Sohag Journal of Sciences
An International Journal

Algorithm for Computing Exact Solution of the First Order Linear Differential System

Mohammed Shehata [1], ***A. A. Khalil*** [2]

[1] **Department of Basic Science, Bilbeis Higher Institute for Engineering, Ministry of Higher Education, Egypt**
[2] **Department of Mathematics, Faculty of Science, Sohag University, Egypt**

Received: 22 May 2022, Revised: 22 June 2022, Accepted: 2 July,
Published online: 1 sept. 2022

Abstract: In this paper, by using Jordan decomposition method, we develop an algorithm and convert it into a Maple procedure to get the exact solution of nonhomogeneous first order linear differential systems. With this procedure, we get the exact solution by just entering the system parameters.

Keywords: First order linear system, matrix exponential, Jordan decomposition, analytic solution, Maple procedure.

1 Introduction

Differential equations play an important role in the understanding of physical sciences. Many differential equations arise from problems in physics, engineering, and other sciences, and these equations serve as mathematical models for solving numerous problems in science and engineering [1-9].Numerous numerical methods exist for solving differential equations, such as Taylor, Picard, Euler, Runge-Kutta and transformation methods.

In this paper, we present a Jordan decomposition method for solving the following nonhomogeneous first order linear differential systems :

$$\begin{aligned}
y'_1(t) &= a_{11}\, y_1(t) + a_{12}\, y_2(t) + \cdots + a_{1n}\, y_n(t) + u_1(t),\\
y'_2(t) &= a_{21}\, y_1(t) + a_{22}\, y_2(t) + \cdots + a_{2n}\, y_n(t) + u_2(t),\\
&\vdots\\
y_n(t) &= a_{n1}\, y_1(t) + a_{n2}\, y_2(t) + \cdots + a_{nn}\, y_n(t) + u_n(t),
\end{aligned}$$

with initial conditions $y_1(t_0) = y_{01}, \dots, y_n(t_0) = y_{0n}$.
In matrix and vector notations, we write it as

$$\textbf{FOLDS}\quad \begin{cases} y'(t) = A\, y(t) + u(t), \\ \text{with initial conditions} \\ y(t_0) = y_0, \end{cases}$$

where $y(t) = [y_1(t), \dots, y_n(t)]^T$,
$u(t) = [u_1(t), \dots, u_n(t)]^T$, $y_0 = [y_{01}, \dots, y_{0n}]^T$ and $A = [a_{ij}]$ is $n \times n$ constant matrix.

We know that the analytic solution of FOLDS with continuous parameters is given by

$$y(t) = e^{At}\, y_0 + e^{At} \int_{t_0}^{t} e^{-A\tau}\, u(\tau) d\tau \quad (1)$$

Jordan decomposition method is based on decomposing matrix exponentials into a product of matrices This allows us to easily find the exact solution to the system throat a simple formula.

After the introduction, the paper is divided into a six sections. In section 2, we introduce the definition and properties of matrix exponentials. Then, in section 3, we provide an overview of Jordan decompositions for matrices. Section 4 presents our algorithm. In section 5, we obtain the Maple procedure. In section 6 we apply the algorithm to find the exact solution to some linear differential systems. Finally, we concluded our result

2. Matrix exponential and its properties

Definition 1 *(* **Matrix exponential***) For each* $n \times n$ *complex matrix* A*, define the exponential of* A *to be the matrix*

$$e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!} = I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \dots .$$

This sum converges for all complex matrices A. The definition 1 immediately reveals many other familiar properties. The following lemma is easy to prove from the definition 1.

* Corresponding author E-mail: amalaboelwafa@science.sohag.edu.eg

Lemma 1 *Let A, B be $n \times n$, complex matrices.*

1. If 0 denotes the zero matrix, then $e^0 = I$, the identity matrix.
2. $A^m e^A = e^A A^m$ for all integers m.
3. $(e^A)^T = e^{A^T}$.
4. If $AB = BA$ then $Ae^B = e^B A$ and $e^A e^B = e^B e^A = e^{A+B}$.

Lemma 2 *Let A be a complex $n \times n$ matrix and let t, s be a real scalar variables. Then*

$$e^{A(t+s)} = e^{At} e^{As}$$

Proof. From the definition 1, we have

$$\begin{aligned} e^{At} e^{As} &= \left(I + At + \frac{A^2 t^2}{2!} + \cdots.\right)\left(I + As + \frac{A^2 s^2}{2!} + \ldots.\right) \\ &= \left(I + A(t+s) + \frac{A^2 (t+s)^2}{2!} + \cdots.\right) = e^{A(t+s)} \end{aligned}$$

Lemma 3 If $A = D = diag(\lambda_1, \lambda_2, \ldots, \lambda_n)$ $= \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & \lambda_n \end{bmatrix}$, then $e^A = diag(e^{\lambda_1}, e^{\lambda_2}, \ldots, e^{\lambda_n})$

Proof. If we directly evaluate the sum of the infinite series in the definition of e^{At}, we find that the entry of e^{At} is given by

$$\sum_{k=0}^{\infty} \frac{\lambda_i^k t^k}{k!} = e^{\lambda_i t}.$$

Lemma 4 *Let A and P be complex $n \times n$ matrices, and suppose that P is invertible. Then*

$$e^{P^{-1}AP} = P^{-1} e^A P$$

Proof. Recall that, for all integers $m \geq 0$, we have $(P^{-1}AP)^m = P^{-1} A^m P$. The definition 1 then yields

$$\begin{aligned} e^{P^{-1}AP} &= I + P^{-1}AP + (P^{-1}\tfrac{A^2}{2!}P) + (P^{-1}\tfrac{A^3}{3!}P) + \ldots. \\ &= P^{-1}\left(I + A + \tfrac{A^2}{2!} + \tfrac{A^3}{3!} + \ldots.\right)P = P^{-1} e^A P \end{aligned}$$

We can now prove a fundamental theorem about matrix exponentials.

Theorem 1 *Let A be a complex square matrix, and let t, be a real scalar variable.*

If $f(t) = e^{At}$ then $f'(t) = Ae^{At}$.

Proof. Applying lemma 2 to the limit definition of derivative yields

$$f'(t) = \lim_{h \to 0} \frac{e^{A(t+h)} - e^{At}}{h} = e^{At}\left(\lim_{h \to 0} \frac{e^{Ah - I}}{h}\right)$$

Applying the definition 1 to $e^{Ah} - I$ then gives us

$$\begin{aligned} f'(t) &= e^{At}\left(\lim_{h \to 0} \tfrac{1}{h}\left[Ah + \tfrac{A^2 h^2}{2!} + \tfrac{A^3 h^3}{3!} + \cdots.\right]\right) \\ &= e^{At} A = A e^{At}. \end{aligned}$$

3. Jordan decomposition

This section contains linear algebraic results [10-13].

Definition 2 (A Jordan block (of size k)) $J_k(\lambda) \in C^k \times C^k$ *is the upper triangular matrix*

$$J_k(\lambda) = \begin{bmatrix} \lambda & 1 & & 0 \\ & \ddots & \ddots & \\ & & \lambda & 1 \\ 0 & & & \lambda \end{bmatrix}$$

From definition 1 and lemma 1, we can compute $e^{J_k(\lambda t)}$.

For example, let us take $k = 3$; $J_3(\lambda) = \begin{bmatrix} \lambda & 1 & 0 \\ 0 & \lambda & 1 \\ 0 & 0 & \lambda \end{bmatrix}$,

we can write $J_3(\lambda) = \lambda I + N$ where $N = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$.

Direct calculation shows that

$$N^2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } N^3 = 0.$$

But then the transition matrix e^{Nt} is easily evaluated to be

$$e^{Nt} = I + Nt + \frac{N^2 t^2}{2!} = \begin{bmatrix} 1 & t & \frac{t^2}{2!} \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}.$$

Since λI commutes with N, we can write, $e^{J_3(\lambda t)} = e^{(\lambda I + N)t} = e^{\lambda I t} e^{Nt} = e^{\lambda t} \begin{bmatrix} 1 & t & \frac{t^2}{2!} \\ 0 & 1 & t \\ 0 & 0 & 1 \end{bmatrix}$. In the same way, we can compute $e^{J_k(\lambda t)}$ for any arbitrary k. Hence,

$$e^{J_k(\lambda t)} = e^{\lambda t} \begin{bmatrix} 1 & t & \frac{t^2}{2!} & \cdots & \frac{t^{k-1}}{(k-1)!} \\ & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & \frac{t^2}{2!} \\ & & & \ddots & t \\ 0 & & & & 1 \end{bmatrix} \quad (2)$$

Theorem 2 (Jordan decomposition) *Every complex matrix $A \in C^n \times C^n$ is similar to a block diagonal matrix J. That is, for every matrix A there exists an invertible matrix P so that $A = PJP^{-1}$, where*

$$J = \begin{bmatrix} J_{k_1}(\lambda_1) & & 0 \\ & \ddots & \\ 0 & & J_{k_m}(\lambda_m) \end{bmatrix}$$

and $\lambda_1, \lambda_2, \ldots, \lambda_m$ are the eigenvalues of A with multiplicity $k_1, k_2, \dots, k_m$ and $k_1 + k_2 + \cdots + k_m = n$.

Corollary 1 *When $m = n$ then $J = D$. (A is diagonalizable). Recall from linear algebra, that an eigenvector v associated with the eigenvalue λ for the matrix A satisfies the equation $(A - \lambda I)v = 0$ and in this case, the nonsingular matrix $p = [v_1, v_2, \dots, v_n]$, consists of n linearly independent eigenvectors $v_1, v_2, \dots, v_n$ of A.*

Definition 3 *If A an $n \times n$ matrix, a generalized eigenvector of A corresponding to the eigenvalue A is a nonzero vector v satisfying $(A - \lambda I)^p v = 0$ for some positive integer p. Equivalently, it is a nonzero element of the nullspace of $(A - \lambda I)^p$.*

The aim of generalized eigenvectors was to enlarge a set of linearly independent eigenvectors to make a basis.

Theorem 3 *If A an $n \times n$ matrix and λ is an eigenvalue with multiplicity k, then the set of generalized eigenvectors for λ consists of the nonzero elements $v_1, v_2, \dots, v_k$ of nullspace $(A - \lambda I)^k$. In other words, we need to take at most k powers of $(A - \lambda I)$ to find all of the generalized eigenvectors for λ. We can calculate $v_1, v_2, \dots, v_k$ from the chain*

$$(A - \lambda I)v_1 = 0, \quad (A - \lambda I)v_2 = v_1, \quad \dots,$$
$$(A - \lambda I)v_k = v_{k-1}.$$

Corollary 2 *If P is the matrix whose columns are n linearly independent of generalized eigenvectors $v_1, v_2, \dots, v_n$ of an $n \times n$ matrix A arranged in chains, then $A = PJP^{-1}$. It is unique up to a rearrangement of the Jordan blocks.*

4. Algorithm for Exact Solution

Based on the previous sections and from (1), we can deduce the following theorem which transform to algorithm (table 1) for computing exact solution:

Theorem 4 (**Formula of exact solution of FOLDS**) *If $u(t)$ is $n-$ vector continuous function on an open interval $I \subset \Re$, A is $n \times n$ constant matrices and if y_0 is any constant vector and t_0 is any constant in I, then there exist only one function y defined on an interval $\hat{I} \subset I$ with $t_0 \in \hat{I}$ solution of* **FOLDS**, *given by*

$$y(t) = P\left[e^{J(t-t_0)}\, p^{-1}\, y_0 + \int_{t_0}^{t} e^{J(t-\tau)}\, p^{-1}\, u(\tau)d\tau\right] \qquad (3)$$

Table 1: Algorithm for exact solution

Step 1	Input $A, u(t), y_0, I, t_0$
Step 2	Calculate the eigenvalues λ_i of the matrix A
Step 3	Calculate the eigenvectors (in the case of multiple eigenvalues from theorem 3) generalized eigenvectors and construct the corresponding Jordan matrix J and construct the nonsingular linear transformation matrix P and its inverse P^{-1}
Step 4	Compute $e^{J\,t}$
Step 5	Compute $y_h(t) = P\, e^{J\,(t-t_0)}\, P^{-1}$
Step 6	Compute $y_p(t) = P \int_{t_0}^{t} e^{J\,(t-\tau)} P^{-1}\, u(\tau)d\tau$
Step 7	Compute $y(t) = y_h(t) + y_p(t)$
Step 8	Plot $y(t)$

5. The procedure

Procedure **Solu1** *computes and plots the exact solution for nonhomogeneous first order linear system with constant coefficients.*

Output:

Plotting solutions if the conditions for finding them are met, or printing an objection sentence if the conditions are not met.

Syntax:

Solu1 $(a, \mathrm{b}, \mathrm{t}_0, A, u, y_0)$;

Input:

a - A real number represents the start of the solution interval;

b - A real number represents the end of the solution interval;

t_0 - A real number in $[a, b]$ which represents the initial time;

A - A square matrix represents system coefficients;

u - Vector of functions represents the nonhomogeneous term;

y_0 - Vector of real numbers represents the initial values..

Definition:

```
> Solu1:=proc(a,b,t0,
A::'Matrix'(square),u::Vector,y0::Vector)
> local u1,m,J1,P,P1,eJ,eJ1,yh,yp1,yp2,x,yp,z,y,i;
> u1:=convert(u,list);m:=nops(u1);
> J1, P := LinearAlgebra[JordanForm](A, output = ['J',
'Q']);
> P1:=LinearAlgebra[MatrixInverse](P);
>
eJ:=LinearAlgebra[MatrixExponential](J1,t);eJ1:=subs(
t = -t, eJ);
```

```
> yh := simplify(P.eJ.P1.y0);
> yp1 := eJ1.P1.u;yp2:=subs(t = x, yp1);yp:=P.eJ.map(int,yp2,x=a..t);
> z:=simplify(yh+yp); z:=convert(z,list);
> if seq(iscont(u1[i], t = a .. b),i=1..m)=seq(true,i=1..m) then for i to m do print(y[i](t));print(plot(z[i],t=a..b,color=red,thickness=2));end do;  else print( "u vector is not continous on "(a,b));fi;
> end :
```

In our procedure Solu1, we have invoked some of the pre-existing Maple Linear Algebra procedures, such as JordanForm, MatrixInverse and MatrixExponential.

6. Application

We now give two applications to illustrate our procedure.

Application I: Solve the following differential system:

$$\begin{aligned} y'_1(t) &= 3y_1 + y_2 + 2t, \quad 0 < t < 1, \\ y'_2(t) &= y_2 - y_1 + t, \\ y_1(0) &= 1, \quad y_2(0) = 1. \end{aligned}$$

1. Determine $A,\ u(t), y_0, I, t_0$;

$$A = \begin{bmatrix} 3 & 1 \\ -1 & 1 \end{bmatrix}, \quad u(t) = \begin{bmatrix} 2t \\ t \end{bmatrix},$$

$$y_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad I = (0,1), \quad t_0 = 0.$$

2. We can show that $\lambda = 2$ is the only eigenvalue of A with multiplicity 2, so $J = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}$

3. Solve $(A - 2I)v_1 = 0$ and $(A - 2I)v_2 = v_1$ to get the generalized eigenvectors $v_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ and $v_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ then, we can evaluate p and calculate p^{-1};

$$p = \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix} \Rightarrow p^{-1} = \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix};$$

4. Compute e^{Jt}; $\quad e^{Jt} = \begin{bmatrix} e^{2t} & te^{2t} \\ 0 & e^{2t} \end{bmatrix}$

5. Calculate $y_h = pe^{Jt}p^{-1}\, y_0$;

$$\begin{aligned} y_h &= \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix} . \begin{bmatrix} e^{2t} & te^{2t} \\ 0 & e^{2t} \end{bmatrix} . \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix} . \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\ &= e^{2t} \begin{bmatrix} 1 + 2\,t \\ 1 - 2\,t \end{bmatrix} \end{aligned}$$

6. Compute $y_p(t) = P \int_0^t e^{J(t-\tau)}\, u(\tau) d\tau$;

$$\begin{aligned} y_p &= \begin{bmatrix} 1 & 1 \\ -1 & 0 \end{bmatrix} . \int_0^t \begin{bmatrix} e^{2(t-\tau)} & (t-\tau)e^{2(t-\tau)} \\ 0 & e^{2(t-\tau)} \end{bmatrix} . \begin{bmatrix} 0 & -1 \\ 1 & 1 \end{bmatrix} . \begin{bmatrix} 1 \\ 1 \end{bmatrix} d\tau \\ &= \begin{bmatrix} \frac{1}{4}\, e^{2t}(3\,t - 1) + \frac{1}{4}(t - 1) \\ \frac{1}{4}\, e^{2t}(-3\,t + 1) - \frac{5}{4}\, t - 1 \end{bmatrix} \end{aligned}$$

7. Compute $y = y_h(t) + y_p(t)$;

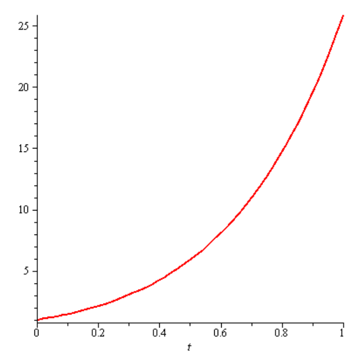
$$\begin{aligned} y &= e^{2t} \begin{bmatrix} 1 + 2\,t \\ 1 - 2\,t \end{bmatrix} + \begin{bmatrix} \frac{1}{4}\, e^{2t}(3\,t - 1) + \frac{1}{4}(t - 1) \\ \frac{1}{4}\, e^{2t}(-3\,t + 1) - \frac{5}{4}\, t - 1 \end{bmatrix} \\ &= \begin{bmatrix} \frac{11}{4}\, t\, e^{2t} + \frac{3}{4}\, e^{2t} - \frac{1}{4}\, t + \frac{1}{4} \\ \frac{-11}{4}\, t\, e^{2t} + 2\, e^{2t} - \frac{5}{4}\, t - 1 \end{bmatrix} \end{aligned}$$

This is the same result using the procedure with

```
>
A := Matrix([[3, 1], [-1, 1]]); y0 := Vector([1, 1]); u
     := Vector([2 t, t]);
```

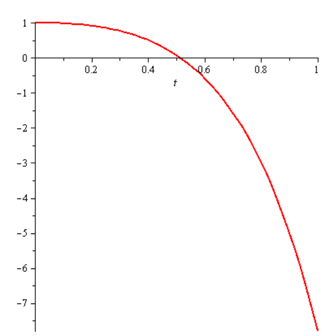
$$A := \begin{bmatrix} 3 & 1 \\ -1 & 1 \end{bmatrix} \quad y0 := \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad u := \begin{bmatrix} 2\,t \\ t \end{bmatrix}$$

```
> Solu1(0, 1, 0, A, u, y0);
```



$y_1(t)$

$y_2(t$



Application II: Solve the following differential system:

$$\begin{aligned} y'_1(t) &= y_2, \quad 0 < t < 3, \\ y'_2(t) &= \tfrac{3}{4}\, y_1 + y_2 + y_4, \\ y'_3(t) &= y_1 - \tfrac{3}{4}\, y_4, \\ y'_4(t) &= -y_3 - y_4, \\ y_1(0) &= 0,\ y_2(0) = 0,\ y_3(0) = 1,\ y_4(0) = 1. \end{aligned}$$

1. Determine $A, u(t), y_0, I, t_0$;

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{3}{4} & 1 & 0 & 1 \\ 1 & 0 & 0 & -\frac{3}{4} \\ 0 & 0 & -1 & -1 \end{bmatrix}, \quad u(t) = \mathbf{0},$$

$$y_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \quad I = (0,3), \quad t_0 = 0.$$

2. Solve the characteristic equation $|A - \lambda I| = 0 \Rightarrow (4\lambda^2 - 5) = 0$ to get two real eigenvalues of A; with multiplicity 2;

$$\lambda_1 = -\frac{\sqrt{5}}{2}, \quad \lambda_2 = -\frac{\sqrt{5}}{2}, \quad \lambda_3 = \frac{\sqrt{5}}{2}, \quad \lambda_4 = \frac{\sqrt{5}}{2}$$

then

$$J = \begin{bmatrix} -\frac{\sqrt{5}}{2} & 1 & 0 & 0 \\ 0 & -\frac{\sqrt{5}}{2} & 0 & 0 \\ 0 & 0 & \frac{\sqrt{5}}{2} & 1 \\ 0 & 0 & 0 & \frac{\sqrt{5}}{2} \end{bmatrix}$$

3. Solve $(A - \lambda_1 I)v_1 = 0$ to get the first generalized eigenvectors

$$v_1 = \begin{bmatrix} \frac{3}{20} + \frac{\sqrt{5}}{20} \\ -\frac{1}{8} - \frac{3\sqrt{5}}{40} \\ \frac{1}{20} \\ \frac{1}{5} + \frac{\sqrt{5}}{10} \end{bmatrix}$$

Solve $(A - \lambda_2 I)v_2 = v_1$ to get the second generalized eigenvectors

$$v_2 = \begin{bmatrix} \frac{1}{2} + \frac{\sqrt{3}}{50} \\ -\frac{\sqrt{5}}{5} \\ -\frac{9\sqrt{5}}{20} \\ \frac{2\sqrt{5}}{25} \end{bmatrix}$$

Solve $(A - \lambda_3 I)v_3 = 0$ to get the third generalized eigenvectors

$$v_3 = \begin{bmatrix} \frac{3}{20} - \frac{\sqrt{5}}{20} \\ -\frac{1}{8} + \frac{3\sqrt{5}}{40} \\ \frac{1}{20} \\ \frac{1}{5} - \frac{\sqrt{5}}{10} \end{bmatrix}$$

Solve $(A - \lambda_4 I)v_4 = v_3$ to get the fourth generalized eigenvectors

$$v_4 = \begin{bmatrix} \frac{1}{2} - \frac{\sqrt{3}}{50} \\ \frac{\sqrt{5}}{5} \\ \frac{9\sqrt{5}}{20} \\ -\frac{2\sqrt{5}}{25} \end{bmatrix}$$

then

$$P = [v_1, v_2, v_3, v_4]$$

$$= \begin{bmatrix} \frac{3}{20} + \frac{\sqrt{5}}{20} & \frac{1}{2} + \frac{\sqrt{3}}{50} & \frac{3}{20} - \frac{\sqrt{5}}{20} & \frac{1}{2} - \frac{\sqrt{3}}{50} \\ -\frac{1}{8} - \frac{3\sqrt{5}}{40} & -\frac{\sqrt{5}}{5} & -\frac{1}{8} + \frac{3\sqrt{5}}{40} & \frac{\sqrt{5}}{5} \\ \frac{1}{20} & -\frac{9\sqrt{5}}{20} & \frac{1}{20} & \frac{9\sqrt{5}}{20} \\ \frac{1}{5} + \frac{\sqrt{5}}{10} & \frac{2\sqrt{5}}{25} & \frac{1}{5} - \frac{\sqrt{5}}{10} & -\frac{2\sqrt{5}}{25} \end{bmatrix}$$

4. Calculate e^{Jt}; $e^{Jt} = \begin{bmatrix} e^{-\frac{\sqrt{5}}{2}t} & te^{-\frac{\sqrt{5}}{2}t} & 0 & 0 \\ 0 & e^{-\frac{\sqrt{5}}{2}t} & 0 & 0 \\ 0 & 0 & e^{\frac{\sqrt{5}}{2}t} & te^{\frac{\sqrt{5}}{2}t} \\ 0 & 0 & 0 & e^{\frac{\sqrt{5}}{2}t} \end{bmatrix}$

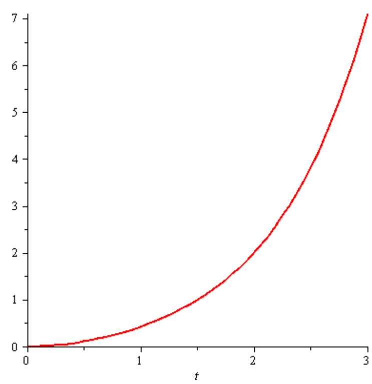
Using the procedure with

\>

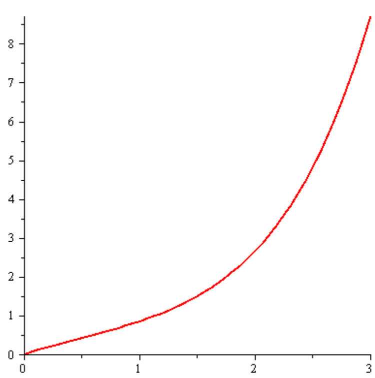
$A := \text{Matrix}\left(\left[[0, 1, 0, 0], \left[\frac{3}{4}, 1, 0, 1\right], \left[1, 0, 0, \frac{-3}{4}\right], [0, 0,\right.\right.$
$\left.\left.-1]\right]\right)$; $y0 := \text{Vector}([0, 0, 1, 1])$; $u := \text{Vector}([0, 0, 0,$

$$A := \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{3}{4} & 1 & 0 & 1 \\ 1 & 0 & 0 & -\frac{3}{4} \\ 0 & 0 & -1 & -1 \end{bmatrix} \quad y0 := \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad u := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

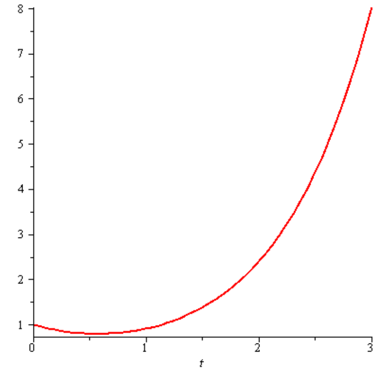
\> $Solu1\,(0, 3, 0, A, u, y0)$;



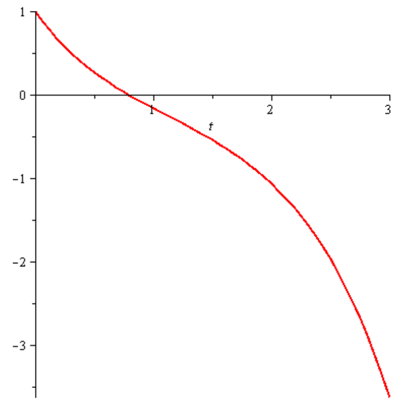
$y_1(t)$



$y_2(t)$



$y_3(t)$



$y_4(t)$

7. Conclusion

In the paper, a Maple procedure was presented for computing and plotting the exact solutions of nonhomogeneous first-order linear differential equations whose coefficients are constant.The advantage of this procedure is that we do not need to enter the system equations and their parameters as we would in Matlab. Our solution is obtained by entering the parameters only as matrices or vectors.

Acknowledgments

The authors are very grateful to the referees for their valuable suggestions, which helped to improve the paper significantly.

References

[1] Abdel-Halim H. I.H. Application to differential transformation method for solving systems of differential equations, *Applied Mathematical Modelling*, 2008 32, 2552–2559.

[2] Agarwal R. P & Jinhai C.. Periodic solutions for first order differential systems, Applied Mathematics Letters, 2010, 23, 337-341.

[3] Higazy. M and Sudhanshu A..Sawi transformation for system of ordinary differential equations with application, *Ain Shams Engineering Journal*, 2021, 12 3173–3182.

[4] Mardanov, M. J, Sharifov,Y. A, Ismayilova, K. E and Zamanova, S. A. Existence and uniqueness of solutions for the system of first-order nonlinear differential equations with three-point and integral boundary conditions, *European Journal of Pure and Applied Mathematics*, 2019, Vol. 12, No. 3, 756-770.

[5] Sankar P. M.; Susmita R. and Biswajit D. ;Numerical Solution of First-Order Linear Differential Equations in Fuzzy Environment by Runge-Kutta-Fehlberg Method and Its Application, *International Journal of Differential Equations*, , 2016 Volume,1-14.

[6] Francesco C. and Farrin P;. Solution of the system of two coupled first-order ODEs with scond-degree polynomial right-hand sides, *Math Phys Anal Geom*, 2021, 24-29.

[7] Mardanov. M.J; Sharifov.Y.A, Ismayilova.K.E, Zamanova.S.A.. Existence and uniqueness of solutions for the system of first-order nonlinear differential equations with three-point and integral boundary conditions. *European Journal of pure and Applied mathematics,* 2019 .Vol. 12, No. 3, 756-770 .

[8] Mohamed A.R.; Abd-El-Aziz E.and Mahmoud N.S.. Numerical solution of system of first-order delay differential equations using polynomial spline functions, *International Journal of Computer Mathematics*, 2006 Vol. 83, No. 12925–937.

[9] Mohammed S.. Exact Solution of First Order Constraints Quadratic Optimal Control Problem, *Advanced Modeling and Optimization,* 2015, Volume 17 , Number 1.

[10] David L.; Steven L. and Judi M.; Linear Algabra and Its Applications, 5th Edition, 2014.

[11] Mohamed K. and Bilal C.; Elementary Proofs of the Jordan Decomposition Theorem for Niloptent Matrices, American *Journal of Mathematics and Statistics*, 2016 ,6(6): 238-241

[12] Sheldon A.; Linear Algebra Done Right, Third edition,*Springer International Publishing*., 2015.

[13] Yulei R.; Jiying W. and Yong C.; Jordan decomposition and geometric multiplicity for a class of non-symmetric Ornstein-Uhlenbeck operators, *Advances in Difference Equations*, 2014, 34.