



HAL
open science

A 3 RD SHORTER SOLUTION OF THE CLAY MILLENNIUM PROBLEM ABOUT $P \neq NP$ =EXPTIME

Constantine Konstantinos Kyritsis

► **To cite this version:**

Constantine Konstantinos Kyritsis. A 3 RD SHORTER SOLUTION OF THE CLAY MILLENNIUM PROBLEM ABOUT $P \neq NP$ =EXPTIME. 2022. hal-03601593v1

HAL Id: hal-03601593

<https://hal.science/hal-03601593v1>

Preprint submitted on 8 Mar 2022 (v1), last revised 19 Oct 2023 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A 3RD SHORTER SOLUTION OF THE CLAY MILLENNIUM PROBLEM ABOUT $P \neq NP = EXPTIME$.

Konstantinos E. Kyritsis Dept. Accounting-Finance University of Ioannina,
Greece, ckiritsi@uoi.gr

ABSTRACT

In this paper I present a 3rd very short proof that $P \neq NP$, and $NP = EXPTIME$ derived by simplification from my 2 previous published proofs, in the context of the Zermelo-Frankel set theory and deterministic Turing machines. The proof also gives that the well-known fact that the complexity class NP is contained in the complexity class EXPTIME, must be stated in the usual formal definition of the NP class (even at the standard formulation of this millennium problem by S Cook) because it is provable that it cannot be derived from the other two usual conditions of the formulation and if no added, the problem would become ILL POSED. The results of these proofs definitely solve the 3rd Clay Millennium Problem about P versus NP, in a simple and transparent way that the general scientific community, but also the experts of the area, can follow, understand and therefore become able to accept.

Key words: 3rd Clay Millennium problem, EXPTIME-complete problems, NP-complexity, P-complexity

Mathematical Subject Classification: 68Q15

1 INTRODUCTION

Two solution of the famous P versus NP problem has been published in [8],[9], [21] , 21] [22], Kyritsis K. and in this paper we present a 3rd very short simplification of the solution. This simplification also gives that the well-known fact that the complexity class NP is contained in the complexity class EXPTIME, must be stated in the usual stated definition of the NP class (even at the standard formulation of this millennium problem by S Cook) because it cannot be derived from the other 2 usual conditions of the formulation and if no added, the problem would become ill posed (see Remark 4.3

In the history of mathematics, it is known that difficult problems that have troubled a lot the mathematicians, turned out to have different proofs one simple and one very complex. Such an example is if the general 5th order polynomial equation can be solved with addition, subtraction, multiplication, division and extraction of radicals starting from the coefficients. The famous mathematician Niels Henrik Abel who gave a very

simple proof. On the other hand, the proof of the same, by the E. Galois theory, is a whole book of dozens of pages!

And a famous mathematician once said that *“Once a proof is known to a mathematical problem, then immediately after it becomes trivial!”*

It is the same with the solution of the P versus NP problem in this paper. We will utilize in our proofs, the **key abstraction** of the existence of an EXPTIME complete language, (it is known that it exists) without specifying which one, which will simplify much the arguments. Then we synthesize other languages and arguments over it, that will solve the problem.

A second issue that is important to mention, is a statement, that is usually attributed to the famous mathematician Yuri

Manin, that *“A correct proof in mathematics is considered a proof only if it has passed the social barrier of being accepted and understood by the scientific community and published in accepted Journals”*

Passing the obstruction of the social barrier, sometimes is more difficult than solving the mathematical problem itself!

We must notice here that the P versus NP problem, is in fact a set of different problems within different axiomatic systems. In the context of what axiomatic system is the Complexity Theory of Turing machines? Since the complexity theory of Turing machines requires entities like infinite sets of words then it is in the context of some axiomatic set theory, together with the axiom of infinite. So we notice that the next are different problems:

- 1) The P versus NP problem in the Zermelo-Frankel axiomatic system of sets without the axiom of choice and this axiomatic system formulated in the 2nd order formal languages.
- 2) The P versus NP problem in the Zermelo-Frankel axiomatic system of sets with the axiom of choice and this axiomatic system formulated in the 2nd order formal languages.
- 3) Etc

We might try to think of the P versus NP problem within the context of the axiomatic system of Peano Arithmetic with or without the axiom of induction and within second order formal languages. But to do so, we must carefully define, what additional axioms or definitions give the existence of infinite subsets of natural numbers that are used in the Complexity Theory.

The main hidden guiding idea in searching for such a simple proof, was that what the “arbitrary human-like free-will” of a non-deterministic Turing machine as human-machine interactive software (e.g. in password setting), can do in polynomial time cannot be done by a purely mechanical deterministic Turing machine in polynomial time. (See also beginning of paragraph 4) After the Key-abstraction mentioned above I had to find the right simple arguments to make a valid proof of this idea. The proof of the P versus NP problem in the direction $P \neq NP$, is supposed also to mean that the standard practice of encryption in the internet, is safe.

We notice also that the P versus NP:

- 1) It is a difficult problem, that has troubled the scientific community for some decades
- 2) It may have simple proofs of a few paragraphs, hopefully not longer than the proof of the Time Hierarchy theorem, which seems to be a deeper result.
- 3) But it can also have very lengthily and complex proofs, that may take dozens of pages.
- 4) There many researchers (more than 5 of them) that have claimed to have solved it, either as $P=NP$, or as $P \neq NP$, and even as suggestion that neither are provable, but only a handful of them seem to have been able to pass the preliminary social barrier and publish their solution in conferences or Journals with referees. The rest of them have published online only preprints (see e.g. the [17] P versus NP page). It seems to me though that it is not probable that all of them have correct solutions. Especially in the direction $P=NP$, there is a common confusion and mistake, that has been pointed out by Yannakakis M. [18]. Furthermore, this confusing situation has contributed so that although there are publications in respectable Journals, the experts and the scientific community does not seem of being able to decide if the P versus NP problem has been solved or not. This is reasonable, as there are proofs of close to 100 pages, and no average reader would feel comfortable to go through them, and decide for himself if there a flaw or error somewhere. Still it is better to have published results than non-published, and then let the large number of readers to try to find errors or flaws in the solutions if there are any.

So here comes the need of a more challenging problem: Not only to solve the P versus NP problem, but also solve it in such an simple, elegant and short way, so that the researchers will know a decisive proof that they can understand and control that $P \neq NP$ or not, so short that anyone familiar with the area, would discover any flaw or error if it existed.

This is I believe the value of the present paper that provides such a proof in the context of the Zermelo-Frankel set theory (we do not use the axiom of choice), formulated e.g. within 2nd order formal languages.

What this proof is or is not:

- 1) It does not introduce new theoretical concepts in computational complexity theory so as to solve the P versus NP.
- 2) It does not use relativization and oracles
- 3) It does not use diagonalization arguments, although the main proof, utilizes results from the time hierarchy theorem
- 4) It is not based on improvements of previous bounds of complexity on circuits
- 5) It is proved with the method of counter-example. Thus it is transparent short and “simple”. It takes any Exptime-complete DTM decision problem, and from it, it derives in the context of deterministic Turing machines a decision

problem language which it is apparent that it belongs in the NP class decision problems while it does not belong the class P of decision problems.

- 6) It seems a “simple” proof because it chooses the right context to make the arguments and constructions and the key-abstraction mentioned above. So it helps the scientific community to accept that this 3rd Clay Millennium problem has already been solved.

In the paragraph 4, we give an advanced, full proof that $P \neq NP$, in the standard context of deterministic Turing machines, solving thus the 3rd Clay Millennium problem.

2. PRELIMINARY CONCEPTS, AND THE FORMULATION OF THE 3RD CLAY MILLENNIUM PROBLEM, P VERSUS NP.

In this paragraph, for the sake of the reader, we will just mention the basics to understand the formulation of the 3rd Clay Millennium problem. The official formulation is found in [3] (Cook, Stephen (*April 2000*), *The P versus NP Problem (PDF)*, *Clay Mathematics Institute site*). Together with an appendix where there is concise definition of what are the Deterministic Turing machines, that is considered that they formulate, in Computational Complexity theory, the notion and ontology of the software computer programs.

In the same paper are also defined the computational complexity *classes P, NP*.

The elements of the classes P, NP etc strictly speaking are not only sets of words denoted by L, that is not only languages, but also for each such set of words or language L at least one DTM, M that decides it, in the specified complexity so they are pairs (L,M). Two such pairs (L_1, M_1) (L_2, M_2) are called *equidecidable* if $L_1 = L_2$ although it may happen that $M_1 \neq M_2$. E.g. if the complexity of M_1 is polynomial-time while that of M_2 exponential-time choosing the first pair instead of the second means that we have turned a high complexity problem to a feasible low complexity problem.

The definition of other computational complexity classes like **EXPTIME** etc. can be found in standard books like [6],[11],[12]. In the official formulation [3] there is also the definition of the concept of *a decision problem language in polynomial time reducible to another decision problem language*.

Based on this definition it is defined that an EXPTIME-complete decision language of EXPTIME is EXPTIME-complete, when all other decision problems languages of EXPTIME have a polynomial time reduction to it. Here is the exact definition

Definition 2.1 Suppose that L_i is a language over all words Σ_i , $i = 1, 2$. Then $L_1 \leq_p L_2$ (L_1 is p-reducible to L_2) iff there is a polynomial-time computable function $f: \Sigma_1 \rightarrow \Sigma_2$ such that $x \in L_1$ if and only if $f(x) \in L_2$, for all $x \in \Sigma_1$.

In the same books [6],[10],[11] can be found the concepts and definitions of **NP-complete and EXPTIME-compet** decision problems. See also [7], [12] where it is proved that specific decision problems are EXPTIME-complete.

For simplicity we will consider here only binary alphabets $\{0,1\}$ and sets of binary words Σ .

3. WELL KNOWN RESULTS THAT WILL BE USED.

We will not use too many results from the computational complexity theory for our proof that $P \neq NP$.

A very deep theorem in the Computational Complexity is the *Time Hierarchy Theorem* (see e.g. [6],[11],[12],[10],[14]). This theorem gives the existence of decision problems that cannot be decided by any other deterministic Turing machine in less complexity than a specified.

Based on this theorem, it is proved that:

Proposition 3.1 *There is at least one EXPTIME-complete decision problem, that cannot be decided in polynomial time, thus $P \neq EXPTIME$.*

The next two propositions indicate what is necessary to prove in order to give the solution of the P versus NP problem.

Proposition 3.2 *If the class NP contains a language L which cannot be decided with a polynomial time algorithm, then $P \neq NP$.*

Proposition 3.3 *If the class NP contains a language L which is EXPTIME complete, then $NP=EXPTIME$.*

4. THE SOLUTION: $P \neq NP=EXPTIME$ IN THE CONTEXT OF DETERMINISTIC TURING MACHINES

We will prove in this paragraph that $P \neq NP$ in the context of second order formal language of the Zermelo-Frankel set theory.

Since we are obliged to take strictly the official formulation of the problem, rather than text books about it, we make the next clarifications.

We will use the next conditions for a Language to be in the class NP, as stated in the official formulation of the P versus NP problem (see [3] Cook, Stephen (April 2000), *The P versus NP Problem (PDF)*, Clay Mathematics Institute.).

We denote by Σ^* all the words of an alphabet Σ .

Definition 4.1 *A language L of binary words is in the class NP if and only if the next conditions hold*

- 1) *There is a deterministic Turing machine M that decides L. In other words for any word x in L, when x is given as input to M, then M accepts it and if x does not belong to L then M rejects it.*

In symbols: \exists a deterministic Turing machine M , such that $\forall x \in \Sigma^*$, x is either accepted or rejected by M and if M accepts $x \rightarrow x \in L$, and if M reject $x \rightarrow x \notin L$

- 2) There is a polynomial-time checkable relation $R(x,y)$, and a natural number k of N , so that for every word x , x belongs to L if and only if there is a word y , with $|y| \leq |x|^k$, and $R(x,y)$ holds.

In symbols: \exists relation R which is polynomial-time checkable, and $\exists k \in N$, such that $\forall x \in \Sigma^*$,

$x \in L \leftrightarrow (\exists y \in \Sigma^*, |y| \leq |x|^k \text{ and } R(x,y) \text{ holds}).$

- 3) The complexity class NP is contained in the complexity class $EXPTIME$.

Remark 4.1. In the official statement of the P versus NP problem (see [3] Cook, Stephen (April 2000), *The P versus NP Problem (PDF)*, Clay Mathematics Institute) the condition 1) is not mentioned. But anyone that has studied complexity theory, knows that it is required. The condition 2) alone cannot guarantee that there is a deterministic Turing machine that decides the language., as the polynomial checkable relation works only if we provide it with certificate y , and not with only x as input. Indeed we shall see below at the end of the proposition in **Remark 4.4**, that there is even an undecidable language L , for which nevertheless there is a polynomial checkable relation R , so that condition R is satisfied. The languages of NP cannot be semidecidable (or undecidable). The NP class is also defined as $NP = \cup_{k \in N} NTIME(n^k)$, but this definition is also in the context of non-deterministic Turing Machines. The situation with P , is more clear, because the mere requirement that a language of P is of polynomial time complexity as it is standard to define it, involves already that there exist a deterministic Turing machines that for every input word, it halts within polynomial time steps and either accepts or rejects it, therefore it decides it. And not only that is simply the language of a deterministic Turing machine, and therefore maybe only semi-decidable.

Remark 4.2. Notice that in the condition 2) the k depends on the relation R and is not changing as the certificate y changes. In other words k does not depend on y and we *did not* state the next:

There is a polynomial-time checkable relation $R(x,y)$, so that for every word x , x belongs to L if and only if there is a word y , and k in N , with $|y| \leq |x|^k$, and $R(x,y)$ holds. In symbols: \exists relation R which is polynomial-time checkable, such that $\forall x \in \Sigma^$, $x \in L \leftrightarrow (\exists y \in \Sigma^* \text{ and } \exists k \in N \text{ such that } |y| \leq |x|^k \text{ and } R(x,y) \text{ holds}).$*

In the official statement of the P versus NP problem (see [3] Cook, Stephen (April 2000), *The P versus NP Problem (PDF)*, Clay Mathematics Institute) this is not made clear, in the natural language that the definition is stated. But that k does not depend on the certificate, but on the polynomial checkable relation becomes clear, when we look at the proof in any good textbook about complexity theory, of how a non-deterministic Turing machine which runs in polynomial time, can define a

deterministic Turing machine with a polynomial time checkable relation, which is considered that replaces it.

Remark 4.3

A further point to notice is that the condition **3) *The complexity class NP is contained in the complexity class EXPTIME is usually omitted in the definition of NP in the books and even in the official formulation of the corresponding Clay Millennium problem. Nevertheless it holds (see [6], [19]) and by omitting it we run the risk to have the problem P vs NP as ILL POSED problem because this condition 3) cannot be derived from the conditions 1) and 2) but it is derived from the initial definition of non-deterministic procedures as half machine half interactive and during the proof that equivalently compute the same results as appropriately defined 100% machine procedures.***

Here is an argument which shows that by omitting it in the definition of NP it becomes ill posed. What is remarkable is that the same arguments is also the simplest and shortest solution of the P vs NP millennium problem! It occurred to me when I was trying to simplify my 2 previous solutions of this millennium problem in [20] and [22].

We shall prove by counter-example that any language of any complexity class can be identified as a language of the complexity class NP, if the complexity class NP is defined as in the definition 4.1 and only with the conditions 1) , 2) without the condition 3).

- 1) Let us take an arbitrary Language $L=\{x_1, x_2, \dots, x_n.. \}$ decidable by a deterministic Turing machine, in any complexity class possible of higher complexity than he EXPTIME complexity class.
- 2) From the previous assumption the language L satisfies the first condition 1) of the definition of the NP class.
- 3) In the definition 4.1 and in the condition 2) the certificate y, just exists and there is not assumption that it requires a particular level of complexity deterministic Turing machine to find it. So we choose for any word x of L as certificate y the same word x itself Thus by definition $x=y$ and the condition $|y|<|x|^k$ holds for $k=1$.
- 4) Then as polynomial checking relation $R(x,y)$ we take the equality of words, which can be checked in polynomial time relative to the length of the words.
- 5) We conclude that all two conditions 1) and 2) are satisfied thus the language L as a type of language which belongs to the complexity class NP.
- 6) This would give that any complexity class can be identified as NP class, thus the definition of the complexity class NP as well as the problem P vs NP are ill posed.
- 7) We conclude that the condition 3) , which we do know that it holds from the very definition of non-deterministic Turing machines, must be added to the definition of the complexity class NP and is not derivable from the conditions 1) and 2)

- 8) Then we just have to start with a language L_e which is EXPTIME complete language. We know from the hierarchy theorem that such a language does exist.
- 9) We repeat the same arguments as in 1), 2), 3), 4), 5) and we conclude that the language L_e belongs to the NP class that $EXPTIME \leq NP$. This together with the condition 3) of the definition 4.1 $NP \leq EXPTIME$ gives that $NP = EXPTIME$. And thus we conclude that P is not equal to NP , $P \neq NP$ and the P vs NP millennium problem has been solved.

We summarize the solution in the next proposition.

Proposition 4.1 (3rd solution of the 3rd Clay Millennium problem P vs NP)
There is at least one decision problem language of the class NP which is not also in the class P. Therefore, $P \neq NP$. It holds also that $NP = EXPTIME$.

Proof:

- 1) We just have to start with a language $L_e = \{x_1, x_2, \dots, x_n\}$ which is EXPTIME complete language. We know from the hierarchy theorem that such a language does exist.
- 2) From the previous assumption the language L_e satisfies the first condition 1) of the definition of the NP class.
- 3) In the definition 4.1 and in the condition 2) the certificate y , just exists and there is not assumption that it requires a particular level of complexity deterministic Turing machine to find it. So we choose for any word x of L_e as certificate y the same word x itself Thus by definition $x=y$ and the condition $|y| < |x|^k$ holds for $k=1$.
- 4) Then as polynomial checking relation $R(x,y)$ we take the equality of words, which can be checked in polynomial time relative to the length of the words.
- 5) We conclude that all two conditions 1) and 2), 3) are satisfied thus the language L_e as a type of language which belongs to the complexity class NP.
- 6) We have concluded that the language L_e belongs to the NP class and as it is EXPTIME complete we get that that $EXPTIME \leq NP$. This together with the condition 3) of the definition 4.1 $NP \leq EXPTIME$ gives that $NP = EXPTIME$. And thus we conclude that P is not equal to NP and the P vs NP millennium problem has been solved. QED.

5. CONCLUSIONS

Sometimes great problems have relatively short and elegant solutions provided we find the **key-abstractions** and convenient context, symbols and semantics to solve them. It requires also a certain power of thinking rather than complexity of thinking, in areas where traditionally and collectively it may not exist before. Even relatively simple paths of reasoning, may be difficult to travel, if there is not, at a certain point of them, the necessary “bridge”, that is the necessary key-abstraction or the right conceptual “coins” of symbols and semantics to exchange and convert. Here the key-abstraction was to start from the class EXPTIME and

an EXPTIME-complete language of it, without specifying which one instead starting from the class NP. If the P versus NP problem is researched without a main strategy, that would require a short proof, it might become a very complex problem to solve. *The main hidden guiding idea in searching for such a simple proof, was that, what the “arbitrary human-like free-will” of a non-deterministic Turing machine as human-machine interactive software (e.g. in password setting), can do in polynomial time, cannot be done by a purely mechanical deterministic Turing machine in polynomial time. In other words the human-like non-deterministic arbitrariness in Turing machines has an exponential nature.* Since in my opinion the Hierarchy Theorem is a deeper result than the P versus NP problem, in principle there should exist a not much more complicated proof of the P versus NP problem, compared to the proof of the Hierarchy Theorem. *The proof of the P versus NP problem in the direction $P \neq NP$, is supposed also to mean that the standard practice of password setting in the internet, is safe when the encryptions is not corrupted and the publicly available hardware computational power is the same for all .*

REFERENCES

- [1] Conway J.H. *On numbers and games*, Academic press 1976
- [2] Cook, Stephen A. (1972). "A hierarchy for nondeterministic time complexity". Proceedings of the fourth annual ACM symposium on Theory of computing. STOC '72. Denver, Colorado, United States: ACM. pp. 187–192
- [3] Cook, Stephen (April 2000), *The P versus NP Problem (PDF)*, Clay Mathematics Institute site.
- [4] Diduch Rodrigo Gilberto (2012), *P vs NP*, International Journal of Computer Science and Network Security (IJCSNS) Volume 2, pp 165-167.
- [5] Gram Seenil 2001 "Redundancy, Obscurity, Self- Containment & Independence" by The 3rd International Conference on Information and Communications Security (ICICS 2001) took place in Xian, China, November 13-16, 2001. Proceedings of ICICS as Volume 2229 of Springer Lecture Notes in Computer Science. Pages 495-501.
- [6] Harry R. Lewis and Christos H. Papadimitriou *Elements of the Theory of Computation*, Prentice-Hall, Englewood Cliffs, New Jersey, 1981, ISBN 0-13-273417-6.
- [7] Hartmanis, J.; Stearns, R. E. (1 May 1965). "On the computational complexity of algorithms". Transactions of the American Mathematical Society. *American Mathematical Society*. **117**: 285–306. . ISSN 0002-9947. JSTOR 1994208. MR 0170805.
- [8] Kyritsis C. *On the solution of the 3rd Clay Millennium problem. A short and elegant proof that $P \neq NP$ in the context of deterministic Turing machines*

and Zermelo-Frankel set theory. Proceedings of the first ICQSBEI 2017 conference, Athens, Greece, pp 170-181

- [9] Kyritsis C THE SOLUTION OF THE 3RD CLAY MILLENNIUM PROBLEM. A SHORT PROOF THAT $P \neq NP=EXPTIME$ IN THE CONTEXT OF ZERMELOFRANKEL SET THEORY. *International Journal of Pure and Applied Mathematics Volume 120 No. 3 2018, pp 497-510 ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version) url: <http://www.ijpam.eu> doi: 10.12732/ijpam.v120i3.1*
- [10] Luca Trevisan, *Notes on Hierarchy Theorems*, U.C. Berkeley.
- [11] John C. Martin (1997). *Introduction to Languages and the Theory of Computation (2nd ed.)*. McGraw-Hill. ISBN 0-07-040845-9.
- [12] Papadimitriou Christos (1994). *Computational Complexity*. Addison-Wesley. ISBN 0-201-53082-1.
- [13] Rustem Chingizovich Valeyev 2013 *The Lower Border of Complexity of Algorithm of the Elementary NP-Complete Task (The Most Condensed Version)* World Applied Sciences Journal 24 (8): 1072-1083, 2013 ISSN 1818-4952 © IDOSI Publications, 2013.
- [14] Stanislav, Žák (October 1983). "A Turing machine time hierarchy". *Theoretical Computer Science*. Elsevier Science B.V. **26** (3): 327–333.
- [15] A. A. Tsay, W. S. Lovejoy, David R. Karger, *Random Sampling in Cut, Flow, and Network Design Problems*, *Mathematics of Operations Research*, 24(2):383–413, 1999.
- [16] Ivanov Viktor V. 2014, *A short proof that NP is not P*. *International Journal of Pure and Applied Mathematics IJPAM* pp 81-88 .
- [17] Woeginger GJ (2016) *The P versus NP page* ,<https://www.win.tue.nl/~gwoegi/P-versus-NP.htm>
- [18] Yannakakis M. 1998 "Expressing combinatorial optimization problems by linear programs" *Proceedings of STOC 1988*, pp. 223-228.
- [19] <https://en.wikipedia.org/wiki/EXPTIME>
- [20]] Kyritsis K. 2021 "REVIEW OF THE SOLUTIONS OF THE CLAY MILLENNIUM PROBLEM ABOUT $P \neq NP =EXPTIME$ ". September 2021 *World Journal of Research and Review* 13(3):21-26 <https://www.wjrr.org/vol-13issue-3>
- [21] Kyritsis K. 2021 Author of chapter 5 in the book "New Visions in Science and Technology Vol.6, October 2, 2021 , Page 60-69 <https://doi.org/10.9734/bpi/nvst/v6/5176F> Published: 2021-10-02 Chapter in a book <https://stm.bookpi.org/NVST-V6/article/view/4135>

[22] Kyritsis K 2021 Author of a book with title: The Solutions of the 3rd and 4th Millennium Mathematical Problems: The solutions of the Millennium Problem P vs NP in computational complexity and the Millennium problem in fluid dynamics. Publisher : Lap Lambert Academic Publishing. ISBN 6204725629. <https://www.lap-publishing.com/catalog/details//store/gb/book/978-620-4-72562-8/the-solutions-of-the-3rd-and-4th-millennium-mathematical-problems>

Author: Konstantinos E. Kyritsis Dept. Accounting-Finance University of Ioannina, Greece, ckiritsi@uoi.gr