



**NTNU – Trondheim**  
Norwegian University of  
Science and Technology

# Cartesian grid methods for the compressible Navier-Stokes equations

**Are Arstad Skøien**

Master of Energy and Environmental Engineering

Submission date: June 2012

Supervisor: Bernhard Müller, EPT

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





EPT-M-2012-80

**MASTER THESIS**

for

student Are Skøien

Spring 2012

Cartesian grid methods for the compressible Navier-Stokes equations

*Kartesiske grid metoder for de kompressible Navier-Stokes ligningene***Background and objective**

The Cartesian grid method has recently become a popular method in computational fluid dynamics (CFD). The reason lies in its simplification of grid generation. Instead of generating a body-fitted structured or unstructured grid, the body is embedded in a simple Cartesian grid and the effect of the body surface is taken into account by proper conditions at grid points near the body surface. We want to investigate Cartesian grid methods for two-dimensional (2D) viscous compressible flow. The accuracy of the methods will be checked for suitable test cases. The 2D Navier-Stokes equations for compressible flow will be solved by an explicit finite volume method on a Cartesian grid. A Cartesian grid MATLAB code based on a simplified ghost point treatment for the 2D compressible Navier-Stokes equations will be used as an example. The goal of the project is to develop and to implement a Cartesian grid method that resolves 2D viscous compressible flow more accurately than the Cartesian grid method based on a simplified ghost point treatment and to compare the Cartesian grid methods with each other and the literature.

**The following tasks are to be considered:**

- 1 Review the literature on Cartesian grid methods for the compressible Navier-Stokes equations.
- 2 Develop and implement a Cartesian grid method different from the simplified ghost point treatment for 2D viscous compressible flow.
- 3 Compare the new Cartesian grid method with the simplified ghost point treatment and with the literature for suitable test cases of 2D viscous compressible flow.

-- " --

Within 14 days of receiving the written text on the master thesis, the candidate shall submit a research plan for his project to the department.

When the thesis is evaluated, emphasis is put on processing of the results, and that they are presented in tabular and/or graphic form in a clear manner, and that they are analyzed carefully.

The thesis should be formulated as a research report with summary both in English and Norwegian, conclusion, literature references, table of contents etc. During the preparation of the text, the candidate should make an effort to produce a well-structured and easily readable report. In order to ease the evaluation of the thesis, it is important that the cross-references are correct. In the making of the report, strong emphasis should be placed on both a thorough discussion of the results and an orderly presentation.

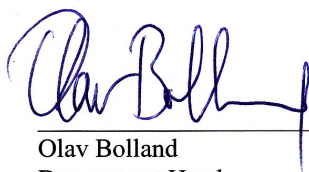
The candidate is requested to initiate and keep close contact with his/her academic supervisor(s) throughout the working period. The candidate must follow the rules and regulations of NTNU as well as passive directions given by the Department of Energy and Process Engineering.

Risk assessment of the candidate's work shall be carried out according to the department's procedures. The risk assessment must be documented and included as part of the final report. Events related to the candidate's work adversely affecting the health, safety or security, must be documented and included as part of the final report.

Pursuant to "Regulations concerning the supplementary provisions to the technology study program/Master of Science" at NTNU §20, the Department reserves the permission to utilize all the results and data for teaching and research purposes as well as in future publications.

The final report is to be submitted digitally in DAIM. An executive summary of the thesis including title, student's name, supervisor's name, year, department name, and NTNU's logo and name, shall be submitted to the department as a separate pdf file. Based on an agreement with the supervisor, the final report and other material and documents may be given to the supervisor in digital format.

Department of Energy and Process Engineering, 16. January 2012



Olav Bolland  
Department Head



Bernhard Müller  
Academic Supervisor

## Sammendrag

En kartesisk grid metode har blitt utviklet for å løse de todimensjonale Euler og Navier-Stokes ligningene for viskøs og ikke-viskøs kompressibel strømning. Både stasjonære og ikke-stasjonære strømninger har blitt løst. Ved å bruke en forenklet ghost punkt behandling, så bruker vi de nærmeste grid punkter som speilbildepunkter til ghost punktene. På randen til det solide legemet setter vi veggbetingelser ved hjelp av ghost punktene. Metoden bygger på en annen forenklet ghost punkt metode, men den nye metoden presterer bedre. Nøyaktigheten til metoden har blitt undersøkt ved å gjenskape godt dokumenterte eksempler funnet i litteraturen. Vi viser utregnede eksempler av supersonisk strømning forbi en diamant-skråvinge og sammenligner med analytiske resultater. Vi regner også ut transiente løsninger for de kompressible Euler ligningene og gjør en sammenligning av tetthet og trykk med annet arbeid ved forskjellige tidspunkter. Den supersoniske viskøse strømningen rundt en NACA0012 vingeprofil simuleres og vi regner ut løft- og drag-koeffisienter samt profilen til trykk-koeffisienten og sammenligner dette med litteraturen. Metoden blir også testet for supersonisk strømning over en sylinder og skjærkraft-koeffisienten blir brukt til å sjekke nøyaktigheten. Resultatene kan sammenlignes med andre mer avanserte kartesisk grid metoder, men er ikke like nøyaktige. Til slutt blir den supersoniske strømningen rundt et todimensjonalt F22 jagerfly inklusiv motorstrømning simulert for å illustrere fleksibiliteten til metoden.

## Abstract

A Cartesian grid method has been developed for solving the 2D Euler and Navier-Stokes equations for viscous and inviscid compressible flow, respectively. Both steady and unsteady flows have been considered. Using a simplified ghost point treatment, we consider the closest grid points as mirror points of the ghost points. Wall boundary conditions are imposed at the ghost points of the immersed boundary. The accuracy of the method has been investigated for various test cases. We show computed examples of supersonic flow past a diamond-wedge airfoil and compare with analytical results. Further we compute time accurate solutions of the compressible Euler equations for an incident shock over a cylinder and compare the pressure time history with other work. The supersonic viscous flow around a NACA0012 airfoil is computed, and the lift and drag coefficients along with the pressure coefficient profile are compared with the literature. The method is also tested for supersonic flow over a cylinder, and the computed skin friction profiles have been used to assess the accuracy. Lastly the supersonic flow around a 2D F-22 fighter aircraft with simulated jet engine outflow is shown to illustrate the flexibility of the method. The present method is built on a previously established simplified ghost point treatment, but performs better. The results are comparable, although not as accurate as other more complex methods.

## Table of Contents

Legend.....	6
List of Figures .....	7
1 Introduction.....	9
2 Governing Equations .....	11
2.1 Compressible Navier-Stokes Equations.....	11
2.2 Pressure Coefficient and Skin Friction Coefficient .....	13
2.3 Geometry for 4-digit NACA Airfoil .....	13
2.4 Lift and Drag on Airfoil .....	14
2.5 Rankine-Hugoniot Conditions for Moving Shock.....	15
3 Numerical Discretization .....	16
3.1 Spatial Discretization .....	16
3.2 Discretization of the Boundary Conditions.....	18
3.3 Temporal Discretization.....	19
3.4 Stability of the Numerical Scheme .....	20
4 Ghost Point Treatment at Embedded Boundary .....	20
4.1 Simplified Ghost Point Treatment for Embedded Boundary in 2D.....	20
4.2 Ghost Point Treatment with Second Order Accuracy.....	22
4.3 The SP and KP Embedded Boundary Method.....	22
4.4 Simplified Ghost Point Treatment for the Compressible Euler Equations .....	22
4.5 New Simplified Ghost Point Treatment for Embedded Boundary in 2D .....	23
4.6 Calculating the Shear Stress on a Body. ....	25
5 Verification of the New Simplified Ghost Point Treatment .....	26
5.1 Supersonic Flow around a Diamond-Wedge Airfoil with Compressible Euler Equations .....	26
5.2 Supersonic Flow over a Circular Cylinder with Compressible Euler Equations.....	29
5.3 Moving Shock Wave over a Circular Cylinder with Compressible Euler Equations.....	30
5.4 Supersonic Flow over a Circular Cylinder with Compressible Navier-Stokes Equations.....	34
5.5 Supersonic Flow past an NACA0012 Airfoil with Compressible Navier-Stokes Equations ....	38
5.6 Supersonic Flow around a 2D F-22 Fighter Aircraft .....	43
5.7 Results and Discussion .....	45
6 Conclusions and Outlook.....	46
Acknowledgments.....	46
References.....	47
Appendix.....	49

## Legend

$\gamma$	Ratio of specific heats
$\mu$ [ $kg/m \cdot s$ ]	Dynamic viscosity
$\rho$ [ $kg/m^3$ ]	Density
$\tau$ [ $N/m^2$ ]	Viscous stress tensor
$c$ [ $m/s$ ]	Speed of sound
$c_p$ [ $J/kg \cdot K$ ]	Specific heat at constant pressure
$c_v$ [ $J/kg \cdot K$ ]	Specific heat at constant volume
$E$ [ $J/kg$ ]	Energy per unit mass
$k$ [ $W/m \cdot K$ ]	Thermal conductivity
$p$ [ $N/m^2$ ]	Pressure
$T$ [ $K$ ]	Temperature
$u$ [ $m/s$ ]	Velocity in x-direction
$v$ [ $m/s$ ]	Velocity in y-direction
$\Delta t$ [ $s$ ]	Time step
$\Delta x$ [ $m$ ]	Grid cell size in x-direction
$\Delta y$ [ $m$ ]	Grid cell size in y-direction
$C_p$	Pressure coefficient
$C_f$	Skin friction coefficient
$C_D$	Drag coefficient
$C_L$	Lift coefficient
CFL	Courant-Friedrichs-Lewy number
$M$	Mach number
$Pr$	Prandtl number
$Re$	Reynolds number
VNN	Von Neumann number

## List of Figures

<b>Figure 1:</b> Sketch of domain and boundary conditions. ....	12
<b>Figure 2a:</b> Coordinate system in which shock is at rest. Steady Euler equations yield Rankine-Hugoniot conditions for steady shock.....	15
<b>Figure 2b:</b> Coordinate system in which flow in region 1 is at rest. Change of coordinate system from figure 2a to figure 2b yields Rankine-Hugoniot conditions for shock moving with velocity $-u_1$ ( $u_1$ from figure 2a). ....	15
<b>Figure 3:</b> Sketch of the numerical fluxes in the Cartesian domain. ....	17
<b>Figure 4:</b> Sketch of domain and discretized boundary conditions. ....	18
<b>Figure 5:</b> Flagged ghost cells on a (81x81) grid. Back points indicate the location of ghost points. ....	20
<b>Figure 6:</b> Illustration of the fluid points chosen as mirror points and the ghost points set to create a solid cylinder. ....	21
<b>Figure 7:</b> Illustration of ghost point calculations for MUSCL. ....	22
<b>Figure 8:</b> Angle used to identify fluid point (left). Normal components on body (right). ....	23
<b>Figure 9:</b> Fluid points (small circles) in white fluid domain used as mirror points for ghost points (small circles) in green solid domain in the New Simplified method. ....	24
<b>Figure 10:</b> Identifying the embedded boundary. ....	25
<b>Figure 11:</b> The points used in the averaging of the shear stress at the wall. ....	26
<b>Figure 12:</b> Illustration of expected shock wave behavior in a channel with a diamond-wedge. ....	27
<b>Figure 13:</b> Residual convergence history for inviscid supersonic flow past a diamond-wedge airfoil at a Mach number of 3 and a deflection angle of $15^\circ$ . ....	27
<b>Figure 14:</b> Mach number contours past a diamond-wedge airfoil with attached shock, and shock reflection at upper wall boundary, $M_\infty = 3$ , $\theta = 15^\circ$ . ....	28
<b>Figure 15:</b> Effects of increasing the deflection angle such that we observe a detached shock instead of an attached shock in flow past a wedge. ....	29
<b>Figure 16:</b> Mach number contours past a diamond-wedge airfoil with a detached shock, $M_\infty = 1.7$ , $\theta = 24^\circ$ . ....	29
<b>Figure 17:</b> Comparison of the pressure coefficient obtained from the New Simplified method, the Simplified method and the results from Sjögreen and Petersson (2007) [8]. ....	30
<b>Figure 18:</b> Comparison of computed density contours for a incident shock $M_s = 2$ past a circular cylinder at different times obtained using the hybrid method [13] and the New Simplified method. ....	31
<b>Figure 19:</b> Comparison of pressure at different times obtained using the hybrid method, the unstructured grid method [13] and the New Simplified method. ....	33
<b>Figure 20:</b> Mach number contours of supersonic flow past a cylinder obtained using the New Simplified method. ....	34
<b>Figure 21:</b> Comparison of the skin friction coefficient obtained from the weighted ghost point New Simplified Method and the New Simplified method without weighted ghost points. ....	35
<b>Figure 22:</b> Skin friction profile obtained from the New Simplified method with grid refinement compared with a reference skin friction profile [3]. ....	35
<b>Figure 23:</b> Surface plot of supersonic flow past a circular cylinder with second order accuracy on a 401x401 grid. ....	36
<b>Figure 24:</b> Convergence history plot obtained from the New Simplified method on supersonic Mach 3 flow with Reynolds number 500 around a circular cylinder. ....	37
<b>Figure 25:</b> Comparison of the skin friction coefficient obtained from the New Simplified method and the Simplified method. ....	37
<b>Figure 26:</b> Residual convergence history for supersonic flow past a NACA0012 airfoil at a Mach Number of 2, angle of attack $10^\circ$ and Reynolds number 1000 on a 801x401 grid. ....	38
<b>Figure 27:</b> Example choice of fluid points near the leading edge of a NACA0012 airfoil. ....	39

<b>Figure 28:</b> Mach number contour results from P. De Palma et al (2006) [14]. $M_\infty = 2$ , $Re_\infty = 1000$ , $\alpha = 10^\circ$ .....	39
<b>Figure 29:</b> Mach number contour results from P. De Palma et al. (2006)[14] in the domain .....	40
<b>Figure 30:</b> Mach number contours for a NACA0012 airfoil obtained from the New Simplified method.....	40
<b>Figure 31:</b> Pressure coefficient obtained from the New Simplified method (left) and from P. De Palma et al. (2006), [14] (right) .....	42
<b>Figure 32:</b> Flagged ghost cells for the F-22 geometry on a 221x161 grid.....	43
<b>Figure 33:</b> Mach number image with cool color mapping and lightning effects for the 2D F-22 fighter aircraft. ....	44
<b>Figure 33:</b> Mach number image with hot color mapping of the 2D F-22 fighter aircraft in vertical accent. ....	45



# 1 Introduction

Computational Fluid Dynamics (CFD), has been an emerging alternative to the experiment. CFD is a very effective tool used to solve complex problems at low costs compared to laboratory experiments. CFD now works hand in hand with laboratory experiments to test physical problems. The cost of repeating an experiment several times can be lowered with the use of CFD and computer technology to run the same problem and suggest corrections to the experiment. The experimental data is still more valuable due to its accuracy, but CFD is essentially used to lower the number of laboratory tests needed. The researcher is also able to introduce new ideas that would have been very hard to replicate in a laboratory without good funding. There are a number of scientific fields where CFD is becoming increasingly important, such as the aircraft industry, meteorology, vehicle design, biology and astrophysics [2].

The Cartesian grid method has recently become a popular method in CFD to compute flows over or in complex geometries [12,14,13,5,8,3,9,2]. The reason lies in its simplification of grid generation, 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. Adding an additional body with the body-fitted method means redoing the entire grid, but with the Cartesian grid method investigated here this can be done in a few simple steps. Instead of generating a body-fitted structured grid, the body is embedded in a simple Cartesian grid and the effect of the body is taken into account by proper conditions at grid points near the body surface.

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 [12]. This problem is not present in the simplified ghost point treatment, as symmetry conditions with respect to the boundary are imposed at ghost points in the solid adjacent to the boundary. However, conservativity is lost in the process.

Other methods for embedding a solid boundary are found in [14,13,8,3] and the results in this work are compared to these methods. These methods are more complex and are able to produce the effect of the solid boundary more accurately. Sjögreen and Petersson [8] used linear interpolation at the embedded boundary. H. Luo et al. [13] used a Cartesian grid as a baseline mesh to cover the computational domain, while the boundary surfaces were addressed using a gridless method [13].

The simplified ghost point method has been verified for the 2D compressible Euler equations using a circular arc bump in a channel as the embedded boundary [9]. In previous work, the simplified ghost point method was extended to the compressible Navier-Stokes equations using a circular cylinder [22]. The method experienced some problems on the part of the body where the mirrored ghost points change direction due to the shape of the body. Unphysical behavior at these points could turn the solution unstable for large time steps or for fine grids, and also affect the accuracy.

A new simplified ghost point method has been developed and is presented in the present work. Extensive numerical tests have been performed to demonstrate the properties of the method. In addition to steady state viscous flow over the circular cylinder, an unsteady time accurate analysis has been done with the compressible Euler equations. Further, supersonic viscous flow over a NACA0012 airfoil has been calculated along with inviscid flow over a diamond-wedge airfoil. The last example flow is that around the profile of a 2D F-22 fighter aircraft. The method experiences some problems for the skin friction profile on the circular cylinder, seen as unwanted bumps in the profile, but the computed results are overall in good agreement with those obtained from other work

and available analytical data. In the development of the code focus has been laid on how to easily add additional solid embedded bodies, although not shown in the paper it can be seen on the cover picture that features two NACA0018 and two NACA0012 airfoils at different angles of attack.

The paper is organized as follows. In section 2, the governing equations are introduced. The compressible Navier-Stokes equations are explained along with the skin friction and the pressure coefficients. Section 2 also covers the 4-digit NACA airfoil geometry and how lift and drag are calculated. The Rankine-Hugoniot conditions for a moving shock are also presented. In section 3 the discretization techniques are presented, and in section 4 the Simplified and the New Simplified ghost point treatments are explained. Results of steady and unsteady inviscid and viscous 2D compressible flow test cases over a circular cylinder and an airfoil are presented and discussed in section 5. The final conclusion is given in the end.

## 2 Governing Equations

### 2.1 Compressible Navier-Stokes Equations

The compressible Navier-Stokes equations for perfect gas in two dimensions are considered. In conservative form they read

$$\frac{\partial U}{\partial t} + \frac{\partial F_c}{\partial x} + \frac{\partial G_c}{\partial y} = \frac{\partial F_v}{\partial x} + \frac{\partial G_v}{\partial y}, \quad (1)$$

where

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

$$F_v = \begin{bmatrix} 0 \\ 2\mu \frac{\partial u}{\partial x} - \frac{2}{3}\mu \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \\ \mu \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \\ \tau_{xx}u + \tau_{yx}v + k \frac{\partial T}{\partial x} \end{bmatrix}, G_v = \begin{bmatrix} 0 \\ \mu \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right) \\ 2\mu \frac{\partial v}{\partial y} - \frac{2}{3}\mu \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \\ \tau_{xy}u + \tau_{yy}v + k \frac{\partial T}{\partial y} \end{bmatrix}, \quad (3)$$

where  $\rho$  is the density,  $u$  and  $v$  the velocity components in  $x$ - and  $y$ -directions, respectively,  $p$  denotes the pressure,  $\mu$  the viscosity and  $E$  the total energy per unit mass. The components of the viscous stress tensor  $\tau$  are

$$\tau_{xx} = 2\mu \frac{\partial u}{\partial x} - \frac{2}{3}\mu \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right), \quad (4)$$

$$\tau_{yy} = 2\mu \frac{\partial v}{\partial y} - \frac{2}{3}\mu \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right), \quad (5)$$

$$\tau_{xy} = \tau_{yx} = \mu \left( \frac{\partial v}{\partial x} + \frac{\partial u}{\partial y} \right). \quad (6)$$

The pressure  $p$  and the temperature  $T$  are given by the equations of state for perfect gas, which can be expressed as

$$T = \frac{p}{R\rho}, \quad (7)$$

$$p = (\gamma - 1) \left( \rho E - \frac{1}{2}\rho(u^2 + v^2) \right). \quad (8)$$

where  $\gamma = \frac{c_p}{c_v}$  is the ratio of specific heats at constant volume and pressure, respectively, and  $R$  is the gas constant for air. The viscosity is dependent on the temperature and is calculated using Sutherland's law

$$\mu(T) = \mu_{\infty} \left( \frac{T}{T_{\infty}} \right)^{1.5} \frac{T_{\infty} + 110K}{T + 110K}, \quad (9)$$

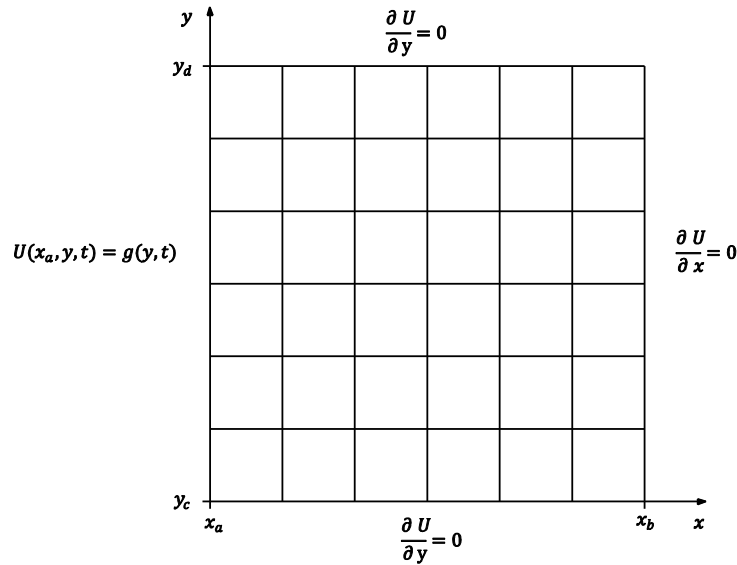
where  $\mu_{\infty}$  is a reference viscosity and  $T_{\infty}$  is a reference temperature.

The speed of sound  $c$  is related to the pressure and density by

$$c^2 = \gamma \frac{p}{\rho}, \quad (10)$$

where  $\gamma = \frac{c_p}{c_v}$  is the ratio of specific heats,  $\gamma = 1.4$  for air, which is used in all simulations. The Prandtl number  $Pr$  is a constant that relates thermal conductivity to viscosity. For air at standard conditions  $Pr = 0.72$  and  $k = \frac{\mu c_p}{Pr}$  is the thermal conductivity.

For boundary conditions we distinguish between three types of boundaries, supersonic inflow, supersonic outflow and solid wall. For supersonic inflow in the x-direction, the conservative variables at the inlet are given as Dirichlet boundary conditions on all flow variables. No boundary conditions must be imposed at the outlet because the flow is supersonic. Extrapolation boundary conditions are assumed, see figure 1.



**Figure 1:** Sketch of domain and boundary conditions.

Later in the report the boundaries are referred to as north, south, east and west, where the west boundary is the inflow boundary in figure 1.

## 2.2 Pressure Coefficient and Skin Friction Coefficient

### Pressure coefficient

The pressure coefficient is calculated along the embedded boundary surface where a wall condition is imposed. It is calculated using the pressure on the ghost point indices. This works because the pressure at the ghost points is taken from the fluid points. The wall lies between a ghost point and a fluid point, so the pressure at the wall must be equal to one of these points.

The pressure coefficient is given as

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty U_\infty^2}, \quad (11)$$

and can also be expressed as

$$C_p = \frac{2}{\gamma M_\infty^2} \left( \frac{p}{p_\infty} - 1 \right), \quad (12)$$

where  $p_\infty$ ,  $\rho_\infty$ ,  $U_\infty$  and  $M_\infty$  is the reference pressure, density, velocity and Mach number respectively.

The Mach number is defined as  $M = \frac{\sqrt{u^2 + v^2}}{\sqrt{\gamma RT}} = \frac{\sqrt{u^2 + v^2}}{c}$ .

### Skin friction coefficient

For viscous flow the no-slip condition at the wall will cause viscous drag. The skin friction coefficient  $C_f$  is calculated by using the shear stress on the wall of the embedded body. The skin friction coefficient at a wall is defined by

$$C_f \equiv \frac{\tau_w}{\frac{1}{2}\rho_\infty U_\infty^2}, \quad (13)$$

where  $\tau_w$  is the local wall shear stress, and is calculated as

$$\tau_w = (\tau_{xx}n_x + \tau_{xy}n_y)n_y - (\tau_{xy}n_x + \tau_{yy}n_y)n_x, \quad (14)$$

with  $n_x$  and  $n_y$  the x- and y-components of the unit normal vector. The components of the viscous stress tensor  $\tau_{xx}$ ,  $\tau_{yy}$  and  $\tau_{xy}$  are given in (4), (5) and (6).

## 2.3 Geometry for 4-digit NACA Airfoil

For the flow around an airfoil we have considered the NACA0012. It is a symmetrical airfoil, and the equation for the upper surface is

$$y = \frac{t}{0.2} c \left[ 0.2969 \sqrt{\frac{x}{c}} - 0.1260 \left( \frac{x}{c} \right) - 0.3515 \left( \frac{x}{c} \right)^2 + 0.2843 \left( \frac{x}{c} \right)^3 - 0.1015 \left( \frac{x}{c} \right)^4 \right], \quad (15)$$

where  $t$  is the maximum thickness with respect to the chord (  $100t$  gives the last two digits in NACA0012),  $c$  is the chord length and  $x$  is the position along the chord.

With equation (15), at the trailing edge where  $\left(\frac{x}{c}\right) = 1$ , the thickness is not zero. In the GAMM-Workshop [16] the airfoil was extended to a closed trailing edge (with zero thickness) by extending the chord to  $c_{extended} = 1.0089c$  [16]. We have instead modified the last coefficient to -0.1036, which will result in a very small change to the overall shape, but will close the gap at the trailing edge.

## 2.4 Lift and Drag on Airfoil

The force acting on the airfoil is given as

$$\vec{F} = - \int_c P \vec{n} dA + \int_c \tau \cdot \vec{n} dA, \quad (16)$$

where  $c$  is the airfoil contour.

The normal and tangential force coefficients are given as

$$C_n = \frac{F_y}{\frac{1}{2}\rho U^2 c}, \quad (17)$$

$$C_t = \frac{F_x}{\frac{1}{2}\rho U^2 c}. \quad (18)$$

Where  $\rho$  and  $U$  are reference density and velocity, respectively.

The lift and drag coefficients are calculated with respect to the flow direction and become

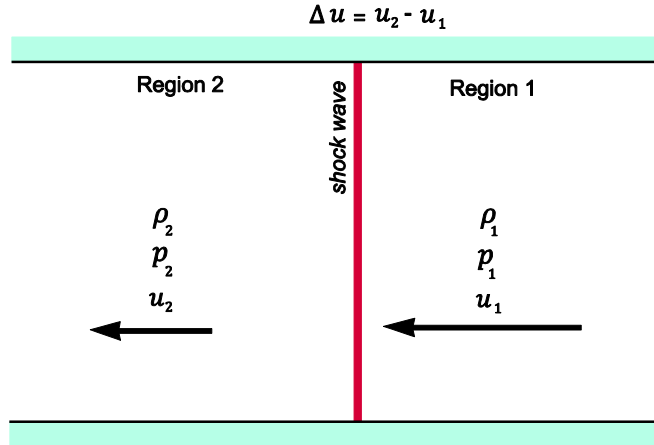
$$C_L = C_n \cos(\alpha) - C_t \sin(\alpha), \quad (19)$$

$$C_D = C_n \sin(\alpha) + C_t \cos(\alpha), \quad (20)$$

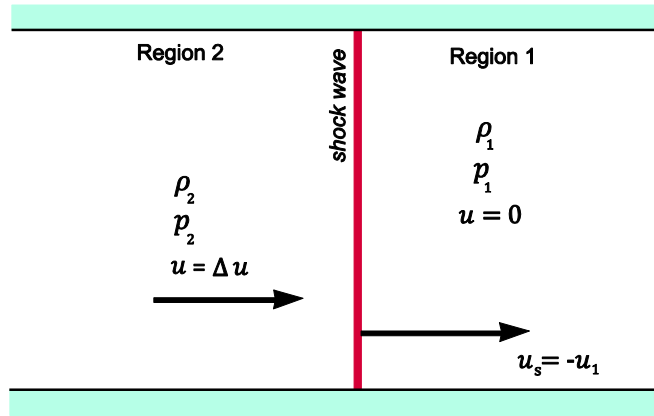
where  $\alpha$  is the angle of attack on the airfoil.

## 2.5 Rankine-Hugoniot Conditions for Moving Shock

**Figure 2a:** Coordinate system in which shock is at rest. Steady Euler equations yield Rankine-Hugoniot conditions for steady shock.



**Figure 2b:** Coordinate system in which flow in region 1 is at rest. Change of coordinate system from figure 2a to figure 2b yields Rankine-Hugoniot conditions for shock moving with velocity  $-u_1$  ( $u_1$  from figure 2a).



The Rankine-Hugoniot conditions for a moving shock are

$$\frac{\Delta u}{c_1} = \frac{2}{\gamma+1} \frac{-u_1}{c_1} \left( 1 - \left( \frac{c_1}{u_1} \right)^2 \right), \quad (21)$$

$$\frac{\rho_1}{\rho_2} = 1 - \frac{2}{\gamma+1} \left( 1 - \left( \frac{c_1}{u_1} \right)^2 \right), \quad (22)$$

$$\frac{p_2}{p_1} = 1 + \frac{2\gamma}{\gamma+1} \left( \left( \frac{u_1}{c_1} \right)^2 - 1 \right), \quad (23)$$

$$M_s = \frac{-u_1}{c_1}, \quad (24)$$

where  $M_s$  is the moving shock Mach number,  $u_1$  and  $c_1$  are the velocity and the speed of sound, respectively, to the right of the shock in figure 2a. In figure 2b the flow ahead of the shock i.e. to the right of the shock, is stagnant. The velocity to the left of the shock must then be set as  $\Delta u$ .

In other words, if a shock wave moves from left to right with a mach Number  $M_s$ , the primitive variables for each side of the shock must be set following the conditions in (21), (22), (23), and (24).

### 3 Numerical Discretization

The convective terms of the Navier-Stokes equations are discretized by a second order accurate total variation diminishing (TVD) finite difference approximation, which reduces to first order at extrema. The viscous terms are discretized by central differences. Time advancement is then done with an explicit Runge-Kutta scheme.

#### 3.1 Spatial Discretization

The rectangular domain is the box  $[-L_x, L_x] \times [-L_y, L_y]$  and a  $[N \times M]$  Cartesian grid with equidistant grid spacing  $\Delta x = \frac{2L_x}{N-1}$  and  $\Delta y = \frac{2L_y}{M-1}$ . The Cartesian coordinates of the grid points  $(i, j)$  are  $(x_i, y_j)$ , where  $x_i = -L_x + (i-1)\Delta x$ ,  $i = 1, 2, \dots, N$  and  $y_j = -L_y + (j-1)\Delta y$ ,  $j = 1, 2, \dots, M$ .

The conservative finite difference method (FDM), which can also be considered as a node-centered finite volume method (FVM), yields the following semi-discretization of the 2D compressible Navier-Stokes equations

$$\frac{dU_{i,j}}{dt} = -\frac{F^c_{i+\frac{1}{2},j} - F^c_{i-\frac{1}{2},j}}{\Delta x} - \frac{G^c_{i,j+\frac{1}{2}} - G^c_{i,j-\frac{1}{2}}}{\Delta y} + \frac{F^v_{i+\frac{1}{2},j} - F^v_{i-\frac{1}{2},j}}{\Delta x} + \frac{G^v_{i,j+\frac{1}{2}} - G^v_{i,j-\frac{1}{2}}}{\Delta y}, \quad (25)$$

where  $U_{i,j}$  is the approximation of  $U$  at the grid point  $(x_i, y_i)$  for the FDM and the average of  $U$  in the cell

$$\Omega_{i,j} = \left[ x_i - \frac{\Delta x}{2}, x_i + \frac{\Delta x}{2} \right] \times \left[ y_j - \frac{\Delta y}{2}, y_j + \frac{\Delta y}{2} \right].$$

The convective part  $F^c$  and  $G^c$  are the numerical fluxes for the 2D compressible Euler equations. The vector of the conservative variables  $U$  and the flux vectors are defined in (2). The numerical fluxes of the local Lax-Friedrichs method for  $F^c$  and  $G^c$  are defined as follows

$$F_{i+\frac{1}{2},j}^{LF} = 1/2[F(U_{i,j}) + F(U_{i+1,j}) - \max(|u_{i+1,j}| + c_{i+1,j}, |u_{i,j}| + c_{i,j})(U_{i+1,j} - U_{i,j})], \quad (26)$$

$$G_{i,j+\frac{1}{2}}^{LF} = 1/2[G(U_{i,j}) + G(U_{i,j+1}) - \max(|v_{i,j+1}| + c_{i,j+1}, |v_{i,j}| + c_{i,j})(U_{i,j+1} - U_{i,j})]. \quad (27)$$

With the MUSCL approach using the minmod limiter we obtain second order accuracy, except for extrema where the accuracy drop to first order, and also avoid undesired oscillations. In (26)  $U_{i,j}$  is replaced by  $U_{i+\frac{1}{2},j}^L$  and  $U_{i+1,j}$  is replaced by  $U_{i+\frac{1}{2},j}^R$  [10].

$$U_{i+\frac{1}{2},j}^L = U_{i,j} + \frac{1}{2} \minmod(U_{i,j} - U_{i-1,j}, U_{i+1,j} - U_{i,j}) \quad (28)$$

$$U_{i+\frac{1}{2},j}^R = U_{i+1,j} + \frac{1}{2} \minmod(U_{i+2,j} - U_{i+1,j}, U_{i+1,j} - U_{i,j}) \quad (29)$$

where



$$\text{minmod}(a, b) = \begin{cases} a & \text{if } |a| \leq |b| \text{ and } ab > 0 \\ b & \text{if } |b| < |a| \text{ and } ab > 0 \\ 0 & \text{if } ab \leq 0 \end{cases} \quad (30)$$

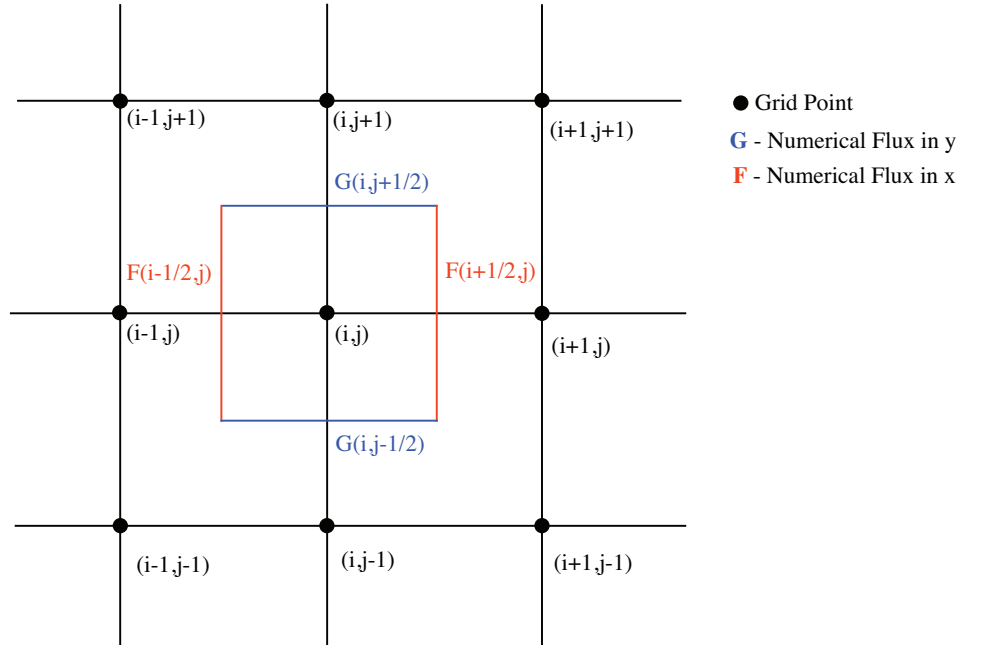
is the minmod limiter. The MUSCL approach is applied similarly to the numerical fluxes  $G_{i,j+\frac{1}{2}}$  in (27).

The viscous terms are discretized by central finite differences, and are shown below.

$$(F_v)_{i+\frac{1}{2},j} = \begin{bmatrix} 0 \\ \frac{4}{3\Delta x}\mu(u_{i+1,j} - u_{i,j}) - \frac{1}{6\Delta y}\mu[(v_{i+1,j+1} - v_{i+1,j-1}) + (v_{i,j+1} - v_{i,j-1})] \\ \frac{1}{4\Delta y}\mu[(u_{i+1,j+1} - u_{i+1,j-1}) + (u_{i,j+1} - u_{i,j-1})] + \frac{1}{\Delta x}\mu(v_{i+1,j} - v_{i,j}) \\ \frac{1}{2}(u_{i+1,j} + u_{i,j})(F_{v,2})_{i+\frac{1}{2},j} + \frac{1}{2}(v_{i+1,j} + v_{i,j})(F_{v,3})_{i+\frac{1}{2},j} + \frac{1}{\Delta x}k(T_{i+1,j} - T_{i,j}) \end{bmatrix} \quad (31)$$

$$(G_v)_{i,j+\frac{1}{2}} = \begin{bmatrix} 0 \\ \frac{1}{\Delta y}\mu(u_{i,j+1} - u_{i,j}) + \frac{1}{4\Delta x}\mu[(v_{i+1,j+1} - v_{i-1,j+1}) + (v_{i+1,j} - v_{i-1,j})] \\ \frac{4}{3\Delta x}\mu(v_{i,j+1} - v_{i,j}) - \frac{1}{6\Delta y}\mu[(u_{i+1,j+1} - u_{i-1,j+1}) + (u_{i+1,j} - u_{i-1,j})] \\ \frac{1}{2}(u_{i,j+1} + u_{i,j})(G_{v,2})_{i,j+\frac{1}{2}} + \frac{1}{2}(v_{i,j+1} + v_{i,j})(G_{v,3})_{i,j+\frac{1}{2}} + \frac{1}{\Delta y}k(T_{i,j+1} - T_{i,j}) \end{bmatrix} \quad (32)$$

**Figure 3:** Sketch of the numerical fluxes in the Cartesian domain.



### 3.2 Discretization of the Boundary Conditions

We use first order boundary conditions, as these have proven to be very robust. Second order conditions are a good choice as well, but a check has to be applied for each iteration to make sure the condition is physical.

When setting the inflow boundary conditions for supersonic flow one needs to specify all the flow variables at the inlet.

$$U_{1,j} = U_{ref} \quad (33)$$

with

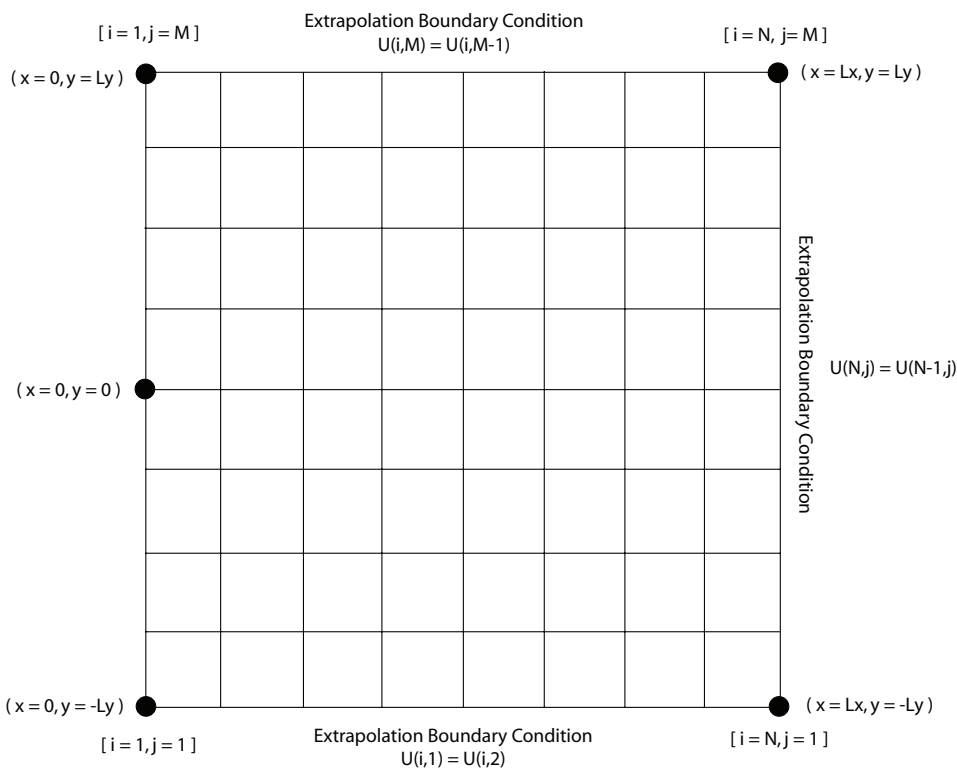
$$\rho_{1,j} = \rho_{ref}, \quad p_{1,j} = p_{ref}, \quad u_{1,j} = u_{ref}, \quad v_{1,j} = 0. \quad (34)$$

Outflow boundary conditions for supersonic flow are not necessary and extrapolation is used.

$$\text{For the east outflow boundary: } U_{N,j} = U_{N-1,j} \quad (35)$$

$$\text{For the north outflow boundary: } U_{i,M} = U_{i,M-1} \quad (36)$$

$$\text{For the south outflow boundary: } U_{i,1} = U_{i,2} \quad (37)$$



**Figure 4:** Sketch of domain and discretized boundary conditions.

When the boundary conditions are handled differently, they are explained for each case in section 5.

### 3.3 Temporal Discretization

When choosing a time integrating method, the possibilities are many. One should take into consideration the computer used for the calculation, the accuracy needed and for how long time the calculation can run. If the computer has fast memory and a lot of it, multistep methods such as the Adam Bashforth method can be used [1]. Other methods that require less memory, but more CPU calculations, are the Runge-Kutta methods. A popular choice is the third order Runge-Kutta (RK3) method. The accuracy is enough for most needs and the method is stable enough to take a relatively large time step. Two different RK3 methods have been considered.

Numerical algorithms with minimal random-access memory (RAM) are essential in computational fluid dynamics. Minimal use of memory is desired to fit a problem in a cache, the high-speed memory tightly coupled to the central processing unit. Effective cache usage is such an important factor in the execution of a numerical code that one is often willing to calculate an extra number of floating point operations if this can fit the problem into a cache more effectively. A RK3 method that can be implemented with reduced storage is the Runge-Kutta-Wray (RKW3) method. This method is third order accurate, but amazingly requires no more memory than the explicit Euler method [1].

For each sub step in the time integrating method we calculate  $\frac{dU}{dt} = \mathbf{R}(U)$ , where  $U$  is the vector of the conservative variables and  $\mathbf{R}$  the residual, i.e. the right hand side of (25), at all interior grid points.

The RKW3 method is given as

$$\begin{aligned} U^{(1)} &= U^n + \frac{8}{15} \Delta t R(U^n), \\ U^{(2)} &= U^{(1)} + \frac{5}{12} \Delta t R(U^{(1)}) - \frac{17}{60} \Delta t R(U^n), \\ U^{(n+1)} &= U^{(2)} + \frac{3}{4} \Delta t R(U^{(2)}) - \frac{5}{12} \Delta t R(U^{(1)}), \end{aligned} \quad (38)$$

where  $R_{i,j} = -\frac{F_{i+\frac{1}{2},j}^c - F_{i-\frac{1}{2},j}^c}{\Delta x} - \frac{G_{i,j+\frac{1}{2}}^c - G_{i,j-\frac{1}{2}}^c}{\Delta y} + \frac{F_{i+\frac{1}{2},j}^v - F_{i-\frac{1}{2},j}^v}{\Delta x} + \frac{G_{i,j+\frac{1}{2}}^v - G_{i,j-\frac{1}{2}}^v}{\Delta y}$  is the residual of the 2D compressible Navier-Stokes equations.

Another choice is the total variation diminishing (TVD) Runge-Kutta methods. These keep the TVD property of the method, as this is a strong stability preserving method (SSP) [4]. Non-TVD but linearly stable Runge-Kutta time discretization can generate oscillations even for TVD spatial discretization, verifying the claim that TVD Runge-Kutta methods are important for such applications [7].

So in order to be sure to keep the TVD property of the discretization, the TVD RK3 method is used.

The TVD RK3 method used is given as [7]

$$\begin{aligned} U^{(1)} &= U^n + \Delta t R(U^n), \\ U^{(2)} &= \frac{3}{4} U^n + \frac{1}{4} U^{(1)} + \frac{1}{4} \Delta t R(U^{(1)}), \\ U^{(n+1)} &= \frac{1}{3} U^n + \frac{2}{3} U^{(2)} + \frac{2}{3} \Delta t R(U^{(2)}). \end{aligned} \quad (39)$$

### 3.4 Stability of the Numerical Scheme

The stability of the time iteration method is dependent on the eigenvalues of the Jacobian matrices of the flux vectors and the properties of the time discretization method. To make sure not to take a too large time step the Courant-Friedrichs-Lewy (CFL) condition number and the von Neumann number (VNN) are used. For the 2D compressible Navier-Stokes they are defined as follows

$$CFL = \Delta t_I \left( \frac{|u|+c}{\Delta x} + \frac{|v|+c}{\Delta y} \right), \quad (40)$$

$$VNN = \Delta t_V \left( \frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \max \left( \frac{A\mu}{3\rho}, \frac{\gamma\mu}{\rho Pr} \right). \quad (41)$$

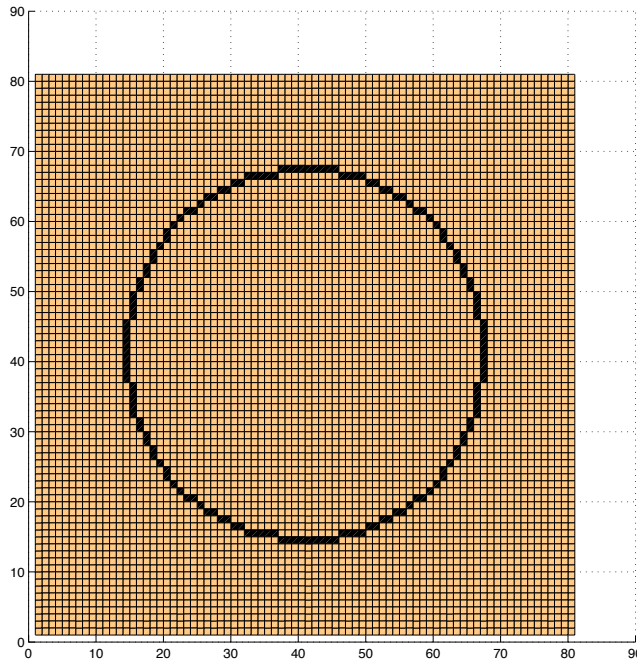
The CFL and VNN condition numbers are then set for each simulation. The VNN has been set as 0.4 while the CFL number is varied from 0.5 to 0.25, depending on the simulation. The time step sizes  $\Delta t_I$  and  $\Delta t_V$  is calculated from the CFL and the VNN condition numbers for every time step and the minimum of both is chosen as  $\Delta t$  in the RK3.

## 4 Ghost Point Treatment at Embedded Boundary

### 4.1 Simplified Ghost Point Treatment for Embedded Boundary in 2D

In the simplified ghost point treatment, the fluid point F adjacent to the embedded boundary is chosen as a mirror point. One then assumes the embedded boundary is in the middle between ghost and fluid points. In figure 6 the idea is shown on a 9x9 grid. In figure 5 the flagged ghost cells of the cylinder generated from the code are shown on a finer grid.

**Figure 5:** Flagged ghost cells on a (81x81) grid. Back points indicate the location of ghost points.



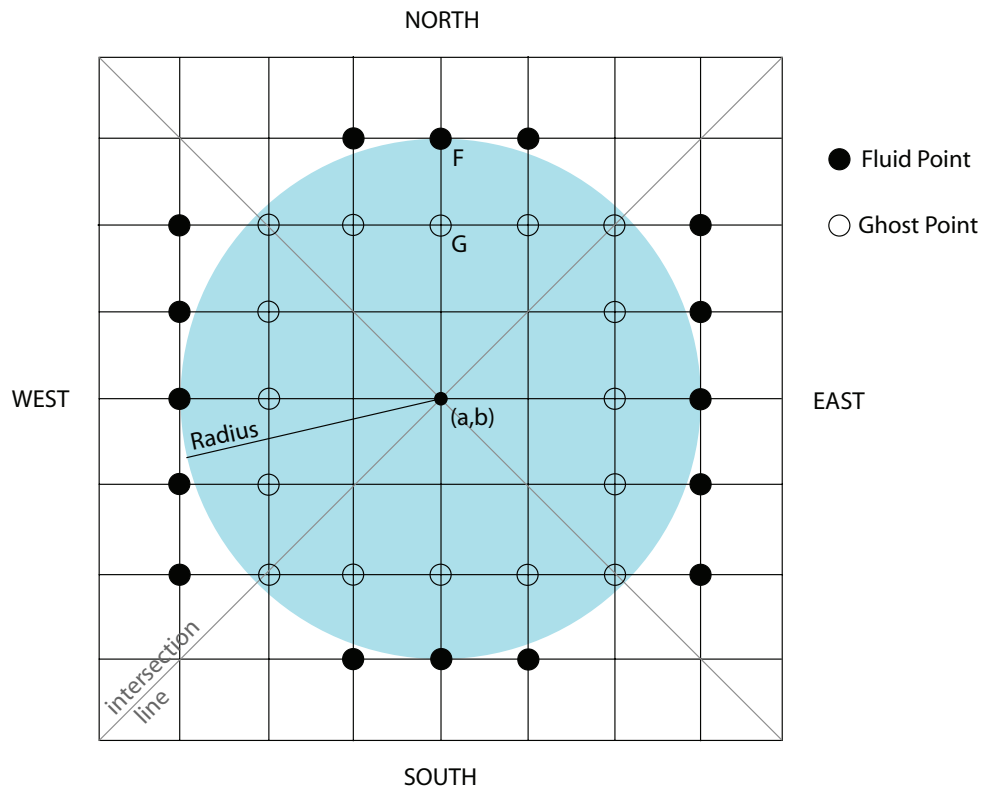
The primitive variables at the ghost points are set as

$$\rho_G = \rho_F, \quad u_G = -u_F, \quad v_G = -v_F, \quad p_G = p_F, \quad (42)$$

assuming the embedded boundary to lie in the middle between ghost point G and mirror point F. At the embedded boundary, the boundary conditions  $u = v = 0$  and  $\frac{\partial \rho}{\partial n} = \frac{\partial p}{\partial n} = 0$  are assumed. The mirror points F have to be carefully chosen. On the west side of the domain the ghost point values are calculated using the values at the fluid points to the west of the ghost points.  $U_{iG,jG} = U_{iG-1,jG}$ .

On the south side of the domain the ghost points are calculated using the fluid points south of the ghost points.  $U_{iG,jG} = U_{iG,jG-1}$ . On the east side of the domain the ghost points are calculated using the fluid points to the east of the ghost points.  $U_{iG,jG} = U_{iG+1,jG}$ . On the north side of the domain the ghost points are calculated using the fluid points to the north of the ghost points.  $U_{iG,jG} = U_{iG,jG+1}$ .

**Figure 6:** Illustration of the fluid points chosen as mirror points and the ghost points set to create a solid cylinder.

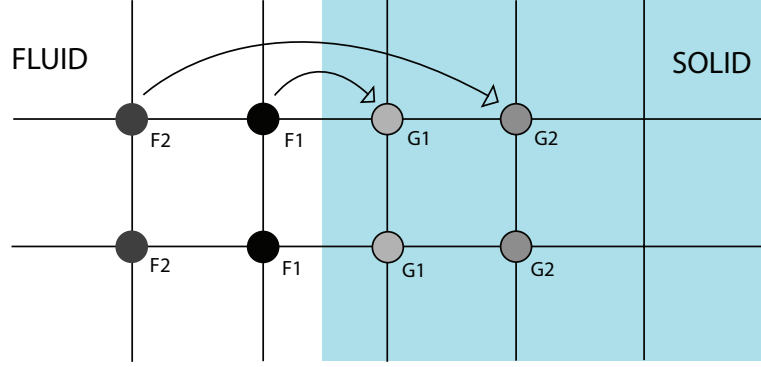


Four points on the grid are located exactly on the circle, i.e. the intersection of the grid lines through the center point (a,b) with the circle. These have to be taken care of explicitly and are set as fluid points. See figure 6.

Four ghost points on the grid are located on the 45-degree intersection lines. On this line a ghost point can be calculated using the nearest fluid point either in the x-direction or in the y-direction. In this code these ghost points are calculated using the nearest fluid point that lies in the x-direction. Another approach could be to choose the nearest fluid point that lies on the intersection line, the diagonal.

## 4.2 Ghost Point Treatment with Second Order Accuracy

When using the minmod limiter in the MUSCL approach to achieve second order accuracy, a new set of ghost points is needed. Since the minmod limiter also uses the values  $U_{i\pm 2, j}$  and  $U_{i, j\pm 2}$ , ghost point values must be set at the points  $(x_{i\pm 2}, y_j)$  and  $(x_i, y_{j\pm 2})$  if they lie inside the solid. This is done like illustrated in figure 7 by setting the primitive variables at the ghost points as  $\rho_{G2} = \rho_{F2}$ ,  $u_{G2} = -u_{F2}$ ,  $v_{G2} = -v_{F2}$  and  $p_{G2} = p_{F2}$ .



**Figure 7:** Illustration of ghost point calculations for MUSCL.

## 4.3 The SP and KP Embedded Boundary Method

A Cartesian grid developed by Kreiss and Petersson [5,6] will from now be referred to as the KP embedded boundary method. This is a very accurate boundary procedure developed for wave propagation problems. A normal to the boundary through the ghost point is defined, and function values in the interior of the domain of the normal are obtained by quadratic interpolation in the  $x$ -direction. If the normal has slope less than one, the interpolation is instead done in the  $y$ -direction. Finally the Dirichlet condition or the Neumann condition is approximated by a formula involving the values found in the interior [3].

Sjögreen and Petersson developed another method [8]. This will be called the SP embedded boundary method. This method is a robust, but less accurate, boundary procedure for problems with discontinuous solutions [8]. The SP method is similar to the KP method, but uses more values along the normal and the interpolation is linear instead of quadratic. These methods are more complex, use more fluid points and do a good job at identifying the exact wall of the embedded boundary.

## 4.4 Simplified Ghost Point Treatment for the Compressible Euler Equations

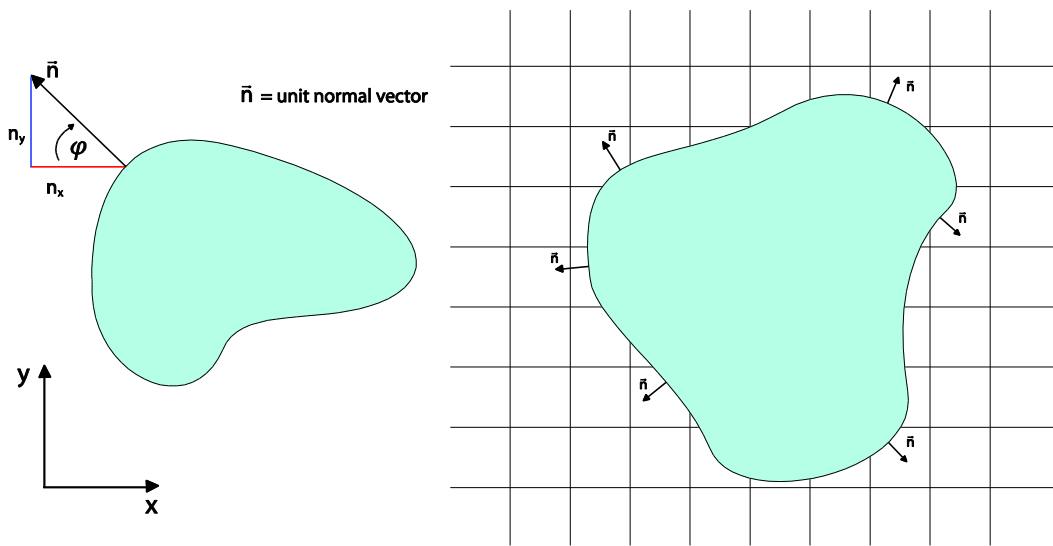
The wall boundary condition is different when dealing with inviscid flow. Although the same fluid and ghost points are chosen as with the compressible Navier-Stokes equations the values assigned are different. For inviscid flow only the normal velocity component is zero at the wall. The tangential velocity component is treated symmetrically like density and pressure. Then the primitive variables at the ghost points are set as in [9].

$$\rho_G = \rho_F, p_G = p_F, u_G = u_F - 2(n_1 u_F + n_2 v_F)n_1, v_G = v_F - 2(n_1 u_F + n_2 v_F)n_2. \quad (43)$$

where  $n_1$  and  $n_2$  are the x- and y-components of the unit normal vector.

#### 4.5 New Simplified Ghost Point Treatment for Embedded Boundary in 2D

The New Simplified method is developed to be more accurate than the Simplified method. The new method is still very similar, but it will choose its mirror fluid points from more locations than the old method. Instead of only choosing the mirror points on the grid lines through the ghost points in the x- or y-directions, the new method also considers all diagonal directions. The New Simplified method chooses its mirror fluid points from a total of eight different locations, depending on the unit normal component of the surface nearest to the ghost point. In figure 8 there are two blunt bodies of random shape. Their normal component is calculated on the surface, and the angle of the unit normal is calculated with respect to the x-axis such that if  $n_x = -1$  and  $n_y = 0$  then  $\varphi = 0^\circ$ .

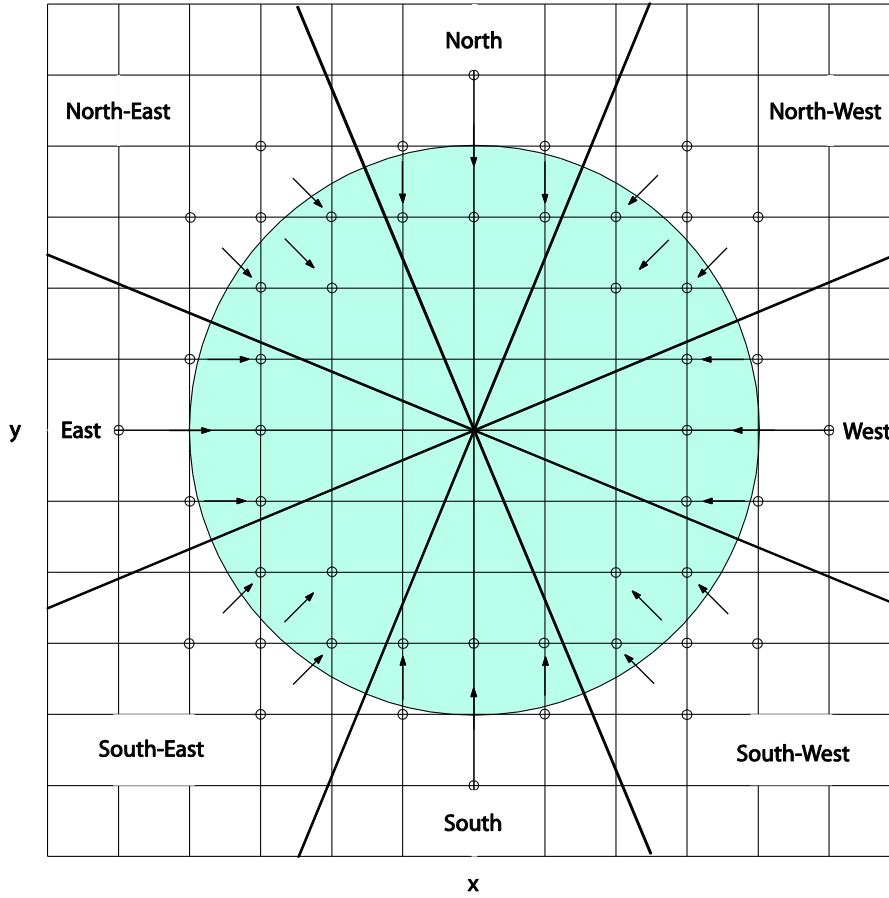


**Figure 8:** Angle used to identify fluid point (left). Normal components on body (right).

If  $(i, j)$  represents the index of a ghost point then  $(iF, jF)$  represent the index of the fluid point chosen as the mirror point. The general relationship to choose all fluid point indices is dependent on the angle  $\varphi$  and the relation is given below.

$$\begin{aligned}
 & \text{if } -22.5 < \varphi \leq 22.5 \text{ then } (iF, jF) = (i - 1, j) \\
 & \text{if } 22.5 < \varphi \leq 67.5 \text{ then } (iF, jF) = (i - 1, j + 1) \\
 & \text{if } 67.5 < \varphi \leq 112.5 \text{ then } (iF, jF) = (i, j + 1) \\
 & \text{if } 112.5 < \varphi \leq 157.5 \text{ then } (iF, jF) = (i + 1, j + 1) \\
 & \text{if } 157.5 < \varphi \leq 202.5 \text{ then } (iF, jF) = (i + 1, j) \\
 & \text{if } 202.5 < \varphi \leq 247.5 \text{ then } (iF, jF) = (i + 1, j - 1) \\
 & \text{if } 247.5 < \varphi \leq 292.5 \text{ then } (iF, jF) = (i, j - 1) \\
 & \text{if } 292.5 < \varphi \leq 337.5 \text{ then } (iF, jF) = (i - 1, j - 1)
 \end{aligned} \tag{44}$$

For a circular cylinder the fluid points will be chosen like shown in figure 9.



**Figure 9:** Fluid points (small circles) in white fluid domain used as mirror points for ghost points (small circles) in green solid domain in the New Simplified method.

In figure 9 the arrows indicate the transfer of fluid data from the mirror points to the ghost points. In this way we can apply the New Simplified method to a variety of different complex geometries.

### Weighted Ghost Point Method

The method of weighing the ghost point values is only applied to the cylinder due to the added complexity of the method. The wall of the immersed body is identified and the velocities at the ghost points are weighed using a scaling factor in order to improve the accuracy of the wall boundary treatment. By applying these weights to the velocity components at the ghost points we try to reflect the embedded boundary closer to its actual location. Without the weights the embedded boundary is assumed to lie in the middle between ghost and fluid points.

The method is illustrated in figure 10. The distance from the ghost point to the immersed wall boundary is called  $a$ , and  $b$  is the distance from the appropriate fluid point to the immersed wall boundary. The distance  $a+b$  is equal to the distance between the ghost point and the corresponding fluid point. The primitive variables at the ghost points for viscous flow are set as

$$\rho_{G1} = \rho_{F1}, \quad u_{G1} = -\left(\frac{a}{b}\right)u_{F1}, \quad v_{G1} = -\left(\frac{a}{b}\right)v_{F1}, \quad p_{G1} = p_{F1}, \quad (45)$$

while with the second order MUSCL scheme the primitive variables for viscous flow are set as

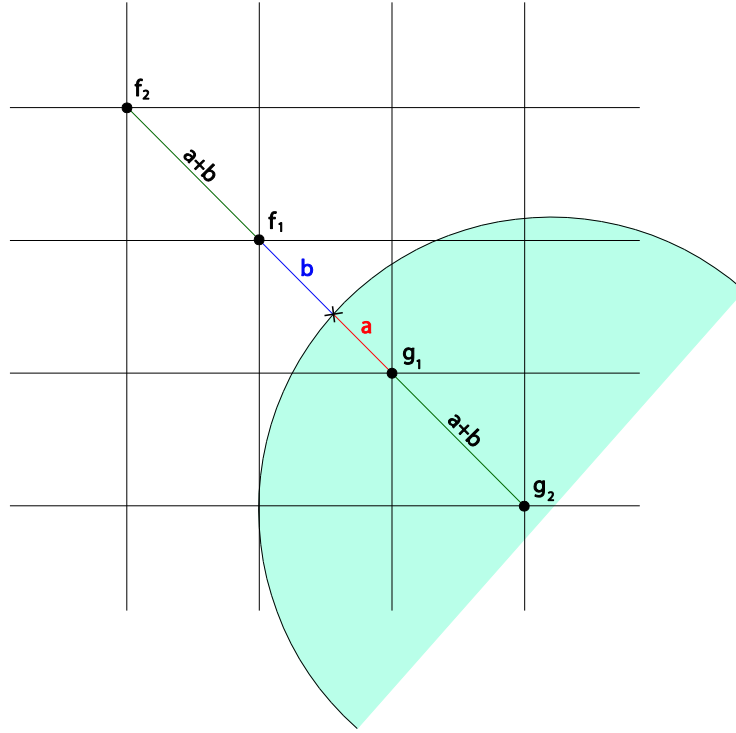


$$\rho_{G2} = \rho_{F2}, \quad u_{G2} = -\left(\frac{a+a+b}{b+a+b}\right)u_{F2}, \quad v_{G1} = -\left(\frac{a+a+b}{b+a+b}\right)v_{F2}, \quad p_{G2} = p_{F2}. \quad (46)$$

To avoid instabilities we enforce a maximum and minimum value to the  $\left(\frac{a}{b}\right)$  scaling such that

$$\left(\frac{a}{b}\right)_{max} = 2 \quad \text{and} \quad \left(\frac{a}{b}\right)_{min} = 0.5. \quad (47)$$

If the maximum or minimum values are enforced, the distances  $a$  and  $b$  are set to mach. For example if  $a = 10$  and  $b = 2$  then the length  $a$  will be set as  $a = 8$  and the length  $b$  will be set as  $b = 4$ . This must be done to keep the right scaling on the velocities in (46).



**Figure 10:** Identifying the embedded boundary.

Note that the normal component at the immersed wall boundary is not used, but only the distance to the wall between the ghost and fluid points. In all test cases featuring a circular cylinder the weighted ghost point method is applied unless it is stated otherwise.

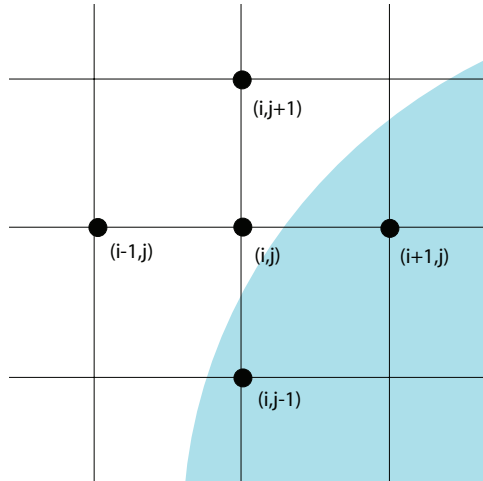
#### 4.6 Calculating the Shear Stress on a Body.

With the simplified ghost point treatment, it is somewhat difficult to exactly identify the wall of the chosen embedded body, as it lies between two points. The solution devised here was to average the skin friction on points near the wall. Several points close to the wall of the embedded body have been used.

If  $(\tau_p)_{i,j}$  is the shear stress of the fluid point closest to the wall the following averaging is used to obtain the averaged wall shear stress  $\tau_{w(i,j)}$ . See figure 11 and (48).

$$\tau_{w(i,j)} = [ 2(\tau_p)_{i,j} + (\tau_p)_{i-1,j} + (\tau_p)_{i+1,j} + (\tau_p)_{i,j-1} + (\tau_p)_{i,j+1} ]^{\frac{1}{6}}. \quad (48)$$

**Figure 11:** The points used in the averaging of the shear stress at the wall.

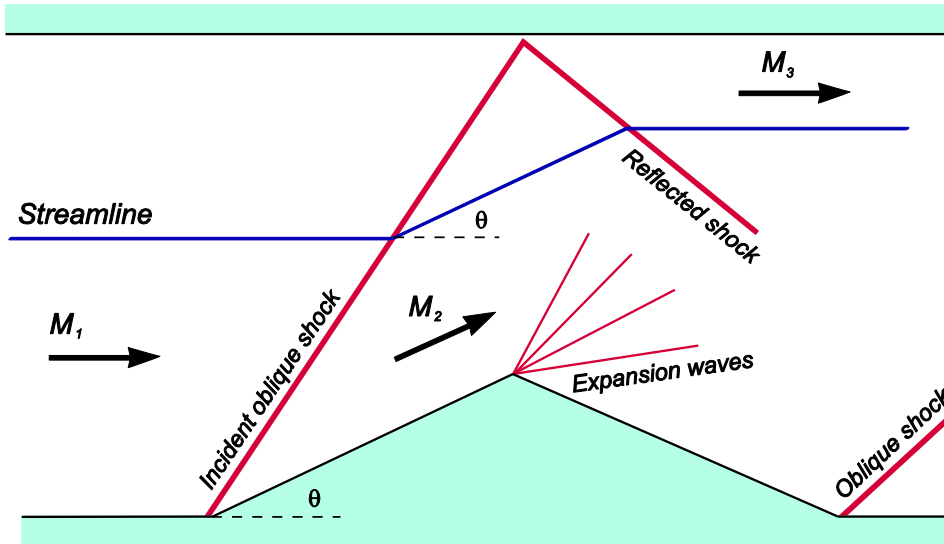


## 5 Verification of the New Simplified Ghost Point Treatment

### 5.1 Supersonic Flow around a Diamond-Wedge Airfoil with Compressible Euler Equations

To test the shock wave behavior of the method a supersonic inviscid flow past a diamond-wedge airfoil has been considered. For this type of flow a number of analytical solutions exist and are compared to the simulations where possible. The first simulation has a free-stream Mach number  $M_\infty = 3$ , and the deflection angle of the diamond is  $\theta = 15^\circ$ . A grid with  $400 \times 200$  cells has been used to discretize the computational domain  $[-0.75, 3.25] \times [-1, 1]$ . The chord length of the diamond-wedge airfoil is  $c = 2$ , whose leading edge is located at the origin. An inviscid wall boundary condition is implemented at the upper boundary  $y = 1$ . There are supersonic inflow conditions at  $x = -0.75$ , and supersonic outflow conditions at  $x = 3.25$  and at  $y = -1$ . First order accuracy is used. The inviscid wall boundary at  $y = 1$  are set as symmetry conditions and the primitive variables are set like

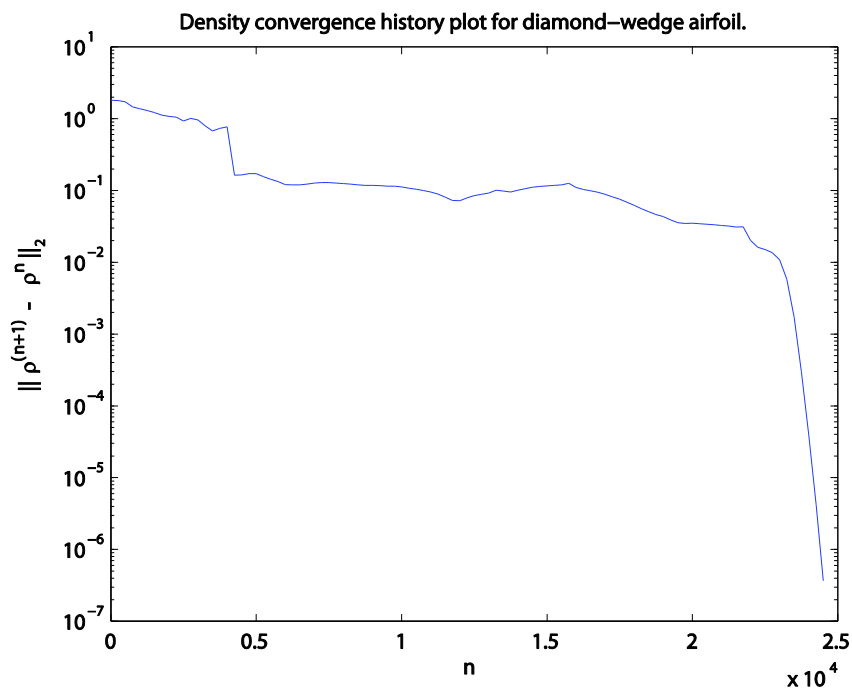
$$\begin{aligned}\rho_{i,M} &= \rho_{i,M-1}, \\ u_{i,M} &= u_{i,M-1}, \\ v_{i,M} &= -v_{i,M-1}, \\ p_{i,M} &= p_{i,M-1}.\end{aligned}$$



**Figure 12:** Illustration of expected shock wave behavior in a channel with a diamond-wedge.

For a channel flow with a diamond-wedge we expect the flow to develop like illustrated in figure 12 [18]. The first incident shock will be reflected at the upper wall, and at the top of the diamond the supersonic flow will accelerate through expansion waves, as the area in this part increases. Finally an oblique shock wave will settle at the trailing edge where the flow is turned parallel to the x-axis.

The residual of the continuity equation measured as  $\|\rho^{n+1} - \rho^n\|_2$  is reduced by more than six order of magnitude in 25000 time steps. The convergence history is shown in figure 13. As seen in figure 13, once the flow settles the solution converges quickly. There are no upstream reflections, because the flow is supersonic at all outflow boundaries.

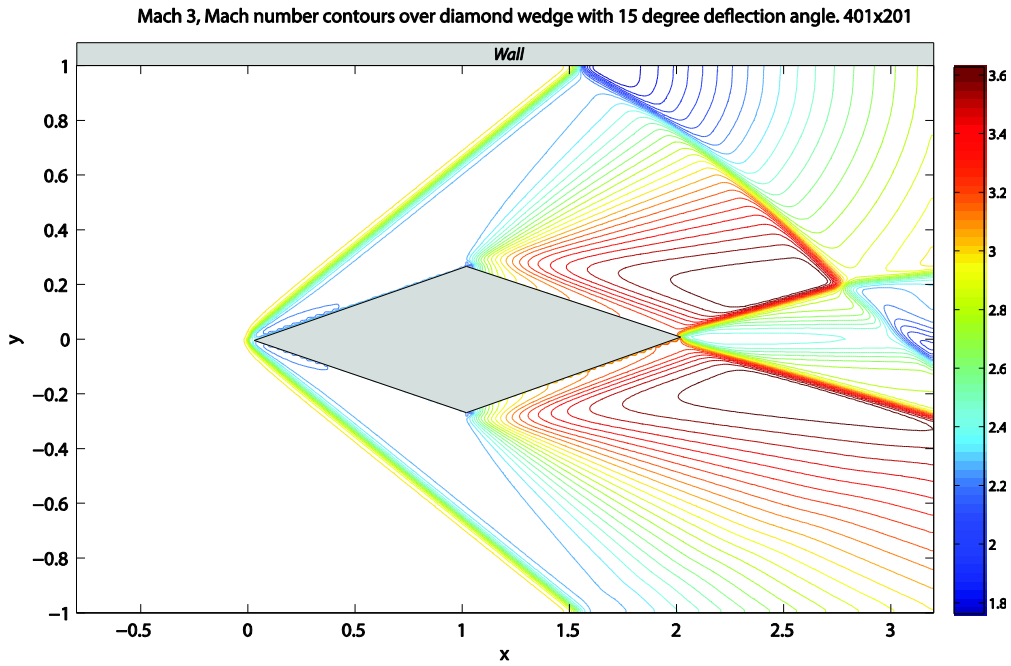


**Figure 13:** Residual convergence history for inviscid supersonic flow past a diamond-wedge airfoil at a Mach number of 3 and a deflection angle of  $15^\circ$ .

The calculated incident oblique shock wave angle from the simulation is  $\beta \cong 32^\circ$  and from the  $\theta, \beta, M$ - relation from (49) and [17] the shock-wave angle should be  $\beta \cong 32.2^\circ$ . Due to the inaccuracy of calculating an angle from a simulation result, we are content with this result. The  $\theta, \beta, M$ - relation is given as

$$\tan(\theta) = 2\cot(\beta) \frac{M_1^2 \sin^2(\beta) - 1}{M_1^2 (\gamma + \cos(2\beta) + 2)}, \quad (49)$$

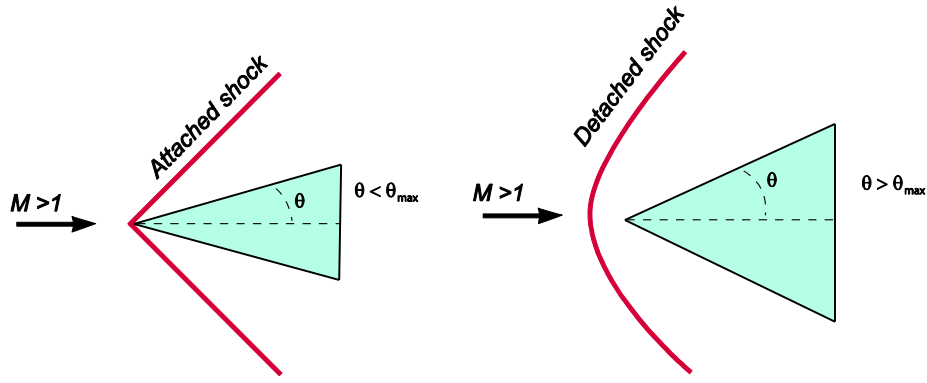
where  $\beta$  is the shock wave angle,  $\theta$  is the deflection angle and  $M_1$  the Mach number in front of the shock (the free stream Mach number in this simulation).



**Figure 14:** Mach number contours past a diamond-wedge airfoil with attached shock, and shock reflection at upper wall boundary,  $M_\infty = 3$ ,  $\theta = 15^\circ$ .

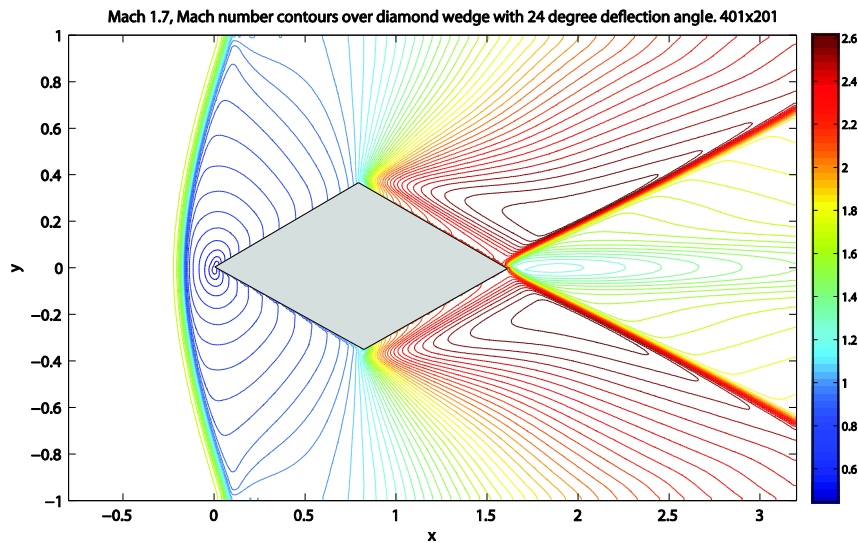
In figure 14 we observe an attached oblique shock wave at the leading edge of the diamond-wedge airfoil. At the wall we see a reflected shock, and at the trailing edge we see oblique shocks attached to the tail of the diamond-wedge. Some shock interactions are also in the wake, where the reflected shock from the wall intersect the oblique shock from the tail.

The second simulation with the diamond-wedge airfoil has a free-stream Mach number  $M_\infty = 1.7$ , and the deflection angle of the diamond is  $\theta = 24^\circ$ . A grid with  $400 \times 200$  cells has been used to discretize the computational domain  $[-0.75, 3.25] \times [-1, 1]$ . The chord length of the diamond-wedge airfoil is  $c = 1.6$ , whose leading edge is located at the origin. Supersonic inflow boundary conditions are set at the inflow boundary  $x = -0.75$ , while supersonic outflow boundary conditions are set at all other boundaries.



**Figure 15:** Effects of increasing the deflection angle such that we observe a detached shock instead of an attached shock in flow past a wedge.

The goal with this simulation is to observe a detached shock. The physical geometry in this case is such that  $\theta > \theta_{max}$ , and no solution exists for a straight oblique shock wave. Instead, nature establishes a curved shock wave detached from the apex of the diamond-wedge [18]. This is confirmed by the numerical simulation using the first order node centered finite volume method with the new simplified ghost point treatment in figure 16 and illustrated in figure 15. The maximum deflection angle  $\theta_{max}$  for a flow with  $\gamma = 1.4$  and  $M_{\infty} = 1.7$  is  $\theta_{max} \cong 17^{\circ}$  [17]. In this simulation we have chosen a deflection angle larger than  $\theta_{max}$ . As seen in figure 15 we observe a strong bow shock-wave detached from the nose of the diamond-wedge airfoil.

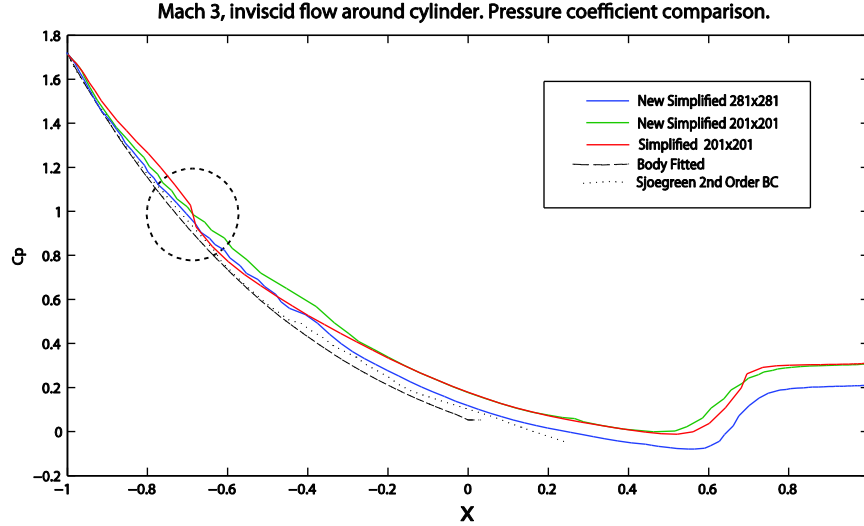


**Figure 16:** Mach number contours past a diamond-wedge airfoil with a detached shock,  $M_{\infty} = 1.7$ ,  $\theta = 24^{\circ}$ .

## 5.2 Supersonic Flow over a Circular Cylinder with Compressible Euler Equations

The supersonic inviscid flow past a circular cylinder with  $M_{\infty} = 3$  has been considered. The computational domain is  $[-2, 2] \times [-2, 2]$ . The diameter of the cylinder is  $D = 1$  and its center is located at  $(x, y) = (0, 0)$ . The boundary conditions for all test cases are for supersonic inflow at the boundary  $x = -2$ , and those for supersonic outflow for all other boundaries. First order accuracy is used.

For the inviscid case two grids with  $200^2$  and  $280^2$  cells have been used with the New Simplified method and one case with a grid of  $200^2$  cells has been used with the Simplified method. These simulations have been used to check the pressure coefficient versus Sjögreen and Petersson (2007) [8] results and are shown in figure 17.



**Figure 17:** Comparison of the pressure coefficient obtained from the New Simplified method, the Simplified method and the results from Sjögreen and Petersson (2007) [8].

The Simplified method has a problem in the area where the fluid points used to mirror the ghost points change direction from  $x$  to  $y$ . This is seen as a sudden bump in the pressure coefficient near  $x = 0.7$  in figure 17. A circle is placed in the region of the bump in figure 17. A goal with developing a new method was to remove that bump. With the New Simplified method that bump is indeed no longer present. The pressure coefficient is slightly higher than the results from Sjögreen and Petersson (2007) [8], but with grid refinement the pressure coefficient comes closer to their results. It is worthwhile mentioning that Sjögreen and Petersson [8] used a  $305 \times 305$  grid.

### 5.3 Moving Shock Wave over a Circular Cylinder with Compressible Euler Equations

To test the transient solution of the method we have chosen to replicate a test case done by H. Luo et al. (2006) [14]. An incident moving shock at  $M_s = 2$  past a circular cylinder in a channel is computed. A uniform grid with  $300 \times 150$  cells has been used to discretize the computational domain  $[-3, 3] \times [-1.5, 1.5]$ . The diameter of the cylinder is  $D = 1$  and its center is located at  $(0, 0)$ .

The moving shock wave is initialized as a moving shock wave at  $x = -3$  and propagates towards the cylinder. The boundary conditions for the inflow are set as for a moving shockwave using the Rankine-Hugoniot conditions, the outflow boundary at  $x = 3$  is set as for supersonic outflow, while the wall has been created using symmetry boundary conditions at the upper and lower boundaries. The relations between the primitive variables to the left of the shock and to the right of the shock for  $M_s = 2$  are  $\frac{p_1}{p_2} = 0.375$ ,  $\frac{p_2}{p_1} = 4.5$  and  $\Delta u = 1.25c_1$ , where  $c_1$  is the speed of sound to the right of the shock.

The primitive variables at the north wall  $y = 1.5$ , are set as

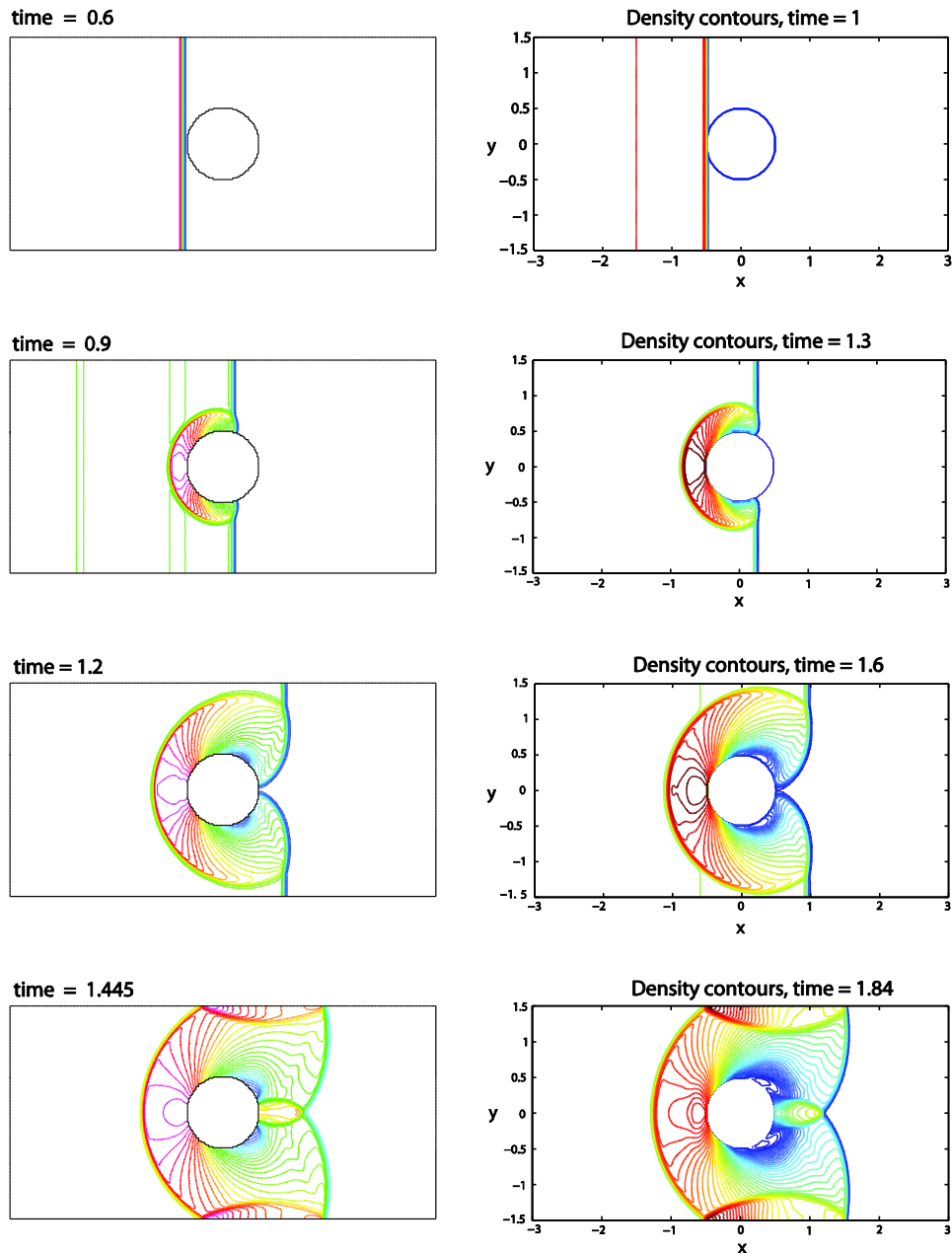
$$\rho_{i,M} = \rho_{i,M-1}, \quad u_{i,M} = u_{i,M-1}, \quad v_{i,M} = -v_{i,M-1}, \quad p_{i,M} = p_{i,M-1}.$$

The primitive variables at the south wall  $y = -1.5$ , are set as

$$\rho_{i,1} = \rho_{i,2}, \quad u_{i,1} = u_{i,2}, \quad v_{i,1} = -v_{i,2}, \quad p_{i,1} = p_{i,2}.$$

Hybrid method, H. Luo et al. (2006)

New Simplified method



**Figure 18:** Comparison of computed density contours for a incident shock  $M_s = 2$  past a circular cylinder at different times obtained using the hybrid method [13] and the New Simplified method.

It is not known how H. Luo et al. (2006) [13] calculated the dimensionless time and where in the channel they initialized the moving shock wave. We have chosen a dimensionless time that we find to give almost the same time intervals as [13], and it is chosen as

$$t^* = \frac{t}{D} (\Delta u), \quad (50)$$

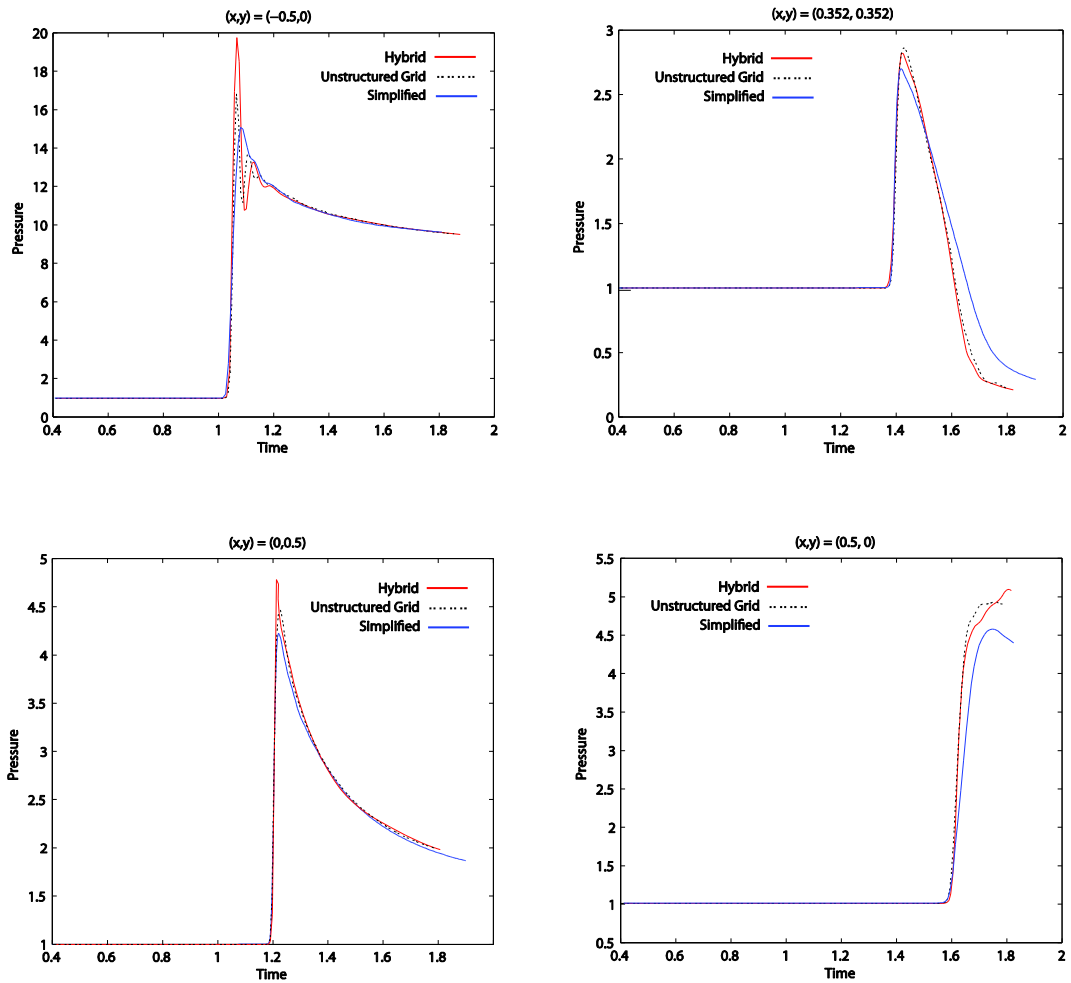
where  $\Delta u = u_2 - u_1$  and  $u_2$  is the velocity of the flow in region 2 while  $u_1$  is the velocity of the flow in region 1 in figure 2a.

Our results give us a time that is consecutively 0.4 higher than H. Luo et al. (2006) [13]. Therefore we conclude that they have started the moving shock wave another place than we have, probably at  $x = -2$ , and continue the comparison.

The computed density contours in the flow field at different times obtained using the New Simplified method are compared to the hybrid method results of H. Luo et al. (2006) [13] in figure 18. The idea behind the hybrid method is to apply a gridless method to cells in the vicinity of a solid boundary and a conventional Cartesian grid method to all other cells. Figure 19 shows a comparison of pressure time histories at different locations between the New Simplified method and the results of H. Luo et al (2006).

As seen in figure 18 the transient results look very similar, although it looks like our solution has traveled slightly faster. The reflected shock wave in the last figure at time = 1.84 looks bigger. One way to explain the difference could be that the shock moves very fast through this domain, and the smallest increase in time makes a relevant change in the result, so the time comparison might not be completely accurate. Overall we get the same behavior, and the time dependent solutions look almost the same.





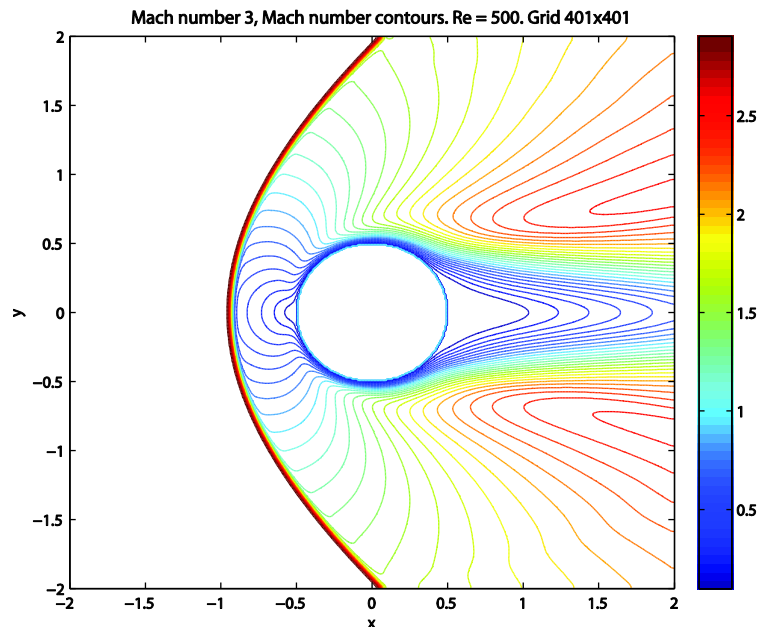
**Figure 19:** Comparison of pressure at different times obtained using the hybrid method, the unstructured grid method [13] and the New Simplified method.

Figure 19 shows the comparison of the pressure time histories at four points  $(x,y) = (-0.5,0)$ ,  $(x,y) = (0.325,0.325)$ ,  $(x,y) = (0,0.5)$ ,  $(x,y) = (0.5,0)$ , [13] between the New Simplified method, the hybrid method [13] and the unstructured grid method from H. Luo et al (2006) [13]. Note that the original results by H. Luo et al (2006) have been moved forward in time by 0.4. This is done to be able to compare the results, because the shock wave was not initialized at the same location in the domain. The pressure obtained using the New Simplified method is generally lower than with the two other methods. We are also not able to catch the oscillating behavior that is seen at  $(x,y) = (-0.5,0)$ . An important difference here is that the New Simplified method uses an equidistant grid, while H. Luo et al. (2006) [13] have more points available near the cylinder's embedded boundary due to the use of a meshless method employed there for the hybrid method.

## 5.4 Supersonic Flow over a Circular Cylinder with Compressible Navier-Stokes Equations

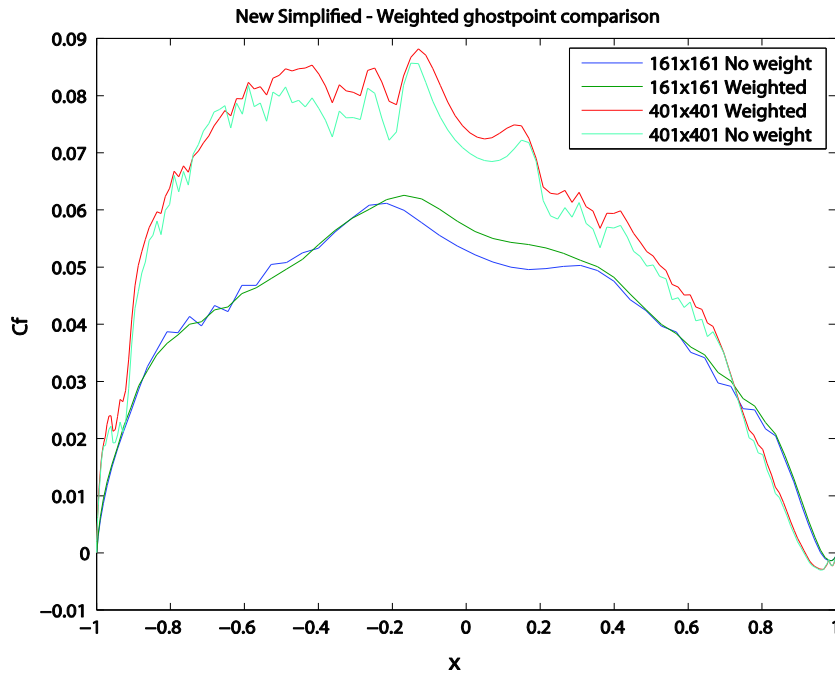
The goal in this subchapter is 1) To compare the New Simplified method versus the Simplified method. 2) To compare the New Simplified method with weighted ghost points versus the New Simplified method without weighted ghost points. 3) Test the New Simplified method versus a well documented Cartesian grid method for viscous flow at supersonic flow conditions.

The supersonic flow past a circular cylinder with  $M_\infty = 3$  and  $Re_D = 500$  has been considered. Three grids with  $160^2$ ,  $280^2$  and  $400^2$  cells have been used to discretize the computational domain  $[-2, 2] \times [-2, 2]$ . The diameter of the cylinder is  $D = 1$  and its center is located at  $(0,0)$ . The cylinder wall is assumed to be adiabatic. The boundary conditions for all test cases are for supersonic inflow at the boundary  $x = -2$ , and those for supersonic outflow for all other boundaries. Second order accuracy except for extrema is reached with the MUSCL scheme.



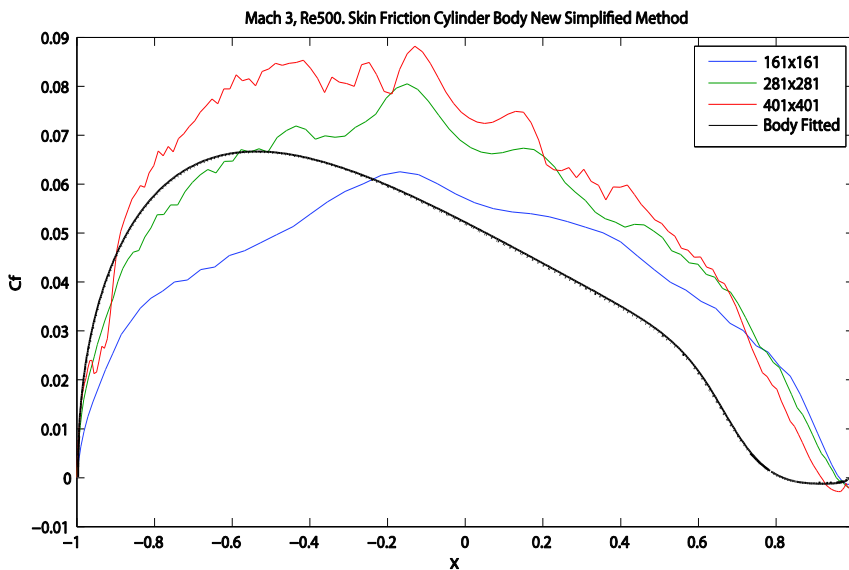
**Figure 20:** Mach number contours of supersonic flow past a cylinder obtained using the New Simplified method.

Figure 20 shows the Mach number contours. We observe a sharply resolved bow shock. Some of the flow in the wake of the cylinder exits the domain at a Mach number lower than one, and supersonic outflow boundary conditions are not correct in this region. However, the area is small and does not seem to affect the solution.



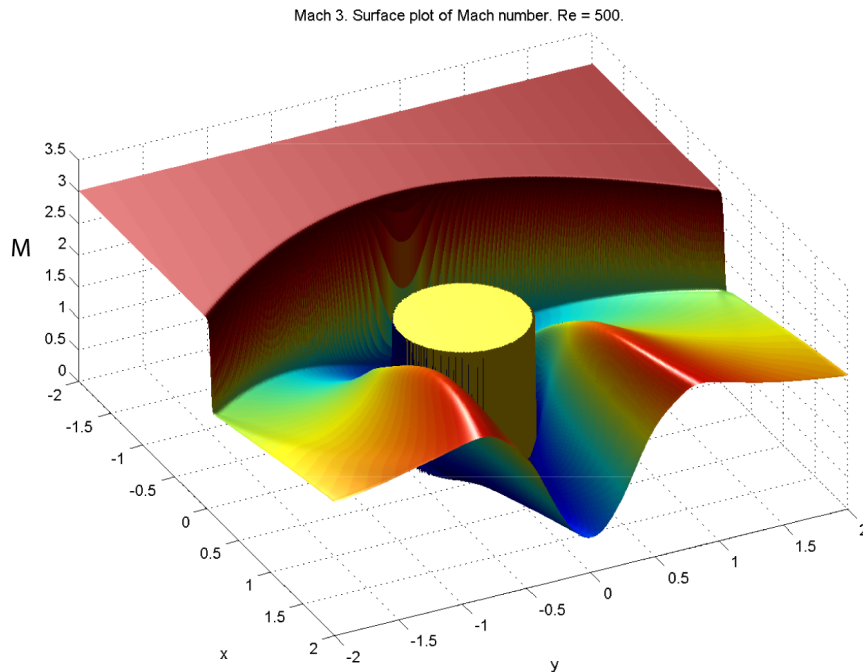
**Figure 21:** Comparison of the skin friction coefficient obtained from the weighted ghost point New Simplified Method and the New Simplified method without weighted ghost points.

The skin friction profiles obtained from the New Simplified method with and without weighted ghost points are shown in figure 21. From these results we see that some of the bumps become smaller or disappear completely, as was the idea of improving the accuracy by weighing the ghost points. Without the weights the embedded boundary would always lie in the middle between ghost and fluid points, which is not always the case, and we believed this to be a reason for the bumps. With the weights one attempts to have the ghost point velocity components better reflect the actual location of the embedded boundary.



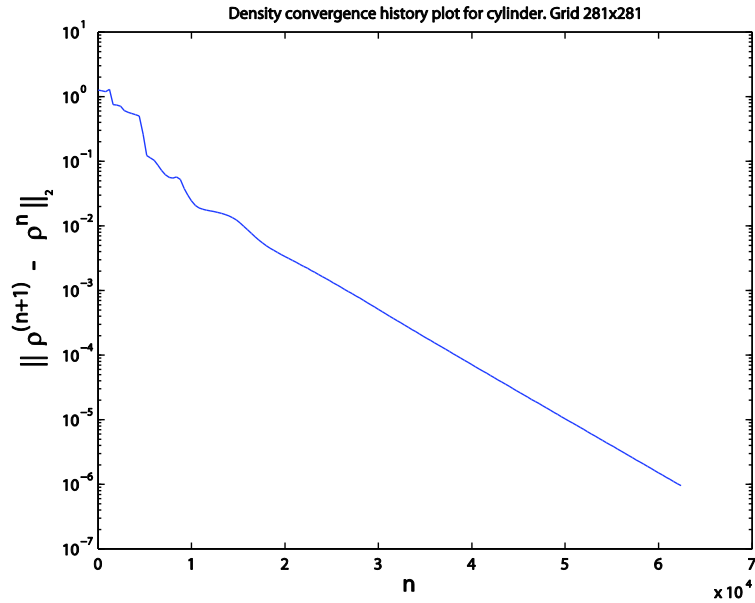
**Figure 22:** Skin friction profile obtained from the New Simplified method with grid refinement compared with a reference skin friction profile [3].

In figure 22 the skin friction is plotted for the various grid sizes and compared to the body fitted results from Sjögreen (2009). As the resolution increases the bumps in the skin friction coefficient become more evident. The results from the coarsest grid actually looks like the best solution, so no clear grid convergence is shown.



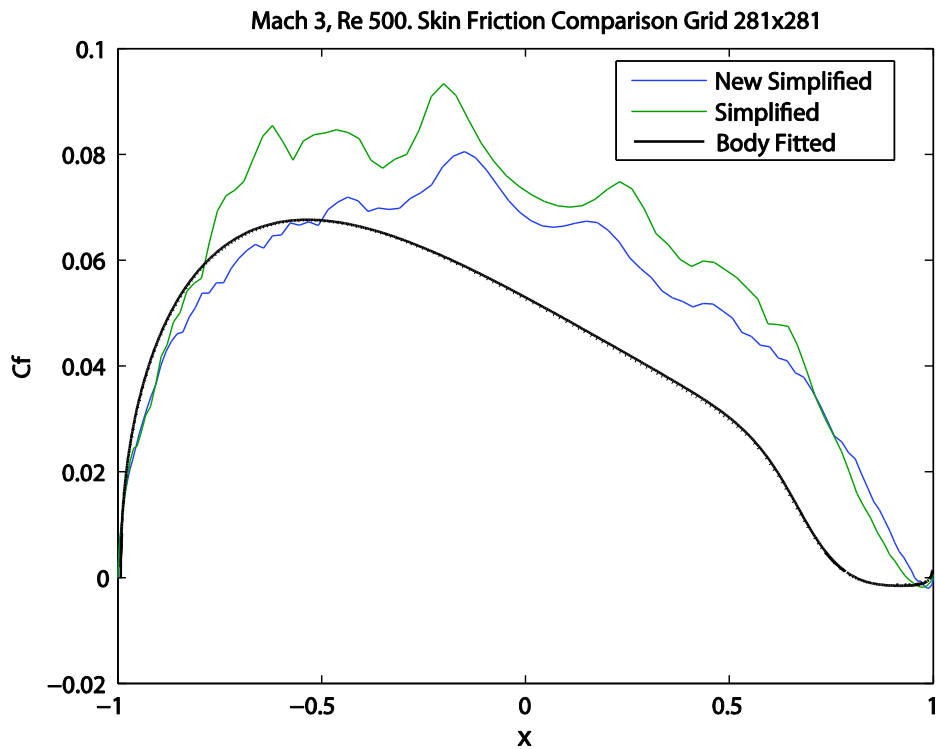
**Figure 23:** Surface plot of supersonic flow past a circular cylinder with second order accuracy on a 401x401 grid.

The surface plot in figure 23 is added to get a good visualization of how the flow behaves around the cylinder. The use of an equidistant grid rewards us with a sharply defined bow shock throughout the domain. A circular three dimensional cylinder is added in post processing.



**Figure 24:** Convergence history plot obtained from the New Simplified method on supersonic Mach 3 flow with Reynolds number 500 around a circular cylinder.

The time stepping is stopped when the residual of the density variable reach  $10^{-6}$ , and the convergence history for the 281x281 grid is seen in figure 24.



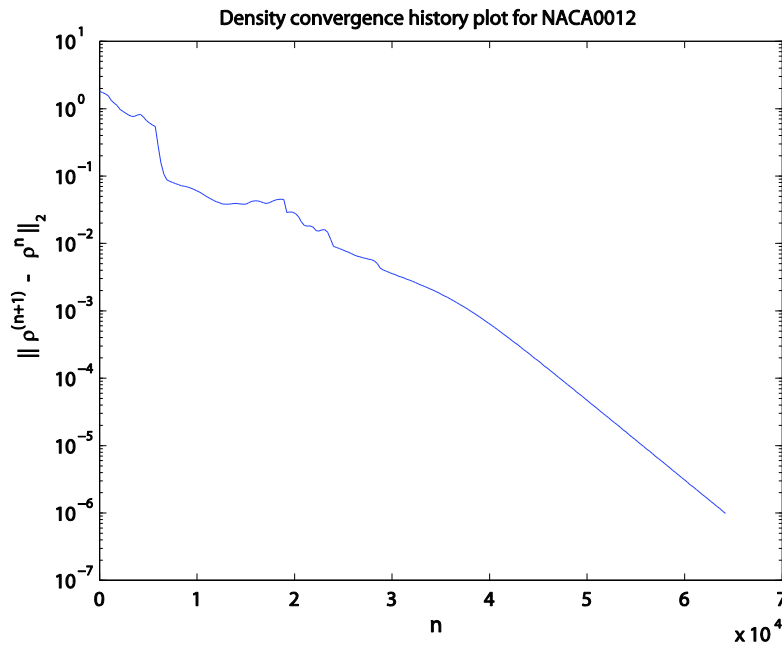
**Figure 25:** Comparison of the skin friction coefficient obtained from the New Simplified method and the Simplified method.

The difference between the skin friction coefficient obtained from the New Simplified method and the Simplified method are shown in figure 25. In this plot it is evident that the New Simplified

method is closer to the reference body fitted skin friction profile from Sjögren and Petersson (2009) [3]. This gives a good incentive to say that the New Simplified method performs better than the Simplified method.

## 5.5 Supersonic Flow past a NACA0012 Airfoil with Compressible Navier-Stokes Equations

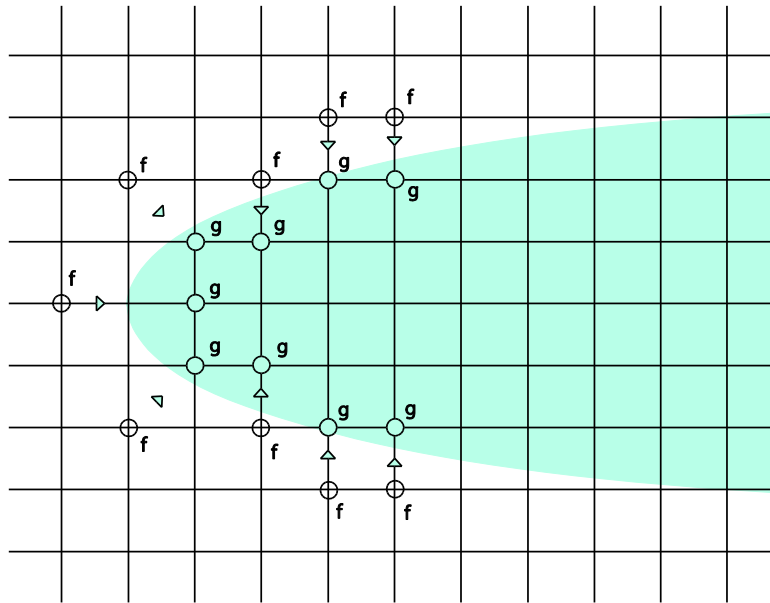
In order to further verify the New Simplified method versus a well documented laminar supersonic flow, the flow past a NACA0012 airfoil has been considered. The following parameters have been used: Free-stream Mach number  $M_\infty = 2$ , angle of attack  $\alpha = 10^\circ$ , and  $Re_\infty = 1000$ . Three grids with  $400 \times 200$ ,  $600 \times 300$  and  $800 \times 400$  cells have been used to discretize the computational domain  $[-0.75, 2.25] \times [-0.75, 0.75]$ . The chord length of the airfoil is  $c = 1$ , whose leading edge is located at the origin. In this simulation the MUSCL scheme is applied to get second order accuracy. The residual measure  $\|\rho^{n+1} - \rho^n\|_2$  is reduced from about 1 to  $10^{-6}$ , as the convergence history for the finest grid shows in figure 26.



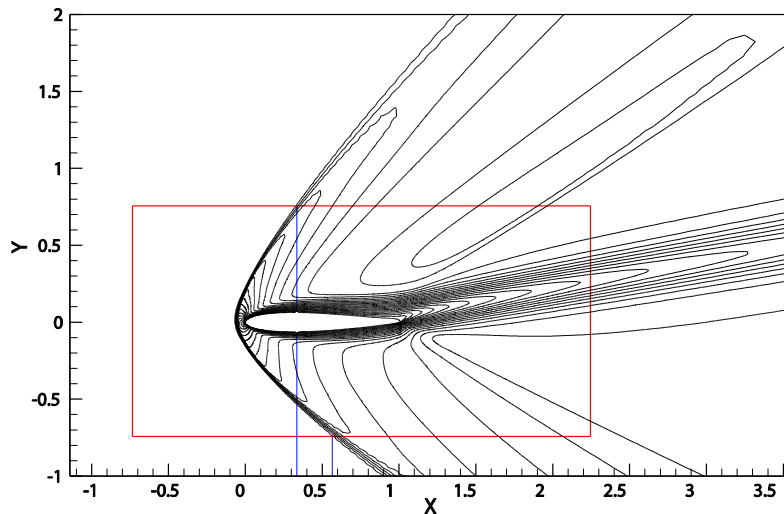
**Figure 26:** Residual convergence history for supersonic flow past a NACA0012 airfoil at a Mach Number of 2, angle of attack  $10^\circ$  and Reynolds number 1000 on a  $801 \times 401$  grid.

Supersonic inflow boundary conditions are used at the inflow  $x = -0.75$ . Supersonic outflow conditions are used at all other boundaries. To achieve an angle of attack the inflow velocities are set as  $u = M_\infty c_\infty \cos(\alpha)$  and  $v = M_\infty c_\infty \sin(\alpha)$ , where  $\alpha$  = the flow angle of attack, and  $c_\infty$  the speed of sound.

On a coarse grid the fluid points near the leading edge are chosen to mirror the ghost points as shown in figure 27. The choice of the fluid points depends on the angle of the normal component with respect to the x-axis as shown in figure 8 and (44).

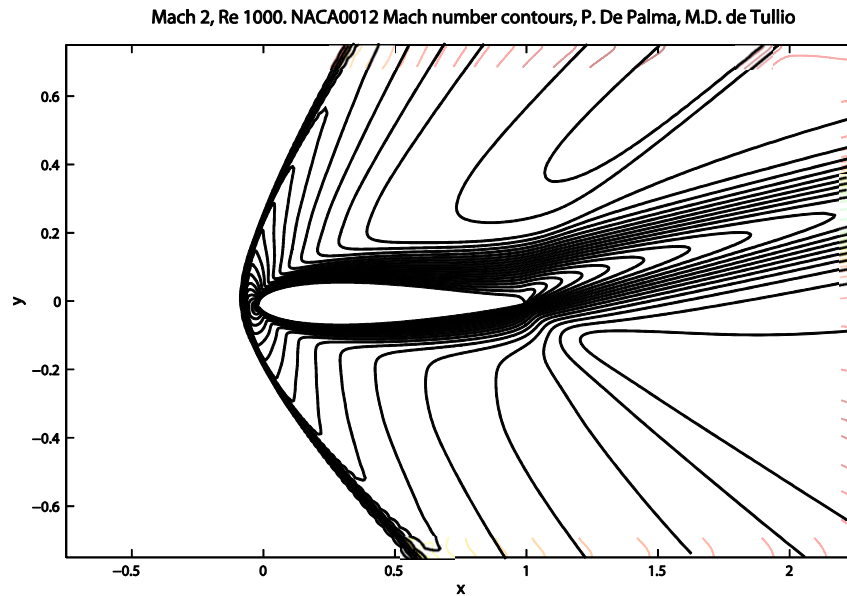


**Figure 27:** Example choice of fluid points near the leading edge of a NACA0012 airfoil.

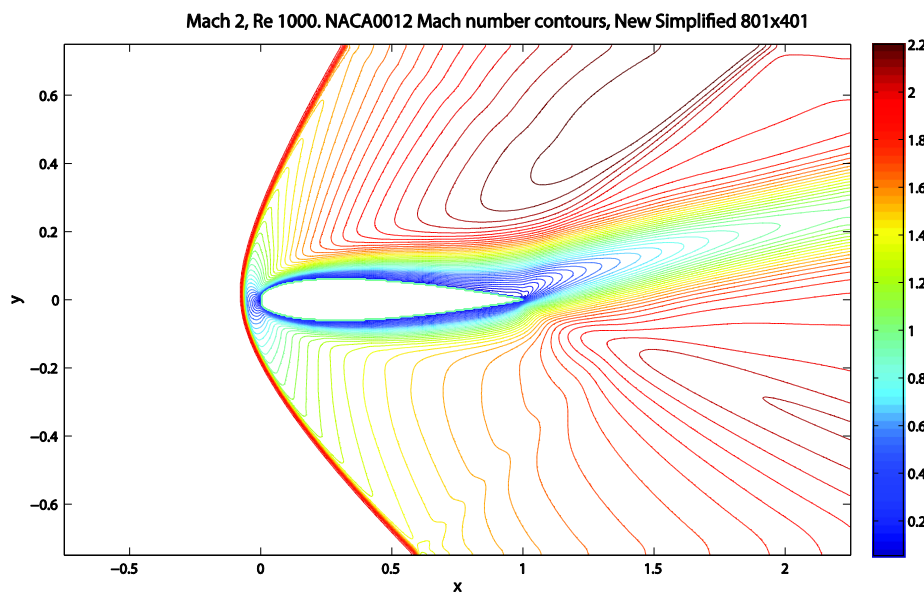


**Figure 28:** Mach number contour results from P. De Palma et al (2006) [14].  $M_\infty = 2$ ,  $Re_\infty = 1000$ ,  $\alpha = 10^\circ$ .

To compare the results qualitatively the work of P. De Palma et al. (2006) [14] has been used, and in figure 28 the Mach number contours from their work are shown. The red rectangle drawn in figure 28 represents the domain  $[-0.75, 2.25] \times [-0.75, 0.75]$  which is chosen in this simulation. That domain is then extracted from the plot and enlarged to make it easier to compare the qualitative results, as shown in figure 29.



**Figure 29:** Mach number contour results from P. De Palma et al. (2006)[14] in the domain  $[-0.75, 2.25] \times [-0.75, 0.75]$ .



**Figure 30:** Mach number contours for a NACA0012 airfoil obtained from the New Simplified method.

From the qualitative comparison of figure 29 and figure 30 we see that we get good results. The shock waves and the wake exit the domain at the same locations in both simulations. Since the grid used with the New Simplified method is equidistant the bow-shock wave is more refined at the outer regions. Shock wave reflections are visible where the bow shock exits the domain, but they are small. Because P. De Palma et al. [14] used a Cartesian grid with local grid refinement, their resolution around the airfoil is good, while they have less points at the outer regions.

The lift and drag coefficients calculated are compared to the results of [14], [15] and [16] in table 1. P. De Palma et al. calculated their lift and drag coefficients performing a momentum balance of the fluid comprised within a rectangle surrounding the body [14]. Also note that the viscosity is taken as a constant in the results from the GAMM-workshop [16] and from B. Müller [15].



The lift and drag coefficients in the present work is calculated using (16),(19) and (20). The equation for the force acting on the airfoil is found in (16), which when discretized using the midpoint rule on the pressure terms and trapezoidal rule on the shear stresses becomes

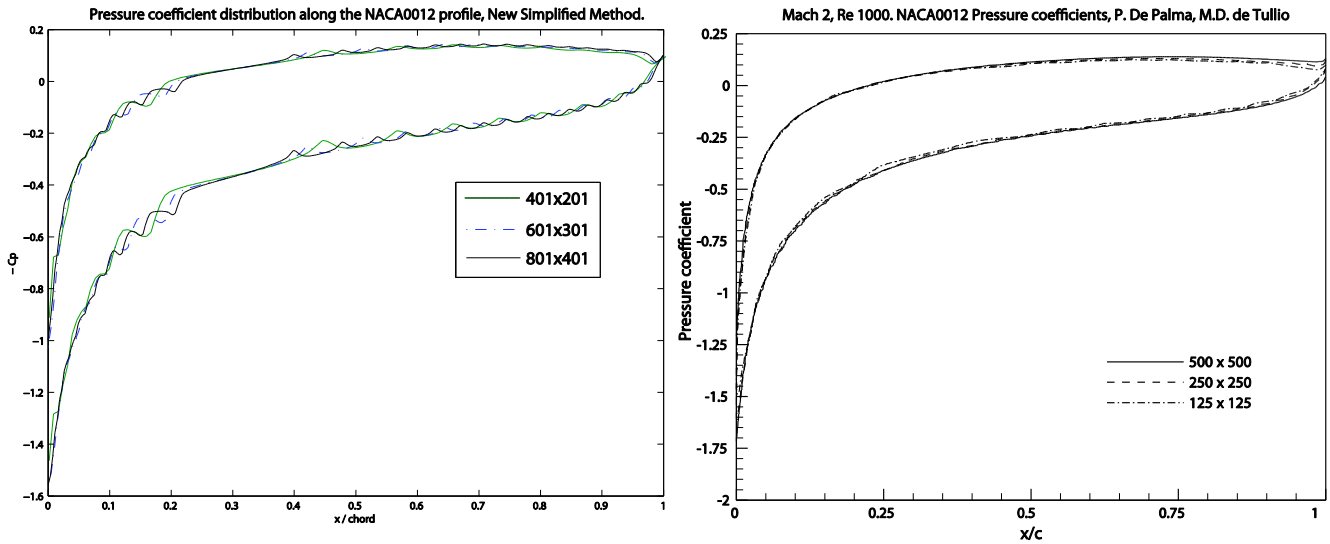
$$\vec{F} = \begin{bmatrix} F_x \\ F_y \end{bmatrix} = - \sum_{i=1}^I P_i \begin{bmatrix} n_x \\ n_y \end{bmatrix} \Delta A_i + \sum_{i=1}^{I-1} \begin{bmatrix} \frac{\tau_{xx_i} + \tau_{xx_{i+1}}}{2} n_x + \frac{\tau_{xy_i} + \tau_{xy_{i+1}}}{2} n_y \\ \frac{\tau_{xy_i} + \tau_{xy_{i+1}}}{2} n_x + \frac{\tau_{yy_i} + \tau_{yy_{i+1}}}{2} n_y \end{bmatrix} \Delta A, \quad (51)$$

where the surface  $\Delta A_i = \left[ (x_{i+1/2} - x_{i-1/2})^2 + (y_{i+1/2} - y_{i-1/2})^2 \right]^{1/2}$ ,  $n_x$  and  $n_y$  are the x- and y-components of the unit normal vector and are calculated directly from the airfoil profile using (16).

**Table 1:** Drag and lift coefficients for supersonic flow around a NACA0012 airfoil at a Mach number of 2, Reynolds number 1000 and with 10 degree angle of attack.

Test Case	Mesh	C <sub>D</sub>	C <sub>L</sub>
<b>B. Müller [15]</b>	49x17 (C-grid)	0.2379	0.3253
	97x33 (C-grid)	0.2484	0.3388
	193x65 (C-grid)	0.2515	0.3388
<b>P. De Palma et al [14]</b>	125x125 (Cartesian with local grid refinement)	0.2448	0.3296
	250x250 (Cartesian with local grid refinement)	0.2485	0.3335
	500x500 (Cartesian with local grid refinement)	0.2514	0.3353
<b>New Simplified method</b>	401x201 (Equidistant Cartesian grid)	0.2354	0.3335
	604x301 (Equidistant Cartesian grid)	0.2525	0.3400
	801x401 (Equidistant Cartesian grid)	0.2597	0.3421
<b>GAMM-Workshop [16]</b>	193x72 Group 4 (L. Cambier)	0.2535	0.3427

The calculated drag and lift coefficients obtained from the New Simplified method are close to the other results in table 1. We have the highest drag coefficient, but observe grid convergence.

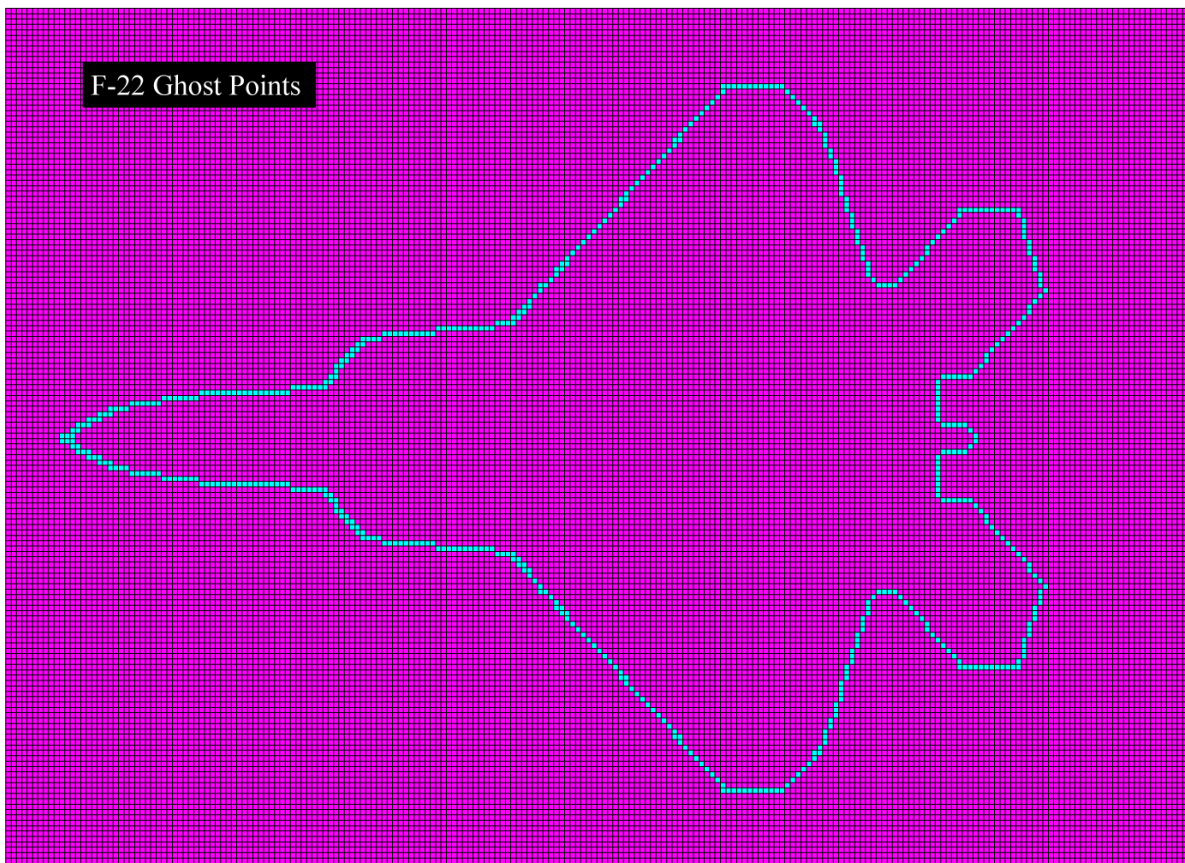


**Figure 31:** Pressure coefficient obtained from the New Simplified method (left) and from P. De Palma et al. (2006), [14] (right).

The pressure coefficient along the upper and lower part of the airfoil is plotted in figure 31. The New Simplified method suffers from wavelike bumps. Although they get smaller with grid refinement, they are still visible. Other than that, the shape and values of the pressure coefficient are fairly similar to the results of P. De Palma (2006). From the reasonable drag and lift coefficients given in table 1 one can say that the wavelike bumps even each other out in total.

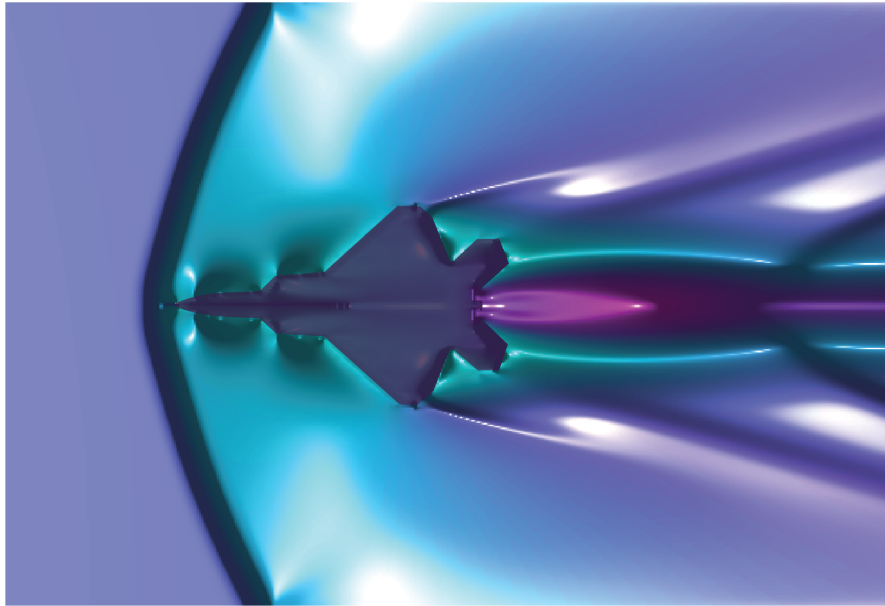
## 5.6 Supersonic Flow around a 2D F-22 Fighter Aircraft

To underline the flexibility of the New Simplified method the supersonic flow around an even more complex geometry has been considered. The 2D profile of the F-22 Jet Fighter has been recreated. The profile has been found on the websites of the United States Air Force (USAF) [20]. The F-22 was chosen because of its similarity to the F-35 JSF aircraft that Norway has ordered. The jet engines have their own boundary conditions to simulate flow from the engines. There is little scientific value in this simulation, other than to show that there are few limits to the geometry of the embedded boundary with the use of the New Simplified or the Simplified method. A grid with  $500 \times 400$  cells has been used to discretize the computational domain  $[0, 50] \times [-20, 20]$ . The aircraft is 18.9 meters long [20]. To get a big jet from the engines the inviscid case with the compressible Euler equations has been used with a free stream Mach number  $M_\infty = 2$ . The outflow at the engine outlets has been set with a Mach number of 3,  $\rho_\infty$  and  $p_\infty$ . The ghost point values near the engine outlets were set equal to the engine outlet conditions. First order accuracy is reached with the first order local Lax-Friedrichs method.



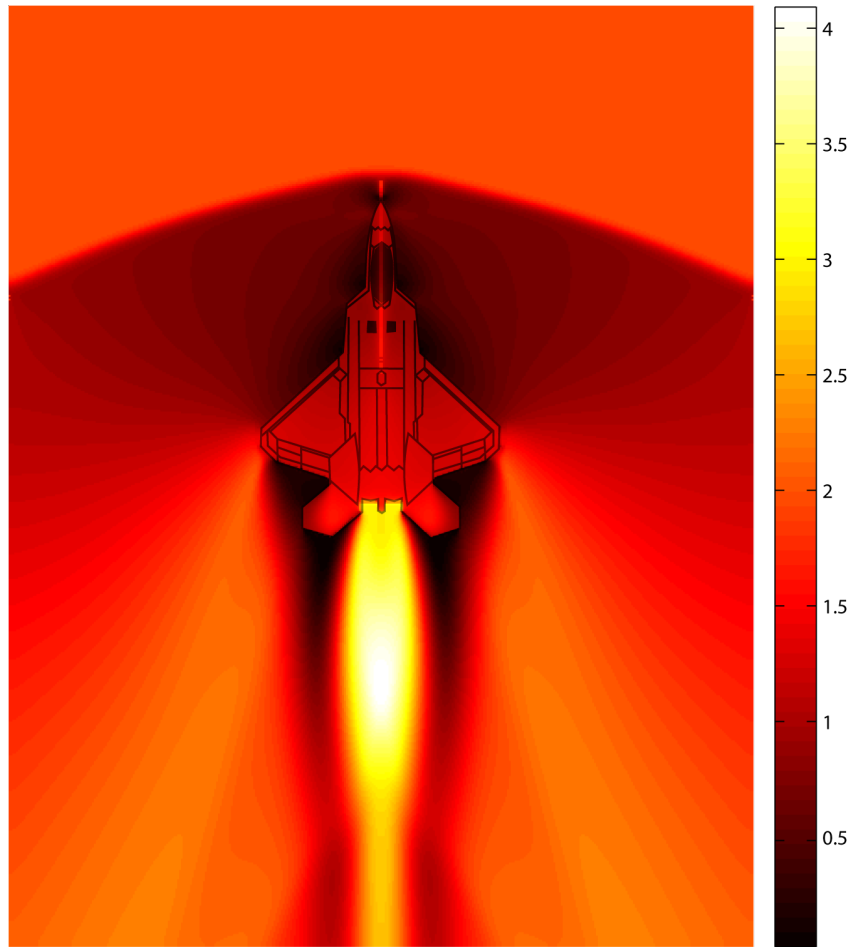
**Figure 32:** Flagged ghost cells for the F-22 geometry on a  $221 \times 161$  grid.

Colorful Fluid Dynamics: Due to the lack of any verification to these simulations we have made an attempt to create more artistic plots for the reader's delight.



**Figure 33:** Mach number image with cool color mapping and lightning effects for the 2D F-22 fighter aircraft.

Some interesting flow patterns are seen in the wake due to the shape of the wings. The flow gets room to expand after the outflow from the engines, so it accelerates downstream. We expect a sharper attached shock than seen in figure 33 and 34, so the ghost point treatment at the nose might need some detailed work and further testing.



**Figure 33:** Mach number image with hot color mapping of the 2D F-22 fighter aircraft in vertical accent.

The outline and details of the F-22 fighter in figure 33 are added in post processing.

## 5.7 Results and Discussion

The New Simplified method has been tested for a variety of cases. The flow around a diamond-wedge airfoil behaves the right way, although details are not checked. The viscous supersonic flow around a NACA0012 airfoil gives good results, and the method can be said to be verified for this case, although we observe wave-like bumps in the computed pressure coefficient profile. The unsteady inviscid supersonic flow around a cylinder in a channel with a moving shock gives good transient results, and the method can be said to be verified for this case. The supersonic viscous flow around a cylinder gives good qualitative results, although the computed skin friction profile differs from the reference profile [3]. Good convergence to steady state is reached in all cases.

The existence of bumps and unwanted discontinuities in pressure coefficients and skin friction coefficients seems to be a recurring problem with simplified ghost point treatment. In comparison with the Simplified method, these problems are less evident in the New Simplified method, though that was a goal with developing a new method. For that we have shown that the bump in the pressure coefficient profile for inviscid flow around a cylinder in the region where the mirror point switches

from the  $x$ - to the  $y$ -grid lines for the Simplified method has disappeared for the New Simplified method (figure 17).

In computational fluid dynamics it is difficult to get something for free. If a simple method is applied, the accuracy will most likely be affected. Although the developed method shows some good results, we have not quite been able to reach the results of other more advance methods. On another note, a simple method is easy to work with, and can be flexible. The flexibility is shown with the F-22 embedded boundary.

## **6 Conclusions and Outlook**

We have presented an embedded boundary method for computing viscous and inviscid compressible flows around complex geometries. The method can give first or second order accuracy and is easy to implement for two-dimensional problems. The method has been verified versus well documented steady and unsteady test problems in supersonic flow regimes using inviscid flow or viscous flow with moderate values of the Reynolds number. We find that more grid points must be used to achieve high resolution near the body and in regions of high flow-gradients, in comparison with methods that use local grid refinement or have gridless points near boundaries. Several numerical tests have been performed to demonstrate the accuracy, robustness and versatility of the proposed method. The numerical results obtained indicate that the use of the New Simplified method leads to an improvement over the Simplified method. The results from the New Simplified method are comparable, but not as accurate as other more complex methods.

For future work it would be interesting to further develop the method to compute three-dimensional problems. The order of accuracy could be increased and it is natural to look at the implementation of local grid refinement. If we get higher resolution in areas of interest we could solve for smaller scales, and could increase the Reynolds number. A more pressing matter is to further develop and test the calculation of the skin friction coefficient, as it has proven to be very sensitive to any change in the methods used to calculate it. Lastly, the weighed ghost point method can possibly be improved and tested for other cases.

## **Acknowledgments**

This work was performed under the excellent supervision of professor Bernhard Müller at NTNU, and with good help from Ph.D. Muhammad Asif Farooq.

## References

- 
- | References |  |
|------------|--|
| [1]        | T. R. Bewley. " <i>Numerical Renaissance</i> ". Not published yet, but available online at <a href="http://renaissance.ucsd.edu/">http://renaissance.ucsd.edu/</a> (2010)  |
| [2]        | M. Farooq. " <i>Cartesian Grid Method for Compressible Flow Simulation</i> ". Doctoral thesis at NTNU: 85 (2012)   |
| [3]        | B. Sjögren, M. Kupiainen. " <i>A Cartesian Embedded Boundary Method for the Compressible Navier-Stokes Equations</i> ". J. Sci. Comput. 41, pp. 94-117 (2009)  |
| [4]        | S. Gottlieb, C.-W. Shu and E. Tadmor. " <i>Strong Stability-Preserving High-Order Time Discretization Methods</i> ". SIAM Review Vol.43, No. 1, pp. 89-112   |
| [5]        | H.-O. Kreiss and N.A. Petersson. " <i>A Second Order Accurate Embedded Boundary Method for the Wave Equation with Dirichlet Data</i> ". SIAM J. Sci. Comput. 27, pp. 1141-1167 (2006)  |
| [6]        | H.-O. Kreiss, N.A. Petersson and J. Yström. " <i>Difference Approximations of the Neumann Problem for the Second Order Wave Equation</i> ". SIAM J. Numer. Anal. 42, pp. 1292-1323 (2004)  |
| [7]        | S. Gottlieb, C.-W. Shu. " <i>Total variation Diminishing Runge-Kutta Schemes</i> ". Mathematics of Computation Vol. 67, No. 221, pp. 73-85 (1998)  |
| [8]        | B. Sjögren and N.A. Petersson. " <i>A Cartesian Embedded Boundary Method for Hyperbolic Conservation Laws.</i> " Commun. Comput. Phys., pp. 1199-1219 (2007)   |
| [9]        | M. Farooq and B. Müller, " <i>Accuracy of the Cartesian Grid Method for Hyperbolic Conservation Laws Using Standard and Simplified Ghost Point Treatments at the Immersed Boundary</i> ". Journal of Structural Mechanics, Vol. 44, No. 3, pp.279-291 (2011) |
| [10]       | B. van Leer. " <i>Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method</i> ". Journal of Computational Physics 32, pp. 101-136 (1979)   |
| [11]       | R. M. Beam and R.F. Warming. " <i>An Implicit Factored Scheme for the Compressible Navier-Stokes Equations</i> ". AIAA Journal vol.16, no.4, pp. 393-402 (1978)  |
| [12]       | J.J. Quirk. " <i>An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies</i> ". Computers and Fluids 23, pp. 125-142 (1994)   |
| [13]       | H. Luo, J.D. Baum and R. Löhner. " <i>A hybrid Cartesian grid and gridless method for compressible flows</i> ". Journal of Computational Physics 214, pp. 618-632 (2006)   |
| [14]       | P. De Palma, M.D. Tullio, G. Pascazio and M. Napolitano. " <i>An immersed-boundary method for compressible viscous flows</i> ". Computers and Fluids 35, pp. 693-702 (2006)  |
| [15]       | B. Müller. " <i>Navier-Stokes solution for laminar transonic flow over a NACA0012 airfoil</i> ". FFA Report 140 The Aeronautical Research Institute of Sweden, Stockholm (1986)  |
| [16]       | M.O. Bristeau, R. Glowinski, J. Periaux and H. Viviand. " <i>Numerical Simulation of Compressible</i>  |
-

---

*Navier-Stokes Flows*". Notes on Numerical Fluid Mechanics Vol 18, Vieweg, Braunschweig (1987)

[17] Ames Research Staff. *"Equations, Tables and Charts for Compressible Flow"*. NACA Report 1135, (1953)

[18] J.D Anderson JR. *"Fundamentals of Aerodynamics"*. Tata McGraw-Hill Publishing Company Limited, New Delhi, Fourth Edition (2007)

[19] J. Zierep. "Theoretische Gasdynamik" G. Braun, Karlsruhe (1976)

[20] <http://www.af.mil/shared/media/ggallery/hires/AFG-060411-002.jpg>

[21] R. Crockett, P. Colella and T. Marthaler. *"A Cartesian grid projection method for the incompressible Euler equations in complex geometries"*. SIAM J. Sci. Comput. 18, pp 1289-1309 (1997)

[22] A. A. Skøien. *"Cartesian Grid Method for Viscous Compressible Flow"*. Project thesis at NTNU (2011)

---



# Appendix

## A - Code Development

### A.1 Memory Handling

On newer computers the bottleneck of performance usually lies in the memory. The CPUs are fast, but the memory handling can be slow if not handled properly. Matlab is not a clever choice if a fast code is a goal. This is because of the way Matlab handles its variables. Every time the CPU asks for a variable, Matlab goes all the way to the RAM memory to get it, and skips the much faster cache memories (modern computers have two cache memories, some have three). This is almost as going to the brick factory for each brick you use, when building a brick house. What you should have is a large pile of bricks next to you, and when that is gone, you go to the brick store, and when the brick store is out of bricks, the brick store goes to the factory. In a compiled language such as C or Fortran, the next variables would already be lined up in the next cache – analog to a pile of bricks next to you.

However, the built in functions in Matlab are faster because these have been optimized, they should send the correct set of variables to the next cache memory. So as often as possible we use built-in Matlab functions.

### A.2 Ghost Point Treatment

When working with the body fitted method, adding a new geometry to the grid at a late point of the development is not possible. One would have to start from the beginning and fit the new grid to the new geometry. This work can be tedious.

With the embedded method the geometries are simply placed on top of the grid, so adding or removing an even complex geometry can be done at any point of the development. The way the ghost point indices are stored in the code makes it very easy to add multiple geometries in the same domain. When the ghost points for a solid body are created, they are in reference to a point on the solid. For example, all the ghost point indices to the circular cylinder are stored as pointers in reference to the center of the cylinder. When the cylinder is applied to the domain, the function that creates these points is called and the center location of the cylinder is set as a parameter. If another cylinder is wanted, the function is simply called again, with a different center location. The function that treats the ghost points (mirror them with appropriate fluid points) use these pointers to treat all the cylinders in the domain.

Example of CPU time use: With 800x400 cells and 50 000 time steps using the second order MUSCL scheme, the code ran on a 3.2GHz CPU for 53.6 hours. With 800x400 cells and 50 000 time steps using the first order approach, the code ran on a 3.2GHz CPU for 34.8 hours.

The program is developed from scratch and consists of 37 functions with a total of 7300 lines. The reason for the number of functions are the way the immersed boundaries are treated. For example the cylinder has one function to generate ghost point, one function to treat ghost points for viscous flow, one function to treat ghost points for inviscid flow, one function to calculate skin and pressure coefficients and one function to post process the flow such that we can observe a solid circle. All immersed geometries share these sets of functions. A schematic of how the code works is presented below.

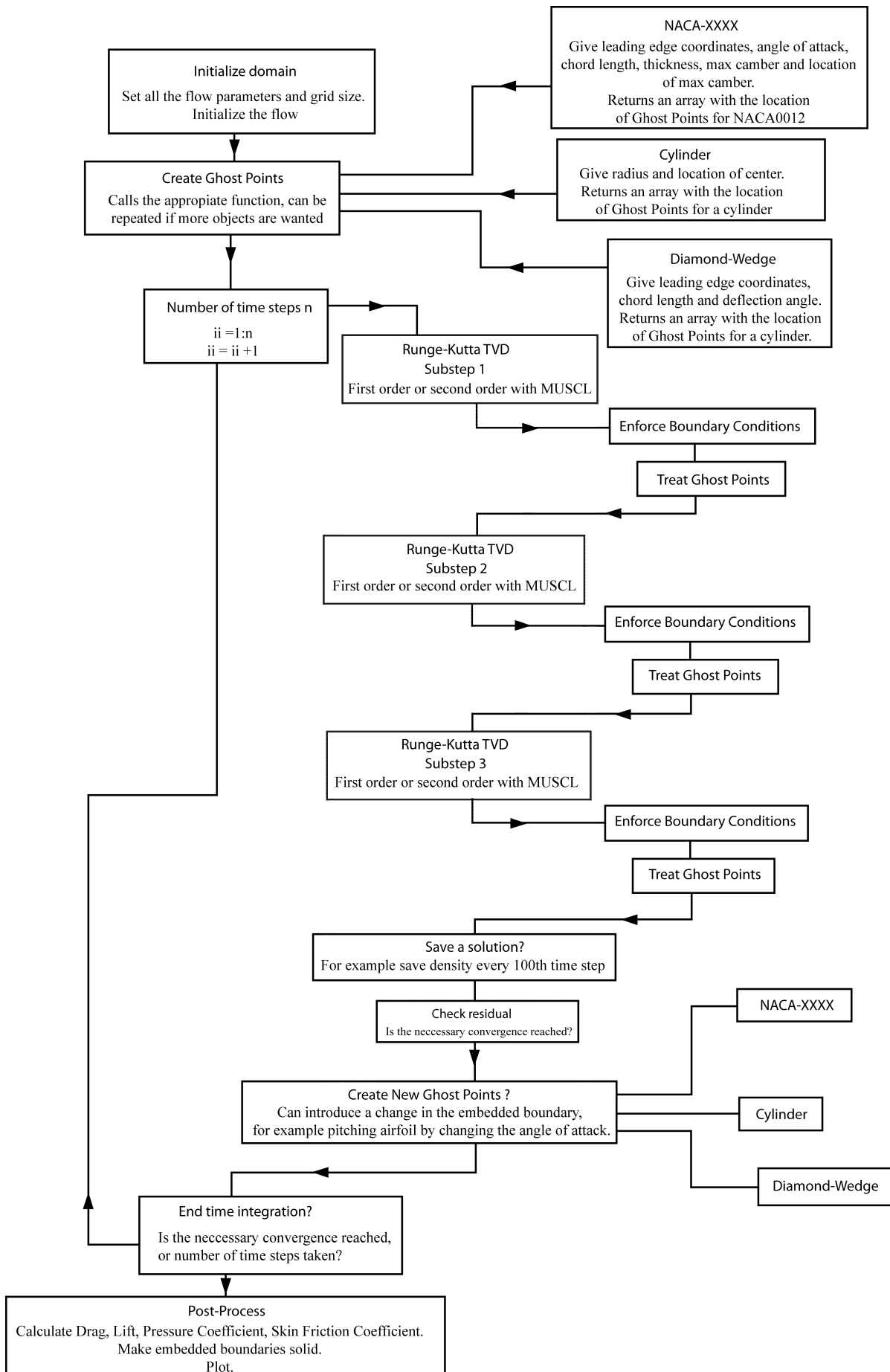


Figure: Flowchart of present Cartesian Grid method

An example of the ghost point treatment functions is given below.

```
[ghostPointIndicesCylinder] = createGhostPointsCylinder(RadiusOfCylinder,  
xLocationOfCenter, yLocationOfCenter).
```

The function "createGhostPointsCylinder" creates the ghost point indices of a given cylinder, and returns the indices to the memory of the computer. The function that mirrors these points with the appropriate fluid points will use the ghost point indices (now available in the memory) after each RK3 sub step, and update the solution matrix U.

```
[U] = treatGhostPointsCylinder(ghostPointIndicesCylinder).
```

If a second cylinder is needed a new set of ghost points will be stored in the memory, and the function "createGhostPointsCylinder" is called with parameters of a new cylinder, such as

```
[ghostPointIndicesCylinder_2] = createGhostPointsCylinder(RadiusOfCylinder_2,  
xLocationOfCenter_2, yLocationOfCenter_2).
```

The function that treats these new points are called again after each RK3 sub step, but with the new set of ghost point indices, such as

```
[U] = treatGhostPointsCylinder(ghostPointIndicesCylinder),  
[U] = treatGhostPointsCylinder(ghostPointIndicesCylinder_2),
```

and now there are two cylinders in the domain.

If a pitching airfoil is needed, the ghost point indices have to be changed during the time steps. The function that creates the ghost point indices to the airfoil is then called during the time integration, but with a different angle of attack parameter.

```
angleOfAttack = angleOfAttack + changeInAngleOfAttack;
```

```
[ghostPointIndicesAirfoil] =  
createGhostPointsAirfoil(ChordLength, Camber, LocationOfMaxCamber, xLocationOfLeading  
Edge, yLocationOfLeadingEdge, angleOfAttack)
```

The function that treats the ghost points and mirrors them with the appropriate fluid points are called as before after every RK3 substep, such as

```
[U] = treatGhostPointsAirfoil(ghostPointIndicesAirfoil).
```

The limitations to handling the ghost points in this way are almost none. New boundaries can be introduced in unsteady flow, objects can be moved (change x- or y-location), changed (radius of circle increased or decreased for example), taken away (stop treating the ghost point indices) and accelerated (change boundary condition during time steps).