Stalking the Perfect Cuboid

Terry Raines
2114 Morning Road
Dyersburg, TN 38024

Part One: The Computer Search

A *cuboid* (or *Euler brick*) is a rectangular parallelepiped with positive integer edges x, y, z whose face diagonals $\sqrt{x^2+y^2}, \sqrt{x^2+z^2}, \sqrt{y^2+z^2}$ are all integers. If the body diagonal $\sqrt{x^2+y^2+z^2}$ is also an integer, the cuboid (x, y, z) is said to be *perfect*, but so far no example of a perfect cuboid has been found. Cuboids that are not perfect are sometimes called *body cuboids*; the smallest example is $(44, 117, 240)$ discovered by Paul Halcke and published in 1719. Our main interest will be *primitive* body cuboids: cuboids with $\gcd(x, y, z) = 1$.

For the past few years, serious computer searches for body cuboids have been based on the following method. Any primitive body cuboid has exactly one odd edge and, given any odd integer x there are only finitely many positive integers y and z such that $x^2 + y^2$ and x^2+z^2 are perfect squares. Whenever $\sqrt{y^2+z^2}$ is an integer, the computer has found a body cuboid.

For example, if $x = 85$ the only choices for y and z are 132, 204, 720, and 3612. These four integers are determined by the prime factors of 85 (for more details see Butler [**4**]). Since $132^2 + 720^2$ is a perfect square, $(85, 132, 720)$ is a body cuboid. The odd edge x must be completely factored; also when x is large there may be many thousands of choices for y and z and every possible pair (y, z) must be checked to determine whether $y^2 + z^2$ is a perfect square. Using this method "Durango Bill" Butler found all primitive cuboids with odd edge less than 3×10^{12} and in 2014 Robert Matson [**26**] extended the search to 25×10^{12}. This search method was first described by Maurice Kraïtchik [**16**] in Volume III of his *Théorie des Nombres* (1947).

Over the years many other computer searches for body cuboids have been undertaken using methods which do not require factorization into primes; this greatly simplifies the computer code. To me many of these methods seemed vaguely similar, and around 1995 I discovered the following simple method which seems to generalize most of the older search methods. Sooner or later it will find every primitive body cuboid exactly once. Let

$$q = a^2 - b^2\ ,\quad r = 2ab\ ,\quad s = a^2 + b^2\ ,\quad t = c^2 - d^2\ ,\quad u = 2cd\ ,\quad v = c^2 + d^2$$

so that $q^2 + r^2 = s^2$ and $t^2 + u^2 = v^2$. In each of the following four cases

Case	x	y	z
I	qu	ru	rt
II	qu	qt	rt
III	qt	rt	ru
IV	qt	qu	ru

$\sqrt{x^2+y^2}$ and $\sqrt{y^2+z^2}$ are always integers; for example in Case I we have

$$x^2+y^2=(qu)^2+(ru)^2=(q^2+r^2)u^2=(su)^2$$

and

$$y^2+z^2=(ru)^2+(rt)^2=r^2(u^2+t^2)=(rv)^2.$$

The other three cases are similar. Thus (x,y,z) is a body cuboid whenever x^2+z^2 happens to be a perfect square, and so if $g=\gcd(x,y,z)$ then $(x/g,y/g,z/g)$ is a primitive body cuboid.

If the right triangles (q,r,s) and (t,u,v) are not primitive, we get quite a lot of duplicate primitive cuboids. To avoid this we require that $\gcd(a,b)=\gcd(c,d)=1$; also $a+b$ and $c+d$ must both be odd. In a computer search we may assume $a>b>0$ and $a\geq c>d>0$, so a is the dominant loop. By symmetry if $a=c$ we may assume that $d<b$. (The conditions $a=c$ and $b=d$ need not be checked: they imply $q=t$ and $r=u$. In Cases I and II, $x^2+z^2=(qu)^2+(rt)^2=2q^2r^2$ cannot be a perfect square. In Cases III and IV, $x^2+z^2=(qt)^2+(ru)^2=q^4+r^4$ and this also cannot be a perfect square.)

Euler observed that if (x,y,z) is a body cuboid then so is (xy,xz,yz), which he called the *derived cuboid* of (x,y,z). I prefer to call these *duals* since the dual of the dual is $(xy\cdot xz, xy\cdot yz, xz\cdot yz)=xyz\cdot(x,y,z)$ which reduces to the same primitive as (x,y,z). Note that Cases I and II are duals, as are Cases III and IV. For this reason the search always finds body cuboids in pairs — each time a body cuboid appears, you get the dual cuboid for free. The table

Case	(x,y)	(y,z)
I	$(a^2-b^2,2ab)u$	$(2cd,c^2-d^2)r$
II	$(2cd,c^2-d^2)q$	$(a^2-b^2,2ab)t$
III	$(a^2-b^2),2ab)t$	$(c^2-d^2,2cd)r$
IV	$(c^2-d^2,2cd)q$	$(a^2-b^2,2ab)u$

shows that the faces (x,y) and (y,z) always reduce to the Pythagorean triangles generated by either (a,b) or (c,d). When (x,y,z) is a body cuboid, the third face (x,z) reduces to a Pythagorean triangle generated by a third pair (e,f). When this happens, it is easy compute the generators e and f: let $h=\gcd(x,z)$ and set $k=x/h$, $l=y/h$, and $m=\sqrt{k^2+l^2}$. Note that since (x,y,z) is a body cuboid, m is always an integer. Without loss of generality we may assume k is odd and l is even. To find e and f such that $e^2-f^2=k$ and $e^2+f^2=m$ we merely note that $2e^2=m+k$, $2f^2=m-k$, and hence

$$(e,f)=(\sqrt{(m+k)/2},\sqrt{(m-k)/2})\ .$$

Then of course $l=\sqrt{m^2-k^2}=\sqrt{(e^2+f^2)^2-(e^2-f^2)^2}=\sqrt{4e^2f^2}=2ef$. Note that dual body cuboids always have the same generators (a,b,c,d,e,f).

The reason for computing the generators e and f is that they decide whether the body cuboid (x,y,z) is new: if $e>a$, or if $e=a$ and $f>b$, then (x,y,z) cannot have appeared earlier in our search and so must be new; otherwise the cuboid and its dual must be discarded.

Referees have asked why my method finds *all* primitive body cuboids. Let (x, y, z) be any primitive body cuboid; compute the generators (a, b, c, d, e, f), and you will know exactly when the search will find (x, y, z).

Many of the older search formulas could not find all primitive cuboids — for example, not all primitive body cuboids can be found with the Case I formulas alone. This was apparently not considered a problem in those more innocent times: after all, a perfect cuboid was expected to appear after a few dozen body cuboids were discovered.

A few years ago I bought eighteen factory refurbished Compaq computers for \$100 each (they stack neatly and are all hooked to the same keyboard and monitor) and in a little over two years they checked all $a \leq 7000$ running nonstop — it is amazing that sixteen of them still work! I stopped the project about two years ago because my little army of computers was averaging one new pair of body cuboids per day. A very good day might produce two or three pairs, followed by several days with nothing. Obviously run time is $O(a^4)$ so reaching $a = 10000$ would have taken several more years assuming no more of these aging computers failed. For $a \leq 7000$ my search found 36830 different primitive body cuboids, and 3725 of these had odd edge larger than 10^{13}.

The story would have ended there, but at a Christmas dinner in 2014 my nephew asked me "How are your cuboids coming along?" I replied that, sadly, my computers had been idle for a long time, and that I had no plans to continue the search. But in January I discovered Tim Roberts' website *unsolvedproblems.org* and sent him a short email: was he interested in an unpublished search method for body cuboids? The same day he replied Yes, but he was on vacation — send him my computer code, and he would take a look. Thanks to his vacation, I had time to prepare a detailed description of my search method, including the following Ubasic code.

```
 10   cls :  print "BODYSAVE" :  open "BXXXX" for output as #1
 20   input "First A";FA : input "Last A";LA
 30   for A=FA to LA : for B=1 to A-1
 40   if or{gcd(A,B)>1,even(A+B)} then 140
 50   Q=A^2-B^2 :  R=2*A*B
 60   for C=2 to A : if C<A then LD=C-1 else LD=B-1
 70   for D=1 to LD : if or{gcd(C,D)>1, even(C+D)} then 130
 80   T=C^2-D^2 :  U=2*C*D
 90   X=Q*U : Z=R*T : XZ=isqrt(X^2+Z^2) :  if res>0 then 110
100   Y=R*U : TT=1 :  gosub 160 :  Y=Q*T : TT=2 :  gosub 160
110   X=Q*T : Z=R*U : XZ=isqrt(X^2+Z^2) :  if res>0 then 130
120   Y=R*T : TT=3 :  gosub 160 :  Y=Q*U : TT=4 :  gosub 160
130   next D : next C
140   next B : next A : end
150   '
160   GG=gcd(X,gcd(Y,Z)) : XX=X/GG : ZZ=Z/GG : gosub 210
170   if FLAG=0 then return else YY=Y/GG
180   inc CT : print CT,A,B,C,D,E,F,GG,TT,XX,YY,ZZ
290   print #1,A,B,C,D,E,F,GG,TT,XX,YY,ZZ : return
200   '
210   G=gcd(XX,ZZ) : K=XX/G : L=ZZ/G : KK=K : if odd(L) then K=L : L=KK
220   M=isqrt(K^2+L^2) :  E=isqrt((M+K)/2) :  F=isqrt((M-K)/2)
230   FLAG=0 :  if or{E>A, and{E=A,F>B}} then FLAG=1
240   return
```

A week later I was astonished to learn that Tim had translated my Ubasic code into Python, and had found the 280 primitive body cuboids with $a \leq 100$ in a matter of seconds. The next step, he said, would be to translate the code into C++ so that a search could begin on the brand new Linux Cluster at the University of Queensland in Australia. For the next two days we exchanged a flurry of emails. He noticed that the b- and d-loops were always either even or odd, so they could be performed twice as fast. Also C++ had a function which returned 'true' when $x^2 + z^2$ was a perfect square, and was much faster than the ordinary square root.

The search on the Linux Cluster began on January 27, and in just two days it completed $a \leq 7000$, a computation that took my eighteen computers over two years! In fact, the Cluster found 36854 primitive body cuboids in this range, which means I had missed twelve dual pairs. I was not at all surprised by this: I had used a $10 flash drive to transfer data from my eighteen computers to a master data file, and some human error was inevitable. Also there were several power outages during that two-year period.

The Cluster search ended on March 23 at $a = 15000$. At this point each node was taking up to ten hours to locate a new pair of cuboids. (Of course with 100 nodes running, the Cluster was still averaging more than ten new pairs every hour.) A total of 75868 primitive body cuboids were found. None were perfect.

Part Two: Analysis of Data

All the data accumulated in the course of this project was stored in fifteen files named $\mathcal{D}_1$ to $\mathcal{D}_{15}$. Specifically, $\mathcal{D}_1$ contained information on all cuboids generated by $a \leq 1000$, $\mathcal{D}_2$ by $1001 \leq a \leq 2000$, and so forth. Here our first objective will be to estimate the number of cuboids in $\mathcal{D}_k$ when k is larger than fifteen.

For $k = 1, \cdots, 15$ let $C(k)$ be the number of primitive body cuboids contained in $\mathcal{D}_k$ and for $k = 1, \cdots, 14$ let $D(k) = C(k+1) - C(k)$ and $R(k) = C(k+1)/C(k)$. The table

k	$C(k)$	$D(k)$	$R(k)$	k	$C(k)$	$D(k)$	$R(k)$
1	4874	608	1.1247	9	5092	-208	0.9592
2	5482	8	1.0015	10	4884	22	1.0045
3	5490	-76	0.9862	11	4906	-4	0.9992
4	5414	-112	0.9793	12	4902	-208	0.9576
5	5302	-212	0.9600	13	4694	-6	0.9987
6	5090	112	1.0220	14	4688	130	1.0277
7	5202	-172	0.9669	15	4818		
8	5030	62	1.1023				

shows that the numbers $C(k)$ appear to be decreasing (though rather irregularly) while the differences $D(k)$ show this irregularity may be hard to predict. On the other hand, the ratios $R(k)$ seem to rise and fall only slightly above and below unity. The values $C(1)$, $D(1)$, and $R(1)$ appear to be statistical outliers, so we ignore them and let

$$R_1 = \sqrt[13]{R(2) \cdots R(14)} = 0.99012$$

$$R_2 = \frac{R(2) + \cdots + R(14)}{13} = 0.99039$$

be the geometric and arithmetic means of the thirteen ratios $R(2), \cdots, R(14)$. Then if $C_1(k) = 4818 \times R_1^{k-15}$ and $C_2(k) = 4818 \times R_2^{k-15}$ we have

k	$C_1(k)$	$C_2(k)$	k	$C_1(k)$	$C_2(k)$
16	4770	4772	50	3403	3436
17	4723	4726	100	2071	2120
18	4677	4680	200	767	807
19	4630	4635	500	39	45
20	4585	4591	1000	0	0

and while the values for $k = 16, 17, 18, 19, 20$ are not unreasonable, the values for larger k seem much too small. Indeed $C_1(1000) = C_2(1000) = 0$ implies there are only finitely many primitive body cuboids, and we know that is not true!

If you plot the fifteen points $(1, C(1)), \cdots, (15, C(15))$ on a sheet of graph paper, it is even more obvious that $(1, (C(1))$ is an outlier while the other fourteen points roughly follow a descending curve. Moreover this curve appears to be concave upward, whereas the two exponential curves in the preceding table are concave downward. So instead we consider the two curves $y = a + b \log k$ and $y = c + d/ \log k$ — both are decreasing and concave upward when $b < 0$ and $d > 0$. Note that the second curve is not defined for $k = 1$, another reason for treating the point $(1, C(1))$ as an outlier.

It is easy to find formulas for a, b, c, d. If $y_i = a + b \log i$ and $y_j = c + d/ \log j$ then

$$b = \frac{y_i - y_j}{\log i - \log j} \ , \quad a = y_i - b \log i = y_j - b \log j$$

and similarly

$$d = \frac{\log i \log j}{\log j - \log i}(y_i - y_j) \ , \quad c = y_i - d/ \log i = y_j - d/ \log j \ .$$

Since we may assume $2 \leq i < j \leq 15$ this gives $\binom{13}{2} = 91$ choices for (a, b) and for (c, d). By taking the arithmetic means of those 182 choices for a, b, c, d and then adjusting both curves to equal $C(k) = 4818$ when $k = 15$, we obtain the two curves $C_3(k) = 6063.66 - 459.98 \log k$ and $C_4(k) = 4123.57 + 1880.55/ \log k$.

k	C_3	C_4	$C_4 - C_3$	k	C_3	C_4	$C_4 - C_3$
16	4788	4802	14	50	4264	4604	340
17	4760	4787	27	100	3945	4532	587
18	4734	4774	40	200	3627	4479	852
19	4709	4762	53	500	3205	4426	1221
20	4686	4751	65	1000	2886	4396	1510

While this is a definite improvement over the C_1 and C_2 values, C_3 and C_4 are still growing further and further apart. This is no surprise, since $C_3(k) \to -\infty$ and $C_4(k) \to 4123.57$ as $k \to \infty$. To fix this, we introduce our final estimation, the weighted average

$$C_5(k) = \frac{C_3 + C_4 \log k}{1 + \log k} = \frac{7944.21 + 3663.59 \log k}{1 + \log k}$$

which is closer to $C_4(k)$ than to $C_3(k)$ when k is large. Note that $C_5(15) = C(15) = 4818$.

k	$C_5(k)$	k	$C_5(k)$	k	$C_5(k)$
16	4798	50	4535	10^4	4006
17	4780	100	4427	10^5	3953
18	4764	200	4257	10^6	3914
19	4749	500	4205	10^7	3884
20	4735	1000	4083	10^8	3861

Finding the 75868 primitive body cuboids up to $k = 15$ took almost two months on the Australian supercomputer, and since run time is $O(k^4)$ it may seem rather bizarre to attempt to estimate $C(k)$ for $k = 10^8$. However, to obtain formulas which approximate the number of primitive body cuboids with body diagonals less than 10^{25} we will require estimates of the form $y = (A + B \log k)/(1 + \log k)$ when k is a number in the billions. Note that if we take the first and second derivatives of y with respect to k we see that if $A < B$ then the curve y is increasing and concave down, and that if $A > B$ then y is decreasing and concave up.

Greatest Common Divisor Counts

The $k = 1, \cdots, 15$ are the same as above, but the d notation requires some explanation. The column $d = 0$ contains the number of body cuboids (x, y, z) in $\mathcal{D}_k$ with $\gcd(x, y, z) = 1$ before they were reduced to primitive; that is, they were already primitive and needed no reduction. The columns for $d > 0$ contain the counts of those cuboids in $\mathcal{D}_k$ for which the greatest common denominators of the edges before reduction to primitive were greater than one and had exactly d digits. Thus $d = 1$ is the number with $2 \leq \gcd \leq 9$, $d = 2$ is the number with $10 \leq \gcd \leq 99$, and so on.

	$d = 0$	1	2	3	4	5	6	7	8	Total
$k = 1$	796	996	1739	1027	300	16	0	0	0	4874
2	791	859	1646	1400	683	98	5	0	0	5482
3	761	777	1513	1383	860	178	18	0	0	5490
4	707	704	1386	1455	902	241	17	0	0	5414
5	705	686	1311	1391	862	310	36	1	0	5302
6	655	674	1208	1321	792	392	46	2	0	5090
7	645	616	1214	1338	905	420	60	4	0	5202
8	647	567	1123	1306	893	421	70	3	0	5030
9	610	593	1115	1358	897	443	73	3	0	5092
10	612	565	1033	1198	920	463	87	5	1	4884
11	578	588	1079	1227	860	487	82	5	0	4906
12	614	565	990	1255	860	512	102	4	0	4902
13	606	545	939	1134	908	462	94	6	0	4694
14	587	505	937	1124	891	511	119	14	0	4688
15	595	548	972	1140	924	496	131	12	0	4818
Total	9909	9988	18207	19047	12457	5450	940	59	1	75868

A somewhat unexpected result is the total number of cuboids for $d = 0$. It is well known that the probability that three random positive integers x, y, z are relatively prime (that is, $\gcd(x, y, z) = 1$) is $1/\zeta(3) = 0.8319074\cdots$ where

$$\zeta(3) = 1 + 1/3^2 + 1/3^3 + 1/3^4 + \cdots = 1.2020569\cdots$$

is Apéry's constant. However the preceding table gives $9909/75868 = 0.1306084\cdots$ so we must conclude there is something decidedly "nonrandom" about body cuboids.

Our next goal is to estimate the counts in the columns when $k > 15$ and again we use the weighted logarithmic curve

$$C(k) = \frac{(a + b \log k) + (c + d/\log k) \log k}{1 + \log k} = \frac{(a + d) + (b + c) \log k}{1 + \log k}$$

where a, b, c, d have the same formulas as before. Using $I = 1$ to 14, $J = I + 1$ to 15, $K = 2$ to 15, and averaging the ten to twenty curves which best fit the data, the following eight estimation formulas were found.

	a	b	c	d
$d = 0$	861	-97.9	475	299
1	991	-164	393	418
2	1972	-369	631	922
3	1608	-173	1059	221
4	823	37.1	996	-194
5	-81.5	213.1	695	-536
6	-52.6	67.8	174.5	-118.1
7	-1.28	4.75	14.51	-7.38

Note that no estimate for $d = 8$ is given because of insufficient data. Note also that for $d = 0, 1, 2, 3$ the curves are decreasing and concave upward, but for $d = 4, 5, 6, 7$ they are increasing and concave downward. Here are some values obtained from these estimation formulas.

	$d = 0$	1	2	3	4	5	6	7	Total	$C(k)$
$k = 15$	595	548	972	1140	924	496	131	12	4818	4818
16	592	542	960	1136	926	503	136	12	4804	4802
17	589	537	949	1132	928	510	135	12	4792	4788
18	586	533	939	1128	929	515	136	12	4778	4775
19	583	529	930	1125	931	521	138	12	4769	4763
20	580	525	920	1122	932	526	139	12	4756	4751
50	544	470	798	1078	951	597	158	14	4610	4592
100	525	440	732	1054	961	635	169	14	4610	4506
200	510	417	680	1036	969	665	177	15	4469	4439
500	494	393	627	1017	977	696	185	15	4404	4370
1000	485	379	595	1005	982	715	189	16	4366	4329
10000	463	345	520	979	994	758	202	17	4278	4231
100000	449	324	473	962	1001	786	209	17	4221	4170
1000000	439	309	440	950	1006	805	214	17	4180	4127
10000000	432	299	416	941	1010	818	218	18	4152	4096
100000000	427	290	398	935	1012	829	221	18	4130	4073

Of course the cuboid count $C(k)$ and the Total agree for $k = 15$ since the a, b, c, d were chosen so that they would. The differences in the right two columns are respectively 0, 2, 4, 3, 6, 5, 18, 24, 30, 34, 37, 47, 51, 53, 56, and 57, an error at most 1.38%, a surprisingly good fit.

More Approximation Formulas

Our ultimate goal is to obtain formulas which estimate the number of body diagonals, odd edges, and least edges for primitive body cuboids contained in $\mathcal{D}_k$ where $k > 15$. The methods are similar for these three estimates so we begin with body diagonals, perhaps the most important, since this is the condition that prevents a body cuboid from being perfect. The Australian supercomputer obtained the following counts for $k \leq 15$. All body diagonals were rounded to the nearest integer, and none were found with more than 18 digits.

Body Diagonal Counts with d Digits

$d =$	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
$\mathcal{D}_1$	5	13	42	159	432	958	1324	1137	600	200	4	0	0	0	0
$\mathcal{D}_2$	0	0	0	0	5	112	746	1489	1549	1050	506	25	0	0	0
$\mathcal{D}_3$	0	0	0	0	0	19	271	1083	1520	1291	841	461	4	0	0
$\mathcal{D}_4$	0	0	0	0	0	8	110	686	1281	1467	1031	681	150	0	0
$\mathcal{D}_5$	0	0	0	0	0	2	57	397	1148	1319	1215	742	422	0	0
$\mathcal{D}_6$	0	0	0	0	0	0	34	266	950	1225	1186	786	596	47	0
$\mathcal{D}_7$	0	0	0	0	0	0	18	188	821	1276	1210	892	628	169	0
$\mathcal{D}_8$	0	0	0	0	0	1	13	127	622	1181	1224	921	607	334	0
$\mathcal{D}_9$	0	0	0	0	0	1	13	92	553	1107	1257	1005	606	458	0
$\mathcal{D}_{10}$	0	0	0	0	0	0	7	92	417	1039	1148	955	678	522	26
$\mathcal{D}_{11}$	0	0	0	0	0	0	7	45	381	952	1115	1066	694	568	78
$\mathcal{D}_{12}$	0	0	0	0	0	0	0	60	321	876	1134	1070	731	558	152
$\mathcal{D}_{13}$	0	0	0	0	0	0	1	30	254	774	1115	973	744	570	233
$\mathcal{D}_{14}$	0	0	0	0	0	0	2	29	205	722	1010	1109	743	543	325
$\mathcal{D}_{15}$	0	0	0	0	0	0	0	34	189	647	1071	1092	826	557	402

It is evident that all cuboids with body diagonal less than 10^9 have been found, and virtually all with body diagonal less than 10^{10}. On the other hand it appears that the computer is not close to finding all body diagonals less than 10^{11} or 10^{12}.

Each row in this table suggests a bell that is thick in the middle and tapers off on both sides, although the right tails appear to be chopped off for $k > 3$. If these are indeed bells (we will soon see they might perhaps be called "logarithmic" or "geometric" bells, though this terminology does not seem to be standard) it would be most useful to know their means and standard deviations. Let the primitive body cuboids have edges x, y, z and let $\delta = \log_{10} \sqrt{x^2 + y^2 + z^2}$. Let n_k be the number of cuboids in $\mathcal{D}_k$ and set

$$\mu_k = \frac{1}{n_k} \sum \delta \quad \text{and} \quad \sigma_k^2 = \frac{1}{n_k} \sum (\mu_k - \delta)^2$$

where the sums are taken over the cuboids (x, y, z) in $\mathcal{D}_k$ for $k = 1, 2, \cdots, 15$. Then

k	μ_k	σ_k	k	μ_k	σ_k	k	μ_k	σ_k
1	8.5855	1.4416	6	12.1201	1.4371	11	13.0297	1.5426
2	10.2778	1.2212	7	12.3278	1.4543	12	13.1409	1.5477
3	10.9947	1.3018	8	12.5457	1.4677	13	13.2901	1.5544
4	11.4678	1.3452	9	12.7046	1.4891	14	13.4067	1.5647
5	11.8405	1.3801	10	12.8792	1.5318	15	13.5044	1.5833

and we see that the graphs of both μ_k and σ_k are increasing and (with minor exceptions) concave down. These seem to describe reasonably good bells: for example, for $k = 1$ the proportions of cuboids in $\mathcal{D}_1$ with the base 10 logarithms of their body diagonals within $\mu_1 \pm 1\sigma_1$, $\mu_1 \pm 2\sigma_1$, and $\mu_1 \pm 3\sigma_1$ are respectively 0.6859, 0.9590, and 0.9951 while for a normal Gaussian bell it is well known that these proportions are 0.6827, 0.9545, and 0.9973. Results for k between 2 and 15 are similar.

The values μ and σ are called *geometric* means and *geometric* standard deviations. Another way to view these results is that in each block $\mathcal{D}_k$ about 68% of body diagonals lie between $10^{\mu-\sigma}$ and $10^{\mu+\sigma}$, about 95% lie between $10^{\mu-2\sigma}$ and $10^{\mu+2\sigma}$, and more than 99% lie between $10^{\mu-3\sigma}$ and $10^{\mu+3\sigma}$.

In order to extend the values μ_k and σ_k to $k > 15$ it was natural to fit them to a weighted logarithmic curve of the form

$$C(k) = \frac{(a+d)+(b+c)\log k}{1+\log k}$$

where a, b, c, d are constants. Unfortunately this curve did not fit the data very well: in particular the curve $c + d/\log k$ was an exceptionally poor fit. Recall that the curve $c + d/\log k$ was introduced because the block count values $C(k)$ were decreasing and the curve $a + b\log k$ converged to $-\infty$ which was unacceptable. On the other hand, μ_k and σ_k are both increasing and curves of the form $a + b\log k$ seem to fit both of them quite well. The approximations

$$\mu_k \approx 9.1475 + 1.6089\log k \quad \text{and} \quad \sigma_k \approx 1.1786 + 0.1494\log k$$

are good to within one percent for $k = 2$ to 14 (as usual, $k = 1$ is a bit of an outlier) and the constants were chosen so that the fit is exact for $k = 15$.

We now have the tools to estimate the number S_d of primitive body cuboids whose body diagonals lie between 10^{d-1} and 10^d. Consider the following code:

for $d = 1, 2, 3, \cdots$: for $k = 1, 2, 3, \cdots$
  compute $C(k)$, μ_k , σ_k , and $z_k = (d - \mu_k)/\sigma_k$
  compute $S_d = \sum_k p(z_k)C(k)$ where $p(z_k) = \exp(-z_k^2)/\sqrt{2\pi}$
  when $p(z_k)C(k)$ gets small, exit the k-loop
next k : next d

Remark 1. For $k \le 15$ we use the exact values of $C(k)$, μ_k , and σ_k and for $k > 15$ we use the approximating formulas.

Remark 2. The function $p(z)$ is the well-known Gaussian bell curve, which has the property $\int_{-\infty}^{\infty} p(z)dz = 1$. Since $\int_{-4}^{4} p(z)dz = 0.9999366658$ and $C(k)$ is usually between 4000 and 5000, we "chop off the tails" by defining $p(z_k) = 0$ when $|z_k| > 4$. This greatly accelerates the program by shortening the k-loops.

Remark 3. Most elementary statistics textbooks use tables to approximate

$$P(x) = \int_{-\infty}^{x} p(z)\,dz = \frac{1}{2} + \frac{1}{\sqrt{2\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{2^n(2n+1)n!} \ .$$

The infinite sum converges very quickly, but the program runs about three times faster using $p(z) = \exp(-z^2)/\sqrt{2\pi}$ instead. Since

$$p(z) = \lim_{\epsilon \to 0} \frac{\epsilon}{2} \left(P(z+\epsilon) - P(z-\epsilon) \right)$$

the two methods yield equivalent results.

A one-day Ubasic run on one of my older computers produced the following table:

$d =$	$k =$	k ratios	S_d	S_d ratios
3	2		1	
4	2	1.000000000	12	12.00000
5	2	1.000000000	88	7.33333
6	4	2.000000000	395	4.48864
7	13	3.250000000	1166	2.95190
8	35	2.692307692	2615	2.24271
9	92	2.628571429	5664	2.16597
10	246	2.673913043	12838	2.26660
11	661	2.686991870	28385	2.21101
12	1777	2.688350983	60825	2.14286
13	4776	2.687675858	129766	2.13343
14	12836	2.687604690	279511	2.15396
15	34504	2.688064818	609404	2.18025
16	92750	2.688094134	1343443	2.20452
17	249318	2.688064690	2991748	2.22693
18	670184	2.688019042	6727002	2.24852
19	1801506	2.688076707	15268918	2.26980
20	4842587	2.688077087	34980169	2.29094
21	13107249	2.688077468	80873859	2.31199
22	34991373	2.688077412	188671085	2.33291
23	94059522	2.688077487	444053392	2.35358
24	252839286	2.688077513	1054140261	2.37390
25	679651601	2.688077521	2523327223	2.39373

It must emphasized that these are merely *approximations* and that the S_d ratios are much more important than the S_d values themselves. Evidently the k ratio column is converging, while the S_d ratios are not. The limit $2.688077\cdots$ is apparently determined by the fact that all bells were chopped off at ± 4 standard deviations. Similar programs for ± 3 and ± 5 standard deviations produced k-ratio columns converging to the limits $2.366\cdots$ and $2.848\cdots$ respectively. The wider bells gave larger k ratio limits, larger k values, and made the programs run more slowly. The choice of ± 4 standard deviations was somewhat arbitrary, but seems to be a reasonable middle-of-the-road decision. The S_d ratios varied only about two percent for tail chops at ± 3, ± 4, and ± 5 standard deviations, so they were relatively unaffected. For a long time I believed a formula for S_d would probably be exponential — that is, the S_d ratios would eventually converge, but clearly this is not happening. The right-hand column is increasing quite steadily for $d > 16$. For $17 \le d \le 25$ the usual curve-fitting methods found the three approximating curves $R_1 = 0.98863 + 0.43652 \log d$, $R_2 = 3.64850 - 4.03896/\log d$, and $R_3 = (R_1 + R_2 \log d)/(1 + \log d)$ with the following error values.

d	$R_1(d)$	$R_2(d)$	$R_1(d) - R(d)$	$R_2(d) - R(d)$	$R_3(d) - R(d)$
17	2.22538	2.22293	-0.00155	-0.00400	-0.00336
18	2.25033	2.25112	0.00181	0.00260	0.00240
19	2.27393	2.27678	0.00413	0.00698	0.00626
20	2.29632	2.30027	0.00538	0.00933	0.00834
21	2.31762	2.32187	0.00563	0.00988	0.00883
22	2.33793	2.34184	0.00502	0.00893	0.00797
23	2.35733	2.36036	0.00375	0.00678	0.00605
24	2.37591	2.37761	0.00201	0.00321	0.00331
25	2.39373	2.39373	0.00000	0.00000	0.00000

The approximation $R_1 = 0.98863 + 0.43652 \log d$ is the winner, by a nose. Since the Australian supercomputer found that S_9 is 1101 (that is, there are 1101 primitive body cuboids whose body diagonals are between 10^9 and 10^{10}) it is now easy to predict S_d for larger d. Remember, these predictions may not agree precisely with some tables above — values for small d are especially erratic.

d	S_d	d	S_d	d	S_d
9	1101	16	1.77×10^5	40	2.82×10^{14}
10	2144	17	3.89×10^5	50	4.71×10^{18}
11	4276	18	8.65×10^5	60	1.10×10^{23}
12	8702	19	1.95×10^6	70	3.33×10^{27}
13	18043	20	4.43×10^6	80	1.26×10^{32}
14	38039	25	3.08×10^8	90	5.78×10^{36}
15	81428	30	2.60×10^{10}	100	3.13×10^{41}

For example, the table on the previous page gives $S_9 = 5664$. Now the approximations R_1 were constructed so that $R_1(25) = R(25)$ but in general $R_1(d) \neq R(d)$ when $d < 25$ and indeed

$$\prod_{d=6}^{24} \frac{R(d)}{R_1(d)} = 6.190 \ , \quad \prod_{d=7}^{24} \frac{R(d)}{R_1(d)} = 2.733 \ , \quad \text{and} \quad \prod_{d=8}^{24} \frac{R(d)}{R_1(d)} = 1.702$$

so disagreements involving small values of d are inevitable.

Odd Edge Counts and Least Edge Counts

The data files $\mathcal{D}_1$ to $\mathcal{D}_{15}$ produced means and standard deviations similar to those for body diagonals. For odd edges the approximation formulas were $\mu_k = 8.6415 + 1.6052 \log k$, $\sigma_k = 1.2456 + 0.1523 \log k$, and $R_1(d) = 0.9373 + 0.4730 \log d$. For least edges we have $\mu_k = 8.2929 + 1.5912 \log k$, $\sigma_k = 1.2490 + 0.1524 \log k$, and $R_1(d) = 0.9272 + 0.4514 \log d$.

There are 749 primitive body cuboids whose odd edges contain exactly eight decimal digits; that is, the odd edge count for $d = 8$ is $S_8 = 749$. For least edges, $S_8 = 1037$. Methods like those above for body diagonals yield the next two tables.

Odd Edge Count Predictions

d	S_d	d	S_d	d	S_d
8	749	15	1.185×10^5	30	5.544×10^{10}
9	1439	16	2.628×10^5	40	8.136×10^{14}
10	2844	17	5.910×10^5	50	1.879×10^{19}
11	5762	18	1.346×10^6	60	6.138×10^{23}
12	11935	19	3.102×10^6	70	2.649×10^{28}
13	25214	20	7.227×10^6	80	1.442×10^{33}
14	54221	25	5.731×10^8	100	7.558×10^{42}

Least Edge Count Predictions

d	S_d	d	S_d	d	S_d
8	1037	15	1.837×10^5	30	1.147×10^{11}
9	2021	16	4.149×10^5	40	2.074×10^{15}
10	4056	17	9.499×10^5	50	5.948×10^{19}
11	8350	18	2.203×10^6	60	2.426×10^{24}
12	17580	19	5.172×10^6	70	1.314×10^{29}
13	37767	20	1.228×10^7	80	9.002×10^{33}
14	82617	25	1.072×10^9	100	7.542×10^{43}

Prognostications on the Existence of a Perfect Cuboid

Let $P(d)$ be the probability that a random positive integer with d decimal digits is a perfect square. If $d = 1$ then of the nine numbers 1, 2, $\cdots$, 9 only 1, 4, and 9 are perfect squares, so $P(1) = 3/9$. If $d = 2$ then of the ninety numbers 10, 11, $\cdots$, 99 only 16, 25, 36, 49, 64, and 81 are perfect squares, so $P(2) = 6/90$. Similarly $P(3) = 22/900$, $P(4) = 68/9000$, $P(5) = 217/90000$, and $P(6) = 683/900000$. In general if $d = 2n$ then

$$P(d) = \frac{10^n - 1 - \lfloor 10^{2n}/\sqrt{10} \rfloor}{9 \cdot 10^{2n-1}}$$

and if $d = 2n + 1$ then

$$P(d) = \frac{\lfloor 10^{2n}\sqrt{10} \rfloor - 10^n + 1}{9 \cdot 10^{2n}} .$$

Computer searches have shown there are no perfect cuboids for $d < 9$. Now $P(9) = 21623/900000000 = 0.0000240256$ so the expected number of perfect primitive cuboids for $d = 9$ is $P(9) \cdot 1101 = 0.0264521 \approx$ one in 38. Similarly we have

d	one in	d	one in	d	one in
9	38	17	1071	25	13494
10	61	18	1522	26	17827
11	97	19	2138	27	23383
12	151	20	2974	28	30463
13	231	21	4095	29	39428
14	346	22	5588	30	50715
15	511	23	7558	31	64842
16	795	24	10139	32	82427

and we can thus estimate the probability that a perfect primitive cuboid (x, y, x) exists with $\sqrt{x^2 + y^2 + z^2} > 10^d$. For example, $1/61 + 1/97 + 1/151 + 1/231 + \cdots \approx 1/21$.

d	one in	d	one in	d	one in
10	21	17	295	24	2333
11	32	18	408	25	3031
12	48	19	559	26	3910
13	71	20	757	27	5009
14	104	21	1016	28	6375
15	149	22	1351	29	8062
16	211	23	1783	30	10133

I was genuinely surprised that these numbers were so small, since $P(d)$ quickly becomes almost zero. The search on the Queensland Cluster found only a few thousand cuboids with $d > 16$, the tiny tip of an astronomical iceberg, and yet there is one chance in 211 that a perfect cuboid might be out there somewhere. Would you fly on Buzzard Airlines if you knew their crash rate was one in 211? So . . . Does a perfect cuboid exist? Probably not. Does this mean that a perfect cuboid cannot exist? Certainly not.

Almost Perfect Cuboids

We conclude with a list of the fifteen cuboids from $\mathcal{D}_1$ to $\mathcal{D}_{15}$ which came closest to perfection, in the sense that their body diagonals were nearly an integer.

k	x	y	z	body error
1	1244484	37835	269280	+ 0.000064764
2	70735208725	583316952720	204903304452	+ 0.000302216
3	16868453745	4379546392	1438005600	− 0.000140602
4	10136008575	4061165768	10894334400	− 0.000025451
5	1677076347375	9891972998336	1156250722800	− 0.000149049
6	2040233435175	853181299080	119179456352	+ 0.000198299
7	163427634589	1461174115120	152788458867660	+ 0.000033865
8	880818229090008	501968724800505	222764235512000	+ 0.000338285
9	7318225497285	6453586104108	381446833845	− 0.000099753
10	205212110499	18596334900	408279173200	− 0.000062970
11	11650397410036	3013275535173	5898062395680	− 0.000126528
12	273884042835	927729434016	561798887188	− 0.000003598
13	1794451687126560	655747029795	733147471004	− 0.000001235
14	9583407188532	8317379716800	1410957090725	− 0.000111462
15	2555561080800	2654119727259	679218272612	− 0.000024763

The grand prize winner is the cuboid in $\mathcal{D}_{13}$ which has generators

$$(a, b, c, d, e, f) = (12238, 5473, 2737, 2736, 14979924, 14973804)\ .$$

The reciprocal of the body error is $1/0.000001235 \approx 809717$, more than ten times larger than would expect since there are only 75868 primitive body cuboids with $a \leq 15000$. Note the unusually small winner in $\mathcal{D}_1$; it has $1/0.000064764 \approx 15441$ but it was only the 578th body cuboid found in the search.

Bibliography

[1] Ayoub B. Ayoub: Integral solutions to the equation $x^2 + y^2 + z^2 = u^2$: a geometrical approach, *Mathematics Magazine*, **57** (1984) 222–223.

[2] Raymond A. Beauregard & E. R. Suryanarayan: Pythagorean Boxes, *Mathematics Magazine*, **74** (2001) 222–226.

[3] Andrew Bremner: The rational cuboid and a quartic surface, *Rocky Mountain Journal of Mathematics*, **18** (1988) 105–121.

[4] Bill Butler: The "integer brick" problem (The Euler brick problem). Look up www.durangobill.com.

[5] E. Z. Chein: On the derived cuboid of an Eulerian triple, *Canadian Mathematical Bulletin*, **20** (1977) 509–510.

[6] Yong–Gao Chen & Shu-Guang Guo: On the rational cuboids with a given face, *Journal of Number Theory* **112** (2005) 205–215.

[7] W. J. A. Colman: On certain semi-perfect cuboids, *Fibonacci Quarterly*, **26** (1988) 54–57.

[8] W. J. A. Colman: Some observations on the classical cuboid and its parameter solutions, *Fibonacci Quarterly*, **26** (1988) 338–343.

[9] W. J. A. Colman: A perfect cuboid in Gaussian integers, *Fibonacci Quarterly*, **32** (1994) 266–268.

[10] Leonard Eugene Dickson: *History of the Theory of Numbers*, Volume II.

[11] Leonhard Euler: *Elements of Algebra* (Fifth Edition, 1840) 443–447.

[12] Richard K. Guy: *Unsolved Problems in Number Theory* (Second Edition, 1994) 173–181.

[13] Ivan Korec: Nonexistence of a small perfect rational cuboid, I, *Acta Mathematica University Comenian*, **42/43** (1983) 73–86.

[14] Ivan Korec: Nonexistence of a small perfect rational cuboid, II, *Acta Mathematica University Comenian*, **44/45** (1984) 39–48.

[15] Ivan Korec: Lower bounds for perfect rational cuboids, *Mathematica Slovaca*, **42** (1992) 565–582.

[16] Maurice Kraïtchik: *Théorie des Nombres*, Volumes I, II, III. (1922, 1926, 1947)

[17] Maurice Kraïtchik: On certain rational cuboids, *Scripta Mathematica*, **11** (1945) 317–326.

[18] Jean Lagrange: Sets of n squares of which any $n - 1$ have their sum square, *Mathematics of Computation*, **41** (1983) 675–681.

[19] M. Lal & W. J. Blundon: Solutions of the Diophantine equations $x^2 + y^2 = l^2$, $y^2 + z^2 = m^2$, $z^2 + x^2 = n^2$, *Mathematics of Computation*, **20** (1966) 144–177.

[20] L. J. Lander, T. R. Parkin, & J. L. Selfridge: A survey of equal sums of like powers, *Mathematics of Computation*, **21** (1967) 336–459.

[21] John Leech: The rational cuboid revisited, *American Mathematical Monthly*, **84** (1977) 518–533.

[22] John Leech: Five tables related to rational cuboids, *Mathematics of Computation*, **32** (1978) 657–659.

[23] John Leech: A remark on rational cuboids, *Canadian Mathematics Bulletin*, **24(3)** (1981) 377–378.

[24] Florian Luca: Perfect cuboids and perfect square triangles, *Mathematics Magazine*, **73** (2000) 400–401.

[25] Allan J. MacLeod: Parametric expressions for a "nearly-perfect" cuboid. PCTex document at http://maths.paisley.ac.uk.

[26] Robert D. Matson: Results of computer search for a perfect cuboid, *unsolvedproblems.org* (January 18, 2015)

[27] H. C. Pocklington: Some diophantine impossibilities, *Proceedings of the Cambridge Philosophical Society*, **17** (1914) 110–121.

[28] Randall L. Rathbun: (1) Table of equal area Pythagorean triangles, form coprime sets of integer generator pairs. (2) The integer cuboid table, with body, edge, and face type of solutions + The integer cuboid auxiliary table (with Torbjörn Granlund). (3) The classical rational cuboid table of Maurice Kraitchik (also with Granlund). Reviewed in *Mathematics of Computation*, **62** (1994) 440–443.

[29] Randall L. Rathbun: Computer searches for the perfect integer cuboid (1999). Article and report at www.math.niu.edu.

[30] W. G. Spohn: On the integral cuboid, *American Mathematical Monthly*, **79** (1972) 57–59.

[31] W. G. Spohn: On the derived cuboid, *Canadian Mathematical Bulletin*, **17** (1974) 575–577.