Use of Haar Wavelets in solving Differential equations

PROJECT REPORT

Submitted in partial fulfillment of the requirements
of MATH F476

by

Kaushik Iyer - 2017A3PS0162G

Under the supervision of

Dr. Amit Setia

Acknowledgements

I would like to express my sincerest gratitude to my project mentor/supervisor Dr. Amit Setia for his constant guidance in helping me get through the theoretical and practical aspects of this project.

List of Figures

List of Tables

Contents

Abstract

This report presents the usage of Haar wavelets in solving Ordinary Differential Equations(ODE), Partial Differential Equations and a system of ODEs. an alternative methodology for the same has also been presented. This is useful in finding numerical solutions to several mathematical models and provides solutions of accuracy better or comparable to existing numerical methods like the Runge-Kutta method, Finite difference method and others.

1 Literature Review

1.1 Introduction to Haar Wavelets

The Haar wavelets, introduced by Alfred Haar in [1] are defined as:

$$H(t) = \begin{cases} 1 & \text{if } 0 \leq t < 1/2 \\ -1 & \text{if } 1/2 \leq t < 1 \\ 0 & \text{elsewhere} \end{cases} \tag{1}$$

For i = 1,2,..., let $i = 2^j + k$ where $j = 0, 1, ..., J$ and $k = 0, 1, ..., 2^j - 1$. For $t \in [0, 1]$, define $h_i(t) = 2^{\frac{j}{2}} H(2^j t - k)$. $h_0(t) = 1$ for $t \in [0, 1]$. It can be shown that $h_i(t)_{i=0}^{\infty}$ forms a set of orthornormal basis in $L^2[0, 1]$. The Haar basis is a set of translated and scaled version of the block function defined above. Here, j denotes the level of resolution and k is the translation parameter. The normalized Haar functions can be obtained by dividing the entire function by the square root of the maximum resolution.

The Haar functions can be discretized by choosing discrete values of t in [0,1]. These points are called collocation points and are defined as $t_j = \frac{j-0.5}{N}$ $j = 1, 2, ..., N$.

The Haar vector and the Haar matrix is therefore formed as shown in [2]

$H_N(t) = [h_0(t), h_1(t), ..., h_N(t)]^T$

$H_{N\times N}(t) = [H_N(\frac{1}{2N}), H_N(\frac{3}{2N}) ...., H_N(\frac{N-1}{2N})]$ at the above mentioned collocation points.

For example, $H_{4x4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & 0 & 0 \\ 0 & 0 & -\sqrt{2} & -\sqrt{2} \end{bmatrix}$

1.2 Function Approximation

Like the Fourier series, the Haar basis can be used to approximate any function defined on $L^2[0, 1]$. Any function f(t) can be expanded as an infinite summation of the Haar basis functions:

$$f(t) = \sum_{i=0}^{\infty} c_i h_i(t) \tag{2}$$

where c_i are called the Haar coefficients and are calculated as follows

$$c_i = \langle f(t) i h_i(t) \rangle = \int_0^1 f(t) h_i(t) dt \tag{3}$$

For computation purposes, the infinite summation is often reduced to N terms, where $N = 2^J$, J being the maximum resolution desired. Let $f(t) = f(t_j)$ be the vector of the function value at the N collocation points
In the matrix form, eqs. (2) and (3) can be represented as

$$f(t) = c^T H_{N \times N} \Rightarrow c^T = f(t) H_{N \times N}^{-1} \tag{4}$$

For a two-dimensional function, f(t,x), the function approximation is given as:

$$f(t,x) = H_t^T C H_x \Rightarrow C = H_t f(t,x) H_x^T \tag{5}$$

using the orthogonality of the Haar matrix ($HH^T = I$). Here, H_t and H_x are $N \times N$ matrices and f(t,x) is discretized to $f(t_j, x_i)$ forming an $N \times N$ matrix.

1.3 Application to Numerical Solution of Differential Equations

The Haar wavelets have been used extensively in solving several types of differential equations. They have found applications in solving linear ODEs (Ordinary Differential Equations) with linear and nonlinear coefficients , PDEs (Partial Differential Equations), System of ODEs and even in Fractional Differential and Partial Differential Equations (FDEs and FPDEs).
It is necessary to introduce the Haar Operational matrix of Integration here which plays a central role in approximation the solution to the differential equation.

1.3.1 Haar Operational Matrix of Integration

The Haar operational matrix of Integration is defined by [3]:

$$P_{i,1}(t) = \int_0^{t_j} h_i(t)dt \tag{6}$$

$$P_{i,n+1}(t) = \int_0^{t_j} P_{i,n} h_i(t)dt \tag{7}$$

n = 1,2,3,..... represents the order of the integral
for example

$$P_{i,1}(t) = \begin{cases} t - \zeta 1_1 & \text{if } \zeta_1 \leq t < \zeta_2 \\ \zeta_3 - t & \text{if } \zeta_2 \leq t < \zeta_3 \\ 0 & \text{elsewhere} \end{cases} \tag{8}$$

$$P_{i,2}(t) = \begin{cases} \frac{1}{2}(t - \zeta_1)^2 & \text{if } \zeta_1 \leq t < \zeta_2 \\ \frac{1}{4m^2} - \frac{1}{2}(\zeta_3 - t)^2 & \text{if } \zeta_2 \leq t < \zeta_3 \\ \frac{1}{4m^2} & \text{if } \zeta_3 \leq t < 1 \\ 0 & \text{elsewhere} \end{cases} \tag{9}$$

where $\zeta_1 = \frac{k}{m}$, $\zeta_2 = \frac{k+\frac{1}{2}}{m}$, $\zeta_3 = \frac{k+1}{m}$, $m = 2^j$, $j = 0, 1, 2, ..., J$

2 Ordinary Differential Equations

2.1 Initial Value Problems

Consider the Simple Harmonic Motion equation

$$y''(t) + y(t) = 0 \tag{2.1}$$

with initial conditions $y(0) = 1, y^{'}(0) = 0$. The exact solution is $y(t) = cos(t)$. The solution to this problem is started by approximating the highest order derivative by the Haar series and using the Haar operation matrix of Integration for the lower order derivatives.

$$y''(t) = \sum_{i=0}^{2M} c_i h_i(t) = c^T H \tag{2.1.1}$$

Integrating the above equation on both sides once and twice, we get
$y'(t) - y'(0) = \sum_{i=0}^{2M} c_i P_{i,1}(t) = c^T P_{i,1}$

$$y'(t) = y'(0) + c^T P_{i,1} \tag{2.1.2}$$

$y(t) - y(0) = \sum_{i=0}^{2M} c_i P_{i,2}(t) + ty'(0) = c^T P_{i,1} + t_j y'(0)$

$$y(t) = c^T P_{i,1} + t_j y'(0) + y(0) \tag{2.1.3}$$

substituting (2.1.1) and (2.1.3) in (2.1), we get
$c^T H + c^T P_{i,1} + t_j y'(0) + y(0) = 0$
which is a linear matrix equation and can be solved for c^T. After finding the coefficients, they can be substituted in (**??**) to get the approximated solution. The following figure shows the results of the simulation.

Consider an ODE with nonlinear coefficients

$$y''(t) + \frac{2}{t} y'(t) - 4(t^2 + 6) y(t) = 0 \tag{2.2}$$

with the initial conditions $y(0) = 1$ and $y'(0) = 0$. The exact solution is $y(t) = e^{-t^2}$. Proceeding in a similar manner, we get

$$y''(t) = c^T H \tag{2.2.1}$$

$$y'(t) = c^T P_{i,1} + y'(0) \tag{2.2.2}$$

$$y(t) = c^T P_{i,2} + ty'(0) + y(0) \tag{2.2.3}$$

$$c^T H + \frac{2}{t}(c^T P_{i,1} + y'(0)) - 4(t^2 + 6)(c^T P_{i,2} + ty'(0) + y(0)) = 0 \tag{2.2.4}$$

Solving 2.2.4 yields the coefficient vector. Note that when t is multiplied by $c^T H$,

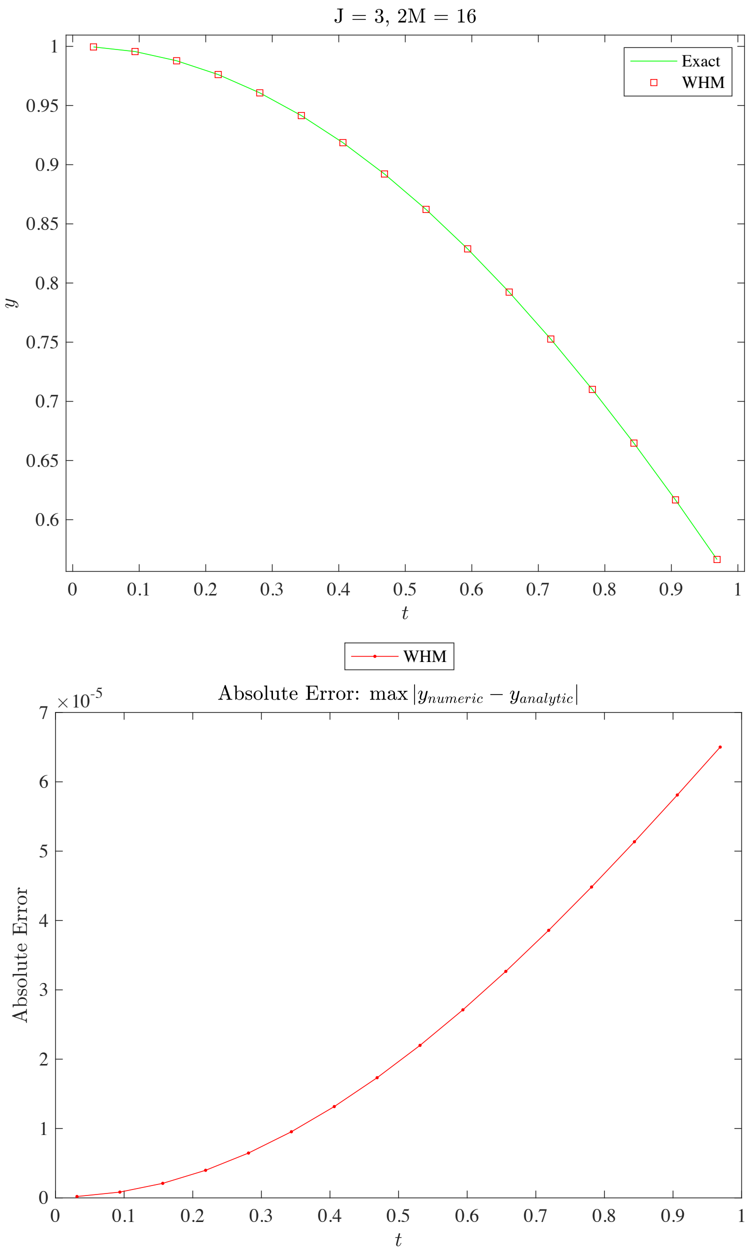

Figure 1: simulation results of $y''(t) + y(t) = 0$

it is the corresponding scalar product that has to be considered. The results are shown in figure 2. Lastly, consider a nonlinear ODE

$$y''(t) + \frac{2}{t}y'(t) + y^n(t) = 0 \tag{2.3}$$

with initial conditions y(0) = 1 and y'(0) = 0.
In this case, the final equation to find the coefficient vector c^T is given by

$$c^T H + \frac{2}{t}(c^T P_{i,1} + y'(0)) + (c^T P_{i,2} + ty'(0) + y(0))^n = 0 \tag{2.2.4}$$

This nonlinear equation in c^T can be solved using the 'fsolve' function in MATLAB. The results are displayed in Figure 3. A similar approach can also be applied for boundary value problems and system of linear ODEs.

3 Partial Differential Equations

In case of partial differential equations, one of the variables is confined to [0,1] and the other is non negative (usually time), generally. the method of approach in this case is to assume the Haar coefficients to be constant for a small duration (say (Δt) of the time variable. This is iterated over the entire duration of the time variable. The results are better if Δt is smaller.
Consider the general Burgers-Huxley equation

$$\frac{\partial u}{\partial t} + \alpha u^\delta \frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} = \beta u(1 - u^\delta)(u^\delta - \gamma) \tag{3.1}$$

Here α,β and γ are parameters >0 and $\gamma \in (0, 1)$. The exact solution subject to the initial condition
u(x,0) = $[\frac{\gamma}{2} + \frac{\gamma}{2}\ tanh(a_1 x)]^{\frac{1}{2}}$
and the boundary conditions
u(0,t) = $[\frac{\gamma}{2} + \frac{\gamma}{2} tanh(-a_1 a_2 t)]^{\frac{1}{2}}$, u(1,t) = $[\frac{\gamma}{2} + \frac{\gamma}{2} tanh(a_1(1 - a_2 t))]^{\frac{1}{2}}$
is given by
u(x,t) = $[\frac{\gamma}{2} + \frac{\gamma}{2} tanh(a_1(x - a_2 t))]^{\frac{1}{2}}$
where $a_1 = \frac{-\alpha\delta + \delta\sqrt{\alpha^2 + 4\beta(1+\delta)}}{4(1+\delta)}$ and $a_2 = \frac{\alpha\gamma}{1+\delta} - \frac{(1+\delta-\gamma)(-\alpha+\delta\sqrt{\alpha^2+4\beta(1+\delta)})}{2(1+\delta)}$

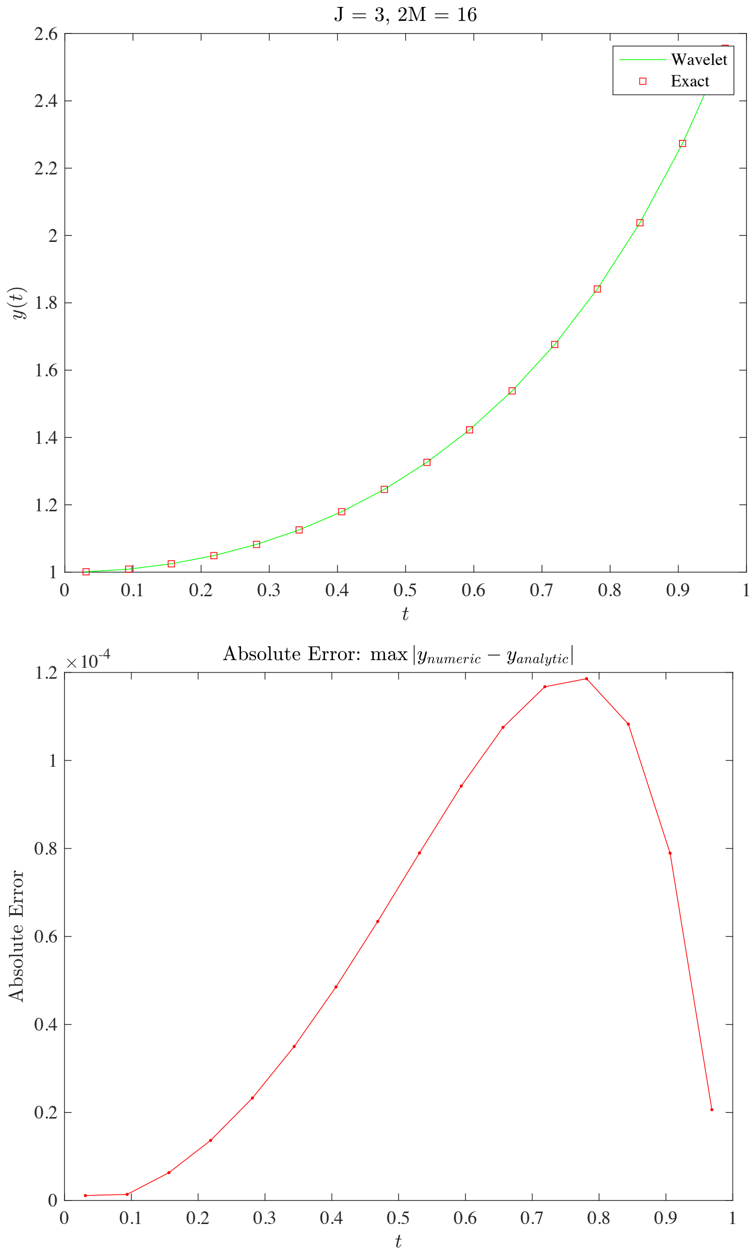

Figure 2: simulation results of $y''(t) + \frac{2}{t}y'(t) - 4(t^2+6)y(t) = 0$

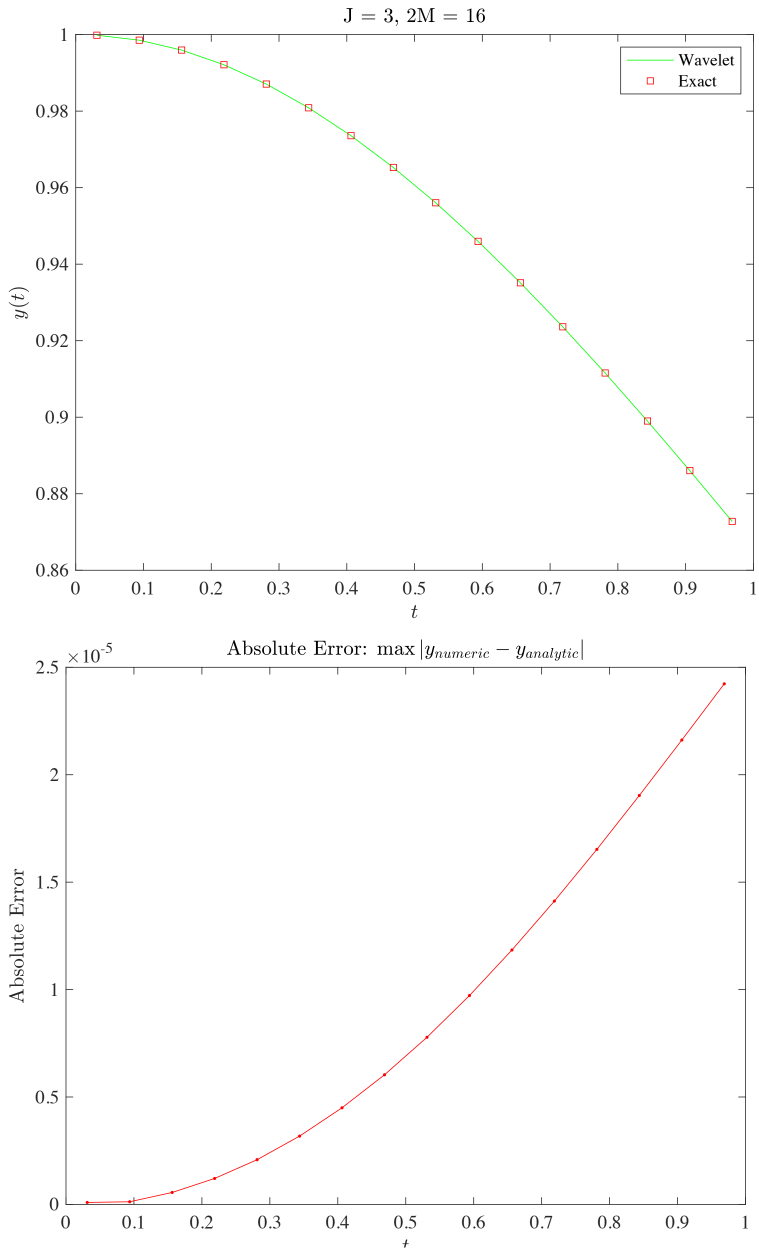

Figure 3: simulation results of $y''(t) + \frac{2}{t}y'(t) + y^n(t) = 0$

Let $\dot{u}$ and u' represent derivatives with respect to t and x respectively. Then, we assume H(x) remain constant for the time subinterval $[t_s, t_{s+1}]$. Proceeding like the cases of ODE,

$$\dot{u}''(x,t) = c^T H(x) \tag{3.2}$$

$$u''(x,t) = (t - t_s)c^T H(x) + u''(x, t_s) \tag{3.3}$$

$$u'(x,t) = (t - t_s)c^T P_{i,1}(x) + u'(x, t_s) - u'(0, t_s) + u'(0,t) \tag{3.4}$$

$$u(x,t) = (t-t_s)c^T P_{i,2}(x) + u(x,t_s) - u(0,t_s) - x[u'(0,t_s) - u'(0,t)] + u(0,t) \tag{3.5}$$

Discretizing $t \to t_{s+1}$ and $x \to x_l$ for the collocation points, using the matrix notation $U_s(l)$ for $u(x_l, t_s)$ and denoting the initial and boundary conditions with the following functions:
$u(x,0) = f(x), u(0,t) = g_0(t), u(1,t) = g_1(t)$ and similarly for their corresponding derivatives, we get the following equation to be solved for the coefficients c^T:

$$c^T[P_{i,2}(l) - x_l P_{i,2}(1)] = U_s''(l) - \alpha U_s^\delta(l) U_s' + \beta U_s(l) U_s^\delta(l))(1 - U_s^\delta(l))(U_s^\delta(l) - \gamma) \tag{3.5}$$

The solution process is started with $t_s = t_0 = 0$. The simulation results are shown in Figures 4 and 5.

4 Fractional Calculus

4.1 Introduction

The Riemann- Liouville Integral operator is defined as

$$J^\alpha u(t) = \begin{cases} \frac{1}{\Gamma(\alpha)} \int_0^t (t-T)^{\alpha-1} u(T) dT & \text{if } \alpha > 0 \\ u(t) & \text{if } \alpha = 0 \end{cases} \tag{4.1}$$

The Caputo fractional derivative is defined as

$$D_*^\alpha u(t) = \begin{cases} \frac{1}{\Gamma(r-\alpha)} \int_0^t \frac{u^{(r)(T)}}{(t-T)^{\alpha-r+1}} dT & \text{if } \alpha = \text{r} \in \text{N} \\ \frac{d^r u(t)}{dt^r} & \text{if } 0 \leq r-1 < \alpha < \text{r} \end{cases} \tag{4.2}$$

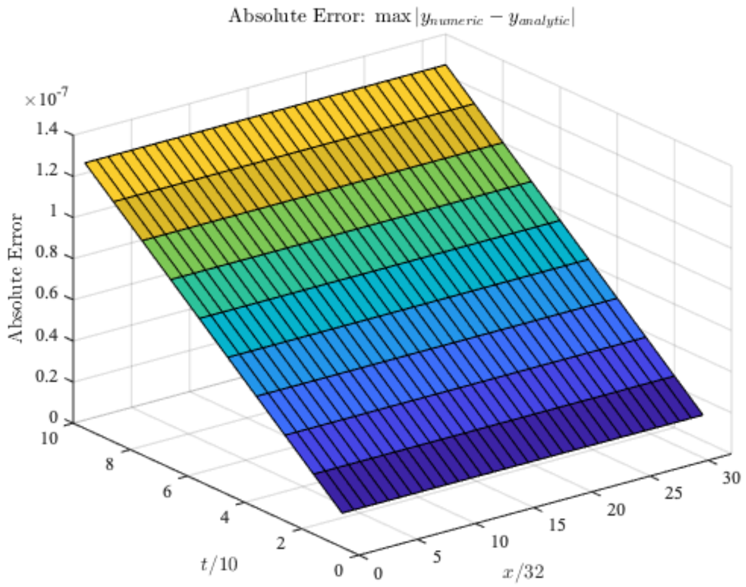

Figure 4: simulation results for $\alpha = 1$, $\beta = 1$, $\gamma = 0.001$ and $\delta = 1$

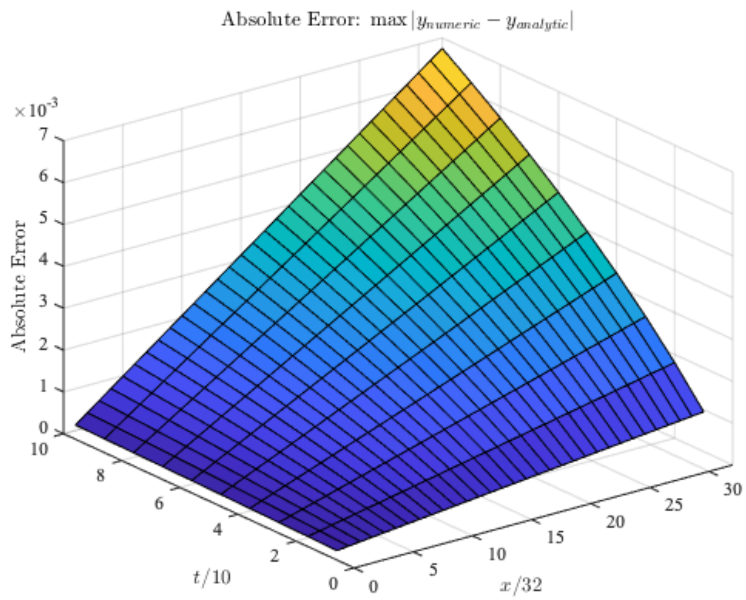

Figure 5: simulation results for $\alpha = 1$, $\beta = 1$, $\gamma = 0.001$ and $\delta = 4$

one useful property linking the definitions in (4.1) and (4.2) is

$$D_*^\alpha J^\alpha u(t) = u(t) \tag{4.3}$$

4.2 Haar Operational Matrix of fractional Order Integration

The definitions given above and the using the definition given in eq. (6) and (7) are used to formulate this matrix P^α, which was derived in [4].

$$P^\alpha H_N(t) = J^\alpha H_N(t) = [Ph_0(t), Ph_1(t), ..., Ph_{N-1}(t)]^T \tag{4.4}$$

where

$$Ph_0(t) = \frac{1}{\sqrt{N}} \frac{t^\alpha}{\Gamma(\alpha+1)} \tag{4.5}$$

$$Ph_i(t) = \frac{1}{\sqrt{N}} \begin{cases} 0 & \text{if } 0 \leq \text{t} < \frac{k-1}{2^j} \\ 2^{j/2}[\frac{1}{\Gamma(\alpha+1)}(t - \frac{k-1}{2^j})^\alpha] & \text{if } \frac{k-1}{2^j} \leq \text{t} < \frac{k-1/2}{2^j} \\ 2^{j/2}[\frac{1}{\Gamma(\alpha+1)}(t - \frac{k-1}{2^j})^\alpha - \frac{2}{\Gamma(\alpha+1)}(t - \frac{k-1/2}{2^j})^\alpha] & \text{if } \frac{k-1/2}{2^j} \leq \text{t} < \frac{k}{2^j} \\ 2^{j/2}[\frac{1}{\Gamma(\alpha+1)}(t - \frac{k-1}{2^j})^\alpha - \frac{2}{\Gamma(\alpha+1)}(t - \frac{k-1/2}{2^j})^\alpha + \frac{1}{\Gamma(\alpha+1)}(t - \frac{k}{2^j})^\alpha] & \text{if } \frac{k}{2^j} \leq \text{t} < 1 \end{cases} \tag{4.6}$$

Therefore, the Matrix of fractional order integration is $P^\alpha = (P^\alpha H)H^T$. From the property in (4.3), we get a formula to calculate the matrix of fractional order differentiation

$$D^\alpha P^\alpha = I \tag{4.7}$$

4.3 Numerical Examples

The first two examples are solved using the method proposed by Yiming Chen, Mingxu Yi, Chunxiao Yu in [5].

Consider the fractional order ODEs

$$D_*^{\frac{1}{3}}u(t) + t^{\frac{1}{3}}u(t) = \frac{3}{2\Gamma(2/3)}t^{\frac{2}{3}} + t^{\frac{4}{3}} \tag{4.3.1}$$

with initial condition u(0) = 0. The exact solution is u(t) = t.

$$D_*^{\frac{1}{4}}u(t) + u(t) = t^4 - \frac{1}{2}t^3 - \frac{3}{\Gamma(4-\alpha)}t^{3-\alpha} + \frac{24}{\Gamma(5-\alpha)}t^{4-\alpha} \tag{4.3.2}$$

with the initial condition u(0)=0. The exact solution is u(t) = t^4 - $\frac{1}{2}t^3$.
The simulation results are shown in figures (6) and (7) respectively.

Consider now, a fractional partial differential equation (FPDE)

$$\frac{\partial^{1/4}u}{\partial x^{1/4}} + \frac{\partial^{1/4u}}{\partial t^{1/4}} = f(x,t) \quad x,t \geq 0 \tag{4.3.3}$$

Given the initial conditions u(0,t) = u(x,0) = 0 and the function f(x,t) = $\frac{4}{3\Gamma(3/4)}(x^{3/4}t + xt^{3/4})$, the exact solution is u(x,t) = xt. The simulation results are shown in figure (8).

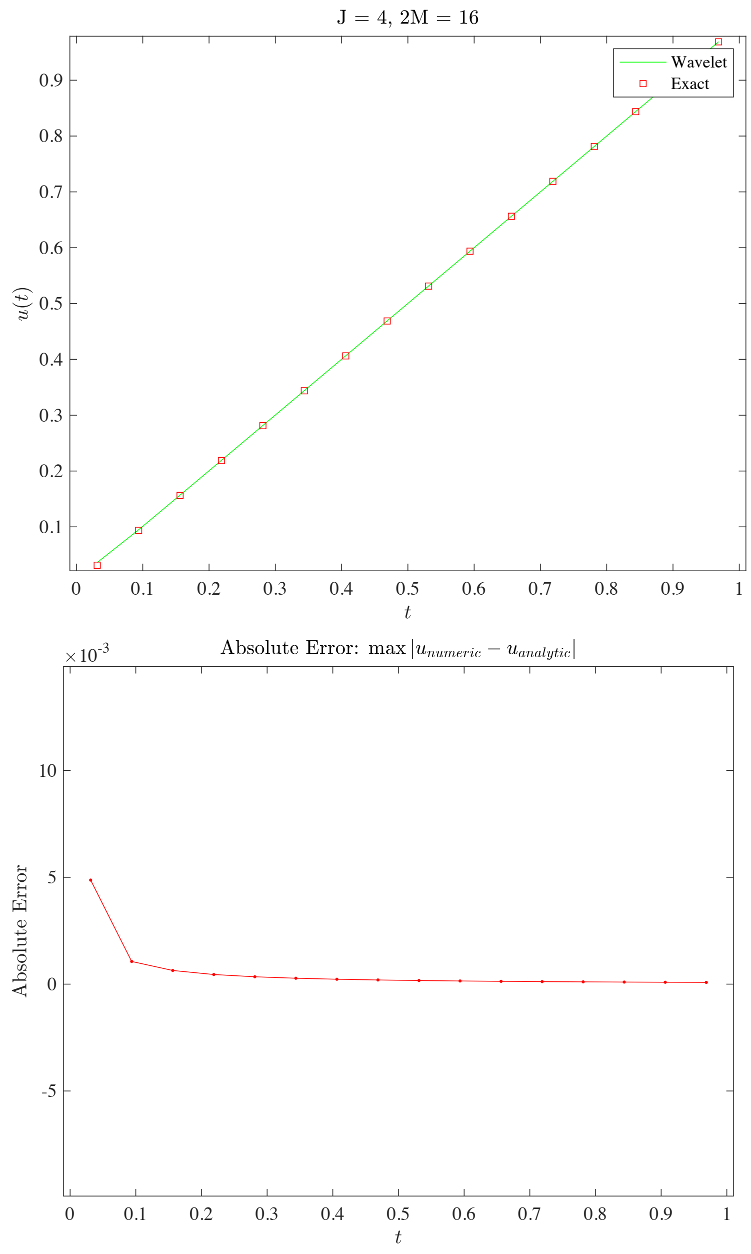

Figure 6: simulation results for $D_*^{\frac{1}{3}}u(t) + t^{\frac{1}{3}}u(t) = \frac{3}{2\Gamma(2/3)}t^{\frac{2}{3}} + t^{\frac{4}{3}}$

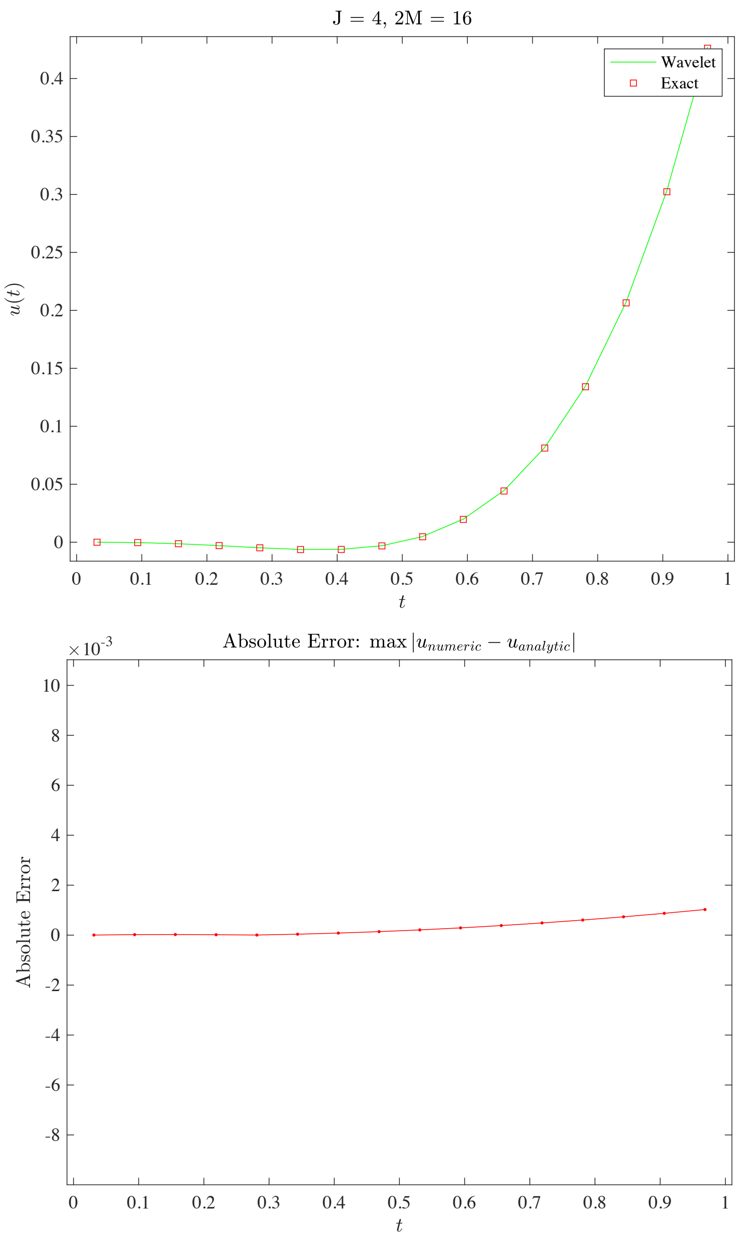

Figure 7: simulation results for $D_*^{\frac{1}{4}}u(t) + u(t) = t^4 - \frac{1}{2}t^3 - \frac{3}{\Gamma(4-\alpha)}t^{3-\alpha} + \frac{24}{\Gamma(5-\alpha)}t^{4-\alpha}$

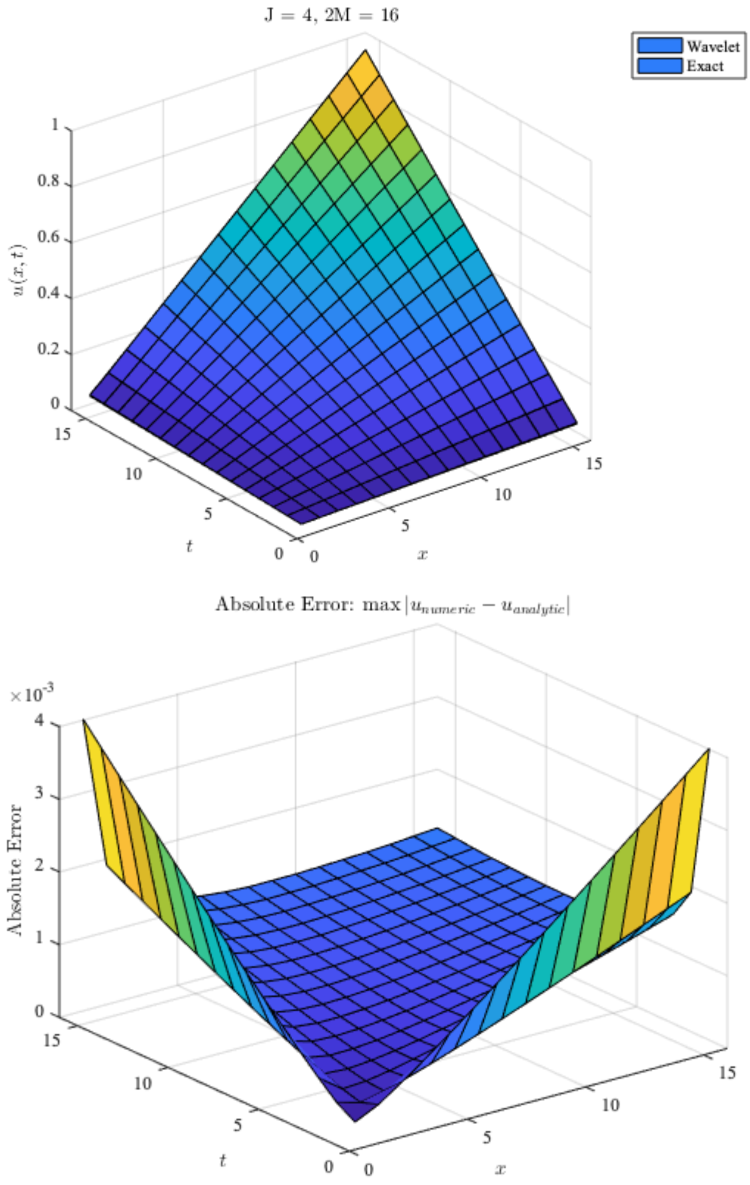

Figure 8: simulation results for $\frac{\partial^{1/4} u}{\partial x^{1/4}} + \frac{\partial^{1/4} u}{\partial t^{1/4}} = f(x,t)$

5 Extension of Fractional Calculus to PDE and ODE

Based on the result derived in the previous section, it can be shown that PDEs and ODEs satisfying certain conditions can be solved using an alternative method which shall be described in this section. The effectiveness of this method is mainly seen in PDEs but can be successfully extended and generalised to several ODEs too.

5.1 First Order PDE

Consider first order partial differential equations of the form

$$\frac{\partial u}{\partial x} + \lambda \frac{\partial u}{\partial t} = f(x,t) \tag{5.1}$$

with the initial conditions $u(x,0) = u(0,t) = k$ where k $\in R$.
The procedure to solve these type of equations is as follows:

$$U = H_x^T C H_t + K \tag{5.2}$$

Here, $U = u(x,t)_{N\times N}$, $H_x and H_t$ are the Haar matrices in x and t respectively and K is the matrix of the constant k. The partial derivatives are then approximated as under:

$$\begin{aligned}\frac{\partial U}{\partial x} = \quad & \frac{\partial (H_x^T C H_t + K)}{\partial x} \\ &= (\frac{\partial H_x}{\partial x})^T C H_t \\ &= (D^\alpha H_x)^T C H_t \\ &= H_x^T D^{\alpha T} C H_t\end{aligned} \tag{5.3}$$

where D^α is as defined in eq. (4.7). Similarly, the time derivative can be approximated as:

$$\frac{\partial U}{\partial t} = H_x^T C D^\alpha H_t \tag{5.4}$$

The function f(x,t) can be approximated in the same way as u(x,t), i.e. $f(x,t) = H_x^T F H_t$. Substituting (5.3) and (5.4) in (5.1),

$$\begin{aligned} H_x^T D^{\alpha T} C H_t + \lambda(H_x^T C D^{\alpha} H_t) &= H_x^T F H_t \\ D^{\alpha T} C + \lambda C D^{\alpha} &= F \end{aligned} \tag{5.5}$$

This equation can be solved for the coefficients C and substituted in (5.1) to estimate the solution u(x,t).

It must be noted here that this method works only for cases where the initial conditions are a constant because the method of approximating a one-dimensional function is different from approximating a two-dimensional function and the two cannot be incorporated in a single equation and solved.

Consider the example

$$\frac{\partial u}{\partial x} + \frac{\partial u}{\partial t} = (x+t)cos(xt)) \tag{5.6}$$

subject to the initial conditions u(x,0) = u(0,t) = 0. The exact solution is u(x,t) = sin(xt).The numerical result is shown in figure(9). The maximum error is also compared between the method where one dimension is approximated and the second iterated over (referred to as M1 in the following table) and the proposed method (referred to as M2 in the following table).

t	Max Error in M1	Max Error in M2
1/64	0.0297817886405494	1.72278006528581e-06
3/64	0.0282890370011744	1.22143437771699e-06
5/64	0.0248128380420626	3.91904590049674e-06
7/64	0.0193508874496209	5.81119053348622e-06
9/64	0.0203432879973288	6.34880745378763e-06
11/64	0.0203432879973288	7.88209411742291e-06
13/64	0.0421403396348165	1.04128907454371e-05
15/64	0.0559560789378517	1.16357289666469e-05
17/64	0.0717078966327158	1.61321804482695e-05
19/64	0.0893852393323705	1.70907574745249e-05
21/64	0.108973275270424	1.90664825713283e-05
23/64	0.130451876396884	1.98245567816224e-05
25/64	0.153794499109762	1.94105395930899e-05
27/64	0.178966968483903	2.00336338916363e-05
29/64	0.205926172385117	2.16737392508271e-05
31/64	0.234618673663449	2.22001601691835e-05
33/64	0.264979250719058	1.76694463425720e-05
35/64	0.296929379138322	1.81957656719689e-05
37/64	0.330375669811293	1.97164108192149e-05
39/64	0.365208281962809	2.02366941896015e-05
41/64	0.401299332848265	1.98196897411851e-05
43/64	0.438501329461676	2.03987686386853e-05
45/64	0.476645651447183	2.19314471681109e-05
47/64	0.515541118451539	2.25886007709786e-05
49/64	0.554972679345680	2.58902048364540e-05
51/64	0.594700265003103	2.66409322511318e-05
53/64	0.634457850558269	2.82860045488498e-05
55/64	0.673952777166892	2.91889220638275e-05
57/64	0.712865387118486	2.94243807030803e-05
59/64	0.750849029554806	3.05313612805058e-05
61/64	0.787530496849142	3.24455899909371e-05
63/64	0.822510953702001	3.37478562590920e-05

Table 1: Maximum error in two methods at different points in the time axis

This method, however fails for higher derivatives of the function and the conventional method has to be used. Even when the initial conditions are non constant, this method does not give accurate results.

5.2 ODEs and System of ODEs

The above methodology can be extended to ODEs because the initial condition in case of a single variable function will always be a constant and hence successive differentiation would not get affected. Take for example, the logistic population growth model

$$\frac{dy}{dt} - ty = te^t \tag{5.7}$$

with initial condition y(0) = 0. The exact solution is $y(t) = te^t$.
The solution is found using the conventional method (M1) of approximating the derivative with the Haar function and with the proposed method (M2) above where the function is approximated using the Haar wavelets and successive differentials are approximated thereafter.

$$\begin{aligned} y &= c^T H \\ \frac{dy}{dt} &= c^T D^\alpha H \end{aligned} \tag{5.8}$$

Here, $\alpha = 1$ in D^α. The equation to find the coefficients is therefore

$$c^T D^\alpha H - t(c^T H) = te^t \tag{5.8}$$

The maximum errors in both the methods is shown in table 2. Lastly, consider the system of linear ordinary differential equations:

$$\begin{aligned} y_1'(t) &= y_3(t) - cost, \quad y_1(0) = 1 \\ y_2'(t) &= y_3(t) - e^t, \quad y_2(0) = 0 \\ y_3'(t) &= y_1(t) - y_2(t), \quad y_3(0) = 2 \end{aligned} \tag{5.9}$$

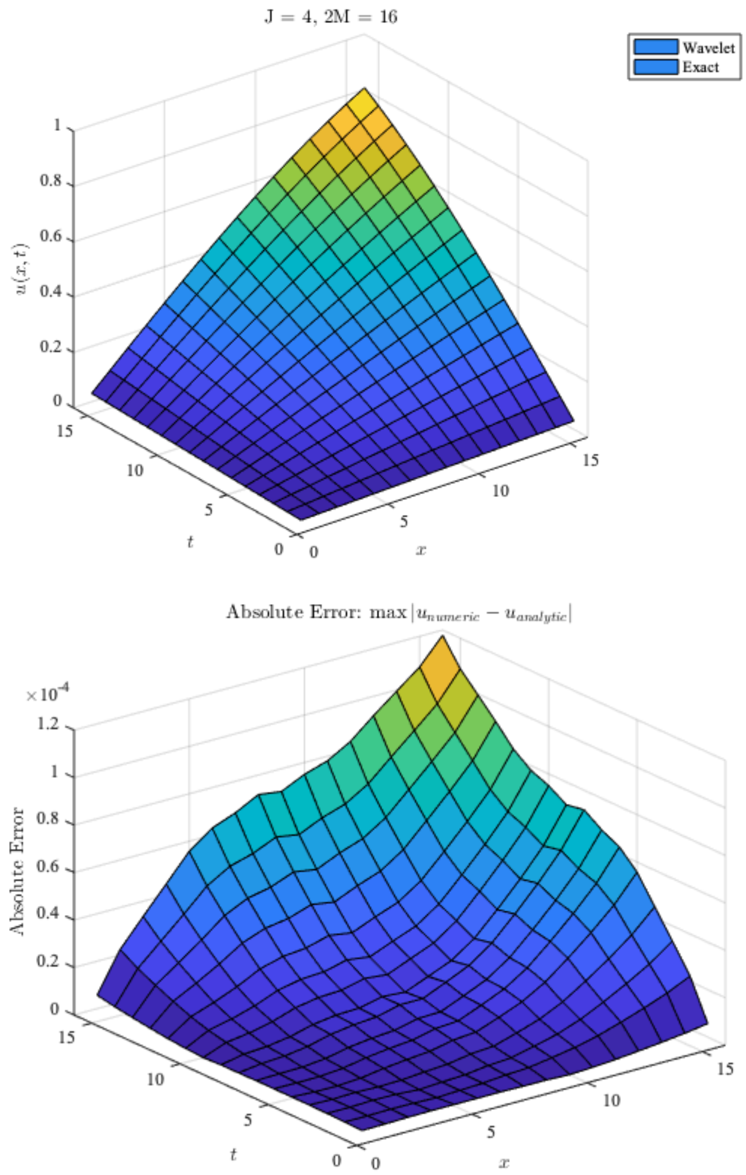

Figure 9: simulation results for $\frac{\partial u}{\partial x} + \frac{\partial u}{\partial t} = (x+t)cos(xt))$

J	Max Error in M1	Max Error in M2
1	0.2593	0.2593
2	0.0762	0.0762
3	0.0209	0.0209
4	0.0057	0.0055
5	0.0016	0.0014

Table 2: Maximum error comparison between the two methods.

y_i	Max Error in M1	Max Error in M2
y_1	0.00230617686339075	0.00231512933214610
y_2	0.00589442817110275	0.00589384656343854
y_3	0.00405803224412917	0.00405803224412917

Table 3: Maximum error comparison between the two methods for J $=$ 4 .

In this case, the solution of $y_{3(t)}$ is found first and then substituted in the equations of $y_2(t)$ and $y_1(t)$. Table 3 compares the errors as in the previous two cases.

6 Conclusions

This report shows the effectiveness of the Haar wavelets in approximating functions, their derivatives and integrals, and also give better results than the Taylor series approximation or finite difference methods in approximating the numerical solutions of Ordinary Differential Equations, Partial Differential Equations and also Fractional Order Differential Equations.

The report also presents an alternative method of approximating the above mentioned equations and functions by extending the definition of fractional order calculus to first order differential equations, be they partial or ordinary. This methodology, presented in Section 5, is a new extension which I believe has not been done before. It has only been used in fractional calculus, but it can be readily seen that it is very effective in case of first order PDEs with constant initial conditions and at par with the conventional method in case of ODEs and a system of ODEs. Since an n^{th} order ODE can be converted to a system of n first order ODEs, this methodology works for any ODE.

7 Future works

This report only focuses on Haar wavelets, but the idea can be generalized for any family of wavelets, as has been done in the case of Legendre and Chebyshev wavelets.

Moreover, the wavelet function can be extended to any domain outside [0,1]. This transformation of variables can help solve equations for any values of the dependent variable. The methodlogy introduced and discussed in **section 5** can be used as a basis to solve PDEs with non constant initial conditions.

Another kind of equations that have yet to be solved using wavelets is the system of ODEs which are coupled, as in the case of the epidemic models and in the Lotka-Volterra models. This is something worthy of research as this has not been done with sufficient accuracy.

References

[1] Haar A. Zur theorie der orthogonalen Funktionsysteme. Math Annal 1910;69:331–71.

[2] Chen, Y., Yi, M., Yu, C. (2012). Error analysis for numerical solution of fractional differential equation by Haar wavelets method. Journal of Computational Science, 3(5), 367-373.

[3] JIN-SHENG GUF WEI-SUN JIANG (1996) The Haar wavelets operational matrix of integration, International Journal of Systems Science, 27:7, 623-628, DOI: 10.1080/00207729608929258

[4] Çelik, İ. (2012). Haar wavelet method for solving generalized Burgers–Huxley equation. Arab Journal of Mathematical Sciences, 18(1), 25-37.

[5] Ü. Lepik (2009) Haar wavelet method for solving stiff differential equations, Mathematical Modelling and Analysis, 14:4, 467-481

[6] Lepik, Ü. (2011). Solving PDEs with the aid of two-dimensional Haar wavelets. Computers Mathematics with Applications, 61(7), 1873-1879.

[7] Wang, L., Ma, Y., Meng, Z. (2014). Haar wavelet method for solving fractional partial differential equations numerically. Applied Mathematics and Computation, 227, 66-76.