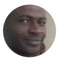
Finite-Difference Approximations to the Heat Equation via C

3 authors:

Olusegun Adeyemi Olaiju
Universiti Teknologi Malaysia

Yeak Su Hoe
Universiti Teknologi Malaysia



Ezekiel Babatunde Ogunbode
Federal University of Technology Minna

JASES
Journal of Applied Sciences & Environmental Sustainability

©Journal of Applied Sciences & Environmental Sustainability 3 (7): 188 - 200, 2017
e-ISSN 2360-8013

Research Article

Finite-Difference Approximations to the Heat Equation via C

Olusegun Adeyemi Olaiju[1], Yeak Su Hoe[1] and Ezekiel Babatunde Ogunbode[2,3]

[1]Faculty of Science, Department of Mathematics, Universiti Teknologi Malaysia.
81310, Skudai. Johor Bahru. Malaysia.
[2]Faculty of Civil Engineering, Department of Structures and Materials,
Universiti Teknologi Malaysia. 81310, Skudai. Johor Bahru. Malaysia.
[3]Department of Building, Federal University of Technology Minna. Niger State. Nigeria.

Corresponding Author: olaiju@yahoo.com

ARTICLE INFO

Article history
Received: 2/06/2017
Accepted: 04/07//2017

Boundary conditions, C program, Finite difference method, Heat equation, Partial differential equations.

A b s t r a c t

Partial differential equations (PDEs) are useful tools for mathematical modelling in the field of physics, engineering and Applied Mathematics. Useful as these equations are, only a few of them can be solved analytically. Numerical methods have been proven to perform exceedingly well in solving difficult partial differential equations. A popularly known numerical method known as finite difference method has been applied expansively for solving partial differential equations successfully. In this study, explicit finite difference scheme is established and applied to a simple problem of one-dimensional heat equation by means of C. These sample calculations show that the accuracy of the predictions depends on mesh spacing and time step. The result of the study reveals that the solutions of the heat equation decay from an initial state to a non-varying fixed state circumstance, the temporary performance of these solutions are smooth and bounded, the solution does not improve local or global utmost that are outside the range of the initial data.

1. Introduction

According to Louise (2015), PDEs classification is important for any numerical solution chosen. The general equation governing partial differential equations is of the form:

$$a(x,y)u_{xx} + 2b(x,y)u_{xy} + c(x,y)u_{yy} = f\left(x,y,u_x,u_y,u\right) \quad (1)$$

PDE are classified into three categories, which are;

i. Elliptic, where $b^2 < ac$, e.g Laplace's equation; $u_{xx} = u_{yy} = 0$, $a = c = 1, b = 0$

©Journal of Applied Sciences & Environmental Sustainability 3 (7): 188 - 200, 2017
e-ISSN 2360-8013

ii. Hyperbolic, where $b^2 > ac$, e.g 1D wave equation; $u_{xx} - \frac{1}{c^2} u_{yy} = 0$,

$a = 1,\ c = -\frac{1}{c^2},\ b = 0$

iii. Parabolic, where $b^2 = ac$, e.g Diffusion equation; $u_t - \alpha u_{xx} = 0$, $a = \alpha,\ c = b = 0$

The one-dimensional heat equation is a parabolic PDE and is of the form

$$Ut = \alpha Uxx \tag{2}$$

where $U(x\ t)$ is the dependent variable, and α is a constant coefficient called the thermal diffusivity which is the material property.

Equation (2) is a model of transient heat conduction in a lump of solid with thickness L. The domain of the solution is a semi-infinite strip of width L that continues indefinitely in time. In a practical computation, the solution is obtained only for a finite time. Solution to equation (2) requires specification of boundary conditions at x = 0, $U(0,t) = g_1(t)$ (Dirichlet boundary conditions) and x = L, $U_x(L,t) = g_2(t)$ (Neumann boundary conditions) and initial conditions at t = 0. , $U(x,0) = f(x)$

As mentioned by Hadamard, a problem is well-posed (or correctly-set) if satisfies the succeeding circumstances;

a. it has a solution,

b. the solution is unique,

c. the solution's behaviour changes continuously with the initial conditions.

So, the heat equation is well-posed (Louise, 2015; Lloyd, 1996).

The finite difference method is one of the various techniques for finding numerical solutions to Partial differential equations. In all numerical solutions, the continuous partial differential equation is substituted with a discrete approximation, which means the numerical solution is known only at a finite number of points in the physical domain which can be selected by the user of the numerical method. In general, increasing the number of points will equally increase the resolution as well as the accuracy of the numerical solution (Gerald, 2011).

The discrete approximation outcomes in a set of algebraic equations that are elucidated for the values of the discrete unknowns. Figure 1 is an illustrative depiction of the numerical solution. The grid is the set of points where the discrete solution is computed which are called nodes. Two basic parameters of the grid are Δx, the local distance amongst contiguous points in space, and Δt, the local distance amid adjacent time steps.

The basic idea of the finite-difference method is to replace continuous derivatives with so-called difference formulas that involve only the discrete values associated with positions on the grid.

Relating the finite-difference method to a differential equation involves replacing all derivatives with difference formulas. In the heat equation, there are derivatives with respect to time and derivatives concerning space. Using various arrangements of mesh points in the difference formula results in difference schemes. In the limit as the mesh spacing (Δx and Δt) go to zero, the numerical solution obtained with any valuable system will approach the true solution to the original differential equation. Though, the rate at which the numerical solution approaches the true solution varies with the system. Several academic writtings have been published on numerical solution of heat equation (william, 1992; Morton and Mayers, 1994; Jeffery, 1998; Clive, 1988; Golub and Ortega, 1993; Burden and Faires, 1997; Thomas, 2013; Strikwerda, 2004 and Hoffman, 1992).

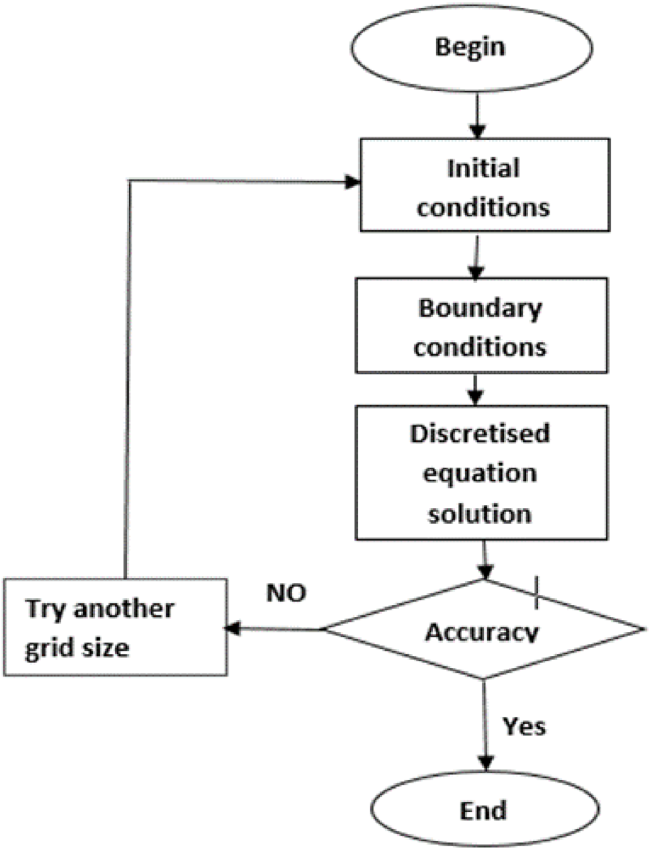


Figure 1: Flow chart for the solution of the heat equation

©Journal of Applied Sciences & Environmental Sustainability 3 (7): 188 - 200, 2017
e-ISSN 2360-8013

Finite difference method

By considering Figure 2, the white squares indicate the location of the initial values which are already known. The grey squares indicate the location of the boundary values which are also known. The black circles indicate the position of the interior points where the finite difference approximation is to be computed.

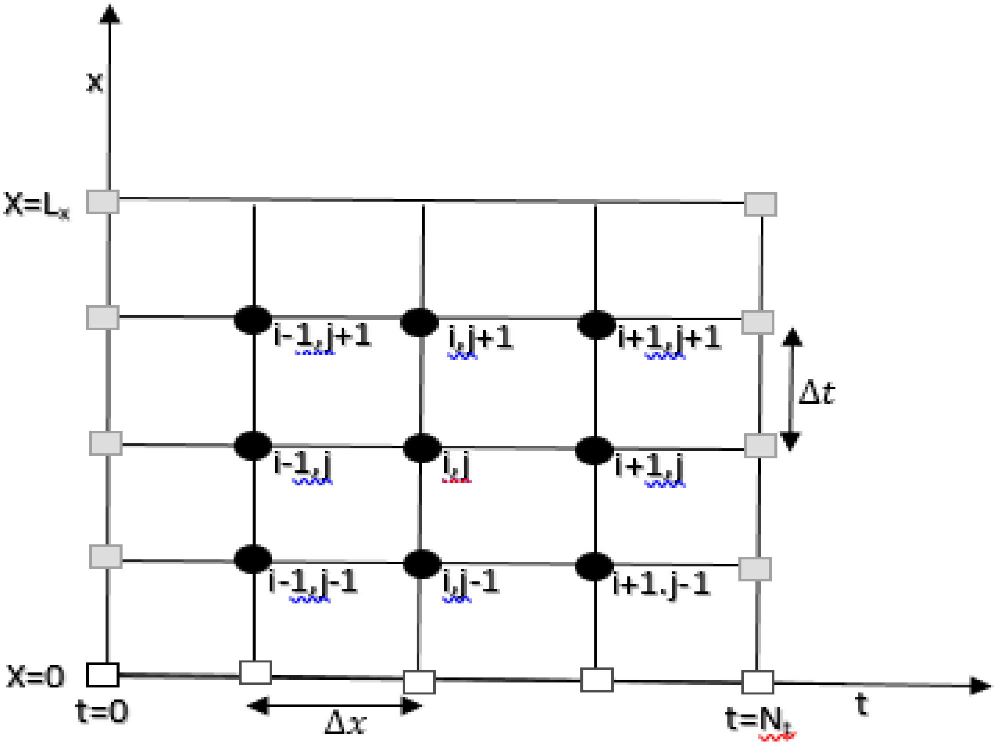


Figure 2: Discrete Grid Points

Consider Taylor series expansion of U_i^{j+1} about the point (i, j) in Figure 2

$$U_i^{j+1} = U_i^j + \Delta t \frac{\partial U_i^j}{\partial t} + \frac{(\Delta t)^2}{2} \frac{\partial^2 U_i^j}{\partial t^2} + \frac{(\Delta t)^3}{6} \frac{\partial^3 U_i^j}{\partial t^3} + \cdots \quad (3)$$

Suppose $O(\Delta t)$ terms is considered in equation (3) then the forward difference in time approximation for U_t will be arrived at,

$$\frac{\partial U_i^j}{\partial t} = \frac{U_i^{j+1} - U_i^j}{\Delta t} + O(\Delta t) \quad (4)$$

©Journal of Applied Sciences & Environmental Sustainability 3 (7): 188 - 200, 2017
e-ISSN 2360-8013

A higher order approximation for U_t can be derived if the Taylor series expansion for $U_i^{j-1}$is equally considered:

$$U_i^{j-1} = U_i^j - \Delta t \frac{\partial U_i^j}{\partial t} + \frac{(\Delta t)^2}{2}\frac{\partial^2 U_i^j}{\partial t^2} - \frac{(\Delta t)^3}{6}\frac{\partial^3 U_i^j}{\partial t^3} + \cdots \qquad (5)$$

By subtracting equation 5 from 3, we have the centred difference equation in time, which always gives higher order accuracy than the forward difference:

$$\frac{\partial U_i^j}{\partial t} = \frac{U_i^{j+1} - U_i^{j-1}}{2\Delta t} + O(\Delta t^2) \qquad (6)$$

Similarly, the approximation for the second order derivative Utt can be derived by the addition of equations 3 and 5:

$$\frac{\partial^2 U_i^j}{\partial t^2} = \frac{U_i^{j+1} - 2U_i^j + U_i^{j-1}}{(\Delta t)^2} + O(\Delta t^2) \qquad (7)$$

The same approximations apply to spatial variable x

The above approximations are used by the finite difference method to solve partial differential equations numerically.

Solution of 1D heat equation

Consider the heat equation (2), $Ut = \alpha Uxx$ for $0 \le x \le Lx,\ 0 \le t \le Lt$ and discretise time and variable x relating to space.

Let $\Delta t = \frac{Lt}{Nt},\ \Delta x = \frac{Lx}{Nx}\quad t_j = j\Delta t,\ 0 \le j \le Nt$ and $x_i = i\Delta x,\ 0 \le i \le Nx$.where Lt and Lx are the length of t and x respectively and $Nx\ and\ Nt$ are number of, the grids on both x and t axis.

If $U_i^j = U(x_i, t_j)$, then, equation 2 has the following finite difference approximation from equation (4) and (7) and by dropping $O(\Delta t)$ and $O(\Delta x^2)$

$$\frac{\partial U(x_i, t_j)}{\partial t} \approx \frac{U_i^{j+1} - U_i^j}{\Delta t}$$

and

$$\frac{\partial^2 U(x_i, t_j)}{\partial x^2} \approx \frac{U_{i+1}^j + U_{i-1}^j - 2U_i^j}{(\Delta x)^2}$$

©Journal of Applied Sciences & Environmental Sustainability 3 (7): 188 - 200, 2017
e-ISSN 2360-8013

So that the discretised version of equation 2 is,

$$\frac{U_i^{j+1} - U_i^j}{\Delta t} = \frac{c}{(\Delta x)^2}\left(U_{i+1}^j - 2U_i^j + U_{i-1}^j\right)$$

Which can be rewritten as,

$$U_i^{j+1} = k\left(U_{i+1}^j + U_{i-1}^j\right) + (1 - 2k)U_i^j \tag{8}$$

where $k = \frac{\alpha \Delta t}{(\Delta x)^2}$.

Thus, U_i^{j+1} gives the solution for the temperature at the next time step.

Assume there exist initial conditions,

$$U(x,0) = U_i^0 = f(x_i) \tag{9}$$

and mixed boundary conditions,

(Dirichlet boundary conditions) for $x = 0$, $U(0,t) = U_0^j = g_1(t_j)$ (10)

(Neumann boundary conditions)for $x = Lx$, $U_x(Lx,t) = \frac{\partial U_{Nx}^j}{\partial x} = g_2(t_j)$ (11)

Then, the solution to equation 2, with initial conditions (9) and boundary conditions (10) and (11) takes the following steps,

From equation (11) we have,

©Journal of Applied Sciences & Environmental Sustainability 3 (7): 188 - 200, 2017
e-ISSN 2360-8013

$$\frac{\partial U_{Nx}^{j}}{\partial x} \approx \frac{U_{Nx}^{j} - U_{Nx-1}^{j}}{\Delta x} = g_2(t_j)$$

which gives,

$$U_{Nx}^{j} = \Delta x g_2(t_j) + U_{Nx-1}^{j} \qquad (12)$$

Combining equations (8),(9),(10) and (12) gives,

$$\begin{pmatrix} U_1^{j+1} \\ U_2^{j+1} \\ U_3^{j+1} \\ \vdots \\ U_{Nx-1}^{j+1} \end{pmatrix} = \begin{pmatrix} 1-2k & k & 0 & 0 & \cdot & \cdot & 0 & 0 \\ k & 1-2k & k & 0 & \vdots & \vdots & 0 & 0 \\ 0 & k & 1-2k & k & . & . & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & , & , & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & k & 1-2k \end{pmatrix} \begin{pmatrix} U_1^{j} \\ U_2^{j} \\ U_3^{j} \\ \vdots \\ U_{Nx-1}^{j} \end{pmatrix} + \begin{pmatrix} kU_0^{j} \\ 0 \\ 0 \\ \vdots \\ kU_{Nx-1}^{j} \end{pmatrix} \qquad (13)$$

$$\begin{pmatrix} U_1^{j+1} \\ U_2^{j+1} \\ U_3^{j+1} \\ \vdots \\ U_{Nx-1}^{j+1} \end{pmatrix} = \begin{pmatrix} 1-2k & k & 0 & 0 & \cdot & \cdot & 0 & 0 \\ k & 1-2k & k & 0 & \vdots & \vdots & 0 & 0 \\ 0 & k & 1-2k & k & . & . & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & , & , & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & k & 1-k \end{pmatrix} \begin{pmatrix} U_1^{j} \\ U_2^{j} \\ U_3^{j} \\ \vdots \\ U_{Nx-1}^{j} \end{pmatrix} + \begin{pmatrix} kg_1(t_j) \\ 0 \\ 0 \\ \vdots \\ k\Delta x g_2(t_j) \end{pmatrix} \qquad (14)$$

Equation (14) can be written in the form:

$$U^{j+1} = AU^{j} + B \qquad (15)$$

Stability of the Numerical Methods

The solutions to Equation (2) subject to the initial and boundary conditions in Equations (9), (10) and (11) are all bounded, decaying functions. Thus the magnitude of the solution will decrease from the initial condition to a constant.

©Journal of Applied Sciences & Environmental Sustainability 3 (7): 188 - 200, 2017
e-ISSN 2360-8013

Explicit finite difference method is only stable if k (the gain parameter) satisfies $k = \frac{\alpha \Delta t}{\Delta x^2} < \frac{1}{2}$ or the time step satisfies: $\Delta t < \Delta x^2/2\alpha$. If the time step exceeds this value, this can yield unstable solutions that oscillate and grow. (Gerald, 2011; Arnold, 2015).

C program

C is categorised as the high-level and general-purpose programming language which is appropriate for the development of portable applications. C is originally intended for writing system software (techopedia.com). Additionally, Techopedia.com also describe C as one of the most extensively used languages in programming. C language has a compiler for most computer systems, and it has generated many popularly known languages such as C++. Consequently, C has been accepted as an influential programming language which belongs to the structured, procedural paradigms of languages. It has been shown to be flexible and may be used for diverse applications. However, despite C been a high-level language, it has been seen to share several characteristics with assembly language (Greg, 2014)

Test Problem

The finite difference codes are verified by solving the heat equation $Ut = \alpha Uxx$ using C codes with boundary conditions$U(x,0) = x(1-x)$,$U(0,t) = 0$,$U(L,t) = 0$. , and initial condition$U(x,0) = x(1-x)$. The exact solution to this problem is, $U(x,t) = 8\sum_{n=1}^{\infty} \frac{1}{(n\pi)^3} \sin(n\pi x) e^{-(n\pi)^2 t}$

Setting $Lt = 0.03$, $Nt = 5$ $\Delta t = \frac{Lt}{Nt} = 0.006$, $Lx = 1$, $Nx = 5$, $\Delta x = \frac{Lx}{Nx} = 0.2$ $t_j = j\Delta t$, $0 \le j \le Nt$ and $x_i = i\Delta x$, $0 \le i \le Nx$, $\alpha = 1$, the following tables are generated which give the Numerical and exact solutions of the problem together with the errors generated by the numerical solution.

Table1: The values of $u[i][j]$ at (i,j) for $Nx = Nt = 5$

u_i^j	0.000 0.160 0.240 0.240 0.160 0.000
	0.000 0.148 0.228 0.228 0.148 0.000
	0.000 0.138 0.216 0.216 0.138 0.000
	0.000 0.129 0.204 0.204 0.129 0.000
	0.000 0.121 0.193 0.193 0.121 0.000
	0.000 0.114 0.182 0.182 0.114 0.000

Table2: The values of exact $u(x,t)$ at (I,j) for $Nx = Nt = 5$

$u(x,t)$	0.000 0.152 0.245 0.245 0.152 0.000

©Journal of Applied Sciences & Environmental Sustainability 3 (7): 188 - 200, 2017
e-ISSN 2360-8013

	0.000 0.143 0.231 0.231 0.143 0.000
	0.000 0.135 0.218 0.218 0.135 0.000
	0.000 0.127 0.205 0.205 0.127 0.000
	0.000 0.120 0.194 0.194 0.120 0.000
	0.000 0.113 0.182 0.183 0.113 0.000

Table3: The error in $u[i][j]$ at (i,j) for $Nx = Nt = 5$

$Errors\ u_i^j$	0.000 0.008 -0.005 -0.005 0.008 -0.000
	0.000 0.005 -0.003 -0.003 0.005 -0.000
	0.000 0.003 -0.002 -0.002 0.003 -0.000
	0.000 0.002 -0.001 -0.001 0.002 -0.000
	0.000 0.001 -0.001 -0.001 0.001 -0.000
	0.000 0.001 -0.000 -0.000 0.001 -0.000

Table 1 shows the Numerical Solutions to the Problem of one-dimensional heat equation. Table two shows the exact solution of the problem while Table 3 gives the error which is the difference between the exact solution and numerical solution. By comparing Tables 1 and 2, it was observed that the values are very close, which shows consistency (Arnold, 2015).

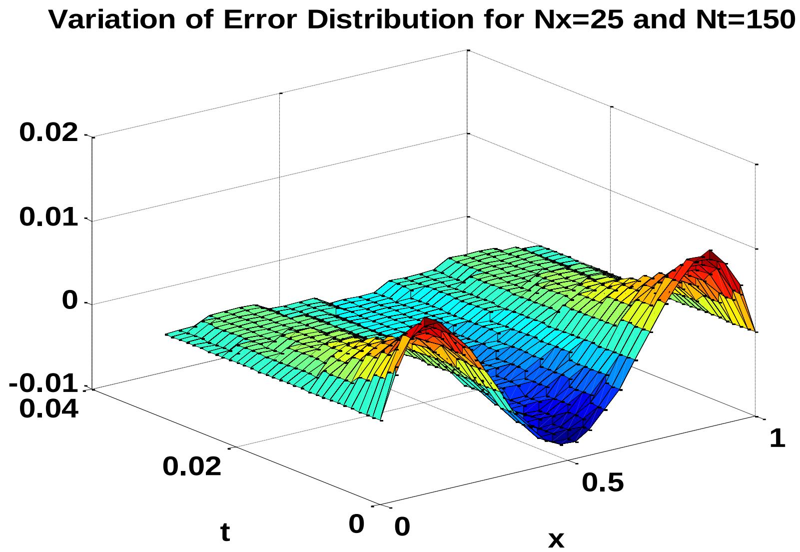


Figure 3: Explicit Finite Difference method error distribution with time for $Nx = 25, Nt = 150$

©Journal of Applied Sciences & Environmental Sustainability 3 (7): 188 - 200, 2017
e-ISSN 2360-8013

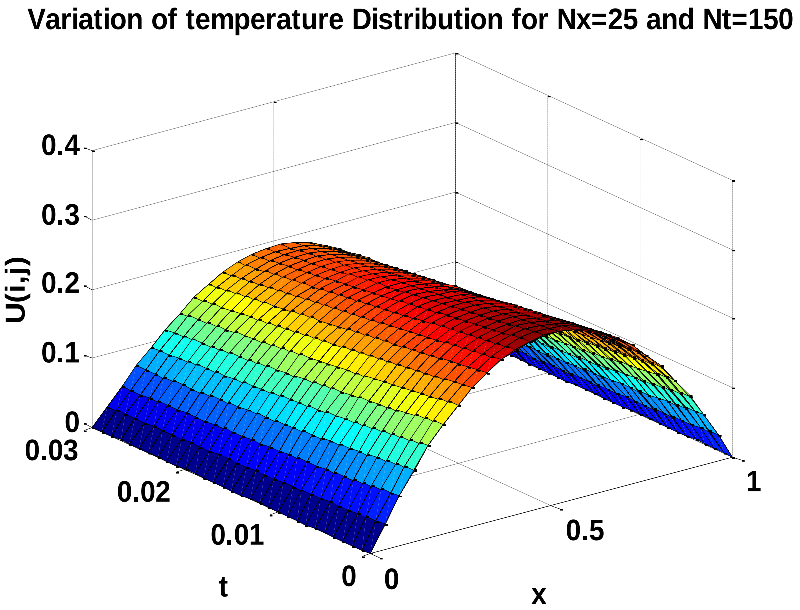


Figure 4: Explicit Finite Difference method for temperature distribution with time for Nx=25, Nt=150

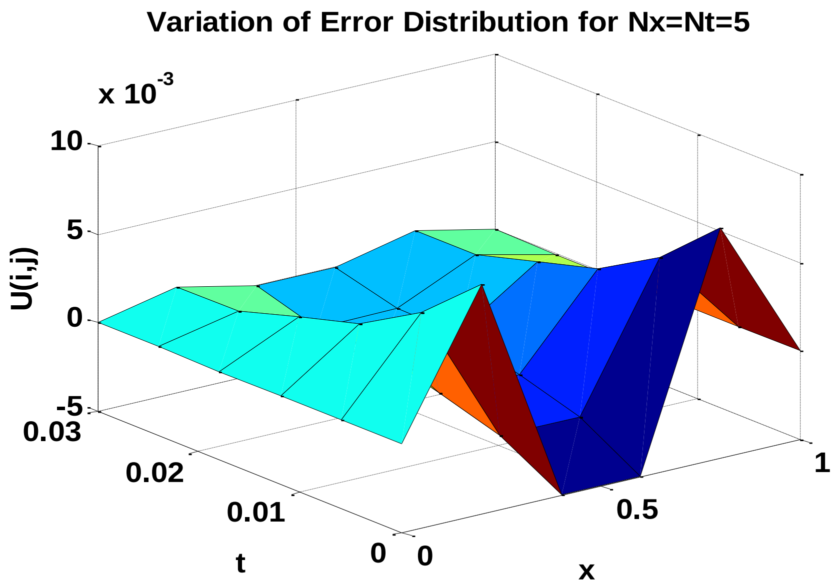


Figure 5: Explicit Finite Difference method error distribution with time for Nx= Nt=5

©Journal of Applied Sciences & Environmental Sustainability 3 (7): 188 - 200, 2017
e-ISSN 2360-8013

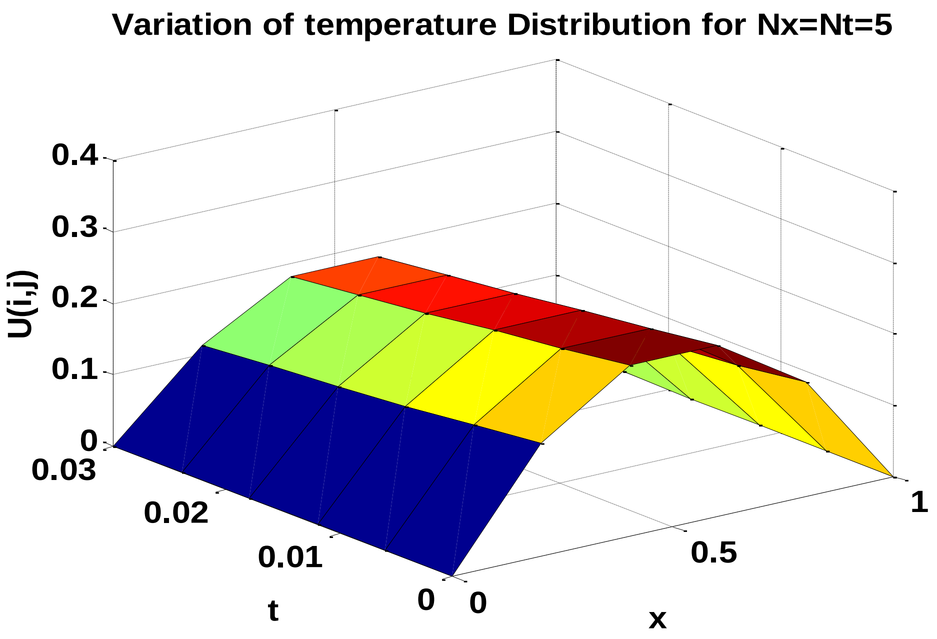


Figure 4: Explicit Finite Difference method for temperature distribution with time for Nx=Nt=5

Figure3 and 5 show the error in temperature on the implementation of finite difference method to 1D heat distribution problem for Nx=25, Nt =150 and Nx=Nt=5. From the above figures, It is clear that the errors become smaller with the increasing number of grids. Since the quality of solution improves with increasing number of elements. We conclude that the result is valid. (Subramanian, 2009).

Conclusion

So far we have used the finite difference method as a solution of one-dimensional heat equation. The explicit method has been used out of the different finite difference methods. The results were compared with the exact solution of the problem. We got the approximate solution by the method using C program, specifically Code:: Block software from www.codeblocks.org and Matlab from https://www.mathworks.com to generate the surface plots.

The research has revealed that the size of the mesh is significant to arrive at an accurate solution when using finite difference method, the smaller the size of the mesh the closer is the numerical result to the exact solution. Also, C program proved to be a powerful tool in programming the solution of one-dimensional heat partial differential equation. It was also observed that the solutions of the heat equation decay from an initial state to a non-varying steady state condition. The transient behaviour of these solutions are smooth and bounded; the solution does not develop

©Journal of Applied Sciences & Environmental Sustainability 3 (7): 188 - 200, 2017
e-ISSN 2360-8013

local or global maxima that are outside the range of the initial data. However, the study is limited to using explicit finite difference method on parabolic PDE only. It should be noted that finite element method and finite volume method are powerful tools to solve difficult partial differential equations.

REFERENCES

Arnold Douglas N. (2015). Lecture notes on Numerical Analysis of Partial Differential Equations. Available at http://www.math.umn.edu/~arnold/8445/notes.pdf

Burden R. L and Faires J. D (1997). Numerical Analysis. Brooks/Cole Publishing Co., New York, sixth edition.

Clive A.J. F (1988). Computational Techniquess for Fluid Dynamics. Springer-Verlag Berlin.

Gerald W. Recktenwald (2011). Finite-Difference Approximations to the Heat Equation www.nada.kth.se/~jjalap/numme/FDheat.pdf

Golub Gene and Ortega James M (1993). Scientific Computing: An Introduction with Parallel Computing. Academic Press, Inc., Boston.

Greg Perry and Dean Miller (2014). C Programming Absolute Beginner's Guide.Third Edition. Pearson Education, Inc.

Jeffery Cooper (1998). Introduction to Partial Differential Equations with Matlab. Birkhauser, Boston.

Lloyd N. Trefethen, (1996). Finite Difference and Spectral Methods for Ordinary and Partial Differential Equations, unpublished text, 1996, available at https://people.maths.ox.ac.uk/trefethen/pdetext.html

Louise Olsen-Kettle (2015), Numerical solution of partial differential Equations retrieved from http://espace.library.uq.edu.au/view/UQ:239427.

Morton K.W. and Mayers D.F(1994) Numerical Solution of Partial Differential Equations: An Introduction. Cambridge University Press, Cambridge, England.

Strikwerda J. C. (2004) Finite difference schemes and partial differential equations, SIAM.

©Journal of Applied Sciences & Environmental Sustainability 3 (7): 188 - 200, 2017
e-ISSN 2360-8013

Subramanian. S. J. (2009). Introduction to Finite Element Method. *Department of Engineering Design*. Indian Institute of Technology. Madras.

Techopedia. C Programming Language (C). Sourced online on 10th Dec, 2016 and available at https://www.techopedia.com/definition/24068/c-programming-language-c

Thomas, James W. (2013) Numerical partial differential equations: finite difference methods. Vol. 22. Springer Science & Business Media.

William F. (1992) Numerical Methods for Partial Differential Equations. Academic Press, Inc., Boston, third edition.