Steps of Exact and Analytic Solutions of Ordinary Differential Equations using MAHA Integral Transform and Its Applications

MAHA ALSAOUDI[1], AHMAD SHARIEH[2], AHLAM GUIATNI[3,*], GHARIB GHARIB[4]

[1]Applied Science Private University,
JORDAN

[2]University of Jordan,
JORDAN

[3]The University of Jijel,
Algeria

[4]Zarqa University,
JORDAN

Corresponding Author

Abstract: - Transformation such as integral transform is needed to obtain the exact solutions for linear ordinary differential equations (ODEs) with constant coefficients of higher orders. MAHA transformation exact solution of ODEs is simpler and easier than the previous with two parameters. The major steps of this transform are applying the MAHA transform on the given equation followed by taking the inverse transform. The general steps in numerical solutions involve defining the ODE as a function, defining initial conditions and the range of the independent variable, using an appropriate ODE solver function, and calling the solver and plotting the solution using a programming language. The exact and analytical solutions are validated. Both methods are easy and simple to be deployed in computing scientific applications such as nuclear physics and medical applications.

Key-Words: - MAHA transform, Integral transform of two parameters, Inverse of MAHA transform, Medical Applications, Nuclear Physics, Ordinary differential equations.

Received: July 9, 2024. Revised: November 13, 2024. Accepted: December 8, 2024. Published: December 31, 2024.

1 Introduction

One of the most important aspects of mathematics is differential and integral equations. These provide tools to solve problems in our world. Thus, this paper explains the applicability of the MAHA transform in solving ordinary differential equations (ODEs) and validation of the solutions by using a programming language.

The MAHA transform was introduced to address common and intermediate differential equations in the time domain. Transformations such as [1], [2], [3], [4], [5], serve as valuable numerical tools for solving differential equations. Similarly, the MAHA transform and its essential properties are applied to tackle differential equations. Algorithms are needed to implements these transforms and their applications, [6]. Thus, we aim to show how MAHA transform solves ODEs and validate the solution using programming language such MATLAB.

The MAHA transform for capacities in the set A is characterized as in formula (1).

$$A = \{f(t): \textit{there exist } m, l_1, l_2 > 0. \quad |f(t)| < m e^{l_i |t|}, \textit{ if } t \in (-1)^i x\, [0,\infty)\} \tag{1}$$

where m is constant limited number, and l_1, l_2 might be limited or boundless. MAHA fundamental change is signified by the administrator M(.), which is characterized by the vital condition as in formula (2).

$$F(u,v) = M[f(t)] = (uv)^{\beta} \int_0^{\infty} f\left(\frac{t}{u}\right) e^{-\frac{t}{u}}\, dt \quad u,v > 0, \beta \in Z, t \geq 0 \tag{2}$$

The parameters u and v in this essential transform are employed to compute the variable t, the argument of the function f. This indispensable transform exhibits additional connections with [7], [8].

The purpose of this study is to demonstrate the significance of this intriguing novel transform and its efficiency in solving linear differential equations, [9], [10], [11].

The MAHA transform is used in this work to solve ODEs, as well as several practical applications which are considered basic in mathematical physical fields. The main contributions of this paper are: a review of MAHA transformation and its properties to tackle differential equations, design numerical algorithm to solve the differential equations, and validate the exact solution by MAHA transformation and the numerical analysis solution. The programming codes are written in MATLAB. In the methodology section, the steps of MAHA transformation and algorithms of exact and analytical solutions with their implementation are presented. The exact and the numerical analysis solution are validated on some sample of ordinary differential equations.

Exact solution of nuclear physic and medical applications are presented. The exact and numerical solutions are validated.

In section 2, a related work is presented. In section 3, the methodology presents the steps of MAHA transformation for exact solution and analysis methods for the selected equations. Section 4 presents results of solving sample of equations and applications by deploying MATLAB. Section 5 concludes the manuscript.

2 Related Works

Through studying and researching integral transformations and their applications in life, we found that there are many integral transformations with two parameters. The most important of these, according are ZZ transformations, [12].

The value of β, in formula (2), belongs to the set of integers, gives transformation the generality and the preference by using it in important applications. Note that when β = 0, the MAHA transform becomes Shehu transform. When β = -1, the MAHA transform becomes KKAT transform and Quideen transform. The Maha integral transformation is more general transformation than the previous transformations with two parameters. Through this β, we can choose the appropriate number for the application of physics, engineering, or a life application to convert the ordinary differential equation into an easy and simple algebraic equation. Then, by taking the inverse of this transform, we can obtain the exact solution.

The value of β can be referred to as, for example, the number of samples taken in the application. There are many mathematical models in the linear modeling that depend on this β, for example, drug concentration problems, chemical problems, and others.

Through our study and our knowledge of many integral transforms with one parameter or two parameters or more, there is no preference between one transform and another, but there is an important matter, which is that each transform has a use for some applications through after converting differential equations, integral equation or systems into algebraic equations or algebraic system. Conversion is made easier by simplifying that application, [13], [14], [15], [16]. Also, algorithms are essential to compute values in such systems.

MATLAB programming language can be utilized to solve ordinary differential equations analytically and numerically. It offers various numerical methods to solve linear ordinary differential equations (ODEs). The primary function for this purpose is ode45, but there are other functions which support in solving different types of ODEs and specific requirements. The ode45 being the most commonly used for non-stiff problems. The ode45 uses the Dormand-Prince method, a variable-step, variable-order (Runge-Kutta) method. To numerically solve a 2^{nd} ode, for example, you first convert it to a system of first-order ODE.

The general steps to solve ODEs numerically in MATLAB are:

1. Define the ODE as a function.
2. Define initial conditions and the range of the independent variable.
3. Use an appropriate ODE solver function (ode45, etc.).
4. Call the solver and plot the solution.

These steps will be followed when the MATLAB code is deployed to solve odes in the methodology and applications sections.

3 Methodology

3.1 MAHA Integral Transform of Some Functions

Assuming the existence of the fundamental data (2) for any function f(t), the sufficient data for the presence of the MAHA integral transform are that, for t t greater than or equal to 0, f(t) should be piecewise continuous and of exceptional order. Otherwise, the MAHA transform may or may not exist. The MAHA integral transform has basic capacities:

1) If f(t) = k, k is an arbitrary constant function, then via the definition we get:

$$M[k] = (uv)^{\beta} \int_0^{\infty} k e^{-\frac{t}{u}}\, dt = (uv)^{\beta}\, k \frac{e^{-\frac{t}{v}}}{\frac{-1}{v}}]_0^{\infty} = k u^{\beta} v^{\beta+1}$$

2) If f(t) = t so:

$$M[t] = (uv)^{\beta} \int_0^{\infty} \frac{t}{u} e^{-\frac{t}{u}}\, dt$$

So, by integration by parts, $\mathrm{M}[t] = u^{\beta-1} v^{\beta+2}$

Also:
(i) $\mathrm{M}[t^2] = 2u^{\beta-2} v^{\beta+3}$
(ii) $\mathrm{M}[t^3] = 6u^{\beta-3} v^{\beta+4}$
(iii) In general, if n positive integer number, then $\mathrm{M}[t^n] = u^{\beta-n} v^{\beta+n+1}\, n!$, and if n > -1, then $\mathrm{M}[t^n] = u^{\beta-n} v^{\beta+n+1}\, \tau(n+1)$, where $\tau(.)$ is

Gamma function.

3) If f(t) = e^{-at} , where a is an arbitrary constant number, so:

$$M[e^{-at}] = (uv)^{\beta} \int_0^{\infty} e^{-\frac{at}{u}} e^{-\frac{t}{u}}\, dt$$
$$= (uv)^{\beta} \frac{e^{-\left(\frac{a}{u}+\frac{1}{v}\right)t}}{-\left(\frac{a}{u}+\frac{1}{v}\right)}]_0^{\infty} = \frac{(uv)^{\beta+1}}{v+au},$$

also

$$M[e^{at}] = \frac{(uv)^{\beta+1}}{u-av}$$

4) If f(t) = sin(at), where a is an arbitrary constant number, so:

$$M[\sin(at)] = (uv)^{\beta} \int_0^{\infty} \sin\left(\mathrm{a}\frac{t}{u}\right) e^{-\frac{t}{v}}\, dt = \frac{a u^{\beta+1} v^{\beta+2}}{u^2 + a^2 v^2}$$

5) If f(t) = cos(at), a is an arbitrary constant number, so:

$$M[\sin(at)] = (uv)^{\beta} \int_0^{\infty} \cos\left(\mathrm{a}\frac{t}{u}\right) e^{-\frac{t}{v}}\, dt$$

After simple computations, we get:

$$M[\cos(at)] = \frac{u^{\beta+2} v^{\beta+1}}{u^2 + a^2 v^2}$$

6) If f(t) = sinh(at), a is an arbitrary constant number, so:

$$M[\sinh(at)] = (uv)^{\beta} \int_0^{\infty} \sinh\left(\mathrm{a}\frac{t}{u}\right) e^{-\frac{t}{v}}\, dt = (uv)^{\beta} \int_0^{\infty} \frac{\left(e^{\frac{at}{u}} - e^{\frac{-at}{u}}\right)}{2}\, e^{\frac{-t}{v}}\, dt$$

After simple computations, we get:

$$M[\sinh(at)] = (uv)^{\beta+1} \left(\frac{av}{u^2 - a^2 v^2}\right)$$

7) If f(t) = cosh(at), a is an arbitrary constant number, so:

$$M[\cosh(at)] = \frac{u^{\beta+2} v^{\beta+1}}{u^2 - a^2 v^2}$$

8) Shifting property of MAHA integral transform

If MAHA integral transform of f(t) is F(u,v), then MAHA transform of function $e^{at} f(t)$ is given by $\frac{uv^{\beta}}{[u(\frac{uv}{u-av})]^{\beta}}\, F(u, \frac{uv}{u-av})$.

Proof

$$M[f(t)e^{at}] = (uv)^{\beta} \int_0^{\infty} f\left(\frac{t}{u}\right) e^{a\frac{t}{u}}\, e^{-\frac{t}{v}}\, dt$$
$$= (uv)^{\beta} \int_0^{\infty} f\left(\frac{t}{u}\right) e^{-t[\frac{1}{v}-\frac{a}{u}]}\, dt$$
$$= (uv)^{\beta} \int_0^{\infty} f\left(\frac{t}{u}\right) e^{-t[\frac{u-av}{uv}]}\, dt$$
$$= (uv)^{\beta} \frac{u(\frac{uv}{u-av})^{\beta}}{u(\frac{uv}{u-av})^{\beta}} \int_0^{\infty} f\left(\frac{t}{u}\right) e^{-t[\frac{u-av}{uv}]}\mathrm{dt}$$
$$= \frac{(uv)^{\beta}}{u(\frac{uv}{u-av})^{\beta}}\, F(u, \frac{uv}{u-av})$$

Theorem (3.1)

(i) $M[f'(t)] = (uv)^{\beta}\, [-uf(0)] + \frac{u}{v}\, \mathrm{F(u,v)}$

(ii) $M[f''(t)] = (uv)^{\beta}\, [-uf'(0) - \frac{u^2}{v}\, f(0)] + \frac{u^2}{v^2}\, \mathrm{F(u,v)}$

(iii) $M[f'''(t)] = (uv)^{\beta}\, [-uf''(0) - \frac{u^2}{v}\, f'(0) - \frac{u^3}{v^2}\, f(0)] + \frac{u^3}{v^3}\, \mathrm{F(u,v)}$

(iv) $M[f^{(4)}(t)] = (uv)^{\beta}\, [-uf'''(0) - \frac{u^2}{v}\, f''(0) - \frac{u^3}{v^2}\, f'(0) - \frac{u^4}{v^3}\, f(0)] + \frac{u^4}{v^4}\, \mathrm{F(u,v)}$

(iiv) $M\left[f^{(m)}(t)\right] = (uv)^{\beta}\,[-uf^{(m-1)}(0) - \frac{u^2}{v}\,f^{(m-2)}(0) - \frac{u^3}{v^2}\,f^{(m-3)}(0) - \cdots - \frac{u^m}{v^{m-1}}\,f(0)] + \frac{u^m}{v^m}\,\mathrm{F(u,v)}$

Proof

(i) By the definition, we get:

$$M[\mathrm{f}'(\mathrm{t})] = (uv)^{\beta}\int_0^{\infty}\mathrm{f}'\,(\frac{t}{u})e^{-\frac{t}{v}}\,dt$$

Integration by parts method, we have:

$$M[\mathrm{f}'(\mathrm{t})] = (uv)^{\beta}\,[-uf(0)] + \frac{u}{v}F(u,v)$$

(ii) Also, by the definition, we get:

$$M[\mathrm{f}''(\mathrm{t})] = (uv)^{\beta}\int_0^{\infty}\mathrm{f}''\,(\frac{t}{u})e^{-\frac{t}{v}}\,dt$$

Also, Integration by parts method, we obtain:

$$M[\mathrm{f}''(\mathrm{t})] = (uv)^{\beta}\left[-uf'(0) - \frac{u^2}{v}f(0)\right] + \frac{u^2}{v^2}F(u,v)$$

The proof of (iii) and (iv) is similar to (ii).

(iiv) We can confirm by Mathematical Induction.

3.1.1 The Inverse of MAHA Integral Transform

In this part, we present the inverse of MAHA transform technique of basic functions:

(1) $M^{-1}\left[ku^{\beta}v^{\beta+1}\right] = k$

(2) $M^{-1}\left[u^{\beta-n}v^{\beta+n+1}\right] = \frac{t^n}{n!}$, where n > 0 integer number.

(3) $M^{-1}\left[\frac{(uv)^{\beta+1}}{v+au}\right] = e^{-at}$, where a is a constant number.

(4) $M^{-1}\left[\frac{(uv)^{\beta+1}}{u-av}\right] = e^{at}$

(5) $M^{-1}\left[\frac{au^{\beta+1}v^{\beta+2}}{u^2+a^2v^2}\right] = \sin(at)$

(6) $M^{-1}\left[\frac{u^{\beta+2}v^{\beta+1}}{u^2+a^2v^2}\right] = \cos(at)$

(7) $M^{-1}\left[\frac{au^{\beta+1}v^{\beta+2}}{u^2-a^2v^2}\right] = \sinh(at)$

(8) $M^{-1}\left[\frac{u^{\beta+2}v^{\beta+1}}{u^2-a^2v^2}\right] = \cosh(at)$

3.2 Application of MAHA Integral Transform of Ordinary Differential Equations (ODEs)

The MAHA necessary transform can be utilized as a successful device to solve ordinary differential equations. The steps of an exact solution of differential equation using MAHA transform are:

1) Apply MAHA transform on the given equation utilizing the initial conditions.
2) Take inverse transform, you will get the exact solution.
3) Write a programming code to find values of the dependent variables based on the values of independent variable.

The steps of a numerical analysis solution of a differential equation are:

1. Define the differential equation as a system of first-order ODE.
2. Fill in the initial conditions.
3. Set the range values of the independent variable as needed.
4. Solve the differential equation (you can use a built in function of a programming language).
5. Extract and plot the solution.

Example 1: 1st order linear ODE

Consider the differential equation:

$$\frac{\mathrm{dy}}{\mathrm{dx}} + \mathrm{y} = 0, \mathrm{y}(0) = 1 \qquad (3)$$

Exact Solution:

1) Take MAHA integral transform to this condition, you get:

$$\mathrm{M}[\frac{\mathrm{dy}}{\mathrm{dx}}] + \mathrm{M}[\mathrm{y}] = 0$$

$$(uv)^{\beta}\left(-uy(0)\right) + \frac{\mathrm{u}}{\mathrm{v}}F(u,v) + F(u,v) = 0$$

$$F(u,v)\left[\frac{\mathrm{u}}{\mathrm{v}} + 1\right] = u(uv)^{\beta}$$

So

$$F(u,v) = \frac{(uv)^{\beta+1}}{\mathrm{u}+\mathrm{v}}$$

2) Take inverse to both sides, you get:

The exact solution:

$$y(x) = e^{-x}$$

The MATLAB code Solution:

3)

```
% Define the range of x
x = linspace(0, 5);% as needed
% Compute y(x)
y = exp(-x);
% Plot the function
figure; plot(x, y, 'LineWidth', 2); xlabel('x');ylabel('y'); title('y(x) = e^{-x}');grid on;
```

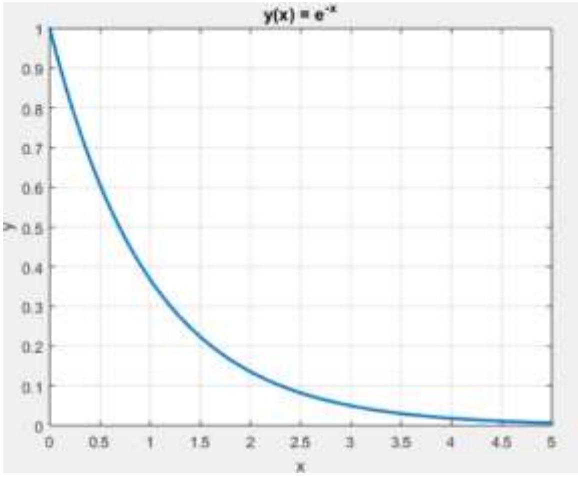

Fig. 1: On the interval [0, 5], plot exact solution of $\frac{\mathrm{dy}}{\mathrm{dx}} + \mathrm{y} = 0, \quad \mathrm{y}(0) = 1$

Numerical Analysis Solution using the built-in function ode45:

```
% Define the differential equation function: y' = -y
ode = @(x, y) -y;
% Define the initial condition
y0 = 1;
% Define the range of x values over which to solve the equation
xspan = [0 5]; %adjust as needed
% Solve the differential equation
[x, y] = ode45(ode, xspan, y0);
% Plot the solution
plot(x, y, '-o'); xlabel('x');
ylabel('y(x)');
title('Solution of dy/dx + y = 0');grid on;
```

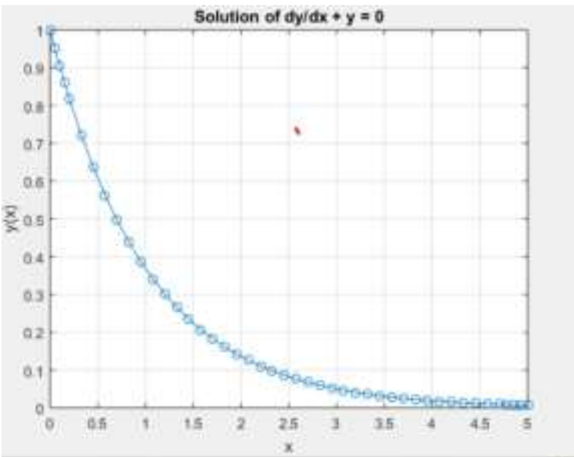

Fig. 2: Numerical solution of dy/dx +y =0, y(0) =1 on the interval [0, 5]

Figure 1 and Figure 2 show the plot of the exact and numerical solutions, respectively. These visually validate the solutions by the two methods.

Example 2: 2nd DE

Find the solution of the second-order differential equation y″+y=0, with initial conditions y(0)=1 and y′(0)=1:

$$y'' + y = 0,\ y(0) = 1,\ y'(0) = 1 \qquad (4)$$

Exact Solution:

1) Take MAHA transform to this differential condition, you will get:

$$(uv)^{\beta}(-u) - \frac{u^2}{v}(uv)^{\beta} + \frac{u^2}{v^2}F(u,v) + F(u,v) = 0$$

$$F(u,v)\left[1 + \frac{u^2}{v^2}\right] = u(uv)^{\beta} + \frac{u^2}{v}(uv)^{\beta}$$

So

$$F(u,v) = \frac{u^{\beta+1}v^{\beta+2}}{u^2+v^2} + \frac{u^{\beta+2}v^{\beta+1}}{u^2+v^2}$$

2) Take the inverse MAHA transform of this equation to get the exact solution:

$$y(x) = \sin x + \cos x$$

The MATLAB code Solution:

3) Use MATLAB code, and compute:

y = sin(x)+ cos(x);

This is validated by the following analytical algorithm (Figure 3).

Numerical Analysis Solution Using the built-in function ode45:

In this case, the equation y″+y=0 is rewritten as a system of first-order ODEs: y1′=y2, y2′=−y1.Here, y1=y, and y2=y′. Therefore, ode is defined as a function that takes x (the independent variable) and y (a dependent variable as a vector containing y and y′) and returns the derivatives [y'(x); y"(x)]. The built-in function ode45 is used.

```
% Rewrite the differential equation y" + y = 0 as a system of first-order ODEs:
% y1 =y and y2 =y' % y1' =y2 and y2' = -y1
% Define the system of ODEs
dydt = @(t, y) [y(2); -y(1)];
y0 = [1; 1];  % y(0) = 1, y'(0) = 1
% Define the time span for the solution
tspan = [0 10]; % adjust as needed
% Solve the system of ODEs using ode45
[t, y] = ode45(dydt, tspan, y0);
% Plot the solution
figure;plot(t, y(:,1), '-o');
xlabel('Time t');ylabel('Solution y');
title('Solution of the differential equation y'''' + y = 0');
grid on;
```

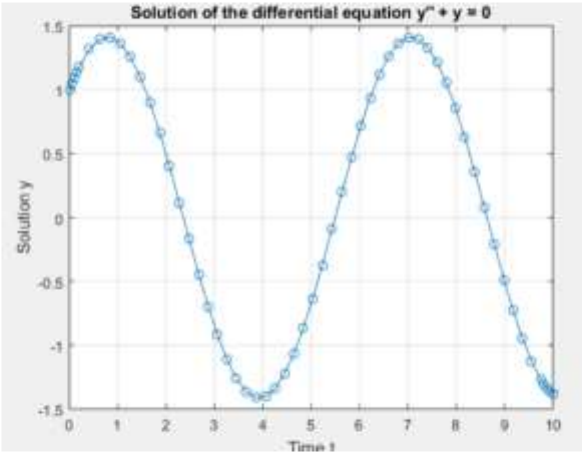

Fig. 3: Virtual Validation of exact and numerical solution of
$y'' + y = 0, y(0) = y'(0) = 1$

Example 3: 2^{nd} *request* ODE

Consider the 2^{nd} *request* differential equation:

$$y'' - 3y' + 2y = 0, y(0) = 1, y'(0) = 4 \quad (5)$$

Exact Solution:

1) Take MAHA transform to above differential equation, we get:

$$(uv)^{\beta}(-u) - \frac{u^2}{v}(uv)^{\beta} + \frac{u^2}{v^2}F(u,v) - 3[(uv)^{\beta}(-u) + \frac{u}{v}F(u,v)] + 2F(u,v) = 0$$

So

$$F(u,v) = \frac{(uv)^{\beta}(u + \frac{u^2}{v})}{(\frac{u}{v} - 2)(\frac{u}{v} - 1)}$$

Now

$$\frac{u + \frac{u^2}{v}}{\left(\frac{u}{v} - 2\right)\left(\frac{u}{v} - 1\right)} = \frac{A}{(\frac{u}{v} - 2)} = \frac{B}{(\frac{u}{v} - 1)}$$

After simple computations, we get: A=3, B=-2.

2)Apply inverse

$$F(u,v) = \frac{3(uv)^{\beta+1}}{u - 2v} - \frac{2(uv)^{\beta+1}}{u - v}$$

to get general exact solution as:

$$y(x) = 3e^{2x} - 2e^{x}$$

3) Use MATLAB and compute:

y = 3*exp(2*x)- 2*exp(x);

This is validated by the following analytical algorithm (Figure 4).

Numerical Analysis Solution Using the built-in function ode45:

To solve y′′−3y′+2y=0, with initial conditions y(0)=1 and y′(0)=4, for example, in MATLAB, you can use the ode45 function as follows. Let y1′=y2, y2′=3y2−2y1; where y1′ =y and y2=y′. Therefore, ode is defined as a function that takes x and y (a vector containing y and y′) and returns the derivatives [y'(x); y"(x)].

```
% Define the differential equation as a system of first-order ODEs.
  ode = @(x, y)[y(2);3*y(2)- 2*y(1)];
% Initial conditions
   y0 = [1; 4];  % [y(0); y'(0)]
   xspan = [0 10];  % Adjust as needed
 % Solve the differential equation
   [x, sol] = ode45(ode, xspan, y0);
   y = sol(:, 1);  % y(x)
  yp = sol(:, 2); % y'(x)
%Plot the solutions
figure; subplot(2, 1, 1);
plot(x, y, '-o'); xlabel('x');
ylabel('y(x)');
title('Solution of y'''' - 3y'' + 2y = 0'); grid on;
subplot(2, 1, 2);plot(x, yp, '-o');
xlabel('x');ylabel('y''(x)');
title('Derivative of y(x)');grid on;
```

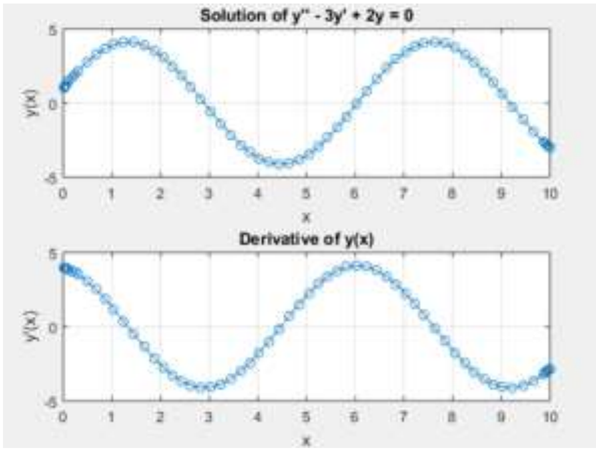

Fig. 4: Virtual validation of exact $solution\ y(x) = 3e^{2x} - 2e^{x}$ and analytical solution of $y'' - 3y' + 2y = 0, y(0) = 1, y'(0) = 4$

Example 4: 2^{nd} order linear nonhomogeneous

Consider 2^{nd} order linear nonhomogeneous request differential equation:

$$y'' + 9y = \cos 2x, y(0) = 1, y\left(\frac{\pi}{5}\right) = -1 \quad (6)$$

Exact Solution:

1) Since *y'* (0) is unknown, let *y'*(0)=*a*. Take MAHA transform of this condition and utilizing beginning conditions, you will have:

$$(uv)^{\beta}\left[-ua - \frac{u^2}{v}\right] + \frac{u^2}{v^2}F(u,v) + 9F(u,v) = \frac{u^{\beta+2}v^{\beta+1}}{u^2 + 9v^2}$$

$$F(u,v)\left[9+\frac{u^2}{v^2}\right]=\frac{u^{\beta+2}v^{\beta+1}}{u^2+9v^2}+(au+\frac{u^2}{v})(uv)^{\beta}$$

So

$$F(u,v)=\frac{(uv)^{\beta}u^2v^{-1}+(uv)^{\beta}u^2v^{-1}(4+\frac{u^2}{v^2})}{(4+\frac{u^2}{v^2})(9+\frac{u^2}{v^2})}+\frac{3(uv)^{\beta}au}{3(9+\frac{u^2}{v^2})}$$

After simple computations, you will get:

$$A=0, B=\frac{4}{5}, C=0, D=\frac{1}{5}$$

2)Take inverse of MAHA transform, then the exact solution is:

$$y(x)=\frac{a}{3}\sin(3x)+\frac{1}{5}\cos(2x)+\frac{4}{5}\cos(3x)$$

To find a, note that $y\left(\frac{\pi}{2}\right)=-1$

Then, we find a =12/5.

Then, the exact solution is
$y(x)=\frac{4}{5}\sin(3x)+\frac{1}{5}\cos(2x)+\frac{4}{5}\cos(3x)$
3) Use MATLAB and compute:
y= 0.8*sin(3*x)+0.2*cos(2*x)+0.8*cos(3*x)

This is validated by the following analytical algorithm (Figure 5).

Numerical Analysis Solution using the built-in function ode45:
To solve the differential equation y′′+9y=cos(2x) with the boundary conditions y(0)=1 and y(π/5)=−1 in MATLAB, you can use the bvp4c function which is designed for boundary value problems.

```
syms y(x)
Dy = diff(y);
D2y = diff(y, 2);
% Define the differential equation
ode = D2y + 9*y == cos(2*x);
cond1 = y(0) == 1; cond2 = y(pi/5) == -1;
sol = dsolve(ode, [cond1, cond2]);
x_vals = linspace(0, 2*pi, 1000);
y_vals = double(subs(sol, x, x_vals));
figure; plot(x_vals, y_vals, '-o');
xlabel('x');
ylabel('y');
title('Solution of the differential equation y'''' + 9y = cos(2x)');
grid on;
```

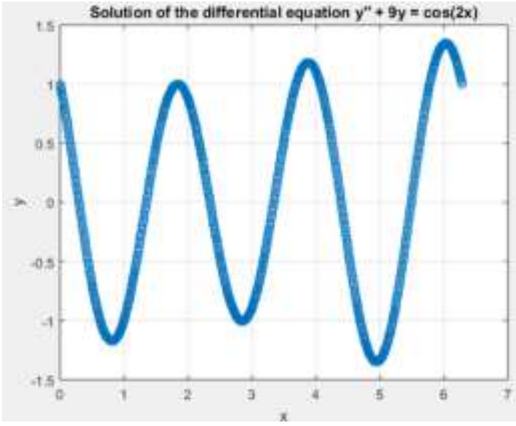

Fig. 5: A solution of
$y''+9y=\cos 2x, y(0)=1, y\left(\frac{\pi}{5}\right)=-1$

Example 5: 2nd ode
Consider the differential equation

$$y''-3y'+2y=4e^{3x}, y(0)=-3, y'^{(0)}=5 \quad (7)$$

Exact Solution:
1) Take MAHA technique of this differential problem and appling the initialdata:

$$(uv)^{\beta}\left[-5u+\frac{3u^2}{v^2}\right]+\frac{u^2}{v^2}F(u,v)-3\left[(uv)^{\beta}+3u\right]-3\frac{u}{v}F(u,v)+2F(u,v)=\frac{4(4v)^{\beta+1}}{4-3v}$$

$$F(u,v)\left[\frac{u^2}{v^2}-\frac{3u}{v}+2\right]=(uv)^{\beta}\left(-5u+\frac{3u^2}{v^2}\right)+9(uv)^{\beta}u+\frac{4(4v)^{\beta+1}}{4-3v}$$

$$F(u,v)=\frac{(uv)^{\beta}[5u^2v^{-1}-38u]}{(\frac{u}{v}-3)(\frac{u}{v}-2)(\frac{u}{v}-1)}$$

So
$F(u,v)=(uv)^{\beta}u[\frac{A}{(\frac{u}{v}-3)}+\frac{B}{(\frac{u}{v}-2)}+\frac{C}{(\frac{u}{v}-1)}]$
We get: $A=2, B=4, C=-9$

$$F(u,v)=\frac{2(uv)^{\beta+1}}{\mathrm{u}-3\mathrm{v}}+\frac{4(uv)^{\beta+1}}{\mathrm{u}-2\mathrm{v}}-\frac{9(uv)^{\beta+1}}{\mathrm{u}-\mathrm{v}}$$

2)Take inverse transform, we obtain:

$$y(x)=2e^{3x}+4e^{2x}-9e^{x}$$

3) Use MATLAB and compute:
y= 2*exp(3*x)+4*exp(2*x)-9*exp(x)

Plot the function on the interval [0, 1], for example (Figure 6).

Numerical Analysis Solution using the built-in function ode45:
To solve the 2nd ode $y''-3y'+2y=4e^{3x}$, with initial conditions $y(0)=-3$ and $y'(0)=5$, the non-homogeneous term $4e^{3x}$ needs to be handled separately from the homogeneous equation. In MATLAB, the analytical solution can be implemented using ode45 as follows.

```
% Define the differential equation as a system of 1st ODEs
 ode = @(x, y) [y(2); 3*y(2) - 2*y(1) +
4*exp(3*x)];
% Initial conditions
 y0 = [-3; 5];  % [y(0); y'(0)]
% Define the range of x values
 xspan = [0 1];
% Solve the differential equation
 [x, sol] = ode45(ode, xspan, y0);
% Extract solutions
 y = sol(:, 1);  % y(x)
 yp = sol(:, 2); % y'(x)
% Plot the solutions
 figure;
 subplot(2, 1, 1);plot(x, y, '-o');
 xlabel('x');
 ylabel('y(x)');
 title(' solution of y′′−3y′+2y=4e3x ');
grid on;
subplot(2, 1, 2);
plot(x, yp, '-o');xlabel('x');
ylabel('y''(x)');
title(' y=2e^(3x)+4e^(2x) − 9e^x');
grid on;
```

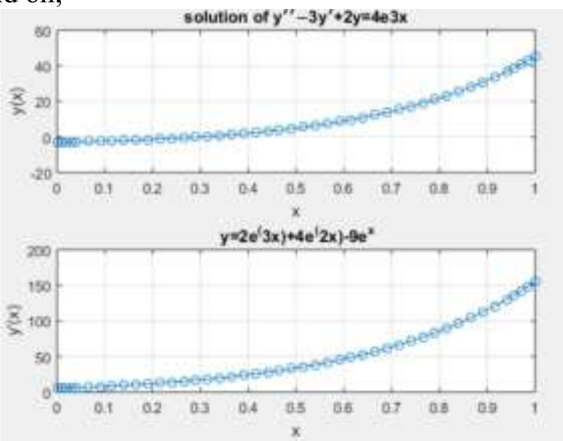

Fig. 6: Visualize validation of exact and numerical solutions, on the interval [0, 1], of $y''-3y'+2y=4e^{3x}, y(0)=-3, y'(0)=5$

3.3 Applications

3.3.1 Maha Integral Transform on "Nuclear Physics"

The following problem is based on nuclear physics fundamentals. Consider the 1st order linear ordinary differential equation:

$$\frac{dN(t)}{dt}=-\alpha N(t) \qquad (8)$$

The essential relationship describing radioactive decay is given in this equation, where *N(t)* during time *t* denotes the number of not decayed atoms left in a sample of radioactive isotope, and α is the decay constant. Apply the Maha integral transform M, to set:

$$M\{N'(t)\}+\alpha M\{N(t)\}=0 \qquad (9)$$

Therefore

$$(uv)^{\beta}[-uN(0)]+\frac{u}{v}F(u,v)+\alpha F(u,v)=0$$

$$(\frac{u}{v}+\alpha)F(u,v)=(uv)^{\beta}uN(0)$$

$$F(u,v)\left(\frac{u}{v}+1\right)= u^{\beta+1}v^{\beta}N_0$$

$$F(u,v)=\frac{u^{\beta+1}v^{\beta}N_0}{\frac{u}{v}+\alpha}$$

So

$$F(u,v)=N_0\frac{(uv)^{\beta+1}}{u+\alpha v} \qquad (10)$$

Take the inverse to both sides, we get the exact solution:

$$N(t)=N_0e^{-\alpha t} \qquad (11)$$

This is the proper type of radioactive decay.
You can solve dN(t)/dt =−αN(t) analytically in MATLAB, where α (alpha) is a constant, using ode45 function as follows. The plot in Figure 7 visualizes how the population N(t) evolves over time t.

```
% Define the parameters
alpha = 0.1;  % for example
% Define the differential equation function
ode = @(t, N) -alpha * N;
N0 = 100;% Initial population size
tspan = [0 10];% Adjust as needed
% Solve the differential equation
[t, N] = ode45(ode, tspan, N0);
% Plot the solution
plot(t, N, '-o');
xlabel('Time (t)');
ylabel('N(t)');
title(['Solution of dN(t)/dt = -\alpha N(t), \alpha = ',
num2str(alpha)]);
grid on;
```

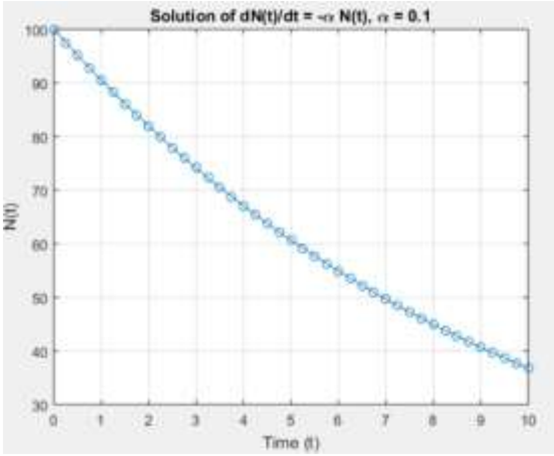

Fig. 7: Plot of the numerical solution of $\frac{dN(t)}{dt} = -\alpha N(t)$, with initial value N(0) =100

3.3.2 Blood Glucose Concentration

During continuous intravenous glucose injection, the concentration of glucose in the blood is G(t) exceeding the baseline value at the start of the infusion. The function G(t) satisfies the initial value problem (I, V, P).

$$G'(t) + kG(t) = \frac{\alpha}{\gamma} \tag{12}$$

Where $t \in (0, \infty)$ and $G(0)=0$.

The variables k, α and γ in equation (12) represent the constant velocity of elimination, the rate of infusion, and the volume, respectively, in which glucose is distributed. The Maha integral transform technique can be utilized to solve the equation in (12). Then, the concentration of glucose presents in the blood stream at time t is $G(t)=\alpha/(\gamma k)(1-e^{-kt})$.

Upon bilateral application of the Maha integral transform on (12), the resulting expression is obtained as in equation (13).

$$M\{G'(t)\} + kM\{G(t)\} = \frac{\alpha}{\gamma} M\{1\} \tag{13}$$

Let $M\{G(t)\} = F(u, v)$. By utilizing the initial value problem (I, V, P) and the integral transform outlined in section 3.1, the equation (13) can be rearranged with the aid of equation (12) as:

$$(uv)^{\beta}[-uG(0)] + \frac{u}{v}F(u,v) + kF(u,v) - \frac{\alpha}{\gamma}u^{\beta}v^{\beta+1} = 0$$

$$\left(\frac{u}{v} + k\right)F(u,v) = \frac{\alpha}{\gamma}u^{\beta}v^{\beta+1}$$

So

$$F(u,v) = \frac{\alpha}{\gamma}\frac{u^{\beta}v^{\beta+2}}{u+kv} \tag{14}$$

After simple computation and using inverse of Maha transform to this expression, we get the concentration of glucose in the blood as in (15).

$$G(t) = \frac{\alpha}{\gamma k}(1 - e^{-kt}) \tag{15}$$

A MATLAB code can be utilized to produce a plot showing the exact solution of G(t) as a function of time t (Figure 8). The exact formula $G(t)=\alpha/(\gamma k)(1-e^{-kt})$ is directly used to compute G(t), ensuring accuracy and validation of the solution. The parameters `alpha`, `gamma`, `k`, and the range of `t` values (`tspan`) can be adjusted as needed for a specific scenario. Note that the "%" in the programming code is for comments and not for execution statements.

```
% Define the parameters
 alpha = 1.0; gamma = 1.0;k = 0.1;
% Define the function for G(t)
 G_exact = @(t)(alpha/(gamma*k)) *(1-exp(-k*t));
 tspan = [0 50]; % example time span
 G0 = 0; % initial value of G
 G = G_exact(t);
% Plot the exact solution
 plot(t,G,'-o'); xlabel('Time (t)');
 ylabel('G(t)');
title('G(t)=\alpha/(\gamma k)(1-e^{-kt})');
 grid on;
```

This code is self-contained and can be executed directly in MATLAB. Adjust the parameter values as needed to fit your specific problem. The following code solves the given differential equation analytically and plots the solution (Figure 8).

```
% Define parameters
alpha = 1;
gamma = 1;
k = 0.1;
tspan = [0 50]; % example time span
G0 = 0; % initial value of G
% Define the differential equation as a function
dGdt = @(t, G) (alpha / gamma) - k * G;
% Solve the differential equation using ode45
[t, G] = ode45(dGdt, tspan, G0);
% Plot the solution
figure;
plot(t, G, 'LineWidth', 2);
title('Solution of dG/dt = (\alpha / \gamma) - k \cdot
G');
xlabel('Time (t)');
ylabel('G(t)');
grid on;
```

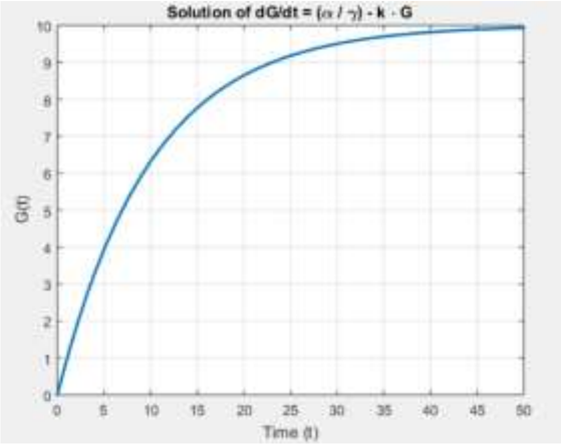

Fig. 8: Plotted solution of G(t) with alpha = 1.0; gamma = 1.0;k = 0.1

3.3.3 Aorta Pressure

The heart's contraction facilitates the transportation of blood into the aorta. The initial value problem is concerned with the aortic pressure function P(t) as in equation (16).

$$P'(t) + \frac{c}{k}P(t) = cAsint(wt) \qquad (16)$$

where $P(0) = p_0$, and c, k, A, w are constants. The Maha integral transform technique is utilized to derive the pressure in the aorta. With the bilateral application of the Maha integral transform to equation (16). The resulting expression is as in (17).

$$M\{P'(t)\} + \frac{c}{k}M\{P(t)\} = cAM\{sint(wt)\} \quad (17)$$

By applying the initial value problem and utilizing the transform outlined in section 3.1, the rearrangement of equation (17) can be expressed as:

$$(uv)^{\beta}[-uP(0)] + \frac{u}{v}F(u,v) + \frac{c}{k}F(u,v) = cA[\frac{wu^{\beta+1}v^{\beta+2}}{u^2 + w^2v^2}]$$

$$\left(\frac{u}{v} + \frac{c}{k}\right)F(u,v) = cAw\left[\frac{u^{\beta+1}v^{\beta+2}}{u^2 + w^2v^2}\right] + P_0\frac{u^{\beta+1}v^{\beta}}{\frac{u}{v} + \frac{c}{k}}$$

$$F(u,v) = \frac{cAwu^{\beta+1}v^{\beta+2}}{(\frac{u}{v}+\frac{c}{k})(u^2+w^2v^2)} + P_0\frac{u^{\beta+1}v^{\beta}}{\frac{u}{v}+\frac{c}{k}} \qquad (18)$$

After preforming basic calculations and applying partition fractions along with the inverse of the Maha integral transform to the given expression, the resultant value obtained represents the amount of pressure in the aorta as in function (19).

$$P(t) = P_0e^{-\frac{c}{k}t} + \frac{cAwk^2}{w^2k^2+c^2}(\frac{c}{wk}\sin(wt) - \cos(wt) + e^{-\frac{c}{k}t}) \qquad (19)$$

Figure 9 shows visualization solution, by the following MATLAB code to analytically solve and plot the given function P(t). Figure 10 shows a numerical solution for the same ODE.

```
% Define the parameters
P0 = 0; % Initial value
c = 1; k = 1; A = 1; w = 1;
t = linspace(0, 100, 1000);
% Define the function P(t)
P = @(t) P0*exp(-c/k * t)+(c*A*w*k^2)/
(w^2*k^2+c^2)*((c/(w*k))*sin(w*t)-
cos(w*t)+exp(-c/k*t));
% Evaluate the function
P_vals =P(t);
plot(t, P_vals, 'LineWidth', 2);
title('Solution to P(t)');
xlabel('Time t');ylabel('P(t)');
grid on;
```

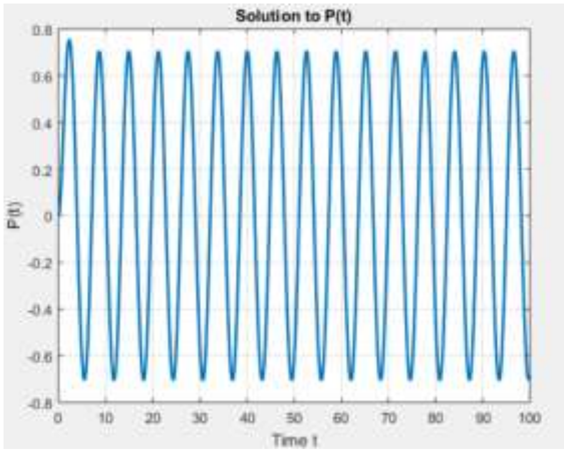

Fig. 9: Plot of exact solution of P(t)'

```
% Define the parameters
c = 1;k = 1; A = 1;w = 1;
% Define the differential equation
ode = diff(P, t) + (c/k) * P == c * A * sin(w * t);
% Define the initial condition
cond = P(0) == 0;
% Solve the differential equation
sol = dsolve(ode, cond);
% Convert the symbolic solution to a MATLAB
function
P_sol = matlabFunction(sol);
t_vals = linspace(0, 100, 1000);
P_vals = P_sol(t_vals);
% Plot the solution
figure; plot(t_vals, P_vals, '-o');
xlabel('Time t'); ylabel('P(t)');
title('Solution of P"(t) + c/k P(t) = cA sin(wt)');grid
on;
```

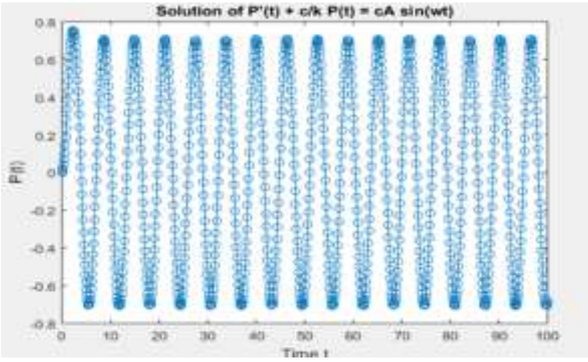

Fig. 10: Plot of numerical solution of P(t) with P(0) =0, and c=k =A=w =1
t = linspace(0, 100, 1000);

The results show that the MATLAB codes for the solutions are matched and agreed with the theoretical approach. The programming codes present examples and will help users in writing computing solutions to problems similar to the presented ones.

4 Conclusions

MAHA integral transform with two parameters conversion of linear ordinary differential equations with constant coefficients and higher orders are extended. The correctness of the transform is proved in the methodology section. Steps to solve ODEs using MAHA integral transform are presented. The steps are applied to find the exact solutions of five different examples of ODEs and three different examples of applications. The steps to solve DOEs numerically and validate the exact solution are presented. MATLAB codes are deployed to show exact (direct) solutions and analytical solutions for the selected eight ODEs. The exact solutions and the numerical ones for given functions are validated and plotted. The two methods are applied to find exact solutions and numerical ones of nuclear physics and two medical applications. It is found that the exact solution is simpler and easier than the previous two parameters, and it can be numerically validated. The presented programming code will be helpful for users interested in computing scientific numerical applications.

As a future work, we intend to investigate the performance of Maha transform in enhancing the security of image encryption/ decryption. The image encryption process starts with representing the image as numerical data. Then, applying Maha transformation on these values through ODE results in an encrypted version of the image. For decryption, if the correct initial conditions and ODE system are given, then the original image can be retrieved by reversing the transformation.

References:

[1] A. Belafhal, R. EL Aitouni, T. Usman, Unification of Integral Transform and Their Applications, *Partial Differential Equations in Applied Mathematics*, vol. 10, 100695, 2024. http://dx.doi.org/10.1016/j.padiff.2024.100695

[2] C. Constanda, Solution techniques for elementary partial differential equations. *Chap- man and Hall/CRC*, 1st Edition, New York, pp, 272, 2002. https://doi.org/10.1201/9781420057515.

[3] D. G. Duffy, *Transform methods for solving partial differential equations*. CRC press, 2nd Edition, New York, pp 728, 2004. https://doi.org/10.1201/9781420035148.

[4] T. M. Elzaki, The new integral transform Elzaki transform', Glob. *J. Pure Appl. Math*., vol. 7, no. 1, pp. 57-64, 2011.

[5] M. Alsaoudi, G. Gharib, E. Kuffi, A. Guiatni. New Two Parameter Integral Transform “MAHA Transform” and its Applications in Real Life, *WSEAS Transactions on Mathematics*, vol. 23, pp. 536-543. 2024. https://doi.org/10.37394/23206.2024.23.56.

[6] J. A. Jasim, E.A. Kuffi and S. A. Mehdi, A Review on the Integral Transforms, *Journal of University of Anbar for Pure Science*, vol. 17, no. 2, pp. 273-310, 2023. DOI: 10.37652/juaps.2023.141302.1090.

[7] A. Kamal and H. Sedeeg, The New Integral Transform Kamal Transform, *Adv. Theor. Appl. Math*., vol. 11, no. 4, pp. 451-458, 2016.

[8] M. A. M. Mahgoub, The new integral transform 'Mahgoub Transform', *Adv. Theor. Appl. Math*., vol. 11, no. 4, pp. 391-398, 2016.

[9] I. A. Mansour, E. Kuffi and S. A. Mehdi, Solving Partial Differential Equations Using Double Complex SEE Integral Transform, *AIP Conference Proceedings*, 2591, 050007, 2023. https://doi.org/10.1063/5.0119609.

[10] S. A. Mehdi, E. Kuffi, J. Adel and J. A. Jasim, Solving the partial differential equation by using SEJI integral transform”, *Journal of Interdisciplinary Mathematics*, vol. 26, no. 6, pp. 1123-1132, 2023. https://doi.org/10.47974/JIM-1611.

[11] E. Kuffi and A. Moazzam, Solution of population growth rate linear differential

models via two parametric SEE transformation, *IHJPAS*. vol. 36, no. 2, pp. 430-435. 2023. https://doi.org/10.30526/36.2.3251.

[12] R. U. Rao, ZZ Transform Method for Natural Growth and Decay Problems, *International Journal of Progressive Science and Technologies*, vol. 5, no. 2, 2017. http://dx.doi.org/10.52155/ijpsat.v5.2.159.

[13] F. Mubarak, MZ. Iqbal, A. Moazzam, U. Amjed, MU. Naeem, Substitution Method Using The Laplace Transformation for Solvin Partial Differential Equation Involving More Than Two Independent Variables, *Bulletin of Mathematics and Statistics Research,* vol. 9, no. 3, pp. 104-116, 2021. https://doi.org/10.33329/bomsr.9.3.104.

[14] R. Saadeh, A. Qazza, K. Amawi, A New Approach Using Integral Transform to Solve Cancer Models, *Fractal and Fractional,* vol. 6, no. 9, pp. 490, 2022. http://dx.doi.org/10.3390/fractalfract6090490.

[15] R.Saadeh, A. Qazza, A. Burqan, A New Integral Transform: ARA Transform and its Properties and Applications, *Symmetry*, vol. 10, no. 6, pp. 925, 2020. http://dx.doi.org/10.3390/sym12060925.

[16] D. Thakur, Utilizing the Upadhyaya Transform to Solve the Linear Second Kind Volterra Intergral Equation, *The Review of Contemporay Scientific and Academic Studies*, vol. 3, no. 4, pp. 6, 2023. http://dx.doi.org/10.55454/rcsas.3.04.2023.007

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Conceptualization, Alsaoudi and Sharieh; methodology, Sharieh; validation Alsaoudi and Guiatni; formal analysis, Alsaoudi; investigation, Alsaoudi; resources, Gharib; data curation, Sharieh; writing/original draft preparation, Guiatni; writing/review and editing, Guiatni; supervision, Gharib; project administration, Gharib; steps of the solutions and programming language code, Sharieh.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare.