American Journal of Systems and Software, 2014, Vol. 2, No. 6, 146-150
Available online at http://pubs.sciepub.com/ajss/2/6/2

DOI:10.12691/ajss-2-6-2

Introduction to Coding Theory for Flow Equations of Complex Systems Models

J. Nescolarde-Selva*, J.L. Usó-Doménech, M. Lloret-Climent

Department of Applied Mathematics, University of Alicante, Alicante, Spain
*Corresponding author: josue.selva@ua.es

Received October 15, 2014; Revised December 06, 2014; Accepted December 10, 2014

Abstract The modeling of complex dynamic systems depends on the solution of a differential equations system. Some problems appear because we do not know the mathematical expressions of the said equations. Enough numerical data of the system variables are known. The authors, think that it is very important to establish a code between the different languages to let them codify and decodify information. Coding permits us to reduce the study of some objects to others. Mathematical expressions are used to model certain variables of the system are complex, so it is convenient to define an alphabet code determining the correspondence between these equations and words in the alphabet. In this paper the authors begin with the introduction to the coding and decoding of complex structural systems modeling.

Keywords: *alphabet, code, complex models, decipherability, flow equations, transformed functions*

Cite This Article: J. Nescolarde-Selva, J.L. Usó-Doménech, and M. Lloret-Climent, "Introduction to Coding Theory for Flow Equations of Complex Systems Models." *American Journal of Systems and Software*, vol. 2, no. 6 (2014): 146-150. doi: 10.12691/ajss-2-6-2.

1. Introduction

Modeling complex systems with a particular methodology mathematical equations are obtained, which analyze and study certain processes. Because of the importance of these systems to simulate different situations, it is convenient to have a tool to compare models with each other. Therefore we must be able to store and save the equations that interest us, and later retrieve and manipulate them. In this sense we need to encode all the words of the language used in mathematical modeling, which has been developed in previous works [1-16]. It is impossible to store all of the equations involved in the selection of intermediate, since the complexity of processes and the number of equations can be excessive.

The modeling process the authors use to deal with complex reality, specifically ecosystems [3,4], is based on the following assumptions:

1. The building of a causal model based on previous theories of reality which can be divided into the following phases:

a) Choose relevant objects or variables related to the proposed goals. Ecological, biological, etc. theories would be the theoretical base of this phase. However, subjective components (intuition, brainstorming, etc.) play an important role.

b) Identify the cause-and-effect relationship between the considered elements. Subsystems diagram, policy structuring diagram, multivariate analysis, etc. may be added.

c) Give a functional representation to the detected relations; that is to say; write them as state equations. The mathematical meta-language gives the laws for this.

2. Experimentation to obtain variable (measurable attributes) data.

3. Creation of flow equations through experimental data.

4. Integration of the system of the ordinary differential equations (state equations) through numerical methods.

Figure 1 shows a clear representation of this process.

We assume [4] that the dynamics of the system can be modeled starting off with a set of ordinary non-lineal differential equations,

$$\frac{dy_j}{dt} = \sum_{i=1}^{n} x_{ij}, \forall j = 1, 2, \ldots, n \tag{1}$$

where the x_{ij} are the flow variables which produce the state variable y_j [17]. The equations associated with flow variables receive the name of flow equations. Said equations represent the biological, chemical and physical processes in the ecosystems. They show the relations between external variables (forcing functions) and state variables (Jorgensen, 1988). Each one of the flow variables can depend either on the input variables or on the state variables. Then (1) can be defined in the following way using transformed functions,

$$\begin{aligned}\frac{dy_j}{dt} &= \sum_{i=1}^{n}\sum_{r=1}^{n} c_r^1 T^1(z_r) + \sum_{i=1}^{n}\sum_{r=1}^{n}\sum_{s=1}^{n} c_{rs}^2 T^2(z_r, z_s) \\ &+ \ldots + \sum_{i=1}^{n}\sum_{r=1}^{n}\sum_{s=1}^{n} \ldots \sum_{u=1}^{n} c_{rs\ldots u}^p T^p(z_r, z_s, \ldots, z_u) + \ldots.\end{aligned} \tag{2}$$

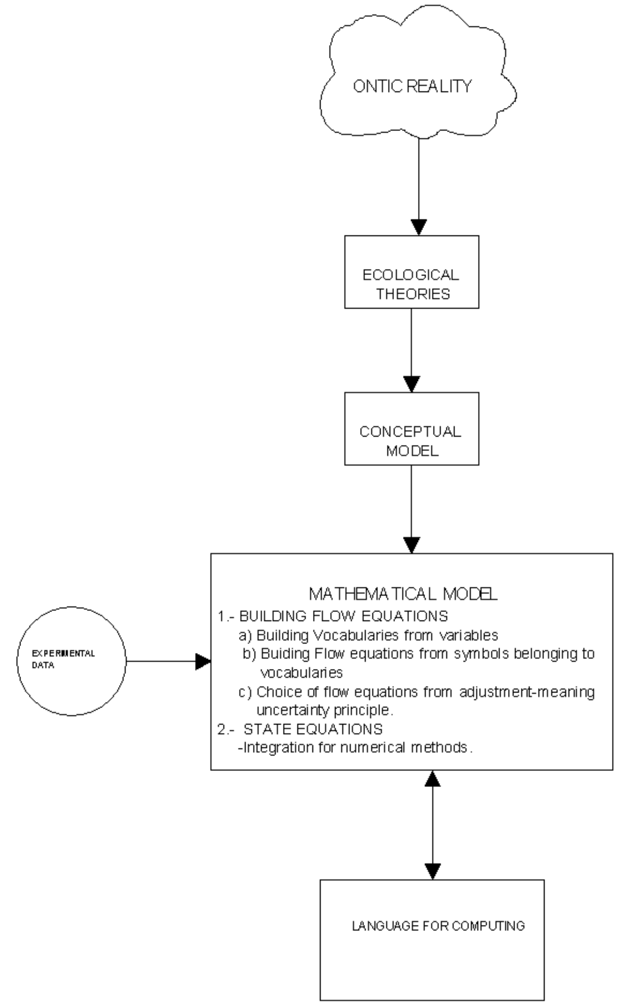

Figure 1. Diagram of authors' methodology

We define as *associative field of a measurable attribute* w and we called Φ_w, the set constituted by all possible symbols of said measurable attribute: $\Phi_w = \left\{\varphi_w^0, \left\{\varphi_w^1\right\}, \left\{\varphi_w^2\right\} .... \left\{\varphi_w^n\right\}, ...\right\}$. The set Φ_w will be a denumerable set. In the practical tool, it will be a requisite to define one subset $V_w \subset \Phi_w$ whose cardinal will be an integer number. The associative field of a measurable attribute w will be called *First Order Vocabulary* (FOV) or Vocabulary of order one and will be denoted by V_w^1. The elements of V_w^1 will be called t-symbols and will be denoted by $\varphi^j{}_i$, where i represents an index of the symbol and j denotes the order of transformation. The measurable attributes are a particular case of the t-symbols. The set X formed by a FOV generated by the set of measurable attributes $W = \{w_1, w_2, ... w_n\}$ will be called *Primary Lexicon* (PL) or *alphabet of the n-order monoads*, $X = \left\{V_{w_1}^1, V_{w_2}^1, ..., V_{w_n}^1\right\}$

The *primitive monoad* or *alphabet* A is formed by a set W of characters used to express measurable attributes $W = \left\{w_1, w_2, ..., w_{n,...}\right\}$, a set D of differential functions in relation to time $D = \left\{\dfrac{d}{dt}\right\}$ and a set Φ of n-order monoads $\Phi = \left\{\left\{\varphi^1\right\}, \left\{\varphi^2\right\}, ..., \left\{\varphi^n\right\}\right\}$. The W set is formed by the input and state variables, and $A = W \cup D \cup \Phi$.

The *textual alphabet* A_t is jointly built with the alphabet A and the set R of real numbers (model parameters) $R = \{r / r \in \Re\}$.

The *Simple Lexical Units* (SLUN) are constituted by the elements of the set A-D.

The *Operating Lexical Units* or operator-LUN (op-LUN) are the mathematical signs +, -.

The *Ordenating Lexical Units* or Ordenating-LUN (or-LUN) are the signs =, <, >.

The *Special Lexical Unit* (SpLUN) is the sign d/dt, which belongs to the alphabet A and defines the beginning of a phrase (state equation). The *differential vocabulary* or d-vocabulary of a measurable attribute w, V_w^{∂}, is the set formed by all partial derivatives of any order of w with respect to any other measurable attribute and the time t.

The *primary differential vocabulary*, $V_w^{1\partial}$, is the set formed by all partial derivatives of order 1 of w with respect to any other measurable attribute and the time t. $V_w^{1\partial} = \left\{\dfrac{\partial w}{\partial t}, \dfrac{\partial w}{\partial y}, ...\right\}$.

Secondary a *higher order differential vocabularies* may also be defined and will be denoted by $V_w^{n\partial}$, $n \geq 1$. For ease of calculation in practical complex system modeling, we define a subset of $V_w^{1\partial}$ called *dimensional primary differential vocabulary*, ${}^{XYZt}V_w^{1\partial}$, consisting of all partial first order derivatives of the measurable attribute w with respect to the three spatial dimensions X, Y, Z and time t, ${}^{XYZt}V_w^{1\partial} = \left\{\dfrac{\partial w}{\partial X}, \dfrac{\partial w}{\partial Y}, \dfrac{\partial w}{\partial Z}, \dfrac{\partial w}{\partial t}\right\}$.

To implement the models of the System Dynamics (Forrester, 1961), a subset of cardinal 1, ${}^tV_w^{1\partial}$ and whose only element is the partial derivative of the p-symbol with respect to the time, will be used.

Let $w_1, w_2, ..., w_n$ be a set of measurable attributes. The *differential Lexicon*, d-L, is the set formed by the d-vocabularies generated by the measurable attributes,

$$d - L = \left\{\begin{matrix} V_{w_1}^{1\partial}, V_{w_2}^{2\partial}, ..., V_{w_2}^{n\partial}; V_{w_2}^{1\partial}, V_{w_2}^{2\partial}, \\ ..., V_{w_2}^{n\partial}; ...; V_{w_n}^{1\partial}, ..., V_{w_n}^{n\partial} \end{matrix}\right\}$$

The Elements of d-L will be called *d-symbols*. The characters (,), {,}, [,], are simply signs since they lack of meaning and they are the equivalent to the signs ?, !,; (,) in the natural languages.

The *Separating of Lexical Units* (s-LUN) are the signs * and /.

The *Composed Lexical Units* (CLUN) are the strings of a SLUN separated by a s-LUN. The *syllables* or composed Lexical units (CLUN) are constituted by a SLUN, or a chain of them, separated by an op-LUN or a or-LUN.

The *word* is the SLUN or CLUN. The symbols [·] preceding the other symbols + or – are word separations.

The *opsep vocabulary* V^S is the one formed by operating and separating LUNs. $\otimes \in V^S; \otimes = \{+, -, *, :\}$ and it will be written a element of *VS* by $\otimes$.

A *simple sentence* is a flow variable [17]. It is built by a CLUN or a combination of CLUNs.

The *vocabulary of order n* $V^n_{w_1 w_2 \dots w_n}$ is the one formed by simple sentences

$$\begin{aligned}&V^n_{w_1 w_2 \dots w_n} = \{\varphi_i \otimes \varphi_j \otimes \dots \otimes \varphi_\omega; \\ &\varphi_i \in V^1_{w_1}, \varphi_j \in V^1_{w_2}, \dots, \varphi_\omega \in V^1_{w_n}\} = \\ &\{\Psi^n_{w_1 \dots w_n} / \Psi^n_{w_1 \dots w_n} = \varphi_i \otimes \varphi_j \otimes \dots \otimes \varphi_\omega; \varphi_i\}\end{aligned}$$

A short notation would be $\phi^n_{w_1, w_2, \dots, w_n} = \varphi_{i_1} \otimes \dots\dots \otimes \varphi_{i_n}$.

The set of all vocabularies of any order is called *t-Lexicon* t-L, and it is formed by the FOV and simple sentence vocabularies.

$$t - L = \left\{\begin{matrix} V^1_{w_1}, V^1_{w_2}, \dots, V^1_{w_n}, V^2_{w_1 w_2}, V^2_{w_2 w_3}, \\ \dots, V^2_{w_1 w_n}, V^2_{w_2 w_3}, \dots, V^2_{w_{n-1} w_n}, V^n_{w_1 w_2 \dots w_n} \end{matrix}\right\}$$

$$\left\{V^1_{x_1}, V^1_{x_2}, \dots, V^1_{x_n}, V^2_{x_1}, V^2_{x_2}, \dots, V^2_{x_n}, \dots, V^n_{x_1}, V^n_{x_2}, \dots, V^n_{x_n}\right\}$$

The set Φ will be a subset of t-L.

Let $\{\phi_n\}_{i=1.,\dots,n} \in V^1_{i=1,\dots,n}$. We say that $\phi_1, \phi_2, \dots, \phi_n$ are related linguistically in a *n-order relationship* and we call it $(\phi_1, \phi_2, \dots, \phi_n) \in r_n$ if and only if $(\exists \otimes \in V^S) \vee (\exists V^n_{12\dots n}) \vee (\exists \Psi^n_{12\dots n} \in V^n_{12\dots n})$ and $\Psi^n_{12\dots n} = \phi_1 \otimes \dots \otimes \phi_n$. We will call R_L the whole of all linguistic relationships $r_L; L = 1, 2, \dots, n$. Let $V^n_{12\dots n}, V^m_{12\dots m}, \dots, V^l_{12\dots l}$ be vocabularies of *n*, *m*,...,*l* orders, respectively. We say that $V^n_{12\dots n}, V^m_{12\dots m}, \dots, V^l_{12\dots l}$ are *related linguistically* and we will call it $(V^n_{12\dots n}, V^m_{12\dots m}, \dots, V^l_{12\dots l}) \in r_V$ if and only if $V^h_{12\dots h} / h = n + m + \dots + l$ vocabulary exists so that

$$\begin{aligned}&(\exists \Psi^n_i \in V^n_{12\dots n}) \wedge (\exists \Psi^m_j \in V^m_{12\dots m}) \wedge \dots \\ &\wedge (\exists \Psi^l_k \in V^l_{12\dots l}) \wedge (\exists \oplus \in V^S) \wedge (\exists A^h_{ij\dots k} \in V^h_{12\dots h})\end{aligned}$$

where $A^h_{ij\dots k} = \Psi^n_i \oplus \Psi^m_j \oplus \dots \oplus \Psi^l_k$.

A *complex sentence* is each ordinary differential equation (ODE) or state equation, which is built by linear combination of simple sentences $A^h_{ij\dots k} = \Psi^n_i \oplus \Psi^m_j \oplus \dots \oplus \Psi^l_k$. A *text* T = (L, A) is the concatenation of complex sentences, determined by the argument A of the text or semantic links between these complex sentences.

The *Lexicon L of a text* is the union between the t-Lexicon and the differential Lexicon, $L = t - L \cup d - L$. We can say that the text is written in a formal language, and we call it as L(M_T). Everything according [13].

The building of flow equations is based on the following processes:

a) With the symbols of the t-Lexicon the word is built (flow equation), whose components are connected together by means of an operator $\otimes$, i.e., $\otimes = \{+, -, x, :\}$. The length l of the word will be $l \geq p$, where p is the number of independent variables (primitives) used in the model.

b) Once the words are built, whose number, say q, will depend on the biggest order of the transformed function, on the modeler and on the experimental data, a process of recognition is generated where only a number of words say w, will be left, that is, those that are "correct". The rest (q-w) words are considered "incorrect". The "correction" criteria will be determined according to different criteria of recognosibility.

c) With the "correct" words, state equations will be constructed,

$$\frac{dy_j}{dt} = A_j = \sum_{i=1}^{n} \Psi_{ij} \quad j = 1, 2, \dots, n \tag{3}$$

where A_j are the flow functions or sentences (the right hand side of ordinary differential or state equations).

d) The procedure of numeric integration of ordinary differential equations will be determined by the modeler according to the needed precision, and in turn depending on the model disaggregation, the economy of calculation, etc., and finally on the preference of the modeling agent.

2. Recognition Code of Flow Equations

Given a complex system and a variable "A", which represents a particular process to be studied, we consider the flow equation:

$$A = F\ (x_1, \dots\dots, x_n) \tag{4}$$

Whatever the method used, the equation (4) will be defined mathematically in a language. The flow equation (4) is expressed by linear combinations of transformed functions (Usó-Domènech, Mateu, and Lopez., 1997).

Elects {f_i}, the flow equation (4) can come modeled as:

$$A = a_o + a_1 f_1(x_1) + \dots\dots + a_n f_2 o f_3(x_n) \tag{5}$$

In complex systems, modeling of the flow equation is complex, so it is necessary to express them by a symbol ϕ_a and a code which allows obtaining immediately the corresponding mathematical expression. Next will be defined an alphabet source of symbols ϕ_a, to represent the flow equations, an alphabet code consisting of elementary functions including in them the identity function and coding rules.

An alphabet U is considered such that U = { ϕ_a / a is a string of length m} where each ϕ_a is a letter. Is defined by S (U) the language generated by U. Denote by S'(U) the subset consisting of the words chosen by the model builder according to certain pre-established criteria.

Let $\mathcal{U}$ be an alphabet consisting of a finite numbers of letters be given

$$\mathcal{U} = \{\varphi_a, \varphi_b, \dots\dots\dots, \varphi_r\} \tag{6}$$

Each symbol φ_i has a subscript, i, formed by a string of m numerical characters (m is the upper order of the used transformed equations by the structural complex model). We call alphabet $\mathcal{U}$ the "transformed equations alphabet".

We call a finite string of symbols

$$\Psi_{i1i2...in} = \varphi_{i1}\varphi_{i2}\cdots\cdots\varphi_{in} \quad (7)$$

a word in $\mathcal{U}$, and the value n its length (to be denoted by $l(\Psi_{i1i2...in})$. Let S = S($\mathcal{U}$) be the set of all non-zero words in $\mathcal{U}$, and S’ a subset of S. S’ is the set of words chosen by L(C).

The object generating words from S’ is called a message source and the words from S’ messages. The words $\Psi_{i1i2...in}$ are flow equations.

Consider that an alphabet $\mathcal{B}$

$$\mathcal{B} = \{f_0, f_1, f_2, \ldots., f_q\} \quad (8)$$

is given. f_0 is the identity function, that is to say $f_0(x) = x$ and f_j, j = 1, 2,,,, q elementary functions.

Let B be a word in $\mathcal{B}$, and by S($\mathcal{B}$) the set of all non-zero words in $\mathcal{B}$.

Let F be a mapping associating the word

$$B = f(\Psi_{i1i2...in}),\ B \in S(\mathcal{B}) \quad (9)$$

with each word $\Psi_{i1i2...in} \in S'(\mathcal{U})$ be given.

We call B the message code, and the transition from the message $\Psi_{i1i2...in}$ to its incoding code.

In coding theory [19,20,21], mappings F are given by an algorithm.

Consider the correspondence between the letters of the alphabet

$$\mathcal{U} = \{\varphi_a, \varphi_b, \ldots\ldots., \varphi_r\} \quad (10)$$

and certain words in the alphabet

$$\mathcal{B} = \{f_0, f_1, f_2, \ldots., f_q\} \quad (11)$$

viz.,

$$\begin{aligned} &\varphi_{\underbrace{00...0i}_{n}} \to f_0 f_0 \cdots\cdots f_0 f_i \\ &\varphi_{\underbrace{00...0ij}_{n}} \to f_0 f_0 \cdots\cdots f_0 f_i f_j \\ &\varphi_{\underbrace{abc...k}_{n}} \to f_a f_b f_c \cdots\cdots f_k \end{aligned} \quad (12)$$

where $f_a f_b \ldots\ldots f_k$ means the composition of the functions, that is to say $f_a o f_b o \ldots\ldots o f_k$. This correspondence is called scheme, and denoted by ∑. It determines alphabet coding as follow: each word $\Psi_{i1i2...in} = \varphi_{i1}\varphi_{i2}\ldots\ldots\ldots \varphi_I$ from S’($\mathcal{U}$) is associated with the word $B = B_{i1}B_{i2}\ldots\ldots.B_{in}$, called the code for $\Psi_{i1i2...in,}$ being each B_i elementary codes of the scheme.

Example 1:

Variables: x_1, x_2, x_3.

Elementary functions:

$$f_1(x) = \sin\ (x)$$

$$f_2(x) = \ \log\ (x)$$

$$f_3(x) = \exp(1/x)$$

$$f_4(x) = x^2$$

Upper order of the transformed equations: 3

Solution:

a) Alphabet source:

$\mathcal{U} = \{$ φ_{001} , φ_{002} , φ_{003} , φ_{004} , φ_{011} , φ_{012} , φ_{013} ,
φ_{014} , φ_{021} , φ_{022} , φ_{023} , φ_{024} , φ_{031} , φ_{032} , φ_{033} ,
φ_{034} , φ_{041} , φ_{042} , φ_{043} , φ_{044} , φ_{111} , φ_{112} , φ_{113} ,
φ_{114} , φ_{121} , φ_{122} , φ_{123} , φ_{124} , φ_{131} , φ_{132} , φ_{133} ,
φ_{134} , φ_{141} , φ_{142} , φ_{143} , φ_{144} , φ_{211} , φ_{212} , φ_{213} ,
φ_{214} , φ_{221} , φ_{222} , φ_{223} , φ_{224} , φ_{231} , φ_{232} , φ_{233} ,
φ_{234} , φ_{241} , φ_{242} , φ_{243} , φ_{244} , φ_{311} , φ_{312} , φ_{313} ,
φ_{314} , φ_{321} , φ_{322} , φ_{323} , φ_{324} , φ_{331} , φ_{332} , φ_{333} ,
φ_{334} , φ_{341} , φ_{342} , φ_{343} , φ_{344} , φ_{411} , φ_{412} , φ_{413} ,
φ_{414} , φ_{421} , φ_{422} , φ_{423} , φ_{424} , φ_{431} , φ_{432} , φ_{433} , φ_{444} $\}$

b) Alphabet code:

$$\mathcal{B} = \{f_0, f_1, f_2, f_3, f_4\}$$

c) Scheme:

$$\Sigma = \begin{cases} \varphi_{001} \to f_0 f_0 f_1 \\ \ldots\ldots\ldots. \\ \ldots\ldots\ldots. \\ \ldots\ldots\ldots. \\ \varphi_{444} \to f_4 f_4 f_4 \end{cases}$$

In the case of the word:

$$\Psi_{011003114} = \varphi_{011}\varphi_{003}\varphi_{114} \to f_0 f_1 f_1 f_0 f_0 f_3 f_1 f_1 f_4$$
$$= a_1(f_1 o f_1)(x_1) + a_2 f_3(x_2) + a_3(f_1 o f_1 o f_4)(x_3) + b =$$
$$a_1 \sin(\sin(x_1)) + a_2 \exp(1/x_2) + a_3(\sin\left(\sin\left((x_3)^2\right)\right) + b,$$
$$a_1, a_2, a_3, b \in R.$$

3. Test for Unique Decipherability

We consider alphabet coding for two alphabets $\mathcal{U}$ and $\mathcal{B}$, specified by the following scheme

$$\begin{aligned} &\phi_{\underbrace{00...0i}_{n}} \to f_0 f_0 \cdots\cdots f_0 f_i \\ &\varphi_{\underbrace{00...0ji}_{n}} \to f_0 f_0 \cdots\cdots f_0 f_i f_j \\ &\varphi_{\underbrace{abc...k}_{n}} \to f_a f_b f_c \cdots\cdots f_k \end{aligned} \quad (13)$$

It is obvious that alphabet coding generates a mapping of the set S($\mathcal{U}$) into the set S($\mathcal{B}$). We denote by $S_\Sigma(\mathcal{B})$ the image of S($\mathcal{U}$) under this mapping.

If the mapping of S($\mathcal{U}$) onto $S_\Sigma(\mathcal{B})$ is one-to-one, then decoding is possible, i.e., it is possible to uniquely reconstruct from a code B the original message with code B. We will say that alphabet is one-to-one.

The decoding procedure is as follows:

Example 2: Suppose that a word

$$a_1 \log(\sin(x_1)) + a_2(\exp(1/x_2))^2 + a_3(\sin(\sin(x_3)) + b$$

is given.

We divide the word into elementary codes and replace each one by its correspondent letter in scheme ∑:

$$a_1 \log(\sin(x_1)) + a_2(\exp(1/x_2))^2 + a_3(\sin(\sin(x_3)) + b$$
$$= a_1(f_2 o f_1)(x_1) + a_2(f_4 o f_3)(x_2) + a_3(f_1 o f_1)(x_1) + b =$$
$$f_0 f_2 f_1\ \ f_0 f_4 f_3\ \ f_0 f_1 f_1 = \varphi_{021}\varphi_{043}\varphi_{011} = \Psi_{021043011}$$

Then we observe that our alphabet coding is one-to-one and the decoding is possible.

4. Conclusions

The application of the code defined in the modeling of the flow equations, provides a simplification of storage processes of these equations. It will therefore be possible to easily compare the flow equations derived in various modeling or simulations of the same model. This code has reduced storage process of flow equations, it being possible to decode because it has been shown that verifies the unique decipherability test.

The application of the results obtained in this work will have a good tool for obtaining better mathematical models.

References

[1] Sastre-Vazquez, P., Usó-Domènech, J.L, Villacampa, Y., Mateu, J. and Salvador, P. 1999. Statistical Linguistic Laws in Ecological Models. *Cybernetics and Systems: An International Journal*. Vol 30. 8. 697-724.

[2] Sastre-Vazquez, P., Usó-Domènech, J.L. and Mateu, J. 2000. Adaptation of linguistics laws to ecological models. *Kybernetes.* 29 (9/10). 1306-1323.

[3] Usó-Domènech, J.L., Villacampa, Y., Stübing, G., Karjalainen, T. & Ramo, M.P. 1995. MARIOLA: a model for calculating the response of mediterranean bush ecosystem to climatic variations. *Ecological Modelling*. 80, 113-129.

[4] Usó-Domènech, J. L., Mateu, J and J.A. Lopez. 1997. Mathematical and Statistical formulation of an ecological model with applications. *Ecological Modelling*. 101, 27-40.

[5] Usó-Domènech, J.L. and Villacampa, Y. 2001. Semantics of Complex Structural Systems: Presentation and Representation. A synchronic vision of language L (MT). *Int. Journal of General Systems*. 30 (4). 479-501.

[6] Usó-Domènech, J.L, Sastre-Vazquez, P. Mateu, J. 2001. Syntax and First Entropic Approximation of L (MT): A Language for Ecological Modelling. *Kybernetes*. 30 (9/10). 1304-1318.

[7] Usó-Domènech, J.L and Sastre-Vazquez, P. 2002. Semantics of L (MT): A Language for Ecological Modelling. *Kybernetes* 31 (3/4), 561-576.

[8] Usó-Domènech, J.L., Vives Maciá, F. and Mateu. J.. 2006^{a}. Regular grammars of L (MT): a language for ecological systems modelling (I) –part I. *Kybernetes* 35 nº6, 837-850.

[9] Usó-Domènech, J.L., Vives Maciá, F. and Mateu. J.. 2006^{b}. Regular grammars of L (MT): a language for ecological systems modelling (II) –part II. *Kybernetes* 35 (9/10), 1137-1150.

[10] Usó-Doménech, J. L., Nescolarde-Selva, J. 2014. Disipation Functions of Flow Equations in Models of Complex Systems. *American Journal of Systems and Software*. 2 (4), pp. 101-107

[11] Usó-Doménech, J. LL., Nescolarde-Selva, J., Lloret-Climent, M. 2014a. Behaviours, Processes and Probabilistic Environmental Functions in H-Open Systems. American Journal of Systems and Software. 2 (3), pp. 65-71.

[12] Usó-Doménech, J. L., Nescolarde-Selva, J., Lloret-Climent, M. 2014b. Saint Mathew Law and Bonini Paradox in Textual Theory of Complex Models. American Journal of Systems and Software. 2 (4), pp. 89-93.

[13] Usó-Doménech, J. L., Nescolarde-Selva, J., Lloret-Climent, M. and González-Franco, L. 2014. Diversity for Texts Builds in Language L (MT): Indexes Based in Theory of Information. American Journal of Systems and Software, 2 (5). pp. 113-120

[14] Villacampa, Y., Usó-Domènech, J.L., Mateu, J. Vives, F. and Sastre, P. 1999. Generative and Recognoscitive Grammars in Ecological Models. Ecological Modelling. 117, 315-332.

[15] Villacampa, Y. and Usó-Domènech, J.L. 1999. Mathematical Models of Complex Structural systems. A Linguistic Vision. Int. Journal of General Systems. Vol 28, no 1, 37-52.

[16] Villacampa-Esteve, Y., Usó-Domènech, J.L., Castro-Lopez-M, A. and P. Sastre-Vazquez. 1999. A Text Theory of Ecological Models. Cybernetics and Systems: An International Journal. Vol 30, 7.587-607.

[17] Forrester, J.W., 1961. Industrial Dynamics. MIT Press, Cambridge, MA.

[18] Jörgensen, S.E., 1988. Fundamentals of Ecological Modelling. Developments in Environmental Modelling 9. Elsevier, Amsterdam.

[19] Abramson, N. 1981. Teoria de la Codificación y la Información. Ed Paraninfo. Madrid. (In Spanish)

[20] Davis, M.D. and Weyuker, E.J, 1983. Computability, Complexity and Languages. Academic Press.

[21] Yablosnsky, S.V. 1989. Introduction to Discrete Mathematics. Ed Mir. Moscou.