n!} \leq \mathrm{vol}_n(K) \cdot \mathrm{vol}_n(K^*)$. This means that the smallest possible volume product for centrally symmetric bodies is found when the body is the n dimensional cube. If we have a polytope, say P,in $\mathbb{R}^n$ with faces $f_1, f_2, ..., f_k$, the dual polytope of P is defined as the convex hull having vertices $v_1, v_2, ..., v_k$, where $0\vec{v_i} \perp f_i$ and $\left\| 0\vec{v_i} \right\| = \frac{1}{h_i}$

where h_i is defined as the distance between the origin and f_i for $1 \leq i \leq k$. Because centrally symmetric convex polytopes are dense in centrally symmetric convex bodies , and the volume product is continuous it suffices to prove Mahler's conjecture only on polytopes.

3 The algorithms

In the program I developed, the initial input is a list of points in three dimensions. With this list of points, the program finds the convex hull containing these points using the QuickHull method via code developed by John Lloyd which can be found at http://www.cs.ubc.ca/spider/lloyd/java/quickhull3d.html. Once the convex hull, say K, is found, the dual is found by taking a face, say f_i, of K and finding the plane P_i containing that face. Once P_i is found the point of projection of the origin on the plane is found, say p_i. Given that the volume product is invariant upon the size of the original body and the fact that through a linear transformation a 'nice' body can be found, it is assumed that this point of projection will lie inside of f_i. A vector $0\vec{p}_i$ is then found and finally the vertex v_i of the dual body is found by

$$v_i = \frac{0\vec{p}_i}{\left\|0\vec{p}_i\right\|^2}$$

This process is then done for all other faces, and the convex hull of these points is found using the same QuickHull method.

Once K and K^* are found, their respective volumes must be found. This is done by taking a face of a body, say f_i, and finding the area of that face. If the face contains only three points p_1, p_2, p_3, then this is done by finding the distance between the points and then using Heron's method,

$$a = \text{dist}(p_1, p_2)$$

$$b = \text{dist}(p_1, p_2)$$

$$c = \text{dist}(p_1, p_2)$$

$$s = \frac{1}{2}(a + b + c)$$

$$\text{area}_i = \sqrt{s(s-a)(s-b)(s-c)}$$

If the face consists of more than three points then an inner point of the face is found by taking the midpoint, say m of two points that do not lie on the same edge, which for simplicity sake were taken to be p_1 and p_3. Then using Heron's method the area of all triangles consisting of two edge points and m were found and summed.

With the area of the face and the distance between the face and the origin, the volume of the pyramid that has the face as it's base and the origin as it's apex is found by using the formula of volume$_i = \frac{1}{3}\cdot\text{dist}(0, f_i)\cdot\text{area}_i$. This was then repeated for all of the faces and the total volume of the body was found by summing these, and the volume product was then trivially found.

4 Findings

Testing my program on the simplest case of three dimensional centrally symmetric convex bodies, i.e. the cube with edge length of 1, the expected volume product of $\frac{32}{3}$ was found with an error of 10^{-15} due only to truncation. Testing the program with a variety of cubes of differing edge lengths and with rectangular prisms of varying lengths, the volume product was invariant.

I then tested my program on octagonal prism and the volume product was found to be $\frac{112}{9}$ which is greater than $\frac{32}{3}$, so my program, while not proving Mahler's conjecture, at least did not contradict it. Interestingly, the volume product of a 'sliced' octahedron, i.e. a decahedron found by placing a face of four edges in place of a vertex for two opposite vertices, was found to also be $\frac{112}{9}$. This is odd because the octagonal prism has 16 vertices and 10 faces, whereas the sliced octahedron has 12 vertices and 10 faces. In two dimensions by removing vertices the area product is reduced, but the same does not hold true for three dimensions.

At first inspection it seemed that bodies with the same number of faces might have the same volume product, but upon further inspection it does not seem to hold true for all cases. An octahedron that has been 'sliced' twice on two pairs of opposite corners has a volume product of approximately 13.307563622661705 and has 12 faces. A decagonal prism that also has 12 faces has a volume product of only $\frac{151}{12}$, so there does not seem to be any hard and fast rule to the volume product as there is in the two dimensional case. In general, a centrally symmetric polytope with more faces and vertices will have a larger volume product, because the sphere has the largest possible volume product.

Mahler's conjecture requires that the body is centrally symmetric with respect to the origin, but interesting things happen when the body is not symmetric about the origin, but the dual is still taken with respect to the origin. If the body were a cube with vertices of $(1+a, 1, 1), (1+a, -1, 1), (-1+a, 1, 1), (-1+a, -1, 1), (1+a, 1, -1), (1+a, -1, -1), (-1+a, 1, -1), (-1+a, -1, -1)$, then the body would be the cube symmetric about the origin when $a = 0$ and will be a cube with one face lying in the y-z plane when $a = 1$. When the dual of the $a = 1$ case is found the face of the original lying in the y-z plane will be mapped to a face lying at $-\infty$ in the x direction. However, the volume of the original body will remain the same no matter how the body is translated, so the only

interesting part of the volume product here is the volume of the dual body. The volume of the dual body is an even function on a that monotonically increases to ∞ for $0 \leq a \leq 1$. If this shifting value of a is added to more than one component of the vertices, then the volume will increase at a greater rate. This is what common sense would predict, because as the number of components is increased then more faces are being mapped to ∞ in different dimensions. See the below table for specific results.

a	one component	two components	three components
0	1.33333333333333	1.33333333333333	1.33333333333333
0.01	1.33346668000133	1.33360004000533	1.3337334133466
0.02	1.33386688008536	1.3344006403415	1.33493461418717
0.03	1.3345344143062	1.33573657722571	1.33693982306647
0.04	1.33547008547008	1.3376102618891	1.33975386807803
0.05	1.33667502088554	1.34002508359453	1.34338354245066
0.06	1.33815067576609	1.34298542328993	1.34783763878957
0.07	1.33989883763775	1.34649667132726	1.35312699359588
0.08	1.3419216317767	1.35056524937268	1.35926454244433
0.09	1.34422152770776	1.35519863666475	1.36626538629373
0.1	1.34680134680134	1.36040540080944	1.37414686950448
0.11	1.34966427101258	1.36619523333595	1.38292867024592
0.12	1.35281385281385	1.37257899027379	1.39263290409273
0.13	1.35625402637914	1.37956873805222	1.40328424173758
0.14	1.35998912008704	1.38717780506633	1.41491004188733
0.15	1.36402387041773	1.39542083930202	1.42754050056473
0.16	1.36836343732895	1.40431387246403	1.44120881821021
0.17	1.37301342120619	1.41387439110924	1.45595138616955
0.18	1.37797988149373	1.4241214153511	1.47180799436865
0.19	1.38326935712556	1.43507558577193	1.488822062218
0.2	1.38888888888888	1.44675925925925	1.50704089506172
0.21	1.3948460438679	1.45919661457046	1.52651596879429
0.22	1.40114894213254	1.47241376852937	1.54730324561724
0.23	1.40780628585506	1.48643890386977	1.56946352430554
0.24	1.41482739105829	1.5013024098666	1.59306282880582
0.25	1.42222222222222	1.51703703703703	1.61817283950617
0.26	1.43000143000142	1.5336780673546	1.64487137210918
0.27	1.43817639233452	1.55126350160125	1.67324290971982
0.28	1.44675925925925	1.5698342656893	1.70337919454134
0.29	1.4557630017833	1.58943443802086	1.73537988647326
0.3	1.46520146520146	1.61011150022139	1.76935329694658
0.31	1.47508942729653	1.6319166138915	1.80541720753568
0.32	1.48544266191325	1.65490492637394	1.84369978428469
0.33	1.49627800845397	1.67913590893723	1.88434060031111
0.34	1.50761344791195	1.70467373124373	1.92749178114397
0.35	1.51946818613485	1.73158767650695	1.97331928946661
0.36	1.53186274509803	1.75995260236447	2.02200436852536
0.37	1.54481906306723	1.78984945321195	2.07374516650672
0.38	1.55836060464391	1.8213658305796	2.12875856776485
0.39	1.57251248181782	1.85459662910464	2.18728226100323
0.4	1.58730158730158	1.8896447467876	2.24957707950905
0.41	1.60275674159554	1.92662187954747	2.31592965446264
0.42	1.61890885543143	1.96564941164574	2.38665542939017
0.43	1.63579110947532	2.00685941537887	2.4621020922327
0.44	1.65343915343915	2.05039577559418	2.54265349156025
0.45	1.67189132706374	2.05039577559418	2.62873411552754
0.46	1.69118890580077	2.14508993632772	2.72081422669675
0.47	1.71137637444915	2.19660682126703	2.81941576340269
0.48	1.73250173250173	2.25117168984112	2.92511913960645
0.49	1.75461683554853	2.30901017969277	3.03857110105641

a	1 dimension	2 dimensions	3 dimensions
0.5	1.77777777777777	2.37037037037037	3.16049382716049
0.51	1.80204532143983	2.43552550539239	3.29169550667982
0.52	1.82748538011695	2.50477711090592	3.43308266297412
0.53	1.85416956380661	2.57845857851009	3.58567456335711
0.54	1.88217579521927	2.65693929308197	3.75062012010443
0.55	1.9115890083632	2.74062940267126	3.92921778160754
0.56	1.94250194250194	2.82998534746786	4.12293902603128
0.57	1.97501604700538	2.92551628944657	4.33345621307447
0.58	2.00924251557163	3.02779161478546	4.56267573053867
0.59	2.04530347190264	3.13744971913275	4.81277760259665
0.6	2.08333333333333	3.25520833333333	5.08626302083334
0.61	2.12348038434995	3.38187670703926	5.3860116372659
0.62	2.16590859865713	3.51837004330269	5.71535094753524
0.63	2.21079975681202	3.66572667354009	6.07814072880135
0.64	2.2583559168925	3.82512858552253	6.47887633049211
0.65	2.30880230880231	3.99792607584816	6.92281571575439
0.66	2.3623907394283	4.18566750430245	7.41613661286757
0.67	2.4194036170084	4.39013539649503	7.96613209307754
0.68	2.48015873015873	4.61339049508693	8.58145553401588
0.69	2.54501495196284	4.85782582928583	9.27242952717281
0.7	2.61437908496732	5.12623349993593	10.0514382351685
0.71	2.68871412247093	5.42188772428097	10.9334295710445
0.72	2.76854928017719	5.74864883757722	11.9365632009494
0.73	2.85449225718975	6.1110945347672	13.0830540243357
0.74	2.94724432655467	6.51468684030654	14.4002803720303
0.75	3.04761904761905	6.96598639455784	15.9222546161322
0.76	3.15656565656566	7.47293005815735	17.6915957816225
0.77	3.27519855891263	8.04519420022757	19.7622063380682
0.78	3.40483486550902	8.69467534603941	22.2029503218575
0.79	3.54704265318791	9.43613368765075	25.1027765034604
0.8	3.70370370370371	10.2880658436214	28.5779606767262
0.81	3.87709605505477	11.2739053650909	32.7825105120412
0.82	4.07000407000408	12.4236998473873	37.9233817075316
0.83	4.28586735240545	13.7764942218112	44.2831701118972
0.84	4.52898550724638	15.3837822936358	52.2546952908829
0.85	4.80480480480482	17.3146119092066	62.394997871015
0.86	5.12032770097288	19.6633168240126	75.5119693702486
0.87	5.48471136706432	22.5615440850035	92.8076679761562
0.88	5.91016548463359	26.1975420418157	116.123856568332
0.89	6.41333974667311	30.8481950296928	148.379966472789
0.9	7.01754385964916	36.9344413665747	194.391796666183
0.91	7.75644754702351	45.1218589122952	262.48899890806
0.92	8.68055555555562	56.5140335648155	367.929906020937
0.93	9.86923266716021	73.0513150789067	540.720318866819
0.94	11.454753722795	98.4085371374156	845.434167847222
0.95	13.6752136752138	140.258601797066	1438.54976202121
0.96	17.0068027210886	216.923504095522	2766.88142978986
0.97	22.5606316976879	381.736576949043	6459.16373856263
0.98	33.6700336700346	850.253375505955	21471.0448360096
0.99	67.0016750418803	3366.9183438133	169191.876573543

5 References

[1] Coxeter, H.S.M., *Regular Polytopes*, The Macmillan Company, New York, 1963

[2] Lloyd, John. "QuickHull3D: A Robust 3D Convex Hull Algorithm in Java." 16 May 2005. University of British Columbia. 18 July 2005 `<http://www.cs.ubc.ca/spider/lloyd/java/quickhull3d.html>`

[3] Dmitry Ryabogin, lecture, June 8,14,23, July 6,14,21,25 , 2005

[3] Dovgard, Roman, *Simulations on Mahler's and volume monotonicity conjectures*, Weizmann Institute of Science, May 10 2001