ISBN 978-82-471-3445-0 (printed ver.)
ISBN 978-82-471-3446-7 (electronic ver.)
ISSN 1503-8181

Doctoral theses at NTNU, 2012:85

Muhammad Asif Farooq

Cartesian Grid Method for Compressible Flow Simulation

Doctoral Thesis

Muhammad Asif Farooq

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology
Thesis for the degree of
philosophiae doctor
Faculty of Engineering Science and Technology
Department of Energy and Process Engineering

Doctoral theses at NTNU, 2012:85

NTNU

NTNU
Norwegian University of
Science and Technology

Muhammad Asif Farooq

Cartesian Grid Method for Compressible Flow Simulation

Thesis for the degree of philosophiae doctor

Trondheim, March 2012

Norwegian University of
Science and Technology
Faculty of Engineering Science and Technology
Department of Energy and Process Engineering

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Thesis for the degree of philosophiae doctor

Faculty of Engineering Science and Technology
Department of Energy and Process Engineering

ISBN 978-82-471-3445-0 (printed ver.)
ISBN 978-82-471-3446-7 (electronic ver.)
ISSN 1503-8181

Doctoral Theses at NTNU, 2012:85

Printed by Fagtrykk Trondheim AS

Cartesian Grid Method for Compressible Flow Simulation

Muhammad Asif Farooq

Dedication

Firstly, this work is dedicated to the people of Pakistan.
Secondly, this work is dedicated to my beloved father (late) and my loving sweet mother for all care, affection and support during my studies and research. It is difficult for me to express in words the love my mother has provided me during these years.
I would also like to dedicate this work to my good friend Kiren Khan for her unconditional support, prayers and love during the course of this research work. Also, I would like to dedicate this work to my younger brothers Athar Farooq, Tayyab Farooq, Zeeshan Farooq and my sweet younger sister, Asma Farooq.
I am overwhelmed by the support, favor, inspirations and patience of my family and Kiren Khan during the course of this study.

Muhammad Asif Farooq
Trondheim December 2011

Abstract

The Cartesian grid method is an alternative to the existing methods to solve a physical problem governed by partial differential equations (PDEs) computationally. Researchers are interested in this method due to its simplicity of grid generation, less computational effort and ease of implementation into a computer code. One of the other options to solve a physical PDE problem is by the body-fitted grid method. In the body-fitted grid method, the boundary points are grid points. This is not the case with the Cartesian grid method where the body wall is embedded as a boundary into a Cartesian grid resulting in irregular cells near the embedded boundary. These irregular cells near the embedded boundary are known as cut-cells. Instead of using special treatments of the cut-cells or enforcing the presence of the embedded boundary by adding source terms at the Cartesian grid points near the boundary, the kinematic and other boundary conditions can be introduced in the Cartesian grid method via ghost points. Those grid points which lie inside the embedded boundary and are also a part of computation are called ghost points. Inactive grid points inside the embedded boundary are referred to as solid points.

In the present Cartesian grid method, based on a ghost point treatment, local symmetry conditions are imposed at the embedded wall boundary. The ghost point treatments available in the literature are difficult to implement due to complex procedures. We are introducing a new approach to approximate the kinematics of the embedded boundary by a very simple ghost point treatment called the simplified ghost point treatment. In this approach, we consider the grid lines in the x- and y- directions as approximations of the lines normal to the embedded boundary depending on whether the angle between the normal and the x- or y-directions is closer.

For 1D hyperbolic nonlinear systems of conservation laws, we use the moving normal shock wave as a test case for the 1D compressible Euler equations. For the 2D compressible Euler equations, we test the simplified ghost point treatment for an oblique shock wave generated by a wedge. Then, we verified our approach for slender bodies, namely for supersonic flow over a circular arc airfoil and for transonic flow over a circular arc bump in a channel. In a final problem, we applied the simplified ghost point treatment to blunt body flow and considered supersonic flows over a cylinder using the 2D compressible Euler and Navier-Stokes equations. The results are good or comparable to those found in the existing literature.

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) for partial fulfilment of the requirements for the degree of philosophiae doctor. The thesis work was carried out at the Department of Energy and Process Engineering (EPT) under the supervision of Professor Bernhard Müller. The research was funded by the Higher Education Commision (HEC) of Pakistan. I am really thankful to the Higher Education Commission (HEC) of Pakistan for allocating a scholarship to me. It was difficult for Pakistan to support students at difficult time but they did provide scholarship with great care. My department was kind enough at the end of five months of my PhD education to provide funding for my research and studies as well.

The main motivation of this work has been to introduce a new and simple ghost point approach for the Cartesian grid method near the embedded boundaries. In the first part of this thesis, the present Cartesian grid method for compressible flow simulation is outlined. Then, our research papers are presented in the second part of this thesis.

Muhammad Asif Farooq
Trondheim, December 2011

Acknowledgements

The support of Prof. Bernhard Müller has been phenomenal in the completion of this work and my personal development. He first accepted me as a PhD student when I was struggling to find a supervisor. Since I had not covered programming in my Master's degree program, he encouraged me to learn programming and helped me a lot. Without his support and encouragement, it was not possible for me to conduct this research work. Thank you Bernd!

I am thankful to all my colleagues. I am thankful to Claudio Walker for useful discussion, whether it is politics or academics. He spent quite a few hours with me discussing my research work and answering Matlab query. Thanks is due to my very good friend Martin Larsson for helping me in computer work and programming. I am also grateful to my colleagues Joris Verschaeve, Kristian Etienne Einarsrud, Simen Andreas Ellingsen, Hatef Khaledi Alidoosti, Mustafa Barri, Anne Line Løvholm, Jan Fredrik Helgaker, Karl Yngve Lervåg, Halvor Lund, Bita Najmi and Nida Razi for creating a friendly environment. I would also like to thank Asif Mushtaq and his family for their support. I also thank my friends Qazi Sohail, Fazal Wahab and Raja Kashif for helping me out in difficult times.

The results in chapter 6 are found with the collaboration of Are Skøien, Master's student at NTNU, and my supervisor, Prof. Bernhard Müller. The complete programming is implemented by Are Skøien, with feedback from Prof. Bernhard Müller and myself, in order to obtain results for supersonic flow over two cylinders for the 2D compressible Navier-Stokes equations. I wish Are Skøien success in the future!

In Pakistan, I would like to thank my closest and dearest friend **Nayyar Afaq**. Nayyar always encouraged me to concentrate on my studies whenever I felt low. Nayyar's advice and encouragement in work will be appreciated as far as I continue my research work. I would also like to thank Haroon Riaz and Mubasher Jamil for their support. Especially, I am grateful to Haroon Riaz for reading and giving feedback on this manuscript. Thanks is also due to my friends Hina Shafique and Farah Asif.

I am grateful to the people at EPT for providing me a conducive and professional environment and all the facilities required for the research. I am grateful to the secretaries of the department for their help at the time when it was most needed.

List of Papers

The thesis is based on and contains the following papers.

Paper 1. M. Asif Farooq & B. Müller. 2010 Analysis of the Accuracy of the Cartesian Grid Method. In: *the* 23^{rd} *Nordic Seminar on Computational Mechanics* (eds. A. Eriksson & G. Tibert), pp. 53–56.

Paper 2. M. Asif Farooq & B. Müller. 2011 Investigation of the Accuracy of the Cartesian Grid Method. In: 8^{th} *International Bhurban Conference on Applied Sciences & Technology*, pp. 45-53.

Paper 3. M. Asif Farooq & B. Müller. 2011 A Cartesian Grid Method for Compressible Inviscid Flows. In: 6^{th} *National Conference on Computational Mechanics* (eds. B. Skallerud & H. I. Andersson), Tapir Academic Press, pp. 47–58.

Paper 4. M. Asif Farooq & B. Müller. 2011 Cartesian Grid Method for the Compressible Euler Equations. In: *Finite Volumes for Complex Applications VI Problems & Perspectives* (eds. J. Fort, J. Fürst, J. Halama, R. Herbin & F. Hubert), Springer-Verlag, pp. 449–456.

Paper 5. M. Asif Farooq & B. Müller. 2011 Accuracy Assessment of the Cartesian Grid Method for Compressible Inviscid Flows Using a Simplified Ghost Point Treatment. *RAKENTEIDEN MEKANIIKKA (Journal of Structural Mechanics)*, **44**, pp. 279–291.

Erratum of Paper 5. Erratum to "Accuracy assessment of the Cartesian grid method for compressible inviscid flows using a simplified ghost point treatment" [Rakenteiden Mekaniikka (Journal of Structural Mechanics) Vol. 44. No. 3, 2011, pp. 279-291].

Paper 6. M. Asif Farooq & B. Müller. 2012 A Cartesian Grid Method for Compressible Flows to Compute Shock Waves. In: 9^{th} *International Bhurban Conference on Applied Sciences & Technology* To appear in conference proceedings.

Contents

Chapter 1

Introduction

Computational Fluid Dynamics, or CFD in short, has been an emerging alternative field to the experiment. CFD is increasingly becoming important in rheology, astrophysics, vehicle design, aircraft industry, aeroacoustics and meteorology. This has also introduced new fields such as computational rheology, computational biology, computational science and engineering and computational astrophysics [2]. Recently, scientists have been able to simulate tsunami waves with the help of CFD [3, 4]. This is an important development to save the precious lives of people from such a devastating catastrophe. The reason of the fame of CFD is due to its effectiveness in the solution of a complex problem at the minimum cost as compared to the laboratory experiment. Generally, the test of a physical problem requires a huge amount of money in the lab. On the other hand, in CFD, thanks to the progress in computer technology, we can test the same problem on a computer with minimum cost and can find out the advantages and disadvantages of a particular idea. Even though the experiments are successful due to accuracy, as it is more close to the reality, but it costs large amounts of money to repeat an experiment. CFD, however, is the solution to repeat the same problem and suggest a correction in the experiment [2].

Since the introduction of a computer and the development of more powerful computers, the interest in CFD has increased exponentially. This development in computer technology also made the researcher capable of introducing new ideas in science and engineering. In the early 1980's, the solutions to the 2D and 3D Euler equations were available [2]. After the solution of inviscid flows, the interest in viscous flows has increased, leading to the search of solutions for the Navier-Stokes equations [2]. This was the time when body-fitted structured grid methods were used [5, 6]. The structured grid method solved many complex problems for the Euler and Navier-Stokes equations. In the 1990's, unstructured grid method appeared in the literature [7]. In the unstructured grid method, the computational domain contains triangles, tetrahedra, prisms and pyramids [2]. An unstructured grid can be composed of a mixture of different elements. The important step in both structured and unstructured grid method is to preserve the conservation properties [2]. The structured grid method is more efficient in terms of accuracy and CPU time than the unstructured grid method [2]. Although, the standard body-fitted grid method

(structured and unstructured grid methods) has gained popularity due to successful results in solving difficult problems, this method requires a lot of effort to generate a grid. This is especially true for unstructured grid methods where extensive effort is required to generate a grid. Nevertheless, the unstructured grid method is often preferred, because unstructured grid generation can more easily be automatized than structured grid generation for complex domains.

The Cartesian grid method is an alternative to the standard body-fitted grid method. The Cartesian grid method [8, 9, 10, 11, 12] has been becoming popular in the researcher community due to its ability to solve the problem with ease and less computational effort for grid generation. The grid adaptation is also possible in the Cartesian grid method. The Cartesian grid method was introduced for incompressible flow in a famous paper [13]. In this approach, a force term is added in the incompressible momentum equation to enforce the presence of the immersed boundary by kinetic conditions on the Cartesian grid. But this approach is not so practical for compressible flow due to sensitive coupling of all variables. It is named as the immersed boundary method for incompressible flows, while for compressible flow this method is often termed as the Cartesian grid method. The difference between the body-fitted and the Cartesian grid methods lies in the representation of the body boundaries on the grids. In the standard body-fitted grid method, the boundary points are grid points. On the other hand, the embedded boundary in the Cartesian grid method intersects the grid lines arbitrarily, hence producing irregular cells near the embedded boundary. The key to solving the Euler and Navier-Stokes equations lies near the embedded boundary where the grid is not regular. Since the cells near the embedded boundary are not regular, we can use different approaches to solve the problem.

1.1 The Cartesian Grid Method

The Cartesian grid method contains uniform grid spacings in x- and y-directions except near the embedded boundary where the grid is irregular. The random intersection of the embedded boundary and the grid lines creates irregular cells near the embedded boundary. These irregular or cut-cells create problems for the scheme to be implemented. Due to these cut-cells, the stencils near the embedded boundary require special treatment. Another weakness of this Cartesian grid method is the stability issue due to the cut-cells. Near the embedded boundary, the time step restriction is quite severe. The time restriction is even more severe for the compressible Navier-Stokes equations due to the presence of boundary layers. One of the solutions is to merge these cut-cells with neighboring cells [14]. Another approach to deal with irregular cells near the embedded boundary is to use the h-box method [15]. The basic idea of the h-box method is to calculate the fluxes at the faces of the small cells. Apart from that, a new approach called Building-Cube Method has

been introduced [16]. In this method, a high density mesh is proposed on the Cartesian grid. Near the wall boundary, a staircase with a high resolution is produced to capture the features in the boundary layer. With the Building-Cube Method, complex geometries can be treated, adaptive grids can be used, and higher order schemes can be implemented.

Another approach is to use ghost point treatment near the embedded boundary. In [17], one such ghost point treatment is introduced near the embedded boundary. In this research work, we have also opted to use ghost point treatment rather than the above mentioned treatments near the embedded boundary. Since this approach is quite easy to implement into a computer program relative to the above mentioned treatments.

One of the weaknesses of the Cartesian grid method is the lack of accuracy near the embedded boundary. Due to the irregular cells near the embedded boundary, we are bound to lose accuracy while using this method.

Fig. 1.1 shows an example of the Cartesian grid for a flow over a wedge. As mentioned above, due to the arbitrary intersection of the grid lines and an embedded boundary, the boundary points and grid points do not coincide with each other. On the other hand, Fig. 1.2 shows a structured body-fitted grid where the boundary points of the circular arc airfoil are grid points. Hence, in the body-fitted grid method we do not need a ghost point treatment near the embedded boundary.

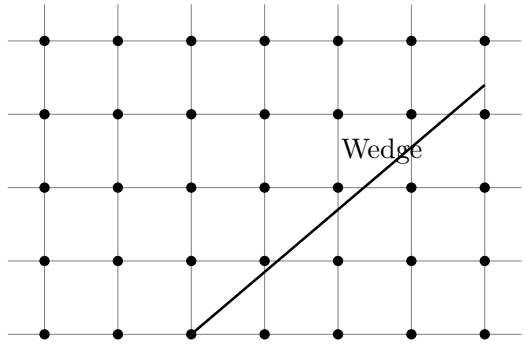


Figure 1.1: Embedded wedge in the Cartesian grid.

In this thesis, we investigate the accuracy of the Cartesian grid method for 1D and 2D nonlinear PDEs. We impose wall boundary conditions at the embedded boundary. The key to the Cartesian grid method is the ghost point treatment near the embedded boundary. The main motivation of this thesis is to introduce a simple ghost point treatment near the embedded boundary for the Cartesian grid method. The ghost point treatments existing in the literature are quite involved and difficult to apply. We are introducing a new approach to handle the ghost points near the embedded boundary. The

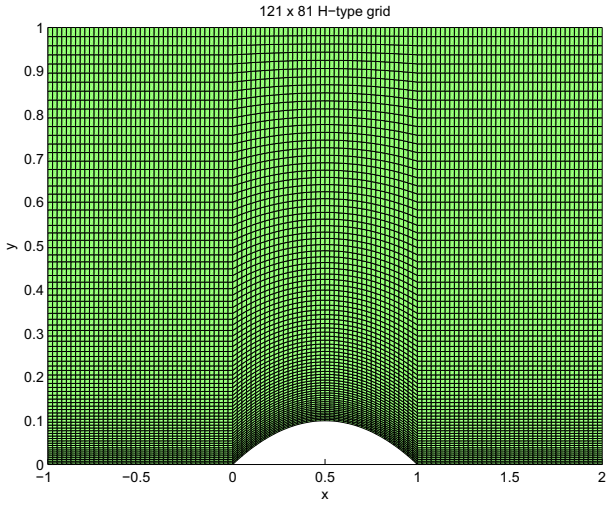


Figure 1.2: Standard body-fitted grid of a circular arc airfoil.

ghost point treatment introduced in this thesis is called the simplified ghost point treatment. The simplified ghost point treatment is not only easy to implement into a computer program, but also the order of the method can easily be increased.

This thesis is outlined as follows. In Chapter 2, we present the governing equations as well as the boundary conditions for the 2D compressible Euler and Navier-Stokes equations. The introduction of the numerical methods is given in Chapter 3. In Chapter 4, we give a brief overview of ghost point treatment near the embedded boundary presented in [1]. The new approach called the simplified ghost point treatment is also presented in Chapter 4. Results for the 2D compressible Euler and Navier-Stokes equations are shown in Chapters 5 and 6, respectively. Conclusions are drawn in Chapter 7. Research papers are given after Chapter 7.

Chapter 2

Governing Equations

The motion of fluids is governed by the following fundamental conservation laws

1. Conservation of mass
2. Conservation of momentum
3. Conservation of energy

The first conservation law states that the mass of the fluid remains conserved. The second conservation law says that the motion of the fluid is based on Newton's second law of motion. The third conservation law is the first law of thermodynamics for fluid motion.

In CFD, the conservation laws of mass, momentum and energy for a compressible fluid are referred to as the compressible Navier-Stokes equations. For inviscid flow, they simplify to the compressible Euler equations. Before moving on to these governing equations, it is important to be familiar with the Burgers' equation as a model equation.

The aim of this chapter is to introduce the reader not only to the model equation but also to the governing equations for compressible inviscid and viscous flows. The other purpose of this chapter is to include the basics of boundary conditions necessary for numerical solutions.

This chapter is organised as follows. In section 2.1, the Burgers' equation is presented. The definition for compressible flow and the equations of state for perfect gas is given in sections 2.2 and 2.3. The governing equations for the 2D compressible Euler and Navier-Stokes equations are presented in sections 2.4 and 2.5. Boundary conditions for the Euler and Navier-Stokes equations are given in sections 2.6 and 2.7.

2.1 Burgers' Equation

In order to understand the convection and diffusion of the fluid, we choose the 1D Burgers' equation as a model equation. The 1D Burgers' equation is given as [18, 19]

$$\frac{\partial u}{\partial t} + \frac{\partial \frac{u^2}{2}}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}, \tag{2.1}$$

with initial condition

$$u(x, 0) = f(x). \tag{2.2}$$

The boundary conditions are given as

$$u(L_1, t) = g(x_a, t), \qquad u(L_2, t) = g(x_b, t). \tag{2.3}$$

In (2.1), u is the conservative variable, and ν is the kinematic viscosity. The equation (2.1) reduces to the inviscid Burgers' equation for $\nu = 0$. The detailed solution of the inviscid Burgers' equation is given in Paper 2 for a test case with periodic boundary conditions $u(0, t) = u(1, t)$.

2.2 Compressibility

In nature, certain fluids exist which largely retain their density when pressure is applied on them. On the other hand, if the pressure is applied to gases, then the density of the gases can be affected. If the density changes significantly with the flow of the applied pressure, then the fluid is called a compressible fluid. Its flow is called as compressible flow. The compressibility of the fluid is also related to the speed of sound. We can relate the velocity of fluid to the speed of sound by the Mach number [20]

$$M = \frac{u}{c}, \tag{2.4}$$

where u is the velocity of the flow and c is the speed of sound. If the Mach number $M < 0.3$ and acoustic effects can be neglected, then the fluid is treated as incompressible [20]. The speed of sound for a perfect gas is defined as

$$c = \sqrt{\left(\frac{\partial p}{\partial \rho}\right)_s} = \sqrt{\frac{\gamma p}{\rho}}. \tag{2.5}$$

In (2.5), p, ρ, s and γ are pressure, density, entropy and ratio of specific heats, respectively. The flow is categorized based on the value of Mach number M.

1. $M < 1$, subsonic flow
2. $M > 1$, supersonic flow
3. $M = 1$, sonic flow
4. $M > 5$, hypersonic flow

2.3 Equations for Perfect Gas

In this section, we consider the equations of state for perfect gas. The relation between pressure p, density ρ and temperature T is given as

$$p = \rho R T \tag{2.6}$$

where T is temperature and $R = c_p - c_v$ is the gas constant. c_p and c_v are the specific heats at constant pressure and volume, respectively. The gas constant is $R = 287\ \frac{\mathrm{m}^2}{\mathrm{s}^2\mathrm{K}}$ for air at standard conditions. The specific internal energy and enthalpy are given by

$$e = c_v T, \quad h = c_p T. \tag{2.7}$$

If we combine (2.6) and (2.7) then we obtain the following relationship for pressure

$$p = (\gamma - 1)(\rho E - \frac{1}{2}\rho(u^2 + v^2 + w^2)), \tag{2.8}$$

where E, u, v, w are the total energy and velocities in x-, y- and z-directions, respectively. The above equation (2.8) is generally used to compute pressure for perfect gas when solving the compressible Euler and Navier-Stokes equations, cf. below.

2.4 Compressible Euler Equations

The 2D compressible Euler equations serve as a model for 2D nonlinear hyperbolic systems. The 2D compressible Euler equations in conservative form read

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0, \tag{2.9}$$

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \ F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho E + p)u \end{bmatrix}, \ G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho E + p)v \end{bmatrix}. \tag{2.10}$$

In (2.10), ρ, u, v, E, and p denoting density, velocity components in x-and y-directions, total energy per unit mass and pressure, respectively.

As mentioned above for perfect gas, we have the following relation

$$p = (\gamma - 1)(\rho E - \frac{1}{2}\rho(u^2 + v^2)), \tag{2.11}$$

where γ is the ratio of specific heats. We consider $\gamma = 1.4$ for air.

If we insert $v = 0$ and $\frac{\partial u}{\partial y} = 0$ in (2.9) - (2.11) then we get 1D compressible Euler equations. If we consider $U = [u_1, u_2, u_3]^{\mathrm{T}}$ and flux function $F = [F_1, F_2, F_3]^{\mathrm{T}}$ and remove the y-momentum equation, then the Jacobian matrix for the 1D compressible Euler equation can be written as

$$\frac{\partial F}{\partial U} = \begin{bmatrix} \frac{\partial F_1}{\partial u_1} & \frac{\partial F_1}{\partial u_2} & \frac{\partial F_1}{\partial u_3} \\ \frac{\partial F_2}{\partial u_1} & \frac{\partial F_2}{\partial u_2} & \frac{\partial F_2}{\partial u_3} \\ \frac{\partial F_3}{\partial u_1} & \frac{\partial F_3}{\partial u_2} & \frac{\partial F_3}{\partial u_3} \end{bmatrix}. \tag{2.12}$$

The eigenvalues of the Jacobian matrix (2.12) for the 1D Euler equations are $\lambda_1 = u + c,\ \lambda_2 = u,\ \lambda_3 = u - c$. If all the eigenvalues are real and the eigenvectors of $\frac{\partial F}{\partial U}$ are linearly independent, then the system is hyperbolic. A closer look at the eigenvalues λ_1, λ_2 and λ_3 shows that all these eigenvalues are real and distinct. Thus, the 1D compressible Euler equations constitute a hyperbolic system [21]. Similarly, the 2D compressible Euler equations are a hyperbolic system.

2.5 Compressible Navier-Stokes Equations

The 2D compressible Navier-Stokes equations in conservative form read as [22]

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0, \tag{2.13}$$

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, \tag{2.14}$$

$$F = \begin{bmatrix} \rho u \\ \rho u^2 + p - \tau_{xx} \\ \rho uv - \tau_{xy} \\ (\rho E + p)u - u\tau_{xx} - v\tau_{xy} + q_x \end{bmatrix}, \tag{2.15}$$

$$G = \begin{bmatrix} \rho v \\ \rho uv - \tau_{xy} \\ \rho v^2 + p - \tau_{yy} \\ (\rho E + p)v - u\tau_{xy} - v\tau_{yy} + q_y \end{bmatrix}, \tag{2.16}$$

with ρ, u, v, E, and p are density, velocities in x- and y-directions, total energy per unit mass and pressure, respectively.

For perfect gas we have the following relation

$$p = (\gamma - 1)(\rho E - \frac{1}{2}\rho(u^2 + v^2)). \tag{2.17}$$

The components of the shear stress and heat flux vector introduced in (2.15) and (2.16) are given as [22]

$$\tau_{xx} = \frac{2}{3}\mu(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}), \tag{2.18}$$

$$\tau_{yy} = \frac{2}{3}\mu(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}), \tag{2.19}$$

$$\tau_{xy} = \tau_{yx} = \mu(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}), \tag{2.20}$$

$$q_x = -k\frac{\partial T}{\partial x}, \tag{2.21}$$

$$q_y = -k\frac{\partial T}{\partial y}. \tag{2.22}$$

The Sutherland's formula can be used to find the viscosity

$$\mu = \frac{C_1 T^{\frac{3}{2}}}{T + C_2}, \tag{2.23}$$

where C_1 and C_2 are constants for a given gas with $C_1 = 1.458 \times 10^{-6}\ \frac{\text{kg}}{\text{ms}\sqrt{\text{K}}}$ and $C_2 = 110.4$ K. We consider $\gamma = 1.4$ for air. The results presented in this thesis are only obtained for a constant viscosity. The Prandtl number is used to find the thermal conductivity

$$k = \frac{c_p \mu}{Pr}. \tag{2.24}$$

For air at standard conditions $Pr = 0.72$. The specific heat at constant pressure c_p can be determined by the following equation

$$c_p = \frac{\gamma R}{\gamma - 1}. \tag{2.25}$$

The value of c_p is 1004.5 $\frac{\text{m}^2}{\text{s}^2\text{K}}$. Similarly, we get the specific heat at constant volume c_v from

$$c_v = \frac{R}{\gamma - 1}. \tag{2.26}$$

The value of c_v is 717.5 $\frac{\text{m}^2}{\text{s}^2\text{K}}$.

2.6 Boundary Conditions for Euler Equations

One of the important steps to solve the 2D compressible Euler equations is to correctly prescribe boundary conditions. In a hyperbolic system, all the information travels on the characteristics. The slope of the characteristics decides how to implement boundary conditions. If the eigenvalues of the Jacobian matrix of the flux in the inner normal direction are positive, then the information travels from exterior to the interior. In this case, physical boundary conditions must be given. If the propagation is from the interior to the exterior of the domain, then numerical boundary condition must be given. In conclusion, the sign of eigenvalues will define to use numerical or physical boundary conditions [23].

2.6.1 Subsonic Inflow

In Fig. 2.1, we show the characteristics for subsonic inflow. If the inlet flow is subsonic, then two eigenvalues are positive and one is negative. So we need to prescribe three boundary conditions and one numerical boundary condition.

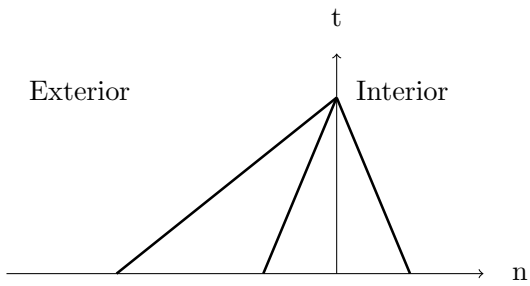


Figure 2.1: The subsonic inflow.

2.6.2 Subsonic Outflow

In Fig. 2.2, we show the characteristics for subsonic outflow. At the subsonic outlet boundary, two eigenvalues are negative and one eigenvalue is positive. Therefore, three numerical and one physical boundary condition must be set at the outlet.

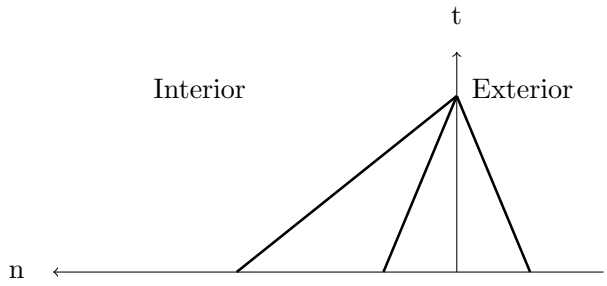


Figure 2.2: The subsonic outflow.

2.6.3 Supersonic Inflow

In Fig. 2.3, we show the characteristics for supersonic inflow. All the eigenvalues are positive so all boundary conditions must be physical and no numerical boundary conditions must be given.

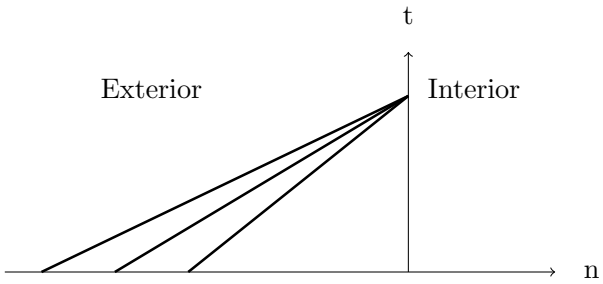


Figure 2.3: The supersonic inflow.

2.6.4 Supersonic Outflow

In Fig. 2.4, we show the characteristics of supersonic outflow. Since all eigenvalues are negative so all boundary conditions must be numerical. Numerical boundary conditions can be given by extrapolation.

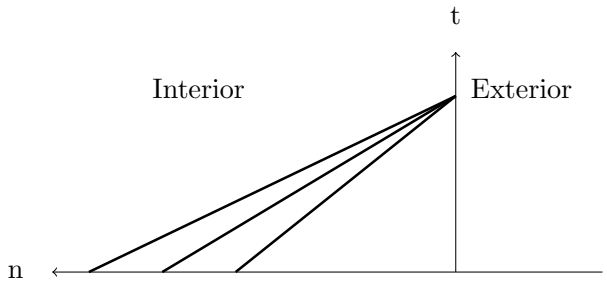


Figure 2.4: The supersonic outflow.

2.6.5 Solid Wall Boundary Conditions

No mass or other convective fluxes penetrate through a solid wall. This implies that the normal velocity is zero. One eigenvalue is positive so we need one physical boundary condition. The other variables, i.e. tangential velocity components, density and pressure have to be determined [23].

2.7 Boundary Conditions for Navier-Stokes Equations

2.7.1 No-Slip Boundary Conditions

At the stationary solid wall we use no-slip boundary conditions $u_w = 0 = v_w$. This means the velocity of the fluid at the solid wall is equal to the velocity of the wall.

2.7.2 Adiabatic Walls

For an adiabatic wall, there is no heat flux through the solid wall. Mathematically it means

$$\frac{\partial T}{\partial n} = 0 \tag{2.27}$$

at the solid wall.

2.7.3 Isothermal Walls

At the solid surface we can also keep the temperature fixed

$$T = T_w \tag{2.28}$$

Chapter 3

Numerical Methods

In this chapter, basic concepts of numerical methods are introduced. We focus on finite volume methods to compute shock waves.

Two important distinctions can be made: shock capturing and shock fitting numerical schemes. Then the important step is to discretize the basic equations. The introduction of explicit methods is also given in this chapter.

This chapter is organised as follows. The discussion on shock capturing or shock fitting methods is presented in section 3.1. The differentiation of different types of discretization methods is drawn in section 3.2 The distinction between cell-vertex and cell-centred is drawn in section 3.3. The spatial discretization for the 2D compressible Euler equations is presented in section 3.4. The total variation diminishing (TVD) property is defined in section 3.5. Courant-Friedrichs-Levy (CFL) condition, MUSCL and spatial discretization for the 2D Navier-Stokes equations are presented in sections 3.6, 3.7 and 3.8, respectively. Time discretization is explained in section 3.9.

3.1 Shock Computations

The compressibility of the fluid can produce discontinuities in a solution. The inviscid Burgers' equation shows that nonlinearity in the model equation can generate discontinuities [2]. A numerical method handles shocks in one of the two following ways:

1. Shock Capturing Methods
2. Shock Fitting Methods

3.1.1 Shock Capturing Methods

Shock capturing techniques are very famous in the research community. This is due to the fact that this method is easy to implement to compute shock waves.

In this method, the governing equations are written in conservation form to capture shocks. Total Variation Diminishing (TVD) schemes like the Godunov method or approximate Riemann solvers can be used to compute shocks. The

shocks captured in this way are quite smeared. In this method, contrary to shock fitting methods, the location of the shock is not treated as a sharp discontinuity and moving boundary, but captured approximately over a number of cells [2, 23, 22].

3.1.2 Shock Fitting Methods

Though, shock capturing methods are famous but the shock produced in this process is quite smeared. To capture shocks correctly one uses shock fitting methods. In this method, the exact location of the shock is calculated from the compressible Euler equations using the Rankine-Hugoniot conditions at the shock boundary, which is allowed to move [2].

3.2 Basic Discretization

Three basic kind of discretization schemes exist to solve a physical problem governed by PDEs

1. Finite Volume Method (FVM)
2. Finite Difference Method(FDM)
3. Finite Element Method (FEM)

In FVM, a variable is placed at the centre of cells and the quantity is averaged over the cells. The derivation involves the integrals which ensure conservation properties and capturing discontinuities. The disadvantage of this discretization is the difficulty of increasing the order of the method. It is not effective due to low order restriction for high accuracy demands.

In FDM, the quantities are placed at the nodes of a grid. The advantage involves the implementation in computer programs apart from the application to higher order methods. For the first and second order discretizations, the FVM and FDM are the same on Cartesian grids, except for the boundary treatments.

One of the powerful techniques in numerical methods is FEM. In FEM, the variables are distributed on the elements. The advantage of FEM is that it is known for handling a complex geometry relatively easily as compared to FDM.

3.3 Cell-Centred Scheme vs Cell-Vertex Scheme

We focus on finite volume methods, because their conservation property is a prerequisite to correctly capture shocks. The computational domain can be discretized by cells, elements or nodes. The solution data can be presented on the computational domain in one of two ways:

1. Cell-centred scheme

2. Cell-vertex scheme

3.3.1 Cell-Centred Scheme

In cell-centered schemes, the data is placed at the centroid of the cells and represented as cell averages. The fluxes are calculated at the cell faces.

3.3.2 Cell-Vertex Scheme

In the cell-vertex scheme, the variables are located at the nodes contrary to the cell-centred scheme. In some cases, the cell-vertex scheme is more suitable and the cell-centred scheme in others. If we consider the accuracy between cell-centred and cell-vertex schemes, then the cell-vertex scheme is first order accurate even on deformed grids, while the cell-centred scheme might become inconsistent [2].

3.4 Spatial Discretization for Compressible Euler Equations

The node-centred finite volume method yields the following semi-discretization of the 2D compressible Euler equations (2.9)

$$\frac{\mathrm{d}U_{i,j}}{\mathrm{d}t} = -\frac{F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}}{\Delta x} - \frac{G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}}}{\Delta y}. \tag{3.1}$$

Where $U_{i,j}$ is the approximation of the average of U in the cell $\Omega_{i,j} = [x_i - \frac{\Delta x}{2}, x_i + \frac{\Delta x}{2}] \times [y_j - \frac{\Delta y}{2}, y_j + \frac{\Delta y}{2}]$, i.e.

$$U_{i,j} \approx \frac{1}{\Delta x.\Delta y} \int_{\Omega_{i,j}} U(x,y,t) \quad dxdy. \tag{3.2}$$

If we interpret (3.1) as a conservative finite difference method, $U_{i,j}$ is an approximation of the exact solution $U(x_i, y_j, t)$. $F_{i+\frac{1}{2},j}$ and $G_{i,j+\frac{1}{2}}$ are numerical fluxes for the 2D compressible Euler equations. The vector of the conservative variables U and the flux vectors F and G are defined in (2.10). We focus here on the following numerical flux functions.

3.4.1 Local Lax-Friedrichs Scheme

The dissipative numerical fluxes of the local Lax-Friedrichs method for F and G are defined as follows

$$F^{lLF}_{i+\frac{1}{2},j} = \frac{1}{2}\left[F(U_{i,j}) + F(U_{i+1,j}) - max(|u_{i+1,j}| + c_{i+1,j}, |u_{i,j}| + c_{i,j})(U_{i+1,j} - U_{i,j})\right], \tag{3.3}$$

$$G^{lLF}_{i,j+\frac{1}{2}} = \frac{1}{2}\left[G(U_{i,j}) + G(U_{i,j+1}) - max(|v_{i,j+1}| + c_{i,j+1}, |v_{i,j}| + c_{i,j})(U_{i,j+1} - U_{i,j})\right]. \tag{3.4}$$

In equations (3.3) and (3.4), c is the speed of sound.

3.4.2 Lax-Friedrichs Scheme

The more dissipative Lax-Friedrichs fluxes for F and G are defined as

$$F^{LF}_{i+\frac{1}{2},j} = \frac{1}{2}\left[F(U_{i,j}) + F(U_{i+1,j}) - \frac{\Delta x}{\Delta t}(U_{i+1,j} - U_{i,j})\right], \tag{3.5}$$

$$G^{LF}_{i,j+\frac{1}{2}} = \frac{1}{2}\left[G(U_{i,j}) + G(U_{i,j+1}) - \frac{\Delta y}{\Delta t}(U_{i,j+1} - U_{i,j})\right], \tag{3.6}$$

where Δx, Δy and Δt are spatial and temporal spacings.

3.5 Total Variation Diminishing (TVD)

We consider a two-level method for a scalar conservation law $u_t + f(u)_x = 0$ like the inviscid Burgers' equation. Let $\mathbf{u}^n$ denote a grid function vectors (u^n_j), $j = -\infty, ..., \infty$, at time $t_n = n.\Delta t$. Let $\mathbf{u}^{n+1}$ denote the numerical solution of the two-level method of time t_{n+1}. The two-level set method is called total variation diminishing (TVD) if for any data $\mathbf{u}^n$ the numerical solution $\mathbf{u}^{n+1}$ satisfies

$$TV(\mathbf{u}^{n+1}) \leq TV(\mathbf{u}^n), \tag{3.7}$$

where $TV(\mathbf{u}) = \sum_j |u_{j+1} - u_j|$. The meaning of the above equation (3.7) is that the total variation of $\mathbf{u}$ is non-increasing. The idea of the TVD is to prevent oscillations in the schemes. Usually, in numerical methods, the total variation can increase or decrease with time. Discontinuities in computational domain cause oscillations. The prevention of such oscillations can be achieved by adding an extra term called numerical dissipation. The numerical dissipation in the numerical flux is based on the characteristic speed $\left|f'(u)\right|$ for the local Lax-Friedrichs scheme and $\frac{\Delta x}{\Delta t}$ for the Lax-Friedrichs scheme. Such dissipative schemes are quite suitable for discontinuities.

3.6 Courant-Friedrichs-Levy Condition

The CFL number for the 2D compressible Euler equations is defined as

$$CFL = \Delta t \max_{i,j} \left(\frac{sp(A_1(U_{i,j}))}{\Delta x} + \frac{sp(A_2(U_{i,j}))}{\Delta y} \right), \tag{3.8}$$

where $sp(A_1(U_{i,j}))$ and $sp(A_2(U_{i,j}))$ are the spectral radii of the Jacobian matrices $A_1 = \frac{\partial F}{\partial U}$ and $A_2 = \frac{\partial G}{\partial U}$, respectively. The CFL condition for the explicit Euler method reads: $CFL \leq 1$.

3.7 MUSCL

At cell face $x_{i+\frac{1}{2},j}$ between cells (i,j) and $(i+1,j)$, the left and right extrapolated variables in the MUSCL (Monotone Upstream-centered Schemes for Conservation Laws) approach [25] are defined as

$$U^L_{i+\frac{1}{2},j} = U_{i,j} + \frac{1}{2}\text{minmod}(U_{i,j} - U_{i-1,j}, U_{i+1,j} - U_{i,j}), \tag{3.9}$$

$$U^R_{i+\frac{1}{2},j} = U_{i+1,j} - \frac{1}{2}\text{minmod}(U_{i+2,j} - U_{i+1,j}, U_{i+1,j} - U_{i,j}), \tag{3.10}$$

where

$$\begin{aligned} \text{minmod}(a,b) &= \begin{cases} a & \text{if} \quad |a| \leq |b| \quad and \quad ab > 0 \\ b & \text{if} \quad |b| < |a| \quad and \quad ab > 0 \\ 0 & \text{if} \quad ab \leq 0 \end{cases} \\ &= \text{sign}(a) \max\{0, \min\{|a|, sign(a)b\}\} \end{aligned} \tag{3.11}$$

is the minmod limiter. The left and right variables $U_{i,j}$ and $U_{i+1,j}$ are replaced by the extrapolated variables $U^L_{i+\frac{1}{2},j}$ and $U^R_{i+\frac{1}{2},j}$ to calculate the flux $F_{i+\frac{1}{2},j}$ in (3.3) and (3.5). The MUSCL approach is applied similarly to the numerical fluxes $G_{i,j+\frac{1}{2}}$ in (3.4) and (3.6).

3.8 Spatial Discretization for Compressible Navier-Stokes Equations

The Navier-Stokes equations contain convective and viscous fluxes. The discretization of the convective fluxes is the same as for the 2D compressible Euler equations. In this section, we explain the discretization of viscous fluxes. The five point stencils is shown in Fig. 3.1. Let us consider only the normal component, for the sake of understanding, of the viscous stress which is given as

$$\tau_{xx} = \frac{2}{3}\mu(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}). \tag{3.12}$$

The discretization of first term on the right hand side of (3.12) is as follows

$$\frac{4}{3}\mu\frac{\partial u}{\partial x}|_{i-\frac{1}{2},j} \approx \frac{4}{3}\mu\frac{u_{i,j} - u_{i-1,j}}{\Delta x}, \tag{3.13}$$

similarly, we can discretize second term on the right hand side of (3.12) as

$$\frac{2}{3}\mu\frac{\partial v}{\partial y}|_{i-\frac{1}{2},j} \approx \frac{2}{3}\mu\frac{v_{i,j+1} - v_{i,j-1} + v_{i-1,j+1} - v_{i-1,j-1}}{4\Delta y}. \tag{3.14}$$

The discretization of the other terms involved in viscous fluxes can be discretized following the above path.

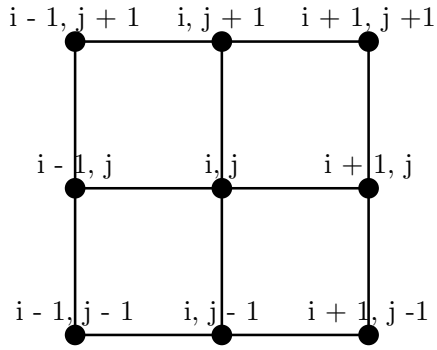


Figure 3.1: Nine point stencil for viscous flows.

3.9 Time Discretization

The spatial discretization above leads to a system of ordinary differential equations (ODEs)

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} = \mathbf{R}(\mathbf{U}), \tag{3.15}$$

with initial condition

$$\mathbf{U}(x, 0) = \mathbf{U_0}(x), \tag{3.16}$$

where $\mathbf{U}$ and $\mathbf{R}$ represent the grid functions of variable and residual, respectively.

Two main types of methods exist to discretize any partial differential equations (PDEs) in time. One is the explicit and another is the implicit method.

Since this thesis only deals with explicit methods because of their ease of application, we do not use implicit schemes. In this thesis, we discuss explicit first and third order time discretization schemes.

3.9.1 Explicit Euler Method

The time discretization with the explicit Euler method reads

$$\frac{\mathbf{U^{n+1}} - \mathbf{U^n}}{\Delta t} = \mathbf{R}(\mathbf{U^n}), \tag{3.17}$$

where $\mathbf{R}$ is the residual.

3.9.2 Explicit Third Order Runge-Kutta Method

The total variation diminishing third order Runge-Kutta method (TVD RK3) is defined as

$$\begin{aligned}
\mathbf{U}^{(1)} &= \mathbf{U}^n + \Delta t \mathbf{R}(\mathbf{U^n}), \\
\mathbf{U}^{(2)} &= \frac{3}{4}\mathbf{U}^n + \frac{1}{4}\mathbf{U}^{(1)} + \frac{1}{4}\Delta t \mathbf{R}(\mathbf{U^{(1)}}), \\
\mathbf{U}^{(n+1)} &= \frac{1}{3}\mathbf{U}^n + \frac{2}{3}\mathbf{U}^{(2)} + \frac{2}{3}\Delta t \mathbf{R}(\mathbf{U^{(2)}}),
\end{aligned} \tag{3.18}$$

where $\mathbf{R}$ is the residual.

The stability region of this method contains on the real axis $[-2.5, 0]$ and $[-1.732, 1.732]$ on the imaginary axis.

Chapter 4

Ghost Point Treatments

In this chapter, we present the treatment near the embedded boundary for the Cartesian grid method. The main focus of this chapter is to emphasize the ghost point treatments near the embedded boundary in 2D. The introduction of the ghost point treatment by Sjögreen and Petersson [1] is also presented in section 4.1. In the next step, the new approach called the simplified ghost point treatment for slender and blunt bodies is presented in sections 4.2 and 4.3, respectively. At the end, the ghost point treatment for a 1D problem is discussed in section 4.4.

4.1 Sjögreen and Petersson Ghost Point Treatment [1]

In this section, the description of the ghost point treatment presented in [1] is described. In Fig. 4.1, we show a 2D graphical description of the ghost point treatment at the embedded boundary [1]. The distance of the ghost point g from the wedge is denoted by b. The straight line through $g(i,j)$ normal to the wedge is intersecting the horizontal lines at three points denoted by the vertical lines. At the first intersection point I, the flow variables u_I are obtained by linear interpolation of the values at the neighboring horizontal grid points. And similarly, the variables u_{II} and u_{III} are obtained in [1]. The linear interpolation is defined as

$$u_I = w_1 u_{i-1,j+1} + (1 - w_1) u_{i,j+1}, \tag{4.1}$$

$$u_{II} = w_2 u_{i-2,j+2} + (1 - w_2) u_{i-1,j+2}, \tag{4.2}$$

$$u_{III} = w_3 u_{i-2,j+3} + (1 - w_3) u_{i-1,j+3}. \tag{4.3}$$

Where $(i-1, j+1)$, $(i, j+1)$ etc are neighboring points. w_1, w_2 and w_3 are the weights.

For flow variables u which have a Dirichlet boundary conditions $u = u_\Gamma$ at the embedded boundary Γ, i.e. at an impermeable wall $\underline{u}.\underline{n} = 0$ for the Euler equations and $u = 0 = v$ for the Navier-Stokes equations, the values $u_{i,j}$ at the ghost point (i,j) are imposed as follows. Points b_1 and b_2 (not shown in Fig. 4.1) on the normal through ghost point $\underline{x}_g = x_{i,j}$ are determined with

a distance Δ and 2Δ from Γ, respectively, where $\Delta = \left|\underline{x}_I - \underline{x}_{i,j}\right|$. The ghost point treatment then explained in [1] interpolates values along the normal.

Linear interpolation is applied for u_{b_1} and u_{b_2} as

$$u_{b_1} = \frac{b}{\Delta} u_{II} + (1 - \frac{b}{\Delta}) u_I, \tag{4.4}$$

$$u_{b_2} = \frac{b}{\Delta} u_{III} + (1 - \frac{b}{\Delta}) u_{II}. \tag{4.5}$$

Then, the minmod limiter S_{mm} is used to determine the boundary slope

$$S_\Gamma = S_{mm}(u_{b_1} - u_\Gamma, u_{b_2} - u_{b_1}). \tag{4.6}$$

The ghost point value is obtained as follows

$$u_{i,j} = u_\Gamma - \frac{b}{\Delta} S_\Gamma. \tag{4.7}$$

For flow variables u, which do not have physical boundary conditions, simple extrapolation with limited slope is used in [1]

$$u_{i,j} = u_I - S(u_{III} - u_{II}, u_{II} - u_I), \tag{4.8}$$

where S is a limiter, e.g. the minmod or another limiter.
Extrapolation (4.8) can give negative density or pressure. In that case, the following simple extrapolation is employed in [1]

$$u_{i,j} = u_I. \tag{4.9}$$

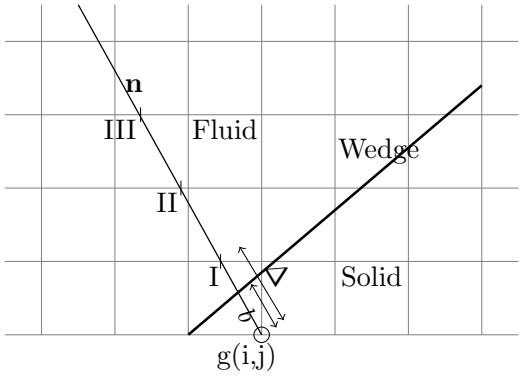


Figure 4.1: Ghost Point Treatment [1].

4.2 Simplified Ghost Point Treatment: A New Approach

In this section, we explain a new approach to determine the ghost point values near an embedded boundary for slender bodies. Let us consider the y-line and the normal which are passing through the ghost point G as it is shown in Fig. 4.2 and Fig. 4.3. The fluid point on the y-line closest to the embedded boundary is denoted by F. Hence, the y-line is passing through fluid and ghost points F and G in Fig. 4.2 and Fig. 4.3. Here, we make the assumption that the y-line can be treated as an approximation to the actual normal line. So rather than considering the normals to assign value for the ghost points, we regard the y-lines as the normal lines to assign values to the ghost points from fluid points. The other assumption we make here is that the wall boundary is in the middle between fluid point F and ghost point G. This assumption is usually not true. If it is true, then we get second order accuracy near the embedded boundary. In the case, when the wall boundary is not in the middle between fluid point F and ghost point G, we lose accuracy near the embedded boundary and get only a first order approximation of the ghost point value. The density ρ, pressure p, and the tangential velocity component at the ghost point are symmetric with respect to the solid boundary and therefore, directly determined by their values at F. The normal velocity component u_n at the ghost point is anti-symmetric and thus gets the negative value of u_n at F.

The following mathematical equations are used to determine the ghost point values

$$\rho_G = \rho_F,\ p_G = p_F,\ u_G = u_F - 2(n_1 u_F + n_2 v_F)n_1,\ v_G = v_F - 2(n_1 u_F + n_2 v_F)n_2, \tag{4.10}$$

where n_1 and n_2 are the x-and y-components of the outer unit normal $\mathbf{n}$.

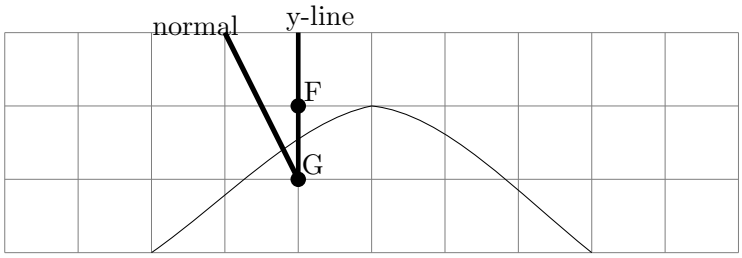


Figure 4.2: The idea of simplified ghost point treatment for slender bodies.

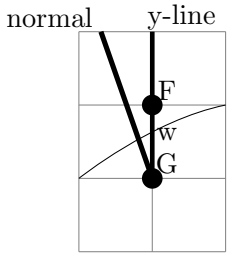


Figure 4.3: Closer look of the fluid and ghost points F and G, respectively, near the embedded boundary.

4.3 Simplified Ghost Point Treatment for Blunt Bodies

In order to apply the simplified ghost point treatment for blunt bodies, we need to modify the ghost point strategy of slender bodies. We consider fluid point F on the grid line in the x-direction, if the angle between the x-axis and the normal passing through the ghost point is less than 45 degrees in the first quadrant. If the angle between normal and x-axis is greater than 45 degrees, then we choose fluid point F on the grid line in the y-direction in the first quadrant. We employ the same strategy in the second, third and fourth quadrants of the Cartesian grid to choose fluid point F and ghost point G. We also assume that the embedded boundary lies in the middle of the fluid point F and the ghost point G. The density ρ, the pressure p, and the tangential velocity component at the ghost point are symmetric with respect to the solid boundary and therefore, directly determined by their values at F as for the slender bodies. The normal velocity component u_n at the ghost point is anti-symmetric and thus gets the negative value of u_n at F.

The mathematical description of this strategy reads

$$\rho_G = \rho_F, p_G = p_F, u_G = u_F - 2(n_1 u_F + n_2 v_F)n_1, v_G = v_F - 2(n_1 u_F + n_2 v_F)n_2, \tag{4.11}$$

where n_1 and n_2 are the x- and y-components of the outer unit normal $\mathbf{n}$ of the embedded boundary, i.e. $\mathbf{n} = \frac{\mathbf{x_G} - \mathbf{x_C}}{|\mathbf{x_G} - \mathbf{x_C}|}$, where $\mathbf{x_C}$ is the vector of the center of the cylinder.

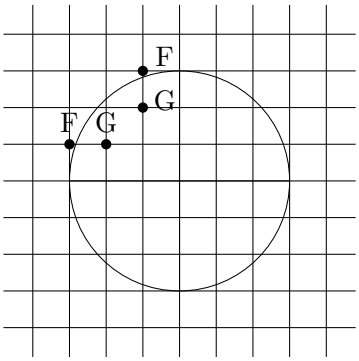


Figure 4.4: Simplified ghost point treatment for a cylinder.

4.4 Ghost Point Treatment for 1D Embedded Boundary

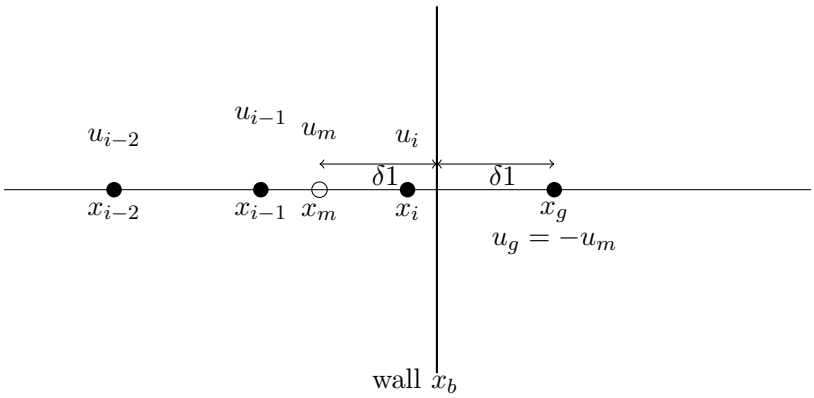


Figure 4.5: Ghost point treatment in 1D.

In 1D, the boundary ghost point treatment is shown in Fig. 4.5. In Fig. 4.5, the full black circles denote grid points. The embedded boundary is a wall located between fluid point x_i and ghost point x_g. The location of the wall is represented by x_b. The procedure to determine the flow variables at the ghost point in 1D is explained as follows:

Step 1: Find the distance between the ghost point and the wall

$$\delta_1 = x_g - x_b, \tag{4.12}$$

Step 2: Determine the location of the mirror point x_m in the fluid domain

$$x_m = x_b - \delta_1, \tag{4.13}$$

Step 3: Apply linear or quadratic interpolation to get the interpolated values. Linear interpolation yields

$$V_m = V_{i-1} + \frac{V_i - V_{i-1}}{\Delta x}(x_m - x_{i-1}), \tag{4.14}$$

where $V = (\rho, u, p)^T$ is the vector of the primitive variables for 1D compressible Euler equations. For quadratic interpolation, the values of V_i, V_{i-1} and V_{i-2} are used to determine V_m.

Step 4: Use reflective boundary conditions to obtain the ghost point values

$$u_g = -u_m, p_g = p_m, \rho_g = \rho_m \tag{4.15}$$

Chapter 5

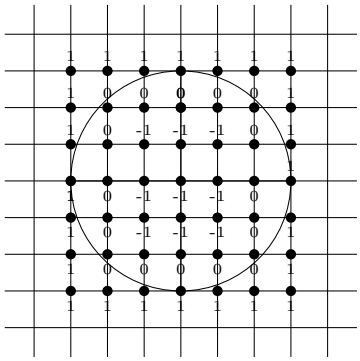
Two-Dimensional Compressible Euler Equations

First, we test the simplified ghost point treatment presented in Chapter 4 for a very simple problem i.e., an oblique shock wave generated by a wedge. In the next step, the simplified ghost point treatment is tested for a curved embedded boundary. We choose a circular arc airfoil for this purpose to test the ghost point treatment for slender bodies. In this chapter, we present new results for blunt bodies. The case under consideration is supersonic flow over a cylinder. We use the first order local Lax-Friedrichs method for spatial discretization. To increase the order of our method, we use MUSCL with the minmod limiter. For time integration, we use the first order explicit Euler and third order total variation diminishing Runge-Kutta (TVD RK3) methods. We compare our results of supersonic flow over a cylinder with results presented in the literature.

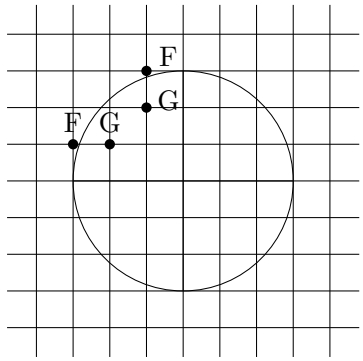
In section 5.1, the flagging strategy and the ghost point treatment for the supersonic flow over the cylinder is presented. Boundary conditions are given in section 5.2. Results are shown and discussed in section 5.3.

5.1 Flagging Strategy and Ghost Point Treatment for Cylinder

In Fig. 5.1(a), we sketch the flagging strategy for the simplified ghost point treatment. We flag fluid, ghost and solid points by assigning them 1, 0 and -1 values, respectively. In Fig. 5.1(b), we show two examples of the choice of the fluid and ghost points as F and G, respectively, in the second quadrant. In the simplified ghost point treatment, we consider the fluid point F on the horizontal or the vertical line through G adjacent to the boundary as the mirror point depending on the angle between the normal passing through the ghost point and the x-axis.



(a) Flagging strategy for cylinder.



(b) The simplified ghost point treatment for cylinder.

Figure 5.1: Left: Flagging strategy for cylinder. Right: The simplified ghost point treatment for cylinder.

5.2 Boundary Conditions

For supersonic flow in the x-direction for blunt bodies, the conservative variables at inflow are given as Dirichlet boundary conditions, cf. Fig. 5.2. No boundary condition must be given at outflow, because the flow is supersonic.

Extrapolation boundary conditions are given at the lower boundary

$$\frac{\partial U}{\partial y}|_{lower\ boundary} = 0$$

Similarly, extrapolation boundary conditions are also assumed at the upper boundary

$$\frac{\partial U}{\partial y}|_{upper\ boundary} = 0$$

5.2.1 Approximation of Boundary Conditions

The inflow boundary conditions for supersonic flow , cf. Fig. 5.2, are imposed as

$$U_{1,j}(t) = U_\infty, \tag{5.1}$$

where U_∞ is the vector of the conservative variables for uniform flow in the $x-$direction. The flow variables at the outflow are approximated as

$$U_{I,j}(t) = U_{I-1,j}(t), \tag{5.2}$$

i.e. by constant extrapolation. This approximation implies that the upwind finite volume method is used to determine the numerical fluxes $F_{I-\frac{1}{2},j}$.

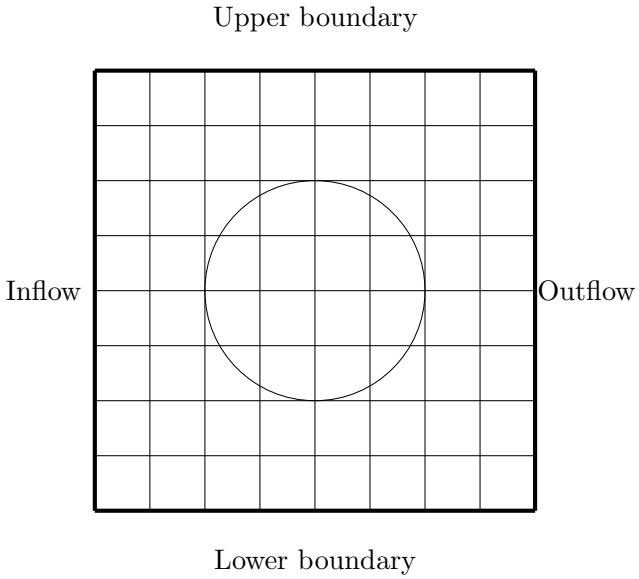


Figure 5.2: Boundaries for flow over blunt bodies.

The extrapolation boundary conditions are implemented at the lower boundary

$$U_{i,1}(t) = U_{i,2}(t). \tag{5.3}$$

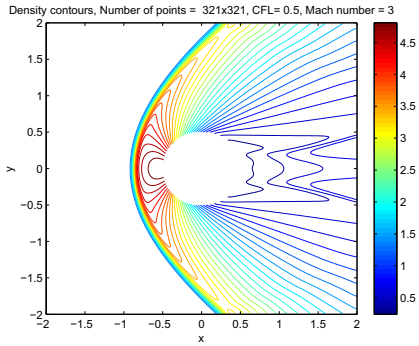
The boundary conditions at the upper boundary are treated as

$$U_{i,J}(t) = U_{i,J-1}(t). \tag{5.4}$$

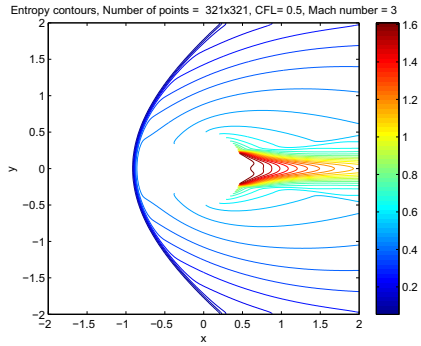
5.3 Results

We consider a supersonic flow past a cylinder of radius 0.5 m as a test case for the simplified ghost point treatment. We use the CFL number 0.5, $n = 80000$ time steps for the first order method, $n = 50000$ time steps for the second order method and 321×321 grid points. For spatial discretization we use the first order local Lax-Friedrichs method. To increase the order of the method we use MUSCL with the minmod limiter. For time integration we use the explicit Euler method for the first order local Lax-Friedrichs scheme and the third order TVD RK3 method for the local Lax-Friedrichs scheme with MUSCL and the minmod limiter. The lengths of the domain in x-and y-directions are $L_x = 4$ and $L_y = 4$, respectively. The supersonic upstream flow conditions are

$$M_\infty = 3, p_\infty = 10^5 \text{ Pa}, \rho_\infty = 1.2 \frac{\text{kg}}{\text{m}^3} \tag{5.5}$$

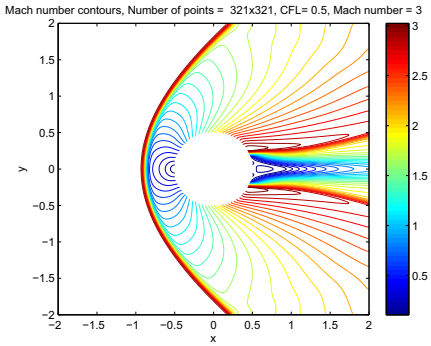


(a) Density contours for a supersonic flow over a cylinder with $M_\infty = 3$.

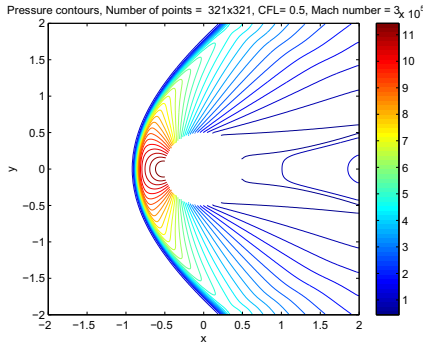


(b) Entropy contours for a supersonic flow over a cylinder with $M_\infty = 3$.

Figure 5.3: Left: Density ρ contours for the first order method. Right: Entropy contours for the first order method. (321 $\times$ 321 grid)



(a) Mach number contours for supersonic flow over a cylinder $M_\infty = 3$.



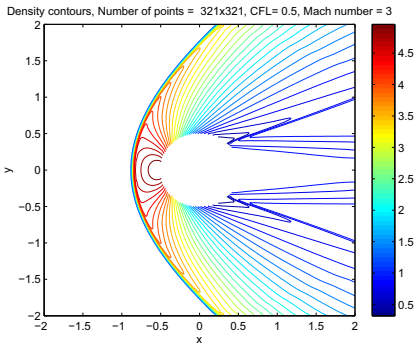
(b) Pressure p contours for supersonic flow over a cylinder $M_\infty = 3$.

Figure 5.4: Left: Mach number contours for the first order method. Right: Pressure p contours for the first order method. (321 $\times$ 321 grid)

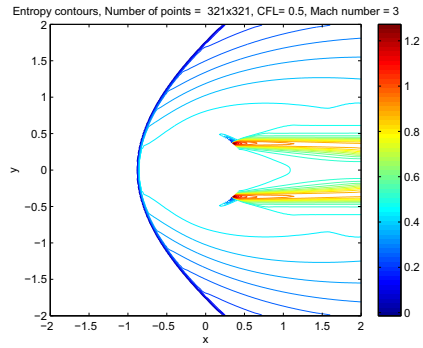
In Figs. 5.3(a) and 5.3(b), we show the density ρ and entropy contours for the supersonic flow past a cylinder. When the supersonic flow hits the front part of the cylinder, a shock wave is produced that is called a bow shock. The bow shock intersects with the upper and lower boundaries of the domain. The entropy remains constant along streamlines for steady compressible flow, except for shocks. The entropy is defined by $\frac{s}{c_v} = \ln(\frac{p}{p_\infty}/(\frac{\rho}{\rho_\infty})^\gamma)$. The flow

behind the cylinder does not have any physical significance, because the flow separation depends on viscous effects that are not taken into account here.

Figs. 5.4(a) and 5.4(b) show the Mach number and pressure contours. The Mach number is uniform upstream of the bow shock. The flow downstream of the bow shock and in front of the cylinder becomes subsonic. When the flow passes the front part of the cylinder, then the velocity of the fluid keeps increasing from the stagnation point at $x = -0.5\ m$ and it becomes supersonic at $x \approx -0.34\ m$ until the outflow boundary except for parts in the unphysical wake.

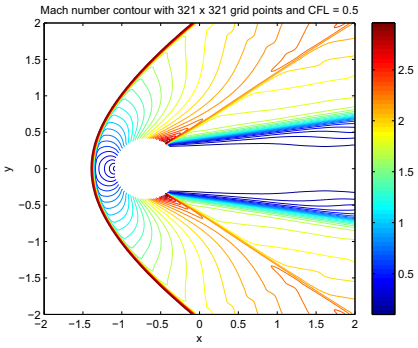


(a) Density contours for the supersonic flow over a cylinder with $M_\infty = 3$.

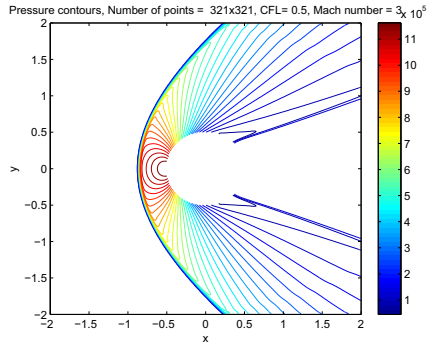


(b) Entropy contours for the supersonic flow over a cylinder with $M_\infty = 3$.

Figure 5.5: Left: Density ρ contours for the second order method. Right: Entropy contours for the second order method. (321 $\times$ 321 grid)

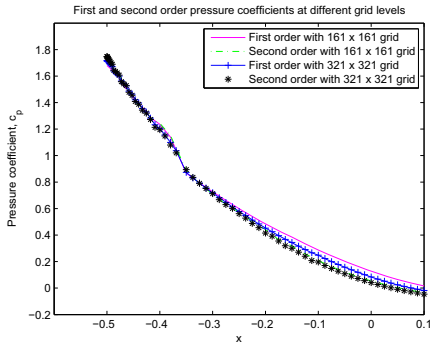


(a) Mach number contours for the supersonic flow over a cylinder $M_\infty = 3$.

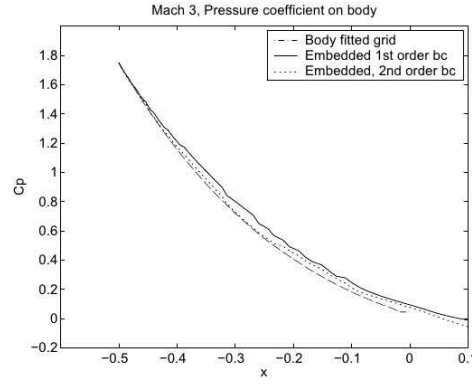


(b) Pressure p contours for the supersonic flow over a cylinder $M_\infty = 3$.

Figure 5.6: Left: Mach number contours for the second order method. Right: Pressure p contours for the second order method. (321 $\times$ 321 grid)



(a) Pressure coefficient of the Cartesian grid method with the simplified ghost point treatment.



(b) Pressure coefficient results from the literature [1].

Figure 5.7: Left: Pressure coefficient for the first and second order methods (Present results with 161 $\times$ 161 and 321 $\times$ 321 grids) Right: Pressure coefficients results from [1]. (305 $\times$ 305 grid)

In Figs. 5.5(a) and 5.5(b), we show the density ρ and entropy contours for the supersonic flow past a cylinder for the second order method. As before, when the supersonic flow hits the front part of the cylinder a bow shock is produced. This shock wave is now sharper resolved than by the first order method. As in the first order results, the flow computed behind the cylinder does not have any physical significance, because the flow separation depends on viscous effects that are not taken into account to produce these results.

Figs. 5.6(a) and 5.6(b) show the Mach number and pressure contours for the second order method. The bow shock in front of the cylinder is also quite sharp compared to the first order results for the Mach numbers and pressure contours. While density, entropy, Mach number and pressure contours between bow shock and cylinder are quite similar computed with the first and second order methods, large differences can be seen in the unphysical wake computed with the compressible Euler equations, cf. Figs. 5.3 - 5.6.

Finally, we show the first and second order pressure coefficient results in Fig. 5.7(a) on different grid levels. We draw a comparison of the pressure coefficient of the simplified ghost point treatment and the results of [1] in Figs. 5.7(a) and 5.7(b). We can see a good agreement of the simplified ghost point treatment and the results in [1]. In the present results, we see a kink where we change direction from x- to y-axis for the simplified ghost point treatment. It is pertinent to note that we observe our 321 $\times$ 321 grid results to compare well with the literature. This shows that we can also achieve similar results even using our simplified ghost point treatment rather than the more involved ghost point treatment in [1].

Chapter 6

Two-Dimensional Compressible Navier-Stokes Equations

This chapter is a continuation of a test cases used to verify the new approach called the simplified ghost point treatment. In this chapter, we include the results for the 2D compressible Navier-Stokes equations. Supersonic flow over two cylinders has been calculated. This shows the wide range of applicability of the simplified ghost point treatment. The convective flux is discretized as for the 2D compressible Euler equations. The viscous terms are discretized with the central difference scheme. We use second order spatial discretization. For time integration we use the third order TVD RK3 method.

In section 6.1, the flagging strategy and the simplified ghost point treatment is the same except a slight change in getting the ghost point values of velocities u and v. The simplified ghost point treatment with modification in mathematical form is repeated here to make it understandable and to present the change made in the simplified ghost point treatment for the 2D compressible Navier-Stokes equations. Boundary conditions are given in section 6.2. Results and discussions are given in section 6.3.

6.1 Flagging Strategy and Ghost Point Treatment for cylinder

In this section, we repeat the flagging strategy for blunt body, i.e. for cylinder. We present this flagging strategy for one cylinder immersed in the computational domain but the results presented contain supersonic flow past two cylinders.

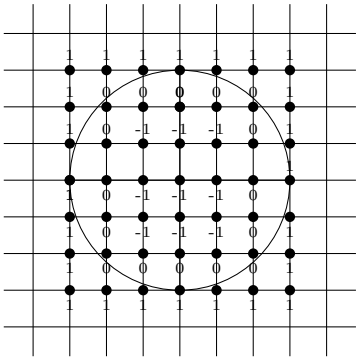
In Fig. 6.1(a), we sketch the flagging strategy for the simplified ghost point treatment for 2D compressible Navier-Stokes equations. We flag fluid, ghost and solid points by assigning them 1, 0 and -1 values, respectively as discussed in Chapter 4. In Fig. 6.1(b), we show two examples of the choice of the fluid and ghost points F and G, respectively, in the second quadrant. In the simplified ghost point treatment, we consider the fluid point F on the horizontal or the vertical line through G adjacent to the boundary as the mirror point depending on the angle between the normal passing through the

ghost point and the x-axis. We consider fluid point F on the grid line in the x-direction if the angle between the x-axis and the normal passing through ghost point is less than 45 degrees in the first quadrant. If the angle between normal and x-axis is greater than 45 degrees then we choose fluid point F on the grid line in the y-direction in the first quadrant. We employ the same strategy in the second, third and fourth quadrants of the Cartesian grid to choose fluid points F and ghost points G. We also assume that the embedded boundary lies between fluid point F and ghost point G. The density ρ and the pressure p are symmetric with respect to the solid boundary and are therefore directly determined by their values at F as before for the 2D compressible Euler equations. Both the tangential and normal velocity components u_t and u_n at the ghost point for 2D compressible Navier-Stokes equations are anti-symmetric and thus get the negative values of u_t and u_n at F. The symmetry conditions for ρ and p implies that the embedded boundary is assumed to be an adiabatic wall, i.e. $\frac{\partial T}{\partial n} = 0$, and that viscous effects on the normal pressure gradient are neglected, i.e. $\frac{\partial p}{\partial n} = 0$ is assumed.

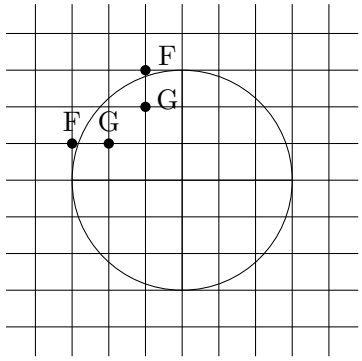
The mathematical description of this strategy is given as

$$\rho_G = \rho_F,\ p_G = p_F,\ u_G = -u_F,\ v_G = -v_F, \tag{6.1}$$

where n_1 and n_2 are the x-and y-components of the outer unit normal $\mathbf{n}$ of the boundary, i.e. $\mathbf{n} = \frac{\mathbf{x_G} - \mathbf{x_C}}{|\mathbf{x_G} - \mathbf{x_C}|}$, where $\mathbf{x_C}$ is the vector of the centre of the cylinder.



(a) Flagging strategy for cylinder.



(b) The simplified ghost point treatment for cylinder.

Figure 6.1: Left: Flagging strategy for cylinder. Right: The simplified ghost point treatment for cylinder.

6.2 Boundary Conditions

For supersonic flow in the x-direction for blunt bodies, the conservative variables at the inflow are given as Dirichlet boundary conditions as for the 2D compressible Euler equations, cf. Fig. 6.2. At the embedded boundary, no-slip boundary conditions are assumed $u = 0 = v$ and an adiabatic wall $\frac{\partial T}{\partial n} = 0$. The normal pressure gradient is approximated by $\frac{\partial p}{\partial n} = 0$.

Extrapolation boundary conditions are given at lower boundary

$$\frac{\partial U}{\partial y}|_{lower\ boundary} = 0$$

Similarly, extrapolation boundary conditions are also assumed at upper boundary

$$\frac{\partial U}{\partial y}|_{upper\ boundary} = 0$$

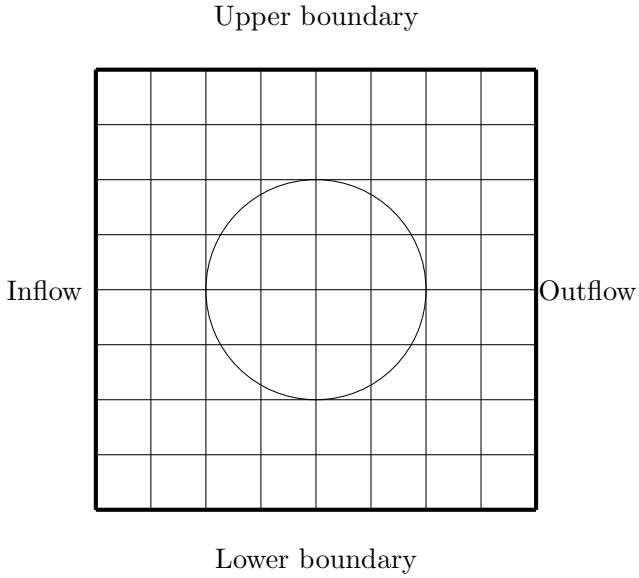


Figure 6.2: Boundaries for flow over blunt bodies.

6.2.1 Approximation of Boundary Conditions

The inflow boundary conditions, cf. Fig. 6.2, are imposed as

$$U_{1,j}(t) = U_\infty, \tag{6.2}$$

where U_∞ is the vector of the conservative variables for uniform flow in the $x-$direction. The flow variables at the outflow are approximated as

$$U_{I,j}(t) = U_{I-1,j}(t), \tag{6.3}$$

i.e. by constant extrapolation.

The extrapolation boundary conditions are implemented at the lower boundary

$$U_{i,1}(t) = U_{i,2}(t). \tag{6.4}$$

The boundary conditions at the upper boundary are treated as

$$U_{i,J}(t) = U_{i,J-1}(t). \tag{6.5}$$

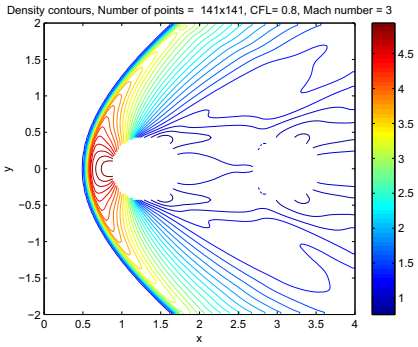
6.3 Results

We consider supersonic flow past two cylinders as a test case for the simplified ghost point treatment. We use the CFL number 0.8 and 141×141 grid points. We use 6000 time steps to calculate the supersonic flow over two cylinders. We use the second order method (except for extrema) for spatial discretization. For time integration we use the third order TVD RK3 method. The lengths of the domain in x-and y-directions are $L_x = 4$ and $L_y = 4$, respectively. The supersonic upstream flow conditions are

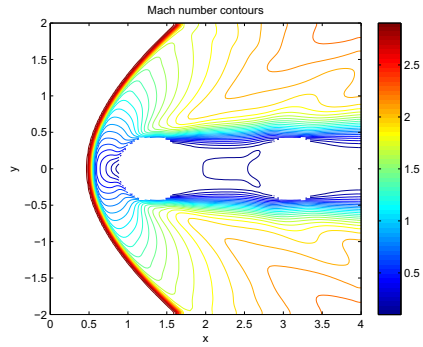
$$M_\infty = 3, p_\infty = 10^5 \text{ Pa}, \rho_\infty = 1.207 \ \frac{\text{kg}}{\text{m}^3}, Pr = 0.72, Re_\infty = 500 \tag{6.6}$$

In Figs. 6.3(a) and 6.3(b), we show the density ρ and Mach number M contours for the supersonic flow past two cylinders. When the supersonic flow hits the front part of the cylinder a bow shock is produced, which intersects with the upper and lower boundaries of the domain. The subsonic wake behind the first cylinder strikes the second cylinder.

Figs. 6.4(a) and 6.4(b) show the velocities u and v contours. In both of these Figs. 6.4(a) and 6.4(b) the wake behind the first cylinder hits with the second cylinder in downstream with low velocity components. We can only see one bow shock in front of the first cylinder. The wake behind the cylinder is physical due to the viscous flow simulation.

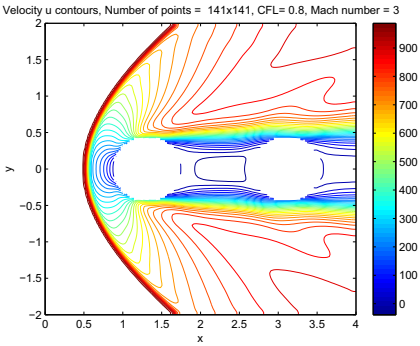


(a) Density contours for supersonic flow over two cylinders with $M_\infty = 3$, $Re_\infty = 500$.

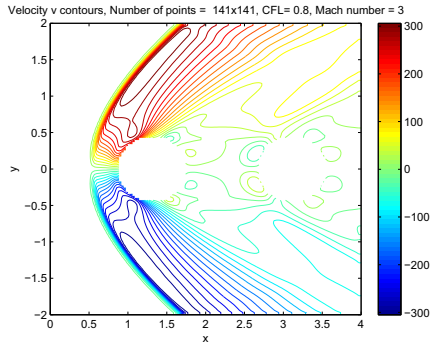


(b) Mach number M contours for supersonic flow over two cylinders with $M_\infty = 3$, $Re_\infty = 500$.

Figure 6.3: Left: Density ρ contours for second order method. Right: Mach number M contours for second order method. (141 $\times$ 141 grid)



(a) Velocity u contours for supersonic flow over two cylinders with $M_\infty = 3$, $Re_\infty = 500$.



(b) Velocity v contours for supersonic flow over two cylinders with $M_\infty = 3$, $Re_\infty = 500$.

Figure 6.4: Left: Velocity u contours for second order method. Right: Velocity v contours for second order method. (141 $\times$ 141 grid)

Chapter 7

Conclusions and Outlook

The main aim of this research work was to find an alternative method to treat the ghost point treatment for the Cartesian grid method. The ghost point treatments presented in the literature are quite involved and difficult to apply, as one example of such a ghost point treatment given in Chapter 4 [1] shows. We introduced a new approach called the simplified ghost point treatment to treat ghost points near the embedded boundary. This simplified ghost point treatment is very easy to apply and to implement in computer programs. The extension from first order to second order is quite easy: we just needed one additional ghost point for second order. The main advantage of the simplified ghost point treatment presented in this thesis work is that no interpolation is needed near embedded boundaries.

The ghost point treatment presented in [1] might be more accurate than the simplified ghost point treatment. But, the main concern, as mentioned above, is the procedure of the ghost point treatment mentioned in [1]. It is a trade off whether to use a more accurate ghost point treatment or use a simple procedure to obtain ghost point values. The common weakness of both of these ghost point treatments is the lack of conservation property near embedded boundaries.

The simplified ghost point treatment is based on two assumptions

1. The x- or y-line passing through fluid and ghost point can be treated as an approximation of the actual normal at the embedded boundary.

2. The wall boundary is in the middle between fluid and ghost points.

As far as the first assumption is concerned, this is a valid justification. This assumption is justified quite well as we tested the simplified ghost point treatment for simple geometries. In fact, the first assumption has made this treatment very simple. The second assumption is not always true, because the embedded boundary cuts the grid lines arbitrarily. If the wall boundary is in the middle of fluid and ghost points then the ghost point values are second order, otherwise they are first order.

We have also learned from this research work that we need a larger number of grid points for the Cartesian grid method compared to the body-fitted grid method. This might be one disadvantage of using the Cartesian grid method.

We investigated the accuracy of the Cartesian grid method for the 1D and 2D compressible Euler equations as well as for the 2D compressible Navier-Stokes equations. We used first and second order methods for spatial discretization. We applied the first order explicit Euler method and the third order total variation diminishing Runge-Kutta method (TVD RK3) for time integration. Local symmetry boundary conditions have been applied near the embedded boundary. For 1D inviscid flows, we tested the code for the 1D Burgers' equation and 1D compressible Euler equations. We verified the ghost point treatment for a moving normal shock wave. To apply the Cartesian grid method for 2D problems, we introduced a new approach called the simplified ghost point treatment near the embedded boundary. We first tested this new approach for a very simple test case, i.e. for supersonic flow over a wedge generating an oblique shock wave. Then, we moved to slender bodies and tested this approach for supersonic and subsonic flow over a circular arc airfoil and a circular arc bump. In order to increase the validity and applicability of the simplified ghost point treatment for blunt bodies, we chose a supersonic flow over a cylinder for the 2D compressible Euler equations. We needed to change the strategy of applying the simplified ghost point treatment for this case as it is explained in Chapter 4. We successfully applied and tested the simplified ghost point treatment for simple to more complex problems.

In the end of this research work, we tested the simplified ghost point treatment for the 2D viscous flows. We chose a supersonic flow over two cylinders for the 2D compressible Navier-Stokes equations. The simplified ghost point treatment is modified near the embedded boundary for the ghost point treatment in order to implement the no-slip condition near the wall boundary.

The simplified ghost point treatment can be further improved by taking curvature of the embedded boundary into account. In a current method, the ghost point values of density and pressure have been directly taken from fluid points without considering the curvature of the embedded boundary. If we consider the curvature of the embedded boundary, then we can expect an improvement of this method.

The other improvement in this method can be done for supersonic flow over a cylinder. Since we need to modify the simplified ghost point treatment for this particular case. In the supersonic flow over a cylinder, we need to change the ghost point treatment strategy if the angle of normal is less or greater than 45 degrees. In that case, where we change the direction of the ghost point treatment, we observe the kink in the pressure coefficient for a cylinder as it is shown in results of Fig. 5.7(a), Chapter 5. The simplified ghost point treatment can be improved where we make a transition from horizontal direction to vertical direction. One of the options is to further extend the line with a slope of 45 degrees and consider to include neighboring fluid points.

In future, the accuracy of the transient problem can be checked. The motion of moving objects can be calculated. Local grid refinement is also possible.

The gridless method can be used near the embedded boundary for accurate solutions. The extension to 3D is straightforward. Parallelization of the code is also easily possible.

Bibliography

[1] B. Sjögreen and N. A. Petersson, *A Cartesian embedded boundary method for hyperbolic conservation laws*, *Communications in Computational Physics* **2** (2007), no. 6 1199 – 1219.

[2] J. Blazek, *Computational fluid dynamics : principles and applications*. 2nd ed., Amsterdam : Elsevier, 2007.

[3] D. L. George and R. J. LeVeque, *High-resolution methods and adaptive refinement for tsunami propagation and inundation*, in *Hyperbolic Problems: Theory, Numerics, Applications* (S. Benzoni-Gavage and D. Serre, eds.), pp. 541–549. Springer Berlin Heidelberg, 2008.

[4] D. L. George and R. J. LeVeque, *Finite volume methods and adaptive refinement for global tsunami propagation and local inundation*, *Science of Tsunami Hazards* **24** (2006) 319–328.

[5] A. Jameson, W. Schmidt, and E. Turkel, *Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes*, *AIAA Paper 81-1259* (1981).

[6] E. Dick, *A flux-difference splitting method for steady Euler equations*, *Journal of Computational Physics* **76** (1988) 19–32.

[7] V. Venkatakrishnan and D. Mavriplis, *Implicit method for the computation of unsteady flows on unstructured grids*, *Journal of Computational Physics* **127** (1996) 380–397.

[8] M. Berger and R. LeVeque, *An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries*, *AIAA Paper 89-1930-CP* (1989).

[9] D. Calhoun and R. J. LeVeque, *A Cartesian grid finite-volume method for the advection-diffusion equation in irregular geometries*, *Journal of Computational Physics* **157** (2000), no. 1 143–180.

[10] D. Clarke, M. Salas, and H. Hassan, *Euler calculations for multielement airfoils using Cartesian grids*, *AIAA Journal* **24** (1986) 353–358.

[11] W. J. Coirier and K. G. Powell, *An accuracy assessment of Cartesian-mesh approaches for the Euler equations*, *Journal of Computational Physics* **117** (1995), no. 1 121–131.

[12] A. S. Almgren, J. B. Bell, P. Colella, and T. Marthaler, *A Cartesian grid projection method for the incompressible Euler equations in complex geometries*, *SIAM Journal on Scientific Computing* **18** (1997), no. 5 1289–1309.

[13] C. S. Peskin, *Flow pattern around heart valves: A numerical method*, *Journal of Computational Physics* **10** (1972) 252–271.

[14] D. Hartmann, M. Meinke, and W. Schröder, *A strictly conservative Cartesian cut-cell method for compressible viscous flows on adaptive grids*, *Computer Methods in Applied Mechanics and Engineering* **200** (2011) 1038–1052.

[15] C. H. Marsha J. Berger and R. J. Leveque, *h-box methods for the approximation of hyperbolic conservation laws on irregular grids*, *SIAM Journal on Numerical Analysis* **41** (2003) 893–918.

[16] L. S. Kim, K. Nakahashi, H. K. Jeong, and M. Y. Ha, *High density mesh flow computations by building cube method*, *Journal of Mechanical Science and Technology* **21** (2007) 1306–1319.

[17] H. Forrer and R. Jeltsch, *A higher-order boundary treatment for Cartesian-grid methods*, *Journal of Computational Physics* **140** (1998), no. 2 259–277.

[18] J. Burgers, *A mathematical model illustrating the theory of turbulence*, *Advances in Applied Mechanics* **1** (1948) 171–199.

[19] E. Hopf, *The partial differential equation* $u_t + uu_x = \mu\ u_{xx}$, *Communications on Pure and Applied Mathematics* **3** (1950), no. 3 201–230.

[20] J. D. Anderson, *Fundamentals of aerodynamics*. Boston, Mass. : McGraw-Hill, 2007.

[21] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics : A Practical Introduction*. Springer-Verlag Berlin Heidelberg, 2009.

[22] J. C. Tannehill, D. A. Anderson, and R. H. Pletcher, *Computational fluid mechanics and heat transfer*. Washington : Hemisphere Publ., 2nd ed., 1997.

[23] C. Hirsch, *Numerical computation of internal and external flows : fundamentals of computational fluid dynamics*. Elsevier/Butterworth-Heinemann, 2007.

[24] H.-O. Kreiss and N. A. Petersson, *A second order accurate embedded boundary method for the wave equation with Dirichlet data*, *SIAM Journal on Scientific Computing* **27** (2006) 1141–1167.

[25] S. Gottlieb and C. Shu, *Total variation diminishing Runge-Kutta schemes*, *Mathematics of Computations* **67** (1998) 73–85.

[26] D. G. J. Burns, A. Balogh and V. I. Shubov, *Numerical stationary solutions for a viscous Burgers' equation*, *Journal of Mathematical Systems, Estimation, and Control* **8** (1998) 1–16.

[27] P. Colella, D. T. Graves, B. J. Keen, and D. Modiano, *A Cartesian grid embedded boundary method for hyperbolic conservation laws*, *Journal of Computational Physics* **211** (2006), no. 1 347–366.

[28] R. Crockett, P. Colella, and D. Graves, *A Cartesian grid embedded boundary method for solving the Poisson and heat equations with discontinuous coefficients in three dimensions*, *Journal of Computational Physics* **230** (2011) 2451–2469.

[29] A. Dadone and B. Grossman, *Surface boundary conditions for the numerical solution of the Euler equations*, *AIAA Journal* **32** (1994) 285–293.

[30] A. Dadone and B. Grossman, *Ghost cell method for inviscid two dimensional flows on Cartesian grids*, *AIAA Journal* **42** (2004), no. 12 2499–2507.

[31] A. Dadone and B. Grossman, *Ghost-cell method for analysis of inviscid three-dimensional flows on Cartesian-grids*, *Computers & Fluids* **36** (2007) 1513–1528.

[32] L. A. Catalano, A. Dadone, V. S. E. Daloiso, and D. Scardigno, *A multigrid procedure for Cartesian ghost cell method*, *International Journal for Numerical Methods in Fluids* **58** (2008) 743–750.

[33] D. Appelö and N. A. Petersson, *A stable finite difference method for the elastic wave equation on complex geometries with free surfaces*, *Communications in Computational Physics* **5** (2009), no. 1 84–107.

[34] R. Donat and A. Marquina, *Capturing shock reflection: An improved flux formula*, *Journal of Computational Physics* **125** (1996) 1–33.

[35] B. Einfeldt, C. D. Munz, P. L. Roe, and B. Sjögreen, *On Godunov-type methods near low densities*, *Journal of Computational Physics* **92** (1991) 273–295.

[36] E. Uzgarn, J. Sim, and W. Shyy, *Marker based 3d adaptive Cartesian grid method for multiphase flow around irregular geometries.*, *Communications in Computational Physics* **5** (2009), no. 1 1–41.

[37] E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yousof, *Combined immersed-boundary finite difference methods for three dimensional complex flow simulations*, *Journal of Computational Physics* **161** (2000) 30–60.

[38] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher, *A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method)*, *Journal of Computational Physics* **152** (1999), no. 2 457–492.

[39] M. Feistauer and V. Kucêra, *On a robust discontinuous Galerkin technique for the solution of compressible flow*, *Journal of Computaional Physics* **224** (2007) 208–221.

[40] Y.-H. Tseng and J. H. Ferziger, *A ghost cell immersed boundary method for flow in complex geometry*, *Journal of Computational Physics* **192** (2003), no. 2 593–623.

[41] D. M. Greaves and A. G. Borthwick, *On the use of adaptive hierarchical meshes for numerical simulation of separated flows*, *International Journal for Numerical Methods in Fluids* **26** (1998) 303–322.

[42] M. D. Griffin and J. D. Anderson, *On the application of boundary conditions to time dependent computaions for quasi-one dimensional fluid flows*, *Computers & Fluids* **5** (1977) 127–137.

[43] B. Grüber and V. Carstens, *The impact of viscous effects on the aerodynamic damping of vibrating transonic compressor blades - A numerical study*, *Journal of Turbomachinery* **123** (2001) 409.

[44] A. Harten, *High resolution schemes for hyperbolic conservation laws*, *Journal of Computational Physics* **49** (1983) 357–393.

[45] A. Harten, *On a class of high resolution total variation stable difference schemes*, *Journal of Numerical Analysis* **21** (1984) 1–23.

[46] Y. Hassen and B. Koren, *Finite-volume discretizations and immersed boundaries*, *in: Lect. Notes Comput. Sci. Eng .71, Springer, Heidelberg,* (2010) 229–268.

[47] H. Luo, J. D. Baum, and R. Löhner, *A hybrid building-block and gridless method for compressible flows*, *International Journal for Numerical Methods in Fluids* **59** (2008) 459–474.

[48] M. D. de Tullio, P. D. Palma, G. Iaccarino, G. Pascazio, and M. Napolitano, *An immersed boundary method for compressible flows using local grid refinement*, *Journal of Computational Physics* **225** (2007) 2098–2117.

[49] D. Ingram, D. Causon, and C. Mingham, *Developments in Cartesian cut cell methods*, *Mathematics and Computers in Simulation* **61** (2003) 561–572.

[50] H. Ji, F.-S. Lien, and E. Yee, *A robust and efficient hybrid cut-cell/ghost-cell method with adaptive mesh refinement for moving boundaries on irregular domains*, *Computer Methods in Applied Mechanics and Engineering* **198** (2008) 432–448.

[51] J. Revstedt and L. Fuchs, *Handling complex boundaries on a Cartesian grid using surace singularities*, *International Journal for Numerical Methods in Fluids* **35** (2001) 125–150.

[52] K. Karagiozis, R. Kamakoti, and C. Pantano, *A low numerical dissipation immersed interface method for the compressible Navier-Stokes equations*, *Journal of Computational Physics* **229** (2010) 701–727.

[53] D. J. Kirshman and F. Liu, *A gridless boundary condition method for the solution of the Euler equations on embedded Cartesian meshes with multigrid*, *Journal of Computational Physics* **201** (2004) 119–147.

[54] H.-O. Kreiss, N. A. Petersson, and J. Yström, *Difference approximations for the second order wave equation*, *SIAM Journal on Numerical Analysis* **40** (2002) 1940–1967.

[55] D. Kirshman and F. Liu, *Cartesian grid solution of the Euler equations using a gridless boundary condition treatment*, *AIAA Paper 2003-3974* (2003).

[56] M. J. Berger and R. J. LeVeque, *A rotated difference scheme for Cartesian grids in complex geometries*, *AIAA Paper CP-91-1602* (1991) 1–9.

[57] J. Liu, N. Zhao, and O. Hu, *The ghost cell method and its applications for inviscid compressible flow on adaptive tree Cartesian grids*, *Advances in Applied Mathematics and Mechanics* **1** (2009) 664–682.

[58] R. Löhner, *Applied computational fluid dynamics techniques : An introduction based on finite element methods.* 2nd edition, John Wiley & Sons, Chichester, 2008.

[59] H. Luo, J. D. Baum, and R. Löhner, *A hybrid Cartesian grid and gridless method for compressible flow*, *Journal of Computational Physics* **214** (2006), no. 2 618–632.

[60] H. Luo, R. Mittal, X. Zheng, S. A. Bielamowicz, R. J. Walsh, and J. K. Hahn, *An immersed-boundary method for flow-structure interaction in biological systems with application to phonation*, *Journal of Computational Physics* **227** (2008) 9303–9332.

[61] D. D. Marshall and S. M. Ruffin, *A new inviscid wall boundary condition treatment for boundary Cartesian grid method*, *AIAA 2004-0583 42nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada* (2004).

[62] R. Mittal and G. Iaccarino, *Immersed boundary methods*, *Annual Review of Fluid Mechanics* **37** (2005), no. 1 239–261.

[63] R. Ghias, R. Mittal, and H. Dong, *A sharp interface immersed boundary method for compressible viscous flows*, *Journal of Computational Physics* **225** (2007) 528–553.

[64] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, and A. von Loebbecke, *A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries*, *Journal of Computational Physics* **227** (2008), no. 10 4825–4852.

[65] B. Müller, *Upwind relaxation method for hypersonic flow simulation*, *Technical Report DLR-FB* (1991) 91–36, DLR, Göttingen, Germany.

[66] K. Nakahashi, D. Sharov, S. Kano, and M. Kodera, *Applications of unstructured hybrid grid method to high-Reynolds number viscous flows*, *International Journal for Numerical Methods in Fluids* **31** (1999) 97–111.

[67] K. Nakahashi, *Current capability of unstructured grid CFD and a consideration for the next step*, *High performance computing on vector systems (book)* **1** (2008) 1–11.

[68] W. F. Noh, *Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux*, *Journal of Computational Physics* **72** (1987) 78–120.

[69] O. K. Olsen, *Embedded boundary method for Navier-Stokes equations*, Master's thesis, Royal Institute of Technology, 2005.

[70] P. D. Palma, M. D. de Tullio, G. Pascazio, and M. Napolitano, *An immersed-boundary method for compressible viscous flows*, *Computers & Fluids* **35** (2006) 693–702.

[71] P. J. S. A. F. de Sousa, J. C. F. Pereira, and J. J. Allen, *Two-dimensional compact finite difference immersed boundary method*, *International Journal for Numerical Methods in Fluids* **65** (2011) 609–624.

[72] R. B. Pember, J. B. Bell, P. Colella, W. Y. Curtchfield, and M. L. Welcome, *An adaptive Cartesian grid method for unsteady compressible flow in irregular regions*, *Journal of Computational Physics* **120** (1995), no. 2 278–304.

[73] Y.-F. Peng, R. Mittal, A. Sau, and R. R. Hwang, *Nested Cartesian grid method in incompressible viscous fluid flow*, *Journal of Computational Physics* **229** (2010) 7072–7101.

[74] J. J. Quirk, *A contribution to the great Riamann solver debate*, *International Journal for Numerical Methods in Fluids* **18** (1994) 555–574.

[75] J. J. Quirk, *An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies*, *Computers & Fluids* **23** (January, 1994) 125–142.

[76] A. Rizzi, *Damped Euler-equation method to compute transonic flow around wing-body combinations*, *AIAA Journal* **20** (1982), no. 10 1321–1328.

[77] P. L. Roe, *Approximate Riemann solves, parameter vectors, and difference schemes*, *Journal of Computational Physics* **135** (1997) 250–258.

[78] H. Shan, *Numerical simulation of flow behind active vortex generators with direct forcing immersed boundary method*, *International Journal of Computational Fluid Dynamics* **21** (2007) 49 – 60.

[79] M. Kupiainen and B. Sjögreen, *A Cartesian embedded boundary method for the compressible Navier-Stokes equations*, *Journal of Scientific Computing* **41** (2009) 94–117.

[80] K. Srinivas, J. Guaruraja, and K. K. Prasad, *On the first order local stability scheme for the numerical solution of time-dependent compressible flows*, *Computers & Fluids* **5** (1977) 87–97.

[81] M. D. de Tullio, P. D. D. Palma, G. Iaccarino, G. Pascazio, and M. Napolitano, *An immersed boundary method for compresible flows using local grid refinement*, *Journal of Computational Physics* **225** (2007) 2098–2117.

[82] H. S. Udaykumar, S. Krishnan, and S. V. Marella, *Adaptively refined parallelisded sharp interface Cartesian grid method for three dimensional moving boundary problems.*, *International Journal of Computational Fluid Dynamics* **23** (2009) 1–24.

[83] E. Uzgoren, J. Sim, and W. Shyy, *Marker based 3-D adapitve Cartesian grid method for multiphase flow around irregular geometries*, *Communications in Computational Physics* **5** (2009) 1–41.

[84] B. van Leer, *Towards the ultimate conservative difference scheme iii. upstream-centered finite-difference schemes for ideal compressible flow*, *Journal of Computational Physics* **23** (1977) 263–275.

[85] B. van Leer, *Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method*, *Journal of Computational Physics* **32** (1979) 101–136.

[86] B. van Leer, *On the relation between the upwind-differencing schemes of Godunov, Engquist-Osher and Roe*, *SIAM Journal on Scientific and Statistical Computing* **5** (1984) 1–20.

[87] Z. Wang, J. Fan, and K. Cen, *Immersed boudnary method for the simulations of 2D viscous flow based on vorticity-velocity formulations*, *Journal of Computational Physics* **228** (2009) 1504–1520.

[88] Y. Younis, A. Bibi, A. Haque, and S. Khushnood, *Vortical flow topology on windward and leeward side of delta wing at supersonic speed*, *Journal of Applied Fluid Mechanics* **2** (2009) 13–21.

[89] J. F. Wendt, ed., *Computational Fluid Dynamics*. Springer Berlin Heidelberg, 2009.

[90] M. Chaudhry, *Open-channel flow*. Springer Verlag, 2008.

Research Papers

Paper 1

Analysis of the Accuracy of the Cartesian Grid Method

M. Asif Farooq and B. Müller

In Proc. 23rd Nordic Seminar on Computational Mechanics
KTH, Stockholm, Sweden, pp. 53-56, 2010.

23rd Nordic Seminar on Computational Mechanics
NSCM-23
A. Eriksson and G. Tibert (Eds)

ANALYSIS OF THE ACCURACY OF THE CARTESIAN GRID METHOD

M. ASIF FAROOQ* AND B. MÜLLER†

* †Department of Energy and Process Engineering, Fluids Engineering Group
Norwegian University of Science and Technology (NTNU)
Kolbjørn Hejes Vei 2 No-7491 Trondheim, Norway
*e-mail: asif.m.farooq@ntnu.no, web page: http://folk.ntnu.no/asiff/

†e-mail: bernhard.muller@ntnu.no - Web page: http://folk.ntnu.no/bmuller

Key words: Cartesian Grid Method, Node-Centred Finite Volume Approach, Embedded Boundary, Ghost Point Treatment, Shock Wave.

Summary. The accuracy of the Cartesian grid method has been investigated for the 1D Burgers' equation and the 1D and 2D compressible Euler equations. Wall boundary conditions are imposed at ghost points by interpolating the numerical solution at the corresponding mirror points linearly or quadratically. We find that linear interpolation does not affect the accuracy of our node-centred finite volume method. When we employ the MUSCL approach with slope limiters, the convergence rate of the Cartesian grid method is reduced similar to corresponding standard body-fitted methods.

1 INTRODUCTION

The Cartesian grid method[1,2,3] has been becoming popular among researchers due to its simplicity, ease of programming and less computational effort compared to body-fitted grid methods. We have been using the ghost point treatment for embedded boundaries.

In this study we analyze the accuracy of the Cartesian grid method for the 1D inviscid Burgers' equation and the 1D and 2D compressible Euler equations. We impose wall boundary conditions at ghost points by interpolating the numerical solution at the mirror points in the fluid domain and mirroring the interpolated values to ensure reflective wall boundary conditions. First order total variation diminishing (TVD) methods are applied for smooth as well as for shock problems. The order of our method is increased by the MUSCL approach with minmod limiter. The first order explicit Euler and the third order TVD Runge-Kutta methods are used for time integration.

For the scalar problem the Cartesian grid method is applied to a smooth solution. For the 1D compressible Euler equations the Cartesian grid method is applied to a normal shock reflection. For the 2D compressible Euler equations we apply the Cartesian grid method to an oblique shock wave.

2 GOVERNING EQUATIONS

2.1 Inviscid Burgers' Equation

The conservative form of the 1D scalar inviscid Burgers' equation with the initial condition reads

$$u_t + (\frac{1}{2}u^2)_x = 0, \qquad u(x,0) = sin\left(\frac{\pi x}{x_b}\right), \tag{1}$$

where x_b is the location of the wall.

2.2 Compressible Euler Equations

The 2D compressible Euler equations for perfect gas in conservative form are given as

$$U_t + (F)_x + (G)_y = 0, \tag{2}$$

where $U = [\rho, \rho u, \rho v, \rho E]^T$, $F = [\rho u, \rho u^2 + p, \rho uv, (\rho E + p)u]^T$ and $G = [\rho v, \rho uv, \rho v^2 + p, (\rho E + p)v]^T$ are the vector of the conservative variables and the flux vectors in $x-$ and $y-$directions, respectively.

3 NUMERICAL METHODS

3.1 DISCRETIZATION SCHEMES

For the spatial discretization we apply the upwind method for inviscid Burgers' equation. We apply the Lax-Friedrichs (LF) and local Lax-Friedrichs (LLF) method for the 1D compressible Euler equations and local Lax-Friedrichs (LLF) method for the 2D compressible Euler equations. To obtain higher order we apply MUSCL with minmod limiter. For time integration we use the first order explicit Euler and the third order TVD Runge-Kutta methods.

3.2 GHOST POINT TREATMENT

In Fig. 1 we show a simplified ghost point treatment for the 2D case. The mathematical form of the ghost point treatment can be written as follows

$$\begin{aligned} u_{n,G} &= -u_{n,M}, \qquad u_{t,G} = u_{t,M}, \\ \rho_G &= \rho_M, \qquad p_G = p_M, \end{aligned} \tag{3}$$

where u_n, u_t, ρ and p denote normal and tangential velocity with respect to the embedded wall boundary, density and pressure, respectively. In Fig. 1, M and G are the mirror and ghost points, respectively. δ is the distance between the ghost point G and the boundary point on the vertical grid line. The ghost point G in the solid is mirrored to the mirror point M in the fluid with respect to the wall (boldface line) on a grid line. The numerical solution at M is interpolated on that grid line.

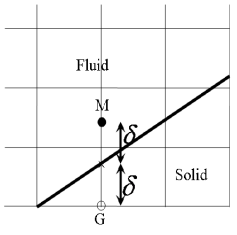


Figure 1: 2D ghost point treatment

4 RESULTS

4.1 INVISCID BURGERS' EQUATION

In this section we present results for the 1D inviscid Burgers' equation. The wall is located at $x_b = 0.5001$ and end time is $tend = 0.02$. In Tables 1 and 2 we show the convergence rates of the first and higher order methods while using linear and quadratic interpolation at the mirror points. It is clear from these tables that linear and quadratic interpolations yield the same error and are not affecting the accuracy of the first and higher order methods.

	Inviscid Burgers' Equation			
	Linear Interpolation		Quadratic Interpolation	
N	Order	L_2-norm	Order	L_2-norm
101	-	0.0018	-	0.0018
201	0.9729	0.009	0.9729	0.009
401	0.9865	0.005	0.9865	0.005
801	0.9932	0.002	0.9932	0.002
1601	0.9965	0.001	0.9965	0.001
3201	0.9980	0.001	0.9980	0.001

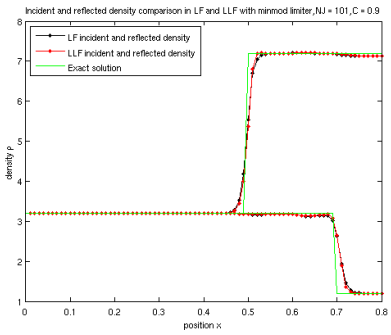
Table 1: First order TVD method.

	Inviscid Burgers' Equation			
	Linear Interpolation		Quadratic Interpolation	
N	Order	L_2-norm	Order	L_2-norm
101	-	0.1772×10^{-3}	-	0.1772×10^{-3}
201	1.5906	0.0588×10^{-3}	1.5906	0.0588×10^{-3}
401	1.5940	0.0195×10^{-3}	1.5940	0.0195×10^{-3}
801	1.6000	0.0064×10^{-3}	1.6000	0.0064×10^{-3}
1601	1.6045	0.0021×10^{-3}	1.6045	0.0021×10^{-3}
3201	1.6079	0.0007×10^{-3}	1.6079	0.0007×10^{-3}

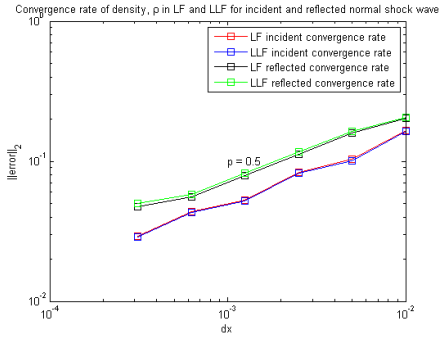
Table 2: Higher order TVD method.

4.2 1D COMPRESSIBLE EULER EQUATIONS

In this section we present results for the 1D compressible Euler equations. The wall is located at $x_b = 0.8001$. At the mirror point we apply linear interpolation. For the spatial discretization we apply the higher order LF and LLF TVD methods. In Figs. 2(a) and 2(b) we present results for a moving normal shock wave. In Fig. 2(a) we draw a comparison between the exact and numerical solutions of density. We observe that the density is lower after reflection from the wall. The convergence rate of the higher TVD method is shown in 2(b). The convergence rate is low (~ 0.5 in the L_2-norm) due to the shock wave.



(a) Comparison of incident and reflected shock for density using the higher order TVD method.

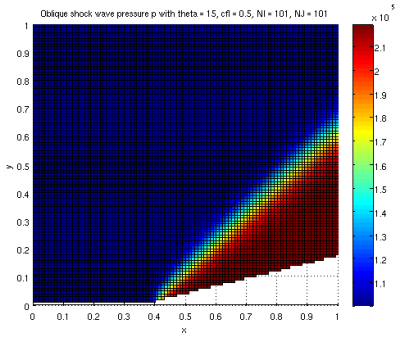


(b) Convergence rate of density for higher order TVD method.

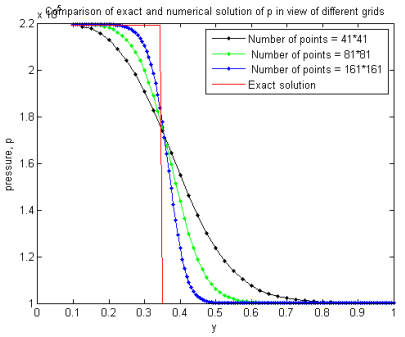
Figure 2: Normal shock wave.

4.3 2D COMPRESSIBLE EULER EQUATIONS

In this section we present results for the 2D compressible Euler equations. We verified our 2D code for an oblique shock wave. We apply the simplified ghost point treatment adjacent to the embedded boundary and use linear interpolation at the mirror points. For spatial discretization we apply the local Lax-Friedrichs (LLF) method. For time integration we employ the explicit Euler method. In Fig. 3(a) we present pressure results for an oblique shock wave at $M_\infty = 2$ and wedge angle of $\Theta = 15$ degrees. In Fig. 3(b) we compare the results for three grids with the exact solution and observe grid convergence for the first order method.



(a) Computed pressure for $M_\infty = 2$ and wedge angle $\Theta = 15$ degrees.



(b) Comparison of exact and numerical solutions for different grids.

Figure 3: Oblique shock wave.

5 CONCLUSIONS

We applied the Cartesian grid method to the scalar 1D inviscid Burgers' equation and the 1D and 2D compressible Euler equations, and both normal and oblique shock waves were computed. Local symmetry boundary conditions were implemented at each ghost point. Accuracy and convergence rate of the Cartesian grid method proved to be similar to standard body fitted methods. We observed the same accuracy for both linear and quadratic interpolation.

REFERENCES

[1] Mittal, R. & Iaccarino, G. Immersed Boundary Methods. *Annual Review of Fluid Mechanics* **37**, 239–261 (2005).

[2] Sjögreen, B. & Petersson, N. A. A Cartesian Embedded Boundary Method for Hyperbolic Conservation Laws. *Commun. Comput. Phys.* **2**, 1199 – 1219 (2007).

[3] Udaykumar, H. S., Krishnann, S. & Marella, S. V. Adaptively Refined Parallelisded Sharp Interface Cartesian Grid Method for Three Dimensional Moving Boundary Problem. *International Journal of Computational Fluid Dynamics* **23**, 1–24 (2009).

Paper 2

Investigation of the Accuracy of the Cartesian Grid Method

M. Asif Farooq and B. Müller

8th International Bhurban Conference on Applied Sciences & Technology, Islamabad, Pakistan, pp. 45-53 , 2011.

Investigation of the Accuracy of the Cartesian Grid Method

M. Asif Farooq*, and Bernhard Müller*
* Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Abstract— **The accuracy of the Cartesian grid method has been investigated for the 1D Burgers' equation as well as the 1D and 2D compressible Euler equations. Wall boundary conditions are imposed at ghost points by interpolating the numerical solution at the corresponding mirror points either linearly or quadratically. We find that linear interpolation does not affect the accuracy of our second order node-centered finite volume method. For smooth problems, our Cartesian grid method exhibits second order convergence. When we employ the MUSCL approach with slope limiters, the convergence rate of the Cartesian grid method is reduced similar to the corresponding standard body fitted methods.**

I. INTRODUCTION

The Cartesian grid method has recently become one of the widely used methods in CFD [1-7]. This is due to its simplicity, faster grid generation, simpler programming, lower storage requirements, lower operation count, and easier post processing compared to body fitted structured and unstructured grid methods. The Cartesian grid method is also advantageous in constructing higher order methods. Problems occur at the boundary, when this method is applied to complex domains [8]. When the Cartesian grid method is applied at curved boundaries the cells at the boundaries are not rectangular and these cut-cells create problems for the scheme to be implemented. One method to solve the time step restriction problem caused by small cut-cells is to merge the cut-cells with neighboring cells [4]. The second method is to use ghost point treatment at the boundary [14]. Here ghost points are introduced outside the domain near the boundary and symmetry conditions are imposed at the ghost points to reflect the presence of the boundary. However, conservativity is lost in this process.

Researchers have been looking into the Cartesian grid method to solve the incompressible Navier-Stokes equations. Often, the immersed boundary method has been used to model the effect of solid boundaries cutting a Cartesian grid by a force term in the incompressible momentum equation [9]. Since this approach is not so practical for compressible flow due to the sensitive coupling of all flow variables, it has not been used for compressible flow simulation except for [10], [2]. Instead, the effect of the tangency or slip condition at solid boundaries for inviscid compressible flow is used in the Cartesian grid method to determine the flow variables in ghost cells or at ghost points near solid boundaries [11-16]. In the ghost point treatment we divide our domain into three types of cells: fluid, ghost and solid cells. For first and second order schemes the methods require one and two ghost points, respectively. Solid and ghost points are flagged inactive.

The goal of the present study is to analyze the accuracy of the Cartesian grid method for the 1D inviscid Burgers' equation and the 1D and 2D compressible Euler equations. Because of its simplicity, we chose the ghost point treatment rather than the cut-cell approach. We impose the wall boundary conditions at the ghost points by interpolating the numerical solution in the fluid domain and mirroring the interpolated values to ensure reflective boundary conditions. We use first order methods for spatial discretization for smooth as well as for shock problems. To increase the accuracy we apply the MUSCL approach with minmod limiter. For time integration we use the first order explicit Euler and the third order TVD Runge-Kutta (RK3) methods. For the scalar Burgers' problem we simulate a smooth solution. For the 1D compressible Euler equations, we use a moving normal shock wave as a test case. For the 2D compressible Euler equations we verify the 2D code for an oblique shock wave.

The paper is organized as follows. In Section II we present the governing equations for the Burgers' equation as well as for the 1D and 2D compressible Euler equations. In Section III we outline the discretization techniques. In Section IV we explain the ghost point treatment at the embedded boundary. In section V we present results and discussions. Conclusions are given in section VI.

II. GOVERNING EQUATIONS

We consider the inviscid Burgers' equation as a model for a 1D scalar nonlinear hyperbolic problem, cf. section A.

The 1D and 2D compressible Euler equations serve as models for 1D and 2D nonlinear hyperbolic systems, cf. section B.

A. *Inviscid Burgers' equation*

The one dimensional scalar inviscid Burgers' equation in conservative form reads

$$\frac{\partial u}{\partial t}+\frac{\partial(\frac{u^2}{2})}{\partial x}=0. \quad (1)$$

The initial condition for the inviscid Burgers' equation is given as

$$u(x,0)=\sin\left(\frac{\pi x}{x_b}\right),\ 0\le x\le 0.5 \quad , \quad (2)$$

where x_b is the location of the wall.

B. *Compressible Euler Equations*

The 2D compressible Euler equations in conservative form read

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=0, \quad (3)$$

where

$$U=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{bmatrix}, F=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (\rho E+p)u\end{bmatrix}, G=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ (\rho E+p)v\end{bmatrix}, \quad (4)$$

with ρ, u, v, E and p denoting density, velocities in $x-$ and $y-$directions, total energy per unit mass and pressure, respectively.

For perfect gas we have the following relation

$$p=(\gamma-1)(\rho E-\frac{1}{2}\rho(u^2+v^2)), \quad (5)$$

where γ is the ratio of specific heats. We consider $\gamma=$ 1.4 for air.

For $v=0$ and $\partial G/\partial y=0$ we obtain from (3) to (5) the 1D compressible Euler equations.

For supersonic flow in the x-direction, the conservative variables at $x=0$ are given as Dirichlet boundary conditions U(0, y , t) = g(y, t). In 1D, we have U(0, t) = g(t). No boundary conditions must be given at the outflow boundary x = L_1, because the flow is supersonic.

Symmetry boundary conditions at y = 0 imply:

$(\rho,\rho u,\rho E)(x,y,t)=(\rho,\rho u,\rho E)(x,-y,t)$)

and

$$\rho v(x,y,t)=-\rho v(x,-y,t)$$

Extrapolation boundary conditions are assumed at y= L_2 :

$$\frac{\partial U(x,y,t)}{\partial y}=0.$$

III. Discretization

A. *Spatial Discretization*

We assume a rectangular domain [0 L_1]×[0 L_2] and a (I+1)×(J+1) Cartesian grid with equidistant grid spacing Δx =L_1 /I and Δy =L_2 /J. The Cartesian coordinates of the grid points (i, j) are (x_i ,y_j), where x_i = iΔx , i = 0, 1, …, I , and y_j = jΔy , j = 0, 1, …, J.

The node-centered finite volume method yields the following semi-discretization of the 2D compressible Euler equations (3)

$$\frac{dU_{i,j}}{dt}=-\frac{1}{\Delta x}(F^n_{i+\frac{1}{2},j}-F^n_{i-\frac{1}{2},j})-\frac{1}{\Delta y}(G^n_{i,j+\frac{1}{2}}-G^n_{i,j-\frac{1}{2}}), \quad (6)$$

Where $U_{i,j}$ is the approximation of the average of U in the cell $\Omega_{i,j}=[x_i-\frac{\Delta x}{2},x_i+\frac{\Delta x}{2}]\times[y_j-\frac{\Delta y}{2},y_j+\frac{\Delta y}{2}]$, i.e.

$$U_{i,j}\approx\frac{1}{\Delta x.\Delta y}\int_{\Omega_{i,j}}U(x,y,t)\,dxdy \ . \quad (7)$$

If we interpret (6) as a conservative finite difference method, $U_{i,j}$ is an approximation of the exact solution U(x_i , y_j , t). $F_{i+1/2,j}$ and $G_{i,j+1/2}$ are numerical fluxes for the 2D compressible Euler equations. The vector of the conservative variables U and the flux vectors F and G are defined in (4). The numerical fluxes of the local Lax-Friedrichs method for F and G are defined as follows

$$F^{LLF}_{i+\frac{1}{2},j}=\frac{1}{2}(F(U_{i,j})+F(U_{i+1,j})-\max(|u_{i,j}|+c_{i,j},|u_{i+1,j}|+c_{i+1,j})(U_{i+1,j}-U_{i,j})), \quad (8)$$

$$G^{LLF}_{i,j+\frac{1}{2}}=\frac{1}{2}(G(U_{i,j})+G(U_{i,j+1})-\max(|v_{i,j}|+c_{i,j},|v_{i,j+1}|+c_{i,j+1})(U_{i,j+1}-U_{i,j})), \quad (9)$$

In equations (8) and (9), c is the speed of sound. The CFL number for the 1D compressible Euler equations is defined as $CFL=\frac{\max_i(|u_i|+c_i)\Delta t}{\Delta x}$. And the CFL number for the 2D compressible Euler equations is defined as

$$CFL=\max_{i,j}\left(\frac{sp(A_1(U_{i,j}))}{\Delta x}+\frac{sp(A_2(U_{i,j}))}{\Delta y}\right)\Delta t,$$

where $sp(A_1)$ and $sp(A_2)$ are the spectral radii of the Jacobian matrices $A_1=\partial F/\partial U$ and $A_2=\partial G/\partial U$, respectively. We choose CFL = 0.9 and CFL = 0.5 for the results of the 1D and 2D compressible Euler equations, respectively. In (8) we replace $U_{i,j}$ by $U^{L}_{i+1/2.j}$ and

$U_{i+1,j}$ by $U^{R}_{i+1/2,j}$ using the MUSCL approach with the minmod limiter to obtain higher order accuracy and also to get rid of undesired oscillations. The extrapolated variables are defined as

$$U^{L}_{i+1/2,j} = U_{i,j} + \frac{1}{2}\min\text{mod}(U_{i,j} - U_{i-1,j}, U_{i+1,j} - U_{i,j}), \tag{10}$$

$$U^{R}_{i+1/2,j} = U_{i+1,j} - \frac{1}{2}\min\text{mod}(U_{i+2,j} - U_{i+1,j}, U_{i+1,j} - U_{i,j}). \tag{11}$$

The MUSCL approach is applied similarly to the numerical fluxes $G_{i,j+1/2}$ in (9).

For the inviscid Burgers' equation (1), the numerical flux for the local Lax-Friedrichs method reads

$$F^{LLF}_{i+\frac{1}{2}} = \frac{1}{2}\left(\frac{u_i^2}{2} + \frac{u_{i+1}^2}{2} - \max(|u_i|, |u_{i+1}|)(u_{i+1} - u_i)\right). \tag{12}$$

To obtain the Lax-Friedrichs and upwind method we replace $\max(|u_i|, |u_{i+1}|)$ by $\Delta x/\Delta t$ and $\left|\frac{1}{2}(u_i + u_{i+1})\right|$, respectively in (12).

B. *Approximation of Boundary Conditions*

The inflow boundary conditions for supersonic flow at $x = 0$ are imposed as

$$U_{0,j}(t) = g(y_j, t) \ . \tag{13}$$

The flow variables at the outflow boundary $x = L_1$ are approximated as

$$U_{I,j}(t) = U_{I-1,j}(t), \tag{14}$$

i.e. by constant extrapolation. This approximation implies that the upwind finite volume method is used to determine the numerical fluxes $F_{I-1/2,j}$.

The symmetry boundary conditions are implemented by considering an extra line below $y = 0$. There we use

$$U_{i,1}(t) = diag(1,1,-1,1)U_{i,3}(t) \ . \tag{15}$$

The boundary conditions at $y = L_2$ are treated as

$$U_{i,J}(t) = U_{i,J-1}(t) \ . \tag{16}$$

The ghost point treatment is described in the next section.

IV. Ghost Point Treatment at Embedded Boundary

In Fig. 1 we show the flagging strategy. We flag the ghost and solid points by assigning them 0 and -1 values. The fluid points are assigned values equal to 1. In Fig. 2 we show a 2D graphical description of the treatment at the boundary. We divide our domain into fluid and ghost points. A ghost point is denoted by G. We used a simplified ghost point treatment at the boundary. The mathematical description of this strategy is given as

$$\begin{aligned} u_G &= u_F - 2(n_1 u_F + n_2 v_F)n_1, \\ v_G &= v_F - 2(n_1 u_F + n_2 v_F)n_2, \\ p_G &= p_F, \rho_G = \rho_F. \end{aligned} \tag{17}$$

Where u, v, p and ρ denote the velocities in x and y directions, pressure and density respectively. In (17) n_1 and n_2 are the components of the unit normal at the boundary.

The location of the fluid point (x_F , y_F) whose flow conditions are mirrored to the ghost point (x_G , y_G) = (x_{iG} , y_{iG}) is chosen as (x_F , y_F) = (x_{iG} , y_{iG+1}), cf. Fig. 2. The 1D boundary treatment is indicated in Fig. 3. In Fig. 3 we show three points x_i, x_{i-1} and x_g as fluid and ghost points. The wall is located between fluid and ghost points. The location of the wall is represented by x_b. The procedure to find the ghost points in 1D is explained as follows:

Step 1: Find the distance between the ghost point and the wall

$$\delta_1 = x_g - x_b \tag{18}$$

Step 2: In this step we determine the mirror point

$$x_m = x_b - \delta_1 \tag{19}$$

Step 3: After steps 1 and 2 we approximate the flow variables at the mirror point in the fluid domain. Here we apply linear or quadratic interpolation to get the interpolated values. Linear interpolation yields

$$V_m = V_{i-1} + \left(\frac{x_m - x_{i-1}}{\Delta x}\right)(V_i - V_{i-1}). \tag{20}$$

Where V = $(\rho, u, p)^T$ here. Now we use reflective boundary conditions to obtain the ghost point values.

$$u_g = -u_m, \rho_g = \rho_m, p_g = p_m \ . \tag{21}$$

The procedure to find the second ghost point for the quadratic interpolation is similar as explained above.

For the inviscid Burgers' equation, the ghost point treatment is the same as for the velocity u in the 1D compressible Euler equations above.

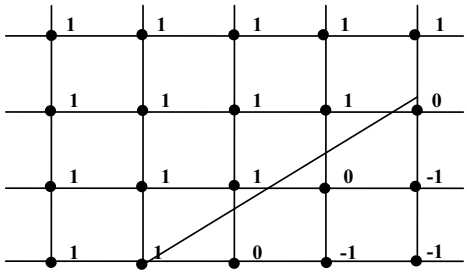


Figure 1. Flagging strategy for fluid, ghost and solid points. Using flags 1, 0 and -1 respectively.

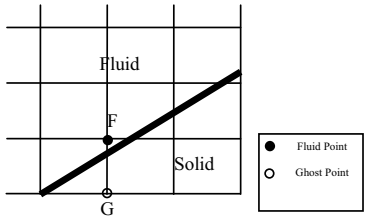


Figure 2. 2D treatment at the boundary.

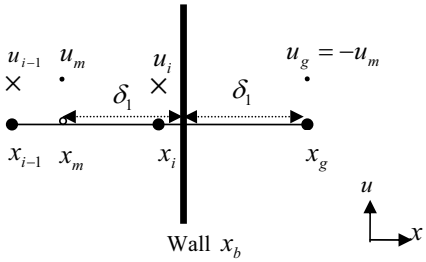


Figure 3. 1D treatment at the boundary.

V. RESULTS

A. *Inviscid Burgers' Equation*

In this section we present results for the inviscid Burgers' equation. We apply the first order upwind method as well as MUSCL with minmod limiter for spatial discretization. For time integration we use the first order explicit Euler method and the third order TVD Runge-Kutta method. Results for the inviscid Burgers' equation are obtained at CFL = 0.1. The wall is located at $x_b = 0.5001$ and we use linear and quadratic interpolation to obtain the ghost point values.

In Table I we present results for the inviscid Burgers' equation. For the spatial discretization we use the first order upwind method and time is integrated with the explicit Euler method. We determine the order and l_2-norm of linear and quadratic interpolation. We observe that the l_2-norm of the error for linear and quadratic interpolation is the same.

TABLE I. CONVERGENCE RATE FOR FIRST ORDER METHOD WITHOUT MUSCL.

Inviscid Burgers' Equation				
	Linear Interpolation		***Quadratic Interpolation***	
N	***Order***	***l_2-norm***	***Order***	***l_2-norm***
101	-	0.0018	-	0.0018
201	0.9729	0.009	0.9729	0.009
401	0.9865	0.005	0.9865	0.005
801	0.9932	0.002	0.9932	0.002
1601	0.9965	0.001	0.9965	0.001
3201	0.9980	0.001	0.9980	0.001

In Table II we show results for the inviscid Burgers' equation. For the spatial discretization we use the higher order MUSCL scheme with minmod limiter and for time integration we use the third order TVD Runge-Kutta method. We obtain the same order and l_2-norm of the error for linear and quadratic interpolation.

TABLE II. CONVERGENCE RATE FOR NUMERICAL SOLUTION WITH MUSCL AND TVD RK3.

Inviscid Burgers' Equation				
	Linear Interpolation		***Quadratic Interpolation***	
N	***Order***	***l_2-Norm***	***Order***	***l_2-Norm***
101	-	0.1772×10^{-3}	-	0.1772×10^{-3}
201	1.5906	0.0588×10^{-3}	1.5906	0.0588×10^{-3}
401	1.5940	0.0195×10^{-3}	1.5940	0.0195×10^{-3}
801	1.6000	0.0064×10^{-3}	1.6000	0.0064×10^{-3}
1601	1.6045	0.0021×10^{-3}	1.6045	0.0021×10^{-3}
3201	1.6079	0.0007×10^{-3}	1.6079	0.0007×10^{-3}

In Fig. 4 we present the convergence rate of the first order upwind method for the inviscid Burgers' equation for a smooth solution. We get the same convergence rate for linear and quadratic interpolation.

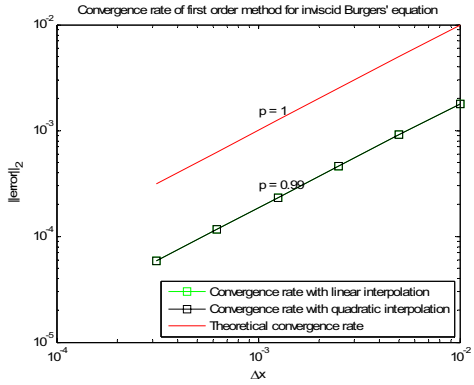


Figure 4. Convergence rate for inviscid Burgers' equation.

In Fig. 5 we present the convergence rate of the upwind method with MUSCL and minmod limiter for the inviscid Burgers' equation for a smooth solution. The computed convergence rate is lower than the theoretical convergence, because the MUSCL scheme gives lower order at the extrema of the smooth solution. We see that for the MUSCL scheme, too, the convergence rates for linear and quadratic interpolation are the same.

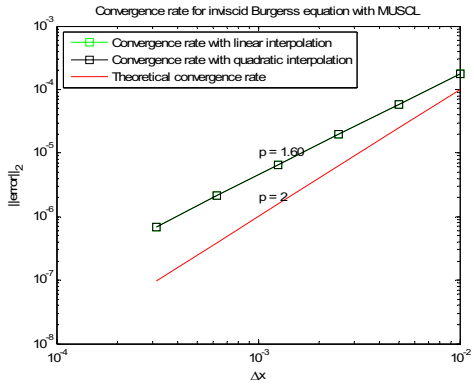


Figure 5. Convergence rate for inviscid Burgers' equation.

B. *1D Compressible Euler Equations*

In this section we present results for a moving normal shock wave. A normal shock wave is moving from left to right with upstream Mach number M = 2, pressure p = 10^5 Pa and density $\rho = 1.2\ kg/m^3$. The downstream conditions can be obtained by the normal shock wave relations. A wall is placed at $x = 0.8001$ in Figs. 6-8. The incident normal shock wave is shown at $x = 0.7$, and the reflected normal shock wave is shown at $x = 0.5$.

In Fig. 6 we show results of the moving normal shock wave for density ρ. The Lax-Friedrichs (LF) and local Lax-Friedrichs (LLF) methods with MUSCL and minmod limiter for spatial discretization are applied. For time integration the third order TVD Runge-Kutta method is employed. We notice that the computed density is lower than the exact one after reflection from the wall. We also observe a wavy behavior behind the normal shock wave which travels along with the shock.

In Fig. 7 we show results of the moving normal shock wave for velocity u. We apply the LF and LLF methods with MUSCL and minmod limiter for spatial discretization. We observe a wavy behavior behind the normal shock wave but this wavy behavior disappears after the reflection.

In Fig. 8 we show results of the moving normal shock wave for pressure p. Again we observe little wavy behavior behind the incident normal shock wave. This wavy behavior disappears after the reflection.

In Fig. 9 we show the convergence rate for density ρ. The convergence rate is low (≈ 0.5 in the l_2 -norm) due to the shock wave. The error l_2 -norm of the reflected wave is higher than for the incident wave.

In Fig. 10 we show the convergence rate of velocity u. Again, the convergence rate is low (≈ 0.5 in the l_2 -norm) due to the shock wave.

In Fig. 11 we show the convergence rate for the pressure p. Here again the convergence rate is low (≈ 0.5 in the l_2 -norm) due to the shock wave. The error l_2 -norm of the reflected wave is higher than for the incident wave.

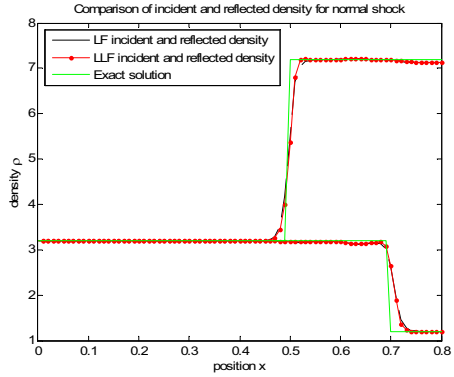


Figure 6. Reflected normal shock waves of density with MUSCL.

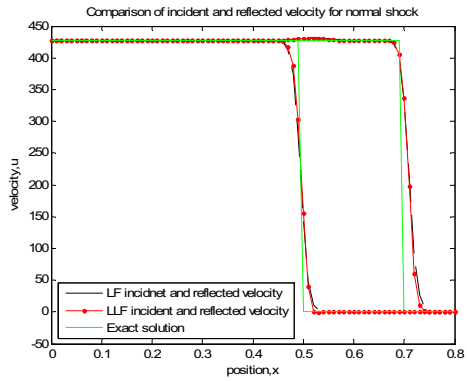


Figure 7. Reflected normal shock waves of velocity with MUSCL.

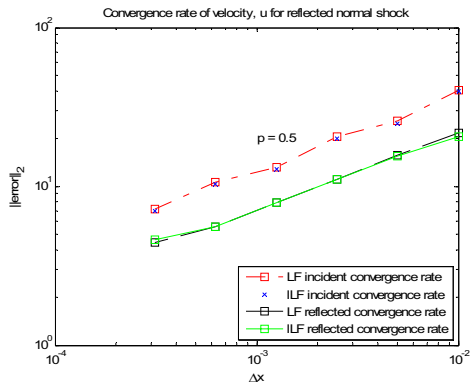


Figure 10. Convergence rates of velocity for incident and reflected normal shock waves with MUSCL.

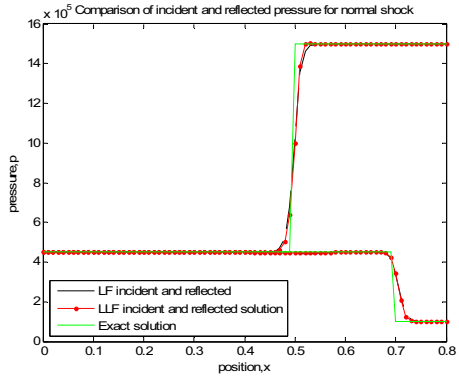


Figure 8. Reflected normal shock waves of pressure with MUSCL.

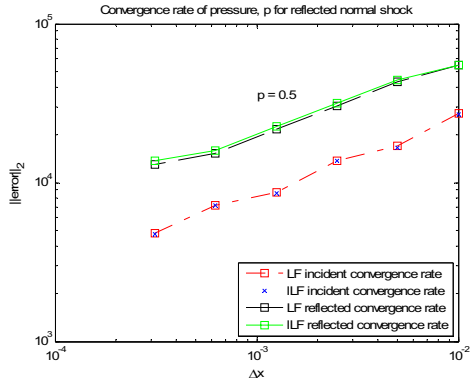


Figure 11. Convergence rates of pressure for incident and reflected normal shock waves with MUSCL.

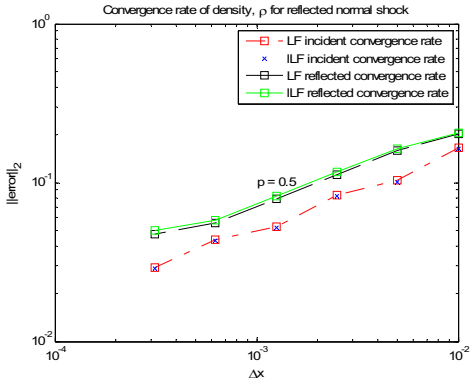


Figure 9. Convergence rates of density for incident and reflected normal shock waves with MUSCL.

C. *2D Compressible Euler Equations*

We verify our 2D code of the Cartesian grid method for an oblique shock wave. A supersonic flow moves from left to right and hits a wedge resulting in a stationary oblique shock wave. The wedge angle is $\theta = 15$ degrees. The supersonic upstream flow conditions are given as

$$M = 2\,,\ p_\infty = 10^5\ \text{Pa},\ \rho_\infty = 1.2\ \text{kg/m}^3 \quad (22)$$

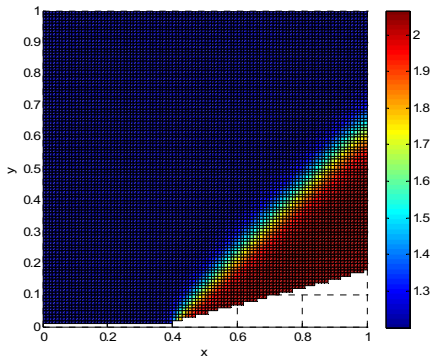


Figure 12. Oblique shock wave of density.

For the spatial discretization the local Lax-Friedrichs method is used. The explicit Euler method is employed for time integration.

In Fig. 12 we present results for the oblique shock wave for density. The apex of the wedge is placed at x =0.4.

In Fig. 13 we compare the exact and numerical solutions for density at x = 0.75. We observe that the density is getting closer to the exact solution as we refine the grid. However, there is some discrepancy between the exact and computed solutions near the wall of the wedge.

In Fig. 14 we compare the exact and numerical solutions for velocity u at x = 0.75. The computed velocity u is becoming closer to the exact solution as we refine the grid but still there is a discrepancy near the wall of the wedge.

In Fig. 15 we compare the exact and numerical solutions for velocity v at x = 0.75. The computed results are in good agreement with the exact solution.

In Fig. 16 we compare the exact and numerical solutions for pressure p at x = 0.75. The computed pressure is in good agreement with the exact solution.

In Fig. 17 we show density contours. This confirms that the variation is occurring at the place where we observe the oblique shock wave. The proximity of the oblique shock wave is unperturbed.

In Fig.18 we show the l_2-norm of the residual $\left(\rho_{i,j}^{n+1} - \rho_{i,j}^{n}\right)/\Delta t$ of the density. We see that the residual becomes stationary after 5500 time levels. It means at this corresponding time we achieved the stationary oblique shock wave solution.

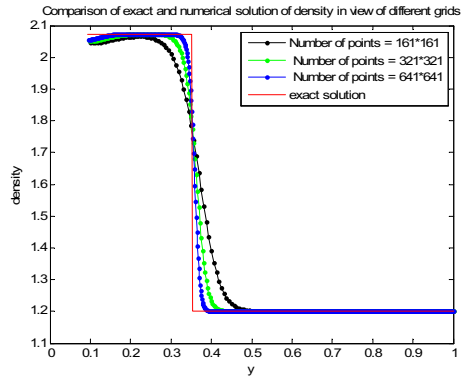


Figure 13. Comparison of exact and numerical solutions for density at different grid levels at x = 0.75.

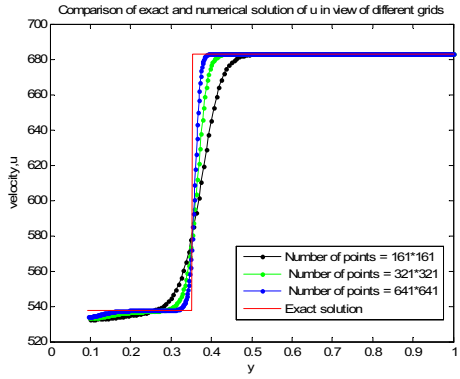


Figure 14. Comparison of numerical and exact solutions for velocity u at different grid levels at x = 0.75.

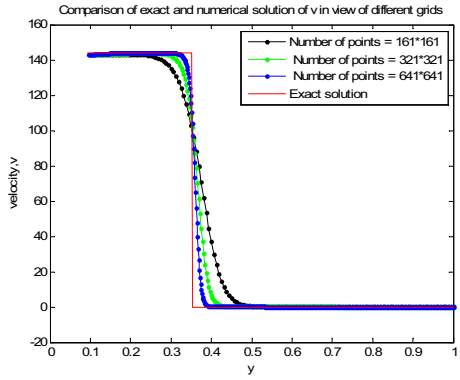


Figure 15. Comparison of exact and numerical solutions for velocity v at different grid levels at x = 0.75.

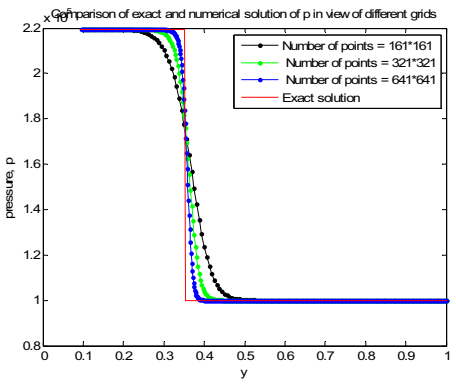


Figure 16. Comparison of exact and numerical solutions for pressure at different grid levels at $x = 0.75$.

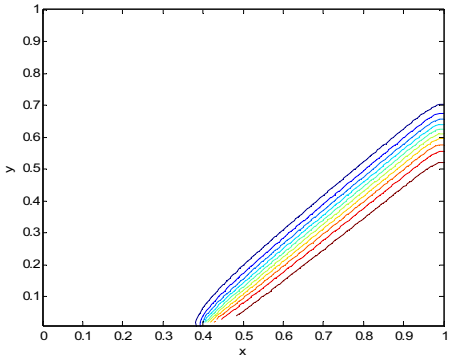


Figure 17. Contour lines for density.

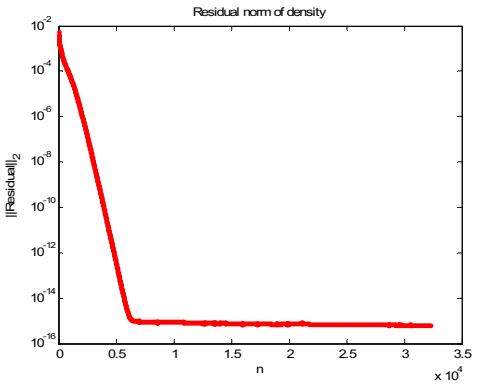


Figure 18. l_2 -norm of residual of density.

VI. Conclusions

We applied the Cartesian grid method to the scalar 1D inviscid Burgers' equation and the 1D and 2D compressible Euler equations, and both normal and oblique shock waves were computed. Local symmetry boundary conditions were implemented at each ghost point. Accuracy and convergence rate of the Cartesian grid method proved to be similar to standard body fitted methods. We observed the same accuracy for both linear and quadratic interpolation.

Acknowledgment

The current research has been funded by the Higher Education Commission (HEC) of Pakistan.

References

[1] A. S. Almgren, J. B. Bell, P. Colella and T. Marthaler, "A Cartesian gird projection method for the incompressible Euler equations in complex geometries", SIAM J. Sci. Computing, vol. 18, pp 1289-1309, 1997.

[2] P. De Palma, M. D. de Tullio, G. Pascazio and M. Napolitano, " An immersed boundary method for compresseible viscous flows", Computers & Fluids, vol. 35, pp. 693-702, 2006.

[3] D. D. Marshall and S. M. Ruffin, " A new inviscid wall boundary condition treatment for boundary Cartesian grid method", AIAA 2004-0583 42nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada , Jan 5 - 8, 2004.

[4] R. Mittal and G. Iaccarino, "Immersed boundary methods", Annu. Rev. Fluid Mech., vol. 37, pp. 239-261, 2005.

[5] H. S. Udaykumar, S. Krishnan and S. V. Marella, " Adaptively refined, parallelized sharp interface Cartesian grid method for three dimensional moving boundary problem", International Journal of Computational Fluid Dynamics, vol. 23, pp. 1-24, 2009.

[6] E. Uzgoren, J. Sim and W. Shyy, "Marker based, 3-D adaptive Cartesian grid method for multiphase flow around irregular geometries", Commun. Comput. Phys., vol.5, pp. 1-41, 2009.

[7] Z. Wang, J. Fan and K. Cen, " Immersed boundary method for the simulations of 2D viscous flow based on vorticity-velocity formulations" J. Comp. Phys., vol. 228, pp. 1504-1520, 2009.

[8] J. J. Quirk, "An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies", Comput. Fluids, vol. 23, pp. 125-142, 1994.

[9] C. S. Peskin, "Flow pattern around heart valves: A numerical method", J. Comput. Physcis, vol. 10, pp. 252-271, 1972.

[10] M. D. de Tullio, P. D. De Palma, G. Iaccarino, G. Pascazio and M. Napolitano, " An immersed boundary method for compressible flows using local grid refinement", J. Comp. Phys, vol. 225, pp. 2098-2117, 2007.

[11] M. J. Berger and R. J. Leveque, "A rotated difference scheme for Cartesian grids in complex geometries", AIAA Paper CP-91-1602, 1991.

[12] R. B. Pember, J. B. Bell, P. Colella, W. Y. Cruthfield and M. L. Welcome, " An adaptive Cartesian grid method for unsteady compressible flow in irregular regions", J. Comput. Phys., vol. 120, pp. 278-304, 1995.

[13] W. J. Coirier and J. G. Powell, "An accuracy assessment of Cartesian mesh approaches for the Euler equations", J. Comput. Phys, vol. 117, pp. 121-131, 1995.

[14] H. Forrer and R. Jeltsch, "A higher order boundary treatment for Cartesian grid methods, J. Comput. Phys., vol. 140, pp. 259-277, 1998.

[15] P. Colella, D. T. Graves, B. J. Keen and D. Modiano, "A Cartesian grid embedded boundary method for hyperbolic conservation laws", J. Comput Phys., vol. 211, pp. 347-366, 2006.

[16] B. Sjögreen and N. A. Petersson, "A Cartesian grid embedded boundary method for hyperbolic conservation laws", Commun. Comput. Phys., vol. 2, pp. 1199-1219, 2007.

[17] M. J. Berger, C. Helzel and R. J. Leveque, "h-box methods for the approximation of hyperbolic conservation laws on irregular grids", SIAM J. Numer. Anal, vol.41, pp. 893-918, 2003.

Paper 3

A Cartesian Grid Method for Compressible Inviscid Flows

M. Asif Farooq and B. Müller

In Proc. 6th National Conference on Computational Mechanics, Trondheim, Norway, pp. 47-58, 2011.

A Cartesian Grid Method for Compressible Inviscid Flows

M. Asif Farooq*, B. Müller
Department of Energy and Process Engineering, Faculty of Engineering Science
The Norwegian University of Science and Technology
e–mail: asif.m.farooq@ntnu.no, bernhard.muller@ntnu.no

Summary The Cartesian grid method has been investigated for the 2D compressible Euler equations. We impose wall boundary conditions by a simplified ghost point treatment for compressible inviscid flows with immersed boundaries. In the simplified ghost point treatment we assume the solid boundary lie in the middle between two grid points in the y direction. Symmetry conditions are used to determine density, pressure, wall tangential, and wall normal velocity components at the ghost points. A first order node-centered finite volume formulation has been used to calculate compressible flows. The order of this formulation is increased by employing MUSCL with the minmod limiter. The code is tested for external flows over a circular arc airfoil. We compare our results for the simplified ghost point treatment with a standard body-fitted grid method. We obtain similar results for the Cartesian grid method as for the standard body-fitted grid method for external flows.

Introduction

The Cartesian grid method has recently become one of the widely used methods in CFD [1, 2, 3, 4, 5, 6, 7, 8, 9]. Since solid boundaries are immersed in a Cartesian grid method with constant grid spacings, the Cartesian grid method is also called immersed boundary method, in particular in application to the incompressible Navier-Stokes equations. The popularity of the Cartesian grid method is due to its simplicity, faster grid generation, simpler programming, lower storage requirements, lower operation count, and easier post processing compared to body fitted structured and unstructured grid methods. The Cartesian grid method is also advantageous in constructing higher order methods. Problems occur at the boundary, when this method is applied to complex domains [10]. When the Cartesian grid method is applied at curved boundaries the cells at the boundaries are not rectangular and these cut-cells create problems for the scheme to be implemented [11]. The time step restriction problem caused by small cut-cells can be solved by merging those cut-cells with neighboring cells [7].

Cut cells are avoided altogether by ghost point treatment at the boundary. In this method symmetry conditions with respect to the boundary are imposed at ghost points in the solid adjacent to the boundary [12]. However, conservativity is lost in this process. Nevertheless, the simplicity of the ghost point treatment has motivated us to use that approach instead of the more complicated cut-cells.

Often the effect of solid boundaries cutting a Cartesian grid has been modelled by a force term in the incompressible momentum equations [13]. Since this approach is not so practical for compressible flow due to the sensitive coupling of all flow variables, it has not been used for compressible flow simulation except for [14, 2]. Instead, the effect of the tangency or slip condition at solid boundaries for inviscid compressible flow is used in the Cartesian grid method to determine the flow variables in ghost cells or at ghost points near solid boundaries [15, 16, 17, 12, 18, 19]. In the ghost point treatment we divide our domain into three types of points: fluid, ghost and solid points. For first and second order schemes the methods require one and two ghost points, respectively. Solid and ghost points are flagged inactive.

In this paper we employ a simplified ghost point treatment for the 2D compressible Euler equations. In the simplified ghost point treatment we assume the solid boundary lie in the middle

between two grid points in the y direction. Symmetry conditions are used to determine density, pressure, wall tangential, and wall normal velocity components at the ghost points. We employ the local Lax-Friedrichs (lLF) method for the spatial discretization. To increase the accuracy we apply the MUSCL approach with the minmod limiter. For time integration we use the first order explicit Euler method for spatially first order method and the third order TVD Runge-Kutta (RK3) method with the more accurate MUSCL approach. As a test case, we consider supersonic external over a circular arc airfoil and solve the 2D compressible Euler equations by time stepping for the steady state.

The paper is organized as follows. First we present the governing equations, i.e. the 2D compressible Euler equations then we outline the boundary conditions and explain the ghost point treatment at the embedded boundary. We present the discretization method, results and discussion and give conclusions.

Compressible Euler Equations

The 2D compressible Euler equations serve as a model for a 2D nonlinear hyperbolic system. In conservative form the 2D compressible Euler equations read

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0, \tag{1}$$

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho E + p)u \end{bmatrix}, G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho E + p)v \end{bmatrix}, \tag{2}$$

with ρ , u, v, E, and p are density, velocity components in x-and y-directions, total energy per unit mass and pressure, respectively.

For perfect gas we have the following relation

$$p = (\gamma - 1)(\rho E - \frac{1}{2}\rho(u^2 + v^2)), \tag{3}$$

where γ is the ratio of specific heats. We consider γ = 1.4 for air.

Approximation of Boundary Conditions

Boundary Conditions for External Flow

The inflow boundary conditions for supersonic flow at $x = 0$ are imposed as

$$U_{1,j}(t) = U_\infty, \tag{4}$$

where U_∞ is the vector of the conservative variables for uniform flow in the $x-$ direction. The flow variables at the outlet $x = L_1$ are approximated as

$$U_{I,j}(t) = U_{I-1,j}(t), \tag{5}$$

i.e. by constant extrapolation. This approximation implies that the upwind finite volume method is used to determine the numerical fluxes $F_{I-\frac{1}{2},j}$.

The symmetry boundary conditions at $y = 0$ are implemented by considering an extra line below $y = 0$. There we use

$$U_{i,1}(t) = diag(1, 1, -1, 1)U_{i,3}(t). \tag{6}$$

The boundary conditions at $y = L_2$ are treated as

$$U_{i,J}(t) = U_{i,J-1}(t). \tag{7}$$

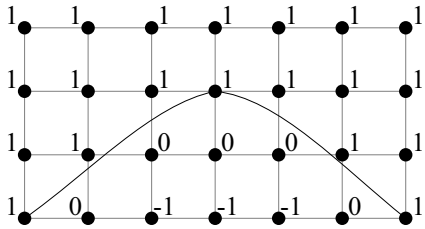
Ghost Point Treatment at Embedded Boundary

Simplified Ghost Point Treatment for Two Dimensional Embedded Boundary

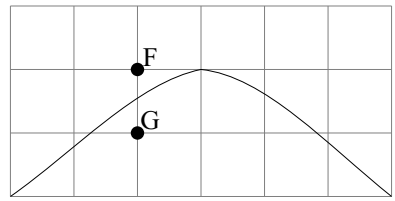
In Fig. 1(a) we show the flagging strategy. We flag the ghost and solid points by assigning them 0 and -1 values. The fluid points are assigned values equal to 1. In Fig. 1(b) we show a simplified ghost point treatment at the solid boundary. A ghost point is denoted by G. In the simplified ghost point treatment we consider the fluid point F on the vertical grid line through G adjacent to the boundary as the mirror point. Then, we assume the boundary is in the middle between ghost and fluid points. The density ρ, pressure p, and the tangential velocity component at the ghost point are symmetric with respect to the solid boundary and therefore directly determined by their values at F. The normal velocity component u_n at the ghost point is anti-symmetric and thus gets the negative value of u_n at F. The mathematical description of this strategy is given as

$$\rho_G = \rho_F, p_G = p_F, u_G = u_F - 2(n_1u_F + n_2v_F)n_1, v_G = v_F - 2(n_1u_F + n_2v_F)n_2, \tag{8}$$

where n_1 and n_2 are the x-and y-components of the outer unit normal **n** of the boundary.



(a) Flagging strategy for fluid (1), ghost (0) and solid points (-1).



(b) Simplified ghost point treatment over a circular arc airfoil.

Figure 1: Ghost point treatment.

Discretization of the Euler Equations

Spatial Discretization

We assume a rectangular domain [0 L_1] $\times$ [0 L_2] and a $(I + 1) \times (J + 1)$ Cartesian grid with equidistant grid spacing $\Delta x = L_1/I$ and $\Delta y = L_2/J$. The Cartesian coordinates of the grid points (i, j) are (x_i, y_j), where $x_i = i\Delta x$, $i = 0, 1, ..., I$, and $y_j = j\Delta y$, $j = 0, 1, ..., J$.

The node-centered finite volume method yields the following semi-discretization of the 2D compressible Euler equations (1)

$$\frac{\mathrm{d}U_{i,j}}{\mathrm{d}t} = -\frac{F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}}{\Delta x} - \frac{G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}}}{\Delta y}. \tag{9}$$

Where $U_{i,j}$ is the approximation of the average of U in the cell $\Omega_{i,j} = [x_i - \frac{\Delta x}{2}, x_i + \frac{\Delta x}{2}] \times [y_j - \frac{\Delta y}{2}, y_j + \frac{\Delta y}{2}]$, i.e.

$$U_{i,j} \approx \frac{1}{\Delta x.\Delta y} \int_{\Omega_{i,j}} U(x,y,t)dxdy. \tag{10}$$

If we interpret (9) as a conservative finite difference method, $U_{i,j}$ is an approximation of the exact solution $U(x_i, y_j, t)$. $F_{i+\frac{1}{2},j}$ and $G_{i,j+\frac{1}{2}}$ are numerical fluxes for the 2D compressible Euler equations at the cell faces ($i + \frac{1}{2}$, j) and (i, $j + \frac{1}{2}$), respectively. The vector of the conservative variables U and the flux vectors F and G are defined in (2). The numerical fluxes of the local Lax-Friedrichs (lLF) method are defined as follows

$$F^{lLF}_{i+\frac{1}{2},j} = \frac{1}{2}\left[F(U_{i,j}) + F(U_{i+1,j}) - max(|u_{i+1,j}| + c_{i+1,j}, |u_{i,j}| + c_{i,j})(U_{i+1,j} - U_{i,j})\right], \tag{11}$$

$$G^{lLF}_{i,j+\frac{1}{2}} = \frac{1}{2}\left[G(U_{i,j}) + G(U_{i,j+1}) - max(|u_{i,j+1}| + c_{i,j+1}, |u_{i,j}| + c_{i,j})(U_{i,j+1} - U_{i,j})\right]. \tag{12}$$

In equations (11) and (12), c is the speed of sound. The CFL number for the 2D compressible Euler equations is defined as $CFL = \Delta t \underset{i,j}{max}\left(\frac{sp(A_1(U_{i,j}))}{\Delta x} + \frac{sp(A_2(U_{i,j}))}{\Delta y}\right)$, where $sp(A_1(U_{i,j}))$ and $sp(A_2(U_{i,j}))$ are the spectral radii of the Jacobian matrices $A_1 = \frac{\partial F}{\partial x}$ and $A_2 = \frac{\partial G}{\partial y}$, respectively. We choose $CFL = 0.5$ for the results of the 2D compressible Euler equations presented below. In (11) we replace $U_{i,j}$ by $U^L_{i+\frac{1}{2},j}$ and $U_{i+1,j}$ by $U^R_{i+\frac{1}{2},j}$ using the MUSCL approach with the minmod limiter to obtain second order accuracy except for extrema without undesired oscillations. The extrapolated variables are defined as

$$U^L_{i+\frac{1}{2},j} = U_{i,j} + \frac{1}{2} minmod(U_{i,j} - U_{i-1,j}, U_{i+1,j} - U_{i,j}), \tag{13}$$

$$U^R_{i+\frac{1}{2},j} = U_{i+1,j} - \frac{1}{2} minmod(U_{i+2,j} - U_{i+1,j}, U_{i+1,j} - U_{i,j}). \tag{14}$$

where

$$minmod(a,b) = \begin{cases} a & if\,|a| \le |b| \quad and \quad ab > 0 \\ b & if\,|b| < |a| \quad and \quad ab > 0 \\ 0 & if ab \le 0 \end{cases} \tag{15}$$
$$= sign(a) max\left\{0, min\left\{|a|, sign(a)b\right\}\right\}$$

is the minmod limiter. The MUSCL approach is applied similarly to the numerical fluxes $G_{i,j+\frac{1}{2}}$ in (12).

Temporal Discretization

For time integration we use the explicit Euler method and the total variation diminishing third order Runge-Kutta (TVD RK3) method. The explicit Euler method is defined as follows

$$U^{n+1} = U^n + \Delta t \mathbf{Res}(U^n), \tag{16}$$

where $\mathbf{Res}(U^n)$ at (i,j) is the residual, i.e. the right hand side of (9). The TVD RK3 method is given as

$$\begin{aligned} U^{(1)} &= U^n + \Delta t \mathbf{Res}(U^n), \\ U^{(2)} &= \frac{3}{4}U^n + \frac{1}{4}U^{(1)} + \frac{1}{4}\Delta t \mathbf{Res}(U^{(1)}), \\ U^{(n+1)} &= \frac{1}{3}U^n + \frac{2}{3}U^{(2)} + \frac{2}{3}\Delta t \mathbf{Res}(U^{(2)}). \end{aligned} \tag{17}$$

Results

Two Dimensional Compressible Euler Equations

We verify our 2D code of the Cartesian grid method for external flow over a circular arc airfoil. For the spatial discretization we use the local Lax-Friedrichs (lLF) method, and to increase the order of our method we employ the MUSCL scheme with the minmod limiter. For time integration we use the first order explicit Euler and third order TVD Runge-Kutta (RK3) methods. We use the CFL number 0.5 and 81 $\times$ 81 grid points to calculate the density ρ, velocities u and v, and pressure p.

External Flow Over a Circular Arc Airfoil

A supersonic flow moves from left to right and hits a circular arc airfoil of which we only consider the upper half. We consider the length of domain in $x-$and $y-$directions is $L_1 = 2m$ and $L_2 = 1m$ respectively. The height of the half circular arc airfoil is 10 $\%$ of its chord length which is assumed to be 1m. The supersonic upstream flow conditions are given as

$$M_\infty = 2, p_\infty = 10^5 Pa, \rho_\infty = 1.2 kg/m^3 \tag{18}$$

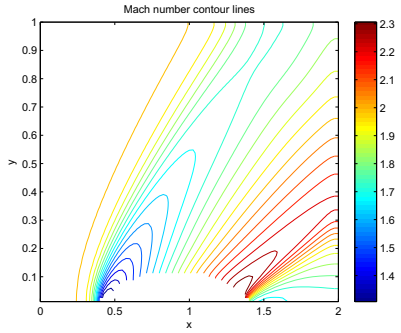
Results for the Cartesian Grid Method

In Fig. 2(a) and Fig. 2(b) we present Mach number and density contours for external flow at $M_\infty = 2$ over the circular arc airfoil. We compute only the flow over the upper half with a height of 10 $\%$ chord. The supersonic flow is moving from left to right and hits the leading edge of the circular arc airfoil. A shock wave is produced at the leading edge. Near the trailing edge another shock wave turns the flow back in the $x-$direction.

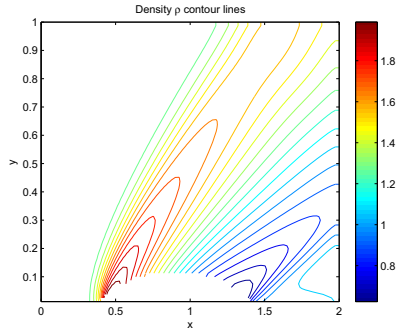
In Fig. 3(a) and Fig. 3(b) we present contours of the velocity components u and v.

Fig. 4(a) shows the behavior of the pressure for external flow over a circular arc airfoil.

In Fig. 4(b) the l_2-norm of the density change, which is equivalent to the density residual, indicated that the density is becoming stationary after around $n = 2500$ iterations. We have not yet explored the kink in the convergence curve leading to slower convergence.

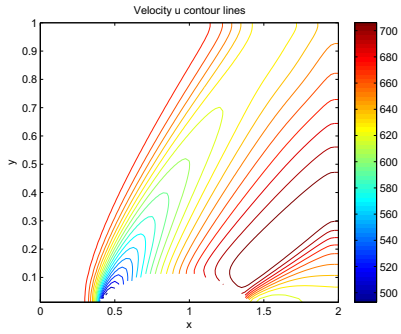


(a) Mach number contours for supersonic external flow over a circular arc airfoil.

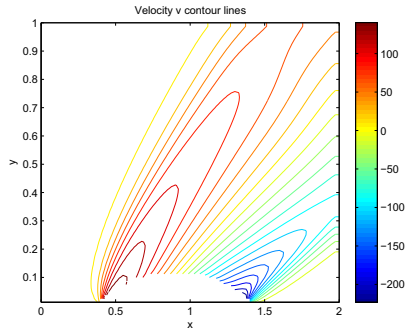


(b) Density contours for supersonic external flow over a circular arc airfoil.

Figure 2: Left: Mach number contours for first order method. Right: Density contours for first order method .

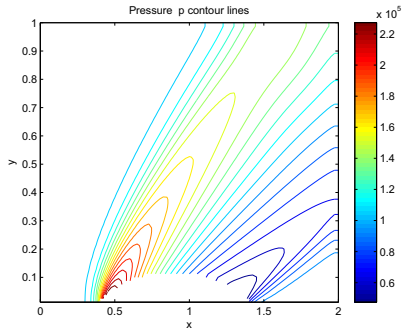


(a) Velocity u component contours for supersonic external flow over a circular arc airfoil.

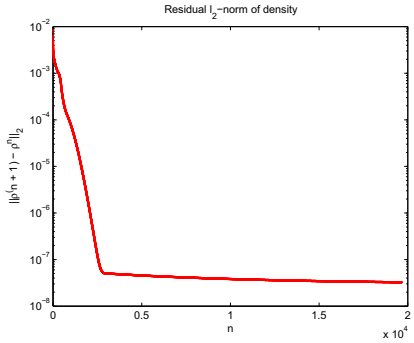


(b) Velocity v component contours for supersonic external flow over a circular arc airfoil.

Figure 3: Left: Velocity component u contours for first order method. Right: Velocity v component contours for first order method.

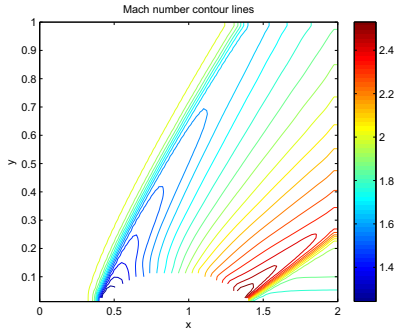


(a) Pressure contours for supersonic external flow over a circular arc airfoil.

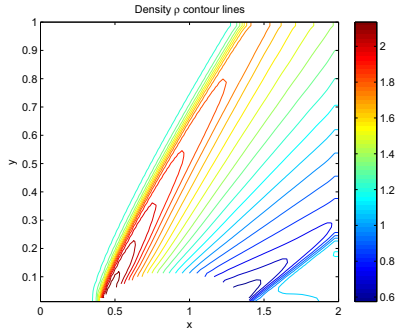


(b) Residual l_2-norm of density for supersonic flow over a circular arc airfoil.

Figure 4: Left: Pressure contours for first order method. Right: Residual l_2-norm of density for first order method.

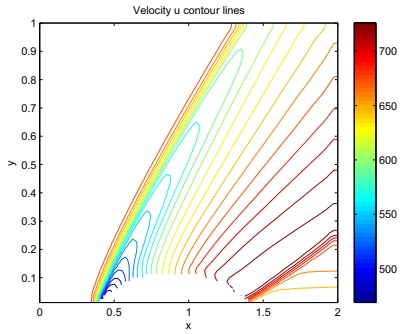


(a) Mach number contours for supersonic external flow over a circular arc airfoil.

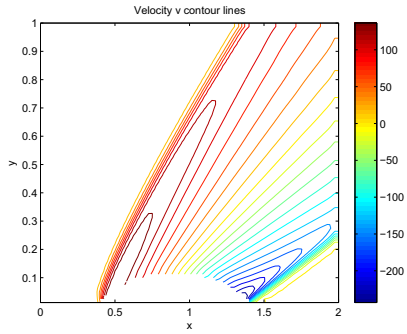


(b) Density contours for supersonic external flow over a circular arc airfoil.

Figure 5: Left: Mach number contours for MUSCL with the minmod limiter. Right: Density contours for MUSCL with the minmod limiter.

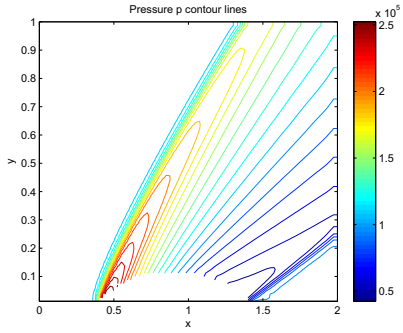


(a) Velocity component u contours for supersonic external flow over a circular arc airfoil.



(b) Velocity component v contours for supersonic external flow over a circular arc airfoil.

Figure 6: Left: Velocity component u contours for MUSCL with the minmod limiter. Right: Velocity v component for MUSCL with the minmod limiter.



(a) Pressure contours for supersonic external flow over a circular arc airfoil.

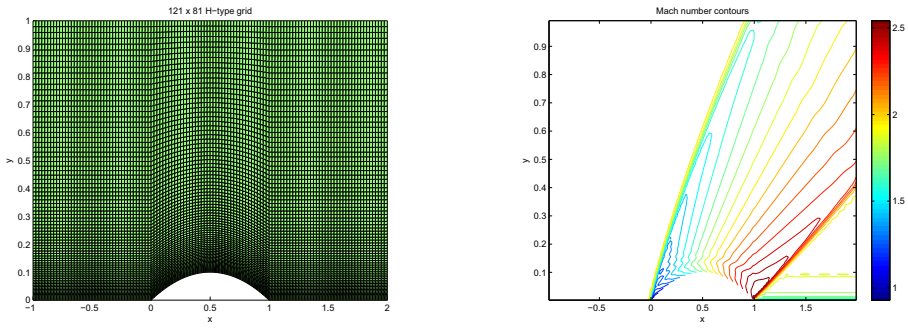
Figure 7: Pressure contours for MUSCL with the minmod limiter.

Now we turn to the results obtained with the higher order method. We use the local Lax-Friedrichs method for spatial discretization with MUSCL and minmod limiter and TVD RK3 method for time integration. We obtain sharp shock waves at the leading and trailing edges , cf. Fig. 5(a) and Fig. 5(b) for Mach number and density contours. Whereas the first order results in Figs. 2, 3, 4(a) yield a detached shock like for blunt body flow, the second order MUSCL results (first order at extrema) indicate an attached oblique shock in agreement with the corresponding body-fitted grid results, cf. below.

In Fig. 6(a) and Fig. 6(b) we present results over the circular arc airfoil for the velocity components u and v for higher order method for external flows. The shock wave at the leading and trailing edges is sharper than for the first order method, cf. Fig. 3.

Also the pressure contours with the higher order method in Fig. 7(a) indicate sharper leading and trailing edge shocks than the first order method, cf. Fig. 4(a).

Results for the Standard Body-Fitted Grid Method



(a) Standard body-fitted grid.

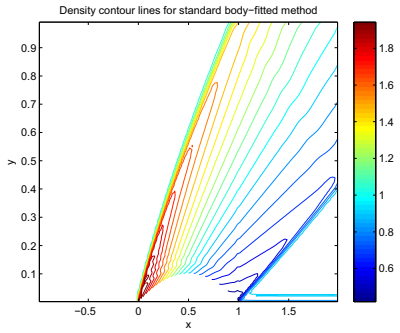
(b) Mach number contours for supersonic external flow over a circular arc airfoil.

Figure 8: Left: Standard body-fitted grid of a circular arc airfoil. Right: Mach number contours for standard body-fitted method.

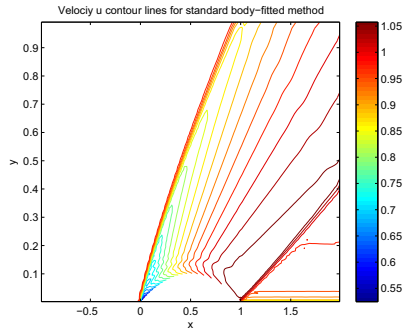
The grid spacing in the $x-$direction is the same as for the Cartesian grid method, i.e. $\Delta x = \frac{1}{40}$. The grid points are clustered towards $y = 0$ and the circular arc airfoil ($\Delta y_{min} = 0.0037$ at $y = 0$ and $\Delta y_{max} = 0.0208$ at $y = 1$). The body-fitted grid for the the domain has 121×81 grid points (cf. Fig. 8(a)), because the left and right boundaries are one chord length away from the leading and trailing edges respectively.

The cell-centered finite volume method with Roe's approximate Riemann solver is used, which is less diffusive than the local Lax-Friedrichs method. The MUSCL approach is applied to the primitve variables, not in the conservative variables as in (13) and (14). While the minmod limiter is used for density and pressure, the less diffusive van Albada limiter is employed for the velocity components. Harten's entropy fix is employed to enforce the entropy condition [20, 21].

The contours for Mach number, density, velocities components u and v and pressure contours in Fig. 8(b), Fig. 9 and Fig. 10 for a same supersonic flow over the circular arc airfoil show that the standard-body fitted grid method yields sharper shocks at the leading and trailing edges. Note that density, velocities components u and v and pressure are non-dimensionalized with respect to reference density ρ_∞, velocity u_∞ and pressure $\rho_\infty u_\infty^2$, respectively. But otherwise, there is

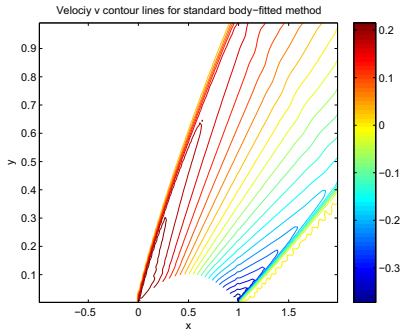


(a) Density contours for supersonic external flow over a circular arc airfoil.

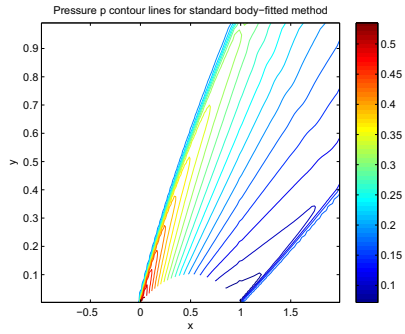


(b) Velocity component u contours for supersonic external flow over a circular arc airfoil.

Figure 9: Left: Density contours for standard body-fitted method. Right: Velocity component u contours for standard body-fitted method.



(a) Velocity component v contours for supersonic external flow over a circular arc airfoil.



(b) Pressure contours for supersonic external flow over a circular arc airfoil.

Figure 10: Left: Velocity contours for standard body-fitted method. Right: Pressure contours for standard body-fitted method.

good qualitative agreement between the body-fitted results and the Cartesian grid method with MUSCL, cf. Fig. 5, Fig. 6 and Fig. 7(a). Note that the wiggles behind the leading edge shock for Mach number M, density ρ and velocity u are not present for the Cartesian grid method.

Conclusions

We investigated the Cartesian grid method for the 2D compressible Euler equations. We used the first order local Lax-Friedrichs method for spatial discretization and increased the accuracy by MUSCL with the minmod limiter. For time integration we used the first order explicit Euler method and the third order TVD RK3 method. At the immersed solid boundary we employed a simplified ghost point treatment in which the fluid point F is chosed on the vertical grid line through G adjacent to the boundary as the mirror point. We tested the 2D code for supersonic external flow over a circular arc airfoil. The comparison between the Cartesian grid method and the standard body-fitted grid method indicated good qualitative agreement.

Acknowledgement

The current research has been funded by the Higher Education Commission (HEC) of Pakistan.

References

[1] Almgren, A. S., Bell, J. B., Colella, P. & Marthaler, T. A Cartesian grid projection method for the incompressible Euler equations in complex geometries. *SIAM Journal on Scientific Computing* **18**, 1289–1309 (1997).

[2] Palma, P. D., de Tullio, M. D., Pascazio, G. & Napolitano, M. An immersed-boundary method for compressible viscous flows. *Computers & Fluids* **35**, 693–702 (2006).

[3] Marshall, D. D. & Ruffin, S. M. . A new inviscid wall boundary condition treatment for boundary Cartesian grid method. *AIAA 2004-0583 42nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada* (2004).

[4] Udaykumar, H. S., Krishnann, S. & Marella, S. V. Adaptively refined parallelisded sharp interface Cartesian grid method for three dimensional moving boundary problem. *International Journal of Computational Fluid Dynamics* **23**, 1–24 (2009).

[5] Uzgoren, E., Sim, J. & Shyy, W. Marker based 3-D adapitve Cartesian grid method for multiphase flow around irregular geometries. *Communications in Computational Physics* **5**, 1–41 (2009).

[6] Wang, Z., Fan, J. & Cen, K. Immersed boudnary method for the simulations of 2D viscous flow based on vorticity-velocity formulations. *Journal of Computational Physics* **228**, 1504–1520 (2009).

[7] Mittal, R. & Iaccarino, G. Immersed boundary methods. *Annual Review of Fluid Mechanics* **37**, 239–261 (2005).

[8] Kirshman, D. & Liu, F. Cartesian grid solution of the Euler equations using a gridless boundary condition treatment. *AIAA Paper 2003-3974* (2003).

[9] Peng, Y.-F., Mittal, R., Sau, A. & Hwang, R. R. Nested Cartesian grid method in incompressible viscous fluid flow. *Journal of Computational Physics* **229**, 7072–7101 (2010).

[10] Quirk, J. J. An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two -dimensional bodies. *Computers & Fluids* **23**, 125–142 (1994).

[11] Ingram, D., Causon, D. & Mingham, C. Developments in Cartesian cut cell methods. *Mathematics and Computers in Simulation* **61**, 561572 (2003).

[12] Forrer, H. & Jeltsch, R. A higher-order boundary treatment for Cartesian-grid methods. *Journal of Computational Physics* **140**, 259–277 (1998).

[13] S.Peskin, C. Flow pattern around heart valves: A numerical method. *Journal of Computational Physics* **10**, 252–271 (1972).

[14] de Tullio, M. D., Palma, P. D. D., Iaccarino, G., Pascazio, G. & Napolitano, M. An immersed boundary method for compresible flows using local grid refinement. *Journal of Computational Physics* **225**, 2098–2117 (2007).

[15] Berger, M. J. & LeVeque, R. J. A rotated difference scheme for Cartesian grids in complex geometries. *AIAA Paper CP-91-1602* **CP-91-1602**, 1–9 (1991).

[16] Pember, R. B., Bell, J. B., Colella, P., Curtchfield, W. Y. & Welcome, M. L. An adaptive Cartesian grid method for unsteady compressible flow in irregular regions. *Journal of Computational Physics* **120**, 278–304 (1995).

[17] Coirier, W. J. & Powell, K. G. An accuracy assessment of Cartesian-mesh approaches for the Euler equations. *Journal of Computational Physics* **117**, 121–131 (1995).

[18] Colella, P., Graves, D. T., Keen, B. J. & Modiano, D. A Cartesian grid embedded boundary method for hyperbolic conservation laws. *Journal of Computational Physics* **211**, 347–366 (2006).

[19] Sjögreen, B. & Petersson, N. A. A Cartesian embedded boundary method for hyperbolic conservation laws. *Communications in Computational Physics* **2**, 1199 – 1219 (2007).

[20] Harten, A. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics* **49**, 357–393 (1983).

[21] Müller, B. Upwind relaxation method for hyperbolic flow simulation. *Technical Report DLR-FB Göttingen, Germany,* 91–36 (1991).

Paper 4

Cartesian Grid Method for the Compressible Euler Equations

M. Asif Farooq and B. Müller

In Proc. Finite Volumes for Complex Applications VI: Problems & Perspectives, Springer-Verlag, Prague, Czech Republic, pp. 449-456, 2011.

Is not included due to copyright

Paper 5

Accuracy Assessment of the Cartesian Grid Method for Compressible Inviscid Flows Using a Simplified Ghost Point Treatment

M. Asif Farooq and B. Müller

RAKENTEIDEN MEKANIIKKA (Journal of Structural Mechanics), Vol. 44, No. 3, pp. 279-291, 2011.

Rakenteiden Mekaniikka (Journal of Structural Mechanics)
Vol. 44, No 3, 2011, pp. 279 – 291

Accuracy assessment of the Cartesian grid method for compressible inviscid flows using a simplified ghost point treatment

M. Asif Farooq and B. Müller

Summary. We introduce a new approach to treat ghost points near embedded boundaries to solve the 2D compressible Euler equations on a Cartesian grid. Solid wall boundary conditions are imposed by our new approach called simplified ghost point treatment for compressible inviscid flows with embedded boundaries. In the simplified ghost point treatment, we assume the solid boundary to lie in the middle between two grid points in the y-direction. Symmetry conditions are used to determine density, pressure, wall tangential, and wall normal velocity components at the ghost points. A cell-vertex finite volume formulation has been used to calculate transonic internal flows over a circular arc bump in a channel.

Key words: compressible Euler equations, Cartesian grid method, simplified ghost point treatment, cell-vertex finite volume method

Introduction

In Computational Fluid Dynamics (CFD) two methods are popular in the scientific community. One is the standard body fitted grid method and the other is the Cartesian grid method. In the former solid boundaries and grid lines conform to each other but in the latter this is not the case. Standard body fitted grid methods both with structured and unstructured grids successfully solved many complex problems in the past [1–4]. But these methods involve a numerical complexity which is sometime discouraging for the scientific community. The popularity of the Cartesian grid method [5–9] is due to its simplicity, faster grid generation, simpler programming, lower storage requirements, lower operation count, and easier post processing compared to body fitted structured and unstructured grid methods. The Cartesian grid method is also advantageous in constructing higher order methods. A disadvantage of this method shows up when it is applied to complex domains. At curved boundaries the cells are not rectangular and these cut-cells create a problem in time-stepping for a scheme to be implemented. The solution to this cut-cell problem is to merge these cut-cells with neighboring cells. The main disadvantage of the cut-cell approach is the complexity of the method in which fluxes are calculated at the interfaces near the embedded boundary.

We avoided the cut-cell approach [10] and opt for the ghost point treatment at the boundary [11]. The main motivation has been the simplicity of the ghost point treatment over the more complicated cut-cell approach. The advantage of the present simplified ghost point approach resides in its ability of implementing the method with ease and extending it to a higher order method. In the ghost point treatment, we divide our

domain into three types of points: fluid, ghost and solid points. Only fluid and ghost points are treated in the scheme, and solid points are cut out of the present computation.

In [11] we successfully compared supersonic external flow results over a circular arc airfoil with standard body fitted grid results. In this paper, we further analyse the simplified ghost point treatment for the 2D compressible Euler equations to simulate transonic internal flow. In the simplified ghost point treatment, we assume the solid boundary to lie in the middle between two grid points in the y-direction. Our assumption is based on the fact that the normal to the wall boundary in slender bodies can be assumed to lie in the y-direction. Symmetry conditions are used to determine density, pressure, wall tangential, and wall normal velocity components at the ghost points. We employ the local Lax-Friedrichs (lLF) method for the spatial discretization. The local Lax-Friedrichs method is chosen over other numerical methods like Roe's method because of its simplicity. We apply the Monotone Upstream-centered Schemes for Conservation Laws (MUSCL) approach with the minmod limiter to get a second order method except for extrema. For time integration we use the third order Total Variation Diminishing Runge-Kutta (TVD RK3) method with the MUSCL approach. As a test case, we consider transonic internal flow over a circular arc bump in a channel and solve the 2D compressible Euler equations by time stepping to the steady state. We compute a shock in the channel with the circular arc bump and compare these results with the results presented in the literature. We need a larger number of grid points than standard body fitted grid methods to obtain results comparable with [12].

In the following section, we discuss the governing equations, i.e. for 2D compressible Euler equations. In the next sections, we outline Boundary conditions, Discretization and Approximation of boundary conditions. Simplified ghost point treatment and Results are discussed in the next two sections. Conclusions are drawn at the end.

2D compressible Euler equations

The 2D compressible Euler equations in conservative form read

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0, \tag{1}$$

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix}, F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho E + p)u \end{bmatrix}, G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho E + p)v \end{bmatrix}, \tag{2}$$

with ρ, u, v, E, and p denoting density, velocity components in x and y-directions, total energy per unit mass and pressure, respectively. We employ the equation of state for perfect gas

$$p = (\gamma - 1)(\rho E - \frac{1}{2}\rho(u^2 + v^2)), \tag{3}$$

where γ is the ratio of specific heats. We consider $\gamma = 1.4$ for air.

Boundary conditions

We consider the following boundaries: (1) subsonic inflow, (2) Subsonic outflow and (3) lower and upper wall. At the subsonic inflow $x = x_a$ in Fig. 1, three flow variables,

e.g. total enthalpy, entropy and velocity component v, are given as Dirichlet boundary conditions and one numerical boundary condition is used. At the subsonic outflow $x = x_b$, we prescribe one Dirichlet boundary condition, e.g. atmospheric pressure, and three numerical boundary conditions. The straight parts of the lower and upper walls are treated as symmetry boundaries. Symmetry boundary conditions near $y = y_c$ imply:

$$(\rho, \rho u, \rho E)(x, y_c + y, t) = (\rho, \rho u, \rho E)(x, y_c - y, t)$$

and

$$\rho v(x, y_c + y, t) = -\rho v(x, y_c - y, t)$$

Similar symmetry boundary conditions are imposed near $y = y_d$. On the curved part of the lower wall, local symmetry boundary conditions with respect to the wall are employed.

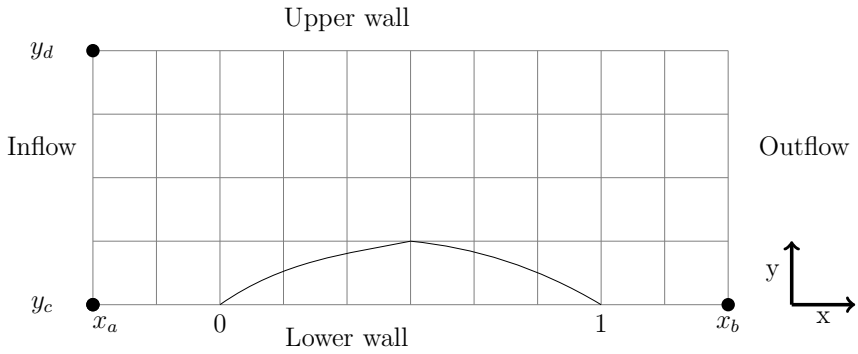


Figure 1. Computational domain and Cartesian grid for channel flow with a circular arc bump.

Discretization

Spatial discretization

We assume Cartesian coordinates (x, y), where $x_a \leq x \leq x_b$, $y_c \leq y \leq y_d$ and a Cartesian grid with $(N_x + 1) \times (N_y + 1)$ grid points. The grid points of the Cartesian coordinates are denoted by (x_i, y_j), where $x_i = x_a + i\Delta x$, with $i = 0, 1, ..., N_x$, $\Delta x = (x_b - x_a)/N_x$ and $y_j = y_c + j\Delta y$, with $j = 0, 1, ..., N_y$, $\Delta y = (y_d - y_c)/N_y$. The semi-discretization of the 2D compressible Euler equations (1) yields the following form

$$\frac{\mathrm{d}U_{i,j}}{\mathrm{d}t} = -\frac{F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}}{\Delta x} - \frac{G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}}}{\Delta y}. \tag{4}$$

$U_{i,j}$ is the approximation of the average of U in the cell $\Omega_{i,j} = [x_i - \Delta x/2, x_i + \Delta x/2] \times [y_j - \Delta y/2, y_j + \Delta y/2]$, i.e.

$$U_{i,j} \approx \frac{1}{\Delta x \Delta y} \int_{\Omega_{i,j}} U(x, y, t) \quad dxdy. \tag{5}$$

Thus, we have been using the cell-vertex finite volume method. We opted for this approach to have a conservative method with the unknowns at the grid points. In (4), $F_{i+\frac{1}{2},j}$ and

$G_{i,j+\frac{1}{2}}$ are numerical fluxes for the 2D compressible Euler equations and $U_{i,j}$ is also an approximation of the exact solution $U(x_i, y_j, t)$. The numerical fluxes $F_{i+\frac{1}{2},j}$ and $G_{i,j+\frac{1}{2}}$ are chosen as the local Lax-Friedrichs method for F and G, which are defined as follows

$$F^{lLF}_{i+\frac{1}{2},j} = \frac{1}{2}\left[F(U_{i,j}) + F(U_{i+1,j}) - \max(|u_{i+1,j}| + c_{i+1,j}, |u_{i,j}| + c_{i,j})(U_{i+1,j} - U_{i,j})\right], \quad (6)$$

$$G^{lLF}_{i,j+\frac{1}{2}} = \frac{1}{2}\left[G(U_{i,j}) + G(U_{i,j+1}) - \max(|v_{i,j+1}| + c_{i,j+1}, |v_{i,j}| + c_{i,j})(U_{i,j+1} - U_{i,j})\right], \quad (7)$$

where c is the speed of sound. Our choice of the local Lax-Friedrichs method has been motivated by the simplicity of its numerical flux functions (6) and (7) and its total variation diminishing (TVD) property for scalar conservation laws with convex flux functions. The time step restriction of the 2D compressible Euler equations is related to the Courant-Friedrichs-Levy (CFL) number. The CFL number is defined as

$$CFL = \Delta t \quad \max_{i,j}\left(\frac{\mathrm{sp}(A_1(U_{i,j}))}{\Delta x} + \frac{\mathrm{sp}(A_2(U_{i,j}))}{\Delta y}\right), \quad (8)$$

where $\mathrm{sp}(A_1(U_{i,j}))$ and $\mathrm{sp}(A_2(U_{i,j}))$ are the spectral radii of the Jacobian matrices $A_1 = \partial F(U)/\partial U$ and $A_2 = \partial G(U)/\partial U$, respectively. We choose $CFL = 0.5$ for the results of the 2D compressible Euler equations. In (6) we replace $U_{i,j}$ by $U^L_{i+\frac{1}{2},j}$ and $U_{i+1,j}$ by $U^R_{i+\frac{1}{2},j}$ using the MUSCL [13] approach with the minmod limiter to obtain higher order accuracy and also to avoid undesired oscillations. The extrapolated variables are defined as

$$U^L_{i+\frac{1}{2},j} = U_{i,j} + \frac{1}{2}\mathrm{minmod}(U_{i,j} - U_{i-1,j}, U_{i+1,j} - U_{i,j}), \quad (9)$$

$$U^R_{i+\frac{1}{2},j} = U_{i+1,j} - \frac{1}{2}\mathrm{minmod}(U_{i+2,j} - U_{i+1,j}, U_{i+1,j} - U_{i,j}), \quad (10)$$

where

$$\mathrm{minmod}(a,b) = \begin{cases} a & \text{if} \quad |a| \le |b| \quad \text{and} \quad ab > 0 \\ b & \text{if} \quad |b| < |a| \quad \text{and} \quad ab > 0 \\ 0 & \text{if} \quad ab \le 0 \end{cases} \quad (11)$$

$$= \mathrm{sign}(a)\max\{0, \min\{|a|, \mathrm{sign}(a)b\}\}$$

is the minmod limiter. The MUSCL approach is applied similarly to the numerical fluxes $G_{i,j+\frac{1}{2}}$ in (7).

Temporal Discretization

We choose the TVD RK3 method because it has a favorable stability domain. The TVD RK3 method [14] is given as

$$\begin{aligned} \mathbf{U}^{(1)} &= \mathbf{U}^n + \Delta t\mathbf{R}(U^n), \\ \mathbf{U}^{(2)} &= \frac{3}{4}\mathbf{U}^n + \frac{1}{4}\mathbf{U}^{(1)} + \frac{1}{4}\Delta t\mathbf{R}(U^{(1)}), \\ \mathbf{U}^{(n+1)} &= \frac{1}{3}\mathbf{U}^n + \frac{2}{3}\mathbf{U}^{(2)} + \frac{2}{3}\Delta t\mathbf{R}(U^{(2)}). \end{aligned} \quad (12)$$

where $R_{i,j} = -(F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j})/\Delta x - (G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}})/\Delta y$ is the residual of the 2D compressible Euler equations define in equation (4).

Approximation of boundary conditions

The inflow and outflow boundary conditions for internal subsonic flow are imposed by means of the local Lax-Friedrichs method (6) at the edge $(1+\frac{1}{2}, j)$ and $(N_x - \frac{1}{2}, j)$, respectively. Since the flow at the inlet is subsonic, there is one characteristic curve (the left going Mach line with an acoustic wave) coming from the interior and two characteristics (the pathline with the entropy and vorticity waves and the right going Mach line with an acoustic wave) are entering from the exterior. Therefore, we need one numerical and three physical boundary conditions. But at the inlet we simply use freestream variables $U_{1,j}(t) = U_\infty$ and let the approximate Riemann solver determine a flux approximation (6) with the correct wave information at the cell face $(1+\frac{1}{2}, j)$ between the boundary point $(1, j)$ and its eastern neighbor $(2, j)$ [15]. At the outlet the flow is subsonic and two characteristics are leaving the domain from the interior and one characteristic curve is entering from outside of the domain. So we need three numerical boundary conditions and one physical boundary condition. As before, we use freestream boundary conditions at the outlet $U_{N_x,j}(t) = U_\infty$ and let the approximate Riemann solver determine a flux approximation (6) with the correct wave information at the cell face $(N_x - \frac{1}{2}, j)$ between the boundary point (N_x, j) and its western neighbor $(N_x - 1, j)$ [15]. Since the ingoing characteristic variables at inlet and outlet are unchanged, the computation of the numerical fluxes at the adjacent cell faces is based on 1D non-reflecting boundary conditions in the x-direction [15]. In summary we use the following numerical boundary conditions at $x = x_a$

$$U_{1,j}(t) = U_\infty, \tag{13}$$

and at $x = x_b$

$$U_{N_x,j}(t) = U_\infty. \tag{14}$$

We emphasize that the uniform flow conditions (13) and (14) at $x = x_a$ and $x = x_b$, respectively, are not considered to be the natural boundary conditions, but suitable flow conditions to get the numerical fluxes approximately correct at the adjacent cell faces, cf. the discussion above. The symmetry boundary conditions are implemented by considering an extra line below $y = y_c$. There we use

$$U_{i,1}(t) = \mathrm{diag}(1, 1, -1, 1) U_{i,3}(t). \tag{15}$$

The symmetry boundary conditions are also implemented at an extra line above $y = y_d$ where we use

$$U_{i,Ny+1}(t) = \mathrm{diag}(1, 1, -1, 1) U_{i,Ny-1}(t). \tag{16}$$

Simplified ghost point treatment at embedded boundary

In Fig. 2 we show the flagging strategy. We flag the ghost and solid points by assigning them 0 and -1 values, respectively. The fluid points are assigned values equal to 1. In Fig. 3 we show a simplified ghost point treatment at the solid boundary [11]. A ghost point is denoted by G. In the simplified ghost point treatment we consider the fluid point F on the vertical grid line through G adjacent to the boundary as the mirror point. Then, we assume the boundary is in the middle between ghost and fluid points. The density ρ, pressure p, and the tangential velocity component at the ghost point are symmetric with respect to the solid boundary and therefore directly determined by their values at

F. The normal velocity component u_n at the ghost point is anti-symmetric and thus gets the negative value of u_n at F. The mathematical description of this strategy is given as

$$\rho_G = \rho_F, p_G = p_F, u_G = u_F - 2(n_1 u_F + n_2 v_F)n_1, v_G = v_F - 2(n_1 u_F + n_2 v_F)n_2, \quad (17)$$

where n_1 and n_2 are the x-and y-components of the outer unit normal $\mathbf{n}$ of the boundary at the intersection with the grid line between F and G.

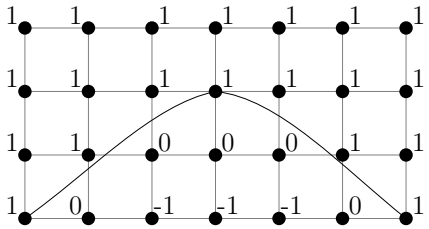


Figure 2. Flagging strategy for fluid (1), ghost (0) and solid points (-1).

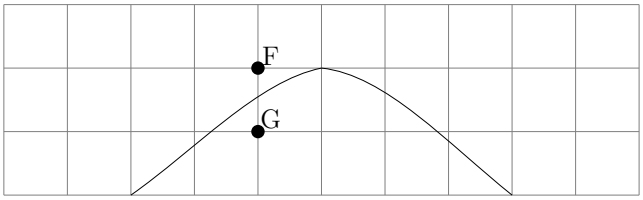


Figure 3. Simplified ghost point treatment over a circular arc bump.

If the embedded solid boundary is in the middle between ghost point G and fluid point F, then the expected normal velocity component at the solid boundary wall (index w below) is zero because the ghost point treatment (17) yields:

$$(u_n)_w = \frac{1}{2}((u_n)_F + (u_n)_G) = 0. \quad (18)$$

If the embedded solid boundary is not in the middle between G and F as in Fig. 4, we do not get zero for the arithmetic average of the normal velocity components, but

$$(u_n)_w = \frac{1}{2}((u_n)_F + (u_n)_G) = (u_n)_F(1 - \frac{2(y_F - y_w)}{\Delta y}). \quad (19)$$

So, we get an error of order $O(y_F - y_w) = O(\Delta y)$, unless $y_F - y_W = \Delta y/2$ when (19) corresponds to (18). For the tangential velocity component, density and pressure, we get for the arithmetic averages at the solid boundary wall, e.g. for pressure, from the ghost point treatment (17)

$$p_w = \frac{1}{2}(p_F + p_G) = p_F. \quad (20)$$

Using Taylor's expansion around the wall, we get

$$p_F = (p_w)_{exact} + O(y_F - y_w) = (p_w)_{exact} + O(\Delta y) \quad (21)$$

Thus with (20) and (new formula for p_F), we see that the simplified ghost point treatment (17) leads to a first order approximation at the embedded solid boundary when assuming that the boundary valves are described by the arithmetic averages at the ghost and fluid points. We see that the normal velocity component is only zero, if the embedded boundary is in the middle between ghost and fluid points.

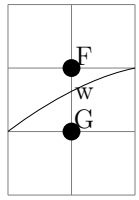


Figure 4. Detail of simplified ghost point treatment near embedded boundary.

Results

Subsonic flow in a channel with a circular arc bump flows from left to right. We use the CFL number 0.5 and 241 × 241 grid points. We assume the lengths of the domain in x-and y-directions are $x_b - x_a = 3$ m and $y_d - y_c = 1$ m, respectively. The height of the circular arc bump is 10 % of its chord length which is assumed to be 1 m. The subsonic upstream flow conditions are chosen as

$$M_\infty = 0.7, \quad p_\infty = 140510 \quad \text{Pa}, \quad T_\infty = 316.2240 \quad \text{K} \tag{22}$$

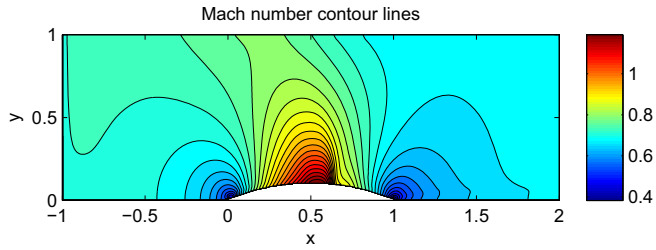


Figure 5. Mach number contours for channel flow over a circular arc bump with 241 × 241 grid points.

In Fig. 5 and Fig. 6, we compare the Mach number contours for internal flow over the circular arc bump with Hirsch [12]. In Fig. 5, we clearly see a shock wave near the downward slope of the circular arc bump. We also see that the our results are lower than Hirsch's results with a structured 65 × 33 grid. The subsonic region in our result shown in Fig. 5 is smaller and further upstream than Hirsch's result in Fig. 6. The reason for getting lower Mach numbers and the supersonic region and the shock wave shifted in the upstream direction is probably due to a relatively coarser grid. As we have shown in [11], we need a larger number of grid points compared to body-fitted grid methods. This observation is also confirmed by the results presented in the literature [6]. Probably,

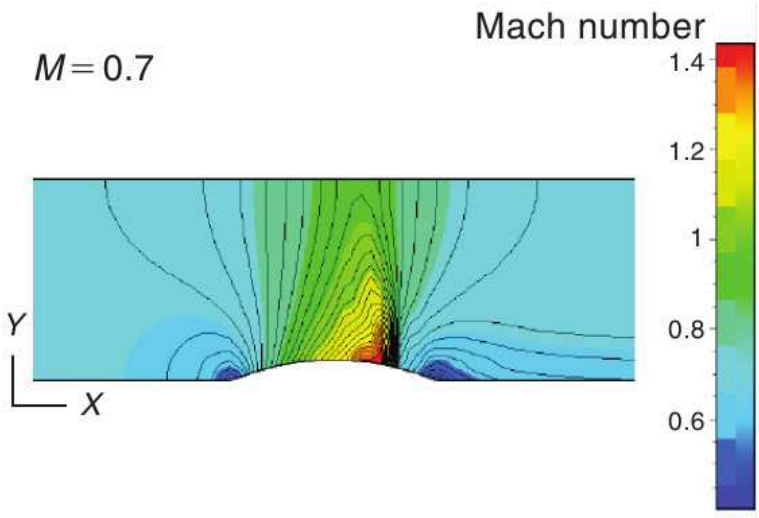


Figure 6. Mach number contours for channel flow over a circular arc bump from [12].

we could have achieved more accurate and thus less diffusive results with fewer grid points by using e.g. Roe's method instead of the local Lax-Friedrichs method (6) and (7).

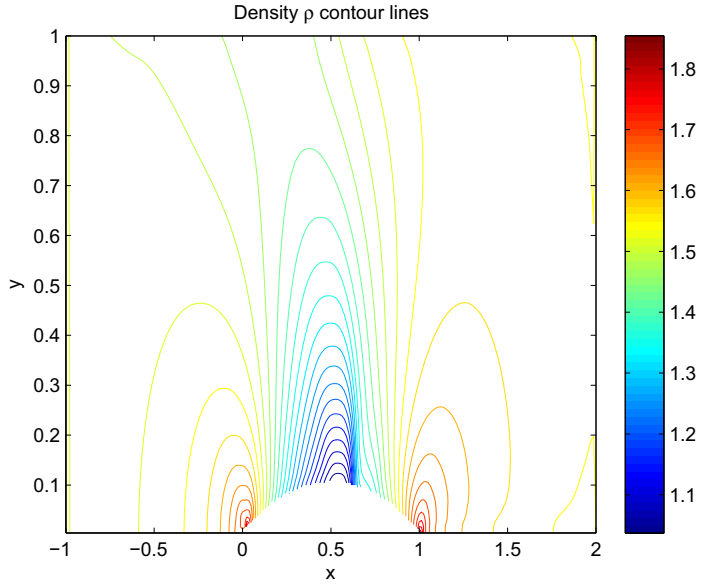


Figure 7. Density contours for channel flow over a circular arc bump with 241 × 241 grid points.

In Figs. 7, 8 and 9, we present the density ρ, velocity component u and pressure p contours for internal flow over the circular arc bump. In Figs. 7, 8 and 9, we see again the shock wave near the downward slope of the circular arc bump. The convergence history for our method based on the local Lax-Friedrichs method, MUSCL and TVD RK3 is

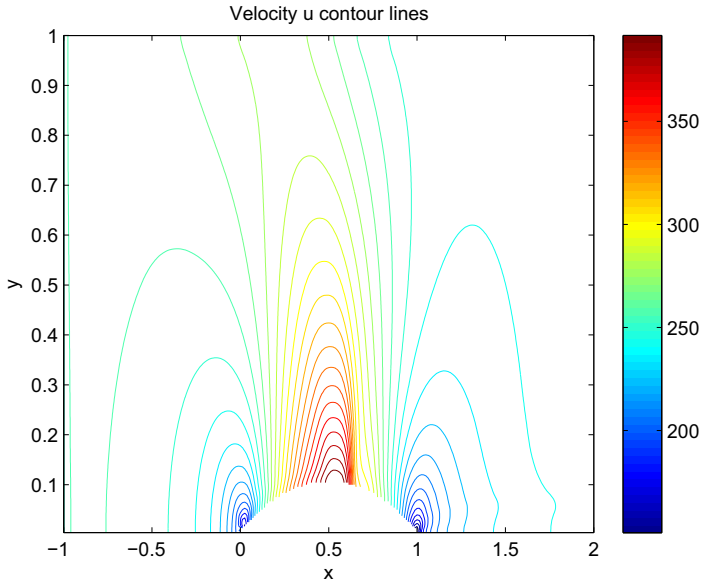


Figure 8. Velocity component u contours for channel flow over a circular arc bump with 241 × 241 grid points.

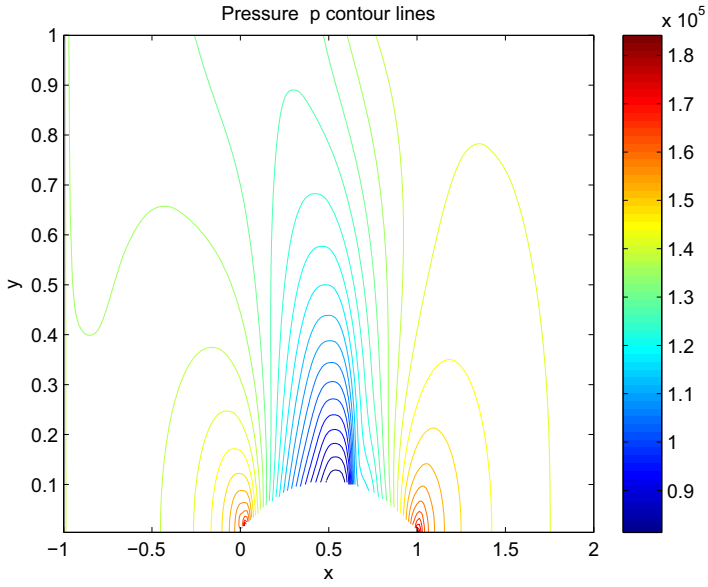


Figure 9. Pressure contours for channel flow over a circular arc bump with 241 × 241 grid points.

presented in Fig. 10. For the current results with the 241 × 241 grid, the steady state is reached at $n = 20000$. The steady state is reached at around $n = 10000$ for the 121 × 121 grid (not shown).

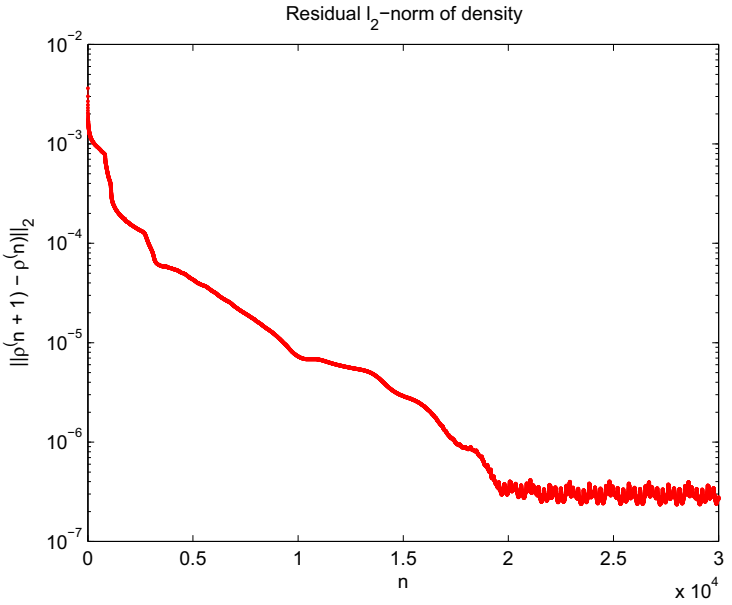


Figure 10. Residual of density for channel flow over a circular arc bump with 241 × 241 grid points.

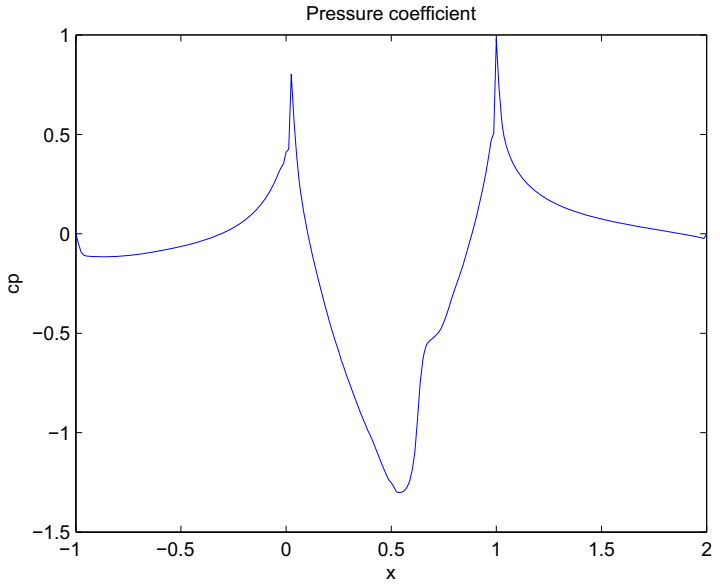


Figure 11. Pressure coefficient for channel flow over a circular arc bump with 241 × 241 grid points at $y = 0$ and on the bump surface.

In Fig. 11, we show the pressure coefficient c_p for internal flow over the circular arc bump on the x-axis $y = 0$ and on the bump surface. The pressure increases from the inflow boundary to the leading edge of the circular arc bump. The pressure decreases due

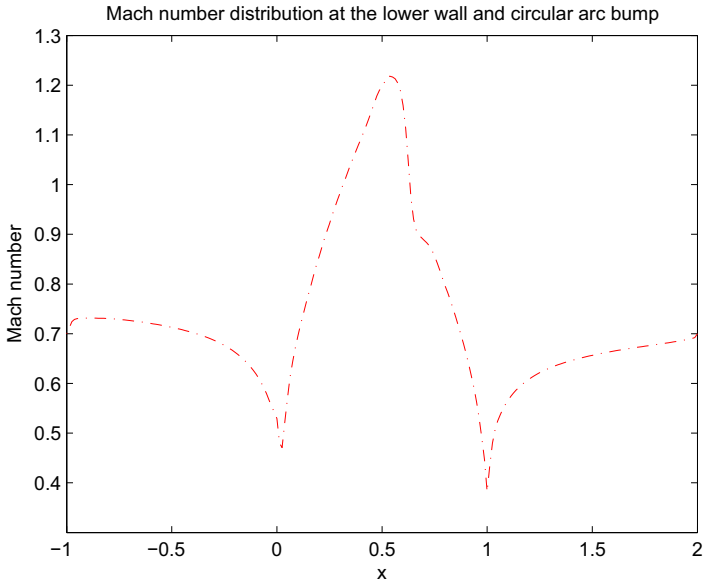


Figure 12. Mach number distribution for channel flow over a circular arc bump with 241 × 241 grid points at $y = 0$ and on the bump surface.

to the acceleration over the circular arc bump and reaches a minimum downstream of the maximum point of the bump. The shock is clearly visible in Fig. 11 and much sharper than with a 121 × 121 grid (not shown). We observe a pressure increase over the shock and towards the trailing edge and a subsequent decrease towards the outflow boundary. In Fig. 12 we show the distribution of the Mach number M for internal flow over the circular arc bump. We observe supersonic flow between $x \approx 0.3$ and $\approx 0.6m$. Again, the shock at $x \approx 0.6m$ is clearly visible.

Conclusions

In this work, we introduced a new approach to account for ghost points at embedded boundaries for the Cartesian grid method. At an embedded solid boundary we employ a simplified ghost point treatment in which the fluid point F is chosen on the vertical grid line through ghost point G adjacent to the boundary as the mirror point. We investigate the accuracy of the Cartesian grid method for the 2D compressible Euler equations simulating transonic flow over a circular arc bump. We use the local Lax-Friedrichs method and the MUSCL approach for spatial discretization. For time integration we use the third order TVD Runge-Kutta method. The Cartesian grid method needs a larger number of grid points than body fitted grid methods to get accurate results. In our Cartesian grid results, the location of the shock is upstream compared to the body-fitted grid result. Further grid refinement is expected to improve the shock location and the quantitative results. This expectation is supported by our observation that the simulated Mach numbers and the supersonic region get larger and closer to the body-fitted grid results when going from a coarse to a finer grid.

Acknowledgements

The current research has been funded by the Higher Education Commission (HEC) of Pakistan.

References

[1] A. Jameson, W. Schmidt and E. Turkel. Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes. *AIAA Paper 81-1259*, 1981.

[2] E. Dick. A flux-difference splitting method for steady Euler equations. *Journal of Computational Physics*, 76(1): 19–32, 1988.

[3] E. Issman, G. Degrez and H. Deconinck. Implicit upwind residual-distribution Euler and Navier-Stokes solver on unstructured meshes. *AIAA Journal*, 34(10): 2021–2028, 1996.

[4] V. Venkatakrishnan and D.J. Mavriplis. Implicit method for the computation of unsteady flows on unstructured grids. *Journal of Computational Physics*, 127(2): 380–397, 1996.

[5] H . S. Udaykumar, S. Krishmann and S. V. Marella. Adaptively refined parallelised sharp interface Cartesian grid method for three dimensional moving boundary problem. *International Journal of Computational Fluid Dynamics*, 23(1): 1–24, 2009.

[6] B. Sjögreen and N. A. Petersson. A Cartesian embedded boundary method for hyperbolic conservation laws. *Communications in Computational Physics*, 2(6): 1199–1219, 2007.

[7] P. Colella, D. Graves, B. Keen and D. Modiano. A Cartesian grid embedded boundary method for hyperbolic conservation laws. *Journal of Computational Physics*, 211(1): 347–366, 2006

[8] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas and A. von Loebbecke. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics*, 227(10): 4825-4852, 2008.

[9] R. Löhner, J. Cebral, F. Camelli, S. Appanaboyina, J. Baum, E. Mestreau and O. Soto. Adaptive embedded and immersed unstructured grid techniques. *Comput.Methods Appl. Mech. Engrg*, 197(25-28): 2173–2197, 2008.

[10] D. Hartmann, M. Meinke and W. Schröder. A strictly conservation Cartesain cut-cell method for compressible viscous flows on adaptive grids *Comput.Methods Appl. Mech. Engrg*, 200(9-12) : 1038-1052, 2011.

[11] M. A. Farooq and B. Müller. A Cartesian grid method for compressible inviscid flows. *in: B. Skallerud, H. I. Andersson (Eds.), MekIT'11, Sixth National Conference on Computational Mechanics Trondheim, Norway, 23-24 May 2011*, 47–57, 2011.

[12] C. Hirsch. *Numerical computation of internal and external flows.* Elsevier, Amsterdam, 2007.

[13] B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov 's method. *Journal of Computational Physics*, 32(1): 101-136, 1979.

[14] S. Gottlieb and C. Shu. Total variation diminishing Runge-Kutta schemes. *Mathematics of Computations*, 67(221): 73–85, 1998.

[15] B. Müller. Influence of inlet and outlet boundary conditions in internal flow computations at low Mach numbers. *in: Desideri,J.-A., Hirsch,C., Le Tallec,P., Pandolfi,M., Periaux,J. (Eds.), Computational Fluid Dynamics '96, Proceedings of 2nd ECCOMAS Congress, Paris, 9-13 Sept. 1996*, John Wiley, Chichester, 637–643, 1996.

M. Asif Farooq, B. Müller
Department of Energy and Process Engineering
Norwegian University of Science and Technology (NTNU)
Kolbjørn Hejes Vei 2 No-7491, Trondheim, Norway
asif.m.farooq@ntnu.no, bernhard.muller@ntnu.no

Erratum of Paper 5

Erratum to “Accuracy assessment of the Cartesian grid method for compressible inviscid flows using a simplified ghost point treatment” [Rakenteiden Mekaniikka (Journal of Structural Mechanics) Vol. 44. No. 3, 2011, pp. 279-291].

Erratum to "Accuracy assessment of the Cartesian grid method for compressible inviscid flows using a simplified ghost point treatment" [Rakenteiden Mekaniikka (Journal of Structural Mechanics) Vol. 44. No. 3, 2011, pp. 279-291]

M. Asif Farooq and B. Müller

The purpose of this note is to correct Figs. 5, 7-12 and the discussion of those figures in our publication [1].

After detecting and correcting a bug in our 2D internal inviscid flow code, we produced the correct figures which we present below. We include the unchanged Fig. 6 from [2].

The new result only affect the discussion of the Mach number contours, i.e. the text between Figs. 5 and 7 in [1], the number of time steps for convergence, i.e. the numbers n = 20000 and n = 10000 on page 287 in [1], the location of supersonic flow, i.e. $x \approx 0.3$ and ≈ 0.6 m on page 289 in [1] and the shock location, i.e. $x \approx 0.6$ m before 'Conclusions' in [1]. Moreover, the last part of the conclusions in [1] is affected by the corrected results.

For completeness, we present the complete corrected sections 'Results' and 'Conclusions' here, although a large part of the text is unchanged as detailed above.

Results

Subsonic flow in a channel with a circular arc bump flows from left to right. We use the CFL number 0.5 and 241 $\times$ 241 grid points. We assume the lengths of the domain in x-and y-directions are $x_b - x_a = 3$ m and $y_d - y_c = 1$ m, respectively. The height of the circular arc bump is 10 % of its chord length which is assumed to be 1 m. The subsonic upstream flow conditions are chosen as

$$M_\infty = 0.7, \quad p_\infty = 140510 \quad \text{Pa}, \quad T_\infty = 316.2240 \quad \text{K} \tag{1}$$

In Fig. 5 and Fig. 6, we compare the Mach number contours for internal flow over the circular arc bump with Hirsch [2]. In Fig. 5, we clearly see a shock wave near the downward slope of the circular arc bump. We also see that our result with a 241 $\times$ 241 Cartesian grid is in good agreement with Hirsch's result with a structured 65 $\times$ 33 grid. The supersonic region in our result shown in Fig. 5 has a similar location as in Hirsch's result in Fig. 6. The maximum Mach numbers in our and Hirsch's results agree to lie above 1.4. Even with a 121 $\times$ 121 grid, we get similar results (not shown) as with the 241 $\times$ 241 grid. However, still the Cartesian grid method needs a larger number of grid points compared to body-fitted grid methods. This observation is also confirmed by our external inviscid flow simulations [3] and by results presented in the literature [4].

In Figs. 7, 8 and 9, we present the density ρ, velocity component u and pressure p contours for internal flow over the circular arc bump. In Figs. 7, 8 and 9, we see again the shock wave near the downward slope of the circular arc bump.

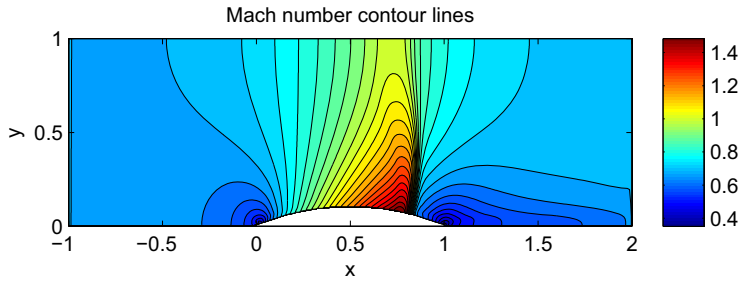


Figure 5. Mach number contours for channel flow over a circular arc bump with 241 × 241 grid points.

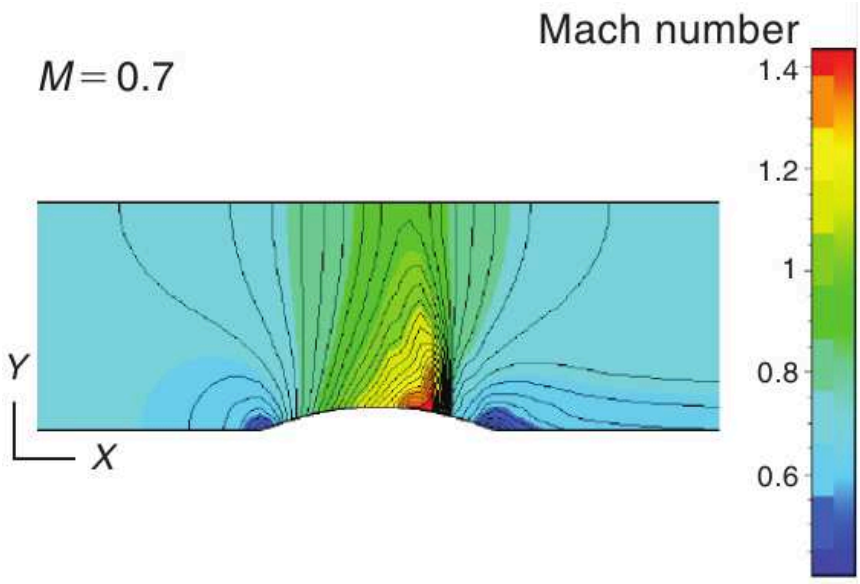


Figure 6. Mach number contours for channel flow over a circular arc bump from [2].

The convergence history for our method based on the local Lax-Friedrichs method, MUSCL and TVD RK3 is presented in Fig. 10. For the current results with the 241 × 241 grid, the steady state is reached at $n = 43000$. The steady state is reached at around

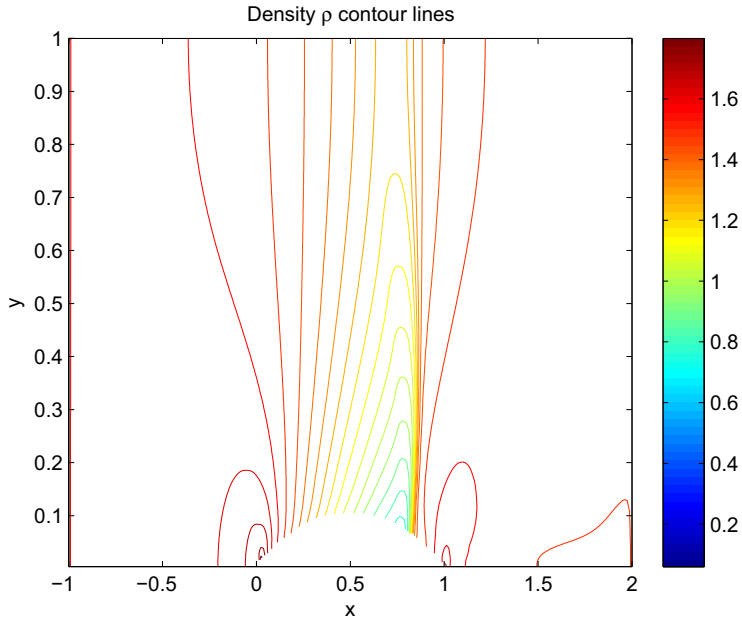


Figure 7. Density contours for channel flow over a circular arc bump with 241 × 241 grid points.

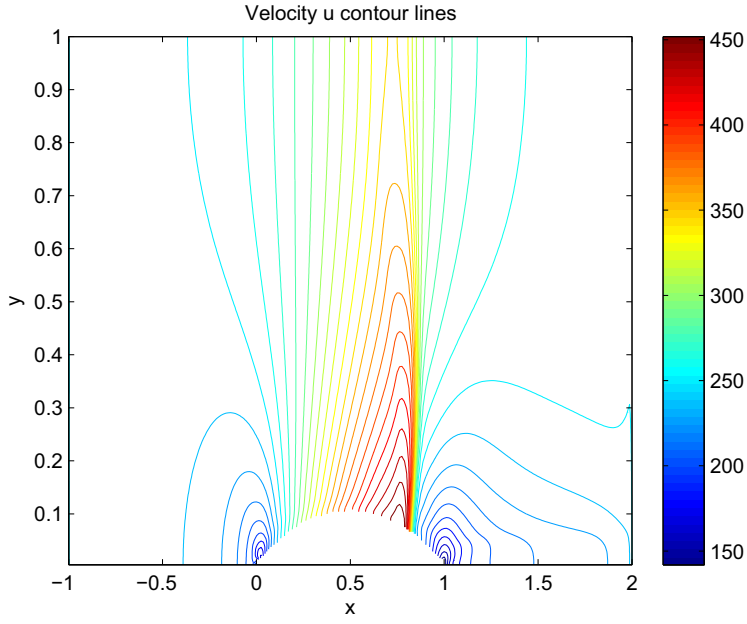


Figure 8. Velocity component u contours for channel flow over a circular arc bump with 241 × 241 grid points.

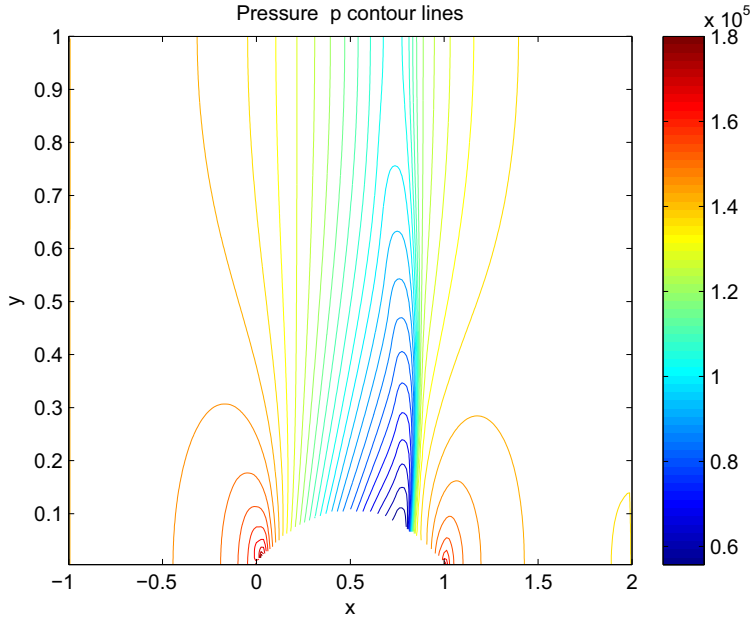


Figure 9. Pressure contours for channel flow over a circular arc bump with 241 × 241 grid points.

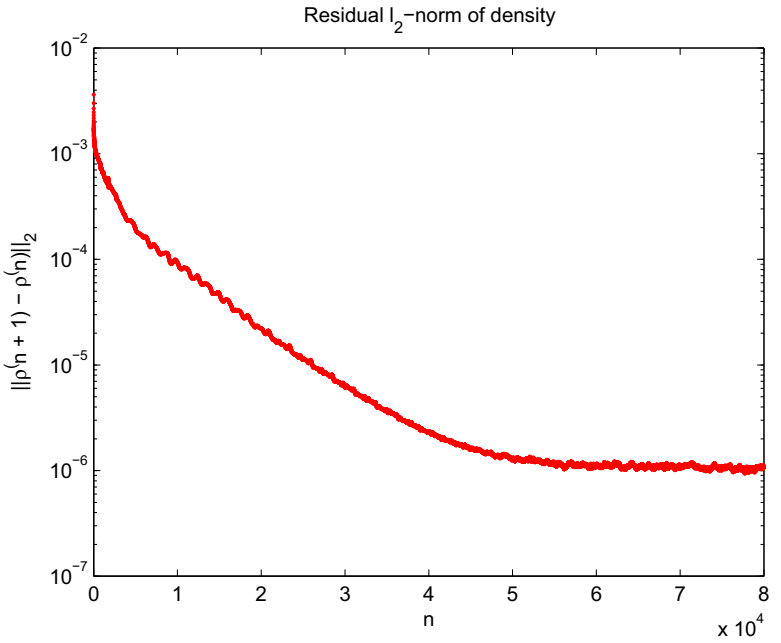


Figure 10. Residual of density for channel flow over a circular arc bump with 241 × 241 grid points.

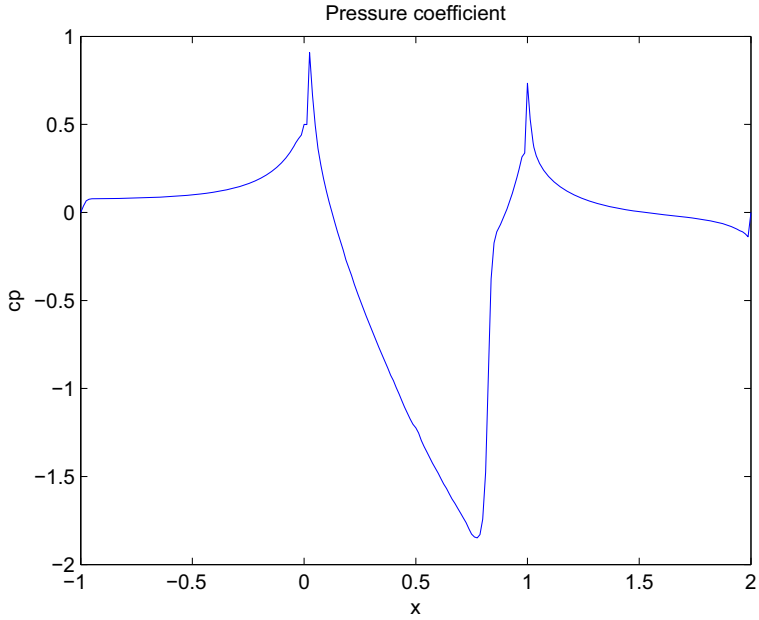


Figure 11. Pressure coefficient for channel flow over a circular arc bump with 241 × 241 grid points at $y = 0$ and on the bump surface.

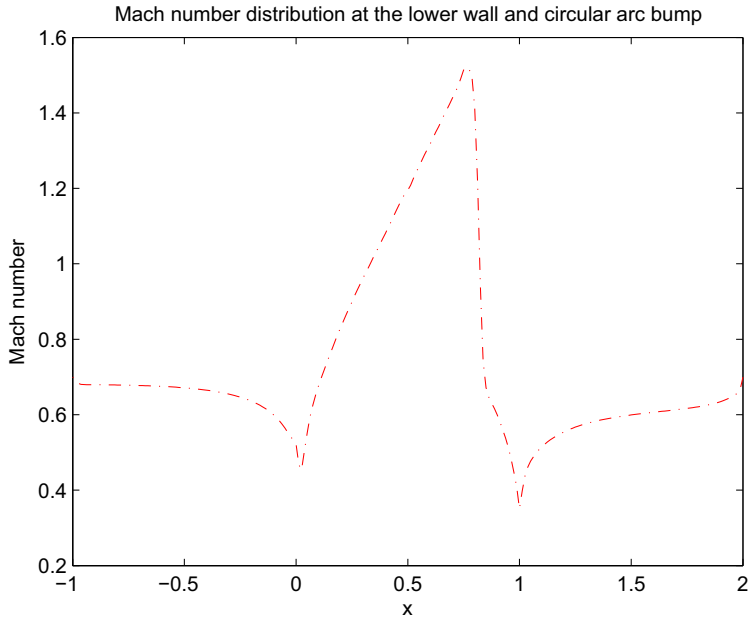


Figure 12. Mach number distribution for channel flow over a circular arc bump with 241 × 241 grid points at $y = 0$ and on the bump surface.

$n = 20000$ for the 121×121 grid (not shown).

In Fig. 11, we show the pressure coefficient c_p for internal flow over the circular arc bump on the x-axis ($y = 0$) and on the bump surface. The pressure increases from the inflow boundary to the leading edge of the circular arc bump. The pressure decreases due to the acceleration over the circular arc bump and reaches a minimum downstream of the maximum point of the bump. The shock is clearly visible in Fig. 11 and much sharper than with a 121×121 grid (not shown). We observe a pressure increase over the shock and towards the trailing edge and a subsequent decrease towards the outflow boundary. In Fig. 12 we show the distribution of the Mach number M for internal flow over the circular arc bump. We observe supersonic flow between $x \approx 0.35$ and ≈ 0.8 m. Again, the shock at $x \approx 0.8$ m is clearly visible.

Conclusions

In this work, we introduced a new approach to account for ghost points at embedded boundaries for the Cartesian grid method. At an embedded solid boundary we employ a simplified ghost point treatment in which the fluid point F is chosen on the vertical grid line through ghost point G adjacent to the boundary as the mirror point. We investigate the accuracy of the Cartesian grid method for the 2D compressible Euler equations simulating transonic flow over a circular arc bump. We use the local Lax-Friedrichs method and the MUSCL approach for spatial discretization. For time integration we use the third order TVD Runge-Kutta method. Our Cartesian grid results are in good agreement with body fitted grid results. However, The Cartesian grid method needs a larger number of grid points than body fitted grid methods to get accurate results.

References

[1] M. Asif Farooq and B. Müller. Accuracy assessment of the Cartesian grid method for compressible inviscid flows using a simplified ghost point treatment. *Rakenteiden Mekaniikka (Journal of Structural Mechanics)*, 44(3): 279-291, 2011.

[2] C. Hirsch. *Numerical computation of internal and external flows*. Elsevier, Amsterdam, 2007.

[3] M. A. Farooq and B. Müller. A Cartesian grid method for compressible inviscid flows. *in: B. Skallerud, H. I. Andersson (Eds.), MekIT'11, Sixth National Conference on Computational Mechanics Trondheim, Norway, 23-24 May 2011*, 47–57, 2011.

[4] B. Sjögreen and N. A. Petersson. A Cartesian embedded boundary method for hyperbolic conservation laws. *Communications in Computational Physics*, 2(6): 1199–1219, 2007.

M. Asif Farooq, B. Müller
Department of Energy and Process Engineering
Norwegian University of Science and Technology (NTNU)
Kolbjørn Hejes Vei 2 No-7491, Trondheim, Norway
asif.m.farooq@ntnu.no, bernhard.muller@ntnu.no

Paper 6

A Cartesian Grid Method for Compressible Flows to Compute Shock Waves

M. Asif Farooq and B. Müller

9th International Bhurban Conference on Applied Sciences & Technology, Islamabad, Pakistan, to appear in conference proceedings, 2012.

A Cartesian Grid Method for Compressible Flows to Compute Shock Waves

M. Asif Farooq* and B. Müller*

* Department of Energy and Process Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Abstract—**The accuracy of the Cartesian grid method has been explored for the 2D compressible Euler equations. We prescribe wall boundary conditions at ghost points near embedded boundaries by using local symmetry conditions. We test two ghost point treatments for supersonic flow over a wedge. In the standard ghost point treatment, the numerical solution at the corresponding mirror points is interpolated either linearly or quadratically. The accuracy of our second order node-centered finite volume method is independent of a linear or quadratic interpolation. In a simplified ghost point treatment, we consider the closest grid point in y-direction as mirror points of the ghost points. The simplified ghost point treatment exhibits lower or comparable mass flow error than the standard ghost point treatment. Moreover, the Cartesian grid and the body-fitted grid methods are applied to supersonic flow over a circular arc airfoil. The comparison of these two methods depicts the requirement of a larger number of grid points for the Cartesian grid method than the body-fitted grid method.**

I. INTRODUCTION

The advancement in computer technology paved the way for researchers and scientists to propose new computational techniques. Two methods are introduced in the literature which have gained popularity among researchers. One of these is the standard body-fitted grid method (structured and unstructured grid methods) and the other is the Cartesian grid method. In the standard body-fitted grid method, the grid points are aligned with the embedded boundary. The advantage of the body-fitted grid method is its potential to resolve the flow features near body surfaces. The disadvantage is the requirement of a great deal of effort for grid generation and connecting different shape elements with each other.

The Cartesian grid method has recently become one of the widely used methods in CFD [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. This is due to its simplicity, faster grid generation, simpler programming, lower storage requirements, lower operation count, and easier post processing compared to body-fitted structured and unstructured grid methods. The Cartesian grid method is also advantageous in constructing higher order methods. Problems occur at the boundary, when this method is applied to complex domains [15, 16, 17]. In the Cartesian grid method, the intersection of a curved embedded boundary and a computational domain create rectangular cells. These cells are known as cut-cells. These cut-cells create problem for the scheme to be implemented due to time step restriction. One solution to the time step restriction problem caused by small cut-cells is to merge the cut-cells with neighboring cells [8, 18]. Another option is to use ghost points at the immersed boundary. In this method, symmetry conditions with respect to the boundary are imposed at ghost points in the solid adjacent to the boundary [19]. While applying ghost point treatment, mass is lost in the process, hence the scheme is no longer conservative. Nevertheless, we preferred the ghost point treatment over a complicated cut-cells approach because of the simplicity of the former approach.

In the ghost point treatment, the domain is categorized into the following types of points: 1. fluid 2. ghost, and 3. solid. At the embedded boundary, the first and second order stencils require one and two ghost points, respectively. In this paper, we present results for the second order method which require two ghost points at the embedded boundary. In order not to have any impact of solid and ghost points on our computation, we flagged these points to remain inactive.

This paper is an extension of our work presented in [20]. The goal of the present study is two-fold. One aim of this paper is to check the accuracy of the two ghost point treatments. The second aim is to compare the results of the Cartesian grid method and the body-fitted grid method. In this way, the pros and cons of the two ghost point treatments are addressed and the accuracy of the Cartesian grid method is related to the accuracy of the structured grid method for the 2D compressible Euler equations. The standard ghost point treatment and a simplified ghost point treatments are compared. In the standard ghost point treatment, we impose the wall boundary conditions at the ghost points by interpolating the numerical solution at mirror points in the fluid domain and mirroring the interpolated values to insure reflective boundary conditions. In the simplified ghost point treatment, we consider the nearest grid points as mirror points of the ghost points. To increase the accuracy, we apply the MUSCL [21] approach with the minmod limiter. The third order TVD Runge-Kutta (RK3) method is used for time integration. The two ghost point treatments have been investigated for an oblique shock wave and the simplified ghost point treatment is applied to a circular arc airfoil.

The paper is organized as follows. In section II., the governing equations for 2D compressible Euler equations are introduced. An outline of the discretization techniques is given in section III. In section IV., the ghost point treatments at embedded boundaries are explained. Results and discussion of 2D application are given in section V. In the end, conclusions are drawn in section VI.

II. COMPRESSIBLE EULER EQUATIONS

The 2D compressible Euler equations in conservative form read

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}=0, \tag{1}$$

where

$$U=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{bmatrix}, F=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ (\rho E+p)u\end{bmatrix}, G=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ (\rho E+p)v\end{bmatrix}, \tag{2}$$

with ρ, u, v, E and p denoting density, velocities in $x-$ and $y-$directions, total energy per unit mass and pressure, respectively.

For perfect gas, we have the following relation

$$p=(\gamma-1)(\rho E-\frac{1}{2}\rho(u^2+v^2)), \tag{3}$$

where γ is the ratio of specific heats. We consider γ = 1.4 for air.

For supersonic flow in the x-direction, the conservative variables at $x=x_a$ are given as Dirichlet boundary conditions U(x_a , y , t) = g(y, t), cf. Fig. 1. No boundary conditions must be given at $x=x_b$, because the flow is supersonic.

Symmetry boundary conditions at y = 0 imply:

$(\rho,\rho u,\rho E)(x,y,t)=(\rho,\rho u,\rho E)(x,-y,t)$)

and

$$\rho v(x,y,t)=-\rho v(x,-y,t)$$

Extrapolation boundary conditions are assumed at $y=y_d$:

$$\frac{\partial U}{\partial y}=0.$$

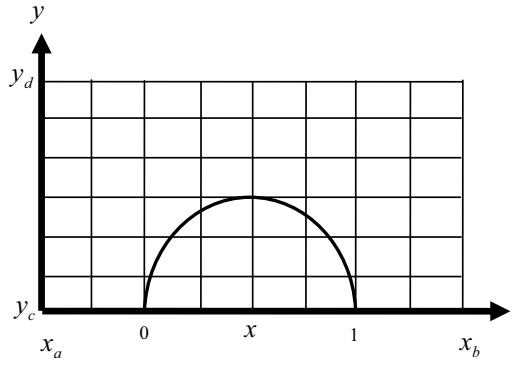


Figure 1. Sketch of domain and Cartesian grid for supersonic flow over a circular arc airfoil.

III. DISCRETIZATION

A. Spatial Discretization

We assume a rectangular domain $[x_a, x_b]\times[y_c, y_d]$ and a (I+1)×(J+1) Cartesian grid with equidistant grid spacing $\Delta x=(x_b - x_a)/I$ and $\Delta y=(y_d - y_c)/J$. The Cartesian coordinates of the grid points (i, j) are (x_i, y_j), where $x_i = i\Delta x$, i = 0, 1, …, I , and $y_j = j\Delta y$, j = 0, 1, …, J.

The node-centered finite volume method yields the following semi-discretization of the 2D compressible Euler equations (1)

$$\frac{dU_{i,j}}{dt}=-\frac{1}{\Delta x}(F^n_{i+\frac{1}{2},j}-F^n_{i-\frac{1}{2},j})-\frac{1}{\Delta y}(G^n_{i,j+\frac{1}{2}}-G^n_{i,j-\frac{1}{2}}). \tag{4}$$

Where $U_{i,j}$ is the approximation of the average of U in the cell $\Omega_{i,j}=[x_i-\frac{\Delta x}{2},x_i+\frac{\Delta x}{2}]\times[y_j-\frac{\Delta y}{2},y_j+\frac{\Delta y}{2}]$, i.e.

$$U_{i,j}\approx\frac{1}{\Delta x.\Delta y}\int_{\Omega_{i,j}}U(x,y,t)\,dxdy\ . \tag{5}$$

If we interpret (4) as a conservative finite difference method, $U_{i,j}$ is an approximation of the exact solution U(x_i, y_j, t). $F_{i+1/2,j}$ and $G_{i,j+1/2}$ are numerical fluxes for the 2D compressible Euler equations. The vector of the conservative variables U and the flux vectors F and G are defined in (2). The numerical fluxes of the local Lax-Friedrichs method for F and G are defined as follows

$$F^{lLF}_{i+\frac{1}{2},j}=\frac{1}{2}(F(U_{i,j})+F(U_{i+1,j})-\max(|u_{i,j}|+c_{i,j},|u_{i+1,j}|+c_{i+1,j})(U_{i+1,j}-U_{i,j})), \tag{6}$$

$$G^{lLF}_{i,j+\frac{1}{2}}=\frac{1}{2}(G(U_{i,j})+G(U_{i,j+1})-\max(|v_{i,j}|+c_{i,j},|v_{i,j+1}|+c_{i,j+1})(U_{i,j+1}-U_{i,j})). \tag{7}$$

In equations (6) and (7), c is the speed of sound. And the CFL number for the 2D compressible Euler equations is defined as

$$CFL = \max_{i,j} \left(\frac{sp(A_1(U_{i,j}))}{\Delta x} + \frac{sp(A_2(U_{i,j}))}{\Delta y}\right)\Delta t,$$

where $sp(A_1(U_{i,j}))$ and $sp(A_2(U_{i,j}))$ are the spectral radii of the Jacobian matrices $A_1 = \partial F/\partial U$ and $A_2 = \partial G/\partial U$, respectively. We choose CFL = 0.5 for the results of the 2D compressible Euler equations, respectively. In (6) we replace $U_{i,j}$ by $U^{L}_{i+1/2.j}$ and $U_{i+1,j}$ by $U^{R}_{i+1/2,j}$ using the MUSCL [21] approach with the minmod limiter to obtain higher order accuracy and also to avoid undesired oscillations. The extrapolated variables are defined as

$$U^{L}_{i+1/2,j} = U_{i,j} + \frac{1}{2}\min\text{mod}(U_{i,j} - U_{i-1,j}, U_{i+1,j} - U_{i,j}), \quad (8)$$

$$U^{R}_{i+1/2,j} = U_{i+1,j} - \frac{1}{2}\min\text{mod}(U_{i+2,j} - U_{i+1,j}, U_{i+1,j} - U_{i,j}). \quad (9)$$

where

$$\min\text{mod}(a,b) = \begin{cases} a & \text{if } |a| \le |b| \text{ and } ab > 0 \\ b & \text{if } |b| < |a| \text{ and } ab > 0 \\ 0 & \text{if } ab \le 0 \end{cases}$$

$$= \text{sign(a)}\max\{0, \min\{|a|, \text{sign(a)b}\}\} \quad (10)$$

is the minmod limiter. The MUSCL approach is applied similarly to the numerical fluxes $G_{i,j+1/2}$ in (7).

B. Approximation of Boundary Conditions

The inflow boundary conditions for supersonic flow at $x = x_a$, cf. Fig. 1, are imposed as

$$U_{1,j}(t) = U_{\infty} \quad , \quad (11)$$

where U_{∞} is the vector of the conservative variables for uniform flow in the x-direction. The flow variables at the outlet $x = x_b$ are approximated as

$$U_{I,j}(t) = U_{I-1,j}(t) \ , \quad (12)$$

i.e. by constant extrapolation. This approximation implies that the upwind finite volume method is used to determine the numerical fluxes $F_{I-\frac{1}{2},j}$.

The symmetry boundary conditions at $y = 0$ are implemented by considering an extra line below $y = 0$. There we use

$$U_{i,1}(t) = diag(1,1,-1,1)U_{i,3}(t) \ . \quad (13)$$

The boundary conditions at $y = y_d$ are treated as

$$U_{i,J}(t) = U_{i,J-1}(t) \ . \quad (14)$$

C. Temporal Discretization

For time integration, we use the third order TVD RK3 method. This method has been chosen because it has a favorable stability domain and preserves the total variation diminishing (TVD) property. Moreover, the TVD RK3 method will be useful for the accuracy of time dependent computations. The TVD RK3 method [22] is given as

$$\mathbf{U}^{(1)} = \mathbf{U}^{(n)} + \Delta t\, \mathbf{R}(\mathbf{U}^{(n)}),$$

$$\mathbf{U}^{(2)} = \frac{3}{4}\mathbf{U}^{(n)} + \frac{1}{4}\mathbf{U}^{(1)} + \frac{1}{4}\Delta t\, \mathbf{R}(\mathbf{U}^{(1)}),$$

$$\mathbf{U}^{(n+1)} = \frac{1}{3}\mathbf{U}^{(n)} + \frac{2}{3}\mathbf{U}^{(2)} + \frac{2}{3}\Delta t\, \mathbf{R}(\mathbf{U}^{(2)}), \quad (15)$$

where

$$R_{i,j} = -\frac{1}{\Delta x}(F^{n}_{i+\frac{1}{2},j} - F^{n}_{i-\frac{1}{2},j}) - \frac{1}{\Delta y}(G^{n}_{i,j+\frac{1}{2}} - G^{n}_{i,j-\frac{1}{2}})$$

is a residual of the 2D compressible Euler equations.

IV. Ghost Point Treatment at Embedded Boundary

A. Standard Ghost Point Treatment for Embedded Boundary in 2D

Fig. 2 illustrates the flagging strategy for 2D flow over a wedge, which we consider as an example of an embedded boundary. We flag the ghost and solid points by assigning them 0 and -1 values, respectively. The fluid points are assigned values equal to 1. In Fig. 3, we show a 2D graphical description of the treatment at the embedded boundary using the standard ghost point treatment which

is a modification of the ghost point treatment mention in [23].

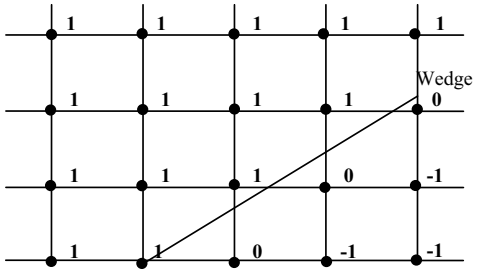


Figure 2. Flagging Strategy for fluid (1), ghost (0) and solid points (-1).

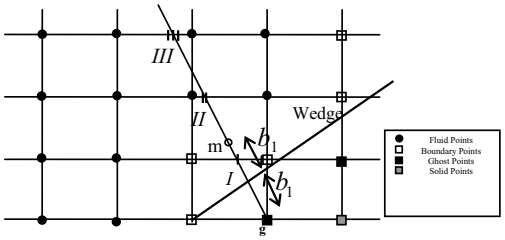


Figure 3. Standard ghost point treatment at the embedded boundary.

We call this ghost point treatment as the standard ghost point treatment. We divide our domain into fluid, boundary, ghost and solid points. A ghost point is denoted by g. The distance of g from the wedge wall is denoted by b_1. The wall unit vector is denoted by **n**. The wall normal line through g is intersecting the horizontal grid lines at three points denoted by vertical lines in Fig. 3. At the first intersection point, we obtain the value of variable vector $\mathbf{V}_I$ by linear interpolation of the values at the neighboring horizontal grid points. And similarly we get $\mathbf{V}_{II}$ and $\mathbf{V}_{III}$. After knowing $\mathbf{V}_I$, $\mathbf{V}_{II}$ and $\mathbf{V}_{III}$ we proceed as follows. Subtract the distance b_1 from the boundary on the normal to obtain the mirror point s_m between $\mathbf{V}_I$ and $\mathbf{V}_{II}$ in the normal direction. Here, we apply linear interpolation between $\mathbf{V}_I$ and $\mathbf{V}_{II}$ and quadratic interpolation among $\mathbf{V}_I$, $\mathbf{V}_{II}$ and $\mathbf{V}_{III}$ for the normal and tangential components of velocity, pressure **p** and density **ρ**. The mathematical description of this strategy to determine the linearly and quadratically interpolated variable vectors $\mathbf{V}_m$ at the mirror point m, cf. (18) and (19), respectively, is expressed as follows.

Let s denote the coordinate in the normal direction.

$$b_1 = s_g - s_{wall} \tag{16}$$

$$s_m = s_{wall} - b_1 \tag{17}$$

$$V_m = V_I + \frac{V_{II} - V_I}{\Delta s}(s_I - s_m), \tag{18}$$

$$V_m = V_{III} + \frac{V_{II} - V_{III}}{s_{II} - s_{III}}(s_m - s_{III}) + \frac{\dfrac{V_I - V_{II}}{s_I - s_{II}} - \dfrac{V_{II} - V_{III}}{s_{II} - s_{III}}}{s_I - s_{III}}(s_m - s_{III})(s_m - s_{II}). \tag{19}$$

where **V** = (ρ, u, v, p) and $\Delta s = s_I - s_{II}$. Then we use reflection boundary conditions

$$u_{t_g} = u_{t_m}, u_{n_g} = -u_{n_m}, p_g = p_m, \rho_g = \rho_m, \tag{20}$$

where u_t and u_n denote the tangential and normal components of the velocity vector, respectively.

B. Simplified Ghost Point Treatment for Embedded Boundary in 2D

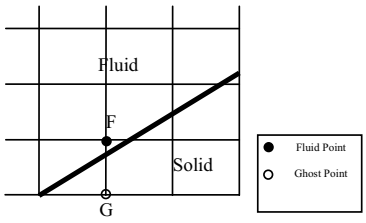


Figure 4. Simplified ghost point treatment [20].

Fig. 4 sketches a simplified ghost point treatment at the embedded boundary of a wedge. A ghost point is denoted by G. In the simplified ghost point treatment, the fluid point F adjacent to the embedded boundary on a grid line parallel to the y-axis is chosen as mirror point. Then, we assume the embedded boundary is in the middle between ghost and fluid points. The density, pressure and tangential velocity component at F are imposed at G,

while the negative value of the normal velocity component **u.n** at F is taken at G. The mathematical description of this strategy is given by

$$\begin{aligned} u_G &= u_F - 2(n_1 u_F + n_2 v_F) n_1, \\ v_G &= v_F - 2(n_1 u_F + n_2 v_F) n_2, \\ p_G &= p_F, \rho_G = \rho_F. \end{aligned} \qquad (21)$$

where u, v, p and ρ denote velocity components in x and y-directions, pressure and density, respectively, cf. [24] for interpolated mirror points. In (21) n_1 and n_2 are the components of the unit normal vector **n** at the boundary. For a curved embedded boundary, the wall normal is taken at the intersection of the boundary and the grid line connecting G and F. The location of the fluid point (x_F, y_F) whose flow conditions are mirrored to the ghost point $(x_G, y_G) = (x_{iG}, y_{iG})$ is chosen as $(x_F, y_F) = (x_{iG}, y_{iG+1})$, cf. Fig. 4. If the fluid point happens to lie on the embedded boundary , we move it one j-index up to obtain proper symmetry conditions with respect to the embedded boundary, i.e. then we choose $(x_G, y_G) = (x_{iG}, y_{iG+2})$.

V. 2D Tests

A. *Test Case: An Oblique Shock Wave*

We verify our 2D code of the Cartesian grid method for an oblique shock wave. For spatial discretization we use the local Lax-Friedrichs (lLF) method, and to increase the order of our method we use the MUSCL approach with the minmod limiter. For time integration, we use TVD RK3 methods. We use the CFL number CFL = 0.5 and N = 101 × 101 grid points to calculate the oblique shock wave. A supersonic flow moves from left to right and hits a wedge resulting in a stationary oblique shock wave. The wedge angle is Θ = 15 degrees. The supersonic upstream flow conditions are given as

$$M = 2, \; p_\infty = 10^5 \text{ Pa}, \; \rho_\infty = 1.2 \text{ kg/m}^3 \qquad (22)$$

B. *Results for Simplified Ghost Point Treatment*

In Fig. 5 and 6, contour lines for the density ρ and velocity component u are presented. The apex of the wedge is placed at $x = 0.4$ m. When the supersonic flow moves from left hits the wedge, then an oblique shock wave is produced which starts from the apex of the wedge and intersects the right boundary. Figs. 5 and 6 show the flow variation in the proximity of the oblique shock wave. A comparison of the exact and numerical solutions for density ρ and velocity component u at $x = 0.75$ m is presented in Figs. 7 and 8. The computed density ρ and velocity component u are becoming closer to the exact solution as the grid is refined. However, there is some discrepancy between the exact and computed solutions near the wall of the wedge. This might be due to the ghost point method not guaranteeing conservativity and due to numerical problems near the apex of the wedge. In Figs. 9 and 10, we have shown the density ρ and velocity u at the bottom wall and wedge.

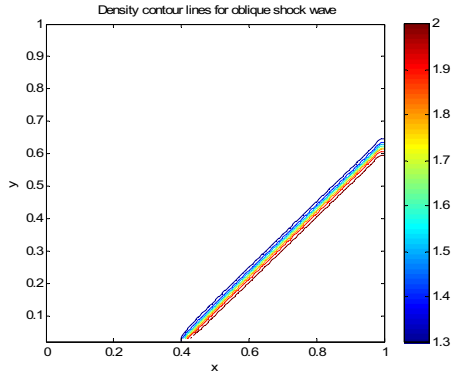


Figure 5. Density contours with MUSCL (101 ×101 grid).

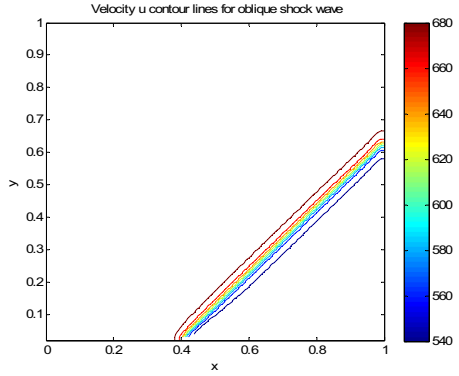


Figure 6. Velocity component u contours with MUSCL (101 ×101 grid).

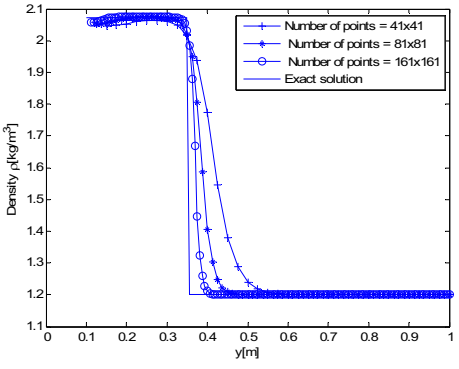


Figure 7. Comparison of exact and numerical solutions of density ρ at different grid levels with MUSCL.

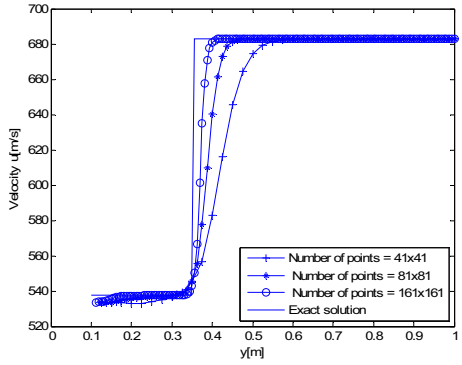


Figure 8. Comparison of exact and numerical solutions of velocity component u at different grid levels with MUSCL.

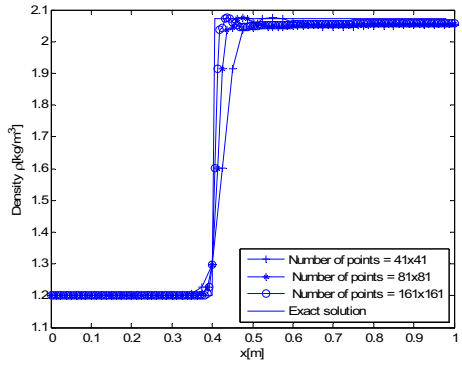


Figure 9. Density ρ at the bottom wall and wedge with MUSCL.

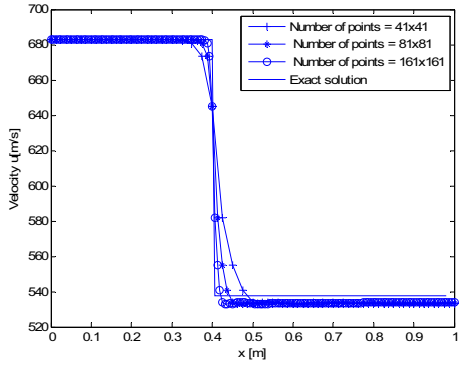


Figure 10. Velocity component u at the bottom wall and wedge with MUSCL.

C. *Results for Standard Ghost Point Treatment*

In this section, results for the standard ghost point treatment are presented and compared with the simplified ghost point treatment.

Figs. 11 and 12 show results for the velocity v and pressure p. We observe almost the same behavior of the standard ghost point treatment as for the simplified ghost point treatment. However, as expected, the standard ghost point treatment is slightly more accurate. Both ghost point treatments yield similar results also for density ρ and velocity component u (not shown).

In Tables I and II, the mass flow errors for the standard and simplified ghost point treatments are presented. The mass flow $\dot{m} = \int_{ywedge}^{1} \rho u(1, y)dy$ of the numerical solutions is approximated by the trapezoidal rule to yield $\dot{m}_{num}$. The mass flow error $\Delta\dot{m}$ is defined by $\Delta\dot{m} = \dot{m} - \dot{m}_{num}$, where $\dot{m} = \rho u_{\infty}.1\text{m}$ is the exact mass flow. The mass flow errors with the standard ghost point treatment using the MUSCL scheme with linear and quadratic interpolation are larger than those for the simplified ghost point treatment shown in Table II. The simplified ghost point treatment is probably more accurate for the mass flow than the standard one, because the former uses the higher order method up to the embedded boundary due to the use of two ghost points while the later drops to first order adjacent to the embedded boundary due to the use of only one ghost point.

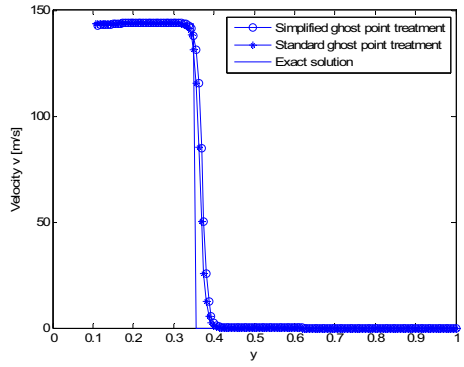


Figure 11. Comparison of simplified and standard ghost point treatments for velocity component v with MUSCL at x = 0.75 m (161 × 161 grid).

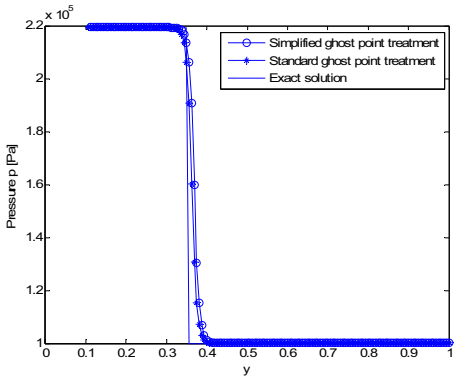


Figure 12. Comparison of simplified and standard ghost point treatments for pressure with MUSCL at $x = 0.75$ m (161 × 161 grid).

TABLE I. MASS FLOW ERROR FOR STANDARD GHOST POINT TREATMENT FOR LINEAR AND QUADRATIC INTERPOLATION WITH MUSCL.

2D Compressible Euler Equations				
Number of points	*Linear Interpolation with MUSCL*		*Quadratic Interpolation with MUSCL*	
$N \times M$	$\Delta \dot{m}$	$\frac{\Delta \dot{m}}{\dot{m}}\%$	$\Delta \dot{m}$	$\frac{\Delta \dot{m}}{\dot{m}}\%$
41 × 41	23.8161	2.9932	22.7270	2.8525
81 × 81	11.7625	1.4563	11.1839	1.3837
161 × 161	5.7399	0.7054	5.4271	0.6667

TABLE II. MASS FLOW ERROR FOR SIMPLIFIED GHOST POINT TREATMENT WITH MUSCL [25].

2D Compressible Euler Equations		
Number of points	*Second Order Method*	
$N \times M$	$\Delta \dot{m}\left[\frac{kg}{s}\right]$	$\frac{\Delta \dot{m}}{\dot{m}}\%$
41 × 41	17.0569	2.1275
81 × 81	8.3530	1.0298
161 × 161	4.0201	0.4930

D. *Test Case: Supersonic Flow Over a Circular Arc Airfoil*

Supersonic flow moves from left to right and hits a circular arc airfoil of which we only consider the upper half. We consider the lengths of the domain in $x-$ and y-directions to be 3m and 1m, respectively. The height of the half circular arc airfoil is 10 % of its chord length which is assumed to be 1m. We use N = 121 × 121 grid points to calculate supersonic flow over that circular arc airfoil at the following uniform flow conditions.

$$M_\infty = 2,\ p_\infty = 10^5 \text{ Pa},\ \rho_\infty = 1.2 \text{ kg/m}^3 \quad (23)$$

1) *Results for the Cartesian Grid Method*

We use the local Lax-Friedrichs method for spatial discretization with MUSCL and minmod limiter and the TVD RK3 method for time integration. We obtain sharp shock waves at the leading and trailing edges, cf. Fig. 13 for Mach number. The second order MUSCL results (first order extrema) indicate an attached oblique shock in agreement with the corresponding body-fitted grid results, cf. below. The convergence history for the higher order method is presented in Fig. 14. The steady state is reached at around n = 2500.

The Mach number distribution on the stagnation streamline is given in Fig. 15. We observe that the flow remains supersonic throughout the domain. At the leading edge, the Mach number decreases due to the shock wave. Then, the Mach number increases from the leading edge to the trailing edge of the arc airfoil due to the flow acceleration. At the trailing edge, the Mach number decreases due to the shock wave and then remains almost constant up to the outlet.

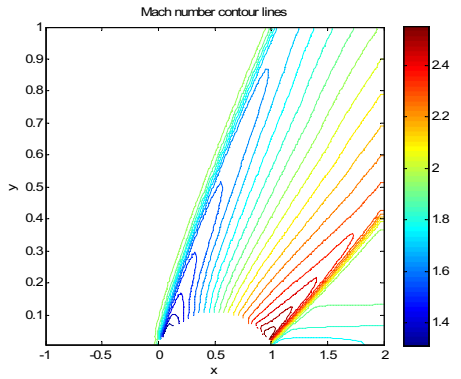


Figure 13. Mach number contours for supersonic flow over a circular arc airfoil.

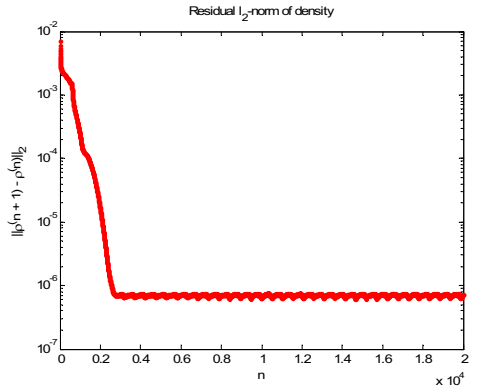


Figure 14. Convergence history for MUSCL with minmod limiter (121 × 121 grid).

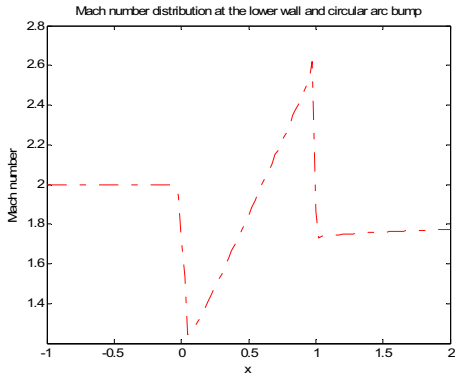


Figure 15. Mach number distribution at the lower wall and circular arc airfoil for supersonic flow over a circular arc airfoil.

2) Results for the Standard Body-Fitted Grid Method

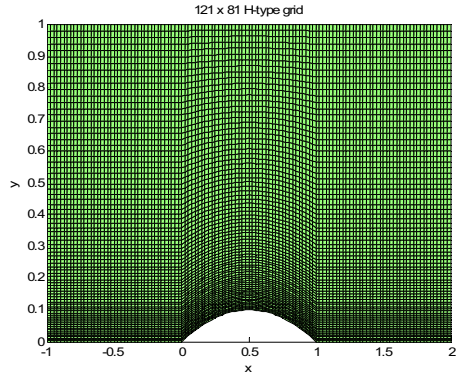


Figure 16. Standard body-fitted grid of a circular arc airfoil.

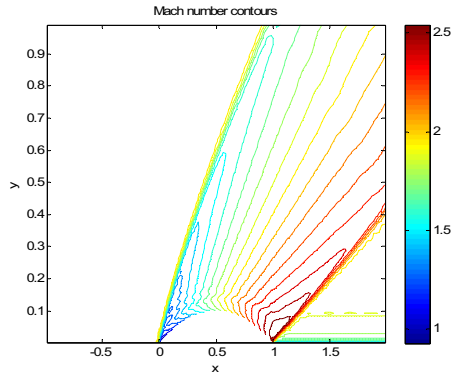


Figure 17. Mach number contours for standard body-fitted grid.

The body-fitted grid for the domain has 121 × 81 grid points, cf. Fig. 16. The grid spacing in the x-direction is the same as for the Cartesian grid method, i.e. $\Delta x = \frac{1}{40}$. The grid points are clustered towards y = 0 and the circular arc airfoil ($\Delta y_{\min}$ = 0.0037 at y = 0 and $\Delta y_{\max}$ = 0.0208 at y = 1).

The cell-centered finite volume method with Roe's approximate Riemann solver is used, which is less diffusive than the local Lax-Friedrichs method. The MUSCL approach is applied to the primitive variables, not to the conservative variables as in (8) and (9). While the

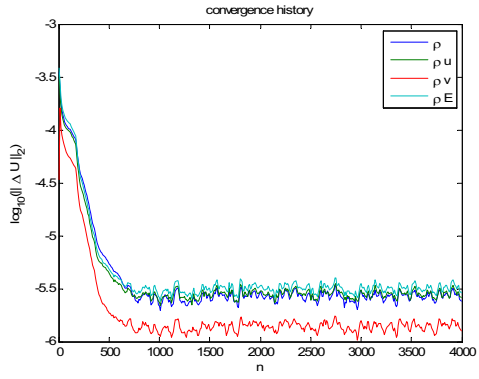


Figure 18. Convergence history for the standard body-fitted grid method.

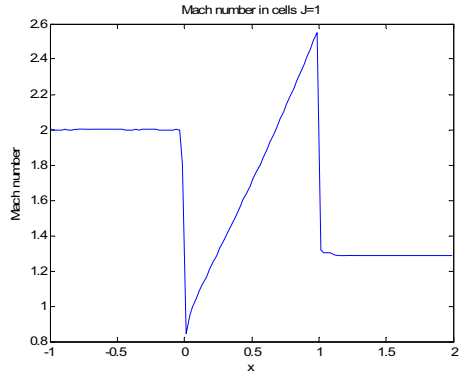


Figure 19. Mach number distribution at y = 0 and circular arc airfoil for supersonic flow over a circular arc airfoil for standard body-fitted grid method.

minmod limiter is used for density and pressure, the less diffusive van Albada limiter is employed for the velocity components. Harten's entropy fix is employed to enforce

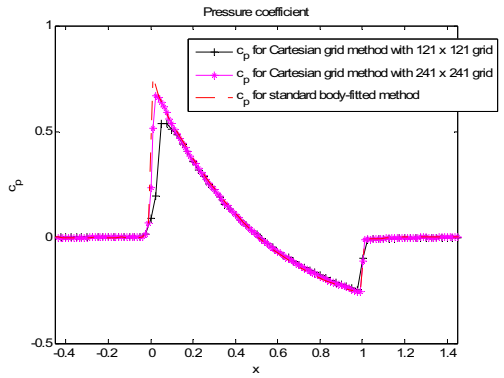


Figure 20. Comparison of pressure coefficient for the Cartesian grid and the standard body-fitted grid methods at different grid levels.

the entropy condition [26]. A low-storage second order Runge-Kutta method was used for time discretization [27].

The contours for Mach number in Fig. 17 for the same supersonic flow over the circular arc airfoil show that the standard body-fitted grid method yields sharper shocks at the leading and trailing edges.

The convergence history for the body-fitted grid method is presented in Fig. 18. The steady state is reached at around n = 600. Note that the l_2 -norms of the changes of the conservative variables stagnate at about 10^{-6}, because single precision was used. The convergence to steady state was accelerated by using local time stepping. The Mach number distribution for the body-fitted grid method in Fig. 19 is similar to the one for the Cartesian grid method in Fig. 15. However, the Mach number at the leading edge and from the trailing edge to outlet is lower than with the Cartesian grid method.

In Fig. 20, we compare the pressure coefficient results obtained with the Cartesian grid method for 121 × 121 and 241 × 241 grids and the standard body-fitted method for the 121 × 81 grid. The Cartesian grid pressure coefficient is lower than the pressure coefficient of the standard body-fitted method for a coarser grid. For the Cartesian grid method, we need a larger number of points to compare with the standard body-fitted method. To achieve a grid spacing Δy corresponding to $\Delta y_{\min}$ of the body-fitted grid, the Cartesian grid method has to employ J + 1 grid points in the y-direction such that $\Delta y = \frac{1}{J} \approx \Delta y_{\min}$.

VI. Conclusions

We applied the Cartesian grid method to solve the 2D compressible Euler equations. Local symmetry boundary conditions were implemented at each ghost point. The two ghost point treatment are applied and compared for an oblique shock wave. The comparison for a circular arc airfoil is drawn between the Cartesian grid method and the standard body-fitted grid method for supersonic flow over a circular arc airfoil. We observed the same accuracy for both linear and quadratic interpolation for the standard ghost point treatment. For the supersonic wedge flow, the presented simplified ghost point treatment proved to be almost as accurate as the standard ghost point treatment, and the mass flow was even predicted slightly better by the former than the latter. For the supersonic arc airfoil flow, e.g. for predicting the pressure coefficient, the Cartesian grid method was found to require a larger number of points.

Acknowledgement

The current research has been funded by the Higher Education Commission (HEC) of Pakistan.

References

[1] A. S. Almgren, J. B. Bell , P. Colella and T. Marthaler, "A Cartesian grid projection method for the incompressible Euler equations in complex geometries," SIAM Journal on Scientific Computing Vol. 18, pp. 1289-1309,1997.

[2] R. Crockett, P. Colella and D. Graves. "A Cartesian grid embedded boundary method for solving the Poisson and heat equations with discontinuous coefficients in three dimensions," Journal of Computational Physics, Vol. 230, pp. 2451-2469, 2011.

[3] P. D. Palma, M. D. de Tullio, G. Pascazio and M. Napolitano, "An immersed boundary method for compressible viscous flows," Computers & Fluids, Vol. 35, pp. 693-702, 2006.

[4] D. D. Marshall and S. M. Ruffin, "A new inviscid wall boundary condition treatment for boundary Cartesian grid method," AIAA 2004-0583 42nd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2004.

[5] H. S. Udaykumar, S. Krishnan and S. V. Marella, "Adaptively refined parallelised sharp interface Cartesian grid method for three dimensional moving boundary problems," International Journal of Computational Fluid Dynamics, Vol. 23, pp. 1-24, 2009.

[6] E. Uzgoren, J. Sim and W. Shyy, "Marker based 3D adaptive Cartesian grid method for multiphase flow around irregular geometries," Communications in Computational Physics, Vol. 5, pp. 1-41, 2009.

[7] Z. Wang, J. Fan and K. Cen, "Immersed boundary method for the simulations of 2D viscous flow based on vorticity-velocity formulations," Journal of Computational Physics, Vol. 228, pp. 1504-1520, 2009.

[8] R. Mittal and G. Iaccarino, "Immersed boundary methods," Annual Review of Fluid Mechanics. Vol. 37, pp. 239-261, 2005.

[9] D. Kirshman and F. Liu, "Cartesian grid solution of the Euler equations using a gridless boundary condition treatment," AIAA Paper 2003-3974 , 2003.

[10] J. Liu, N. Zhao and O. Hu, "The ghost cell method and its applications for inviscid compressible flow on adaptive tree Cartesian grids," Advances in Applied Mathematics and Mechanics, Vol. 1, pp. 664-682, 2009.

[11] A. Dadone and B. Grossman, "Ghost cell method for inviscid two dimensional flows on Cartesian grids," AIAA Journal, Vol. 42, pp. 2499-2507, 2004.

[12] L. A. Catalano, A. Dadone, V. S. E. Daloiso and D. Scardigno, "A multigrid procedure for Cartesian ghost cell method," International Journal for Numerical Methods in Fluids, Vol. 58, pp. 743-750, 2008.

[13] L. S. Kim, K. Nakahashi, H. K. Jeong and M. Y. Ha, "High density mesh flow computations by building cube method," Journal of Mechanical Science and Technology, Vol. 21, 1306-1319, 2007.

[14] Y. Hassen and B. Koren, "Finite-volume discretizations and immersed boundaries", in: Lec. Notes Comput. Sci.Eng. 71, Springer, Heidelberg, pp. 229-268, 2010.

[15] J. J. Quirk, "An alternative to unstructured grids for computing gas dynamics flows around arbitrarily complex two-dimensional bodies," Computers & Fluids, Vol. 23, pp. 125-142, 1994.

[16] M. Berger and R. LeVeque, "An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries" AIAA Paper 89-1930-CP, 1989.

[17] D. Clarke, M. Salas and H. Hassan, "Euler calculations for multielement airfoil using Cartesian grids", AIAA Journal, Vol. 24, pp. 353-358, 1986.

[18] D. Hartmann, M. Meinke and W. Schröder, "A strictly conservative Cartesian cut-cell method for compressible viscous flows on adaptive grids," Computer Methods in Applied Mechancis and Engineering, Vol. 200, pp. 1038-1052, 2011.

[19] H. Forrer and R. Jeltsch, "A higher order boundary treatment for Cartesian grid methods," Journal of Computational Physics, Vol. 140, pp, 259-277, 1998.

[20] M. A. Farooq and B. Müller, "Investigation of the accuracy of the Cartesian grid method," Processings of International Bhurban Conference on Applied Sciences and Technology, Islamabad, Pakistan, January 10-13, 2011, pp. 45-53, 2011.

[21] B. van Leer, "Towards the ultimate conservative difference scheme," V. A second order sequel to Godunov's method, Journal of Computational Physics, Vol. 32, pp. 101-136, 1979.

[22] S. Gottlieb and C. Shu, "Total variation diminishing Runge-Kutta schemes," Mathematics of Computations, Vol. 67, pp. 73-85, 1998.

[23] B. Sjögreen and N. A. Petersson, "A Cartesian embedded boundary method for hyperbolic conservation laws," Communications in Computational Physics, Vol. 2, pp. 1199-1219, 2007.

[24] R. Löhner, "Applied computational fluid dynamics technique: An introduction based on finite element methods", (2nd edition, John Wiley & Sons, Chichester, 2008).

[25] M. A. Farooq and M. Müller, "Cartesian grid method for the compressible Euler equations", in: J. Fort, J. Fürst, J. Halama, R. Herbin, F. Hubert (Eds), Finite Volumes for Complex Applications VI Problems & Perspectives: FVCA , International Symposium, Prague, June 6-10, 2011, Springer-Velag, Berlin, pp. 449-456, 2011.

[26] A. Harten, "High resolution schemes for hyperbolic conservation laws," Journal of Computational Physics, Vol. 49, pp. 357-393, 1983.

[27] B. Müller, "Methods for Euler and Navier-Stokes equations," Lecture Notes, NGSSC, CFD Course, Uppsala, Sweden, 2003.