



IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY



VOLUME : 4 ISSUE : 07 Print / Issue Publication Date: 07-Jan-2020



ISSN : 2455-2143



DOI : 10.33564/IJEAST.2019.v04i07.051

Indexed In



WWW.IJEAST.COM

editor@ijeast.com



NUMERICAL ANALYSIS OF 1D LINEAR CONVECTION EQUATION USING UPWIND SCHEME IN MATLAB

Joy Mandal

Department of Mechanical Engineering
Jadavpur University, Kolkata, West Bengal, India

Abstract- The overall purpose of this study is to critically analyze one of the important areas of Computational Fluid Dynamics i.e. the Linear Convection Problem. There had been, quite a significant research in this field. The main issue that is faced while numerically solving this problem whether in 1D, 2D or 3D is the oscillation or diffusion which eventually makes the solution unstable. Hence, my study will deal with this in details and I had also incorporated the effect of mesh size and time step size on the solution and how they affect the stability individually. I had opted for Upwind Scheme to numerically solve this problem and to comment on the factor (CFL No.) which affects the stability. I had also derived the optimum condition for the CFL No., and the simulation results of the proposed scheme are also shown in this paper.

Keywords- Linear Convection, oscillation, mesh size, time step size, Upwind Scheme, CFL No.

I. INTRODUCTION

The equation given below is the most accessible equation in CFD; from the Navier Stokes equation we kept only the accumulation and convection terms for the x component of the velocity - as we already know, in CFD the variables to be computed are velocities; to make things even simpler, the coefficient of the first derivative of the velocity is constant, making the equation linear.

$$\frac{\partial u}{\partial \tau} + c \frac{\partial u}{\partial x} = 0$$

This equation represents the propagation of a wave without change of shape; the speed of the wave is c and with the initial condition given by:

$$u(x, 0) = u_o(x)$$

The exact solution is:

$$u(x, t) = u_o(x) - u_o(ct)$$

This means that on an axis with the positive direction to the right, the wave will move from left to right with the speed c.

To discretize this equation in space and time we will use the Forward Difference scheme for time derivative and Backward Difference scheme for space derivative. The partial derivative can be approximated using the derivative definition, removing the limit operator:

$$\frac{\partial u}{\partial x} \approx \frac{u(x+\delta x) - u(x)}{\delta x}$$

This approach will be applied both for space and time; for time steps we will use the superscript n and for space steps (grid cells) we will use the subscript i.

$$\frac{\partial u}{\partial \tau} + c \frac{\partial u}{\partial x} = 0$$

Now applying forward difference scheme for time derivative and backward difference scheme for space derivative.

$$\frac{u_i^{n+1} - u_i^n}{\delta \tau} + c \frac{u_i^n - u_{i-1}^n}{\delta x} = 0$$

$$\frac{u_i^{n+1} - u_i^n}{\delta \tau} = -c \frac{u_i^n - u_{i-1}^n}{\delta x}$$

$$u_i^{n+1} = u_i^n - \frac{c\delta \tau}{\delta x} (u_i^n - u_{i-1}^n)$$

where n+1 and n are two consecutive steps in time and i and i-1 are two neighbouring points of the discretized x coordinate (grid points). The only unknown in the equation written in this way is u_i^{n+1} . We must solve for this unknown to get an equation that allows us to advance in time.

So, firstly I had run the simulation for 1D linear convection equation with different grid points only. Then I had also changed the time step size value to see the effect on the stability of the numerical scheme.

Below will be results of MATLAB simulation for the problem stated above along with the plots, discussions and at last the conclusion.

A. PROBLEM STATEMENT

1. Assume that the domain length is $L = 1\text{m}$.
2. The initial velocity profile is a step function. It is equal to 2m/s between $x = 0.1$ and 0.3 and 1m/s everywhere else.
3. Use Upwind Scheme for discretizing the convection equation.
4. Time step and number of grid points should be variables in the calculation with $C=1$.

Part I

1. Set time step = 0.01 .

Part II

Write a function that accepts time step as an argument and solves the problem.

1. Set $n = 80$.
2. Time steps to use = $1\text{e-}4$, $1\text{e-}3$, $1\text{e-}2$, and $1\text{e-}1$.

Expected outputs

1. For $C = 1$ and for time = 0.4 seconds compare the original and final velocity profiles and make the comparison for $n=20, 40, 80$ and 160 .
2. In a single graph compare the numerical solution at $t = 0.4$ seconds and compare the effect of time step on the numerical solution.
3. Plot the simulation time as a function of the time step and explain each of the plots in details.

B. NUMERICAL SCHEME

The Upwind Differencing is a one-sided differencing scheme used to numerically discretize the hyperbolic partial differential equation and the side of the differencing is determined by the direction of the flow of information. It takes into account the "flow of information" from the characteristics of the system of equations. If we consider the linear convection or the linear wave equation.

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0 \quad \text{where } c > 0$$

J.D. Anderson et al.(1992) in his book had stated what method of characteristics is and how the information flows in a system. Since, information flows from left to right (positive direction), upwind scheme in space consists of backward difference formula and since, this is a hyperbolic P.D.E therefore, the time derivative is evaluated by marching forward in time (time marching method) and so, the forward differencing formula for time.

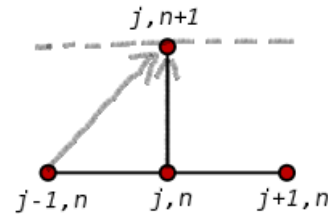


Figure 1: Stencil for Upwind Scheme

Stencil helps us to visualize how the information of the flow properties (velocity, temperature and other parameters) flow through the grid points for different numerical schemes. As far this scheme is considered, we know that it's a time marching problem, so the information gets transferred from time step n to time step $n+1$. Here index j represents the special grid points. If we look at this equation-

$$u_i^{n+1} = u_i^n - \frac{c \delta t}{\delta x} (u_i^n - u_{i-1}^n)$$

we can clearly see that the only unknown is u_i^{n+1} , to evaluate this unknown, information in space travels from previous space grid point at time step n to current space grid at time step $n+1$.

Once the unknown variable is evaluated at all grid points at time step $n+1$, the solver, with the help of these new values calculates the unknown variable at time step $n+2$ and so on and so forth till a steady state condition is reached and we have a convergence. This is how the solver works. I had coded this problem using MATLAB software and I had elaborately presented all my findings below.

II. PLOTS AND DISCUSSIONS

Part I

We will be looking at the convective velocity plot and we will see how the wave propagates through the domain. To visualize the accuracy of the solution, the numerical velocity profile is

being compared with the exact velocity profile for different number of grid points. There are multiple plots to examine

how an increase or decrease in the number of grid points deviates the velocity profile from the exact one.

For number of grid points = 20

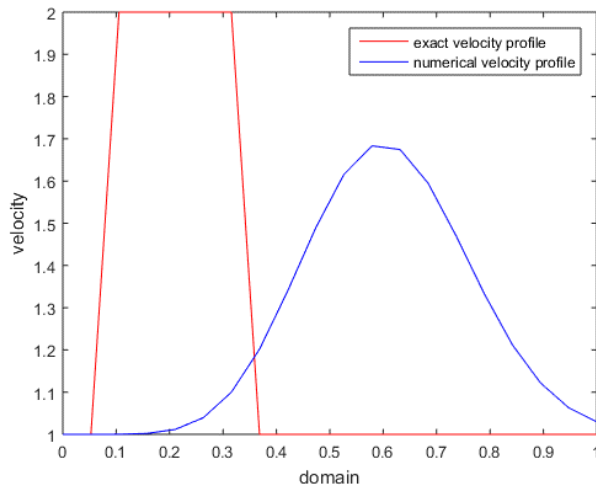


Fig. 2: with hold off command

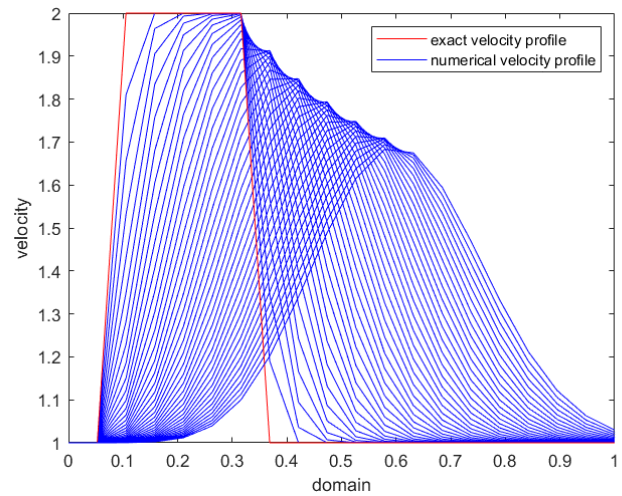


Fig 3: without hold off command

For number of grid points=40

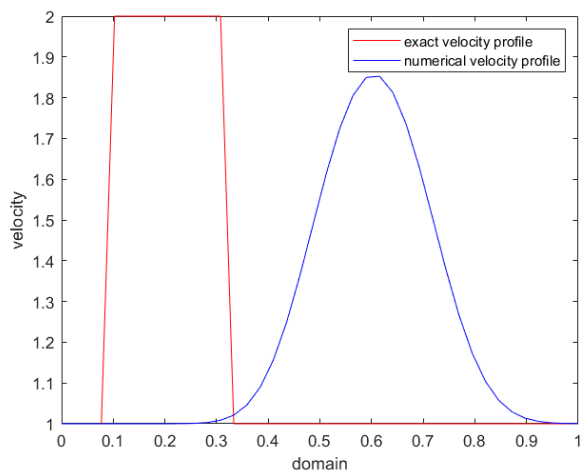


Fig. 4: with hold off command

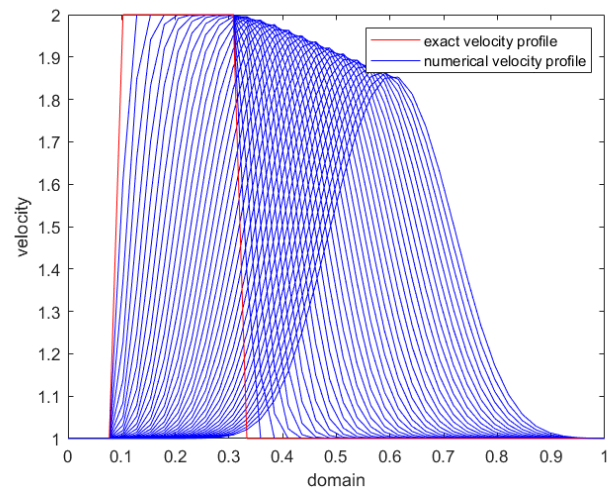


Fig. 5: without hold off command

For number of grid points=80

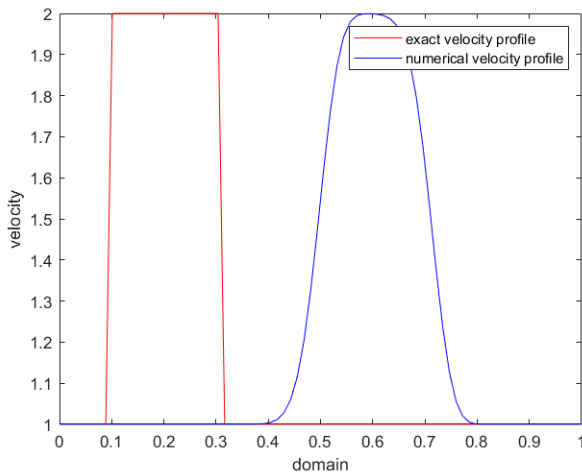


Fig. 6: with hold off command

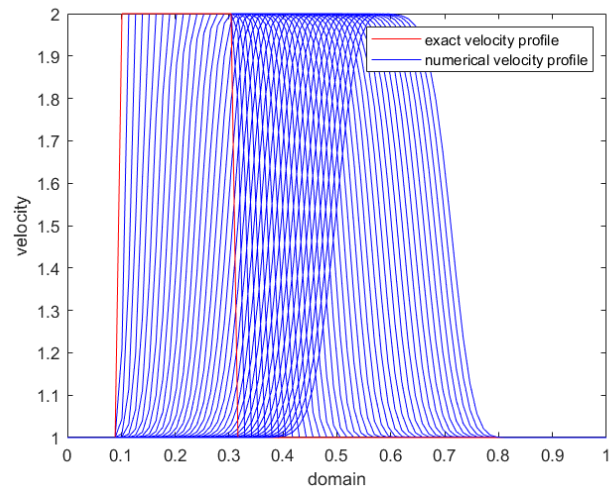


Fig. 7: without hold off command

For number of grid points=160

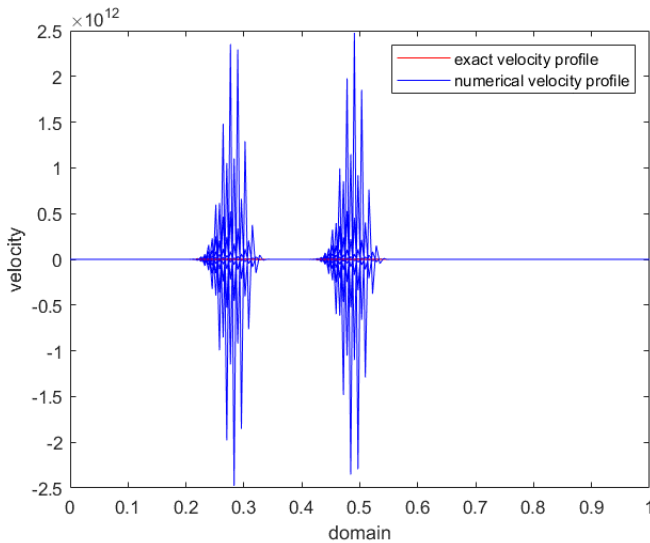


Figure 8: with hold off command

We can see that the square wave is not maintained in the numerical method of solving the equation. This is because, the first order backward differencing scheme in space creates false diffusion. This is also the reason why the peak is not maintained in the numerical profile and it is eventually becoming smooth and diffused. However, if we take a closer look at the plots for different values of space points, we can say that if the spatial step is reduced, the error reduces and the peak is maintained, see the plot for figure 7 and figure 6.

But that will mean that we can get the best solution with max. no. of 'n'. That is, however, not the case, because the next plot shows that when $n=160$ grid points we get an unstable solution. There is something in our numerical approach which is killing the solution. Therefore, we need to analyse our numerical approach for stability. The term $c(dt/dx)$ plays a vital role in determining the stability of a numerical approach.

Now, we will be looking at the numerical velocity profile for different time steps and I am also going to make a comment on the Courant-Friedrichs-Lewy or CFL condition on the stability of linear convection problems. I had also calculated the time taken by the solver to complete the simulation for each time step and then plotted them for inference.

Part II

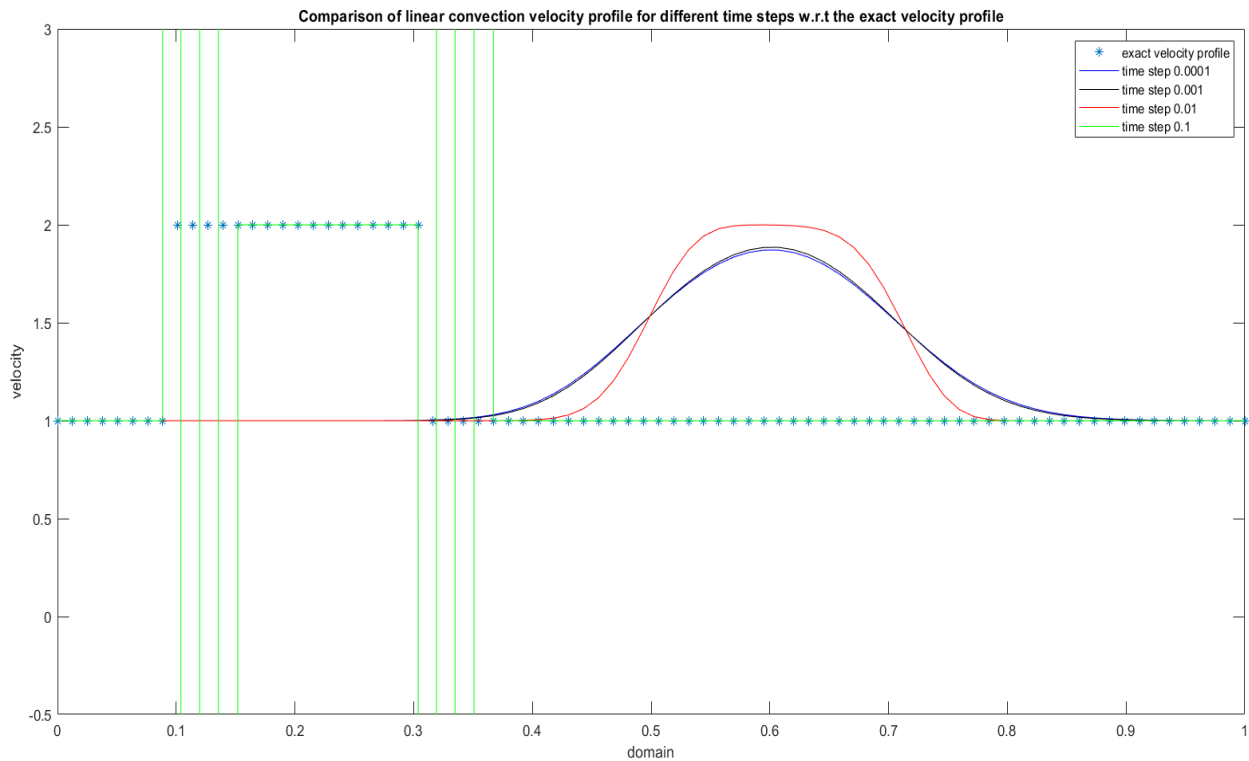


Fig. 9: Velocity profiles for different time steps

The velocity profile depicted by '*' is the actual velocity profile.

We can see that as we make the time step size bigger and bigger, we are getting better solutions. When the time step size is 0.01, the velocity profile kind of matches the actual profile, the peak is also maintained somewhat. Therefore, we might

think that, okay, let's make it even bigger but boom our solution gets killed if we implement the same in our coding. As we can see that when the step size is 0.1, the solution becomes unstable due to diffusion. Initially, we can see that the solution is exactly a square but after that there are oscillations which make the solution unstable.

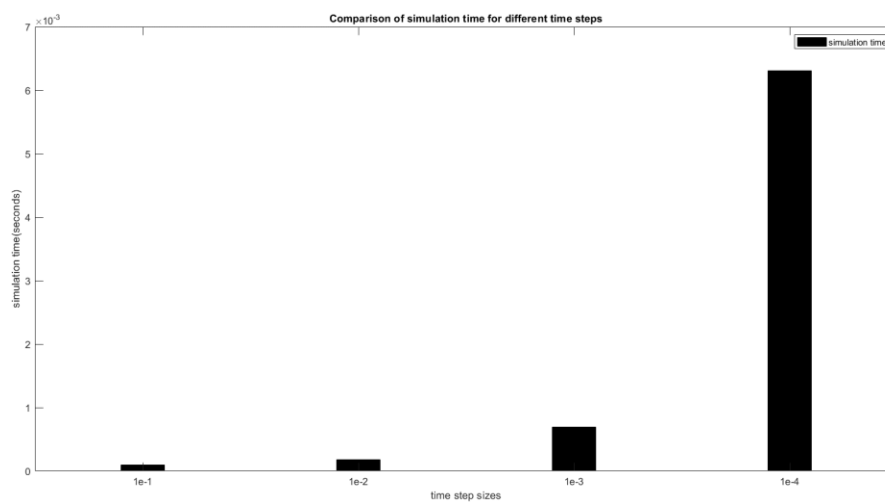


Fig. 10: Simulation time versus time step size

This plot is very simple to understand. As we are making Δt /time step size smaller and smaller, the simulation time is getting more and more. We know,

$$\text{no. of time steps} = \frac{0.4}{\Delta t}$$

for this particular problem, so lower Δt means more no. of time steps. So, the time loop will run for more seconds. This is evident from fig.10 in which it can be clearly seen that $1e-4$ is having maximum simulation time while $1e-1$ is having minimum simulation time.

C. CFL stability condition for 1D Linear Convection:

$$\frac{\partial u}{\partial t} + c \frac{\partial u}{\partial x} = 0$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c \frac{u_i^n - u_{i-1}^n}{\Delta x} = 0$$

Taylor Series expansion of u_i^{n+1} ,

$$u_i^{n+1} = u_i^n + \Delta t \frac{\partial u^n}{\partial t_i} + \frac{\Delta t^2}{2!} \frac{\partial^2 u^n}{\partial t^2_i} + h.o.t$$

Where h.o.t refers to higher order terms

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{\partial u^n}{\partial t_i} + \frac{\Delta t}{2!} \frac{\partial^2 u^n}{\partial t^2_i} + O(\Delta t^2)$$

where $O(\Delta t^2)$ refers to the error which is of 2nd order,

Similarly,

$$c \frac{u_i^n - u_{i-1}^n}{\Delta x} = c \left[\frac{\partial u^n}{\partial x_i} - \frac{\Delta x}{2!} \frac{\partial^2 u^n}{\partial x^2_i} \right] + O(\Delta x^2)$$

Adding the above 2 equations we get,

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c \frac{u_i^n - u_{i-1}^n}{\Delta x} = \frac{\partial u^n}{\partial t_i} + \frac{\Delta t}{2!} \frac{\partial^2 u^n}{\partial t^2_i} + c \left[\frac{\partial u^n}{\partial x_i} - \frac{\Delta x}{2!} \frac{\partial^2 u^n}{\partial x^2_i} \right] + O((\Delta t^2), (\Delta x^2))$$

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c \frac{u_i^n - u_{i-1}^n}{\Delta x} - \left[\frac{\partial u^n}{\partial t_i} + c \frac{\partial u^n}{\partial x_i} \right] = \frac{\Delta t}{2!} \frac{\partial^2 u^n}{\partial t^2_i} - c \frac{\Delta x}{2!} \frac{\partial^2 u^n}{\partial x^2_i} + O((\Delta t^2), (\Delta x^2))$$

Now, the first two terms denote the Numerical Scheme while the next two terms denote the P.D.E (Exact Solution).

Therefore, we can write

$$\text{Numerical Scheme} - \text{P.D.E} = \text{Truncation error}(\epsilon_T) \quad (1)$$

So,

$$\text{Truncation error}(\epsilon_T) = \frac{\delta t}{2!} \frac{\partial^2 u^n}{\partial t^2_i} - c \frac{\delta x}{2!} \frac{\partial^2 u^n}{\partial x^2_i} + O((\delta t^2), (\delta x^2))$$

Let U_i^{n+1} be the exact solution of the Numerical Scheme

$$\frac{U_i^{n+1} - U_i^n}{\delta t} + c \frac{U_i^n - U_{i-1}^n}{\delta x} = 0$$

If we put U_i^{n+1} in equation 1 we will get,

$$-\left[\frac{\partial U^n}{\partial t_i} + c \frac{\partial U^n}{\partial x_i} \right] = \frac{\delta t}{2!} \frac{\partial^2 U^n}{\partial t^2_i} - c \frac{\delta x}{2!} \frac{\partial^2 U^n}{\partial x^2_i} + O((\delta t^2), (\delta x^2)) \quad (2)$$

Substituting $\frac{\partial U}{\partial t}$ with U_t and $\frac{\partial U}{\partial x}$ with U_x and so on and so forth,

$$-(U_t + cU_x)_i^n = O((\delta t), (\delta x)) \quad \text{Reducing } \epsilon_T \text{ to the lowest degree.}$$

$$U_{t_i}^n = -cU_{x_i}^n - O((\delta t), (\delta x))$$

Now taking $\frac{\partial}{\partial t}$ of $U_{t_i}^n$, we get,

$$\begin{aligned} U_{tt_i}^n &= -cU_{xt_i}^n - O((\delta t), (\delta x)) \\ U_{tt_i}^n &= -c(U_{t_i}^n)_x - O((\delta t), (\delta x)) \end{aligned}$$

$$\text{But, } U_{t_i}^n = -cU_{x_i}^n - O((\delta t), (\delta x))$$

Substituting it back into the previous equation we get,

$$U_{tt_i}^n = -c(-cU_{x_i}^n - O((\delta t), (\delta x)))_x - O((\delta t), (\delta x))$$

$$U_{tt_i}^n = c^2 U_{xx_i}^n + O((\delta t), (\delta x))$$

Now putting this value in equation 2 in place of $U_{tt_i}^n$ we get an interesting result,

$$\left[\frac{\partial U^n}{\partial t_i} + c \frac{\partial U^n}{\partial x_i} \right] = -\frac{\delta t}{2!} (c^2 U_{xx_i}^n + O((\delta t), (\delta x))) + c \frac{\delta x}{2!} \frac{\partial^2 U^n}{\partial x^2_i} + O((\delta t^2), (\delta x^2))$$

$$\left[\frac{\partial U^n}{\partial t_i} + c \frac{\partial U^n}{\partial x_i} \right] = \frac{\partial^2 U^n}{\partial x^2_i} c \frac{\delta x}{2!} \left(1 - \frac{c\delta t}{\delta x} \right) + O((\delta t^2), (\delta x^2))$$



V. REFERENCE

This linear equation when discretized using Finite Difference Method changes to convection diffusion problem. The term on the R.H.S within the first braces is the numerical diffusion which creates instability and blows up the solution and that's why it needs to be positive. So, now we have understood in details why some of the plots mentioned above did not fetch us the desired result. In other words, we have the answer to the question that why the convection problems have oscillations? and how can we reduce them in order to achieve the results?

Therefore,

$$\frac{c\delta t}{\delta x} = CFL\ no. < 1$$

III. CONCLUSION

Therefore, what we can conclude, is that, not only there is a limit to decreasing δx for getting an exact solution but also there is a limit to increasing time step size for obtaining an accurate result. Putting it all together, we can say that, CFL no. has a range in which we can get good results but outside that range our solution blows up. For convection problems,

$$CFL\ no. < 1$$

for the numerical scheme to be stable.

Consistency does not always implies stability of the solution. It might happen that a consistent solver might give unstable results owing to CFL criteria.

One interesting thing to recall is that though we do not have any diffusion terms in our governing equation still we encountered oscillations. This is because instability is added to the system, the moment we discretized it using a numerical scheme. We introduced errors, ϵ_T , which became significantly high at certain conditions and ultimately destroyed our solution. Hence, CFL criteria is equally important for higher dimensional problems.

If we refine our mesh, then we also have to reduce our time step size in such a way that the CFL condition is maintained. This on the other hand will result in, increase in the simulation time.

IV. ACKNOWLEDGEMENT

This study was supported by Sarangarajan Iyengar, Director of Product, Skill Lync. I want to thank him and the whole team of Skill Lync for providing me the correct insight and the necessary assistance. I also want to thank my brother-in-law and my friend for helping me out in this study.

- [1] Anderson J.D., Degrez G., and Grundmann R., 1992, Computational Fluid Dynamics An Introduction, DOI 10.1007/978-3-662-11350-9, (pp. 203-216).
- [2] Anderson, D.A., Tannehill, J.e. And Pletcher, R.H. Computational Fluid Mechanics and Heat Transfer, Washington, Hemisphere Publishing Company, 1984.
- [3] Beam, R.H. and Warming, R.F. 'An implicit factored scheme for the compressible Navier-Stokes, equations', AIAA Journal, Vol. 16, No.4, pp. 393-402, 1978. P. S. Huang, C. S. Chiang, C. P. Chang, and T. M. Tu, "Robust spatial watermarking technique for colour images via direct saturation adjustment," Vision, Image and Signal Processing, IEE Proceedings -, vol. 152, pp. 561-574, 2005.
- [4] <https://courses.skill-lync.com/career-programs>
- [5] ME 702- CFD, Open courseware prepared by Prof. Lorena A.Barba. <https://www.youtube.com/watch?V=cdy5xgookby&t=9s>
- [6] Brandt, A. 'Algebraic multigrid theory, the symmetric case', Proceedings of the International Multigrid Conference, Copper Mountain, Colorado, 1953
- [7] Brooks, A.N. and Hughes, T.J.R. 'Streamline upwind/petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations', Computer Methods in Applied Mechanics and Engineering, Vol. 32, pp. 199-259, 1982.
- [8] Casier, F., Deconinck, H. And Hirsch, C. 'A class of bidiagonal schemes for solving the Euler equations', AIAA Journal, Vol. 22. No. 11, pp. 1556-1563, 1984.
- [9] Dahlquist, G. The theory of linear multistep methods and related mathematical topics. Lecture Notes (microfilm), Department of Numerical Analysis, Royal Institute of Technology, Stockholm, 1976.
- [10] Dahlquist, G., Bjork, A. Numerical Methods, Englewood Cliffs, Prentice Hall, 1974.
- [11] Godunov, S.K. 'A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics', Mat. Sb., Vol. 47, pp. 271-290, 1959.
- [12] Green, J.E. 'Numerical methods in aeronautical fluid dynamics-an introduction', in Numerical Methods in Aeronautical Fluid Dynamics, (ed. P.L. Roe), New York, Academic Press, pp. 1-32, 1982.
- [13] Harten, A.: 'High resolution schemes for hyperbolic conservation laws', J. Compo Phys. 49, 357, 1983.
- [14] J arneson, A., Schmidt, W. Turkel, E. 'Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time-stepping', AIAA Paper, 81-1259, 1981.



- [15] Lerat, A. 'Une classe de schemas aux differences implicites pour les systemes hyperboliques de lois de conservation', Comptes Rendus Acad. Sc. Paris, T. 288A, pp. 1033-1036, 1979.
- [16] Maccormack, R.W. 'The effect of viscosity in hypervelocity impact cratering', AIAA Paper 69-354, 1969.
- [17] Maccormack, R.W. 'A numerical method for solving the equations of compressible viscous flow', AIAA Journal, Vol. 20, No.9, pp. 1275-1281, 1982.
- [18] Moretti, G. 'The A-scheme', Journal of Computational Physics, Vol. 7, pp. 191-205, 1979.

IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

ABOUT IJEAST

International Journal of Engineering Applied Science and Technology (IJEAST) is a peer-reviewed, open access journal that publishes high-quality research papers in the field of Engineering, Applied Science and Technology.

IJEAST aims to provide a platform for researchers, academicians, and professionals to share their innovative ideas, research findings, and practical experiences with the global scientific community.

FOCUS AREAS

- Engineering
- Applied Science
- Technology
- Innovation & Development
- Interdisciplinary Studies



PEER REVIEWED

All submissions are rigorously peer reviewed to ensure quality.



OPEN ACCESS

Free and unrestricted access to research for all.



GLOBAL REACH

Connecting researchers and professionals worldwide.



TIMELY PUBLICATION

We ensure a swift and efficient publication process.



For more information, visit our website

www.ijeast.com



INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

✉ editor@ijeast.com

🌐 www.ijeast.com

📍 India



2455-2143