International Journal of Mathematics and
Computer Science, **16**(2021), no. 2, 613–638

On a numerical approach for the powers of the doubly Leslie and doubly companion matrices with applications

Amal Aloui[1], **Mustapha Rachidi**[2], **Bouazza El Wahbi**[1]

[1]Département de Mathématiques et Informatique,
Faculté des Sciences, Université Ibn Tofail,
Kénitra, Morocco

[2]Institute of Mathematics
INMA, Federal University of Mato Grosso do Sul
UFMS, Campo Grande, MS 79070-900, Brazil

email: amal.alaoui@uit.ac.ma, mu.rachidi@gmail.com,
bouazza.elwahbi@uit.ac.ma

(Received September 30, 2020, Accepted October 28, 2020)

Abstract

This study aims to provide some explicit formulas for the entries of powers of doubly Leslie and doubly companion matrices, by using various methods, based on the recursive, the analytic, and the derivative formula of some special linear difference equations. For these methods, three numerical algorithms are proposed for computing entries of powers of doubly Leslie matrices. Moreover, the mean elapsed time of these three algorithms is compared. Furthermore, illustrative applications are given. Finally, illustrative numerical examples are furnished.

Key words and phrases: Doubly Leslie matrix, Entries of the powers of doubly Leslie matrix, Linear difference equations, Companion matrix.
AMS (MOS) Subject Classifications: 15A57, 34K60.
ISSN 1814-0432, 2021, http://ijmcs.future-in-tech.net

1 Introduction

Recently, some studies are devoted to the so-called doubly Leslie matrices, defined by,

$$\mathbf{L} = \mathbf{L}([a_i],[s_j],[b_j]) = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 & \cdots & a_{r-1} \\ s_0 & 0 & 0 & 0 & \cdots & b_{r-2} \\ 0 & s_1 & 0 & 0 & \cdots & b_{r-3} \\ \vdots & \cdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & s_{r-2} & b_0 \end{bmatrix}, \tag{1.1}$$

where b_j $(0 \leq j \leq r-2)$, a_j $(0 \leq j \leq r-1)$ are in $\mathbb{R}$ and $0 < s_j \leq 1$ $(0 \leq j \leq r-2)$. The properties of these matrices have been studied in [16], where a special form of a Schur complement has been explored to obtain the determinant, inverse, and explicit eigenvector formulas of this type of matrices. In [4], the doubly Leslie matrices are constructed from the maximal eigenvalues of its all leading principal sub-matrices. Moreover, the doubly Leslie matrices viewed as Lefkovitch matrices ([8]), can be considered as an extension of the usual Leslie matrices $\mathbf{L} = \mathbf{L}([a_i],[s_j]]) = \mathbf{L}([a_i],[s_j],[0])$, where $0 \leq b_j \leq 1$ $(0 \leq j \leq r-2)$ and $0 < s_j \leq 1$ $(0 \leq j \leq r-2)$, whereas a_j $(0 \leq j \leq r-1)$ takes values in $\mathbb{R}^+$ (see, for example, [14, 11]). On the other side, the doubly Leslie matrices can be viewed as the doubly companion matrices $\mathbf{C} = \mathbf{L}([a_i],[1],[b_j])$, which have been also used as a fundamental tool to analyze some important numerical methods in [5].

The present paper concerns a computational and a numerical approach for studying the powers of the doubly Leslie and the doubly companion matrices. More precisely, we establish some compact formulas for the entries of the powers of the doubly Leslie matrices, using some properties of the linear difference equation of constant coefficients, defined by,

$$v_{n+1} = \gamma_0 v_n + \gamma_1 v_{n-1} + \cdots + \gamma_{r-1} v_{n-r+1}, \quad \text{for} \quad n \geq r-1, \tag{1.2}$$

where $\gamma_0, \gamma_1, ..., \gamma_{r-1}$ are constant coefficients and $v_0, v_1, ..., v_{r-1}$ are the initial conditions (see [3], [10]). Our formulas are based on three methods for expressing the general terms v_n given by (1.2), namely, the recursive, the analytic, and the derivative method (respectively). Moreover, three numerical algorithms, for computing the entries of the powers of the doubly Leslie matrices, are proposed. These algorithms can be applied for providing the entries of the powers of the doubly companion matrices.

The remainder of the paper is organized as follows. In section 2, we establish explicit formulas for the entries of the powers of the doubly Leslie matrices (1.1). For this purpose, three methods derived from properties of the linear difference equations (1.2) are used. Namely, we used the recursive, the analytic, and the derivative methods. Section 3 is devoted to the powers of the doubly companion matrices. In section 4, an application to dynamical populations is given, namely, some compact expressions for the entries of the vector of the population dynamics are provided. In section 5, we propose three algorithms for computing the entries of powers of the doubly Leslie matrices. Discussion and comparison with the literature are provided in Section 6. Finally, some concluding remarks are given in Section 7.

2 Entries of the powers of the doubly Leslie matrix

2.1 Entries of the powers of doubly Leslie matrix by the linear recursive method

The main goal here is to establish compact formulas for the entries of the $n-th$ powers of the doubly Leslie matrix (1.1), in terms of a family of sequences defined by the linear recursive relation (1.2). To this aim, the first step consists in considering the similarity between the doubly Leslie matrix and companion matrix proved in [16]. Then, we utilize a tool based on the $n-th$ power of the $r \times r$ companion matrix established in [13, 1]. That is, it has been proved in [16] that every doubly Leslie matrix $\mathbf{L} = \mathbf{L}([a_i], [s_j], [b_j])$ is similar to a companion matrix, namely, $\mathbf{B} = \mathbf{P}^{-1}\mathbf{L}\mathbf{P} = \mathbf{L}([\gamma_i], [1], [0])$, where $\mathbf{B}$ is the companion matrix of the form,

$$\mathbf{B} = \begin{bmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \cdots & \gamma_{r-1} \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 0 \end{bmatrix}, \tag{2.3}$$

such that the coefficients γ_j $(0 \leq j \leq r-1)$ are as follows,

$$\begin{cases} \gamma_0 &= c_0 + d_0, \\ \gamma_j &= -\sum_{i+k=j-1} c_i d_k + c_j + d_j, \quad \text{for} \quad 1 \leq j \leq r-2, \\ \gamma_{r-1} &= -\sum_{i+k=r-2} c_i d_k + c_{r-1}, \end{cases}$$

with $c_0 = a_0$, $c_i = a_i \prod_{j=0}^{i-1} s_j$, for $1 \leq i \leq r-1$, and $d_0 = b_0$, $d_j = b_j \prod_{k=r-1-j}^{r-2} s_k$, for $1 \leq j \leq r-2$. Moreover, the entries $\Gamma_{i,j}$ $(0 \leq i, j \leq r-1)$ of the matrix $\mathbf{P}$ are defined by,

$$\begin{cases} \Gamma_{i,j} &= \frac{-d_{j-i-1}}{\prod_{k=i}^{r-2} s_k}, \quad \text{if} \quad j > i, \\ \Gamma_{i,j} &= 0, \quad \text{if} \quad i > j, \\ \Gamma_{i,i} &= \frac{1}{\prod_{k=i}^{r-2} s_k}, \quad \text{if} \quad 0 \leq i \leq r-2, \\ \Gamma_{r-1,r-1} &= 1. \end{cases} \tag{2.4}$$

In the sequel $\mu_{i,j}$ $(0 \leq i, j \leq r-1)$ will denote the entries of the matrix $\mathbf{P}^{-1}$. Since $\mathbf{L} = \mathbf{P}\mathbf{B}\mathbf{P}^{-1}$, then we have,

$$\mathbf{L}^n = \mathbf{P}\mathbf{B}^n\mathbf{P}^{-1}, \tag{2.5}$$

for every integer $n \geq 0$. Therefore, the entries of the powers of the doubly Leslie matrix $\mathbf{L} = \mathbf{L}([a_i], [s_j], [b_j])$, will be readily available from the entries of the powers of the companion matrix $\mathbf{B}$. It is known, that the computation of the powers $\mathbf{B}^n$ of the companion matrix $\mathbf{B}$, in terms of a family of sequences (1.2), has been established in [13, 1]. More precisely, the entries of $\mathbf{B}^n$ are given in terms of the family of sequences $\{v_n^{(s)}\}_{n \geq 0}$, indexed by s $(0 \leq s \leq r-1)$, defined as follows,

$$\begin{cases} v_{n+1}^{(s)} &= \gamma_0 v_n^{(s)} + \gamma_1 v_{n-1}^{(s)} + \cdots + \gamma_{r-1} v_{n-r+1}^{(s)}, \quad \text{for} \quad n \geq r-1, \\ v_n^{(s)} &= \delta_{s,n}, \quad \text{for} \quad 0 \leq n \leq r-1. \end{cases} \tag{2.6}$$

That is, it was established in [13, 1] that the scalars $b_{ij}^{(n)}$, the entries of the matrix power $\mathbf{B}^n$, are expressed in terms of sequences (2.6) as follows,

$$b_{ij}^{(n)} = v_{n+r-i-1}^{(r-j-1)}. \tag{2.7}$$

By combining expressions (2.5) and (2.7), we get an explicit formula for the entries of the powers of the doubly Leslie matrix.

Theorem 2.1. *(entries of the powers of the doubly Leslie matrix.) Let* $\mathbf{L} = (L_{ij})_{0\leq i,j\leq r-1}$ *be the doubly Leslie matrix* (1.1) *and* $\mathbf{B}$ *the associated companion matrix* (2.3)*. Then, the entries* $L_{ij}^{(n)}$ *of the power* $\mathbf{L}^n$*, are defined by*

$$L_{ij}^{(n)} = \sum_{p=i}^{r-1} \Gamma_{i,p} \sum_{k=0}^{j} \mu_{k,j} v_{n+r-p-1}^{(r-1-k)}, \tag{2.8}$$

for every $n \geq 0$*, where the sequences* $\{v_n^{(s)}\}_{n\geq 0}$ $(0 \leq s \leq r-1)$ *are defined by* (2.6)*, the coefficients* $\Gamma_{i,j}$ *are defined by* (2.4) *and* $\mu_{i,j}$ *are the entries of the matrix* $\mathbf{P}^{-1}$.

Proof. For every $n \geq 0$, we have $\mathbf{B}^n = \mathbf{P}^{-1}\mathbf{L}^n\mathbf{P}$. Since the entries of $\mathbf{B}^n$ are identified by expression (2.7) and multiplying this matrix by $\mathbf{P}$ and $\mathbf{P}^{-1}$, where the entries of matrix $\mathbf{P}$ are given by formula (2.4) and entries of $\mathbf{P}^{-1}$ are denoted by $\mu_{i,j}$, we get the entries of the matrix $\mathbf{L}^n$ as follows $L_{ij}^{(n)} = \sum_{p=i}^{r-1} \Gamma_{i,p} \sum_{k=0}^{j} \mu_{k,j} v_{n+r-p-1}^{(r-1-k)}$.

For the purpose illustration of Theorem 2.1, we detail the following example.

Example 2.2. *Let us consider the doubly Leslie matrix* (1.1) *defined by,*

$$\mathbf{L} = \begin{bmatrix} 12 & 3 & 5 \\ 0.4 & 0 & 0.3 \\ 0 & 0.5 & 0.25 \end{bmatrix}. \tag{2.9}$$

Then the associated companion matrix $\mathbf{B}$ *and the matrices* $\mathbf{P}$ *and* $\mathbf{P}^{-1}$ *are as follows*

$$\mathbf{B} = \begin{bmatrix} 12.25 & -1.65 & -1.1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} 5 & -1.25 & -0.75 \\ 0 & 2 & -0.5 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{P}^{-1} = \begin{bmatrix} 0.2 & 0.125 & 0.2125 \\ 0 & 0.5 & 0.25 \\ 0 & 0 & 1 \end{bmatrix}.$$

The family of sequences (2.6) *associated to the matrix* $\mathbf{B}$ *are defined as follows,*

$$\begin{cases} v_{n+1}^{(s)} = 12.25 v_n^{(s)} - 1.65 v_{n-1}^{(s)} - 1.1 v_{n-2}^{(s)}, & \text{for } n \geq 2, \\ v_i^{(s)} = \delta_{s,i}, & \text{for } 0 \leq i \leq 2. \end{cases}$$

Suppose that the computation of $\mathbf{L}^4$ *is required. First, Expression* (2.7) *shows that we have,*

$$\mathbf{B}^4 = \begin{bmatrix} 21751.71703 & -3127.875781 & -1976.414688 \\ 1796.740625 & -258.355625 & -163.25375 \\ 148.4125 & -21.3125 & -13.475 \end{bmatrix}.$$

Second, since the entries $L_{ij}^{(4)}$ *of* $\mathbf{L}^4$ *are given by Expression* (2.8), *we derive that,*

$$\mathbf{L}^4 = \begin{bmatrix} 21280.27 & 5649.94375 & 9117.27436 \\ 703.855 & 186.881875 & 301.5621875 \\ 29.6825 & 7.8953125 & 12.73453125 \end{bmatrix}.$$

.

As a consequence of Theorem 2.1, we can observe that if $b_0 = b_1 = \cdots = b_{r-2} = 0$, the doubly Leslie matrix $\mathbf{L} = \mathbf{L}([a_i], [s_j], [0])$ defined by (1.1), is nothing else but the usual Leslie matrix. Therefore, the entries of powers of Leslie matrix, are given by,

$$L_{ij}^{(n)} = \Gamma_{i,i}\mu_{j,j}v_{n+r-i-1}^{(r-j-1)},$$

where $\Gamma_{i,i} = \frac{1}{\prod_{k=i}^{r-2} s_k}$ for $0 \leq i \leq r-2$, $\Gamma_{r-1,r-1} = 1$; $\mu_{j,j} = \prod_{k=j}^{r-2} s_k$ $0 \leq j \leq r-2$, $\mu_{r-1,r-1} = 1$ and $\gamma_0 = a_0$, $\gamma_j = a_j \prod_{k=0}^{j-1} s_j$ for $1 \leq j \leq r-1$.

2.2 Powers of doubly Leslie matrix by the analytic method

In this subsection, we consider the analytic properties of the sequences $\{v_n^{(s)}\}_{n\geq 0}$ $(0 \leq s \leq r-1)$ defined as in (2.6), for establishing another expressions for the entries of the powers of doubly Leslie matrix.

Let $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ be the characteristic polynomial of sequences (1.2). Suppose that $\lambda_1, \lambda_2, \cdots, \lambda_l$ are the roots of $P(X)$, with multiplicities $m_1, m_2, \cdots, m_l$, respectively. It is well known, that the analytic expression of the general term v_n for the sequence (1.2) is given by $v_n = \sum_{i=1}^{l} \left(\sum_{j=0}^{m_i-1} \beta_{i,j} n^j \right) \lambda_i^n$, for every $n \geq 0$, where the $\beta_{i,j}$ are determined by the initial conditions $\{v_j\}_{0\leq j\leq r-1}$ (see, for example [12, 3, 15]). More precisely, under the preceding data the analytic formula of the general term $v_n^{(s)}$ $(0 \leq s \leq r-1)$ defined as in (2.6), is given by,

$$v_n^{(s)} = \sum_{i=1}^{l} \left(\sum_{j=0}^{m_i-1} \beta_{i,j}^{(s)} n^j \right) \lambda_i^n, \quad 0 \leq s \leq r-1, \tag{2.10}$$

where, for each fixed s $(1 \leq s \leq l)$, the $\beta_{i,j}^{(s)}$ are computed by solving the following system of r linear equations,

$$\sum_{i=1}^{l} \left(\sum_{j=0}^{m_i-1} \beta_{i,j}^{(s)} n^j \right) \lambda_i^n = \delta_{s,n}, \quad 0 \leq n \leq r-1. \tag{2.11}$$

Combining Expressions (2.8) and (2.10), we show that the entries of the powers of doubly Leslie matrix can be formulated in terms of the characteristic roots λ_i $(1 \leq i \leq l)$ and their multiplicities m_i, and the initial conditions $\beta_{i,j}^{(s)}$. Indeed, we get the following result.

Theorem 2.3. *(Analytic representation of the entries of the powers of doubly Leslie matrix.) Let* $\mathbf{L} = (L_{ij})_{0 \leq i,j \leq r-1}$ *be the doubly Leslie matrix defined in* (1.1). *Suppose that* $\lambda_1, \lambda_2, \cdots, \lambda_l$ *are the distinct roots of the polynomial* $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ *of multiplicities* $m_1, m_2, \cdots, m_l$*, respectively. Then, the entries* $L_{ij}^{(n)}$ *of the power* $\mathbf{L}^n$ *are given by,*

$$L_{ij}^{(n)} = \sum_{p=i}^{r-1} \Gamma_{i,p} \sum_{k=0}^{j} \mu_{k,j} \sum_{d=1}^{l} \sum_{u=0}^{m_d-1} \beta_{d,u}^{(r-1-k)} (n+r-p-1)^u \lambda_d^{n+r-p-1}, \tag{2.12}$$

for every $n \geq 0$*, where the coefficients* $\Gamma_{i,j}$ *are defined* $(0 \leq i,j \leq r-1)$ *by* (2.4) *and* $\mu_{i,j}$ *are the entries of the matrix* $\mathbf{P}^{-1}$.

Proof. Following Theorem 2.1, the entries $L_{ij}^{(n)}$ of each power $\mathbf{L}^n$ are given by $L_{ij}^{(n)} = \sum_{p=i}^{r-1} \Gamma_{i,p} \sum_{k=0}^{j} \mu_{k,j} v_{n+r-p-1}^{(r-1-k)}$ for every $n \geq 0$. Upon substitution of expression (2.10) into expression (2.8) we get the desired expression,

$$L_{ij}^{(n)} = \sum_{p=i}^{r-1} \Gamma_{i,p} \sum_{k=0}^{j} \mu_{k,j} \sum_{d=1}^{l} \sum_{u=0}^{m_d-1} \beta_{d,u}^{(r-1-k)} (n+r-p-1)^u \lambda_d^{n+r-p-1}.$$

□

Now, if we suppose that $b_0 = b_1 = \cdots = b_{r-2} = 0$, it follows from Theorem 2.4 that, the entries of the powers of the usual Leslie matrix is given by,

$$L_{ij}^{(n)} = \Gamma_{i,i}\mu_{j,j}\sum_{d=1}^{l}\left(\sum_{u=0}^{m_d-1}\beta_{d,u}^{(r-j-1)}(n+r-i-1)^u\right)\lambda_d^{n+r-i-1},$$

where $\Gamma_{i,i} = \frac{1}{\prod_{k=i}^{r-2} s_k}$ for $0 \le i \le r-2$, $\Gamma_{r-1,r-1} = 1$; $\mu_{j,j} = \prod_{k=j}^{r-2} s_k$ $0 \le j \le r-2$, $\mu_{r-1,r-1} = 1$ and $\gamma_0 = a_0$, $\gamma_j = a_j \prod_{k=0}^{j-1} s_j$ for $1 \le j \le r-1$.

When the roots of the polynomial $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ are simple, we get the following corollary.

Corollary 2.4. *Let* $\mathbf{L} = (L_{ij})_{0\le i,j\le r-1}$ *be the doubly Leslie matrix defined in* (1.1). *Suppose that the roots of the polynomial* $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ *are simple. Then for every* $n \ge 0$ *, the entries* $L_{ij}^{(n)}$ *of the power* $\mathbf{L}^n$ *are given by,*

$$L_{ij}^{(n)} = \sum_{p=i}^{r-1}\Gamma_{i,p}\sum_{k=0}^{j}\mu_{k,j}\sum_{d=1}^{l}\beta_{d,0}^{(r-1-k)}\lambda_d^{n+r-p-1}, \tag{2.13}$$

where the coefficients $\Gamma_{i,j}$ $(0 \le i,j \le r-1)$ *are defined by* (2.4) *and the* $\mu_{i,j}$ *are the entries of the matrix* $\mathbf{P}^{-1}$.

Moreover, it was established in [12] that there is no need to solve the linear system (2.11), when the characteristic roots of $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ are simple. That is, if $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ $(\gamma_{r-1} \neq 0)$ owns r distinct roots $\lambda_1, \cdots, \lambda_r$, then the analytic formula (2.10) of each $v_n^{(s)}$ takes the following form

$$v_n^{(s)} = \sum_{i=1}^{r}\frac{1}{P'(\lambda_i)}\left(\sum_{p=0}^{r-1}\frac{A_p^{(s)}}{\lambda_i^{p+1}}\right)\lambda_i^n, \quad n \ge r, \tag{2.14}$$

where $A_m^{(s)} = \gamma_{r-1}v_m^{(s)} + \cdots + \gamma_m v_{r-1}^{(s)}$ (for more details see [12]). Therefore, a straightforward computation, combining expressions (2.8) and (2.14), allows us to derive the following theorem.

Theorem 2.5. *Let* $\mathbf{L} = (L_{ij})_{0\le i,j\le r-1}$ *be the doubly Leslie matrix defined in* (1.1). *Suppose that the roots of the polynomial* $P(X) = X^r - \gamma_0 X^{r-1} - \cdots -$

γ_{r-1} are simple. Then for every $n \geq 0$, the entries $L_{ij}^{(n)}$ of the power $\mathbf{L}^n$, are expressed as follows:

$$L_{ij}^{(n)} = \sum_{p=i}^{r-1} \Gamma_{i,p} \sum_{k=0}^{j} \mu_{k,j} \sum_{d=1}^{r} \frac{1}{P'(\lambda_d)} \left(\sum_{u=0}^{r-1} \frac{A_u^{(r-1-k)}}{\lambda_d^{u+1}} \right) \lambda_d^{n+r-p-1}, \qquad (2.15)$$

where the coefficients $\Gamma_{i,j}$ are defined ($0 \leq i, j \leq r-1$) by (2.4) and $\mu_{i,j}$ are the entries of the matrix $\mathbf{P}^{-1}$.

Example 2.6. *Let consider again the doubly Leslie matrix (2.9) used of Example 2.2. The characteristic polynomial is $P(X) = X^3 - 12.25X^2 + 1.65X + 1.1$ and the roots of this polynomial are $\lambda_1 = -0.23799082$, $\lambda_2 = 0.38179006$ and $\lambda_3 = 12.10620076$. The analytic formula (2.14) of each sequence $\{v_n^{(s)}\}_{n\geq 0}$ ($0 \leq s \leq 2$), associated with this doubly Leslie model, is given as follows,*

$$v_n^{(s)} = \frac{1}{7.65} B(s)(-0.24)^n - \frac{1}{7.27} C(s)(0.38)^n + \frac{1}{144.73} D(s)(12.11)^n$$

such that $B(s) = \frac{A_0^{(s)}}{-0.24} + \frac{A_1^{(s)}}{(-0.24)^2} + \frac{A_2^{(s)}}{(-0.24)^3}$, $C(s) = \frac{A_0^{(s)}}{0.38} + \frac{A_1^{(s)}}{(0.38)^2} + \frac{A_2^{(s)}}{(0.38)^3}$ and $D(s) = \frac{A_0^{(s)}}{12.11} + \frac{A_1^{(s)}}{(12.11)^2} + \frac{A_2^{(s)}}{(12.11)^3}$, where for the real numbers $A_m^{(s)}$ ($0 \leq m \leq 2$, $0 \leq s \leq 2$) are given by: $A_0^{(s)} = -1.1v_0^{(s)} - 1.65v_1^{(s)} + 12.25v_2^{(s)}$, $A_1^{(s)} = -1.1v_1^{(s)} - 1.65v_2^{(s)}$ and $A_2^{(s)} = -1.1v_2^{(s)}$. Then, we have $A_0^{(0)} = -1.1$, $A_0^{(1)} = -1.65$, $A_0^{(2)} = 12.25$, $A_1^{(0)} = 0$, $A_1^{(1)} = -1.1$, $A_1^{(2)} = -1.65$ and $A_2^{(0)} = 0$, $A_2^{(1)} = 0$, $A_2^{(2)} = -1.1$.

Suppose that the computation of the power $\mathbf{L}^4$ is required. Therefore, using formula (2.15), we have

$$L_{ij}^{(4)} = \sum_{p=i}^{2} \Gamma_{i,p} \sum_{k=0}^{j} \mu_{k,j} \sum_{d=1}^{3} \frac{1}{P'(\lambda_d)} \left(\sum_{u=0}^{2} \frac{A_u^{(2-k)}}{\lambda_d^{u+1}} \right) \lambda_d^{6-p}.$$

Then, a straightforward computation allows us to obtain $L_{00}^{(4)} = 21280.27$, $L_{01}^{(4)} = 5649.94375$, $L_{02}^{(4)} = 9117.27436$, $L_{10}^{(4)} = 703.855$, $L_{11}^{(4)} = 186.881875$, $L_{12}^{(4)} = 301.5621875$, $L_{20}^{(4)} = 29.6825$, $L_{21}^{(4)} = 7.8953125$, $L_{22}^{(4)} = 12.73453125$.

For the doubly Leslie matrix (2.9) of Examples 2.2 and 2.3, we show that both Expressions (2.8) and (2.15) of the entries of the powers $L_{ij}^{(n)}$, give the same numerical results.

2.3 Powers of doubly Leslie matrix by the derivative method

In [12], the authors had expressed the analytic formula of sequences (1.2) can take various forms. Among these forms the so-called the derivative expression, when applied to sequences (2.6), we have,

$$v_n^{(s)} = \sum_{i=1}^{l} \sum_{p=0}^{r-1} A_p^{(s)} \times \frac{f_{i,n-p}^{(m_i-1)}(\lambda_i)}{(m_i-1)!}, \quad n \geq r, \tag{2.16}$$

where the function f_i for $1 \leq i \leq l$ is defined by $f_{i,n}(x) = \frac{x^{n-1}}{\prod\limits_{k=1,k\neq i}^{l} (x-\lambda_k)^{m_k}}$ and $A_j^{(s)} = \gamma_{r-1} v_j^{(s)} + \cdots + \gamma_j v_{r-1}^{(s)}$ for $0 \leq j \leq r-1$. Combining Expressions (2.8) and (2.16), we can obtain another analytic expression for the entries $L_{ij}^{(n)}$ of the power $\mathbf{L}^n$. That is, we have the result.

Theorem 2.7. *Let $\mathbf{L} = (L_{ij})_{0\leq i,j\leq r-1}$ be the doubly Leslie matrix defined in (1.1). Suppose that $\lambda_1, \lambda_2, \cdots, \lambda_l$ are the distinct roots of the polynomial $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ of multiplicities $m_1, m_2, \cdots, m_l$, respectively. Then, the entries $L_{ij}^{(n)}$ of the power $\mathbf{L}^n$ are given by,*

$$L_{ij}^{(n)} = \sum_{p=i}^{r-1} \Gamma_{i,p} \sum_{k=0}^{j} \mu_{k,j} \sum_{u=1}^{l} \sum_{d=0}^{r-1} A_d^{(r-1-k)} \frac{f_{u,n+r-p-1-d}^{(m_u-1)}(\lambda_u)}{(m_u-1)!}, \tag{2.17}$$

for every $n \geq 0$, where the coefficients $\Gamma_{i,j}$ are defined $(0 \leq i, j \leq r-1)$ by (2.4) and $\mu_{i,j}$ are the entries of the matrix $\mathbf{P}^{-1}$.

Once again, if $b_0 = b_1 = \cdots = b_{r-2} = 0$, it follows from Theorem 2.7 that, the entries of the powers of the usual Leslie matrix is given by,

$$L_{ij}^{(n)} = \Gamma_{i,i}\mu_{j,j} \sum_{d=1}^{l} \sum_{p=0}^{r-1} A_p^{(r-j-1)} \times \frac{f_{d,n+r-i-1}^{(m_d-1)}(\lambda_d)}{(m_d-1)!},$$

where $\Gamma_{i,i} = \frac{1}{\prod\limits_{k=i}^{r-2} s_k}$, for $0 \leq i \leq r-2$, $\Gamma_{r-1,r-1} = 1$; $\mu_{j,j} = \prod\limits_{k=j}^{r-2} s_k$, $0 \leq j \leq r-2$, $\mu_{r-1,r-1} = 1$ and $\gamma_0 = a_0$, $\gamma_j = a_j \prod\limits_{k=0}^{j-1} s_j$ for $1 \leq j \leq r-1$.

On the other side, if the roots of the polynomial $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ are simple, Expression (2.17) allows us to derive the corollary.

Corollary 2.8. *Let* $\mathbf{L} = (L_{ij})_{0\leq i,j\leq r-1}$ *be the doubly Leslie matrix defined in* (1.1). *Suppose that the roots of the polynomial* $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ *are simple. Then, for every* $n \geq 0$, *the entries* $L_{ij}^{(n)}$ *of the power* $\mathbf{L}^n$ *are given by,*

$$L_{ij}^{(n)} = \sum_{p=i}^{r-1} \Gamma_{i,p} \sum_{k=0}^{j} \mu_{k,j} \sum_{u=1}^{l} \sum_{d=0}^{r-1} A_d^{(r-1-k)} f_{u,n+r-p-1-d}(\lambda_u),$$

where the coefficients $\Gamma_{i,j}$ *are defined* ($0 \leq i,j \leq r-1$) *by* (2.4) *and* $\mu_{i,j}$ *are the entries of the matrix* $\mathbf{P}^{-1}$.

Example 2.9. *Again, let's consider the doubly Leslie matrix used in Example 2.2 The characteristic polynomial is* $P(X) = X^3 - 12.25X^2 + 1.65X + 1.1$ *and the roots of this polynomial are* $\lambda_1 = -0.23799$, $\lambda_2 = 0.38179$, $\lambda_3 = 12.10620$. *The analytic formulas of the r-gfs* $\{v_n^{(s)}\}_{n\geq 0}$ ($0 \leq s \leq 2$) *associated with this Doubly Leslie model are,*

$$\begin{aligned} v_n(s) = &\frac{1}{7.65068}\left[A_0^{(s)}(-0.23799)^{n-1} + A_1^{(s)}(-0.23799)^{n-2} + A_2^{(s)}(-0.23799)^{n-3}\right] \\ &- \frac{1}{7.26655}\left[A_0^{(s)}(0.38179)^{n-1} + A_1^{(s)}(0.38179)^{n-2} + A_2^{(s)}(0.38179)^{n-3}\right] \\ &+ \frac{1}{144.72834}\left[A_0^{(s)}(12.10620)^{n-1} + A_1^{(s)}(12.10620)^{n-2} + A_2^{(s)}(12.10620)^{n-3}\right], \end{aligned}$$

where the real numbers $A_p^{(s)}$ *for* $0 \leq p,s \leq 2$ *are defined as follows* $A_0^{(s)} = -1.1v_0^{(s)} - 1.65v_1^{(s)} + 12.25v_2^{(s)}$, $A_1^{(s)} = -1.1v_1^{(s)} - 1.65v_2^{(s)}$,$A_2(s) = -1.1v_2^{(s)}$. *Then, we have,* $A_0^{(0)} = -1.1$, $A_0^{(1)} = -1.65$, $A_0^{(2)} = 12.25$, $A_1^{(0)} = 0$, $A_1^{(1)} = -1.1$, $A_1^{(2)} = -1.65$ *and* $A_2^{(0)} = 0$, $A_2^{(1)} = 0$, $A_2^{(2)} = -1.1$. *Suppose that the computation of the power* $\mathbf{L}^4$ *is required. Then, using formula* (2.8), *we get,*

$$L_{ij}^{(4)} = \sum_{p=i}^{2} \Gamma_{i,p} \sum_{k=0}^{j} \mu_{k,j} \sum_{u=1}^{l} \sum_{d=0}^{2} A_d^{(2-k)} f_{u,6-p-d}(\lambda_u).$$

Therefore, a straightforward computation allows us to obtain $L_{00}^{(4)} = 21280.27$, $L_{01}^{(4)} = 5649.94375$, $L_{02}^{(4)} = 9117.27436$, $L_{10}^{(4)} = 703.855$, $L_{11}^{(4)} = 186.881875$, $L_{12}^{(4)} = 301.5621875$, $L_{20}^{(4)} = 29.6825$, $L_{21}^{(4)} = 7.8953125$, $L_{22}^{(4)} = 12.73453125$

Once again, for the doubly Leslie matrix (2.9) of Examples 2.2, 2.6, and 2.9, we have shown that the three Expressions (2.8), (2.13), and (2.8), give the same numerical results.

The primordial tasks in the next sections is to promote an application of the results of this section and propose three algorithms for computing the entries of powers of a given doubly Leslie matrix. More precisely, an algorithm is elaborated for each method for computing the entries of powers $\mathbf{L}^n = (L_{ij}^{(n)})_{0\leq i,j\leq r-1}$. This allows us to discuss and analyze the efficiency of each method.

First, in the next section, we are interested in applying results of this section for computing entries of powers for a special case of doubly Leslie matrices, namely, the doubly companion matrices.

3 Entries of powers of doubly companion matrix

In this section, we are interested in compact formulas for the entries of the powers of doubly companion matrix. These matrices have first been studied recently in the literature by Butcher and Chartier in [5] and Butcher and Wright [6], where the doubly companion matrices are used as a tool to analyze numerical methods and some general linear methods property. Later, Wanicharpichat [17] study some properties of the doubly companion matrices, especially, the explicit form of its minimal polynomials is obtained. Since the doubly companion matrices represent a particular case of the doubly Leslie matrices, by taking $s_0 = s_1 = \cdots = s_{r-2} = 1$, we utilize the results of Section 2 to establish some compact formulas for the entries of the powers for a given doubly companion matrix. First, Theorem 2.1 allows us to show that the entries of the powers of the doubly companion matrix can be expressed in terms of sequences (2.6) as follows.

Proposition 3.1. *Let* $\mathbf{L} = \mathbf{L}([a_i],[1],[b_j])$ *(*$0 \leq i \leq r-1, 0 \leq j \leq r-2$*)be a doubly companion matrix, and* $\mathbf{B}$ *its associated companion matrix* (2.3)*. Therefore, for every* $n \geq 0$*, the entries of powers of doubly companion matrix, are given by*

$$\mathbf{L}_{ij}^{(n)} = \sum_{p=i}^{r-1} \Gamma_{i,p} \sum_{k=0}^{j} \mu_{k,j} v_{n+r-p-1}^{(r-1-k)},$$

where

$$\begin{cases} \Gamma_{i,j} = -b_{j-i-1}, & \text{if } j > i, \\ \Gamma_{i,j} = 0, & \text{if } i > j, \\ \Gamma_{i,i} = 1, & \text{if } 0 \leq i \leq r-1, \end{cases} \tag{3.18}$$

and

$$\begin{cases} \gamma_0 &= a_0 + b_0, \\ \gamma_j &= -\sum\limits_{i+k=j-1} a_i b_k + a_j + b_j, \quad \textit{for} \quad 1 \le j \le r-2, \\ \gamma_{r-1} &= -\sum\limits_{i+k=r-2} a_i b_k + a_{r-1}. \end{cases} \tag{3.19}$$

Using Theorem 2.3, we can derive the analytic formulas of the entries of the powers of the doubly companion matrices, namely, these entries are expressed in terms of the roots λ_i $(1 \le i \le l)$ of the polynomial $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ and their multiplicities, and the initial conditions $\beta_{i,j}^{(s)}$. That is, we have the following result.

Proposition 3.2. *Let* $\mathbf{L} = \mathbf{L}([a_i],[1],[b_j])$ *be a doubly companion matrix. Suppose that* $\lambda_1, \lambda_2, \cdots, \lambda_l$ *are the distinct roots of the polynomial* $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ *of multiplicities* $m_1, m_2, \cdots, m_l$*, respectively. Then, for every* $n \ge 0$*, the entries* $L_{ij}^{(n)}$ *of the power* $\mathbf{L}^n$ *are given by,*

$$L_{ij}^{(n)} = \sum_{p=i}^{r-1} \Gamma_{i,p} \sum_{k=0}^{j} \mu_{k,j} \sum_{d=1}^{l} \sum_{u=0}^{m_d-1} \beta_{d,u}^{(r-1-k)} (n+r-p-1)^u \lambda_d^{n+r-p-1}, \tag{3.20}$$

where $\Gamma_{i,j}$ $(0 \le i,j \le r-1)$*and* $\gamma_j (0 \le j \le r-1)$ *are defined respectively by* (3.18) *and* (3.19)

Once again, from Theorem 2.7, we can derive another analytic expression for the entries of the doubly companion matrix as follows.

Proposition 3.3. *Let* $\mathbf{L} = \mathbf{L}([a_i],[1],[b_j])$ *be a doubly companion matrix. Suppose that* $\lambda_1, \lambda_2, \cdots, \lambda_l$ *are the distinct roots of the polynomial* $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ *of multiplicities* $m_1, m_2, \cdots, m_l$*, respectively. Then, the entries* $L_{ij}^{(n)}$ *of the power* $\mathbf{L}^n$ *are given by,*

$$L_{ij}^{(n)} = \sum_{p=i}^{r-1} \Gamma_{i,p} \sum_{k=0}^{j} \mu_{k,j} \sum_{u=1}^{l} \sum_{d=0}^{r-1} A_d^{(r-1-k)} \frac{f_{u,n+r-p-1-d}^{(m_u-1)}(\lambda_u)}{(m_u-1)!} \tag{3.21}$$

for every $n \ge 0$*, where* $\Gamma_{i,j}$ $(0 \le i,j \le r-1)$*and* $\gamma_j (0 \le j \le r-1)$ *are defined respectively by* (3.18) *and* (3.19)

When the roots of the polynomial $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ are simple, the analytic, and the derivative Expressions (3.20)-(3.21), take respectively the following forms,

Corollary 3.4. *Let* $\mathbf{L} = \mathbf{L}([a_i], [1], [b_j])$ *be a doubly companion matrix. Suppose that the roots of the polynomial* $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ *are simple. Then, the entries* $L_{ij}^{(n)}$ *of the power* $\mathbf{L}^n$ *are given by,*

$$L_{ij}^{(n)} = \sum_{p=i}^{r-1} \Gamma_{i,p} \sum_{k=0}^{j} \mu_{k,j} \sum_{d=1}^{l} \beta_{d,0}^{(r-1-k)} \lambda_d^{n+r-p-1},$$

$$L_{ij}^{(n)} = \sum_{p=i}^{r-1} \Gamma_{i,p} \sum_{k=0}^{j} \mu_{k,j} \sum_{u=1}^{l} \sum_{d=0}^{r-1} A_d^{(r-1-k)} f_{u,n+r-p-1-d}(\lambda_u),$$

for every $n \geq 0$*, where* $\Gamma_{i,j}$ $(0 \leq i, j \leq r-1)$*and* $\gamma_j (0 \leq j \leq r-1)$ *are defined respectively by* (3.18) *and* (3.19)

4 Applications to some matrix dynamical systems

4.1 General settings

In this subsection we are interested in applying the results of Section 2, to illustrate the fundamental role of the powers of doubly Leslie matrix, for studying some matrix dynamical systems. Let consider the linear matrix difference equation

$$N(n+1) = \mathbf{L}N(n), \quad for \ \ n \geq 0, \tag{4.22}$$

where $\mathbf{L}$ is a doubly Leslie matrix and $N(n) =^t (x_0(n); x_1(n); \cdots; x_{r-1}(n))$ is the dynamical vector described by the discrete matrix equation (4.22), whose initial condition is $N(0) =^t (x_0(0); x_1(0); \cdots; x_{r-1}(0))$. When the entries $\mathbf{L}_{ij}$ are nonnegative, the matrix $\mathbf{L}$ is known as a Lefkovitch matrix and the dynamical system $\mathbf{L}$ describes the evolution of a population dynamics system ([8]). The well known special case of (4.22) is the classical Leslie matrix model, associated to $b_0 = ... = b_{r-2} = 0$ (see [14]). In this case, the entries of $N(n) =^t (x_0(n); x_1(n); \cdots; x_{r-1}(n))$, called the population dynamical vector, represent the number of individuals in each stage class at time $t = n$. On the other hand, the entries of the matrix $\mathbf{L}$, are defined through the characteristics of the population considered (see [2, 14]). In the numerical of the next subsection, the doubly Leslie matrices are related to some population dynamical systems (see [2]).

Let $\mathbf{L}$ be a doubly Leslie matrix, a standard iterative process shows that, the matrix equation (4.22) takes the form,

$$N(n) = \mathbf{L}^n N(0), \quad for \ \ n \geq 0, \tag{4.23}$$

where $N(0) = {}^t(x_0(0); x_1(0); \cdots ; x_{r-1}(0))$ is the initial vector of the population dynamics. Expression (4.23) shows that the computation of the powers $\mathbf{L}^n$ is a fundamental tools for controlling the discrete time evolution of the dynamical system (4.22), with the aid of the initial vector of the population dynamics $N(0) = {}^t(x_0(0); x_1(0); \cdots ; x_{r-1}(0))$. Therefore, results of Theorem 2.1, Theorem 2.3, Theorem 2.7 and expression (4.23) permit us to give various formulas of the entries of the sequence of vectors $N(n) = {}^t(x_0(n); x_1(n); \cdots ; x_{r-1}(n))$. That is, we have the following result.

Theorem 4.1. *(entries of the vector of the population dynamics.) Let* $\mathbf{L} = (L_{ij})_{0\leq i,j\leq r-1}$ *be the doubly Leslie matrix defined in* (1.1). *Suppose that* $\lambda_1, \lambda_2, \cdots , \lambda_l$ *are the distinct roots of the polynomial* $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ *of multiplicities* $m_1, m_2, \cdots , m_l$*, respectively. Then, for every* $n \geq 0$*, the entries* $x_p(n)$ $(0 \leq p \leq r-1)$*, of the vector* $N(n) = {}^t(x_0(n); x_1(n); \cdots ; x_{r-1}(n))$*, are given by,*

$$x_p(n) = \sum_{s=0}^{r-1}\sum_{i=p}^{r-1}\Gamma_{p,i}\sum_{k=0}^{s}\mu_{k,s}v_{n+r-i-1}^{(r-1-k)}x_s(0), \tag{4.24}$$

$$x_p(n) = \sum_{s=0}^{r-1}\sum_{i=p}^{r-1}\Gamma_{p,i}\sum_{k=0}^{s}\mu_{k,s}\sum_{d=1}^{l}\sum_{u=0}^{m_d-1}\beta_{d,u}^{(r-1-k)}(n+r-i-1)^u\lambda_d^{n+r-i-1}x_s(0), \tag{4.25}$$

$$x_p(n) = \sum_{s=0}^{r-1}\sum_{i=p}^{r-1}\Gamma_{p,i}\sum_{k=0}^{s}\mu_{k,s}\sum_{u=1}^{l}\sum_{d=0}^{r-1}A_d^{(r-1-k)}\frac{f_{u,n+r-i-1-d}^{(m_u-1)}(\lambda_u)}{(m_u-1)!}x_s(0), \tag{4.26}$$

where $N(0) = {}^t(x_0(0); x_1(0); \cdots ; x_{r-1}(0))$ *is the initial vector,* $\{v_n^{(s)}\}_{n\geq 0}$ $(0 \leq s \leq r-1)$ *are the sequences* (2.6) *and the coefficients* $\Gamma_{i,j}$ $(0 \leq i, j \leq r-1)$ *are as in* (2.4) *and* $\mu_{i,j}$ *are the entries of the matrix* $\mathbf{P}^{-1}$.

Especially, when the roots of the polynomial $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ are simple, we can derive that the analytical expressions (4.25) and (4.26) take the following forms.

Corollary 4.2. *Under the data of Theorem 4.1, suppose that the roots* $\lambda_1, \lambda_2, \cdots , \lambda_r$ *the roots of the polynomial* $P(X) = X^r - \gamma_0 X^{r-1} - \cdots - \gamma_{r-1}$ *are simple.*

Then, for every $n \geq 0$, the entries $x_p(n)$ $(0 \leq p \leq r-1)$ are given by,

$$x_p(n) = \sum_{s=0}^{r-1}\sum_{i=p}^{r-1}\Gamma_{p,i}\sum_{k=0}^{s}\mu_{k,s}\sum_{d=1}^{l}\beta_{d,0}^{(r-1-k)}\lambda_d^{n+r-i-1}x_s(0),$$

$$x_p(n) = \sum_{s=0}^{r-1}\sum_{i=p}^{r-1}\Gamma_{p,i}\sum_{k=0}^{s}\mu_{k,s}\sum_{u=1}^{l}\sum_{d=0}^{r-1}A_d^{(r-1-k)}f_{u,n+r-i-1-d}(\lambda_u)x_s(0),$$

where $N(0) = {}^t(x_0(0); x_1(0); \cdots; x_{r-1}(0))$ is the initial vector, the scalars $\Gamma_{i,j}$ are defined $(0 \leq i, j \leq r-1)$ by (2.4) and $\mu_{i,j}$ are the entries of the matrix $\mathbf{P}^{-1}$.

In this section, we expressed the entries $x_p(n)$ $(0 \leq p \leq r-1)$ of the vector of the population vector, as in terms three expressions of the family of sequences $\{v_n^{(s)}\}_{n\geq 0}$ $(0 \leq s \leq r-1)$ defined as in (2.6). Especially, since each entries $x_p(n)$ $(0 \leq p \leq r-1)$ is a linear combination of $v_n^{(s)}$ $(0 \leq s \leq r-1)$, therefore $\{x_p(n)\}_{n\geq 0}$ satisfies also the linear recursive relation (1.2).

In the aim to illustrate the application of Corollary 4.2, we present a practical numerical application in the next subsection.

4.2 Numerical Application

In this subsection, we present an example of the American Bison model to illustrate the conclusion of Theorem 4.1 and Corollary 4.2. All computations are carried out using Python 2.7.4 installed in Ubuntu 13.04 on Samsung PC with Intel (R) Core (TM) i5-3230M processor that has 4 cores.

It was established in [2] that evolution of the American Bison is a matrix dynamical model, described by the doubly Leslie matrix and the initial population vector, given as follows,

$$\mathbf{L} = \begin{bmatrix} 0 & 0 & 0.42 \\ 0.60 & 0 & 0 \\ 0 & 0.75 & 0.95 \end{bmatrix}; \quad and \quad N(0) = {}^t(0, 0, 100)$$

We show that the associated companion matrix $\mathbf{B}$ and the two matrices $\mathbf{P}$, $\mathbf{P}^{-1}$, are as follows,

$$\mathbf{B} = \begin{bmatrix} 0.95 & 0 & 0.189 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \mathbf{P} = \begin{bmatrix} \frac{20}{9} & -\frac{19}{9} & 0 \\ 0 & \frac{4}{3} & -\frac{19}{15} \\ 0 & 0 & 1 \end{bmatrix} \quad \mathbf{P}^{-1} = \begin{bmatrix} \frac{9}{20} & \frac{57}{80} & \frac{361}{400} \\ 0 & \frac{3}{4} & \frac{19}{20} \\ 0 & 0 & 1 \end{bmatrix}.$$

Therefore, the family of sequences (2.6) associated with the matrix $\mathbf{B}$ is defined as follows,

$$\begin{cases} v_{n+1}^{(s)} &= 0.95v_n^{(s)} + 0.189v_{n-2}^{(s)}, \quad \text{for} \quad n \geq 2, \\ v_i^{(s)} &= \delta_{s,i}, \qquad\qquad\qquad\quad \text{for} \quad 0 \leq i \leq 2. \end{cases}$$

Suppose we need to compute N(5), then using expression (4.24) we have,

$$x_p(5) = \sum_{s=0}^{2} \sum_{i=p}^{2} \Gamma_{p,2-i} \sum_{k=0}^{s} \mu_{2-k,s} v_{7-i}^{(2-k)} x_s(0)$$

Therefore, a straightforward computation allows us to obtain $x_0(5) = 49.2914625$, $x_1(5) = 26.36865$, $x_2(5) = 128.54984375$.

We show that our method allows us to get the same result as in [2].

5 Algorithms and numerical approach for the power of doubly Leslie matrix

5.1 Algorithm 1: recursive method

In this subsection, we describe an algorithm that takes a doubly Leslie matrix defined by (1.1) as input and produces its powers, using the recursive method. The proposed algorithm works as follows. The first step consists of constructing the matrix $\mathbf{P}$, and this is done using expression (2.4). Then, the matrix $\mathbf{P}^{-1}$ is constructed using the usual function numpy.linalg.inv in Python, and matrix $\mathbf{B}$ is constructed using the following equation $\mathbf{B} = \mathbf{P}^{-1} \times \mathbf{L} \times \mathbf{P}$ and it is computed with the aid of the usual function numpy.dot in python.The next step is dedicated to calculating the terms of the sequences $\{v_n^{(s)}\}_{n\geq 0}$ using expression (2.6). The last step consists of computing the entries of the powers of doubly Leslie matrix using equation (2.8). Algorithm 1, summarized in Table 1 below, and detailed in Appendix A, describes the steps needed for computing entries of powers of doubly Leslie matrix using the recursive method.

5.2 Algorithm 2: analytic method

Given an $r \times r$ doubly Leslie matrix $\mathbf{L}$ as in (1.1), the first step is to construct the matrix $\mathbf{P}$, and this is done using expression (2.4). Then the matrix $\mathbf{P}^{-1}$

Table 1: Recursive method

Input: The r × r matrices $\mathbf{L}$, $\mathbf{P}$ and $\mathbf{P}^{-1}$. The integer n. Output: The r × r matrix $\mathbf{L}^n$
Step1: Constructing doubly Leslie matrix
Step2: Computing matrix $\mathbf{P}$ from (2.4)
Step3: Computing matrix $\mathbf{P}^{-1}$ using numpy.linalg.inv()
Step4: Computing matrix $\mathbf{B}$ using the equation $\mathbf{B} = \mathbf{P}^{-1} \times \mathbf{L} \times \mathbf{P}$
Step 5: Computing $v_n^{(s)} from$ (2.6)
Step 6: Computing $\mathbf{L}^n$ from (2.8)

is constructed using the usual function numpy.linalg.inv in Python, and the matrix $\mathbf{B}$ is constructed using equation $\mathbf{B} = \mathbf{P}^{-1} \times \mathbf{L} \times \mathbf{P}$, and it is done using the usual function numpy.dot in Python. The next step is to generate the numbers $\beta_{i,j}^{(s)}$ $(0 \leq s \leq r-1)$ using expression (2.11). Then, to calculate entries of powers of doubly Leslie matrix using the analytic method, we use equation (2.12). Algorithm 2, summarized in Table 2 below, and detailed in Appendix B, describes the steps needed for computing entries of powers of doubly Leslie matrix using the analytic method.

Table 2: Analytic method

Input: The r × r matrices $\mathbf{L}$, $\mathbf{P}$ and $\mathbf{P}^{-1}$. The integer n. Output: The r × r matrix $\mathbf{L}^n$
Step1 to 4: The same steps as those in Algorithm 1
Step5: Computing eigenvalues and their multiplicities
Step6: Computing $\beta_{i,j}^{(s)}$ using (2.11)
Step7: Computing $\mathbf{L}^n$ from (2.12)

5.3 Algorithm 3: derivative method

Given an $r \times r$ doubly Leslie matrix $\mathbf{L}$ as in (1.1), the first step is to construct the matrix $\mathbf{P}$, and this is done using expression (2.4). Then the matrix $\mathbf{P}^{-1}$ is constructed using the usual function numpy.linalg.inv in Python, and the matrix $\mathbf{B}$ is constructed using the following equation $\mathbf{B} = \mathbf{P}^{-1} \times \mathbf{L} \times \mathbf{P}$ and it is done using the usual function numpy.dot in Python. The next step is to generate functions $f_{i,n}(x)$ defined by $f_{i,n}(x) = \frac{x^{n-1}}{\prod_{k=1,k\neq i}^{l} (x-\lambda_k)^{m_k}}$ and numbers

```
>>> power_matrix_doublyL(1,100,gamma,mu)
array([[  2338.74031419,   4306.5345568 ,   6344.00998738],
       [  1270.09459348,   2338.74031419,   3445.22764544],
       [  6152.192224  ,  11328.58926318,  16688.28671422]])
```

Figure 1: $\mathbf{L}^{100}$ by recursive algorithm

```
>>> power_matrix_analytic_doublyL(1,100,gamma,mu)
array([[  2338.74031419 +0.00000000e+00j,   4306.53455680 +0.00000000e+00j,
          6344.00998738 -3.26265223e-55j],
       [  1270.09459348 +0.00000000e+00j,   2338.74031419 +0.00000000e+00j,
          3445.22764544 +1.30506089e-54j],
       [  6152.19222400 +0.00000000e+00j,  11328.58926318 +0.00000000e+00j,
         16688.28671422 -4.07831529e-56j]])
```

Figure 2: $\mathbf{L}^{100}$ by analytic algorithm

$A_j^{(s)}$ defined by $A_j^{(s)} = \gamma_{r-1}v_j^{(s)} + \cdots + \gamma_j v_{r-1}^{(s)}$ for $0 \leq j \leq r-1$. Finally, to compute $\mathbf{L}^n$ with $n \geq r$ we use the expression (2.17). Algorithm 3, detailed in Appendix C and summarized in Table 3 below, gives the steps necessary for computing the entries of the powers of the doubly Leslie matrices, using the derivative method.

Table 3: Derivative method

Input: The r $\times$ r matrices $\mathbf{L}$, $\mathbf{P}$ and $\mathbf{P}^{-1}$. The integer n. Output: The r $\times$ r matrix $\mathbf{L}^n$
Step1 to 5: The same steps as those in Algorithm 1
Step6: Computing eigenvalues and their multiplicities
Step7: Computing $f_{i,n}(x)$ using equation $f_{i,n}(x) = \frac{x^{n-1}}{\prod_{k=1,k\neq i}^{l}(x-\lambda_k)^{m_k}}$
Step8: Computing $A_j^{(s)}$ using equation $A_j^{(s)} = \gamma_{r-1}v_j^{(s)} + \cdots + \gamma_j v_{r-1}^{(s)}$ for $0 \leq j, s \leq r-1$
Step 9: Computing $\mathbf{L}^n$ from (2.17)

5.4 Numerical examples and elapsed time comparison

The results given in this subsection were obtained using Python 2.7.4 on a 2.60GHz computer. As an illustration, we consider the doubly Leslie matrix used in example 3.3. Suppose we need to compute $\mathbf{L}^{100}$, then using the three proposed algorithms and the usual function *numpy.linalg.matrix_power* existing in Python, we have the following results given in Fig. 1, Fig. 2 and Fig. 3

```
>>> power_matrix_derivative_method(1,100,gamma,mu)
array([[  2338.74031419 +6.77626358e-21j,   4306.53455680 +6.77626358e-21j,
          6344.00998738 +1.17549435e-38j],
       [  1270.09459348 -1.35525272e-20j,   2338.74031419 +5.87747175e-39j,
          3445.22764544 +2.93873588e-39j],
       [  6152.19222400 +5.42101086e-20j,  11328.58926318 +7.34683969e-40j,
         16688.28671422 +2.16840434e-19j]])
>>>
```

Figure 3: $\mathbf{L}^{100}$ by derivative algorithm

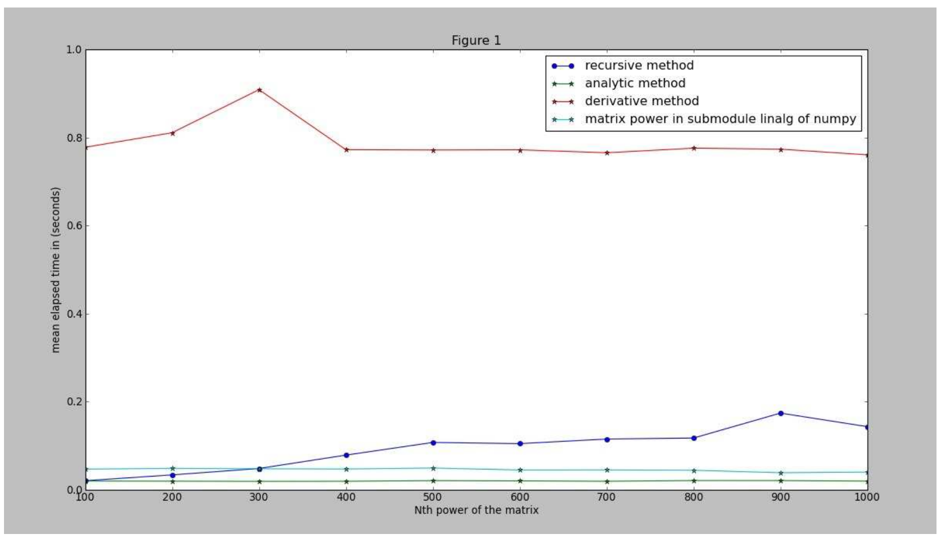

Figure 4: Mean elapsed time by four algorithms over 100 trials in (seconds) for computing the $n-th$ power of the matrix considered in Example 3.3

The outcomes of the three proposed algorithms give the exact values as the usual function *numpy.linalg.matrix_power* existing in python.

To perform reasonable comparative computational studies, we compare the mean elapsed time of the three proposed algorithms with the usual function *matrix_power* existing in the sub-module linalg of numpy module in Python.

The mean elapsed time by the three proposed algorithms after 100 trials are given in Fig. 4 and are compared with the mean elapsed time of the usual function *matrix_power* existing in Python. We can observe that the analytic algorithm provides the best mean elapsed time.

6 Discussion

In the present work, we have first developed explicit formulas for entries of the powers of doubly Leslie matrix. More exactly, these formulas are based on three methods, namely recursive, analytic, and derivative. These expressions allowed us to extract three new formulas concerning the entries of the powers of the doubly companion matrix, which does not exist in the literature.

For these methods, we have proposed three algorithms for computing the entries of the powers of the doubly Leslie matrices. Moreover, a comparison of the mean elapsed time of these three algorithms is provided as shown in Figure 1. Note that, these algorithms are also valuable for computing the entries of the powers of the doubly companion matrices $\mathbf{L} = \mathbf{L}([a_i], [1], [b_j])$, by taking $s_j = 1$ for $0 \leq j \leq r-2$, Leslie matrix $\mathbf{L} = \mathbf{L}([a_i], [s_j], [0])$ by taking $b_j = 0$ for $0 \leq j \leq r-2$, and companion matrix $\mathbf{L} = \mathbf{L}([a_i], [1], [0])$. Two algorithms have been proposed to compute the entries of powers of companion matrix using successive products (see [7]).

To the best of our knowledge, algorithms for computing entries of powers of doubly companion matrix and doubly Leslie matrix are not current in literature.

7 Concluding remarks and perspectives

In this study, we have shown the fundamental role of sequences (2.6) in computing the powers of the doubly Leslie matrices. The main idea used here consists of transforming doubly Leslie matrix into companion matrix then, using the family of sequences (2.6), we have computed the entries of the powers of the doubly Leslie matrices. Moreover, the analytic and the derivative properties of the linear recurrence relations (1.2) had contributed to elaborate more formulas for the entries of the powers of the doubly Leslie matrices. Also, the entries of the powers of the doubly companion matrices are established.

Furthermore, three algorithms were proposed for computing the entries of the powers of doubly Leslie matrices. Note that these algorithms can be also used for computing the entries of the powers of the usual Leslie matrices, the companion matrices, and the doubly companion matrices.

On the other hand, compared with methods of the literature, we can say that the results of this study are not commune. Finally, it seems for us that, the preceding algorithms can be also adapted to the treatment of other population dynamical systems of Leslie type, and more generally of

Lefkovitch type. Recently, the nonlinear Leslie matrix model approach and the asymptotic behavior of its solutions, have been studied in [9], taking into account the effects of the stress factor on the biological population. So, we can also use our approach for studying the nonlinear Leslie matrix model given in [9].

Appendix A.

```
from __future__ import division
import numpy as np
import numpy.linalg as alg
from numpy.linalg import inv
from scipy.linalg import circulant
from scipy.linalg import companion
from numpy.linalg import matrix_power
import time

#function witch generate doubly_leslie matrix(expression 3 in the article)
def doubly_leslie(f, s, b):
    f = np.atleast_1d(f)
    s = np.atleast_1d(s)
    b = np.atleast_1d(b)
    if f.ndim != 1:
        raise ValueError("Incorrect shape for f.  f must be one-dimensional")
    if s.ndim != 1:
        raise ValueError("Incorrect shape for s.  s must be one-dimensional")
    if b.ndim != 1:
        raise ValueError("Incorrect shape for b.  b must be one-dimensional")
    if f.size != s.size + 1:
        raise ValueError("Incorrect lengths for f and s.  The length"
                         " of s must be one less than the length of f.")
    if f.size != b.size + 1:
        raise ValueError("Incorrect lengths for f and b.  The length"
                         " of b must be one less than the length of f.")
    if s.size == 0:
        raise ValueError("The length of s must be at least 1.")
    if b.size == 0:
        raise ValueError("The length of b must be at least 1.")
    tmp = f[0] + s[0] + b[0]
    n = f.size
    l = np.zeros((n, n), dtype=tmp.dtype)
    l[0] = f
    l[list(range(1, n)), list(range(0, n - 1))] = s
    l[list(range(1, n)),n-1]=b
    return l

def mat_p(l):
    len_matrix= len(l)
    gamma = np.zeros((len_matrix ,len_matrix))
    for i in range (0,len_matrix):
        for j in range(0,len_matrix):
            prod =1
            if i==j & i!=len_matrix -1:
                for k in range(i,len_matrix -1):
                    prod*=l[k+1][k]
                    gamma[i][i]=1/prod
            if i> j:
                gamma[i][j]=0
            if j>i:
                if j-i-1==0:
                    fois=1
                    for u in range(i,len_matrix -1):
                        fois*=l[u+1][u]
                        gamma[i][j]=-(l[len_matrix -1])[len_matrix -1]/fois
                else:
                    produit=1
                    fois=1
                    for s in range (len_matrix -j+i,len_matrix -1):
                        produit*=l[s+1][s]
                    for h in range(i,len_matrix -1):
                          fois*=l[h+1][h]
```

```
                    gamma[i][j]=-(l[len_matrix-j+i][len_matrix-1]*produit)/fois
            else :
                gamma[len_matrix-1][len_matrix-1]=1
    return gamma

def inv_mat_p(gamma):
    mu= inv(gamma)
    return mu

def matrix_compan(l,gamma,mu):
    e= np.dot(mu,l)
    b=np.dot(e,gamma)
    return b

def delta(i, s):
    if i == s:
        return 1
    else:
        return 0

def v3(n,s,l,b):
    len_matrix = len(l)
    gamma2 = np.zeros(len_matrix)
    for i in range(0, len_matrix):
        gamma2[i] = b[0][i]

    V = np.zeros(n+1)
    for i in range(0,n+1):
        V[i] = delta(i,s)
    for j in range (len_matrix-1, n):
        for k in range (0,len_matrix):
            V[j+1] += gamma2[k]*V[j-k]
    return V


def power_matrix_doublyL(l,n,gamma,mu):
    len_matrix= len(l)
    power_l=np.zeros((len_matrix,len_matrix))

    for i in range(0,len_matrix):
        for j in range(0,len_matrix):
            som=0
            for p in range(i,len_matrix):
                for k in range(0,j+1):
                    som+=gamma[i][p]*mu[k][j]*v3(n+len_matrix-p-1,len_matrix-1-k,l,b)[n+len_matrix-p-1]
                    power_l[i][j]=som
    return power_l
```

Appendix B.

```
def beta_vanderm(l,s):
    len_matrix=len(l)
    beta=np.zeros(len_matrix)
    mat=np.zeros((len_matrix,len_matrix),dtype=complex)
    vect=np.zeros(len_matrix)
    vect2=np.zeros(len_matrix,dtype=complex)
    evals, evecs = la.eig(l)
    unique_elements, counts_elements = np.unique(evals, return_counts=True)
    for n in range(0,len_matrix):
        for f in range(0,len_matrix):
            vect[f]=delta(s,f)
        result = []
        for k in range(0,len(counts_elements)):
            for j in range(0,counts_elements[k]):
                result.append((n**j)*unique_elements[k])
    mat=np.vander(result)
    mat2=np.rot90(mat)
    beta=np.linalg.solve(mat2,vect)
    return beta

def power_matrix_analytic_doublyL(l,n,gamma,mu):
    len_matrix= len(l)
    power_l=np.zeros((len_matrix,len_matrix),dtype=np.complex_)

    for i in range(0,len_matrix):
        for j in range(0,len_matrix):
```

```
            som=0
            for p in range(i,len_matrix):
                for k in range(0,j+1):
                    f=beta_vanderm(l,len_matrix-1-k)
                    for d in range(1,len(unique_elements)+1):
                        for u in range(0,counts_elements[d-1]):
                            som+=gamma[i][p]*mu[k][j]*f[d-1]*((n+len_matrix-p-1)**u)*((
     unique_elements[d-1])**(n+len_matrix-p-1))
                            power_l[i][j]=som
    return power_l
```

Appendix C

```
def mul_valpropre(l):
    eigenvalues = np.linalg.eigvals(l)
    test, multiplicity = np.unique(eigenvalues, return_counts=True)
    return test,multiplicity

def fun_f(i,n):
    x=sym.symbols('x')
    y=1
    c=1
    for k in range(0,len(test)):
        if k!=i-1:
            y*=pow((x-test[k]),multiplicity[k])
            c=pow(x,n-1)/y
    return c

#fonction calcul ajs verifiee
def fun_a(j,s,l,b):
    sommer=0
    len_matrix= len(l)
    gamma2 = np.zeros(len_matrix)
    for i in range(0, len_matrix):
        gamma2[i] = b[0][len_matrix-i-1]
    #print gamma2
    for o in range(j,len(l)):
        #print v3(o,s,l,b)[len(v3(o,s,l,b))-1]
        #print gamma2[o-j] , v3(o,s,l,b)[len(v3(o,s,l,b))-1]
        sommer+=gamma2[o-j]*v3(o,s,l,b)[len(v3(o,s,l,b))-1]
    return sommer

def fact(n):
    """fact(n): calcule la factorielle de n (entier >= 0)"""
    if n<2:
        return 1
    else:
        return n*fact(n-1)


#derivative method (expression 37)
def power_matrix_derivative_method(l,n,gamma,mu):
    len_matrix= len(l)
    power_l=np.zeros((len_matrix,len_matrix),dtype=np.complex_)
    x= sym.symbols('x')
    for i in range(0,len_matrix):
        for j in range(0,len_matrix):
            additionner=0
            for p in range(i,len_matrix):
                for k in range(0,j+1):
                    for u in range(1,len(test)+1):
                        for d in range(0,len_matrix):
                            additionner+=gamma[i,p]*mu[k,j]*fun_a(d,len_matrix-1-k,l,b)*sym.diff(
     fun_f(u,n+len_matrix-p-1-d),x,multiplicity[u-1]-1).subs({x:test[u-1]})/fact(multiplicity[u-1]-1)
                            #print additionner.dtype
                            power_l[i,j]=additionner
    return power_l
```

Acknowledgements. The authors thank an anonymous referee for useful suggestions. The authors are grateful to present their sincere thanks to the LAGA. The second author expresses his sincere thanks to the INMA and the UFMS for their valuable support and encouragement.

References

[1] B. El-Wahbi, M. Mouline, M. Rachidi, Solving nonhomogeneous recurrences relations by matrix method, Fibonacci Quarterly, **40,** no. 2, (2002), 109–117.

[2] Erin N. Bodine, Suzanne Lenhart, Louis J. Gross, Mathematics for The Life Sciences, Princeton University Press, 2014.

[3] F. Dubeau, W. Motta, M. Rachidi, O. Saeki, On weighted r-generalised Fibonacci sequences, Fibonacci Quarterly, **35**, (1997), 102–110.

[4] H. Pickmann-Soto S. Arela-Pérez, Hans Nina , Elvis Valero, Inverse maximal eigenvalues problems for Leslie and doubly Leslie matrices, Linear Algebra and its Applications, **592,** (2020), 93–112.

[5] J. C. Butcher, P. Chartier, The effective order of singly-implicit Runge-Kutta methods, Numer. Algorithms, **20,** (1999), 269–284.

[6] J. C. Butcher, W. M. Wright, Applications of doubly companion matrices, Applied Numerical Mathematics, **56,** (2006), 358–373.

[7] K. Aydin, On a power of the companion matrix, Süleyman Demirel Üniversitesi Fen Edebiyat Fakültesi Fen Dergisi, **3,** no. 2, (2008), 230–235.

[8] L. P. Lefkovitch, Study of population growth in organisms grouped by stages, Biometrics, **21,** (1965), 1–18. doi: 10.2307/2528348.

[9] L. Monte, Nonlinear Leslie models for the assessment of the effects of stressors on the development of wild populations: Reviewing of the basic properties, Journal of Interdisciplinary Mathematics, **21,** no. 1, (2018), 83–109.

[10] M. Mouline, M. Rachidi, Application of Markov chains properties to r-generalized Fibonacci sequences, Fibonacci Quarterly, **37,** (1999), 34–38.

[11] P. H. Leslie, On the use of matrices in certain population mathematics, Biometrica, **33,** (1945), 183–212.

[12] R. Ben Taher, M. Rachidi, Solving some generalized Vandermonde systems and inverse of their associate matrices via new approaches for the Binet formula, Applied Mathematics and Computation, **290,** (2016), 267–280.

[13] R. Ben Taher, M. Rachidi, On the matrix powers and exponential by the r-generalized Fibonacci sequences methods, The Companion matrix case, Linear Algebra and its Applications, **370,** (2003), 341–353.

[14] R. Ben Taher, N. Naassi, M. Rachidi, On the Leslie matrices, Fibonacci sequences and population dynamics, Journal of Discrete Mathematical Sciences & Cryptography, **20,** (2017), 565–594.

[15] R. P. Stanley, Enumerative combinatorics, **I,** Cambridge University Press, 1997.

[16] W. Wanicharpichat, Explicit minimum polynomial, eingenvalues and inverse formula of Doubly Leslie matrix, Journal of applied mathematics & informatics, **33,** nos. 3-4 , (2015), 247–260.

[17] W. Wanicharpichat, Nonderogatory of Sum and Product of Doubly Companion Matrices, Thai Journal of Mathematics, **9,** no. 2, (2011), 337–348.