ANNALS OF "DUNAREA DE JOS" UNIVERSITY OF GALATI
MATHEMATICS, PHYSICS, THEORETICAL MECHANICS
FASCICLE II, YEAR XVI (XLVII) 2024, No. 2
DOI: https://doi.org/10.35219/ann-ugal-math-phys-mec.2024.2.05

The connection between the stochastic series and the time depending phenomena

Nina N. Cazacu*

Bucharest University of Economic Studies, Piața Romană, nr. 6, Sector 1, 010552, Romania
**Corresponding author: cazanina@yahoo.com*

Abstract
The present paper employs computer-assisted methods in order to solve different type of equations and systems. The most significant goal is to transform, by modeling, time-depending series, such as stochastic series for instance, into dynamic systems that can be solved, while also obtaining visual representations of their solutions.
So, we have started with an introduction including different computer-assisted methods for solving differential equations and systems, followed by a proposed algorithm that leads to both an analytical formula and a graphical representation of the optimal solution and command for a specific class of dynamical systems.
Through this work, we have shown that stochastic series can be expressed, through mathematical modeling, as a time-dependent system of equations or even as dynamic system, which we have established an obvious similarity with the generalised dynamic systems of the age-structured populations, discovered by J.R. Chasnov.
The connection which we have established allows for the application of the classical computer-assisted methods, as well as particular algorithms, for many other time-dependent series, which can be modeled in the dynamic system form, particularly when the main term follows a time-recurring formula.

Keywords: modeling, stochastic series, computer methods, algorithm, dynamic systems.

1. INTRODUCTION

We first present some usual computer-assisted methods for solving different type of differential equations and differential systems, which will be useful for this research. There are two ways for gaining the solutions:

When an exact result is needed, the value of the function can be calculated at any point or represented over an interval within the domain of definition.

When approximate solutions are acceptable, and the solution depends on one or more parameters, a table of parametric values and the corresponding solution values is generated.

- Algorithms for finding the **exact** solutions of the differential equations

Example No. 1

First equation to solve is:

$$\frac{dy}{dx} = y + x \text{, with } y(0)=0, \text{ y:R}\rightarrow\text{R} \qquad (1)$$

```
DSolve[{y'[x]=y[x]+x}, y[x], x]
```

It results: {{y[x]→-1-x+e^x C[1]}}

```
sol=DSolve[{y'[x]=y[x]+x, y[0]=0}, y[x], x]
```

It results: {{y[x]→-1+e^x-x}}

For the graphic representation: Plot[y[x]/.sol,{x,-3,2}, PlotStyle → {Red}];

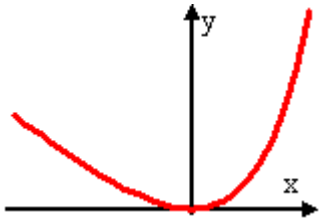

Fig. 1. Graphic representation of the exact solution for the differential equation (1)

Example No. 2

We will use application tools for solving a first-order differential equation in which the unknown to be found is a bilinear function:

$$u + 3\frac{\partial u}{\partial y} + 2\frac{\partial u}{\partial x} = 0, \text{ with: } u:R^2 \rightarrow R,\ u=u(x,y) \qquad (2)$$

z:=u[x,y]; p:=D[u[x,y],x]; q:=D[u[x,y],y]; eqn=2*p+3*q +z=0;
sol=DSolve[eqn, u, {x,y}].

It results: $u[x,y]+3\ u^{(0,1)}[x,y]+2\ u^{(1,0)}[x,y]=0$

eqn/.sol[[1]]//Simplify

It results: {{u→Function[{x,y},$e^{\frac{x}{2}}$ C[1][1/2 (-3 x+2 y)]]}}

particularsolution = u[x,y]/.sol[[1]]/.C[1][a_]→ Sin[a]

It results: $e^{-x/2}$ Sin[1/2 (-3 x+2 y)]

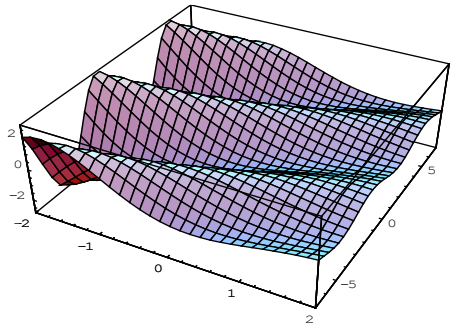

Fig. 2 Graphic representation of the exact solution for the differential equation (2)

For 3D visualization, 30 points are represented. The points' coordinates are: (x,y, $e^{-x/2}$sin[1/2(-3x+2 y)]), with x∈[-2,2], y∈[-7,7], so that the bilinear function z=u(x,y) ∈[e^{-1}sin(4), -e·sin(4)].
Plot3D[particularsolution, {x,-2,2}, {y,-7,7}, PlotPoints → 30];
The solution u[x,y] depends on a parameter, but it is replaced by the function sin(x), and it was obtained only **an exact particular solution**.

- Algorithms for finding the **approximate** solutions of the differential equations

Example No. 1

We used the differential equation (1) once again, in order to find and represent the approximate solutions, and compare it with the exact one.
eqn=y'[x]=x+y[x]; sol=DSolve[eqn,y,x]

It results: {{y→Function[{x},-1-x+e^x C[1]]}}
tab1=Table[y[x]/.sol[[1]]/.C[1]→k,{k,-3,2,0.5}];

It results in a table of values, which the computer will take account of for the graphical representation of the solutions with parameter k=C[1] in the interval [-3,2], with a step size of 0.5, and the variable x also within the same interval. The number of graphical branches is calculated as: 2-(-2)=5; 5:0.5=10. For the graphical representation(Fig.3), we used the following function:
Plot[Evaluate[Join[tab1]],{x,-3,2},PlotStyle→{Blue}];

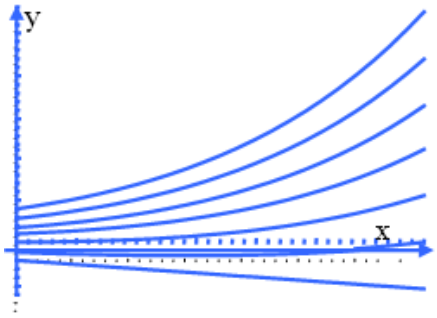

Fig. 3 Graphical representation of the approximate solution for the equation (1)

Example No. 2

The next equation we have proposed to solve using "*Mathematica*" tools is the following:

$$\frac{dy}{dx} = x^2 + y^2 \text{, with: } y=y(x),\ y:R\rightarrow R \qquad (3)$$

The solving code lines are:

eqn=y'[x]=x^2+y [x]^2; sol=DSolve[eqn,y,x]

It results:

{{y→Function[{x},(-BesselJ[-(1/4), $\frac{x^2}{2}$]C[1]+x²(-2BesselJ[-(3/4), $\frac{x^2}{2}$]-BesselJ,[-(5/4) $\frac{x^2}{2}$]C[1]+

BesselJ[3/4, $\frac{x^2}{2}$]C[1]))/

(2x(BesselJ[1/4, $\frac{x^2}{2}$]+ BesselJ[-(1/4), $\frac{x^2}{2}$]C[1]))]}}

tab1=Table[y[x]/.sol[[1]]/.C[1]→ k, {k,0,3, 0.5}];

For this equation, the approximate solutions are given by the tabled values of the calculated solution sol[[1]] for each parameter k=C[1], where the values of *k* increase from 0 to 3, with a step size of 0.5.

For the graphical visualization, the values of the variable y were calculated , with x∈[0,1.7] . The graphs for each parameter value have similar shapes(Fig. 4).

Plot[Evaluate[Join[tab1]], {x,0,1.7}, PlotStyle → {Blue}];

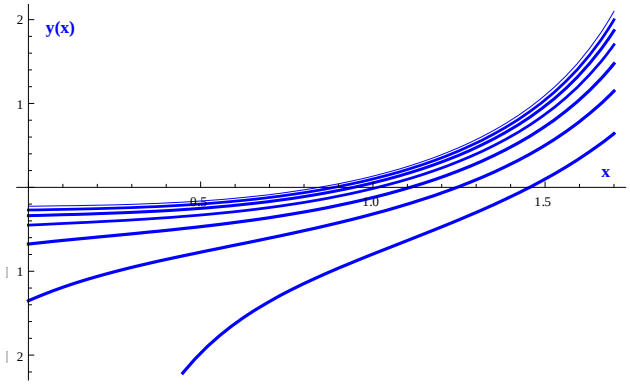

Fig. 4 Graphic representation of the approximate solution for the equation (3)

- Computer methods for finding the solutions of the differential systems

a) First method refers to the **eigenvalues** of the system matrix, noted with A. We have considered, for exemplification, the following system of differential equations:

$$\frac{dX}{dt} = \begin{pmatrix} 1 & -1 \\ -1 & -3 \end{pmatrix} \cdot \begin{pmatrix} x(t) \\ y(t) \end{pmatrix}; X = \{x(t), y(t)\} \qquad (4)$$

A={{1,-1},{-1,-3}}; Eigenvalues[A]

It results: {{1,-1},{-1,-3}}; {-1-$\sqrt{5}$,-1+$\sqrt{5}$ }

X[t] = {x[t], y[t]}; system = MapThread[#1=#2&, {X'[t],A.X[t]}]

It results: {x'[t]=x[t]-y[t],y'[t]=-x[t]-3 y[t]}

sol=DSolve[system, {x,y},t]

The solution includes two parameters, which we have associated with two sets of values and represent over an established interval. The eigenvalues method leads to the following solution with two components, x(t) and y(t) and two parameters C[1] and C[2]:

$$\{\{x \to \text{Function}[\{t\}, \frac{1}{10}\left(5e^{(-1-\sqrt{5})t} - 2\sqrt{5}e^{(-1-\sqrt{5})t} + 5e^{(-1+\sqrt{5})t} + 2\sqrt{5}e^{(-1+\sqrt{5})t}\right)C[1] + \frac{\left(e^{(-1-\sqrt{5})t} - e^{(-1+\sqrt{5})t}\right)C[2]}{2\sqrt{5}}],$$

$$y \to \text{Function}[\{t\}, \frac{\left(e^{(-1-\sqrt{5})t} - e^{(-1+\sqrt{5})t}\right)C[1]}{2\sqrt{5}} + \frac{1}{10}\left(5e^{(-1-\sqrt{5})t} + 2\sqrt{5}e^{(-1-\sqrt{5})t} + 5e^{(-1+\sqrt{5})t} - 2\sqrt{5}e^{(-1+\sqrt{5})t}\right)C[2]]\}\}$$

As a consequence, this results in the corresponding particular solutions, whose images were represented on the interval [-1.5,2], for parameters values between [-1,2]. The image is then transformed to the corresponding scale(Fig. 5).

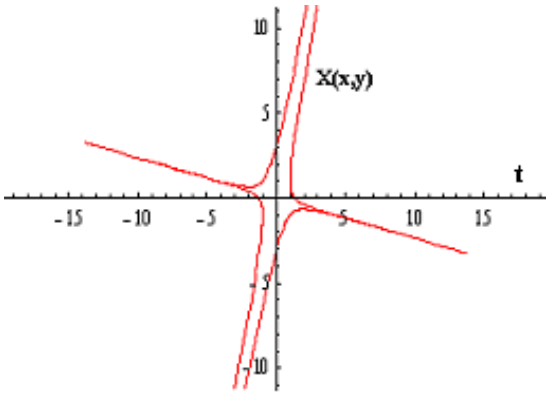

Fig. 5 Graphic representation of the solution for the differential system (4), based on the eigenvalues method

The parameters C[1] and C[2] had two sets of values, chosen from an arbitrarily established interval, with a fixed step. The last two lines of code led to the graphical visualization of the solution:
particularsols=Partition[Flatten[Table[{x[t], y[t]}/.sol/.{C[1] → 1/i, C[2]→ 1/j}, {i,-1,2, 2}, {j,-1,2, 2}]], 2] ; ParametricPlot[Evaluate[particularsols], {t,-1.5,2}, PlotStyle → {{Red}}];

b) The second method used **interpolation** to solve a differential system of two equations, which can also be written in matrix form, as follows:

$$\frac{dx}{dt} = \text{matA} \cdot x \Rightarrow \frac{dx}{dt} = \begin{pmatrix} 0 & -\frac{1}{3} \\ -\frac{1}{3} & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}; x(t_0) = x(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad (5)$$

This method can be applied to any differential system, with respect to the initial conditions. In particular, by solving the system using the "Wolfram Mathematica" program, we were led to the representation in Fig.6.
The code is as follows:

matA=N[{{0,-1/3},{-1/3),0}}]; DAE=$\begin{pmatrix} x_1[t] = -1/3 \cdot x_2[t] \\ x_2[t] = -1/3 \cdot x_1[t] \end{pmatrix}$; x(0) = {1,0}

sol = NDSolve[{DAE,$x_1[0]=1, x_2[0]=0$},{x_1,x_2}, {t,0,1}];

It results: {{x_1→InterpolatingFunction[{{0.,1.}}, <>],x_2→ InterpolatingFunction[{{0.,1.}},<>]}}

The interpolation is developed over the time interval [0,1]. The solution is not approximate, it has to satisfy the initial condition.

For the graphical representation, classic instructions were used for this purpose, so the graphical result had two branches(Fig. 6):
Plot[Evaluate[{x_1[t],x_2[t]}/.First[sol]],{t,0,1};
PlotStyle→{{RGBColor[1,0,0]},{RGBColor [0,0,0]} }];

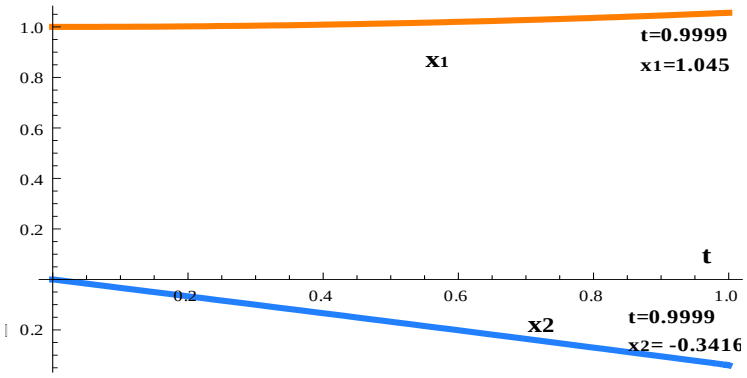

Fig. 6 Graphic representation of the solution for the differential system (5), based on interpolation

c) The third method uses a **proposed algorithm**, also written in the "Wolfram Mathematica" application, based on a semi-inverse method for determining the analytical solution of a class of particular dynamic systems, along with the graphical representation of the solution trajectory([5,1]). Thus, for any dynamic system (Σ) with square constant matrix coefficients the analytical formula for the optimal solution trajectory and command can be determined.

The matrices we have considered for this purpose verify the specific restrictions: $\det(A)\neq 0_n$, $\det(A+A^*)\neq 0_n$, A, B, $C\in M_n(R)$, $X=Y=R^n$, so the algorithm which emerges from the mathematical method can be used(it was noted: A*=Transpose(A)).
The dynamic system we are going to study has the following form:

$$(\Sigma)\quad \begin{aligned} &\frac{dx}{dt} = A\cdot x(t) + B\cdot u(t) \\ &y(t) = C\cdot x(t), x_0 = x(t_0) \end{aligned} \tag{6}$$

with: $x\in X$, the state vector, X= the state space, $y\in Y$, the output vector, Y= the output space, $u\in U$, the command, U= the admissible commands, $(T1,T2)\subset R$ the time interval, $t\in (t_0,t_1)=(0,1)\subset(T1, T2)$ the time variable.

To apply the method, we have particularized the system coefficients, A-the state system matrix, B-the disturbing matrix and C-the output system matrix, for **n=2**. The time interval is [0,1] and the specific restrictions are verified. The initial condition for the system solution was particularized in two-dimensional case.

$$\begin{aligned} &\frac{dx}{dt} = \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix}\cdot\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & c \end{pmatrix}\cdot\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \\ &\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} e & 0 \\ 0 & f \end{pmatrix}\cdot\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}; x(t_0) = x_i = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \end{aligned} \tag{6'}$$

The algorithm, which uses the semi-inverse method adapted to the dynamic system (6'), with x=optimal trajectory, u0=optimal command, written with help of "Wolfram Mathematica" application, is displayed as follows:

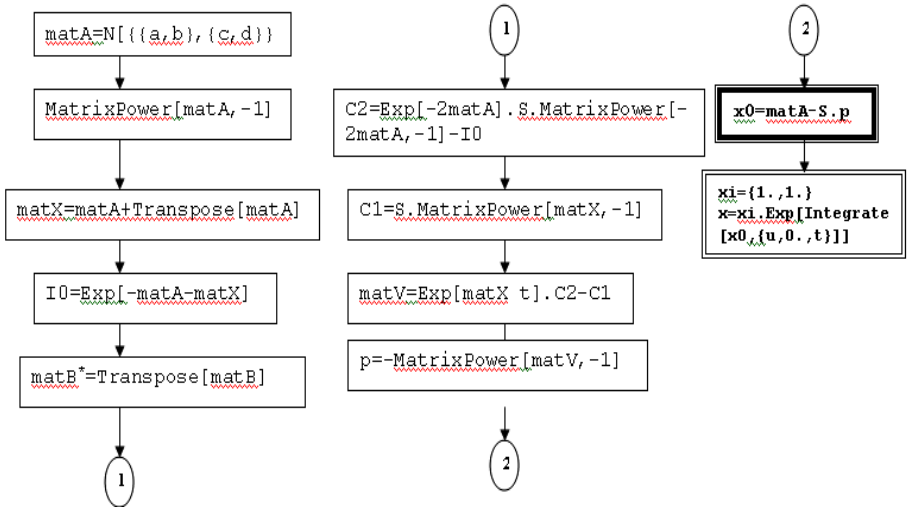

Fig. 7 The algorithm semi-inverse scheme

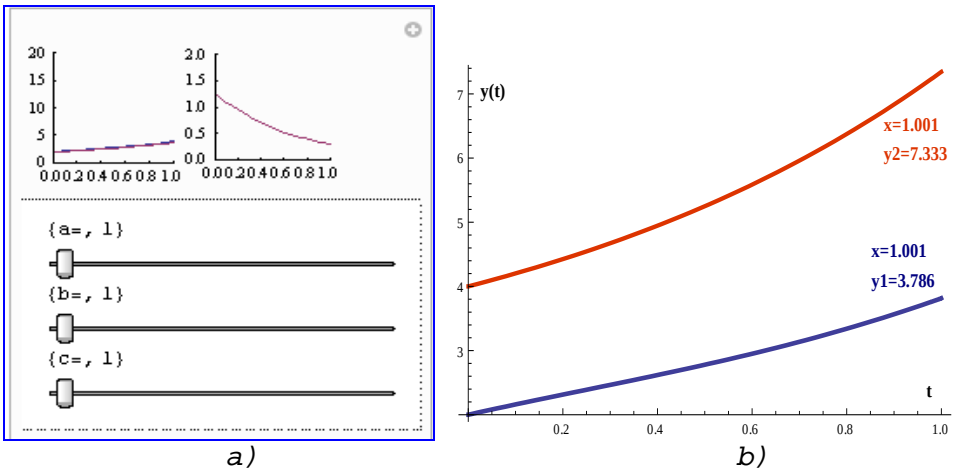

*Fig. 8 Graphic representation of the system (6') solution x(t) and optimal command u(t)(**a**); the system (6') output y(t) trajectories(**b**)*

The graphical commands for obtaining the solution trajectory are the following:
Manipulate[Column[{
{"a=",a},Slider[Dynamic[a],{1,40,1}],{"b=",b},Slider[Dynamic[b],{1,10,1}],{"c=",c},Slider[Dynamic[c],{1,20,1}]}], Delimiter,
Row[{Dynamic[Plot[{x},{t,0,20},AspectRatio→1,PlotRange→{{0,1},{0,20}}],SynchronousUpdating→True],
Dynamic[Plot[{u0},{t,0,2},AspectRatio→1,PlotRange→{{0,1},{0,2}}],SynchronousUpdating→True] }]], with the result is described in Fig. 8.

2. DEVELOPING THE RESEARCH

A. Modeling the stochastic series as differential dynamic systems

The formula of the Stochastic series, highlighting the recurrent connection of terms, also the time dependence is the following:

$$x_t = ax_{t-1} + b + u_t \tag{7}$$

with: a, b scalars, u_t=the disturbing factor, $\bar{u}_t = 0$. It is obvious that the media $\bar{x}_t$ is the solution of the determinist equation: $\bar{x}_t = a\bar{x}_{t-1} + b$.

One can express x_t as a function depending on x_0 and u_t, by applying successive substitutions in the right-hand side of the equation. This leads to the following formulas:

$$x_t = u_t + a^t x_0 + b\frac{1-a^t}{1-a} \Rightarrow \bar{x}_t = a^t x_0 + b\frac{1-a^t}{1-a}$$

We proceeded in a similar way also for a series of series. This time, the constant coefficients a_i were replaced by the coefficients A_i, which are matrices. For the media calculus, two stages must be followed:

a) Repeated elimination of delays, which leads to the following formula for xt:

$$x_t = A(Ax_{t-2} + b + u_t) + b + u_t = \ldots = A^t x_0 + b\frac{I-A^t}{I-A} + u_t\frac{I-A^t}{I-A}(\bar{u}_t = 0)$$

b) Starting from the form: $x_t = Ax_{t-1} + b + u_t$, it moves to the averages and obtains: $\bar{x}_t = A\bar{x}_{t-1} + b(\bar{u}_t = 0)$.

We have solved the equation from sequence a) and passed to the media. It has resulted: $\bar{x}_t = A(A\bar{x}_{t-2} + b) + b\ldots = A^t \cdot x_0 + b\frac{I-A^t}{I-A}$

very similar to the one previously obtained for a single series. From here, all properties of a series can be generalised for a number of dynamic series.

Let's consider a vector x_t of series terms:

$$x_t = A_1 \cdot x_{t-1} + A_2 \cdot x_{t-2} + \ldots + A_m \cdot x_{t-m} + b + u_t \qquad (7')$$

The equation (7') can be written in the *form of a matrix system*:

$$\begin{bmatrix} x_t \\ x_{t-1} \\ \vdots \\ x_{t-m+1} \end{bmatrix} = \begin{bmatrix} A_1 & A_2 & \ldots & A_{m-1} & A_m \\ 0 & A_2 & \ldots & \ldots & A_m \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & A_{m-1} & \ldots \\ 0 & \ldots & \ldots & \ldots & A_m \end{bmatrix} \cdot \begin{bmatrix} x_{t-1} \\ x_{t-2} \\ \vdots \\ x_{t-m=1} \\ x_{t-m} \end{bmatrix} + \begin{bmatrix} b+u_t \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad (8)$$

We have decreased in both members with vector $x=[x_{t-1},x_{t-2},\ldots,x_{t-m-1},x_{t-m}]$, and obtained the system in equations (9):

$$\begin{bmatrix} x_t - x_{t-1} \\ x_{t-1} - x_{t-2} \\ \vdots \\ x_{t-m+1} - x_{t-m} \end{bmatrix} = \begin{bmatrix} A_1 - 1 & A_2 & \ldots & A_{m-1} & A_m \\ 0 & A_2 - 1 & \ldots & \ldots & A_m \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & A_{m-1} - 1 & \ldots \\ 0 & \ldots & \ldots & \ldots & A_m - 1 \end{bmatrix} \cdot \begin{bmatrix} x_{t-1} \\ x_{t-2} \\ \vdots \\ x_{t-m-1} \\ x_{t-m} \end{bmatrix} + \begin{bmatrix} b+u_t \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad (9)$$

For simplifying, we assumed that the determinants A_i are of minimum dimensions: $a_{i,\ i=1,2,3,\ldots,p}$. For the beginning, we studied a single Stochastic series.

We have noted: $x_{t-k} \rightarrow x_k$, $_{k=1,2,\ldots,m}$; $m=p$; $u_t+b=\omega_t$

Taking account of these conventions, system (9) becomes:

$$\begin{bmatrix} dx_1 \\ dx_2 \\ \vdots \\ dx_p \end{bmatrix} = \begin{bmatrix} a_1 - 1 & a_2 & \ldots & a_{p-1} & a_p \\ 0 & a_2 - 1 & \ldots & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & a_{p-1} - 1 & \ldots \\ 0 & \ldots & \ldots & \ldots & a_p - 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{p-1} \\ x_p \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \omega \\ \omega \\ \omega \\ \omega \end{bmatrix} \qquad (10)$$

which, after adding an output vector y(t)=matC·x(t), it was led to a dynamic system.

The vectorial variables x, y and ω are dependent on time t, and have p components.

$$\begin{pmatrix}\dot{x}_1\\ \dot{x}_2\\ \vdots\\ \dot{x}_p\end{pmatrix}=\begin{pmatrix}s_1m_1 & s_2m_2 & \dots & s_{p-1}m_{p-1} & s_pm_p\\ s_2 & -1 & \dots & 0 & 0\\ 0 & s_3 & \dots & 0 & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \dots & s_p & -1\end{pmatrix}\cdot\begin{pmatrix}x_1\\ x_2\\ \vdots\\ x_p\end{pmatrix}+\begin{pmatrix}\alpha_1 & 0 & \dots & 0\\ 0 & \alpha_2 & \vdots & 0\\ 0 & 0 & \dots & 0\\ \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \dots & \alpha_p\end{pmatrix}\cdot\begin{pmatrix}\omega\\ \omega\\ \vdots\\ \omega\end{pmatrix} \quad (11)$$

$$\begin{pmatrix}y_1\\ y_2\\ \vdots\\ y_p\end{pmatrix}=\begin{pmatrix}a & 0 & \dots & 0\\ 0 & b & \vdots & 0\\ 0 & 0 & \vdots & 0\\ \vdots & \vdots & \vdots & \vdots\\ 0 & 0 & \dots & c\end{pmatrix}\cdot\begin{pmatrix}x_1\\ x_2\\ \vdots\\ x_p\end{pmatrix};\ x_0=x(0)=\begin{pmatrix}1\\ 0\\ \vdots\\ 0\end{pmatrix}$$

System (10) can be considered as a particular case of the generalized form in equations (11), which describes, within a similar dynamic system, the evolution of aged-structured populations([2,3]), generalised for **p** classes and **n** generations[1], with **n=p**.

B. Solving the Stochastic system using computer assisted methods

Like any other differential system, Stochastic systems can be solved using classic computer-assisted methods, as well as specific proposed algorithms, if available.

a) We considered some particular parameters values in the mathematical context: s_1=0.75; m_1=2, s_2=0.8; m_2=2, s_3=0.6; m_3=1, $a_3=s_3\times m_3$=0.6, $a_2=s_2\times m_2$=1.6, $a_1=s_1\times m_1$=1.5. As a consequence: a_1-1=0.5, a_2-1=0.6, a_3=-0.4. We started from these real values, but the algorithm can be applied to many other values.

Thus, it was considered that the system matrix has elements based on the values of a_1, a_2, and a_3 , multiplied by an arbitrary scalar(α=**2**):

$$A_1=\begin{pmatrix}0.5 & 1.6 & 0.6\\ 0. & 0.6 & 0.\\ 0. & 0. & -0.4\end{pmatrix},\ A=2\cdot A_1=\begin{pmatrix}1 & 3.2 & 1.2\\ 0. & 1.2 & 0.\\ 0. & 0. & -0.8\end{pmatrix}$$

Using the **classic eigenvalues** algorithm, the first method presented in the previous paragraph, we were led to the differential system to solve, with the following form:

$$\frac{dX}{dt}=A\cdot X=\begin{pmatrix}1 & 3.2 & 1.2\\ 0 & 1.2 & 0\\ 0 & 0 & -0.8\end{pmatrix}\cdot\begin{pmatrix}x\\ y\\ z\end{pmatrix} \quad (12)$$

The corresponding code in the "Mathematica" application led to:

A={{1, 3.2, 1.2},{0,1.2,0}, {0,0,-0.8}}; Eigenvalues[A];It results: {1.2,1.,-0.8}

X[t] = {x[t], y[t],z[t]}; system = MapThread[#1=#2&,{X'[t],A.X[t]}]

It results: {x'[t]=x[t]+3.2 y[t]+1.2 z[t],y'[t]=1.2 y[t],z'[t]=-0.8 z[t]}

sol=DSolve[system, {x,y,z},t]

The result can be tabled and represented in the graphical mode, using next code:

particularsols=Partition[Flatten[Table[{x[t], y[t],z[t]}/.sol/.{C[1] → 1/i, C[2]→ 1/j, C[3]→ 1/k}, {i,-1,20, 6}, {j,-1,20, 6}, {k,-1,20, 6}]], 2]

ParametricPlot[Evaluate[particularsols], {t,0.001,6}, PlotStyle → {{Red}}]

It can be observed that when the graphic trajectories were plotted(Fig. 9), the three parameters had real values, and the time interval for the graphical representation was established. The solutions resulted from interpolation, with the three parameters taking ascending values between -1 and 20, over the time interval [0.001, 6]. The trajectory showed an ascending evolution, and the same behavior was observed also for greater intervals of time.

[1] The parameters s_i, m_i are introduced by Chasnov: s_i = female survivors in class i, m_i= number of expected births per one female.

The solution of the system follows an upward path; however, we cannot determine the path of the *optimal solution*. This can be solved with another procedure: the semi-inverse method algorithm.

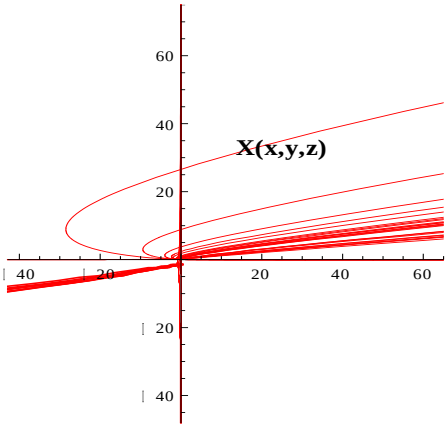

Fig.9 Graphic representation of the system (11) solution

The components of the solution are x(t),y(t) and z(t):
{{x→Function[{t},$e^{-0.8t}$ (3.70074×10^{-17}+1. $e^{1.8t}$+6.82831×10^{-32} $e^{2.t}$) C[1]+$e^{-0.8\,t}$ (6.24267×10^{-16}-16. $e^{1.8t}$+16. $e^{2.t}$) C[2]+$e^{-0.8t}$ (-0.666667+0.666667 $e^{1.8t}$-1.23008×10^{-15} $e^{2.t}$) C[3]],
y→Function[{t},$e^{-0.8t}$(-4.26769×10^{-33}+4.26769×$10^{-33}$$e^{2.t}$)C[1]+$e^{-0.8t}$ (-7.19904×10^{-32}+1. $e^{2.t}$) C[2]+$e^{-0.8t}$ (7.68799×10^{-17}-7.68799×10^{-17} $e^{2.t}$) C[3]],
z→Function[{t},e-$^{0.8t}$(-5.55112×10^{-17}+5.55112×$10^{-17}$$e^{1.8t}$+7.78674×$10^{-48}$$e^{2.t}$) C[1]+e-$^{0.8t}$ (-9.36401×10^{-16}-8.88178×10^{-16} $e^{1.8t}$+1.82458×10^{-15} $e^{2.t}$)C[2]+$e^{-0.8t}$(1. +3.70074×10^{-17} $e^{1.8t}$-1.40273×10^{-31} $e^{2.t}$) C[3]]}}

b) Solving the problem with help of the semi-inverse method algorithm
We continued to follow the similarity with the Chasnov generalized biological model for age-structured populations([2,3]) and applied the proposed algorithm to the system (10). The same matrix A was used for better comparison with the emerged solutions. The new dynamic system was as follows:

$$\frac{dx}{dt}=\begin{pmatrix}1 & 3.2 & 1.2\\ 0 & 1.2 & 0\\ 0 & 0 & -0.8\end{pmatrix}\cdot\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix}+\begin{pmatrix}0 & 0 & 0\\ 0 & 0 & 0.\\ 0. & 0. & 1\end{pmatrix}\cdot\begin{pmatrix}u\\ u\\ u\end{pmatrix} \qquad (13)$$

$$y=\begin{pmatrix}1 & 0 & 0\\ 0 & 0 & 0.\\ 0. & 0. & 0\end{pmatrix}\cdot\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix};x(0)=x_i=\begin{pmatrix}1\\ 0\\ 0\end{pmatrix}$$

In equation (13), matrix B has only the last element different from zero, because we inverted the index order to maintain the similarity with Chasnov system. By applying the same algorithm in the three-dimensional case, we obtained the following results.
The calculated coefficients, for the involved state variable had the following values:
C2={{-0.115502,0.222048,-0.221496},{-0.508018,0.759031,-0.0365854},
{-2.41693,4.62417,5.88894}};
C1={{0.,0.,0.},{0.,0.,0.},{-0.54878,0.731707,-1.03659}}
The following matrices resulted from applying the specific algorithm:

$$X=A+A^*=\begin{pmatrix}2 & 2.2 & 2.2\\ 3.2 & 2.4 & 0.\\ 1.2 & 0. & -1.6\end{pmatrix};\ I0=-e^{-2A-A^*}=\begin{pmatrix}-0.05 & -0.002 & -0.09\\ -0.04 & -0.027 & -1.\\ -0.3 & -1. & -11.\end{pmatrix}$$

Also, for the graphical visualization, in order to better understand the trajectory behavior, ascending type intervals were selected, such as: x∈[-1,1], x∈[0,1] and x∈[0.001,6] (Fig. 10).

The code which has been used for the visual representations:
Plot[x,{t,-1,1},AxesLabel→{"t","x(t)"},LabelStyle→Directive[Blue,Bold],
Plot[x,{t,0,1},AxesLabel→{"t","x(t)"},LabelStyle→Directive[Blue,Bold]],
Plot[x,{t,0.001,6},AxesLabel→{"t","x(t)"},LabelStyle→Directive[Blue,Bold]]

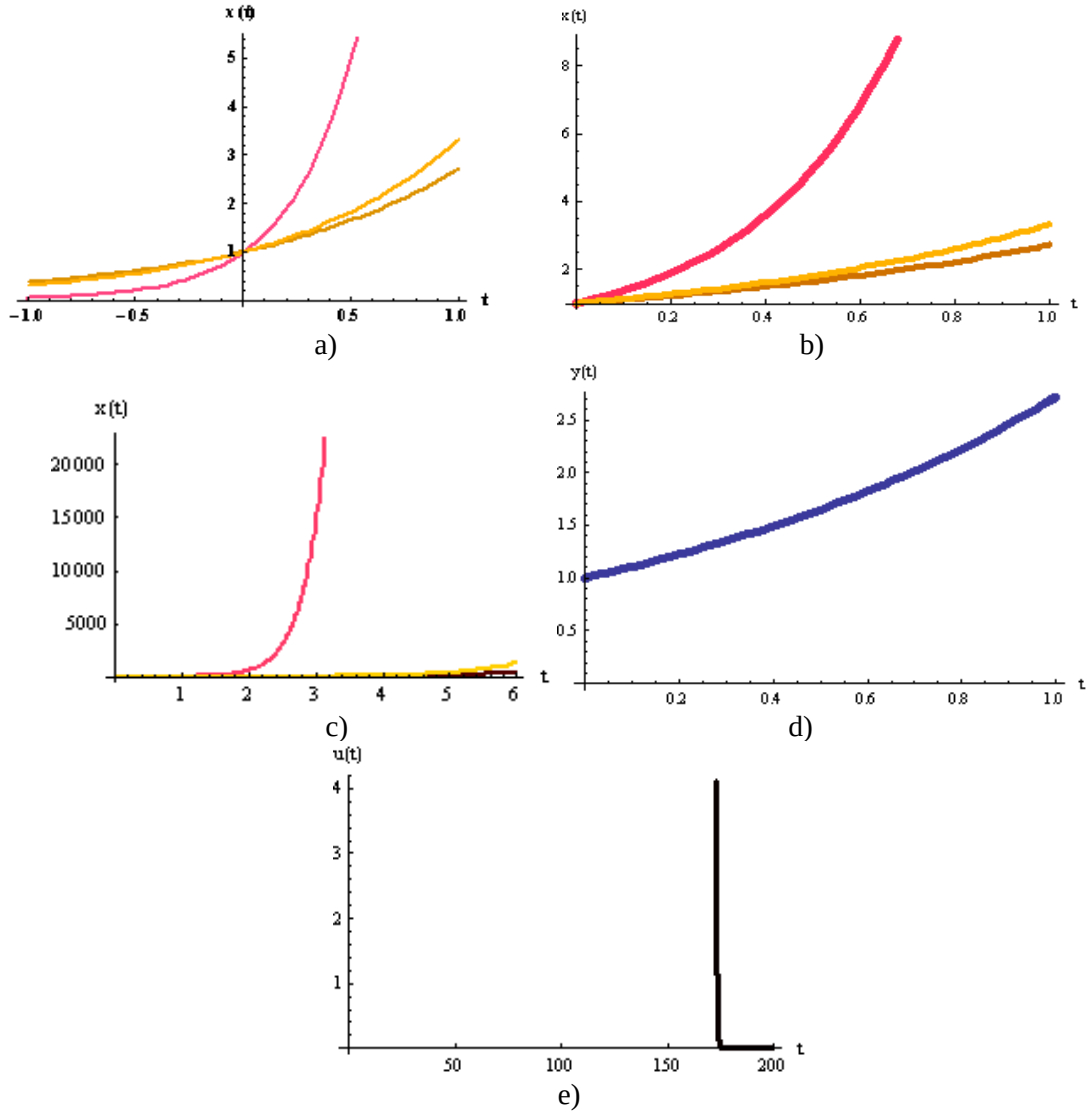

Fig.10 Graphic representation of the system (13) solution trajectory, on [-1,1], [0,1],[0.001,6] in a),b),c), system output in d) and optimal command in e)

The system output $y(t)=(y_1(t),y_2(t),\ldots,y_p(t))$, has a trajectory similar to the optimal solution. If only the first component of the system output were followed, then the image in Fig. 10d would be obtained (for the time interval [0,1]). The solution trajectory presented in Fig. 10 is ascending, though not identical to the one in Fig. 9. The distinction lies in the method in which they have been obtained: in the case presented in Fig. 9, the interpolation method leads to several particular trajectories, on similar directions, whereas in Fig. 10, only one trajectory branch **of interest** is represented(which has three components of its own).

As for the optimal command, the graphical visualization became visible only after 150 units of time. From that moment, it followed a descending trajectory(Fig.10e), indicating that no further external impulse was needed.

RESULTS AND DISCUSSION

The present paper had the purpose of modeling the Stochastic series in the form of dynamic systems, and, as consequence, a lot of research possibilities opened up.

1) One of these possibilities was the study of the similarity between the Stochastic series and the evolution of age-structured populations. The solution trajectory followed an ascending path, regardless of the applied solving method, which means that the involved variable exhibited continuous growth. In our case, this involved the adult population and its corresponding main term x_t in the Stochastic series.

2) In the social life, a practical example could be the old well-known strategy called Martingale, a class of betting strategies originating from France, the eighteenth century, which content could be presented like this: "*one man who bets would have won if, at throwing a coin, it would come to the head drawing, and would have lost otherwise. After each loss, the bet had to be doubled, and at the first profit the previous losses would be recovered plus a profit equal to the initial stake. The condition was that the one who bets to have an infinite fortune, which is impossible, and not imposed a limit on the money earned at a bet*."

Considering the initial stake equal to 1, this corresponds to the initial conditions of a dynamic system associated with the phenomenon. Then, supposing the man has no luck and the bet must be doubled, the state matrix A would have only negative numbers on the first diagonal, corresponding to powers of 2. The disturbing matrix B changes the system's state starting from the last throw, so that the betting could continue. The resulting possible model of the phenomenon in this case could be represented by the following equations, assuming <u>three</u> unlucky throws:

$$\frac{dx}{dt} = \begin{pmatrix} -1 & -2 & -4 \\ 0 & -2 & 0 \\ 0 & 0 & -4 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0. \\ 0. & 0. & 1 \end{pmatrix} \cdot \begin{pmatrix} u \\ u \\ u \end{pmatrix}$$
$$y = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0. \\ 0. & 0. & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} ; x(0) = x_i = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad (14)$$

The state matrix elements have different values but the same arrangement as in system (13), because the three throws, occurring in consecutive ascending time intervals, are similar to three consecutive terms of a Stochastic series. The main difference lies in the meaning of these elements. In system (13), the elements followed the significance in the Chasnov model, representing the age categories of a structured population. In the system (14), the terms represent the bet, which is doubled each time, reflecting the increasing losses for the person who bets. These last elements have to be negative, so that the system output represents the total loss after the third throw, with respect to the initial conditions and the mathematical context.

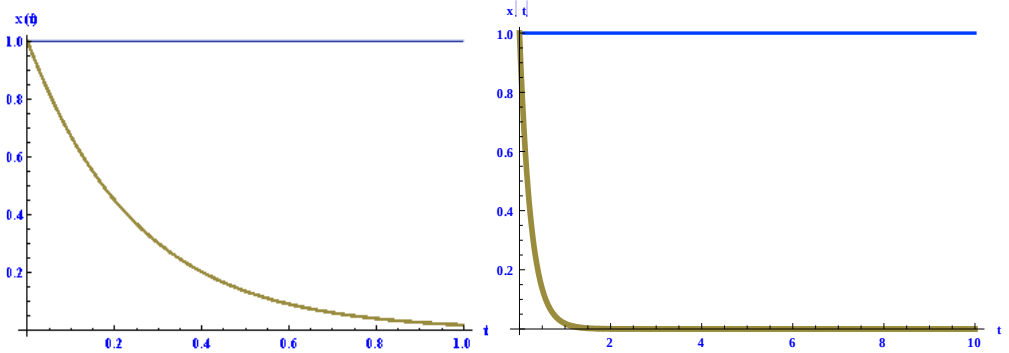

Fig.11 Graphic representation of the system (14) solution trajectory, on the time interval [0,1] and [0,10], in order, from left to right

From the graphical representation in Fig. 11, it can be seen that the initial stake, which is equal to 1, remains unchanged because the person who is betting has not won yet.

The main component of the system's output trajectory is descending, starting from the middle of any time interval considered (Fig. 12). The system output represents the betting result, which is verified as descending, as expected in this context.

There are many other time-dependent series, which can be modeled in the state form of a dynamic system (presented in formula no.7), particularly those with a time-recurring formula for the main term, (without considering the constant b):

$$x_t = ax_{t-1} + \omega_t .$$

This is a first-order recurrent formula. By subtracting x_{t-1} from both sides of the equation, it leads to the next relation: $x_t - x_{t-1} = (a-1)x_{t-1} + \omega_t$, t=1, 2, 3,…. Simplifying, it results: $dx_k = (a-1)x_k + \omega_k$, k=0,1,2,..., n(dt=1), which finally leads to the system approach.

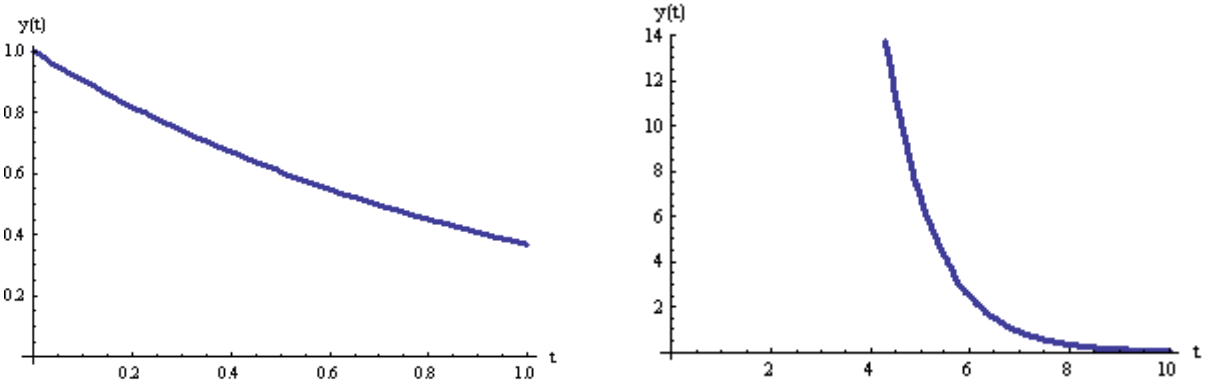

Fig.12 Graphic representation of the system (14) output trajectory, on the time intervals [0,1] and [0,10], in order, from left to right

The method of transforming the Stochastic-type series, representing time-dependent phenomena, into dynamic systems was completed using computer-assisted algorithms, which emerged from the application of "Wolfram Mathematica" or were proposed by the author. Many other phenomena could be analyzed, based on the same modeling procedure, establishing connections between mathematical notions and fields such as economics, society, biology, strategy and so on ([1,4]).

CONCLUSIONS

This paper provides both a theoretical and practical approach for studying a large number of phenomena by means of mathematical modeling and computer-assisted methods for solving differential equations or systems. We used the "Wolfram Mathematica" application to solve different samples of differential equations and systems, applying both classical and mathematical algorithms.

The connection between social phenomena(such as age-structured population evolution) and time-dependent series was analyzed. Both of these can be modeled and studied as dynamic systems.

Any time-dependent series of elements which verify a specific formula of recurrence can be transformed and treated as a dynamic system, leading to a better understanding of its future behavior.

We have mentioned and used the first order recurrence time-dependent series, with the most representative example being the Stochastic-type series. This led us to the first-order differential systems, which were transformed into dynamic systems by adding the output system equation and the initial conditions.

The proposed algorithmic method([1,3,5] used for solving dynamic systems can be applied to any pair of square matrices A and B, particularly constant, symmetric, positive-definite matrices that satisfy the specific restrictions requested in system (6). This method can also be applied to any initial data where $n \geq 2$, $n \in N$.

References

1. Cazacu, A.N., Systemic Approach of the Consumer Behavior, Management & Marketing, XV, 1, pp.118-125, 2016.
2. Chasnov, J. R., *Mathematical Biology*, Hong Kong: Science and Technology University of Hong Kong, 2009
3. Drăgoescu (Cazacu), N., Study of the age-structured populations, modeled as dynamic systems, *"Politehnica" University Scientific Paper*, 55(69), 2, Series: Mathematics-Physics, pp.53-59, 2010
4. Murray, J.D., *Biomathematics Texts: Mathematical Biology*, Berlin, Heidelberg, New York, Barcelon, Budapest, Hong Kong, London, Milan, Paris, Tokyo: Springer-Verlag, 1993
5. Kalman, R.E., Falb, P.L., Arbib, M.A., *Dynamic Systems*, Bucharest :Technical Publishing House, 1969