Numerical Solutions of (1+1)-Dimensional Diffusion Equation via Lucas and Fibonacci Polynomials

Kumail Raza
Supervised By: Dr. Abdul Ghafoor

Institute of Numerical Sciences
Kohat University of Science and Technology (KUST)
Kohat KPK Pakistan

November 11, 2022

Outlines

Motivation

- Most of the physical problems can be modelled using partial differential equations (PDEs).
- When a PDE is non-linear then analytical solution is not always possible so that's why numerical method is a good way for the solution of those PDE's.
- In literature the finite difference method is a good way to use for the solution of PDE's.
- Our proposed scheme is concerned to solve (1+1)-dimensional diffusion equation via Lucas and Fibonacci polynomials together with finite difference scheme.

Introduction to Numerical Methods

- In numerical methods, Euler used finite difference method FDM in 1768 [1]. But did not gain much attention until the beginning of 20^{th} century. FDM are efficient in case of regular domain but it lacked the feasibility incase of irregular domain.
- To overcome this difficulty, finite element method FEM and finite volume method FVM were introduced that have great flexibility to deal problems having complex geometry. Mesh generation in FEM is pain staking process.
- Recently people are using polynomial based approximation.

[1] W.F. Ames, Academic Press, Orlando, 1977.

Introduction to Numerical Methods

- In 2017, Ömer Oruc studied a numerical solution of generalized Benjamin Bona Mahony Burger's equation by using polynomials . [2]
- Nowadays people are using hybridized scheme in which finite differences are used for the temporal part while for space discretization different kind of polynomials are used.
- In our work we used Lucas and Fibonacci polynomials for the space discretization and forward difference is for temporal discretization.
- After this the PDE with collocation approach is converted into system of algebric equations which are easily solvable.

[2] Ömer Oruc, A new algorithm based on Lucas polynomials for approximate solution of 1D and 2D nonlinear generalized Benjamin-Bona-Mahony-Burgers equation, 2017.

Fibonacci Polynomials

The Fibonacci polynomials are defined as

$$F_n(x) = \begin{cases} 0, & if \quad n = 0 \\ 1, & if \quad n = 1 \\ xF_{n-1}(x) + F_{n-2}(x), & if \quad n \geq 2 \end{cases} \tag{1}$$

The first few Fibonacci polynomials can be calculated as:

$$F_0(x) = 0,$$
$$F_1(x) = 1,$$
$$F_2(x) = x,$$
$$F_3(x) = x^2 + 1,$$
$$F_4(x) = x^3 + 2x.$$

By evaluating Fibonacci polynomials at $x = 1$ the Fibonacci numbers can be obtained. These Fibonacci numbers are 1, 1, 2, 3, 5, 8, ... etc.

Function approximation with Fibonacci polynomials

Suppose that a function $u(x)$ is continuous and can be expanded into Fibonacci polynomials as

$$u(x) = \sum_{n=0}^{\infty} a_n F_n(x). \qquad (2)$$

In practice, a truncated version of the above expansion is needed. So, Eq. (1) can be written in truncated form as follows:

$$u(x) \cong \sum_{n=0}^{N} a_n F_n(x) = F(x)A, \qquad (3)$$

where $F(x) = [F_0(x), F_1(x), ..., F_N(x)] \quad and \quad A = [a_0, a_1, ..., a_n]^T$.
There is a relation between the vector $F(x)$ and its derivative vector $F'(x)$ as,

$$F'(x) = F(x)D, \tag{4}$$

where

$$D = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & d & \\ 0 & & & \end{bmatrix}_{(N+1)\times(N+1)}$$

and d is $N \times N$ matrix which is given in [3]

$$d_{i,j} = \begin{cases} i \sin \frac{(j-i)\pi}{2}, & j > i, \\ 0, & j \leq i. \end{cases} \tag{5}$$

[3] A.B.Koç, M.Çakmak, A.Kurnaz, K.Uslu, A new Fibonacci type collocation procedure for boundary value problems, 2013.

For example if we take N = 3 then we have

$$D= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

By plugging the calculated value of D in Eq.(4) we obtain

$$F'(x) = F(x)D = [0, 1, x, x^2 + 1] \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{bmatrix} = [0, 0, 1, 2x]. \qquad (6)$$

Now taking k-th derivative , Eq.(4) reduces to

$$F''(x) = F'(x)D = F(x)D^2$$

,

$$F'''(x) = F''(x)D = F(x)D^3,$$

$$\vdots$$

$$F^{(k)}(x) = F(x)D^k.$$

Lucas Polynomials

Lucas polynomials are defined as:

$$L_n(x) = \begin{cases} 2, & if \quad n = 0 \\ x, & if \quad n = 1 \\ xL_{n-1}(x) + L_{n-2}(x), & if \quad n \geq 2. \end{cases} \tag{7}$$

From Eq.(7), the first few Lucas polynomials can be found as,

$$L_0(x) = 2,$$

$$L_1(x) = x,$$

$$L_2(x) = x^2 + 2,$$

$$L_3(x) = x^3 + 3x,$$

$$L_4(x) = x^4 + 4x^2 + 2.$$

By letting x = 1 in the Lucas polynomials the Lucas numbers are obtained.

Function approximation with Lucas Polynomials

Let $u(x)$ be a continuous function and can be expressed in terms of Lucas polynomials as,

$$u(x) = \sum_{n=0}^{\infty} a_n L_n(x). \tag{8}$$

The above series expansion of $u(x)$ contains infinite terms. If this expansion terminated at N finite terms, Eq. (8) takes following terminated form,

$$u(x) \cong \sum_{n=0}^{N} a_n L_n(x) = L(x)A, \tag{9}$$

where $L(x) = [L_0(x), L_1(x), ..., L_N(x)]$ and $A = [a_0, a_1, ..., a_n]^T$.
The derivative of Eqs.(8)-(9) can be written as:

$$u^{(k)}(x) = \sum_{n=0}^{\infty} a_n L_n^{(k)}(x). \tag{10}$$

$$u^{(k)}(x) \cong \sum_{n=0}^{N} a_n L_n^{(k)}(x) = L^{(k)}(x)A, \quad k = 0, 1, ..., m, \tag{11}$$

in which

$$L^{(k)}(x) = [L_0^{(k)}(x), L_1^{(k)}(x), ..., L_N^{(k)}(x)]$$

.

Proposed Methodology for (1+1)-Dimensional Diffusion Equation

Here we discuss the proposed scheme for (1+1)-dimensional diffusion equation. First we use forward difference for temporal part with θ-weighted $(0 \leq \theta \leq 1)$ scheme and then Lucas and Fibonacci Polynomial for space discretization. Firstly using forward difference with θ-weighted scheme.

$$\frac{u^{n+1} - u^n}{\tau} = \theta[\alpha^2 u_{xx}^{n+1}] + (1-\theta)\alpha^2 u_{xx}^n. \tag{12}$$

Now further simplification of Eq. (12) leads to :

$$u^{n+1} - \tau\theta\alpha^2 u_{xx}^{n+1} = u^n + \tau(1-\theta)\alpha^2 u_{xx}^n. \tag{13}$$

Next we approximate the solution at $(n+1)$-time level by Lucas polynomials as :

$$u^{n+1} \cong \sum_{n=0}^{N} c_n^{n+1} L_n(x) = L(x)C, \tag{14}$$

where $L(x) = [L_0(x), L_1(x), ..., L_N(x)]$ and $C = [a_0, a_1, ..., a_n]^T$. We can obtain higher derivatives of u^{n+1} in terms of Lucas and Fibonacci polynomials by taking derivatives of Eq. (14) with respect to x as follows:

$$u_x^{n+1} \cong \sum_{n=0}^{N} c_n^{n+1} L'_n(x) = nF(x)C, \tag{15}$$

$$u_{xx}^{n+1} \cong \sum_{n=0}^{N} c_n^{n+1} L''_n(x) = n(F(x)D)C, \tag{16}$$

where $F(x) = [F_0(x), F_1(x), ..., F_N(x)]$ and $u^{n+1} = u(x, t^{n+1})$. The method is based on collocation approach so we use the following collocation points :

$$x_r = a + \frac{b-a}{N}(r-1), \quad r = 1, 2, ..., N+1, \quad a \leq x_r \leq b.$$

Using the boundary points in Eq. (14), we get :

$$\begin{aligned} u(a, t_{n+1}) &= \sum_{n=0}^{N} c_n^{n+1} L_n(a) = \psi_a(t_{n+1}), \\ u(b, t_{n+1}) &= \sum_{n=0}^{N} c_n^{n+1} L_n(b) = \psi_b(t_{n+1}), n = 0, 1, ..., M-1. \end{aligned} \tag{17}$$

Plugging Eqs. (14)-(16) into Eq. (13) and discretizing $x \to x_r$, a full discrete system of equations obtained as follows

$$\sum_{n=0}^{N} c_n^{n+1} L_n(x_r) - \tau\theta\alpha^2 (\sum_{n=0}^{N} c_n^{n+1} L_n''(x_r)) = u^n + \tau(1-\theta)\alpha^2 u_{xx}^n.$$

Also writing derivative expressions from Eq. (14) and Eq. (16) , we get above equation as :

$$[L(x_r) - \tau\theta\alpha^2 n(F(x_r D)]C = u^n + \tau(1-\theta)\alpha^2 u_{xx}^n. \tag{18}$$

So Eq. (18) in compact form can be written as

$$\Omega C = \Theta,$$

where $\Theta = u^n + \tau(1-\theta)\alpha^2 u^n_{xx}$ is a column vector that is calculated at every time step at x_r , Ω is $(N+1)\times(N+1)$ matrix given as $\Omega = L(x_r) - \tau\theta\alpha^2 n(F(x_r)D)$ and C is $(N+1)\times 1$ unknown coefficient vector. The Eq. (18) contains $(N+1)$ unknowns which needs to be computed. Once the unknown coefficients computed, then numerical solutions can be obtained from Eq. (14)

Test Problems

Problem 1.

$$\frac{\partial u}{\partial t} - \frac{1}{\pi^2}\frac{\partial^2 u}{\partial x^2} = 0, \quad 0 < x < 1, \quad 0 < t,$$

with initial condition:

$$u(x,0) = \cos\pi(x - \frac{1}{2}), \quad 0 \leq x \leq 1,$$

and homogeneous boundary condition:

$$u(0,t) = u(1,t) = 0, \quad 0 < t.$$

Comparing results at $t = 0.4$ to the actual solution $u(x,t) = e^{-t}\cos\pi(x - \frac{1}{2})$.

Discussion

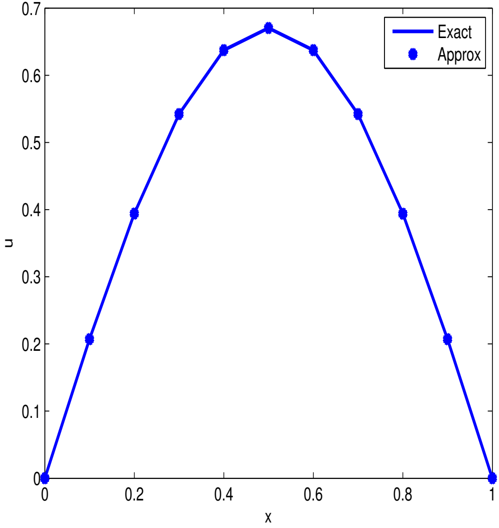
The problem 1 has been solved for different values of dt and the error norms L_2 and L_∞ have been addressed in Table 1 . From table 1 it is clear that when we decrease time step size, the error norms decreases. Table 2 shows the comparison of the exact, forward and computed results for $N = 10$. It is clear from table 2 that the computed results at different points are in good agreement with exact solutions as compared to forward difference results. The graphical behaviour of the solution has been addressed in Figure 1. One can see clearly that exact and approximate solutions promises well with each other. Figure 2 shows the absolute error of problem 1.

Table: Error norms of problem 1 for $N = 10$ and for different time step size

dt	L_∞	L_2
0.04	5.2439×10^{-3}	1.1725×10^{-2}
0.03	3.8946×10^{-3}	8.7087×10^{-3}
0.02	2.6514×10^{-3}	5.9288×10^{-3}
0.01	1.3338×10^{-3}	2.9827×10^{-3}

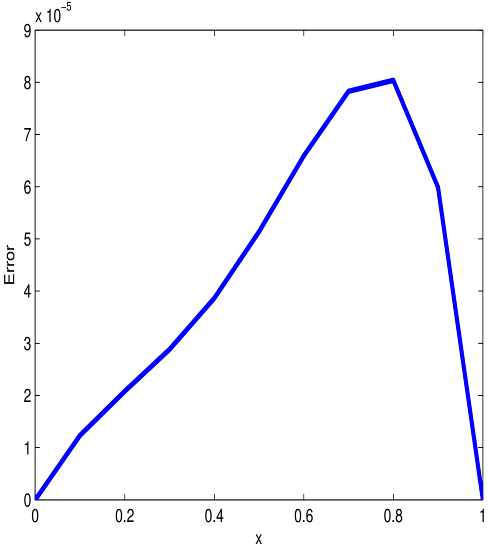
Table: Comparison of exact v/s approximate and forward difference

x	Approximate	Exact	Forward
0.1	0.207125527752174	0.207140285895210	0.206147397870097
0.2	0.393980990074312	0.394004237375760	0.392115652123291
0.3	0.542272072655695	0.542300308913029	0.539700894316262
0.4	0.637480421658031	0.637512247785462	0.634456452656315
0.5	0.670284855654593	0.670320046035639	0.667106992892330
0.6	0.637473920327625	0.637512247785462	0.634456452656315
0.7	0.542260453086130	0.542300308913029	0.539700894316262
0.8	0.393967324573462	0.394004237375760	0.392115652123291
0.9	0.207114957537685	0.207140285895210	0.206147397870097



chch

Figure: Graphical behaviour of problem 1



chch

Figure: Graph of absolute error

Problem 2.

$$\frac{\partial u}{\partial t}(x,t) - \frac{\partial^2 u}{\partial x^2}(x,t) = 0, \quad 0 < x < 1, \quad 0 \le t,$$

with initial condition:

$$u(x,0) = \sin(\pi x), \quad 0 \le x \le 1,$$

and homogeneous boundary condition:

$$u(0,t) = u(1,t) = 0, \quad 0 < t.$$

Comparing results at $t = 0.5$ to the actual solution $u(x,t) = e^{-\pi^2 t}\sin(\pi x)$.

Discussion

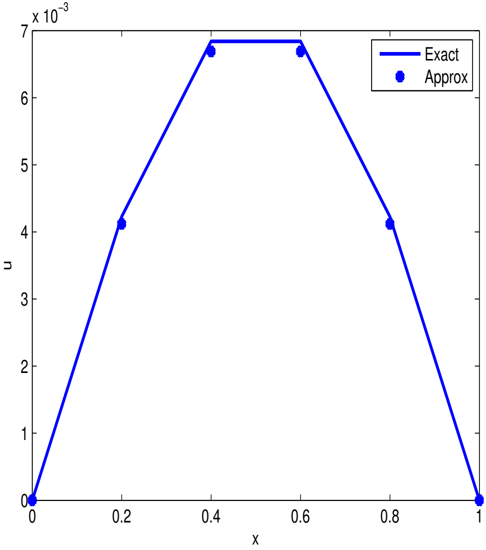
The problem 2 has been solved for different values of dt and the error norms L_2 and L_∞ have been addressed in Table 3 . From table 3 it is clear that when we decrease time step size, the error norms decreases. For $N = 5$ the comparison table of proposed scheme and forward difference scheme results with exact solution of this problem is given in table 4. It is clear from table 4 that the computed results at different points promises well with exact solutions as compared to forward difference results. The solution profile in 1D has been plotted in Figure 3. It is clear that exact and computed solutions are in good agreement. Figure 4 shows the absolute error of problem 2.

Table: Error norms of problem 2 for $N = 5$ and for different time step size

dt	L_∞	L_2
0.04	9.6852×10^{-3}	2.1656×10^{-2}
0.03	7.0092×10^{-3}	1.5673×10^{-2}
0.02	3.8766×10^{-3}	8.6684×10^{-3}
0.01	1.8471×10^{-3}	4.1303×10^{-3}

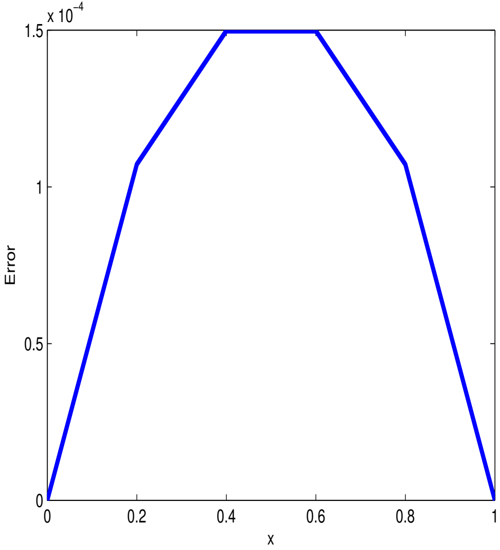
Table: Comparison of exact v/s approximate and forward difference

x	Approximate	Exact	Forward
0.2	0.004120097725974	0.004227282972762	0.004905474506864
0.4	0.006690107236734	0.006839887529993	0.007937224483052
0.6	0.006690107236352	0.006839887529993	0.007937224483052
0.8	0.004120097725243	0.004227282972762	0.004905474506864



chch

Figure: Graphical behaviour of problem 2



chch

Figure: Graph of absolute error

Problem 3.

$$\frac{\partial u}{\partial t}(x,t) - \frac{1}{2}x(1-x)\frac{\partial^2 u}{\partial x^2}(x,t) = 0, \quad 0 < x < 1, \quad 0 < t, \tag{19}$$

Here we considered a variable coefficient diffusion equation which is given in Eq. (19) along with initial condition:

$$u(x,0) = x(1-x), \quad 0 \leq x \leq 1, \tag{20}$$

and homogeneous boundary condition:

$$u(0,t) = u(1,t) = 0, \quad 0 < t. \tag{21}$$

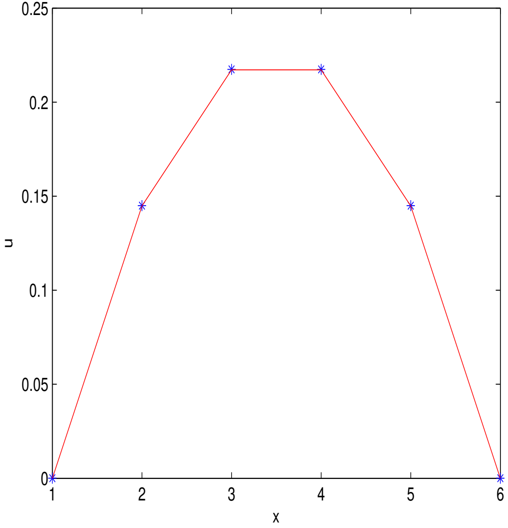
The theoretical solution of this problem is $u(x,t) = x(1-x)e^{-t}$.

Discussion

The numerical computation of this problem has been done at $T = 0.1$, $dt = 0.025$ and $N = 10$. In Table 5 we recorded the obtained results. From table it is obvious that the results of the proposed scheme are better than forward difference scheme.

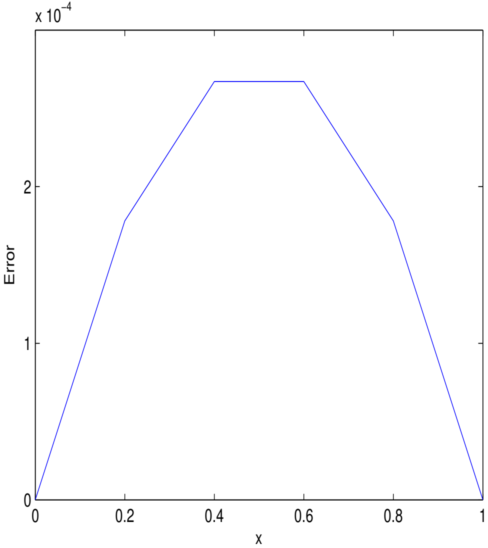
Table: Comparison of exact v/s approximate and forward difference

x	Approximate	Exact	Forward
0.1	0.0815	0.814	0.900
0.2	0.1450	0.1448	0.1600
0.3	0.1902	0.1900	0.2100
0.4	0.2174	0.2172	0.2400
0.5	0.2265	0.2262	0.2500
0.6	0.2174	0.2172	0.2400
0.7	0.1902	0.1900	0.2100
0.8	0.1450	0.1448	0.1600
0.9	0.0815	0.0814	0.1900



chch

Figure: Graphical behaviour of problem 3



chch

Figure: Graph of absolute error

Conclusion

In this work we discussed a hybridized method based on finite difference approach and Lucas polynomials. We solved three test problems and the results have been reported in tabulated data as well as in graphical form. From computed solutions we observed that the proposed method is suitable for the diffusion problem with variable and constant coefficients so one can apply this method to other test problems like diffusion.
In future we will try to extend proposed method to some non-linear PDE's and time-fractional PDE's. Also we will discuss the stability and convergence criteria of the proposed scheme.

Thank You

Questions?