A FAST MODULO PRIMES ALGORITHM FOR SEARCHING PERFECT CUBOIDS AND ITS IMPLEMENTATION.

R. R. Gallyamov, I. R. Kadyrov, D. D. Kashelevskiy,
N. G. Kutlugallyamov, R. A. Sharipov

Abstract. A perfect cuboid is a rectangular parallelepiped whose all linear extents are given by integer numbers, i. e. its edges, its face diagonals, and its space diagonal are of integer lengths. None of perfect cuboids is known thus far. Their non-existence is also not proved. This is an old unsolved mathematical problem.

Three mathematical propositions have been recently associated with the cuboid problem. They are known as three cuboid conjectures. These three conjectures specify three special subcases in the search for perfect cuboids. The case of the second conjecture is associated with solutions of a tenth degree Diophantine equation. In the present paper a fast algorithm for searching solutions of this Diophantine equation using modulo primes seive is suggested and its implementation on 32-bit Windows platform with Intel-compatible processors is presented.

1. Introduction.

Conjecture 1.1 (Second cuboid conjecture). *For any two positive coprime integer numbers $p \neq q$ the tenth-degree polynomial*

$$\begin{aligned}Q_{pq}(t) = t^{10} + (2\,q^2 + p^2)\,(3\,q^2 - 2\,p^2)\,t^8 + (q^8 + 10\,p^2\,q^6 +\\
+ 4\,p^4\,q^4 - 14\,p^6\,q^2 + p^8)\,t^6 - p^2\,q^2\,(q^8 - 14\,p^2\,q^6 + 4\,p^4\,q^4 +\\
+ 10\,p^6\,q^2 + p^8)\,t^4 - p^6\,q^6\,(q^2 + 2\,p^2)\,(3\,p^2 - 2\,q^2)\,t^2 - q^{10}\,p^{10}\end{aligned} \tag{1.1}$$

is irreducible over the ring of integers $\mathbb{Z}$.

Theorem 1.1. *A perfect cuboid associated with the polynomial (1.1) does exist if and only if for some positive coprime integer numbers $p \neq q$ the Diophantine equation $Q_{pq}(t) = 0$ has a positive solution t obeying the inequalities*

$$t > p^2, \qquad t > p\,q, \qquad t > q^2, \qquad (p^2 + t)\,(p\,q + t) > 2\,t^2.$$

Theorem 1.1 can be found in [1]. It stems from the results of [2] and [3]. As for the perfect cuboid problem itself, it has a long history reflected in [4–51]. There are also two series of ArXiv publications. The first of them [52–54] continues the research on cuboid conjectures. The second one [55–67] relates perfect cuboids with multisymmetric polynomials.

2000 *Mathematics Subject Classification.* 11D41, 11D72, 68U99, 65-04.

The scope of perfect cuboids in the case of the second cuboid conjecture is restricted by the following theorem derived from [1].

Theorem 1.2. *In the case of the second cuboid conjecture there are no perfect cuboids outside the region given by the inequalities*

$$\min\left(\sqrt[3]{\frac{p}{9}},\,\frac{p}{59}\right)\leqslant q\leqslant 59\,p. \tag{1.2}$$

In [1] the region given by the inequalities (1.2) was presented as a union of two regions which were called the linear and the nonlinear regions respectively. In this paper we present an algorithm for searching cuboids in the region (1.2).

2. A modulo primes seive.

Let p, q, and t be a triple of integer numbers satisfying the Diophantine equation $Q_{pq}(t)=0$ with the polynomial (1.1) and let r be some prime number. Then we can pass from $\mathbb{Z}$ to the quotient ring $\mathbb{Z}_r=\mathbb{Z}/r\mathbb{Z}$ and denote

$$\tilde p=p\bmod r,\qquad\qquad \tilde q=q\bmod r,\qquad\qquad \tilde t=t\bmod r. \tag{2.1}$$

The numbers $\tilde p$, $\tilde q$, and $\tilde t$ are interpreted as division remainders after dividing p, q, and t by the prime number r. They obey the quotient equation

$$Q_{\tilde p\tilde q}(\tilde t)\bmod r=0. \tag{2.2}$$

Once r is given there are only a finite number of remainders (2.1):

$$\tilde p=0,\,\ldots\,,r-1,\qquad \tilde q=0,\,\ldots\,,r-1,\qquad \tilde t=0,\,\ldots\,,r-1.$$

The values in the left hand side of the equation (2.2) for them can be precomputed. They can be either zero or nonzero modulo r. We can use them as a fast computed test for sweeping away those values of p, q, and t, where $Q_{pq}(t)\neq 0$.

Definition 1.1. A pair of integer numbers $0\leqslant\tilde p\leqslant r-1$ and $0\leqslant\tilde q\leqslant r-1$ is called solvable modulo r if there is at least one integer number $0\leqslant\tilde t\leqslant r-1$ such that $Q_{\tilde p\tilde q}(\tilde t)\bmod r=0$. Otherwise it is called unsolvable.

We can represent solvable and unsolvable pairs in the form of bit-arrays u_r:

$$u_r(\tilde p,\tilde q)=\begin{cases}0 & \text{if } (\tilde p,\tilde q)\text{ is solvable;}\\ 1 & \text{if } (\tilde p,\tilde q)\text{ is unsolvable.}\end{cases} \tag{2.3}$$

The value $u_r(\tilde p,\tilde q)$ of the function (2.3) is called the unsolvability bit. Bit-arrays of the form (2.3) can be stored as tables. For $r=2$ this table looks like

$u_2(\tilde p,\tilde q)$	$\tilde q=0$	$\tilde q=1$
p=0	0	0
p=1	0	0

(2.4)

As we see in (2.4), the values of the function $u_2(\tilde{p},\tilde{q})$ are identically zero. The same is true for the functions $u_3(\tilde{p},\tilde{q})$ $u_5(\tilde{p},\tilde{q})$, and $u_7(\tilde{p},\tilde{q})$ associated with the prime numbers $r=3$, $r=5$, and $r=7$. The case of $r=11$ is different:

u_{11}	0	1	2	3	4	5	6	7	8	9	10
p=0	0	0	0	0	0	0	0	0	0	0	0
p=1	0	0	1	1	1	1	1	1	1	1	0
p=2	0	1	0	1	1	1	1	1	1	0	1
p=3	0	1	1	0	1	1	1	1	0	1	1
p=4	0	1	1	1	0	1	1	0	1	1	1
p=5	0	1	1	1	1	0	0	1	1	1	1
p=6	0	1	1	1	1	0	0	1	1	1	1
p=7	0	1	1	1	0	1	1	0	1	1	1
p=8	0	1	1	0	1	1	1	1	0	1	1
p=9	0	1	0	1	1	1	1	1	1	0	1
p=10	0	0	1	1	1	1	1	1	1	1	0

(2.5)

In the memory of a computer bit-arrays like (2.5) are packed into byte-arrays with 8 bits per 1 byte, e. g. the array u_{11} looks like

00000000	11100000	10011111 ... 11111111	00000000

(2.6)

Note that bits in a byte are written in the reverse order — the highest bit is the leftmost. This is because bytes are designed to represent binary numbers. Note also that the last byte of the table (2.5) in (2.6) is incomplete. It is appended with zero bits which are shown in blue.

Bytes associated with the prime number $r=11$ can be written into some linear locus of memory. Similarly, bytes associated with several other prime numbers can be written into adjacent loci. Altogether they constitute a bit seive. Accessing a proper bit of this seive, we can easily decide whether for a certain pair of integer numbers p and q the equation $Q_{pq}(t)=0$ is unsolvable modulo some prime number r enclosed in the seive. Then it is unsolvable in the ring of integers $\mathbb{Z}$ as well. Quickening the search algorithm is reached through sweeping away those (p,q) pairs that do not go through the bit seive for several prime numbers. Indeed, it is clear that calculating the remainders

$$\tilde{p}=p \bmod r, \qquad\qquad \tilde{q}=q \bmod r.$$

and then addressing bits in a memory locus are much faster operations than factoring a polynomial with numeric coefficients.

In our particular case we use the bit seive for 96 consecutive prime numbers from 11 to 541. This bit seive is stored in the binary file `Cuboid_pq_bit_tables.bin`.

In order to access effectively bit-tables for each particular prime number from 11 to 541 one should know their offsets within this file. These offsets are written to the separate binary file Cuboid_primes.bin. They are enclosed in the structures described as follows in C++ language:

```
struct primes_item
{
short prime;           // prime number
unsigned int p_offset; // prime bit-table offset
};
```

In our implementation the values of prime numbers are restricted not only by `short=2 bytes` data format used for them. Each prime number r is associated with the $r \times r$ bit-table that occupies $r^2/8$ bytes in memory. Using `unsigned int=4 bytes` format for offsets, we have the following restriction:

$$\sum_{i=5}^{N} \frac{r_i^2}{8} < 2^{32}, \text{ where } r_5 = 11,\ r_6 = 13,\ \ldots\ . \tag{2.7}$$

From (2.7) we derive $N < 1198$ and $r_N < 9697$. These inequalities fit the 4 Gb RAM (random access memory) limit. Actually we have chosen $N = 100$ in which case 1.5 Mb RAM is sufficient.

3. Code for generating binary files.

The code for preparing Cuboid_pq_bit_tables.bin and Cuboid_primes.bin binary files is implemented as a DLL library interacting with a Maple code. The DLL file Cuboid_search_v01.dll is generated within the 32-bit x86 makefile project for Microsoft Visual C++ 2005 Express Edition package. The project files

1) make.bat
2) makefile
3) Cuboid_search_v01.h
4) Cuboid_search_v01.cpp

are suppled as ancillary files to this paper. The C++ file Cuboid_search_v01.cpp is the main source file of the project. It comprises a C++ and inline assembly language code for running on 32-bit Windows machines with Intel compatible processors. The DLL library file Cuboid_search_v01.dll is produced from this code by running the batch file make.bat in a command prompt window:

```
>make.bat
```

Along with Cuboid_search_v01.dll several other files are generated, including two LOG files compiler.log and linker.log. They can be removed by running the same batch file with the `clean` option:

```
>make.bat clean
```

Note that the Visual C++ 2005 Express Edition package should be installed for successfully running the above files. In our implementation it was installed on Windows XP machine with Intel Pentium 4 Prescott CPU 2.80 GHz.

The generated DLL library Cuboid_search_v01.dll exports several functions. Their declarations are in the C++ header file Cuboid_search_v01.h. Three of these functions are declared as follows:

```
extern "C" __declspec(dllexport)
                         void __stdcall Open_pq_file_stream();
extern "C" __declspec(dllexport)
   unsigned int __stdcall Write_pq_file_stream(unsigned int rrr);
extern "C" __declspec(dllexport)
                         void __stdcall Close_pq_file_stream();
```

These declarations correspond to the following Maple worksheet declarations:

```
> DLL_file:="./Cuboid_search_v01.dll":
> Open_pq_file_stream:=define_external('Open_pq_file_stream',
    LIB=DLL_file):
> Write_pq_file_stream:=define_external('Write_pq_file_stream',
    'rrr'::(integer[4]), RETURN::(integer[4]), LIB=DLL_file):
> Close_pq_file_stream:=define_external('Close_pq_file_stream',
    LIB=DLL_file):
```

Maple worksheets are supplied in XML format as ancillary files to this paper. They can be imported to Maple. Here is the list of these files:

5) `Create_binary_seive_files.xml`
6) `Test_external_DLL_procedures_01.xml`
7) `Test_external_DLL_procedures_02.xml`
8) `Search_for_cuboids.xml`

The external function `Write_pq_file_stream(r)` imported to the Maple worksheet creates the bit-seive table for a given prime number r in its argument and writes it to the binary file `Cuboid_pq_bit_tables.bin`. It returns the integer value equal to the number of bytes written to the file `Cuboid_pq_bit_tables.bin`. The other binary file `Cuboid_primes.bin` is written simultaneously using the Maple worksheet code in `Create_binary_seive_files.xml`.

The external function `Write_pq_file_stream(r)` exploits another external function `Calculate_Q_pq_mod_prime(p,q,t,r)`. This function returns the value of the polynomial (1.1) modulo prime number r taken as its fourth argument. Though its arguments are declared as 32-bit integers, its code is designed to deal with 16-bit unsigned integers only. Due to the restriction $r < 9697$ derived from (2.7) its usage in `Write_pq_file_stream(r)` does not require 32-bit integers in its arguments:

$$0 \leqslant p \leqslant r-1 < 9797, \qquad\qquad 0 \leqslant q \leqslant r-1 < 9797.$$

The function `Calculate_Q_pq_mod_prime(p,q,t,r)` is a delicate part of the project. It is written in assembly language. Therefore it is carefully tested in the Maple worksheet code file `Test_external_DLL_procedures_01.xml`.

4. Code for loading and unloading binary files.

Once the binary files `Cuboid_pq_bit_tables.bin` and `Cuboid_primes.bin` are generated, they should be used in searching for perfect cuboids. For this purpose they should be loaded into the memory easily accessible from the DLL library functions. This task is performed by the function `Load_Cuboid_Binaries()` residing within the same DLL library. The opposite task is to unload the binary files, i. e.

to release the memory occupied by them. This task is performed by the function `Release_Cuboid_Binaries()` also residing within the DLL library.

The loading and unloading functions are imported and tested within the Maple worksheet code file `Test_external_DLL_procedures_02.xml`.

As an auxiliary test for two generated binary files `Cuboid_pq_bit_tables.bin` and `Cuboid_primes.bin` we visualize the bit-tables from `Cuboid_pq_bitmaps.bin` in the form of the text file `Cuboid_bit_tables.txt`. This text file is written by the code from the Maple worksheet file `Test_external_DLL_procedures_02.xml`.

5. Code for searching cuboids.

According to Theorem 1.2 the search for cuboids in the case of the second cuboid conjecture consists in scanning the region given by the inequalities (1.2). For each positive p these inequalities specify a finite segment of the real axis, which comprises a finite number of integer points. For $p \leqslant 151$ this segment is given by

$$\frac{p}{59} \leqslant q \leqslant 59\,p. \tag{5.1}$$

For $p \geqslant 152$ the inequalities are different:

$$\sqrt[3]{\frac{p}{9}} \leqslant q \leqslant 59\,p. \tag{5.2}$$

The inequalities $0 < p \leqslant 151$ and the inequalities (5.1) outline a finite set of integer points on the coordinate pq-plane. One can easily verify that these points do not produce perfect cuboids. For this reason the software in the DLL library `Cuboid_search_v01.dll` is designed to search cuboids for $p \geqslant 152$ within each segment specified by the inequalities (5.2). Roughly speaking, it is an infinite loop on $p \geqslant 152$ and an enclosed loop on q obeying the inequalities (5.2) for each p.

Both loops on p and on q are started from within the Maple worksheet file `Search_for_cuboids.txt` by executing the commands

```
> Load_Cuboid_Binaries();
> Start_searching(152,3);
```

Here 152 and 3 are initial values for p and q respectively. They should obey the inequalities (5.2). The function `Start_searching` is an external function imported from the DLL library `Cuboid_search_v01.dll`. It starts the looping process and returns just immediately with the value `0` indicating that the search is successfully started. The multithreading mechanism is used in the code of this function:

```
_beginthread(Look_for_cuboids_thread,0,(void*)12);
return(0);
```

Here `Look_for_cuboids_thread` is an internal function which is not exported from the DLL library. It is executed within a new thread, while the initial function `Start_searching` returns control to the Maple worksheet.

You can do anything in the Maple worksheet while the search function `Look_for_cuboids_thread` is running its infinite loops on p and q, provided you do not stop the Maple session by closing the worksheet. In particular, you can control the process of searching by executing the following function in the Maple worksheet:

```
> Get_current_p();
```

This function returns the current value of the loop variable p with the discreteness equal to 100. There is another control function:

```
> Get_current_r_max();
```

This function returns the maximal prime number is used to seive cuboids within current hundred values of p. The number $r_{\max}$ is flushed to 1 for each next hundred values of p and then is recalculated again.

The infinite loop on p cannot ever terminate by itself. Therefore it is terminated manually. This can be done at any time by executing the following function in the Maple worksheet that initiated the thread with this loop:

```
> Stop_searching();
```

Upon doing it you can read the time stamp and the exit values of p, q, and $r_{\max}$ at the end of the file `Cuboid_search_report.txt`, e. g. it could be

```
2016-2-20 21:36
Stop with p=112618, q=5691455, r_max=131
```

Then you can restart the search from this point on by executing the command

```
> Start_searching(112618,5691455);
```

Or you can terminate the session by executing the command

```
> Release_Cuboid_Binaries();
```

and then closing the Maple worksheet.

Normally the function `Start_searching(p,q)` returns 0 indicating that the search is successfully started. It returns 1 if the search is already running. So you cannot initiate several search threads running simultaneously with this software. This limitation is planned to be removed in further versions of the DLL library `Cuboid_search_v01.dll`.

The function `Start_searching(p,q)` returns 2 if $p < 152$ (see (5.1) and (5.2) above for explanation). This function returns 3 if $p > 72796055$, which breaks the 32-bit limit for $q = 59\,p$ in (5.2).

The function `Start_searching(p,q)` returns 4 if q is below the lower limit set by the inequalities (5.2). Similarly, it returns 5 if q is above the upper limit set by the inequalities (5.2).

The function `Start_searching(p,q)` returns 6 if it is invoked before the binary files `Cuboid_pq_bit_tables.bin` and `Cuboid_primes.bin` are loaded into the memory by the function `Load_Cuboid_Binaries()`.

The function `Load_Cuboid_Binaries()` normally returns the number of bytes loaded from the file `Cuboid_primes.bin`, i. e. the size of this file. However, if it is invoked when the binary files `Cuboid_pq_bit_tables.bin` and `Cuboid_primes.bin` are already loaded, it does not load them again and returns 0.

The function `Release_Cuboid_Binaries()` normally returns 0. However it returns 6 if the binary files `Cuboid_pq_bit_tables.bin` and `Cuboid_primes.bin` are not loaded. This function cannot release the memory occupied by the bit-tables if the search is running. In this case it returns 1.

The function `Stop_serching()` normally returns 0. However, if this function is invoked when the search is not on, it returns 1. Thus the functions

```
> Load_Cuboid_Binaries();
> Start_searching(p,q);
> Stop_serching();
> Release_Cuboid_Binaries();
```

should be invoked in the above order. Otherwise they signal misuse, but do not lead to a crash. These four functions constitute a toolkit we used in the present numerical research of perfect cuboids.

6. Results.

At present date 01.04.2016 the values of p from 1 to 154000 are scanned. For each such p all values of q limited by the inequalities (1.2) are scanned. This stands for about 700 billions (p, q) pairs that have been tested. Indeed, we have

$$N \approx \sum_{p=1}^{154000} 59\,p = 699626543000 \approx 0.7 \cdot 10^{12}.$$

None of these (p, q) pairs produces a perfect cuboid. Moreover, none of them goes through our primes seive composed by 96 consecutive primes from 11 to 541. Actually this seive is so dense that the maximal depth reached thus far is 29th prime in our sequence, which is equal to 137. This result is negative in the sense of finding a perfect cuboid. However it shows that the Second cuboid conjecture 1.1 is rather firm for to believe that it might be valid.

The total time spent for the above computations is 4130 minutes, i. e. about 69 hours. Then we can calculate the time per one (p, q) pair:

$$\Delta\,t = \frac{4130}{699626543000}\,min \approx 3.54 \cdot 10^{-7}\,sec.$$

The upper limit for p with our 32-bit code is given by the formula

$$p_{\,\max} = \frac{2^{\,32}}{59} \approx 72796055.$$

Here is the estimate for the time needed to reach this limit:

$$t = \Delta\,t \sum_{p=1}^{72796055} 59\,p = 5.53 \cdot 10^{10}\,sec \approx 1755\,years.$$

This estimate means that our code should be improved not only at the expense of multithreading and multiprocessing. Some fresh theoretical ideas are required.

7. Acknowledgement.

The authors are grateful to A. A. Gubarev for helpful advices on linking C++ code with a Maple worksheet session.

References

1. Sharipov R. A., *Asymptotic approach to the perfect cuboid problem,* **7,** (2015), no. 1, 100–112; see also e-prints arXiv:1504.07161, arXiv:1505.00724, arXiv:1505.02745, and arXiv:1506.04705 in Electronic Archive http://arXiv.org.
2. Sharipov R. A., *A note on a perfect Euler cuboid.*, e-print arXiv:1104.1716 in Electronic Archive http://arXiv.org.
3. Sharipov R. A., *Perfect cuboids and irreducible polynomials*, Ufa Mathematical Journal **4,** (2012), no. 1, 153–160; see also e-print arXiv:1108.5348 in Electronic Archive http://arXiv.org.
4. *Euler brick*, Wikipedia, Wikimedia Foundation Inc., San Francisco, USA.
5. Halcke P., *Deliciae mathematicae oder mathematisches Sinnen-Confect*, N. Sauer, Hamburg, Germany, 1719.
6. Saunderson N., *Elements of algebra,* Vol. 2, Cambridge Univ. Press, Cambridge, 1740.
7. Euler L., *Vollständige Anleitung zur Algebra,* 3 Theile, Kaiserliche Akademie der Wissenschaften, St. Petersburg, 1770-1771.
8. Pocklington H. C., *Some Diophantine impossibilities*, Proc. Cambridge Phil. Soc. **17** (1912), 108–121.
9. Dickson L. E, *History of the theory of numbers,* Vol. 2*: Diophantine analysis*, Dover, New York, 2005.
10. Kraitchik M., *On certain rational cuboids*, Scripta Math. **11** (1945), 317–326.
11. Kraitchik M., *Théorie des Nombres,* Tome 3*, Analyse Diophantine et application aux cuboides rationelles*, Gauthier-Villars, Paris, 1947.
12. Kraitchik M., *Sur les cuboides rationelles*, Proc. Int. Congr. Math. **2** (1954), Amsterdam, 33–34.
13. Bromhead T. B., *On square sums of squares*, Math. Gazette **44** (1960), no. 349, 219–220.
14. Lal M., Blundon W. J., *Solutions of the Diophantine equations* $x^2 + y^2 = l^2$, $y^2 + z^2 = m^2$, $z^2 + x^2 = n^2$, Math. Comp. **20** (1966), 144–147.
15. Spohn W. G., *On the integral cuboid*, Amer. Math. Monthly **79** (1972), no. 1, 57-59.
16. Spohn W. G., *On the derived cuboid*, Canad. Math. Bull. **17** (1974), no. 4, 575-577.
17. Chein E. Z., *On the derived cuboid of an Eulerian triple*, Canad. Math. Bull. **20** (1977), no. 4, 509–510.
18. Leech J., *The rational cuboid revisited*, Amer. Math. Monthly **84** (1977), no. 7, 518–533; see also Erratum, Amer. Math. Monthly **85** (1978), 472.
19. Leech J., *Five tables relating to rational cuboids*, Math. Comp. **32** (1978), 657–659.
20. Spohn W. G., *Table of integral cuboids and their generators*, Math. Comp. **33** (1979), 428–429.
21. Lagrange J., *Sur le dérivé du cuboide Eulérien*, Canad. Math. Bull. **22** (1979), no. 2, 239–241.
22. Leech J., *A remark on rational cuboids*, Canad. Math. Bull. **24** (1981), no. 3, 377–378.
23. Korec I., *Nonexistence of small perfect rational cuboid*, Acta Math. Univ. Comen. **42/43** (1983), 73–86.
24. Korec I., *Nonexistence of small perfect rational cuboid II*, Acta Math. Univ. Comen. **44/45** (1984), 39–48.
25. Wells D. G., *The Penguin dictionary of curious and interesting numbers*, Penguin publishers, London, 1986.
26. Bremner A., Guy R. K., *A dozen difficult Diophantine dilemmas*, Amer. Math. Monthly **95** (1988), no. 1, 31–36.
27. Bremner A., *The rational cuboid and a quartic surface*, Rocky Mountain J. Math. **18** (1988), no. 1, 105–121.
28. Colman W. J. A., *On certain semiperfect cuboids*, Fibonacci Quart. **26** (1988), no. 1, 54–57; see also *Some observations on the classical cuboid and its parametric solutions*, Fibonacci Quart. **26** (1988), no. 4, 338–343.
29. Korec I., *Lower bounds for perfect rational cuboids*, Math. Slovaca **42** (1992), no. 5, 565–582.
30. Guy R. K., *Is there a perfect cuboid? Four squares whose sums in pairs are square. Four squares whose differences are square*, Unsolved Problems in Number Theory, 2nd ed., Springer-Verlag, New York, 1994, pp. 173–181.
31. Rathbun R. L., Granlund T., *The integer cuboid table with body, edge, and face type of solutions*, Math. Comp. **62** (1994), 441–442.
32. Rathbun R. L., Granlund T., *The classical rational cuboid table of Maurice Kraitchik*, Math. Comp. **62** (1994), 442–443.

33. Peterson B. E., Jordan J. H., *Integer hexahedra equivalent to perfect boxes*, Amer. Math. Monthly **102** (1995), no. 1, 41–45.
34. Van Luijk R., *On perfect cuboids,* Doctoraalscriptie, Mathematisch Instituut, Universiteit Utrecht, Utrecht, 2000.
35. Luca F., *Perfect cuboids and perfect square triangles*, Math. Magazine **73**, no. 5, 400–401; see http://www.maa.org/sites/default/files/269082206010.pdf.bannered.pdf.
36. Rathbun R. L., *The rational cuboid table of Maurice Kraitchik*, e-print math.HO/0111229 in Electronic Archive http://arXiv.org.
37. Narumiya N., Shiga H., *On Certain Rational Cuboid Problems*, Nihonkai Math. Journal **12** (2001), no. 1, 75–88.
38. Hartshorne R., Van Luijk R., *Non-Euclidean Pythagorean triples, a problem of Euler, and rational points on K3 surfaces*, e-print math.NT/0606700 in Electronic Archive http://arXiv.org.
39. Waldschmidt M., *Open diophantine problems*, e-print math.NT/0312440 in Electronic Archive http://arXiv.org.
40. Ionascu E. J., Luca F., Stanica P., *Heron triangles with two fixed sides*, e-print math.NT/0608185 in Electronic Archive http://arXiv.org.
41. Ortan A., Quenneville-Belair V., *Euler's brick*, Delta Epsilon, McGill Undergraduate Mathematics Journal **1** (2006), 30-33.
42. Knill O., *Hunting for Perfect Euler Bricks*, Harvard College Math. Review **2** (2008), no. 2, 102; see also http://www.math.harvard.edu/~knill/various/eulercuboid/index.html.
43. Sloan N. J. A., *Sequences A031173, A031174, and A031175*, On-line encyclopedia of integer sequences, OEIS Foundation Inc., Portland, USA.
44. Roberts T. S., *Some constraints on the existence of a perfect cuboid*, Australian mathematical society gazette **37** (2010), no. 1, 29–31; see http://www.austms.org.au/Publ/Gazette/2010/Mar10/TechPaperRoberts.pdf.
45. Stoll M., Testa D., *The surface parametrizing cuboids*, e-print arXiv:1009.0388 in Electronic Archive http://arXiv.org.
46. Meskhishvili M., *Perfect cuboid and congruent number equation solutions*, e-print arXiv:1211.6548 in Electronic Archive http://arXiv.org.
47. Beauville A., *A tale of two surfaces*, e-print arXiv:1303.1910 in http://arXiv.org.
48. Freitag E., Manni R. S., *Parametrization of the box variety by theta functions*, e-print arXiv:1303.6495 in Electronic Archive http://arXiv.org.
49. Meskhishvili M., *Parametric solutions for a nearly-perfect cuboid*, e-print arXiv:1502.02375 in Electronic Archive http://arXiv.org.
50. Kitchen S., *On the existence of perfect cuboids*, OURE publication, Missouri University of Science and Technology, 2015.
51. Wyss W., *On perfect cuboids*, e-print arXiv:1506.02215 in Electronic Archive http://arXiv.org.
52. Sharipov R. A., *A note on the first cuboid conjecture*, e-print arXiv:1109.2534 in Electronic Archive http://arXiv.org.
53. Sharipov R. A., *A note on the second cuboid conjecture. Part I*, e-print arXiv:1201.1229 in Electronic Archive http://arXiv.org.
54. Sharipov R. A., *A note on the third cuboid conjecture. Part I*, e-print arXiv:1203.2567 in Electronic Archive http://arXiv.org.
55. Sharipov R. A., *Perfect cuboids and multisymmetric polynomials*, e-print arXiv:1205.3135 in Electronic Archive http://arXiv.org.
56. Sharipov R. A., *On an ideal of multisymmetric polynomials associated with perfect cuboids*, e-print arXiv:1206.6769 in Electronic Archive http://arXiv.org.
57. Sharipov R. A., *On the equivalence of cuboid equations and their factor equations*, e-print arXiv:1207.2102 in Electronic Archive http://arXiv.org.
58. Sharipov R. A., *A biquadratic Diophantine equation associated with perfect cuboids*, e-print arXiv:1207.4081 in Electronic Archive http://arXiv.org.
59. Ramsden J. R., *A general rational solution of an equation associated with perfect cuboids*, e-print arXiv:1207.5339 in Electronic Archive http://arXiv.org.
60. Ramsden J. R., Sharipov R. A., *Inverse problems associated with perfect cuboids*, e-print arXiv:1207.6764 in Electronic Archive http://arXiv.org.
61. Sharipov R. A., *On a pair of cubic equations associated with perfect cuboids*, e-print arXiv:1208.0308 in Electronic Archive http://arXiv.org.

62. Sharipov R. A., *On two elliptic curves associated with perfect cuboids*, e-print arXiv:1208.1227 in Electronic Archive http://arXiv.org.
63. Ramsden J. R., Sharipov R. A., *On singularities of the inverse problems associated with perfect cuboids*, e-print arXiv:1208.1859 in Archive http://arXiv.org.
64. Ramsden J. R., Sharipov R. A., *On two algebraic parametrizations for rational solutions of the cuboid equations*, e-print arXiv:1208.2587 in Electronic Archive http://arXiv.org.
65. Sharipov R. A., *A note on solutions of the cuboid factor equations*, e-print arXiv:1209.0723 in Electronic Archive http://arXiv.org.
66. Sharipov R. A., *A note on rational and elliptic curves associated with the cuboid factor equations*, e-print arXiv:1209.5706 in Electronic Archive http://arXiv.org.
67. Ramsden J. R., Sharipov R. A., *Two and three descent for elliptic curves associated with perfect cuboids*, e-print arXiv:1303.0765 in Archive http://arXiv.org.

Bashkir State University, 32 Zaki Validi street, 450074 Ufa, Russia
E-mail address: r-sharipov@mail.ru