Cihan University-Erbil Scientific Journal (CUESJ)

Research Article

Solution of Multi-order Fractional Differential Equation Based on Conformable Derivative by Shifted Legendre Polynomial

Salim S. Mahmood[1], Kamaran J. Hamad[1], Milad A. kareem[1], Asrin F. Shekh[2]

[1]Department of Mathematic, Faculty of Science, Soran University, Kurdistan Region, Iraq, [2]Department Information Technology, Erbil Polytechnic University, Kurdistan Region, Iraq

ABSTRACT

The aim of this article is the way for finding approximation solution of multi-order fractional differential equation with conformable sense with use approximated function by shifted Legendre polynomial, the method is easy and powerful for get our results of linear and non-linear equation, the background idea behind this method is finding system of algebra after achieve messing variable is that mean obtains approximate solution, a few example are illustrate for presented how much our method is capable.

Keywords: Conformable derivative, fractional derivative, shifted Legendre polynomial, fractional deferential equation, Python program

INTRODUCTION

There are numerous phenomena in various disciplines of research can work quite well represented by models using fractional calculus mathematic techniques. In many circumstances, there are no analytic or precise solutions to fractional differential equations. As a result, novel approaches for deriving analytical and approximation fractional differential equation solutions have been devised.[1,2]

Little number techniques and methods have gotten a lot of attention in recent years and have proven to be a useful tool for fractional calculus.

According to the studies, the Caputo fractional derivative is commonly incorporated in fractional calculus computed using spectral methods. To compute fractional differential equations with Caputo derivatives, Legendre polynomial has been employed.[3]

In addition, Chebyshev wavelet is used for solving non-linear fractional differential equation,[4,5] operation matrix is used for solving fractional multi-order differential equation,[3] and spectral collocation method is used for compute this kind of differential equation by Laguerre polynomial (LP).[6]

The aim of this article is finding a solution to our fractional equations both kinds linear and non-linear using shifted LP. In this case, if number of N is large means our approximate, you can't use hand writing for find solution, so we prefer write a program for such case by Python.

Hence, in our method, try to find approximating solution of multi-order fractional differential equation based on conformable derivative using shifted Legendre polynomial with program of Python.

PRELIMINARIES

Definition 1: Given function h: $[0, \infty) \to R$ then the conformable fractional derivative of h of order α is determined by

$$D_\alpha h(x) = \lim_{\epsilon \to 0} \frac{h(x + \epsilon x^{1-\alpha}) - h(x)}{\epsilon}, \quad \forall x > 0, \alpha \in (0,1)$$

Some time, write $h^a(x)$ for $D_a h(x)$ to indicate the conformable derivative of h of order α

Definition 2: In this paper, we need to use Legendre polynomials in distance $x=[0,1]$ so using change variable $x=2x-1$, these polynomials can be obtained as follows

$$E_n(x) = (-1)^n \sum_{k=0}^{n} \binom{n}{k} \binom{n+k}{k} (-x)^k \tag{2}$$

Where $E_0(x)=1$, $E_0(x)= 2x-1$ on the interval [0,1], the shifted LP is defined as $E_n(x) = \sum_{k=0}^{n} (-1)^{n+k} \frac{(n+k)!\,x^k}{(n-k)!(k!)^2}, n = 0,1,2,\ldots$ (3)

Corresponding Author:
Salim S. Mahmood, Department of Mathematic, Faculty of Science, Soran University, Kurdistan Region, Iraq.
E-mail: salimsaeedmahmood@gmail.com

Received: August 25, 2021
Accepted: November 11, 2021
Published: December 20, 2021

DOI: 10.24086/cuesj.v5n2y2021.pp64-68

And explicit way for finding shifted Legendre polynomial is "Rodrigues formula"

$$E_n(x) = \frac{1}{n!}\frac{d^n}{dx^n}(x^2 - x)^n \tag{4}$$

Each continuous function h (x) in the interval [0,1] can be written using the transferred Legendre polynomials as follows:

$$h(x) \approx \sum_{i=0}^{N} a_i E_i(x) \tag{5}$$

Where, a_i is obtained using the following relation:

$$a_i = (2i+1)\int_0^1 h(x)E_i(x)dx \tag{6}$$

Definition 3: The orthogonal property of these polynomials is as follows

$$\int_0^1 E_i(x)E_j(x)dx = \begin{cases} \frac{2}{2i+1} & i = j \\ 0 & i \neq j \end{cases} \tag{7}$$

The orthogonal property of the expansion of any function makes it possible in terms of Legendre polynomials.

Theorem 1: Let h_N (x) function can be approximate by shifted LP like that

$$h_N(x) = \sum_{i=0}^{N} a_i E_i(x) \tag{8}$$

After that, for any derivative we use[7]

$$D^{\alpha}(h_N(x)) = \sum_{i=\lceil\alpha}^{N}\sum_{k=\lceil\alpha}^{i} a_i W_{i,k}^{\alpha} x^{k-\alpha} \tag{9}$$

Where, $\lceil\alpha$ is define as least integer greater than or equal to α and

$$W_{i,k}^{(\alpha)} x^{k-\alpha} = \frac{(-1)^{i+k}\Gamma(i+k+1)}{\Gamma(i-k+1)\Gamma(k+1)\Gamma(k-n+1)} \tag{10}$$

Proof: According to the property of the conformable derivative and using the estimated function definition h_N (x), we get the following results:

$$D^{\alpha}(h_N(x)) = \sum_{i=0}^{N} a_i D^{\alpha}(E_i(x)) \tag{11}$$

$$D^{\alpha}(E_i(x)) = \sum_{k=\lceil\alpha}^{i} \frac{(-1)^{i+k}\Gamma(i+k+1)}{\Gamma(i-k+1)\Gamma(k+1)\Gamma(k-n+1)} x^{k-\alpha} \tag{12}$$

By substitution, we get

$$D^{\alpha}(h_N(x)) = \sum_{i=\lceil\alpha}^{N}\sum_{k=\lceil\alpha}^{i} a_i \frac{(-1)^{i+k}\Gamma(i+k+1)}{\Gamma(i-k+1)\Gamma(k+1)\Gamma(k-n+1)} x^{k-\alpha} \tag{13}$$

NUMERICAL SCHEME

In here presented how solve the multi-order fraction differential equation by Python program for get analytic and numerical results.

Assume the initial condition for this equation which is a multi-order fractional differential equation[8]

$$D^{\alpha}(h(x)) + \sum_{k=1}^{r-1} B_k(x)h(x) + B_r(x)h(x) = f(x), \qquad x \in [0,1] \tag{14}$$

$$h^{i}(0) = q_i, \qquad i = 0,1,\ldots,\lceil\alpha - 1 \tag{15}$$

Where, $n-1<\alpha\leq n$, B_k (x), $k=1,2,\ldots.r$ and f (x) are functions that are well-known to be continuous on [0,1] and $q_i, i = 0,1,\ldots,\lceil\alpha - 1$, constants are provided. Assume the fractional differential Equation (14) has an answer that may be represented as

$$h_N(x) = \sum_{i=0}^{N} a_i E_i(x) \tag{16}$$

By the initial equation, we have $\lceil\alpha$ equation and is depend on our approximate about N so all equation number is N+1, except $\lceil\alpha$ equation other equation get by the finding root of shifted Legendre polynomial $E_{N+1-\lceil\alpha}(x)$ and put on Equation (14). So solving this system of equation is our approximation solution for multi-order fractional differential Equation (14).

Example 1: (Bagley-Torvik equation). The inhomogeneous Bagley-Torvik initial value problem is

$$D^2(h(x)) + D^{\frac{3}{2}}(h(x)) + h(x) = 1 + x, \quad x \in [0,1] \tag{17}$$

With initial assumption $h(0)=1$, $h'(0)=1$ and $N=2$

Solution:

$$h_N(x) = \sum_{i=0}^{N} a_i E_i$$

$$h_2(x) = \sum_{i=0}^{2} a_i E_i = a_0 E_0(x) + a_1 E_1(x) + a_2 E_2(x)$$

$$h_2(x) = a_0 + 2a_1 x - a_1 + 6a_2 x^2 - 6a_2 x + a_2$$

So use initial condition for finding the α equation and for any derivative, we use

$$D^{\alpha}(h_N(x)) = \sum_{i=\lceil\alpha}^{N}\sum_{k=\lceil\alpha}^{i} a_i W_{i,k}^{\alpha} x^{k-\alpha}$$

$$h_2(0) = a_0 - a_1 + a_2 = 1 \rightarrow (1)$$

$$D^1((h_2(0)) = 2a_1 - 6a_2 = 1 \rightarrow (2)$$

For finding N+1 equation, we need use the root of the polynomial $E_{N+1-\lceil\alpha}(x) = E_1(x) = 2x - 1 = 0 \rightarrow x = 0.5$

So put the root in the Equation (17) after, we calculate all structure instead each x variable.

$$D^2(h_2(0.5)) = 12a_2$$
$$D^{\frac{3}{2}}(h_2(0.5)) = 12\sqrt{0.5}a_2$$
$$h_2(0.5) = a_0 - \frac{1}{2}a_2$$

Substitute each one of $D^2(h_2(0.5)), D^{\frac{3}{2}}(h_2(0.5))$ and $h_2(0.5)$ in the Equation (17) and instead of each x put root of shifted Legendre polynomial

$$12a_2 + 12\sqrt{0.5}a_2 + a_0 - \frac{1}{2}a_2 = 1 + \frac{1}{2}$$
$$= a_0 + (\frac{23}{2} + 6\sqrt{2})a_2 = \frac{3}{2} \to (3)$$

So we have a linear system after solve by matrix operation, we get

$$\begin{pmatrix} 1 & -1 & 1 \\ 0 & 2 & -6 \\ 1 & 0 & \frac{23}{2} + 6\sqrt{2} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ \frac{3}{2} \end{pmatrix}$$
$$a_0 = \frac{3}{2}, a_1 = \frac{1}{2}, a_2 = 0$$
$$h_2(x) = 1 + x$$

Therefore, the obtained solution is the exact solution of this problem, code of Python for this example is given.[9]

Example 2: Suppose the following fractional linear differential equation with variable coefficients

$$D^{\frac{3}{2}}(h(x)) + 2D(h(x)) + 3\sqrt{x}D^{\frac{1}{2}}(h(x)) + (1-x)$$
$$h(x) = 2\sqrt{x} + 4x + 7x^2 - x^3, \quad x \in (0,1]$$

With the primary conditions $h(0)=0$, $h'(0)=0$ and $N=2$

Solution:

$$h_N(x) = \sum_{i=0}^{N} a_i E_i$$
$$h_2(x) = a_0 E_0(x) + a_1 E_1(x) + a_2 E_2(x)$$
$$h_2(x) = a_0 + 2a_1 x - a_1 + 6a_2 x^2 - 6a_2 x + a_2$$

For finding $N+1$ equation, we need use the root of the polynomial $E_{N+1-\lceil\alpha}(x) = E_1(x) = 2x - 1 = 0 \to x = 0.5$

So we put the root in the Equation (18) after, we calculate all structure instead each x variable

$$D^{\frac{3}{2}}(h_2(0.5)) = 12a_2(\frac{1}{2})^{\frac{1}{2}}$$
$$D^1(h_2(0.5)) = 2a_1$$
$$D^{\frac{1}{2}}(h_2(0.5)) = 2a_1\sqrt{0.5} - 6a_2\sqrt{o.5} + 12a_2\sqrt{0.5}$$
$$h_2(0.5) = a_0 - \frac{1}{2}a_2$$

We substitute each one of $D^{\frac{3}{2}}(h_2(0.5)), D^1(h_2(0.5)), D^{\frac{1}{2}}(h_2(0.5))$ and $h_2(0.5)$ in the Equation (18) and instead of each x we put root (0.5)

$$\frac{1}{2}a_0 + 7a_1 + 8.2352813742238570a_2 = 5.03921356237095 \to (1)$$

$$h_2(0) = a_0 + a_1 + a_2 = 0 \to (2)$$
$$D^1((h_2(0)) = 2a_1 - 6a_2 = 0 \to (3)$$

So we have a linear system, we solved by matrix operation

$$\begin{pmatrix} \frac{1}{2} & 7 & 8.2352813742238570 \\ 1 & -1 & 1 \\ 0 & 2 & -6 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} 5.03921356237095 \\ 0 \\ 0 \end{pmatrix}$$

$$a_0 = \frac{1}{3}, a_1 = \frac{1}{2}, a_2 = \frac{1}{6}$$
$$h_2(x) = x^2$$

Therefore, the obtained solution is the precise solution to this problem.

Example 3: Consider the following problem with an initial value.

$$D^4(h(x)) + D^{\frac{7}{2}}(h(x)) + h(x)^3 = x^9, \quad x \in (0,1] \qquad (18)$$

With initial condition $h(0)=0$, $h'(0)=$ $h''(0)=0$ and $h'''(0)=6$, $N=4$

$$h_N(x) = \sum_{i=0}^{N} a_i E_i = a_0 E_0(x) + a_1 E_1(x) + a_2 E_2(x) + a_3 E_3(x) + a_4 E_4(x)$$
$$h_4(x) = 70a_4 x^4 + (20a_3 - 140a_4)x^3 + (6a_2 - 30a_3 + 90a_4)x^2 +$$
$$(2a_1 - 6a_2 + 12a_3 - 20a_4)x + a_0 - a_1 + a_2 - a_3 + a_4$$

For finding $N+1$ equation we need use the root of the polynomial $E_{N+1-\lceil\alpha}(x) = E_1(x) = 2x - 1 = 0 \to x = 0.5$

So we put the root in the Equation (19) after, we calculate all structure instead each x variable so our equation is

$$\left(a_0 + \frac{a_2}{2} + \frac{3a_4}{8}\right)^3 + (1680 + 840\sqrt{2})a_4 - \frac{1}{512} = 0 \to (1)$$

So by initial condition, we have α equation

$$h_4(0) = a_0 - a_1 + a_2 - a_3 + a_4 = 0 \to (2)$$
$$D^1(h_4(0)) = 2a_1 - 6a_2 + 12a_3 - 20a_4 = 0 \to (3)$$
$$D^2(h_4(0)) = 12a_2 - 60a_3 + 180a_4 = 0 \to (4)$$
$$D^3(h_4(0)) = 120a_3 - 840a_4 - 6 = 0 \to (5)$$

Now use Newton iteration method for finding the messing variable and our result will be

$$a_0 = \frac{1}{4}, a_1 = \frac{9}{20}, a_2 = \frac{1}{4}, a_3 = \frac{1}{20} \quad and \; a_4 = 0$$

If we put those number in our $h_4(x)$ our result will be $h_4(x)=x^3$ which is the precise solution to this issue.

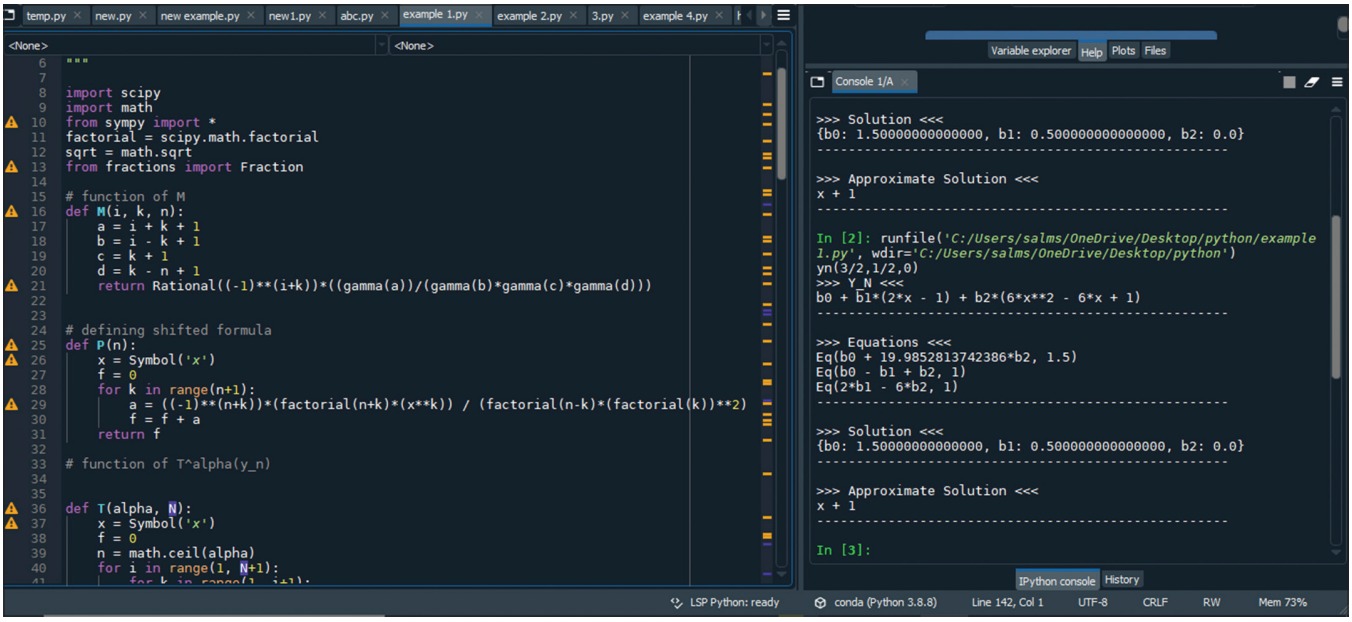
CONCLUSION

In this paper, using the method and technique for get approximate solution of fractional differential equations with initial conditions and conformable sense are used to characterize the fractional derivatives, the solution can be easily analyzed for any number of different values of x since it is written as a truncated Legendre series. Furthermore, the suggested strategy's validity and applicability are proved by numerical findings. The numerical results show that as the number of N terms is raised, the method converges. With tiny N, we get the precise answer for several examples, all examples are achieved by the use of Python programming to create quick algorithms. The method presented here can be used to solve the high order of N for conformable fractional differential equations as well as conformable fractional partial differential equations.

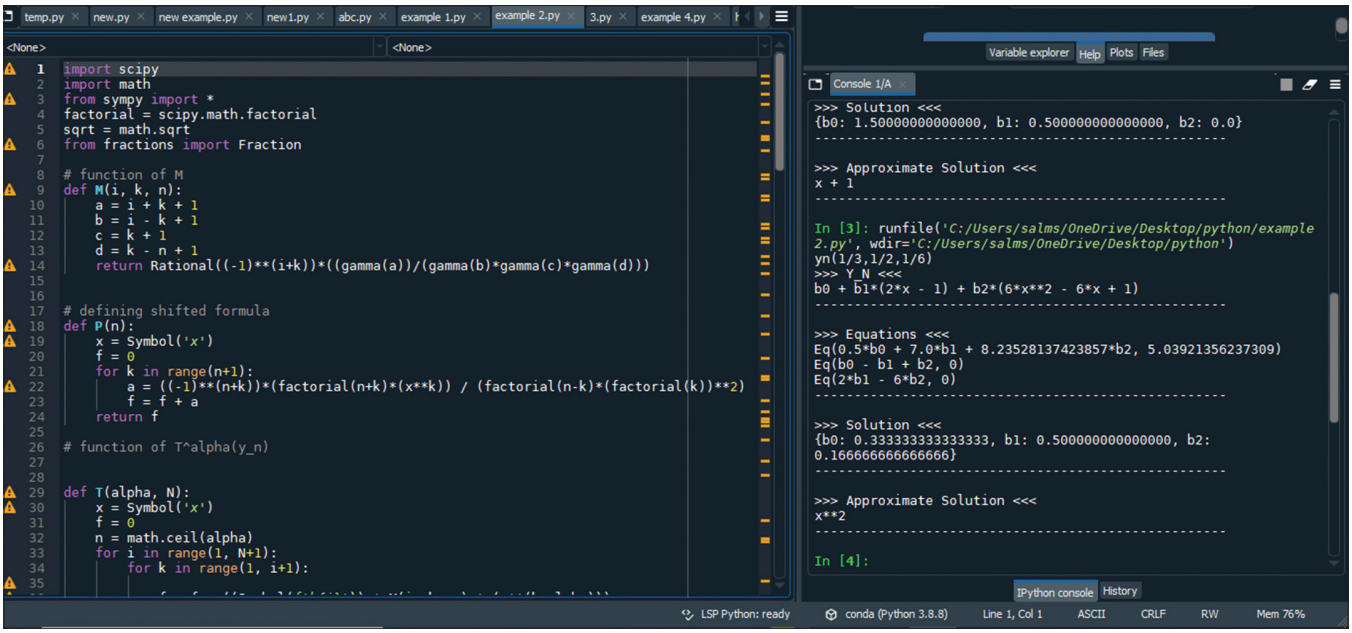
REFERENCES

1. H. Khalil, R. H. Ali Khan, M. A Al-Smadi, A. Freihat and N. Shawagfeh. New operational matrix for shifted legendre polynomials. *Punjab University Journal of Mathematics*, vol. 47, no. 1, pp. 1-23, 2020.
2. R. A. Khalil. A new definition of fractional derivative. *Journal of Computational and Applied Mathematics*, vol. 264, pp. 65-70, 2014.
3. A. B. Kisabo, U. C. Uchenna, and F. A. Adebimpe. Newton's method for solving non-linear system of algebraic equations (NLSAEs) with MATLAB/Simulink® and MAPLE®. *American Journal of Mathematical and Computer Modelling*, vol. 2, no. 4, pp. 117-131, 2017.
4. S. S. Ezz-Eldien, J. A. Machado, Y. Wang and A. A. Aldraiweesh. An algorithm for the approximate solution of the fractional Riccati differential equation. *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 20, no. 6, pp. 661-674, 2019.
5. L. Yuanlu. Solving a nonlinear fractional differential equation using Chebyshev wavelets. *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 9, pp. 2284-2292, 2010.
6. E. Hesameddini and E. Asadollahifar. Numerical solution of multi-order fractional differential equation. *IJNAO*, vol. 5, no. 1, pp. 37-48, 2015.
7. M. Abul-Ez, M. Zayed, A. Youssef and M. De la Sen. On conformable fractional Legendre polynomials and their convergence properties with applications. *Alexandria Engineering Journal*, vol. 59, no. 6, pp. 5231-5245, 2020.
8. H. Çerdik Yaslan and F. Mutlu. Numerical solution of the conformable differential equations via shifted Legendre polynomials. *International Journal of Computer Mathematics*, vol. 97, no.5, pp. 1016-1028, 2020.
9. M. M. Khader, T. S. El Danaf and A. S. Hendy. Efficient spectral collocation method for solving multi-term fractional differential equations based on the generalized Laguerre polynomials. *J. Fractional Calc. Appl*, vol. 3, pp. 1-14, 2012.

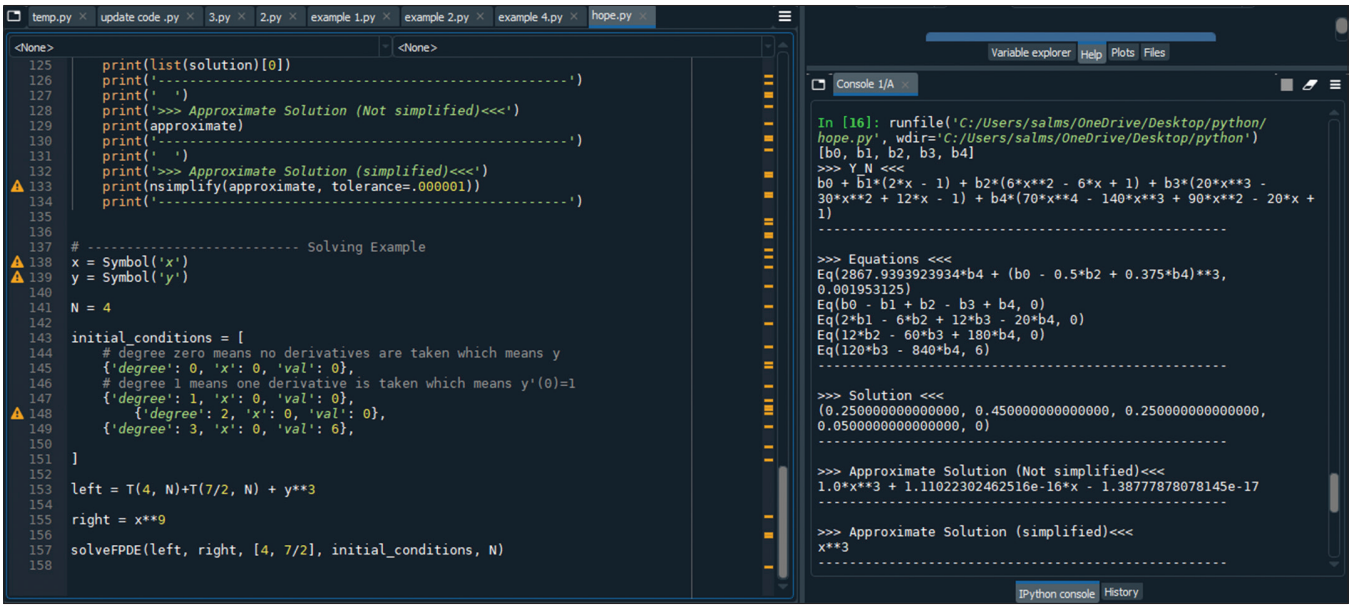
APPENDIX



Result of Python code for example 1



Result of Python code for example 2



Result of Python code for example 3