

淡江大學航空太空工程學系 110學年度 專題實驗成果報告

指導教授：牛仰堯 博士

預處理方法應用於一維歐拉方程在噴嘴配置問題

Preconditioning Method Applied to One-Dimensional
Euler Equations for Diverging Nozzle Configuration
Problem

專題學生：李漢偉 撰

學號：407435055

系級：航太四 A

中華民國 110年1月11日

Abstract

The overall purpose of this research is started by deriving a so-called preconditioning matrix based on a density-based solution with the idea of using a time-marching scheme with the employment of an artificial-compressibility approach that will provide a more efficient solution for both constant and variable density flows at all speeds.

The process of derivation will first define a governing equation that controls the behavior of the fluid, then introducing the preconditioning matrix into the system and finding the updated eigenvalues of the preconditioned system. It is followed by deriving the new flux values which uses the flux difference method.

The preconditioning matrix is then applied to a diverging nozzle configuration problem for both supersonic and subsonic outflow from a supersonic inflow, paired up with a Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL) finite volume scheme as a high reconstruction scheme with Roe's finite difference upwind Riemann solver to split wave speeds into two different components consisting of positive and negative values. A one-dimensional Euler equation will be set as the governing equation of this problem. The computer code will be briefly discussed to show how it works and produce the plots for each fluid components.

The results of this research will then be compared with the solution without adding the preconditioning matrix to give an insight of how the preconditioning process enhances the overall convergence rates of density-based, time-marching schemes, which will then provide steady-state and time-accurate solutions for both variable and constant density fluids at all speeds over a wide range of Reynold numbers in the diverging nozzle configuration problem, producing a higher accuracy and performance in simulating a fluid flow problem concerning from supersonic, compressible into subsonic and incompressible flow.

Such preconditioning method poses a significant purpose for solving multi-phase flows that have a complex component of different chemical reactions and heat releases of a real gases, as well as a cavitating flow that is proved difficult in simulating such flows due to the emerging cavitating bubbles that interrupt the overall fluid behavior by noises and vibration that damages the medium of the flow.

It is thus expected that the preconditioned system's flow solver will be regarded as a universal single numerical solution method that can be applied in any given fluid in various frameworks and situations in hope of improving the simulation of the fluid by removing unwanted oscillations and discontinuities.

Table of Contents

Abstract.....	ii
Table of Contents	iii
Introduction.....	v
List of Figures.....	vi
List of Tables.....	vii
List of Symbols and Abbreviations.....	viii
Chapter 1: Motivation and Research Problem.....	1
1.1 Motivation	1
1.2 Research Problem.....	1
Chapter 2: Background.....	2
2.1 Supersonic, Compressible Flow	2
2.2 Subsonic, Incompressible Flow.....	3
Chapter 3: Literature Review.....	4
3.1 Roe’s Approximate Riemann Solver.....	4
3.2 Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL).....	4
3.3 Weiss-Smith Preconditioning	4
Chapter 4: Research Methodology and Procedure.....	5
4.1 Derivation of Eigenvalues for Preconditioned System.....	5
4.1.1 Governing Equations.....	5
4.1.2 Preconditioning Matrix.....	6
4.2 Flux Difference Splitting for Preconditioned System.....	9
4.3 Solution for a One-Dimensional Fluid Flow Problem.....	11
4.3.1 Steps in Solving a One-Dimensional Fluid Flow Problem.....	11

Chapter 5: Simulation Results.....	12
5.1 Case 1: Supersonic Inflow Supersonic Outflow.....	12
5.2 Case 2: Supersonic Inflow Subsonic Outflow.....	16
Chapter 6: Conclusions	19
Chapter 7: Future Research and Recommendations.....	20
Appendix A: MUSCL type Roe scheme for divergent nozzle configuration problem.....	21
Appendix B: Roe Flux Difference Splitting for divergent nozzle configuration problem. .	23
Appendix C: Computer codes for divergent nozzle configuration problem.....	25
References.....	35

Introduction

In the field of fluid mechanics, the importance of computational fluid dynamics (CFD) should not be neglected as one of the engineering tools for simulating and predicting real life action of a thermo-fluids in a system due to the fact that not only it can help reduce the costs of a conventional experiments, it also provides a high resolution in time and space solution in a short amount of time without disregarding the real physical meaning behind the problem, thus imposing a great significance in the engineering industry, making it a crucial skill and knowledge for engineers to possess to solve modern problems relating to moving fluids.

For decades, researchers have been searching and inventing better and more efficient numerical solution methods in a sense to obtain the closest approximate outcome comparing with the exact solution for various types of fluid problems, as different scenarios require separate methods since each flows have their own behavior, how they interact with the boundary condition, both internally and externally. It has been concluded that a single flow solver for any given fluid problems is one of the main keys in reaching their objectives, remaining a challenge in the industry.

In this paper, the preconditioning idea will be proposed and briefly discussed the mathematical meaning and how it relates to the physical world by including related derivations aligning with how it will act in a real situation. Several assumptions will be presented in the process of the derivation with a logical reason supporting it.

The selected fluid problem will be determined by choosing a medium that interact with both subsonic and supersonic flow, ranging from incompressible to compressible flow. A great example that can implement those criteria are a diverging nozzle configuration which is applied in both aircraft and spacecraft engines, making it a convenient topic for mostly in the field of aerospace engineering.

Several widely known and used numerical solution methods such as Monotonic Upstream-centred Scheme for Conservation Laws (MUSCL) and Roe's finite difference Riemann solver will be implemented in the said problem, and the simulation results will be plotted in a two-dimensional plots and compared with solutions performed with the preconditioning method.

Finally, a conclusion will be made from the comparison done and present how the act of preconditioning affects the overall solution it provides, also what does this bring for the development progress of fluid mechanics in the future that affects the overall engineering field.

List of Figures

Figure	Page
1.2 Diverging nozzle configuration.....	1
5 Physical domain for a diverging nozzle configuration	12
5.1.1 Comparison of Mach number distributions for supersonic outflow	13
5.1.2 Comparison of pressure distributions for supersonic outflow	14
5.1.3 Comparison of density distributions for supersonic outflow	14
5.1.4 Comparison of temperature distributions for supersonic outflow	15
5.1.5 Comparison of velocity distributions for supersonic outflow	15
5.2.1 Comparison of Mach number distributions for subsonic outflow	16
5.2.2 Comparison of pressure distributions for subsonic outflow	17
5.2.3 Comparison of density distributions for subsonic outflow	17
5.2.4 Comparison of temperature distributions for subsonic outflow	18
5.2.5 Comparison of velocity distributions for subsonic outflow	18

List of Tables

Table	Page
2 Inflow and outflow boundary conditions.....	3

List of Symbols and Abbreviations

Symbol or Abbreviation	Units	Description
ρ	slugs/ft ³	density of the fluid
ρ_p	slugs/ft ³	density to pressure derivative
ρ_T	slugs/ft ³	density to temperature derivative
u	ft/s	velocity of the fluid
U_R	ft/s	reference velocity of the fluid
c	ft/s	speed of light
T	°R	temperature of the fluid
M	—	Mach number of the fluid
p	lbf/ft ²	pressure of the fluid
S	ft ²	cross-sectional area of medium
e_t	ft ² /s ²	total energy of the fluid
e	ft ² /s ²	internal energy of the fluid
Δx	ft	spatial step
x	ft	position of the fluid
C_p	ft lb _f /slugs °R	specific heat at constant pressure
R	ft lb _f /slugs °R	gas constant
δ	—	very small number
σ	—	very small number
t	s	physical time-step
CFL	—	Courant-Friedrichs and Lewy number
γ	—	specific heat ratio at constant pressure
Q	—	conservative variable matrix
F	—	flux matrix
H	—	wall pressure force matrix
W	—	transformation matrix
Γ	—	preconditioning matrix
Λ	ft/s	eigenvalue matrix
λ	ft/s	eigenvalues of the fluid
A	—	Jacobian matrix
M'	—	modal matrix
Q'	—	primitive variable matrix

Chapter 1: Motivation and Research Problem

1.1 Motivation

Numerical solution methods still lacks the ability to provide the minimum requirement in producing a high accuracy and performance of fluid flows, especially facing with flows having special conditions that could destroy the overall stability of its behaviour. Condition number of the governing system of equations also remains as a significant obstacle, limited to only a certain range that is still considered huge.

In this century, computational fluid dynamics acts as one of the important tools used in dealing with fluid flow simulations by not only engineers that deal with flow problems such as aerospace engineers, but also other concentrations that involves in understanding the behaviour and characteristics of how fluid interacts that also includes chemical reactions and heat releases. Due to its widely usage, struggles quickly risen up for fluid flow cases with low-Mach-number speed, specifically for a variable density flows, making it a tricky situation to get off with. Convergence rates is proven to be limited in such example, making this research a potential solution in dealing with it.

Furthermore, due to most of the numerical solution methods are dealing with explicit, multistage solution algorithms, it is crucial in providing good stability and convergence rates when dealing with compressible flows at supersonic Mach numbers as explicit schemes although provide a low operation count per iteration, it greatly suffers from serious time-step limitations, especially for solving low-Mach-number compressible flows. This results in surfacing problems such as having a higher computational costs per solution, hindering with progresses made by engineers further.

Thus, this research seeks to produce a solution for present numerical methods with problems dealing with simulating flows that encounter error difficulties such as numerical dissipation and oscillations from incompressible low-Mach-number flows to high speed compressible flows by introducing the idea of a preconditioning method that is capable of smoothing the results, specifically in parts that includes extreme situations.

It is expected to provide a higher accuracy and better performance for fluid flow problems comparing to an existing solution algorithm without the implementation of the preconditioning method.

1.2 Research Problem

In this paper, the preconditioning matrix will be developed and applied into the diverging nozzle configuration problem according to figure 1.2, which is governed by the Euler equations, solved alongside with Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL) and Roe's Approximate Riemann Solver to show how it enhances the convergence rates of density-based, time-marching schemes that will provide steady-state solutions and time-accurate solutions for variable and constant density fluids at all speeds over a wide range of Reynold numbers.

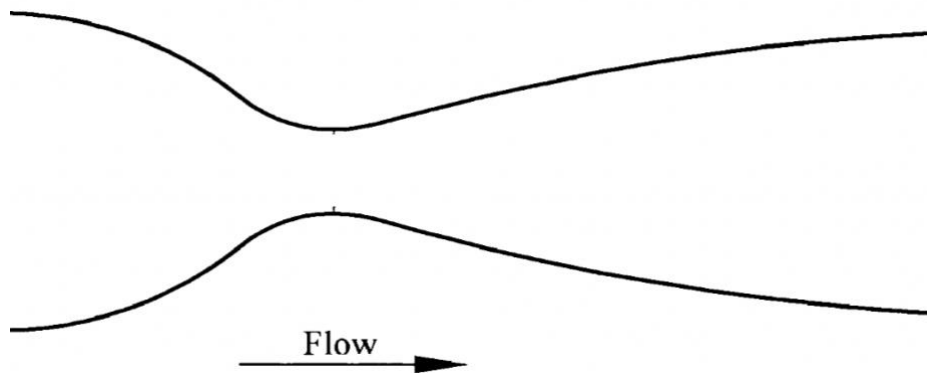


Figure 1.2 Diverging nozzle configuration

Chapter 2 introduces the overlying behavior and characteristics of each different fluid flows and how they interact with various mediums with different requirements for numbers of boundary conditions, including stating the initial conditions necessary for the problem to be solvable by numerical means. Chapter 3 reviews previous researches that are used to numerically solve the diverging nozzle configuration problem and help shape the idea of the derivation process that is discussed detailly in chapter 4. The simulation results are shown in chapter 5, and conclusions will be made according to the comparison of plots produced before and after implementing the preconditioning method into an existing one-dimensional fluid problem in chapter 6. Finally, chapter 7 presents recommendations and future possibilities of what this research will bring and how it will benefit the overall community in contributing a cure for problems faced in most numerical simulation tools.

Chapter 2: Background

2.1 Supersonic, Compressible Flow

A flow is considered supersonic for Mach numbers greater than one, usually ranging from 1.2 to 3. In a diverging nozzles, a flow is conditioned to have a lower pressure to initiate the phenomenon of a supersonic speed.

Due to the local flow velocity being greater than the speed of sound, the disturbances created at a point of the flow are not able to work their way upstream, thus inevitably producing the famous shock waves.

In efforts of conserving the mass, the change in velocity and density are considered, in which the behaviour of changing the density of a supersonic flow is much greater than the change in velocity. Additionally, the velocity is directly proportional with the area of the nozzle and density being indirectly proportional.

2.2 Subsonic, Incompressible Flow

Subsequently, a subsonic flow has a Mach numbers lower than one, usually in the range between 0.3 and 0.5. Smooth streamlines can be observed as no such discontinuities are present in the slope, and because the local flow velocity is lesser than the speed of sound, the disturbances are able to propagate to both upstream and downstream of the flow field.

Being incompressible means the density does not varies with any happenings in the flow, thus making the density being fairly constant. This also means that the density neglected when considering the change in the area of the nozzle, while the velocity remains as a variable relating with the area of the nozzle, being indirectly proportional with it.

Due to the weak relationship between the density and pressure of the flow, the analysis and simulation of it is much simpler comparing to that of a supersonic compressible flow comprising shock waves.

	Inflow		Outflow	
	Subsonic	Supersonic	Subsonic	Supersonic
Analytical Boundary Condition	2	3	1	0
Analytical Boundary Condition	1	0	2	3

Table 2: Inflow and outflow boundary conditions

Chapter 3: Literature Review

3.1 Roe's Approximate Riemann Solver

Philip L. Roe (1981) developed an approximate Riemann solver for compressible flow with shocks by finding an estimate for the intercell flux at the interface between two computational cells. It is based on an upwind scheme which is a first-order accurate in space.

Roe showed that in his paper 'Approximate Riemann solvers, parameter vectors, and difference schemes', the Riemann problem are solved by using an exact solution, splitting the linearized Jacobian to seek the system's eigenvalues and eigenvectors. Most Riemann problem are solved using this method as it presents promising results in most cases with the least complexity involved in running the simulation of the fluids.

3.2 Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL)

Bram Van Leer (1979) developed a Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL) which is a finite volume method that is capable of providing highly accurate numerical solutions with a second order of spatial accuracy.

'Towards the Ultimate Conservative Difference Scheme, V. A Second Order Sequel to Godunov's Method' proposed showed that for both one and two-dimensional fluid flow problems, the methods used are able to show promising results derived using Lagrangean schemes that may be regarded as a second order sequel of Godunov's scheme.

3.3 Weiss-Smith Preconditioning

Jonathan M. Weiss and Wayne A. Smith (1955) developed a preconditioning matrix using time-derivative preconditioning in low-Mach-number with artificial compressibility coupled with a dual time-stepping scheme to provide better accuracy performance for variable and constant density flows at all speeds.

In the paper 'Preconditioning Applied to Variable and Constant Density Flows, several test cases ranging from low-speed to unsteady flow were demonstrated to present the effect of the preconditioning matrix on flows where the density variations played a dominant role. This method is also applicable in flows with chemical reactions and heat release for real gases, thus it is expected to be widely used in fluid flow problems that may benefit from it.

Chapter 4: Research Methodology and Procedure

4.1 Derivation of Eigenvalues for Preconditioned System

The process of derivation is initially started by defining the governing equations of the one-dimensional fluid flow problem, where a quasi-one-dimensional Euler equations is chosen, then inserting the preconditioning matrix Γ into the system of equations and rearranging it to construct the new Jacobian matrix used to find the new corresponding eigenvalues and eigenvectors of the preconditioned system. The procedure is further developed by deriving the new formula value for its flux difference splitting using the derived eigenvalues and eigenvectors however with the following exceptions for compressible flow:

1. The derivatives of density with respect to pressure and temperature are not assumed for an ideal gas law equation of state,
2. The characteristic reference velocity term is used for representing pseudo-acoustic speed instead of its local speed of sound and Mach number, and
3. The derivatives of density towards temperature are retained for cases that implied density as a function of temperature only.

4.1.1 Governing Equations

The governing equations set for a quasi-one-dimensional flow is expressed as:

$$S \frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} - H = 0, \quad (1)$$

where Q as the conservative variables and F and H are the flux and wall pressure force vectors given as:

$$Q = \begin{bmatrix} \rho \\ \rho u \\ \rho e_t \end{bmatrix}; \quad F = S \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(\rho e_t + p) \end{bmatrix}; \quad H = \frac{\partial S}{\partial x} \begin{bmatrix} 0 \\ p \\ 0 \end{bmatrix}, \quad (2)$$

and S, ρ, u, e_t and p are the cross-sectional area, density, velocity, total energy per unit mass, and pressure of the fluid, respectively.

The cross-sectional area is assumed to be independent of time and is only a function of the x -axis:

$$S = S(x), \quad (3)$$

and the total energy with the internal energy e is defined by:

$$e_t = e + \frac{1}{2}|u|^2. \quad (4)$$

4.1.2 Preconditioning Matrix

The derivation of the preconditioning matrix is started by replacing the first term of the governing equation $S \frac{\partial Q}{\partial t}$ into $S \frac{\partial Q}{\partial Q'} \frac{\partial Q'}{\partial t}$, where Q is the conserved quantity and Q' is the primitive variable set to make it a preconditioned system. The equation is then transformed as shown below:

$$S \frac{\partial Q}{\partial t} \frac{\partial Q'}{\partial t} + \frac{\partial F}{\partial x} - H = 0, \quad (5)$$

where the primitive variable Q' and the Jacobian $\frac{\partial Q}{\partial Q'}$ is given by the equation below:

$$Q' = \begin{bmatrix} p \\ u \\ T \end{bmatrix}; \quad \frac{\partial Q}{\partial Q'} = \begin{bmatrix} \rho_p & 0 & \rho_T \\ \rho_p u & \rho & \rho_T u \\ \rho_p e_t - 1 & \rho u & \rho_T e_t + \rho C_p \end{bmatrix}, \quad (6)$$

where C_p is the specific heat coefficient at constant pressure, and the derivative of density with respect to pressure ρ_p and temperature ρ_T are given as:

$$\rho_p = \left. \frac{\partial \rho}{\partial p} \right|_T; \quad \rho_T = \left. \frac{\partial \rho}{\partial T} \right|_p, \quad (7)$$

The first term $S \frac{\partial Q}{\partial Q'} \frac{\partial Q'}{\partial t}$ is then multiplied by the transformation matrix W , turning it into a non-conservative system, given by:

$$\left(W \frac{\partial Q}{\partial Q'} \right) S \frac{\partial Q'}{\partial t} + \frac{\partial F}{\partial x} - H = 0, \quad (8)$$

where the transformation matrix W is given by:

$$W = \begin{bmatrix} 1 & 0 & 0 \\ -u & 1 & 0 \\ -(e_t - |u|^2) & -u & 1 \end{bmatrix}. \quad (9)$$

The transformation matrix W is then multiplied with the Jacobian $\frac{\partial Q}{\partial Q'}$, producing the result as follow:

$$W \frac{\partial Q}{\partial Q'} = \begin{bmatrix} \rho_p & 0 & \rho_T \\ 0 & \rho & 0 \\ -1 & 0 & \rho C_p \end{bmatrix}. \quad (10)$$

The preconditioning of the system is done by replacing the matrix $W \frac{\partial Q}{\partial Q'}$ with the preconditioning matrix Γ below:

$$\Gamma = \begin{bmatrix} \theta & 0 & \rho_T \\ 0 & \rho & 0 \\ -1 & 0 & \rho C_p \end{bmatrix}, \quad (11)$$

where Θ is the newly replaced derivative of density with respect to pressure, defined as:

$$\Theta = \left(\frac{1}{U_r^2} - \frac{\rho_T}{\rho C_p} \right). \quad (12)$$

The system of equations is later given as:

$$\Gamma S \frac{\partial Q'}{\partial t} + \frac{\partial F}{\partial x} - H = 0. \quad (13)$$

In Equation 12, U_r is the reference velocity defined with conditions and restrictions as follow:

1. For an ideal gas,

$$U_r = \begin{cases} \sigma c, & |u| < \sigma c \\ |u|, & \sigma c < |u| < c, \\ c, & |u| > c \end{cases} \quad (14)$$

where σ is a small number approximately 10^{-5} that is included to prevent singularities at stagnation points, and c is the local speed of sound.

2. For viscous flows,

$$U_r = \max \left(U_r, \frac{u}{\Delta x} \right) = \frac{1}{2} \left(U_r + \frac{u}{\Delta x} + \left| U_r - \frac{u}{\Delta x} \right| \right), \quad (15)$$

where U_r is limited to not become smaller than the local diffusion velocity $\frac{u}{\Delta x}$. Due to the fact that diffusion effects dominate most of the regions in viscous flows such as boundary and shear layers, a limitation is required to prevent it from happening.

3. For an incompressible fluid and for flows of variable density fluids in the form $\rho = \rho(T)$,

$$U_r = \begin{cases} \sigma U_{\max}, & |u| < \sigma U_{\max} \\ |u|, & |u| > \sigma U_{\max} \end{cases}, \quad (16)$$

where $U_{\max}$ is the maximum velocity allowed in the flow-field.

The unsteady term on the left side system in conservation form is then multiplied by the inverse of the transformation matrix W^{-1} to give the following result equation:

$$(W^{-1}\Gamma)S \frac{\partial Q'}{\partial t} + \frac{\partial F}{\partial x} - H = 0. \quad (17)$$

The matrix A' is the new Jacobian $S \frac{\partial Q}{\partial Q'}$, thus replacing the $W^{-1}\Gamma$ matrix with the matrix A' gives us the system:

$$A' \frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} - H = 0, \quad (18)$$

where the matrix A' is given as:

$$A' = S \frac{\partial Q}{\partial Q'} = S \begin{bmatrix} \Theta & 0 & \rho_T \\ \Theta u & \rho & \rho_T u \\ \Theta e_t - 1 & \rho u & \rho_T e_t + \rho C_p \end{bmatrix}. \quad (19)$$

The system is then multiplied with the inverse of the preconditioning matrix Γ to revert the equation back to its original form:

$$\Gamma^{-1} A' \frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} - H = 0, \quad (20)$$

where the associated eigenvalues are derived from the matrix $\Gamma^{-1} \frac{\partial Q}{\partial Q'}$, shown below:

$$\lambda_1 = u, \quad (21)$$

$$\lambda_2 = u \frac{\rho C_p - U_f^2 (\rho_p \rho C_p + \rho_T)}{2 \rho C_p} + \left(u^2 \left(\frac{\rho C_p - U_f^2 (\rho_p \rho C_p + \rho_T)}{2 \rho C_p} \right)^2 + U_f^2 \right)^{\frac{1}{2}}, \quad (22)$$

$$\lambda_3 = u \frac{\rho C_p - U_f^2 (\rho_p \rho C_p + \rho_T)}{2 \rho C_p} - \left(u^2 \left(\frac{\rho C_p - U_f^2 (\rho_p \rho C_p + \rho_T)}{2 \rho C_p} \right)^2 + U_f^2 \right)^{\frac{1}{2}}. \quad (23)$$

Rearranging the terms, the following results are found:

$$\lambda_1 = u, \quad (24)$$

$$\lambda_2 = u' + c', \quad (25)$$

$$\lambda_3 = u' - c', \quad (26)$$

where the newly introduced variables are defined as:

$$u' = u(1 - \alpha), \quad (27)$$

$$c' = (\alpha^2 u^2 + U_f^2)^{\frac{1}{2}}, \quad (28)$$

$$\alpha = \frac{(1 - \beta U_f^2)}{2}, \quad (29)$$

$$\beta = \rho_p + \frac{\rho_T}{\rho C_p}. \quad (30)$$

4.2 Flux Difference Splitting for Preconditioned System

The flux difference splitting of the new preconditioned system is defined by the following equation:

$$F = \frac{1}{2}(F_R + F_L) - \frac{1}{2}|A'| (Q'_R - Q'_L), \quad (31)$$

where the right face flux F_R and left face flux F_L are fluxes computed using the right primitive variable Q'_R and left primitive variable Q'_L on the left and right side of the face of the cells, with:

$$\Delta Q' = (Q'_R - Q'_L). \quad (32)$$

Multiplying the dissipation term on the right-hand side of the system equation with the preconditioning matrix $\Gamma\Gamma^{-1}$ gives the formula:

$$\Gamma\Gamma^{-1}|A'|\Delta Q' = \Gamma\Gamma^{-1} \left| S \frac{\partial Q}{\partial Q'} \right| \Delta Q'. \quad (33)$$

The absolute value of the Jacobian matrix is assumed to be values with positive and negative sign to simplify grouping the equations.

$$\Gamma\Gamma^{-1}|A'|\Delta Q' = \Gamma\Gamma^{-1}A'\Delta Q'. \quad (34)$$

The matrix A'_Γ is defined as the Jacobian $S \frac{\partial Q}{\partial Q'}$ multiplied with the inverse of preconditioning matrix Γ , shown below:

$$\Gamma \left(\Gamma^{-1} S \frac{\partial Q}{\partial Q'} \right) \Delta Q' = \Gamma |A'_\Gamma| \Delta Q'. \quad (35)$$

The notation A'_Γ is also expressed as:

$$|A'_\Gamma| = M' |\Lambda| M'^{-1}, \quad (36)$$

where Λ is the complete set of the new eigenvalues and M' is the modal matrix that diagonalizes the matrix A'_Γ , which are defined as:

$$\Lambda = \begin{bmatrix} u & 0 & 0 \\ 0 & u' + c' & 0 \\ 0 & 0 & u' - c' \end{bmatrix}, \quad (37)$$

and

$$M' = \begin{bmatrix} 1 & \frac{\rho}{2c'} & -\frac{\rho}{2c'} \\ u & \frac{\rho}{2c'}(u' + c') & -\frac{\rho}{2c'}(u' - c') \\ \frac{1}{2}u^2 & \frac{\rho}{2c'}(e_t + u'c') & -\frac{\rho}{2c'}(e_t - u'c') \end{bmatrix}. \quad (38)$$

The resultant flux splitting can then be expressed in the following form:

$$\Gamma|A'_\Gamma|\Delta Q' = \begin{bmatrix} \Theta & 0 & \rho_T \\ 0 & \rho & 0 \\ -1 & 0 & \rho C_p \end{bmatrix} M' |\Lambda| M'^{-1} \Delta Q' \quad (39)$$

Finally, after rearranging the terms, the resultant flux splitting can be expressed in the following scalar form:

$$\Gamma A'_\Gamma \Delta Q' = |u| \begin{bmatrix} \Delta p \\ \Delta(\rho u) \\ \Delta(\rho e_t) \end{bmatrix} + \delta u \begin{bmatrix} \rho \\ \rho u \\ \rho e_t \end{bmatrix} + \delta p \begin{bmatrix} 0 \\ 1 \\ u \end{bmatrix}, \quad (40)$$

where the newly presented variables are defined as:

$$\delta u = M^* \Delta u + \frac{\Delta p}{\rho U_r^2} [c^* - (1 - 2\alpha)|u| - \alpha u M^*], \quad (41)$$

$$\delta p = M^* \Delta p + \rho \Delta u [c^* - |u| - \alpha u M^*], \quad (42)$$

$$c^* = \frac{|u' + c'| + |u' - c'|}{2}, \quad (43)$$

$$M^* = \frac{|u' + c'| - |u' - c'|}{2c'}. \quad (44)$$

For the nonpreconditioned system, with $\alpha = 0$, $u' = u$, and $c' = U_r = c$, this reduces into the Roe's flux difference splitting when Roe-averaged values are used to reconstruct the right flux F_R and left flux F_L . Thus it is proven that the preconditioning method works logically for any conservative systems, applicable for any cases involving fluid flow problems.

4.3 Solution for a One-Dimensional Fluid Flow Problem

This research shall use the method of Monotonic Upstream-centred Scheme for Conservation Laws (MUSCL) type Roe to calculate the right and left conservative variables, Q_R and Q_L , and substitute it into the formula for calculating the flux using Roe's approximate Riemann solver as $F(Q_R)$ and $F(Q_L)$, then inserting the preconditioning formula derived in the previous subsection to solve a one-dimensional fluid flow problem. It is expected to produce a higher accuracy and better performance of simulation comparing with methods without applying the preconditioning method.

4.3.1 Steps in Solving a One-Dimensional Fluid Flow Problem

The steps involved in computing the algorithm of the one-dimensional fluid flow problem are:

1. Define the governing equation as a quasi-one-dimensional Euler equations,
2. Derive the eigenvalues and eigenvectors of the preconditioned system,
3. Derive the flux difference splitting for the preconditioned system,
4. Examine one-dimensional fluid flow problem that involves different cases of characteristic flows,
5. Using the MUSCL type Roe scheme to solve for the conservative variables Q_R and Q_L ,
6. Insert the calculated conservative variables into Roe's flux difference splitting method to find the flux of the system,
7. Implement the preconditioned system of equation with its corresponding variables into Roe's flux,
8. Applied the formula steps above with the necessary initial and boundary conditions of the one-dimensional fluid flow problem into a computer code,
9. Plotting a two-dimensional cartesian coordinates for each element of the flow such as the velocity, density, temperature, pressure and Mach number of the flow,
10. Compare the results of the preconditioned system with plots without adding the preconditioning matrix, and lastly
11. Conclude and state important remarks regarding the result and how it can benefit in solving one-dimensional fluid flow problem for future researches.

Chapter 5: Simulation Results

Consider a nozzle whose cross-sectional area is defined by

$$S(x) = 1.398 + 0.347 \tanh(0.8x - 4)$$

Locate the nozzle entrance at $x = 0.0$ and the nozzle exit at $x = 10.0$ ft. Assume air with $\gamma = 1.4$ and $R = 1716 \text{ ft lb}_f/\text{slugs}^\circ\text{R}$ enters the nozzle at supersonic speed. The flow leaves the nozzle under two difference conditions specified as

- (1) supersonic and
- (2) subsonic.

Determine the steady-state solution of the quasi one-dimensional model equation within the domain specified in this problem. The physical domain is illustrated below.

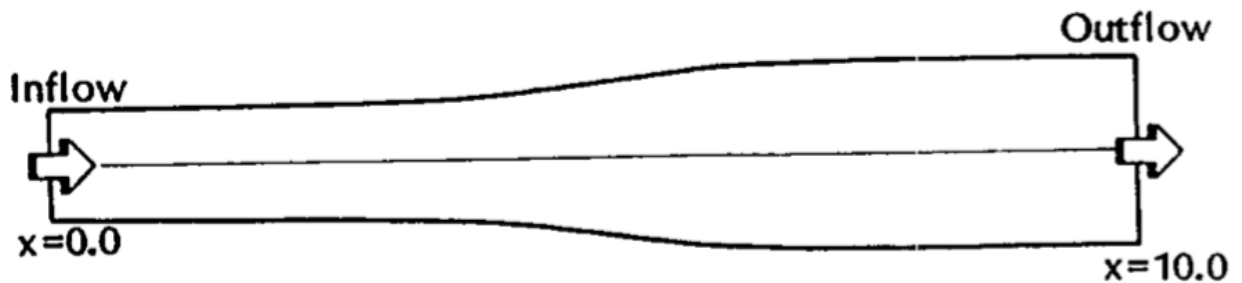


Figure 5: Physical domain for a diverging nozzle configuration

5.1 Case 1: Supersonic Inflow Supersonic Outflow

The supersonic flow properties at the inlet may be determined from the given data to be as:

$$\begin{aligned}u &= 1676.55 \text{ ft/s} \\e_t &= 3636204 \text{ ft}^2/\text{s}^2 \\ \rho &= 0.002241 \text{ slugs/ft}^3\end{aligned}$$

The flow at the inlet is supersonic and, therefore, the flow properties specified above are used as the inflow boundary conditions. The flow at the outlet is specified as supersonic; therefore, no boundary conditions can be specified.

The values of the dependent variables at the outflow must be computed based on the information reaching the outflow from the interior points, thus, an extrapolation schemes are used for this purpose, where it may be either explicit or implicit. For an explicit approach, the new value at $n+1$ is determined from the value n or n and $n-1$, while for an implicit scheme, the values are determined at the time level $n+1$ as a part of the solution. Few of the extrapolation schemes are reported to be as the following:

$$f_{IM}^{n+1} = f_{IMM1}^{n+1} \quad (45)$$

$$f_{IM}^{n+1} = 2f_{IMM1}^{n+1} - f_{IMM2}^{n+1} \quad (46)$$

$$f_{IM}^{n+1} = f_{IMM1}^n \quad (47)$$

$$f_{IM}^{n+1} = 2f_{IMM1}^n - f_{IMM2}^{n-1} \quad (48)$$

$$f_{IM}^{n+1} = 2f_{IMM1}^n - f_{IMM2}^n \quad (49)$$

The spatial step Δx is set to be 0.1 ft and temporal step as 0.00001 sec. The Courant-Friedrichs and Lewy (CFL) number is about 0.15, which would satisfy the stability requirement of the explicit scheme. The extrapolation formula in Equation 46 is used in this solution.

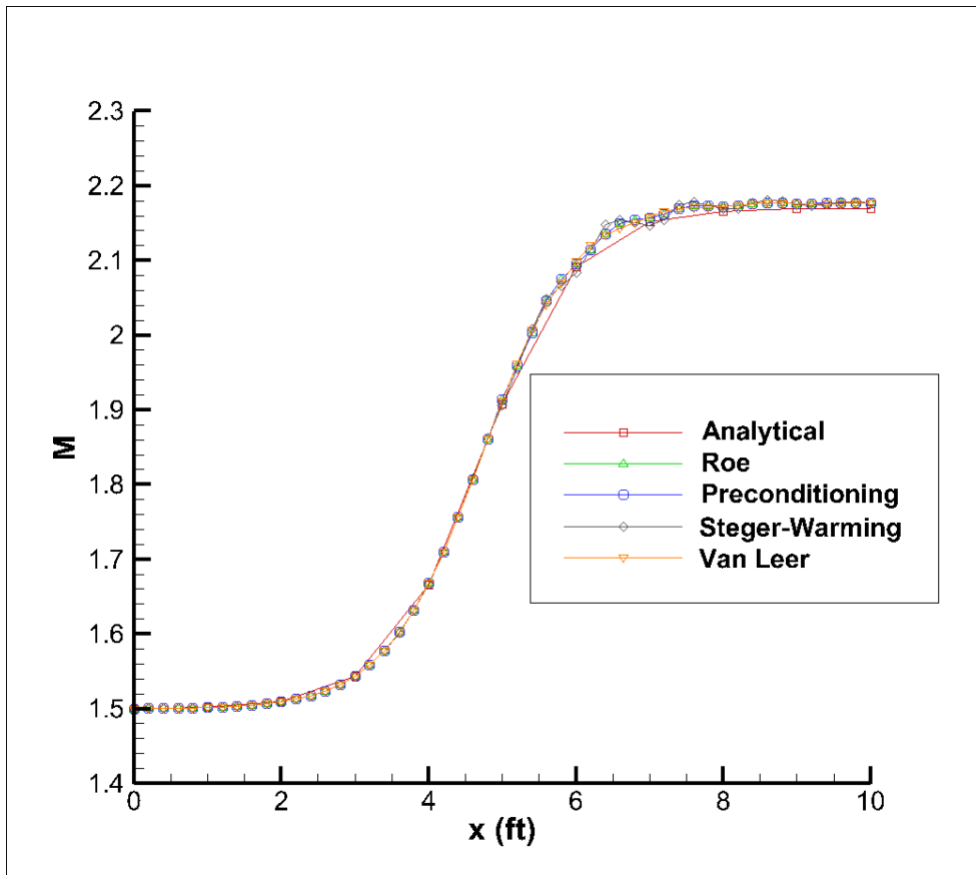


Figure 5.1.1: Comparison of Mach number distributions for supersonic outflow

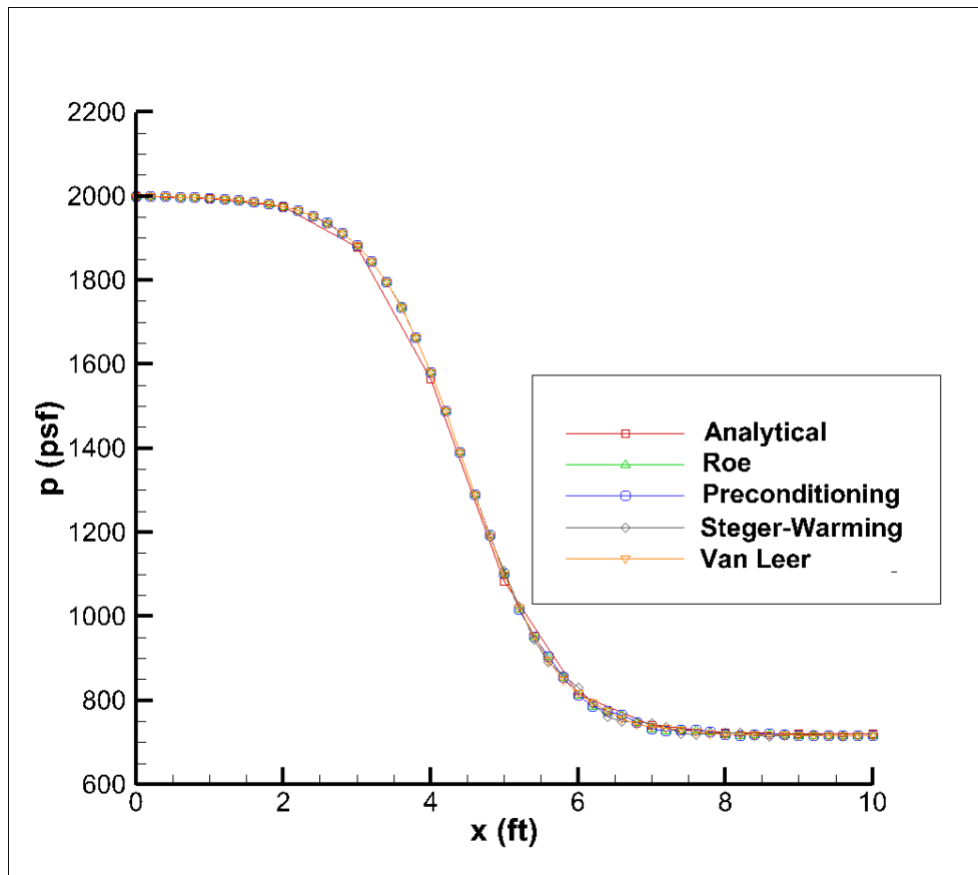


Figure 5.1.2: Comparison of pressure distributions for supersonic outflow

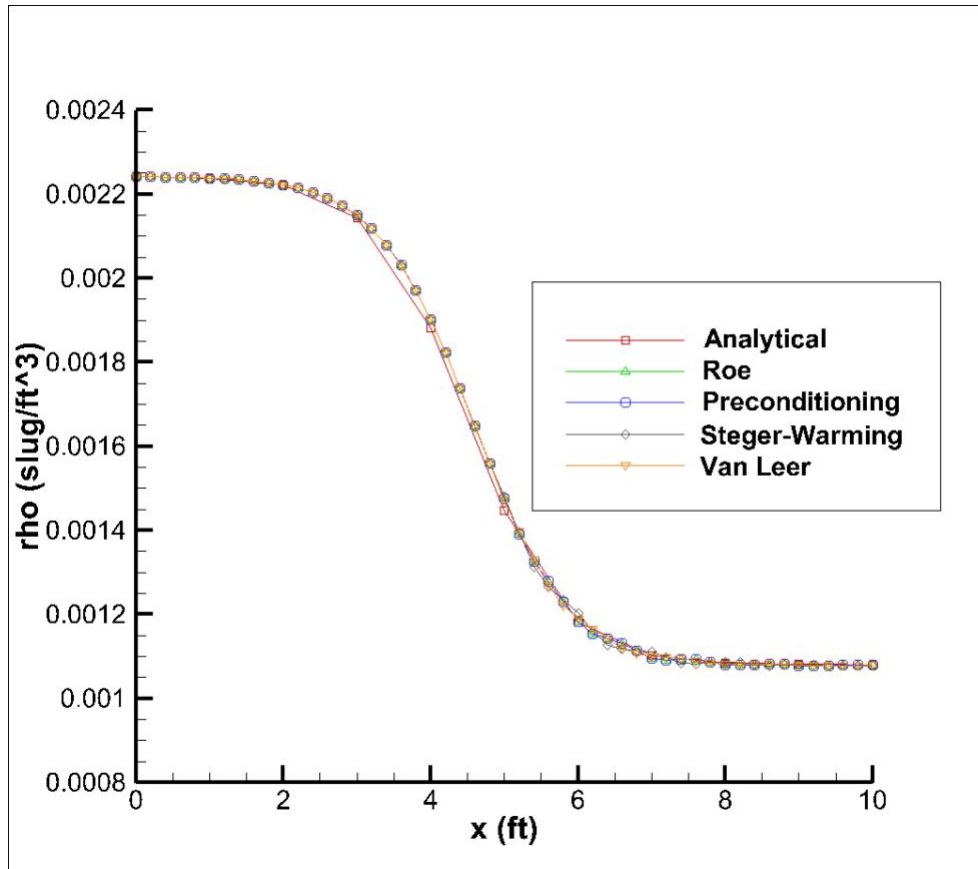


Figure 5.1.3: Comparison of density distributions for supersonic outflow

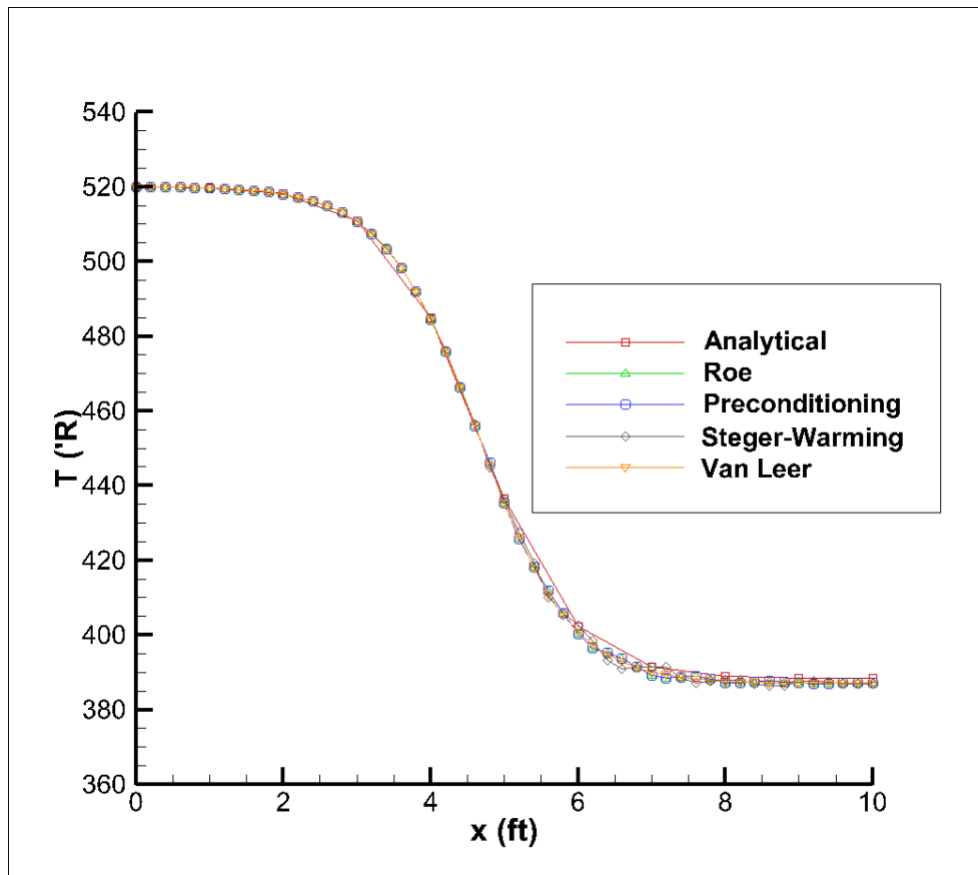


Figure 5.1.4: Comparison of temperature distributions for supersonic outflow

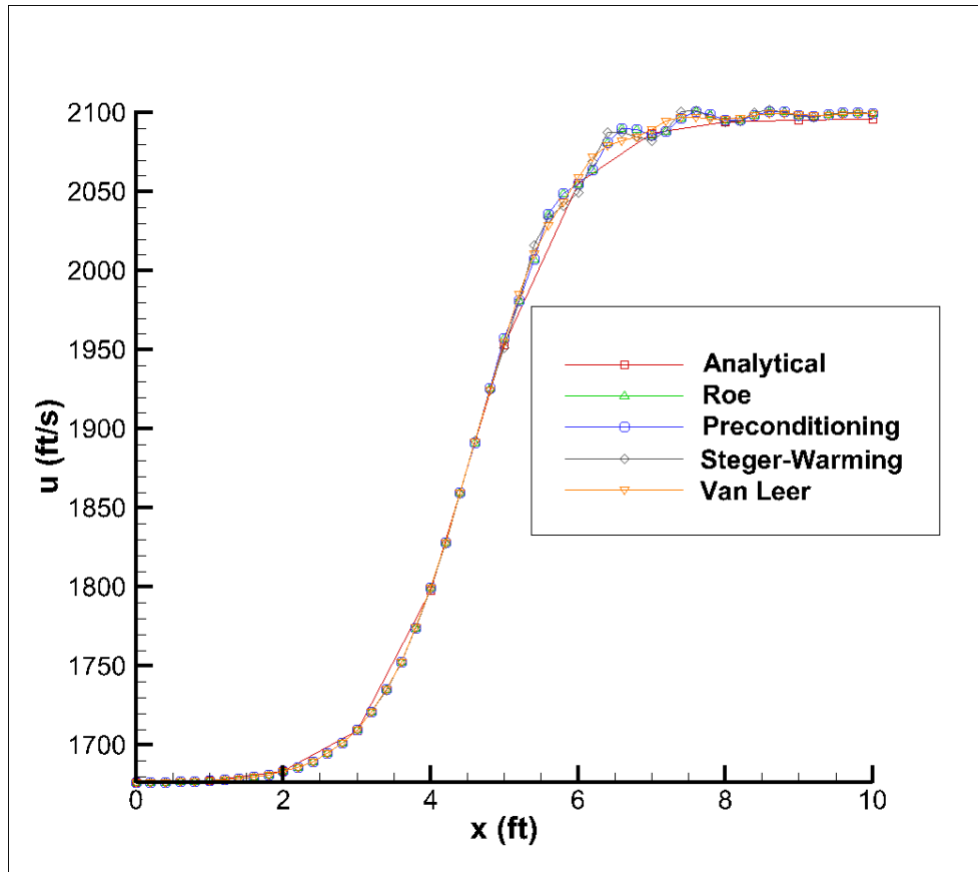


Figure 5.1.5: Comparison of velocity distributions for supersonic outflow

5.2 Case 2: Supersonic Inflow Subsonic Outflow

The second case for the proposed problem includes a shock wave within the domain followed by a subsonic flow. When the flow is subsonic, one of the eigenvalues is negative, which dictates that one of the characteristics is providing information into the domain. Thus, one analytical outflow boundary condition may be specified at the exit. The specification may be accomplished by defining ρ, u, p at the exit. In the following result, the velocity corresponds to a normal shock location of 5.0 ft within the domain

$$u = 1676.55 \text{ ft/sec} \quad x \leq 2.8$$

$$u = 572.76 \text{ ft/sec} \quad x > 2.8$$

$$\rho = 0.002241 \text{ slugs/ft}^3$$

$$p = 2000.00 \text{ lbf/ft}^2$$

The spatial step Δx is set to be 0.1 ft and temporal step as 0.00001 sec. The Courant-Friedrichs and Lewy (CFL) number is about 0.15, which would satisfy the stability requirement of the explicit scheme.

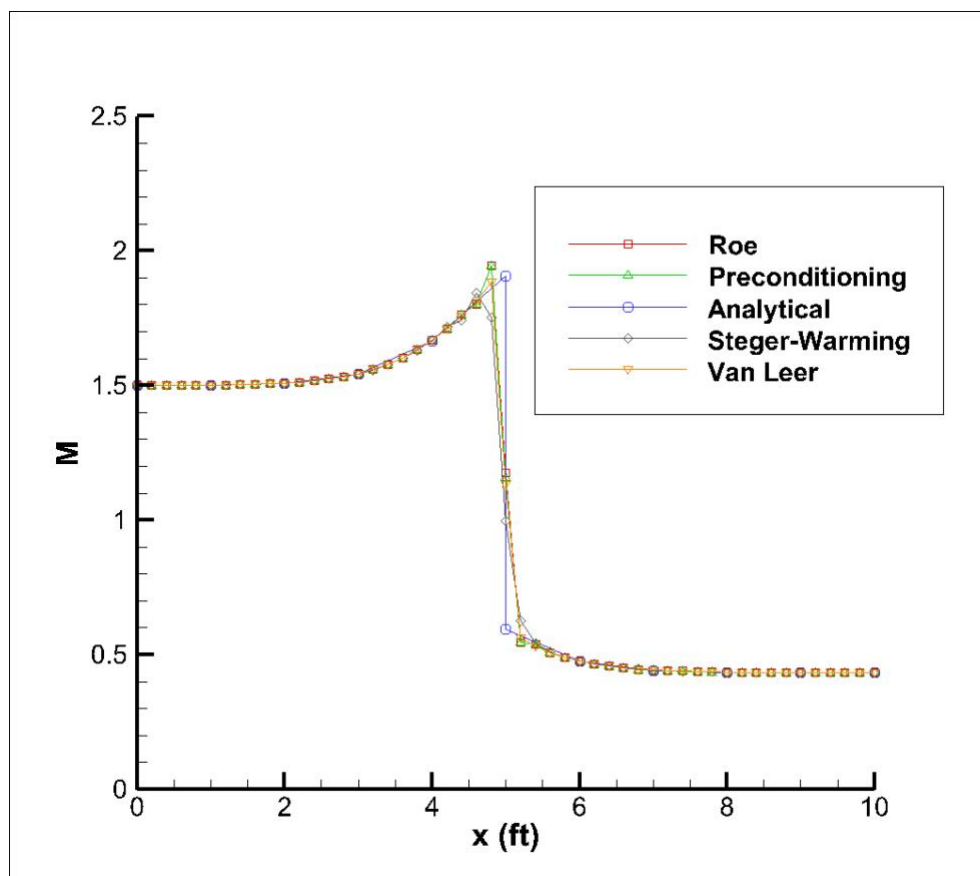


Figure 5.2.1: Comparison of Mach number distributions for subsonic outflow

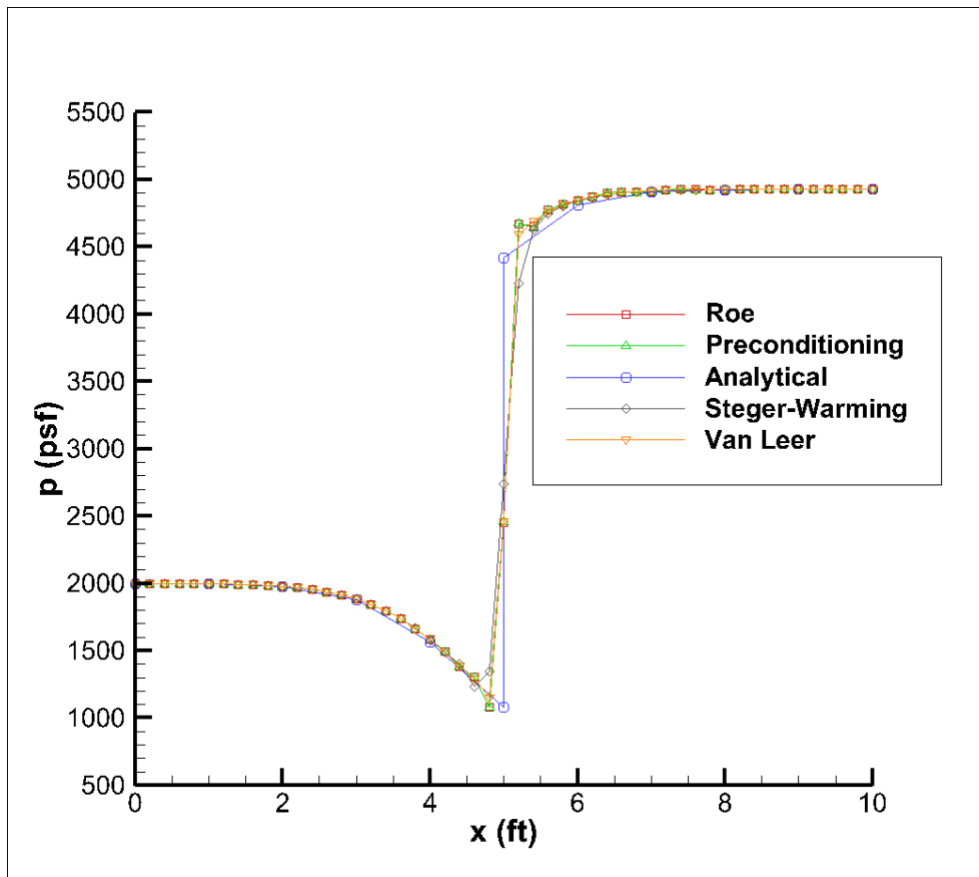


Figure 5.2.2: Comparison of pressure distributions for subsonic outflow

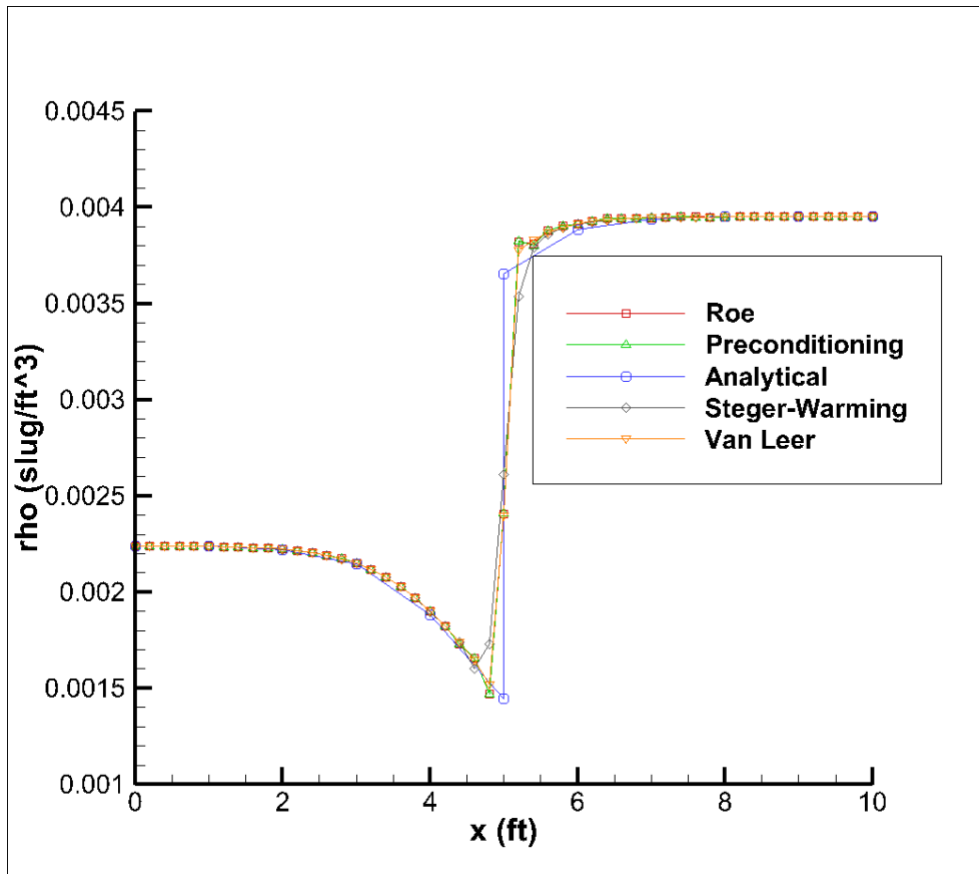


Figure 5.2.3: Comparison of density distributions for subsonic outflow

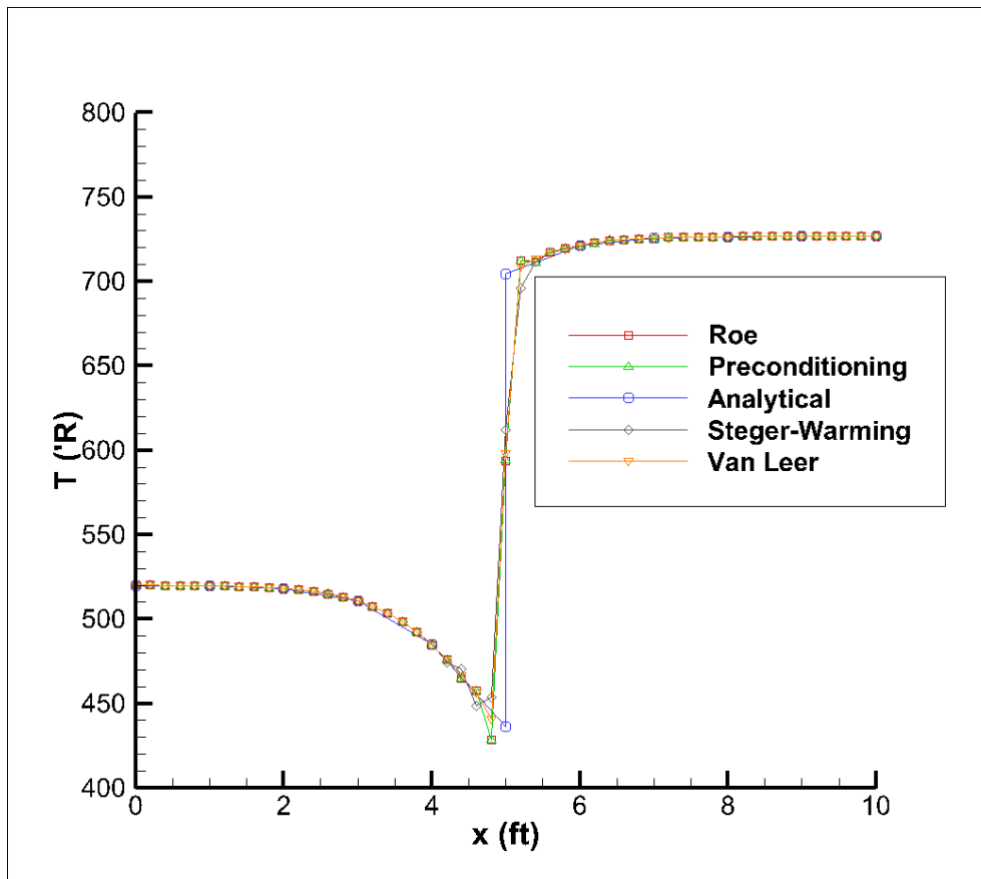


Figure 5.2.4: Comparison of temperature distributions for subsonic outflow

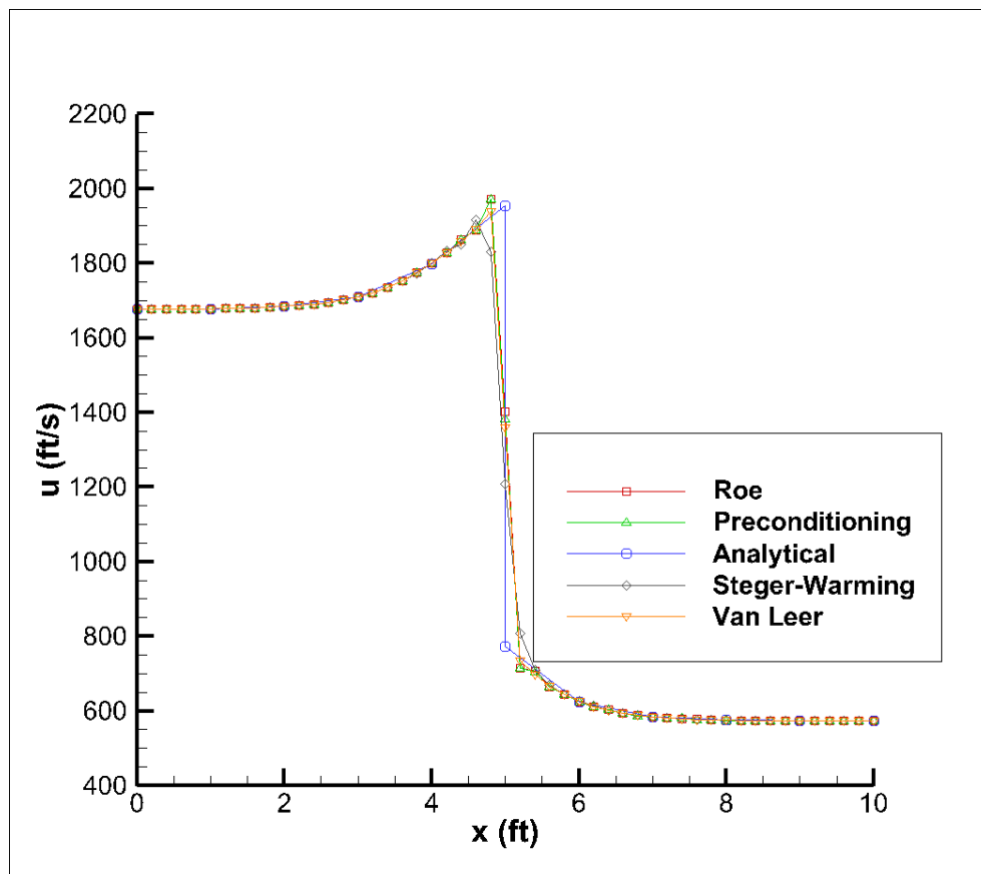


Figure 5.2.5: Comparison of velocity distributions for subsonic outflow

Chapter 6: Conclusions

The derivation of a time-derivative preconditioning from a quasi-one-dimensional Euler equations as the governing equation is presented, which is suitable for both variable and constant density fluids, thus enabling efficient solution of incompressible and variable density flows over a wide range of speeds.

The eigenvalues of the preconditioned system are initially derived by applying the preconditioning matrix Γ to replace the nonpreconditioned system's matrix, then using that new formula to find the associated eigenvalues of the system. The new eigenvalues are then used to calculate the equation of flux difference splitting for the preconditioned system, where several limitations and assumptions are done to address different needs for various cases of fluid flows. The derived preconditioning matrix is then implemented into an existing numerical solution for one-dimensional fluid problem of the diverging nozzle configuration for both supersonic and subsonic outflow, which is comprised of methods of MUSCL scheme with Roe's Riemann solver.

The steps of solving a one-dimensional fluid flow problem through applying the preconditioned flux difference splitting scheme that is derived by first introducing the preconditioning matrix into the system, gaining new eigenvalues are shown with the aid of a MUSCL type Roe scheme to calculate the left and right face elements of the flow.

Several cases are taken as an example of experimenting the newly derived flux difference splitting method, which shows that the results tend to have a better accuracy and performance than it is without preconditioning from the figures plotted, though only small changes can be observed when comparing both plots due to the fact that the conditions of the flow given are not extreme enough.

It can be observed from the first case where the outflow of the nozzle is supersonic, that the preconditioning method and Roe's approximate solver shows very similar results, with other numerical solution methods presented such as Steger-Warming's, Van Leer's and using analytical methods to give a wider perspective of how far do they differ.

On the other hand, the second case in which the outflow is subsonic, it can be seen that the preconditioning method approaches the most to its analytical solution and improving Roe's method, thus proving to have better results when compared with other numerical methods such as Steger-Warming's or Van Leer's. This shows that the preconditioning method reveals its usage clearer for low-speed-flows such as subsonic outflow, matching our initial goals of the research.

Chapter 7: Future Research and Recommendations

The majority of numerical solution methods used to solve quasi-one-dimensional fluid flow problem ranges from mainly explicit schemes algorithms, which suffers from serious time-step limitations comparing in using implicit approach. It is commonly implemented in conjunction with choosing unstructured meshes as the building blocks in discretizing the complex domains geometrically, which is capable of resolving various length scales of an evolving flow-field.

For low-Mach-number compressible flows, mentioned problem poses a greater threat due to the constraints present in the time-accuracy part, resulting in a low convergence rate observed as the effect of a more decreased time-step size, making the overall computational costs per solution higher than expected. It is thus recommended in using a time-marching schemes to cope up with the above deficiencies to further provide good stability and convergence characteristics. However, new setbacks emerge when trying to solve incompressible flows as its system is not fully hyperbolic, unlike for compressible flows, disabling the update of the pressure from its equation of state.

Another useful recommendation in replying to such cases is to introduce a pressure time derivative into not only the continuity, but also the entire equations involved in the system to ensure that the calculated results would bring an overall optimal convergence rate, thus making the system fully hyperbolic and the pressure could be updated. Due to the fact that a time-derivative preconditioning could destroy the time accuracy of the system, it is advised to apply a dual time-stepping method for future researches which involves an inner iteration loop in pseudo-time that is confined by an outer loop of a physical time-stepping to avoid it being affected by the stiffness of the system.

The potential of the imposed preconditioning method has proven to be that much of an impact whether it is for present or future researches. The presented method is applicable for flows such as multispecies that includes chemical reactions with mixtures of characteristics for each component and heat releases that take place in flows of real gases situations. Cavitating flows is also an important case that greatly needs the preconditioning method due to the trouble it presents by the cavitating bubbles created.

It is also expected to only rely on a density-based flow solver which is based on an unstructured mesh to keep up with the progress of this industry field, which has proven to be a better fluid flow solver choice in providing a steady-state solutions for time-dependent flows of incompressible and variable density ranging from different Reynolds number speeds.

Appendix A: MUSCL type Roe scheme for divergent nozzle configuration problem

The solution by the three-point explicit difference scheme MUSCL (Monotonic Upstream Schemes for Conservation Law) is to be given by the equations below. Various “slope” limiters are sued to eliminate unwanted oscillations, where the popular one such as the ”minmod” limiter is applied into the method. By $u_{j+\frac{1}{2}}^R$ and $u_{j+\frac{1}{2}}^L$. where u^R , u^L are defined for second- and third-order spatial differencing, it modifies the upwind-biased interpolation as follows:

$$u_{j+\frac{1}{2}}^R = u_{j+1} - \frac{1}{4} \left[(1 - \bar{\eta}) \widetilde{\Delta}_{j+\frac{3}{2}} + (1 + \bar{\eta}) \widetilde{\widetilde{\Delta}}_{j+\frac{1}{2}} \right], \quad (51)$$

$$u_{j+\frac{1}{2}}^L = u_j - \frac{1}{4} \left[(1 - \bar{\eta}) \widetilde{\widetilde{\Delta}}_{j-\frac{1}{2}} + (1 + \bar{\eta}) \widetilde{\Delta}_{j+\frac{1}{2}} \right], \quad (52)$$

$$\widetilde{\Delta}_{j+\frac{1}{2}} = \text{minmod} \left(\Delta_{j+\frac{3}{2}}, \omega \Delta_{j-\frac{1}{2}} \right), \quad (53)$$

$$\widetilde{\widetilde{\Delta}}_{j+\frac{1}{2}} = \text{minmod} \left(\Delta_{j+\frac{1}{2}}, \omega \Delta_{j+\frac{3}{2}} \right), \quad (54)$$

where the “minmod” limiter and delta functions are defined as:

$$\text{minmod}(z, \omega y) = \text{sgn}(z) \cdot \max\{0, \min[|z|, \omega y \text{sgn}(z)]\}, \quad (55)$$

$$\Delta_{j+\frac{1}{2}} = \frac{1}{2} (u_j + u_{j+1}), \quad (56)$$

$$\Delta_{j-\frac{1}{2}} = \frac{1}{2} (u_{j-1} + u_j), \quad (57)$$

and the omega is given as:

$$1 \leq \omega \leq \frac{3-\bar{\eta}}{1-\bar{\eta}}, \quad (58)$$

with $\bar{\eta} \neq 1$ and $\text{sgn}(z)$ denotes the sign of the variable z .

Here, the spatial order of accuracy is determined by the value of $\bar{\eta}$:

$$\bar{\eta} = -1, \text{ fully upwind scheme}, \quad (59)$$

$$\bar{\eta} = 0, \text{ Fromm scheme}, \quad (60)$$

$$\bar{\eta} = 1/3, \text{ third-order upwind-biased scheme}, \quad (61)$$

$$\bar{\eta} = 1, \text{ three-point central-difference scheme}, \quad (62)$$

where the value is taken as 1/3, making the scheme a three-point central-difference.

The scheme's solution and numerical flux by Roe's method are then redefined as below:

$$u_j^{n+1} = u_j - \lambda \left[h \left(u_{j+\frac{1}{2}}^R, u_{j+\frac{1}{2}}^L \right) - h \left(u_{j-\frac{1}{2}}^R, u_{j-\frac{1}{2}}^L \right) \right]^n, \quad (63)$$

where the definition of the flux function is given as:

$$\tilde{h}_{j+\frac{1}{2}}^{VL} = h \left(u_{j+\frac{1}{2}}^R, u_{j+\frac{1}{2}}^L \right) \quad (64)$$

$$= \frac{1}{2} \left[f \left(u_{j+\frac{1}{2}}^R \right) + f \left(u_{j+\frac{1}{2}}^L \right) - \left| \frac{f \left(u_{j+\frac{1}{2}}^R \right) - f \left(u_{j+\frac{1}{2}}^L \right)}{\left(u_{j+\frac{1}{2}}^R - u_{j+\frac{1}{2}}^L \right)} \right| \left(u_{j+\frac{1}{2}}^R - u_{j+\frac{1}{2}}^L \right) \right]. \quad (65)$$

Appendix B: Roe Flux Difference Splitting Scheme for divergent nozzle configuration problem

The average relationship for Roe's flux difference splitting scheme based on a quasi-one-dimensional Euler equations is defined as:

$$F[q(x=0)] = \frac{1}{2} + [F(u_{\text{right}}) + F(u_{\text{left}})] - \frac{1}{2} \sum_{k=1}^3 [e_{\text{RL},i} * \phi(\lambda_i) * \Delta Q_i]. \quad (66)$$

The values for the conservative variables are divided into three parts for a one by three matrix for a one-dimensional fluid flow problem, which are given as below:

$$\Delta Q_1 = \Delta \rho - \frac{\Delta p}{a_{\text{RL}}^2}, \quad (67)$$

$$\Delta Q_2 = \Delta u + \frac{\Delta p}{\rho_{\text{RL}} a_{\text{RL}}}, \quad (68)$$

$$\Delta Q_3 = \Delta u - \frac{\Delta p}{\rho_{\text{RL}} a_{\text{RL}}}. \quad (69)$$

where the each variables denoted by delta is defined as the difference between the right and left side of the face of the cells, given as:

$$\Delta \rho = \rho_{\text{R}} - \rho_{\text{L}}, \quad (70)$$

$$\Delta u = u_{\text{R}} - u_{\text{L}}, \quad (71)$$

$$\Delta e_t = e_{t\text{R}} - e_{t\text{L}}. \quad (72)$$

The matrix of the eigenvalues is derived from the Jacobian matrix in Roe's exact Riemann solver, and is found to be as the following:

$$\phi(\lambda_i) = \Lambda = \begin{bmatrix} u & 0 & 0 \\ 0 & u + c & 0 \\ 0 & 0 & u - c \end{bmatrix}. \quad (73)$$

An additional correction is added for the value of the matrix containing the eigenvalues of the system above to prevent any singularities at stagnation points of the fluid flow, given by:

$$\phi(\lambda_i) = \begin{cases} \frac{\lambda_i^2 + \delta^2}{2\delta} & \text{for } |\lambda_i| < \delta \\ |\lambda_i| & \text{for } |\lambda_i| \geq \delta \end{cases}, \quad (74)$$

$$\delta \approx 10^{-5}. \quad (75)$$

The corresponding eigenvectors for each of the eigenvalues are divided into three components, arranged as a matrix given as:

$$e_{RL,1} = \begin{bmatrix} 1 \\ u_{RL} \\ \frac{1}{2}u_{RL}^2 \end{bmatrix}, \quad (76)$$

$$e_{RL,2} = \frac{\rho_{RL}}{2a_{RL}} \begin{bmatrix} 1 \\ u_{RL} + a_{RL} \\ e_{t0,RL} + u_{RL}a_{RL} \end{bmatrix}, \quad (77)$$

$$e_{RL,3} = -\frac{\rho_{RL}}{2a_{RL}} \begin{bmatrix} 1 \\ u_{RL} - a_{RL} \\ e_{0,RL} - u_{RL}a_{RL} \end{bmatrix}. \quad (78)$$

The flux for Roe's method is then rearranged into a manner where only 3 components of flows, given as the absolute velocity, the difference in velocity and pressure in the following scalar form:

$$F = -\frac{1}{2} \left\{ \zeta \begin{bmatrix} \Delta\rho \\ \Delta(\rho u) \\ \Delta(\rho e_t) \end{bmatrix} + \delta u \begin{bmatrix} \rho \\ \rho u \\ \rho e_t \end{bmatrix} + \delta p \begin{bmatrix} 0 \\ 1 \\ u \end{bmatrix} \right\}, \quad (79)$$

where the variables for the flux formula is defined as:

$$\zeta = |u|, \quad (80)$$

$$\delta u = \frac{1}{\rho} \left(\frac{|u-c|-|u+c|}{2} - |u| \right) \beta + \frac{|u-c|-|u+c|}{2\rho c} [u\Delta\rho - \Delta(\rho u)], \quad (81)$$

$$\delta p = -\frac{|u-c|-|u+c|}{2} c\beta + \left[|u| - \frac{|u-c|-|u+c|}{2} \right] [u\Delta\rho - \Delta(\rho u)], \quad (82)$$

$$\beta = \frac{\gamma-1}{c^2} \left[\frac{V_M^2}{2} \Delta\rho - u\Delta(\rho u) + \Delta(\rho e_t) \right]. \quad (83)$$

Appendix C: Computer codes for solving the divergent nozzle configuration problem

```
program nozzle_preconditioning
  common /cdt/dx,dt,del,r,r1,pi,imm1,imm2,imax,imp1,imp2,xmax,er
  dimension p(-2:500),u(-2:500),rho(-2:500),s(-2:500),ds(-2:500)
  dimension q(-2:500,3),qq(-2:500,3),h(-2:500,3),ept(-2:500,3)
  dimension ef(-2:500,3),et(-2:500),fpt(-2:500,3),gpt(-2:500,3)
  dimension qpr(-2:500,3),qpl(-2:500,3),qrm(-2:500,3),qlm(-2:500,3),q1(-2:500,3)
  dimension fpn(-2:500,3),fpp(-2:500,3),gpn(-2:500,3),gpp(-2:500,3),sig(-2:500)
  dimension hp(-2:500,3),hn(-2:500,3),vpr(-2:500,3),vpl(-2:500,3),dsp(-2:500,3)
  r=1.4
  r1=0.4
  er=1716.
  ifrog=100
  nt=30000
  imax=51
  dx=0.2
  dt=0.00001
  del=dt/dx
  t=0.
  tf=0.1
  imp2=imax+2
  imp1=imax+1
  imm1=imax-1
  imm2=imax-2

  !Area
  do i=1,imax
    x=(i-1)*dx
    ds(i)=0.2776*(1./(cosh(0.8*x-4.))**2))
    s(i)=1.398+0.347*tanh(0.8*x-4.)
  end do

  !Initial condiiton
  !subsonic
  do i=1,imax
    x=(i-1)*dx
    if(x<=2.8)then
      u(i)=1676.55
    else
      u(i)=572.76
    end if
    rho(i)=0.002241
    p(i)=2000.0
  end do
```

```

!supersonic
do i=-1,imp2
  u(i)=1676.55
  rho(i)=0.002241
  et(i)=3636204. !p(i)/(rho(i)*r1)+0.5*u(i)**2
  p(i)=rho(i)*(r1)*(et(i)-0.5*u(i)**2)
  temp=p(i)/(rho(i)*er)
  a=(r*er*temp)**0.5
  xm=u(i)/a
end do

!Boundary condition
!subsonic
do i=-1,0
  rho(i)=0.002241
  p(i)=2000.0
  u(i)=1676.55
end do

do i=imp1,imp2
  rho(i)=0.002241
  p(i)=2000.0
  u(i)=572.76
end do

!supersonic
do i=-1,imp2
  x=(i-1)*dx
  q(i,1)=rho(i)
  q(i,2)=rho(i)*u(i)
  q(i,3)=p(i)/(r1)+0.5*rho(i)*u(i)**2
end do

!Time iteratoin
!do m=1,nt
do while(t<=tf)
t=t+dt

  !call muscl
  call muscl(q,qpr,qpl,qrm,q1m,vpr,vpl,q1)

  !call roe
  call roe(vpr,vpl,qpr,qpl,rhopr,rhopl,upr,u1p,ppr,ppl,epr,epl,dsp)

  !call fvs
  call fvsm(s,qpr,q1,fpn,rhopn,upn,ppn,etpn)
  call fvsp(s,qpl,q1,fpp,rhopp,upp,ppp,etpp)

```

```

do j=1,3
  x=(i-1)*dx
  !subsonic
  do i=1,imax
    !fvs
    ef(i,j)=((fpp(i,j)+fpn(i,j)))

    !roe
    ef(i,j)=0.5*((fpp(i,j)+fpn(i,j))-dsp(i,j))
  end do
end do

do i=1,imax
  x=(i-1)*dx
  h(i,1)=0 !ds(i)*q(i,2)
  h(i,2)=ds(i)*(p(i))
  h(i,3)=0 !ds(i)*(r*q(i,3)*q(i,2)/q(i,1)-0.5*r1*q(i,2)**3/q(i,1)**2)

  do k=1,3
    qq(i,k)=q(i,k)-(del*(1./s(i))*(ef(i,k)-ef(i-1,k)))+((1./s(i))*dt*h(i,k))
  end do

end do

!New value subsonic
do i=2,imm1
  do j=1,3
    q(i,j)=qq(i,j)
  end do

  rho(i)=q(i,1)
  u(i)=q(i,2)/q(i,1)
  et(i)=q(i,3)/q(i,1)
  p(i)=(r1)*(q(i,3)-0.5*q(i,2)**2/q(i,1))
end do

!New value supersonic
do i=2,imax
  do j=1,3
    q(i,j)=qq(i,j)
  end do

  rho(i)=q(i,1)
  u(i)=q(i,2)/q(i,1)
  et(i)=q(i,3)/q(i,1)
  p(i)=(r1)*(q(i,3)-0.5*q(i,2)**2/q(i,1))
end do

```

```

!Boundary condition subsonic
do i=imax,imp2
  q(i,1)=q(imm1,1)
  q(i,2)=q(imm1,1)*572.76
  q(i,3)=(p(imm1)/(r1))+(0.5*q(imm1,1)*572.76**2)
end do

do i=-1,1
  q(i,1)=q(i+1,1)
  q(i,2)=q(i,1)*1676.55
  q(i,3)=(2000./(r1))+(0.5*q(i,1)*1676.55**2)
end do

!Extrapolation supersonic for roe
do i=imp1,imp2
  q(i,1)=q(imax,1)
  q(i,2)=(q(imax,1)*1676.55)
  q(i,3)=((p(imax)/(r1))+(0.5*q(imax,1)*1676.55**2))
  q(i,1)=2.*q(imax,1)-q(imm1,1)
  q(i,2)=2.*(q(imax,1)*u(imax))-(q(imm1,1)*u(imm1))
  q(i,3)=2.*((p(imax)/(r1))+(0.5*q(imax,1)*u(imax)**2))-
  ((p(imm1)/(r1))+(0.5*q(imm1,1)*u(imm1)**2))
end do
end do

!Error
terr=0.
if(mod(i,ifrog)==0) go to 776

!=0
do i=0,imax
  trmax=qq(i,1)-q(i,1)
  if(abs(terr)<abs(trmax))then
    terr=trmax
    ii=i
  end if
end do

write(*,*)i,ii,terr,qq(ii,1)

if (terr<1e-5)go to 1001

!New value
776 do j=1,3
  do i=2,imax
    q(i,j)=qq(i,j)
  end do
end do

```

```

!Output
1001 open(11,file='p.plt')
    open(12,file='m.plt')
    open(13,file='u.plt')
    open(14,file='rho.plt')
    open(15,file='temp.plt')
    open(16,file='ft.plt')

do i=1,imax
    rho(i)=q(i,1)
    u(i)=q(i,2)/q(i,1)
    et(i)=q(i,3)/q(i,1)
    p(i)=rho(i)*(r1)*(et(i)-0.5*u(i)**2)
    temp=p(i)/(rho(i)*er)
    a=(r*er*temp)**0.5
    xm=u(i)/a

    xx=(i-1)*dx
    write(*,100)xx,s(i),xm,p(i),temp,rho(i),u(i)
100  format(' ',2f10.4,f10.3,2f10.2,f10.6,f10.2)
    write(11,*)xx,p(i)
    write(12,*)xx,xm
    write(13,*)xx,u(i)
    write(14,*)xx,rho(i)
    write(15,*)xx,temp
    write(16,*)xx,et(i)
end do
end program

!minmod
function smd(x,y)
    sgn=sign(1.0,x)+sign(1.0,y)
    smd=amin1(abs(x),abs(y))*sgn*0.5
return
end function

!muscl
subroutine muscl(q0,qpr,qpl,qrm,qlm,vpr,vpl,q1)
    common /cdt/dx,dt,del,r,r1,pi,imm1,imm2,imax,imp1,imp2,xmax,er
    dimension q1(-2:500,3)
    dimension q0(-2:500,3)
    dimension vpr(-2:500,3),vpl(-2:500,3)
    dimension vnr(-2:500,3),vnl(-2:500,3)
    dimension qpr(-2:500,3),qpl(-2:500,3)
    dimension qrm(-2:500,3),qlm(-2:500,3)

    real niu
    niu=1./3.

```

```

do i=-1,imp2
  q1(i,1)=q0(i,1) !rho
  q1(i,2)=q0(i,2)/q0(i,1) !rho*u/rho=u
  q1(i,3)=0.4*(q0(i,3)-0.5*q0(i,2)*q1(i,2))
end do

do k=1,3

  do i=1,imax
    qp2=q1(i+2,k)-q1(i+1,k)
    qp1=q1(i+1,k)-q1(i,k)
    qp0=q1(i,k)-q1(i-1,k)
    qm1=q1(i-1,k)-q1(i-2,k)

    vpr(i,k)=q1(i+1,k)-1./4.*((1.-niu)*smd(qp2,qp1)+(1.+niu)*smd(qp1,qp2))
    vpl(i,k)=q1(i,k)+1./4.*((1.-niu)*smd(qp0,qp1)+(1.+niu)*smd(qp1,qp0))
    vnr(i,k)=q1(i,k)-1./4.*((1.-niu)*smd(qp1,qp0)+(1.+niu)*smd(qp0,qp1))
    vnl(i,k)=q1(i-1,k)+1./4.*((1.-niu)*smd(qm1,qp0)+(1.+niu)*smd(qp0,qm1))

    qpr(i,1)=vpr(i,1) !rho
    qpr(i,2)=vpr(i,1)*vpr(i,2) !rho*u
    qpr(i,3)=vpr(i,3)*2.5+0.5*vpr(i,2)*qpr(i,2)

    qpl(i,1)=vpl(i,1)
    qpl(i,2)=vpl(i,1)*vpl(i,2)
    qpl(i,3)=vpl(i,3)*2.5+0.5*vpl(i,2)*qpl(i,2)

    qrm(i,1)=vnr(i,1)
    qrm(i,2)=vnr(i,1)*vnr(i,2)
    qrm(i,3)=vnr(i,3)*2.5+0.5*vnr(i,2)*qrm(i,2)

    qlm(i,1)=vnl(i,1)
    qlm(i,2)=vnl(i,1)*vnl(i,2)
    qlm(i,3)=vnl(i,3)*2.5+0.5*vnl(i,2)*qlm(i,2)
  end do

end do

end subroutine muscl

!flux plus
subroutine fvsp(s,qp,q1,fps,rho,u,p,et)
common /cdt/dx,dt,del,r,r1,pi,imm1,imm2,imax,imp1,imp2,xmax,er
dimension qp(-2:500,3),fp(-2:500,3),fps(-2:500,3),S(-2:500),q1(-2:500,3)
real miu,ms
!          | rho*u |      | rho |
!  subroutine OBTAINS f=| rho*u**2+p | from q=| rho*u |
!          | u*(e+p) |      | e |
miu=1e-4

```

```

do j=1,imax

  rho=qp(j,1)
  u=qp(j,2)/qp(j,1)
  p=r1*(qp(j,3)-0.5*qp(j,2)**2/qp(j,1))
  et=qp(j,3)/qp(j,1)
  a=sqrt(r*p/qp(j,1))
  xm=u/a

  !van leer
  if(xm>=1.) then
    fp(j,1)=rho*u
    fp(j,2)=fp(j,1)*u+p
    fp(j,3)=(qp(j,3)+p)*u
  else if(xm<=-1.)then
    fp(j,1)=0.
    fp(j,2)=0.
    fp(j,3)=0.
  else
    fp(j,1)=(0.25*rho*a*(1.+xm)**2)
    fp(j,2)=((0.25*rho*a**2*(1.+xm)**2)*((r1)/r*xm+2./r))
    fp(j,3)=(r**2/(2.*(r**2-1.)))*(fp(j,2)**2/fp(j,1))
  end if

  !steger-warming
  fp(j,1)=rho/(2.*r)*(2.*r*u+a-u)
  fp(j,2)=rho/(2.*r)*(2.*r1*u**2+(u+a)**2)
  fp(j,3)=rho/(2.*r)*(r1*u**3+((u+a)**3)/2.+((3.-r)*(u+a)*a**2)/(2.*r1))

  !roe
  fp(j,1)=rho*u
  fp(j,2)=(r1/r)*rho*(et+p/rho)+(((r+1.)/(2.*r))*rho*u**2)
  fp(j,3)=rho*u*(et+p/rho)

  fps(j,1)=s(j)*fp(j,1)
  fps(j,2)=s(j)*fp(j,2)
  fps(j,3)=s(j)*fp(j,3)
end do

return

end subroutine fvsp

!flux min
subroutine fvsm(s,qn,q1,fms,rho,u,p,et)
common /cdt/dx,dt,del,r,r1,pi,imm1,imm2,imax,imp1,imp2,xmax,er
dimension qn(-2:500,3),fn(-2:500,3),fms(-2:500,3),s(-2:500),q1(-2:500,3)
real miu,ms

```

```

!          | rho*u |      | rho |
!  subroutine contains f=| rho*u**2+p | from q=| rho*u |
!          | u*(e+p) |      | e |
miu=1e-4

do j=1,imax

  rho=qn(j,1)
  u=qn(j,2)/qn(j,1)
  p=r1*(qn(j,3)-0.5*qn(j,2)**2/qn(j,1))
  et=qn(j,3)/qn(j,1)
  a=sqrt(r*p/qn(j,1))
  xm=u/a

  !van leer
  if(xm>=1.0) then
    fn(j,1)=0.
    fn(j,2)=0.
    fn(j,3)=0.
  else if(xm<=-1.0)then
    fn(j,1)=qn(j,2)
    fn(j,2)=fn(j,1)*u+p
    fn(j,3)=(qn(j,3)+p)*u
  else
    fn(j,1)=(-0.25*rho*a*(1.-xm)**2)
    fn(j,2)=((0.25*rho*a**2*(1.-xm)**2)*(-(r1)/r*xm+2./r))
    fn(j,3)=(r**2/(2.*(r**2-1.)))*(fn(j,2)**2/fn(j,1))
  end if

  !steger-warming
  fn(j,1)=rho/(2.*r)*(u-a)
  fn(j,2)=rho/(2.*r)*(u-a)**2
  fn(j,3)=rho/(2.*r)*(((u-a)**3)/2.+((3.-r)*(u-a)*a**2)/(2.*r1))

  !roe
  fn(j,1)=rho*u
  fn(j,2)=(r1/r)*rho*(et+p/rho)+(((r+1.)/(2.*r))*rho*u**2)
  fn(j,3)=rho*u*(et+p/rho)

  fms(j,1)=s(j)*fn(j,1)
  fms(j,2)=s(j)*fn(j,2)
  fms(j,3)=s(j)*fn(j,3)
end do

return

end subroutine fvsm

```

```

subroutine roe(vpr,vpl,qr,ql,rhopr,rhopl,upr,upl,ppr,ppl,epr,epl,dsp)
  common /cdt/dx,dt,del,r,r1,pi,imm1,imm2,imax,imp1,imp2,xmax,er
  dimension qr(-2:500,3),fr(-2:500,3),frs(-2:500,3),S(-2:500),ql(-2:500,3),fl(-2:500,3),fls(-
2:500,3),dsp(-2:500,3)
  dimension e(-2:500,3),w(-2:500,3),qz(-2:500,3),sig(-2:500,3),eig(-2:500,3),q1(-2:500,3)
  dimension e12(-2:500,3),e13(-2:500,3),e21(-2:500,3),e22(-2:500,3),e23(-2:500,3),e31(-
2:500,3)
  dimension e32(-2:500,3),e33(-2:500,3),e11(-2:500,3),vpr(-2:500,3),vpl(-2:500,3)
  real miu,ms
  !          | rho*u |      | rho |
  !  subroutine OBTAINS f=| rho*u**2+p | from q=| rho*u |
  !          | u*(e+p) |      | e |
  miu=1e-4

do j=1,imax
  rhopr=vpr(j,1)
  rhopl=vpl(j,1)
  upr=vpr(j,2)
  upl=vpl(j,2)
  ppr=vpr(j,3)
  ppl=vpl(j,3)
  epr=qr(j,3)/qr(j,1)
  epl=ql(j,3)/ql(j,1)

  rhoud=rhopr*upr-rhopl*upl
  rhoed=rhopr*epr-rhopl*epl

  rhod=0.5*(rhopr+rhopl)
  ed=0.5*(epr+epl)
  ud=0.5*(upr+upl)
  pd=0.5*(ppr+ppl)
  ad=sqrt(r*pd/rhod)

  !PRECONDITIONING
  xc=miu*ad

  !viscous flows
  ur=0.5*(ur+ud/dx+abs(ur-ud/dx))

  !ideal gas
  if(abs(ud)<xc)then
    ur=xc
  else if(abs(ud)>ad)then
    ur=ad
  else
    ur=abs(ud)
  end if

```

```

!max incompressible
if(abs(ud)<miu*ur)then
    ur=miu*ur
else if(abs(ud)>miu*ur)then
    ur=abs(ur)
end if

bet=1./(ad*ad)!ideal gas

!For nonpreconditioned
ur=ad !(nonpreconditioned)
alf=0 !(nonpreconditioned)
cx=ur !(nonpreconditioned)
ux=ud !(nonpreconditioned)

!For preconditioned
alf=0.5*(1-bet*ur*ur) !(preconditioned)
cx=(alf*alf*ud*ud+ur*ur)**0.5 !(preconditioned)
ux=ud*(1-alf) !(preconditioned)

cs=0.5*(abs(ux+cx)+abs(ux-cx))
ms=0.5*(abs(ux+cx)-abs(ux-cx))/cx
delu=ms*(upr-upl)+(cs-(1-2*alf)*abs(ud)-alf*ud*ms)*(ppr-ppl)/(rhod*ur*ur)
delp=ms*(ppr-ppl)+(cs-abs(ud)+alf*ud*ms)*rhod*(upr-upl)

dsp(j,1)=abs(ud)*(rho pr-rho pl)+delu*rhod
dsp(j,2)=abs(ud)*(rho ud)+delu*rhod*ud+delp
dsp(j,3)=abs(ud)*(rho ed)+delu*rhod*(ed+pd/rhod)+delp*ud
end do
return
end subroutine roe

```

References

- [1] van Leer, B., “Towards the Ultimate Conservative Difference Scheme V. A Second-Order Sequel to Godunov’s Method” *Journal of Computational Physics*, Vol. 32, pp. 101-136, 1979.
- [2] Roe, P. L., “Approximate Riemann Solvers, Parameter Vectors and Difference Scheme” *Journal of Computational Physics*, Vol. 43, pp. 357-372, 1981.
- [3] Jonathan M. Weiss & Wayne A. Smith, “Preconditioning Applied to Variable and Constant Density Flows”, Vol.33, No. 11, November 1995.
- [4] Beam, R. M., and Warming, R. F., "An Implicit Finite Difference Algorithm for Hyperbolic Systems in Conservation Law Form," *Journal of Computational Physics*, Vol. 22, 1976, p.87-110.
- [5] Briley, W. R., and McDonald, H., "Solution of the Multi-Dimensional Compressible Navier-Stokes Equations by a Generalized Implicit Method," *Journal of Computational Physics*, Vol. 24, 1977, p. 372-397.
- [6] Pulliam, T. H., and Stager, J. L., "Recent Improvements in Efficiency, Accuracy and Convergence for Implicit Approximate Factorization Algorithms," *AIAA Paper 85-0360*, Jan. 1985.
- [8] Choi, Y. H., and Merkle, C. L., "The Application of Preconditioning in Viscous Flows," *Journal of Computational Physics*, Vol. 105, 1993, pp.207-233.
- [9] Turkel, E., "Preconditioned Methods for Solving the Incompressible and Low Speed Compressible Equations," *Journal of Computational Physics*, Vol. 72, 1987, pp. 277-298.
- [10] Merkle, C. L., and Athavale, M. M., "Time-Accurate Unsteady Incompressible Flow Algorithms Based on Artificial Compressibility," *AIAA Paper 87-1137*, 1987.
- [11] Roe, P. L., "Characteristic Based Schemes for the Euler Equations," *Annual Review of Fluid Mechanics*, Vol.18,1986, p337-365.
- [12] van Leer, B., Lee, W. T, and Roe, P., "Characteristic Time-Stepping or Local Preconditioning of the Euler Equations," *AIAA Paper 91-1552*,1991.
- [13] X.s. Li, “On the All-Speed Roe-type Scheme for Large Eddy Simulation of Homogeneous Decaying Turbulence.”, *International Journal of Computational Fluid Dynamics*, September 2015.
- [14] Shin, B. R. “Numerical Analysis of Unsteady Cavitating Flow by a Homogeneous Equilibrium Model” *AIAA Paper*, pp. 2001-2909, 2001.
- [15] Kunz, R. F. et al. “A Preconditioned Navier-Stokes Method for Two- Phase Flows With Application to Cavitation Prediction” *Computational Fluids*, Vol. 29, pp. 849-875, 2000.