Pressure Boundary Conditions for Compressible Flow Simulations using Conservation Variables

Seminar: CES

Author: *Mario M. Koretz*

Supervisor: *Max von Danwitz, M. Sc.*

30.11.2016

Abstract:

The following document discusses methods of prescribing subsonic pressure boundary conditions for compressible flow problems which is part of a seminar thesis handed out by the Chair of Computational Analysis of Technical Systems at the RWTH Aachen. Thorough evaluations of the test runs of boundary values methods are preceded by the underlying mathematical preliminaries crucial for understanding the theoretical background.

Contents

1 Introduction **1**

1.1 Topic of this Seminar Paper . . . 1

1.2 Properties of Compressible Flows . . . 1

2 Mathematical Preliminaries **3**

2.1 The Governing Equations . . . 3

2.1.1 Euler Equations . . . 3

2.1.2 Weak Formulation . . . 4

2.1.3 Solution Strategies . . . 5

2.2 Applying Boundary Conditions . . . 6

2.2.1 Prescribing the Total Energy per Unit Volume (Method A) . . . 6

2.2.2 Adapting the Euler Fluxes (Method B) . . . 7

2.2.3 Riemann Problem on Boundaries (Method C) . . . 7

3 Test Case Definitions **10**

3.1 Tested Geometries . . . 10

3.2 Tested Flow Conditions . . . 11

3.2.1 Supersonic . . . 11

3.2.2 Critical . . . 11

3.2.3 Shocked . . . 12

3.3 Simulation Setup . . . 13

4 Results and Discussion **15**

4.1 Solution Visualizations . . . 15

4.1.1 Supersonic Nozzle Flow . . . 16

4.1.2 Critical Nozzle Flow . . . 16

4.1.3 Shocked Nozzle Flow . . . 18

4.2 Convergence Behavior . . . 18

4.2.1 Supersonic Nozzle Flow . . . 20

4.2.2 Critical Nozzle Flow . . . 20

4.2.3 Shocked Nozzle Flow . . . 21

4.3 Conclusion . . . 22

Appendices

A XNS Example Input Files **24**

Bibliography **26**

1 Introduction

1.1 Topic of this Seminar Paper

The focus of this document lies on working out the differences between different approaches of prescribing subsonic pressure boundary conditions in the regime of compressible flow problems by documenting the results of multiple characteristic test cases. These methods are applied to a two-dimensional model of a converging-diverging nozzle within XNS, the finite element solver developed by CATS.

Additionally, this document presents the underlying mathematical preliminaries forming the bases of the tested methods and the simulated compressible flow. The corresponding section 2 leads from the Euler equations governing the physical problem over the actual implemented weak form to the mathematical representation of the tested methods. These topics are treated rather roughly to establish a knowledge base within the surrounding context. Three distinct methods are tested and evaluated originating from specific mathematical approaches. Descriptions of these are given in detail in section 2.2. The final evaluations of the methods are of course based on the error in the results but supplemented with considerations of the observed convergence behavior. Utilizing the won insights creates a report and feedback on the correct implementation of the methods and the functionality of the compressible flow solver in XNS. Section 4 consists of the pertaining compilation of the respective discussions whereas section 3 gives an overview of the applied test cases and general remarks about the simulation setup in XNS.

1.2 Properties of Compressible Flows

In contrast to *incompressible* flows, the density ρ is not a constant quantity in the regime of *compressible* flow problems but rather needs to be modeled as another field variable $\rho = \rho(x, t)$ changing in time and varying across the entire domain. Generally, compressible flow is characterized by a Mach number greater than 0.3 which describes a threshold where, due to the acting forces on the fluid, effects of compression can no longer be neglected [2]. Apart from physical considerations of the flow, the governing equations and respective stabilization techniques differ from incompressible flow problems. This creates a demand for a separated consideration of this group of problems altogether.

Introducing boundary conditions for compressible flow problems generally works in the same way it does for incompressible flows. However, by the virtue of the underlying physical

problems, being able to prescribe a certain pressure at an arbitrary boundary, for instance the outlet of a jet engine, is necessary. This can neither be achieved through a Dirichlet nor a Neumann boundary condition as the pressure is not part of the conservation variables which are used to model compressible flows in the given context.

As a consequence, methods need to be developed in order to resolve this issue and explicitly define pressure values at arbitrary points of the finite element model. However, those are restricted to the condition that there is no negative impact on the calculated results. This can obviously not be achieved completely and therefore leads to the given work of testing and comparing different approaches to obtain a final report on how this issue is resolved optimally.

2 Mathematical Preliminaries

2.1 The Governing Equations

2.1.1 Euler Equations

The following derivation follows the pertaining chapter to the treatment of compressible flows in "Finite Element Method for Flow Problems" [1]. Note that, although an inviscid is assumed for the derivations below, the actual implementation in the solver takes viscosity into account which is then simply set to zero.

From the conservation of mass, momentum and total internal energy per unit volume, the compressible Navier-Stokes equations for describing inviscid, compressible flow with the neglect of heat-conduction, which are known as the Euler equations, can be written as

$$\begin{aligned} U_t + \nabla \cdot F &= B && \text{in } \Omega \times]0, T[\,, \\ F^{in} \cdot n &= G && \text{on } \Gamma \times]0, T[\,, \\ U(x, 0) &= U_0(x) && \text{on } \Omega \text{ at } t = 0\,, \end{aligned}$$

where

$$U = \begin{bmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho E \end{bmatrix}, \; F = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}, \; F_i = \begin{bmatrix} \rho u_i \\ \rho u_i \, u_1 + \delta_{i1} p \\ \rho u_i \, u_2 + \delta_{i2} p \\ \rho u_i \, u_3 + \delta_{i3} p \\ (\rho e + p) \, u_i \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ \rho b \\ u \cdot \rho b \end{bmatrix}.$$

The vector U represents the conservation variables and F_i defines the respective fluxes in each dimension. B contains the source terms for the conservation equations of mass, momentum and energy with ρb being "the external force per unit volume". This term is assumed to be zero as body forces are of vanishing influence for the considered Mach numbers. This is taken into account in the derivations below.

Applying the chain rule $\frac{\partial F_i}{\partial x_i} = \frac{\partial F_i}{\partial U} \frac{\partial U}{\partial x_i}$, one obtains the quasi-linear form

$$U_t + A_i \frac{\partial U}{\partial x_i} = 0, \tag{2.1}$$

where

$$A = \frac{\partial F_i}{\partial U}$$

is called the *Euler Jacobian Matrix*. Note that the formulations of the boundary and initial conditions above still hold.

2.1.2 Weak Formulation

The equation given above is supposed to be discretized by means of the finite element method. In order to so, a so called weak formulation needs to be derived. This is achieved by a multiplication by a test function W^h and an integration over the whole space-time slab $Q = \Omega \times]0, T[$ leading to the new formulation of the problem

$$\int_Q W^h \cdot \left(U_t^h + A_i \cdot U_{x_i}^h \right) \, dQ = 0 \,, \tag{2.2}$$

where the choice of the test function and trial solution spaces conforms to the usual Galerkin approach. The "h" superscript denotes the consideration of the equation in a discrete function space, which is necessary for the numerical approximation of the solution. Additional terms which, among others, take care of stabilization and jump conditions are to be added to the expression above. This is further described in the next part of this section.

In order to compensate for the naturally given discontinuity of the solution between discrete time steps due to inhomogeneous Neumann boundary conditions, a so called "jump condition" is added to the weak formulation. It can also be considered as a "coupling between consecutive time-slabs" as it is formulated in the respective technical notes to the solver. This, the theory of which is not discussed here, leads to the term

$$\mathcal{T}_{jump} = \int_{\Omega_n} (W^h)_n^+ \cdot ((U^h)_n^+ - (U^h)_n^-) \, d\Omega \,. \tag{2.3}$$

Stabilization of the solution by means of the SUPG-method is introduced with the term

$$\mathcal{T}_{SUPG} = \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_{MOM} \left[(A_k^h)^T \frac{\partial W^h}{\partial x_k} \right] \cdot \left[\frac{\partial U^h}{\partial t} + A_i^h \frac{\partial U^h}{\partial x_i} \right] \, dQ \,, \tag{2.4}$$

where τ_{MOM} is a diagonal matrix containing stabilization parameters for each degree of freedom. This term accounts for missing physical diffusivity by added numerical diffusivity which supports the effort of avoiding oscillations of the solution between time steps.

Unresolved discontinuities of the conservation variables in space can occur which lead to convergence issues in terms of oscillations in proximity to this discontinuity when resolving the solution at such a "jump" point. Apart from the numerical origin of this occurrence due to the discretization of the physical space Ω, so called "shocks", which are actually

a physical phenomenon, also lead to such discontinuities in the solution. Given by the mathematical similarity of these issues, both can be taken into account by the term

$$\mathcal{T}_{DC} = \sum_{e=1}^{(n_{el})_n} \int_{Q_n^e} \tau_{DC} \cdot \frac{\partial W^h}{\partial x_i} \cdot \frac{\partial U^h}{\partial x_i} \, dQ \,, \tag{2.5}$$

where τ_{DC} is a scalar coefficient. Adding this term to the weak form has the effect of damping the potential oscillations in the presence of high gradients which is consequently known as "discontinuity capturing". Another achievement of this method is the prevention of so called under- and overshootings of the solution in approach or close after the discontinuity respectively. As the discretization of the physical space obviously always leads to discontinuities, this stabilization is of vanishing importance using a discretization with a potentially infinitely large resolution. This is an important fact that must be considered when evaluating the final results given in section 3.

Adding all of the terms above to the previous weak formulation, one arrives at the final problem formulation implemented in the solver.

Given $(U^h)_n^-$ at a time step, find $U^h \in (S_U^h)_n$ such that $\forall\, W^h \in (\mathcal{V}_U^h)_n$

$$\int_{Q_n} W^h \cdot \left(\frac{\partial U^h}{\partial t} + A_i^h \frac{\partial U^h}{\partial x_i} \right) dQ + \mathcal{T}_{jump} + \mathcal{T}_{SUPG} + \mathcal{T}_{shock} = 0 \,, \tag{2.6}$$

where $(S_U^h)_n$ and $(\mathcal{V}_U^h)_n$ denote the trial solution and test function spaces respectively.

2.1.3 Solution Strategies

As mentioned before, equation 2.6 will be solved using the finite element method. This involves a discretization using so called shape functions leading to a system of equations. Thorough explanations and derivations concerning the discretization by means of finite elements is found in common text books about this topic (e.g. [1]) and is not discussed here. However, regardless of the applied discretization, a suitable solution method for solving the given *nonlinear* equation is needed. Here, the Newton-Raphson method is the usual choice which is motivated by a linearization of the form

$$f(\bar{U} + \Delta U) = f(\bar{U}) + \left. \frac{df}{dU} \right|_{\bar{U}} \cdot \Delta U \,, \tag{2.7}$$

where f is the residual described by equation 2.6 and $\left. \frac{df}{dU} \right|_{\bar{U}}$ denotes the Jacobian of f at $\bar{U}$. Setting this equation to zero and reorganizing the terms, one obtains a linear system of

equations

$$K \cdot \Delta U = -f, \tag{2.8}$$

which is solved in every Newton step till a suitable threshold of convergence in f is achieved. $K = \frac{df}{dU}\big|_{\bar{U}}$ is the system's stiffness matrix calculated element-wise and, depending on the algorithm, assembled afterwards to a global stiffness matrix. More details and further descriptions of the necessary calculations and the derived weak form above can be taken from the technical notes on the compressible flow solver of XNS published on the official website of CATS.

2.2 Applying Boundary Conditions

Adding a method for prescribing pressure boundary values involves a respective adjustment of the solver implementation including potential changes in the weak form and thereby the residual and its Jacobian. The necessary extensions for each tested method are discussed below.

2.2.1 Prescribing the Total Energy per Unit Volume (Method A)

Each conservation variable is independent of an explicit expression of p. However, the total energy per unit volume (ρe) is part of the conservation variable formulation and has a direct relation to p. In general, let p_{out} denote the pressure which is supposed to be set at an arbitrary point in the mesh. Following the statement above, we can define the pressure as

$$p_{out} = p_{out}((\rho e)) \,, \tag{2.9}$$

which, by means of the definition of the total internal energy, leads to prescribing $(\rho e)_{out}$ as a Dirichlet boundary condition by using the expression

$$(\rho e)_{out} = \frac{p_{out}}{(\gamma - 1)} + \frac{1}{2}\frac{\sum_{i=1}^{n_{sd}} (\rho u_i)^{*2}}{\rho^*} \,, \tag{2.10}$$

where $(\rho e)_{out}$ depends on the desired pressure p_{out} and the remaining current values of the states at the respective nodes super-scripted with "$*$". $n_{sd} = 3$ is the number of spatial dimensions in the general three dimensional case and γ denotes the ratio of specific heats of the fluid.

Now this relation is utilized for prescribing the internal energy and thereby the pressure at for instance the outlet boundary of the domain by replacing p_{out} by the desired boundary pressure value but inserting the actual current values of ρ and (ρu_i) at each point of the

associated boundary. The "reset" of the pressure is invoked after each Newton iteration step such that the pressure boundary conditions are incorporated in the next solution step.

In the consecutive discussions, this method is denoted by *Method A*.

2.2.2 Adapting the Euler Fluxes (Method B)

Another method involves introducing the prescribed pressure values as part of the Euler fluxes projected to the corresponding boundary normal. These are generally defined as

$$F_i = \begin{bmatrix} \rho u_i \\ \rho u_i\, u_1 + \delta_{i1} p \\ \rho u_i\, u_2 + \delta_{i2} p \\ \rho u_i\, u_3 + \delta_{i3} p \\ (\rho e + p)\, u_i \end{bmatrix}, \tag{2.11}$$

where p in this expression is replaced by the respective boundary value p_{out}. Denoting the prescribed boundary flux by $F_i(p_{out})$, the difference

$$l = n \cdot F_i(p_{out}) - n \cdot F_i = \begin{bmatrix} 0 \\ \delta_{i1}(p_{out} - p^*) \\ \delta_{i2}(p_{out} - p^*) \\ \delta_{i3}(p_{out} - p^*) \\ u_i(p_{out} - p^*) \end{bmatrix} \tag{2.12}$$

must vanish. The weak form of this expression is then formulated as

$$\int_{(P^h)_p} W^h \cdot l^h \, dP = 0\,, \tag{2.13}$$

where the index p indicates a "pressure boundary". This expression is placed on the left hand side of equation 2.6. The proceeding of solving this extended weak form is analogous to the previously described steps in section 2.1.3.

In the consecutive discussions, this method is denoted by *Method B*.

2.2.3 Riemann Problem on Boundaries (Method C)

The next and last method is based on the diagonalization of the Euler equations in 1D, the derivation of which is given in [1]. The result of this decomposition is expressed by the so

called *Riemann variables*

$$W = \left(s,\, v + \frac{2}{\gamma - 1},\, v - \frac{2}{\gamma - 1} \right)^T . \tag{2.14}$$

At a domain boundary, the corresponding eigenvalues of 2.14

$$\Lambda = diag(v,\, v + c,\, v - c)\,, \tag{2.15}$$

represent information about whether a boundary condition is imposed or not. To be precise "as many conditions as there are negative eigenvalues" [1] are prescribed at the boundary. Note that c is the speed of sound in the given notation of the eigenvalues.

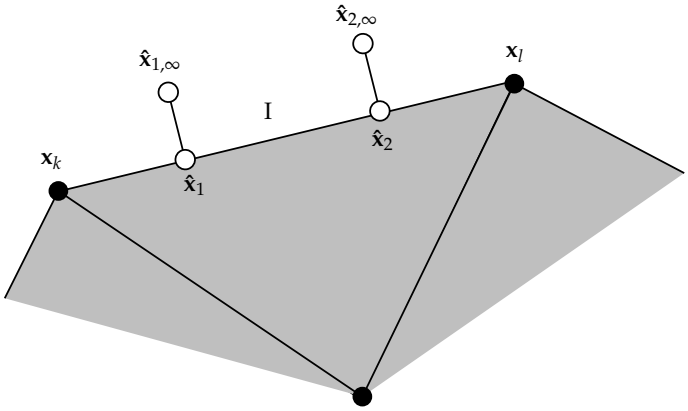

Figure 2.1: Ghost Points at corresponding quadrature points (Source: [3])

This theory is further utilized and developed by [3], where so called *ghost points* extend the boundary at the quadrature points as can be seen in Figure 2.1. Now, depending on the eigenvalues, a conclusion can be made whether the Riemann variables are determined based on the the interior, which corresponds to $\lambda_d^* > 0$, or the boundary conditions.

The respective ghost state W_∞ is transformed to the conservation variables U_∞ from which the resulting Euler fluxes are calculated. Consequently, this means that a *Riemann Problem* must be solved at the boundary using a suitable solver. Such a problem is generally defined by looking for a piecewise constant solution, given that a single discontinuity is imposed on the data. The "data" is represented by the states at the boundary in this case.

Similar to *Method B*, the resulting fluxes are incorporated in the defined difference

$$l = n \cdot F_i(U^*, U_\infty) - n \cdot F_i(U^*) \tag{2.16}$$

which has to vanish as well. By means of the formulation

$$\int_{(P^h)_p} W^h \cdot l^h \, dP = 0 \,, \tag{2.17}$$

the given expression extends the left hand side of the weak form the same way as *Method B*. This term obviously vanishes in case the ghost state W_∞ is determined by the current flow state in the interior.

A more detailed study of the underlying mathematics can be found in the previously mentioned thesis. Alternatively, the technical notes on the CATS website contain more conclusive details on this method. In the consecutive discussions, this method is denoted by *Method C*.

3 Test Case Definitions

In order to adequately test the implementations and the quality of the respective methods of applying the pressure boundary conditions, three distinct test cases are defined. These will pose a basis of an evaluation of the applied methods which is thoroughly done in section 4. Following the section of numerical results from "Characteristics Finite Element Methods in Computational Fluid Dynamics" [4], the following test cases are recreated in the developed compressible flow solver. However, this is only used as a reference solution as it is formulated as a one-dimensional problem, whereas the tests documented here were done on a two-dimensional grid.

3.1 Tested Geometries

A two-dimensional representation of converging-diverging nozzle, also known as a "Laval Nozzle", is chosen as the basic and only geometry used in the given piece of work. Test cases for generating the most important phenomena are easily defined and evaluated. The created nozzle model is based on the geometries defined in [4] which leads to a model which is qualitatively represented in Figure 3.1. All the presented flow conditions below and their respective boundary values are applied to this geometry. Furthermore it is taken advantage of the nozzle's symmetry along its central horizontal axis allowing a reduction of the model. This has the consequence, as calculations are done in 2D, that the computational effort is half of what computations on the entire geometry would cause.

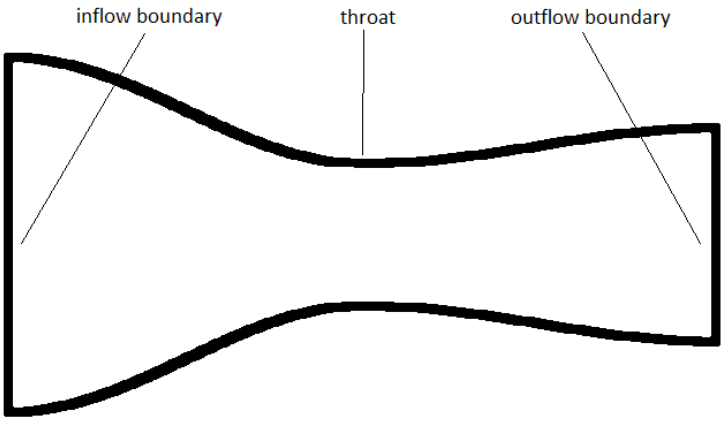

Figure 3.1: Two-dimensional Outline of the Test Geometry

3.2 Tested Flow Conditions

3.2.1 Supersonic

In this case, no outlet pressure boundary conditions are prescribed and thereby, caused by the prescribed inflow conditions, admits the development of a *supersonic* flow (See Figure 3.2). This test case is not suitable for testing any pressure boundary condition method as none is prescribed but rather serves as a reference for the general correct functioning of the underlying solver. Comparisons in the convergence behavior of this test case and the ones below are treated in the results section.

The corresponding boundary values are listed in table 3.1. Note that the inflow boundary conditions are the same for each test case presented here.

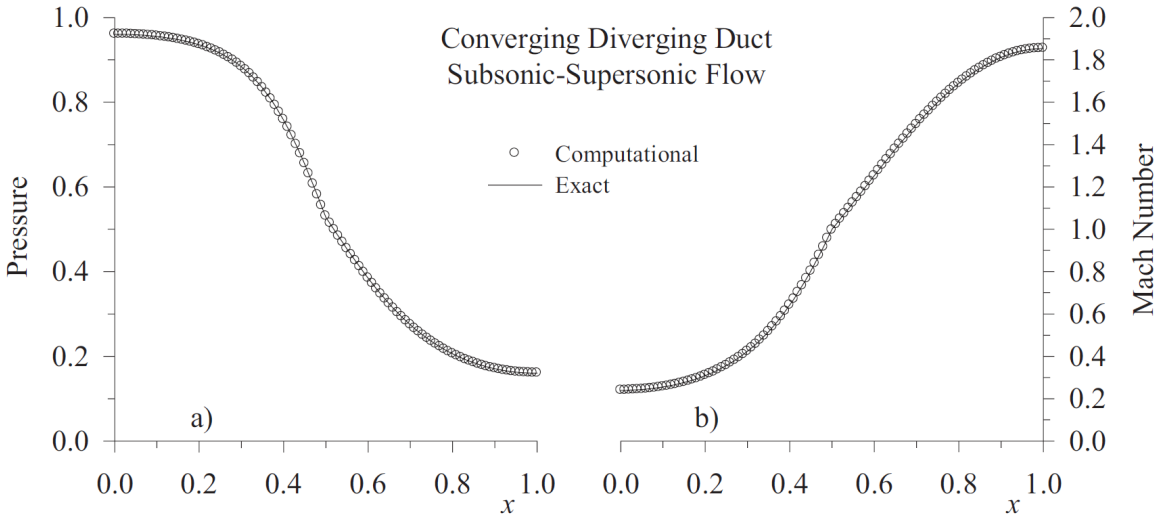

Figure 3.2: Reference solution for the *supersonic* nozzle flow test case (Source: [4])

3.2.2 Critical

Critical flow, also known as "choked" flow, describes the compressible flow phenomenon of the decreasing pressure towards the throat of the nozzle against a higher pressure in the environment, i.e. the outlet boundary of the domain. Within the throat, which is the nozzle position with the minimal cross-sectional area, a sonic state is reached which then declines towards the increasing pressure at the outlet. As this test case leads to a "shock of vanishing strength", [4], it poses a challenge to both the correct implementation of a boundary condition method and the overall discontinuity capturing and stabilization

capabilities of XNS. Visible through the kink in the plotted pressure field, this condition can be examined in the reference solution in Figure 3.3.

For more detailed reference purposes, table 3.1 lists the prescribed boundary values set in XNS for this test case and the other two.

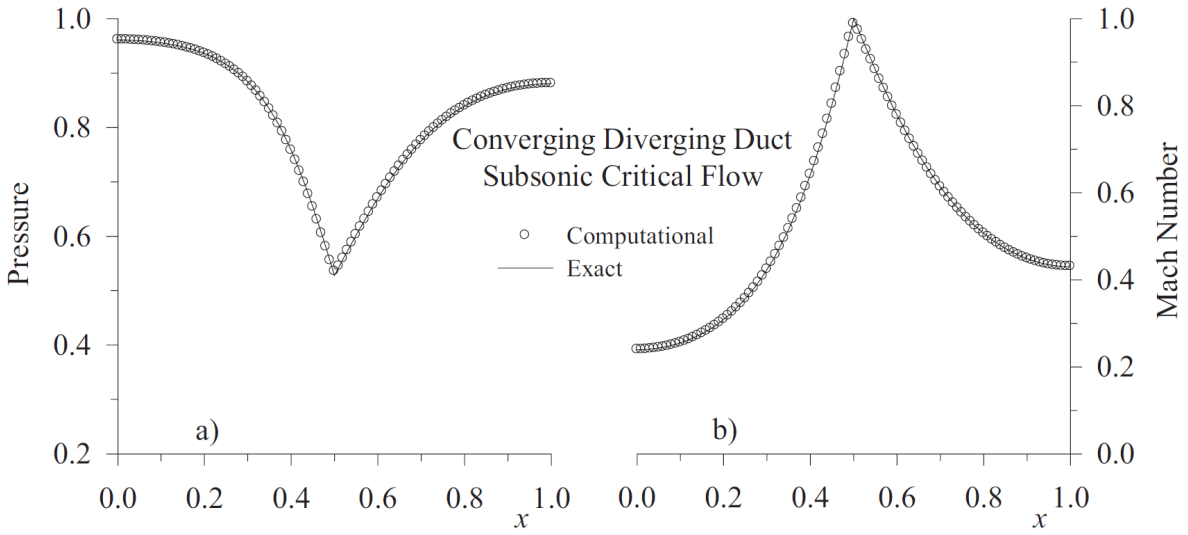

Figure 3.3: Reference solution for the *critical* nozzle flow test case (Source: [4])

3.2.3 Shocked

The next test case is similar to the *critical* test case but involves the generation of a so called "shock". Like in the previous test case, a pressure on the outlet boundary is prescribed which leads to an increase in the pressure after passing the nozzle throat. Said pressure, however, is configured in such a way that this expansion is spatially "delayed" and leads to the aforementioned shock (see Figure 3.4) in order to reach equilibrium to the pressure at the outlet boundary. It should be noted that the prescribed pressure on the outflow boundary is less than for the *critical* test case. As in the previous case, this abrupt increase in the pressure is not a trivial task for the *compressible* flow solver and makes it thereby especially interesting to test in combination with the different invoked pressure boundary value methods.

The exact boundary values, which are equal to the *critical* and the *supersonic* case at the inflow boundary, can be viewed in table 3.1.

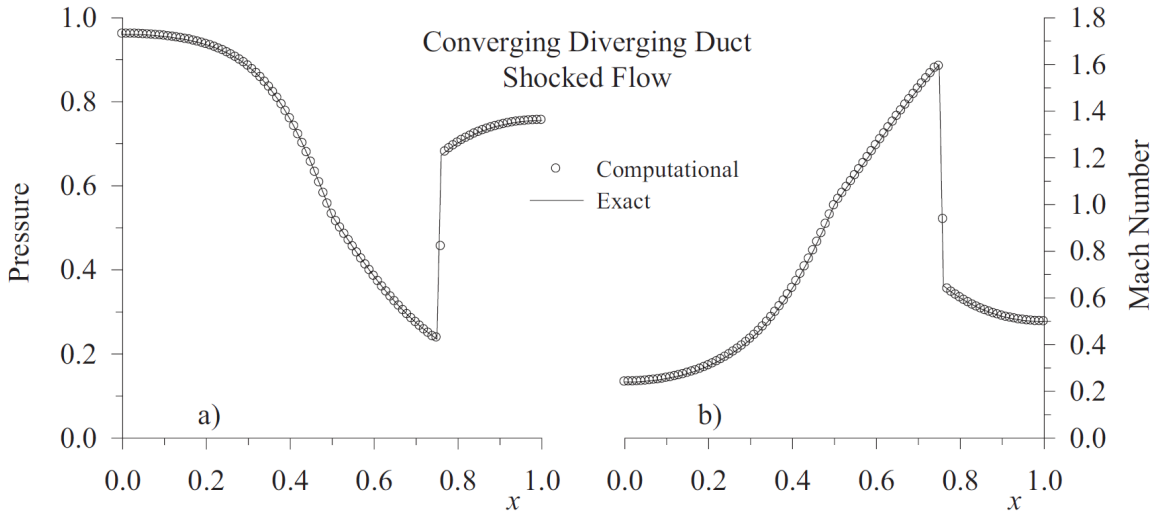

Figure 3.4: Reference solution for the *shocked* nozzle flow test case (Source: [4])

Physical Quantity	Inflow	Outflow:	Supersonic	Critical	Shocked
ρ	0.9720		-	-	-
ρe	2.4410		-	-	-
p	-		-	0.8810	0.7560

Table 3.1: Boundary conditions for all defined flow test cases

3.3 Simulation Setup

XNS works on input files setting the type of simulation, paths to additional data files, time stepping, solver configurations, boundary conditions and initial conditions. Usually, boundary conditions are set via so called expressions, passing the corresponding boundary and degree of freedom as arguments. For instance the density of 0.972 for the inflow boundary (value: 1) of the shocked test case is set via the line "`rngdexp 1 1 0.972`". Setting the outflow pressure in method *A* for this test case is done similarly except replacing the exact value of 0.972 with the expression "`0.756/0.4+1/(2*r)*(ru*ru+rv*rv+rw*rw)`" where 0.756 is the prescribed pressure on the outflow boundary (value: 3). For methods *B* and *C* another kind of expression is implemented enabling to "directly" set the pressure to a numerical value. Again following the same test case, in both cases this expression is "`rngdpres 3 0.756`". Of course, the underlying implementation of processing this prescription differs in both methods. The initial conditions are set via linear interpolations of the precalculated or prescribed values between the in- and outflow boundaries for the *shocked* and the *supersonic* test cases and a uniform flow across the entire domain for the *critical*

test case. An exemplary XNS input file is given in appendix A.

The simulation time within XNS is also defined in the input file by defining the number of time steps `nts` and the time step size `dt`. 1000 time steps and a time step size of $dt = 0.01$ are the standard configuration of each test run. Following the configurations described above, this results in a time of 10 seconds for every executed simulation.

4 Results and Discussion

Each method for prescribing pressure boundary conditions as they are described in section 2.2 was applied in multiple simulations in accordance to the test cases presented in section 3. The following chapter aims at revealing the differences between the results and thereby the characteristic of the methods. These characteristics are illustrated by a plot of the pressure along the center of the nozzle (which is the lower boundary in the grid in Figure 4.1) as well as the evolution of the solution over time at a chosen sampling node (the red dot in the aforementioned figure). The sampling node is chosen this way as it is desirable to properly capture the sensitivity of the solver's shock capturing capabilities for the "shock test case" while applying the different methods. Given by the virtue of the corresponding test case, this node lies "behind" the shock. The discussion of these plots is followed by a conclusion, attempting to analyze the won insights in order to potentially develop ideas for improvement.

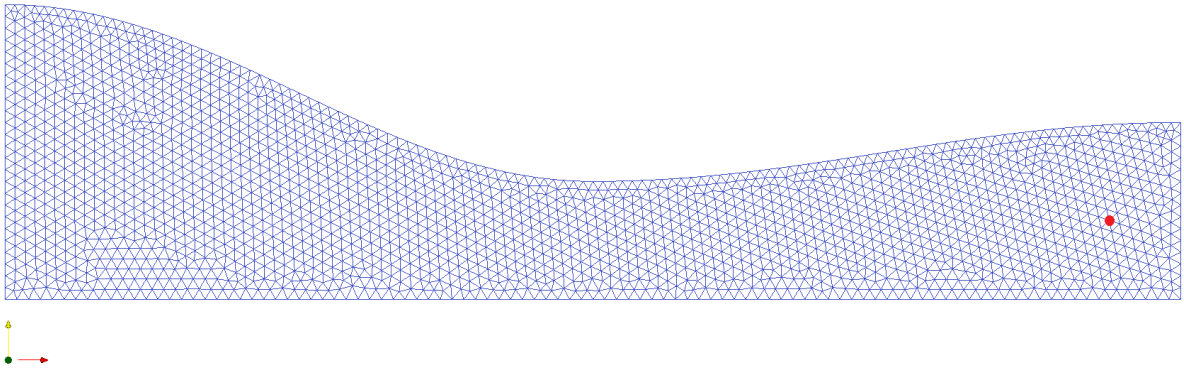

Figure 4.1: Nozzle Grid and Sampling Node

4.1 Solution Visualizations

In the following analysis it shall be discussed how well the numerical solutions in XNS approximate the reference solutions for the respective test cases. The reference solutions, analogous to [4], correspond to the one-dimensional solutions of the isentropic flow relation which is an expression relating the Mach number, the cross-sectional area and the ratio of specific heats. For each test case and tested method, the same time stepping and number of time steps was set in order to obtain a qualitative comparison between the achieved approximations after a certain time.

4.1.1 Supersonic Nozzle Flow

In order to avoid the wrong conclusion that the difference between the reference solution and our solution in the results solely depends on the implementation of the pressure boundary condition methods and the resolution of shocks, the general functionality of the XNS solver for *compressible* flows is tested by means of the *supersonic* flow test case. In this test case there is no prescribed pressure on the outflow boundary leading to *supersonic* flow conditions towards the outflow boundary following the passing of the nozzle throat. This also means that none of the implemented boundary condition methods is supposed to affect the generated results and thus serves as a reference for comparing the errors in the solution. As expected and verifiable in Figure 4.2, the solutions for testing the different implementations are identical.

Additional simulations of this test case, the results of which are illustrated in Figure 4.3, prove that the difference to the reference solution does not differ significantly when increasing the simulation time from 10 to 100 seconds. In the given plots, it is taken into account that no boundary value method is invoked for the *supersonic* test case. Therefore the different stages of the numerical results are clearly distinguishable.

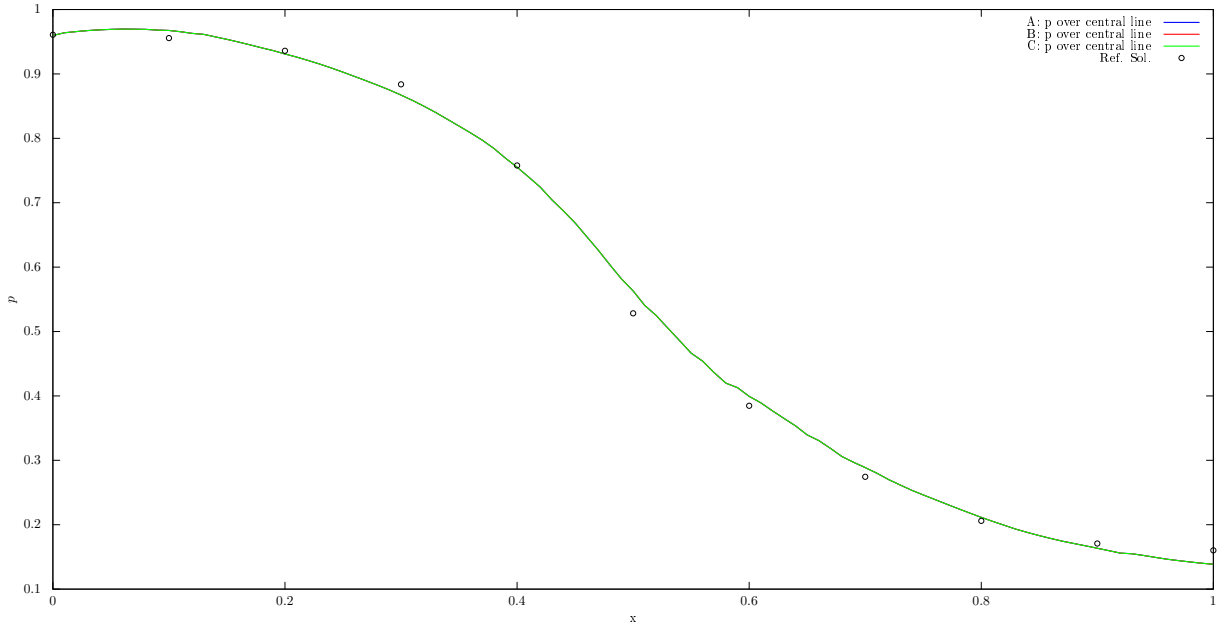

Figure 4.2: Pressure over central line, *supersonic* nozzle flow, methods A, B, C ($\Delta t = 0.01$)

4.1.2 Critical Nozzle Flow

The critical flow condition is chosen as it is to be expected that the resolution of the kink in the pressure field (see Figure 3.3) in combination with the prescribed pressure at the

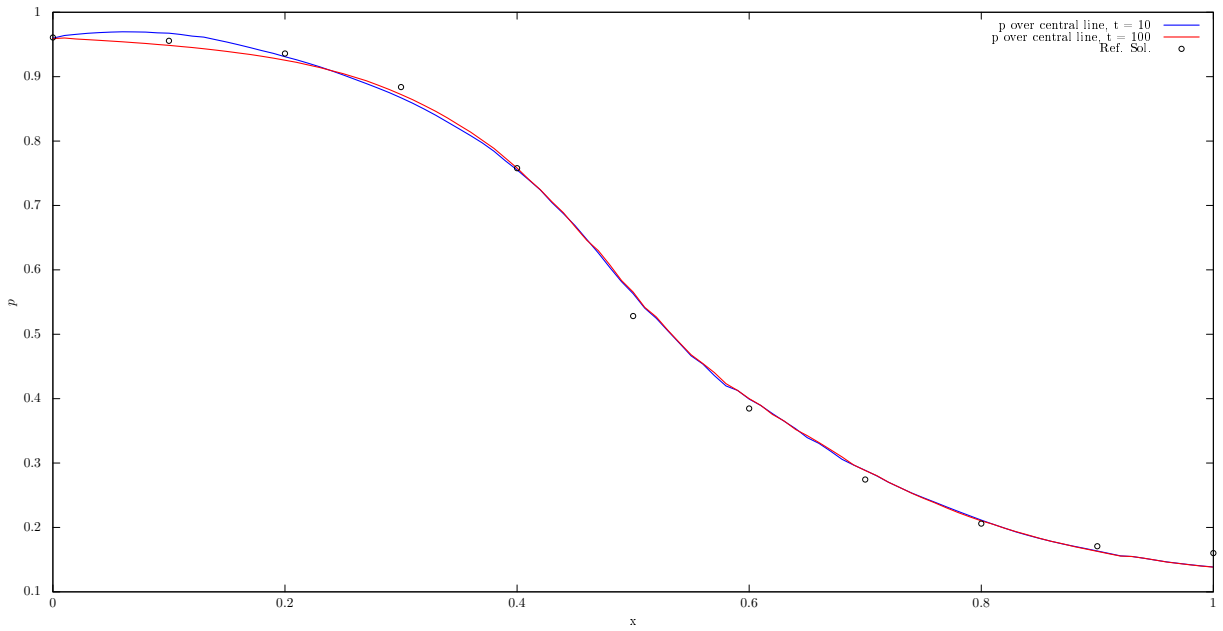

Figure 4.3: Pressure over central line, *supersonic* nozzle flow ($\Delta t = 0.01, 0.1$)

outflow boundary presents a challenging task to the solver. Considering Figure 4.4, one concludes that this assumption is justified as the difference between the reference solution and our solution close to this position is considerably greater than in the remaining plot. It must be noted, however, that this shortcoming applies to each of the tested methods without any significant difference at any of the calculated nodal values.

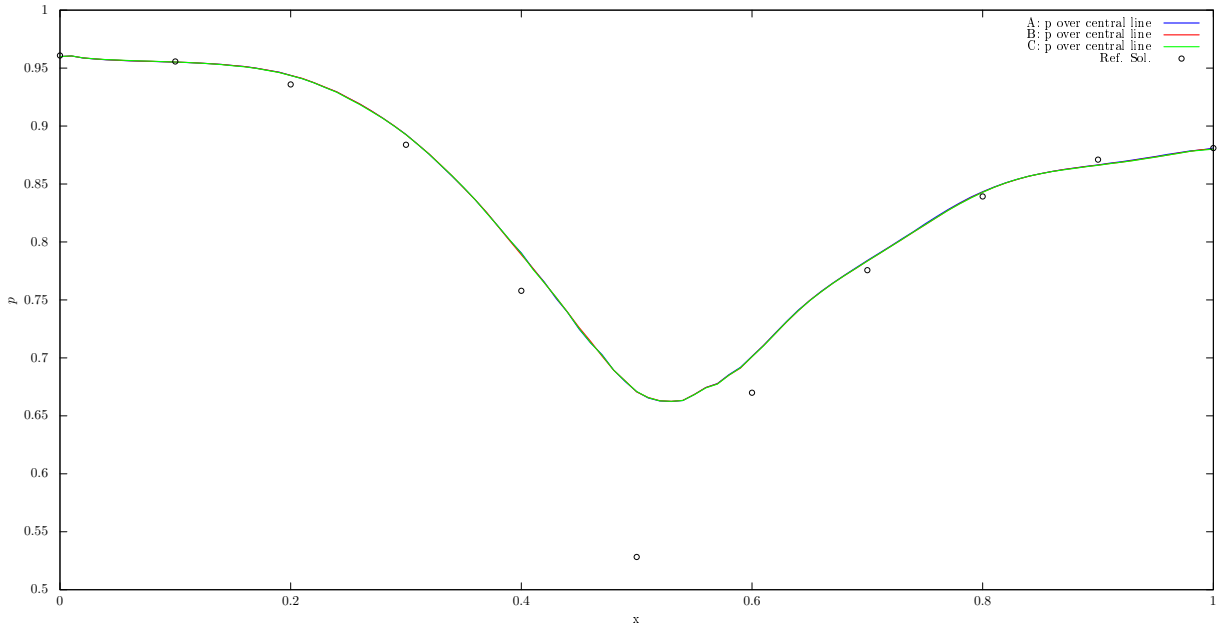

Figure 4.4: Pressure over central line for a *critical* nozzle flow ($\Delta t = 0.01$)

4.1.3 Shocked Nozzle Flow

This is further investigated by means of the previously presented test case in which the pressure at the outflow boundary is chosen in a such a way that a shock is induced at approximately $x = 0.75$. In the referenced test case in [4] it is also stated that *at most* one node must be included in the generated shock which, assuming that a similar grid resolution is chosen, should at least approximately be met by the generated results.

As can be seen in Figure 4.5, the calculated solution qualitatively follows the reference solution of the expected shock. The applied discontinuity capturing works to a satisfactory degree as neither overshooting and undershooting nor oscillations are present close to the shock. It is also obvious that an non-matching (compared to the reference solution by [4]) number of nodes is captured within the shock. However, this issue can be easily ignored and only serves as a reference consideration as this behavior strongly depends on the chosen grid resolution and, as such, is resolved when increasing it.

Similar to the critical flow case, the results and the overall error do not significantly differ from each other. Increasing differences occur only in the middle section of the domain in approach to the shock at $x = 0.75$. Here, the solution of method B diverges the most from the "main course" of the condensed plot without any improvement on the error compared to the other methods. This, at least, signals some influence on the resolution of the shock in the solver residing from the applied methods. In the passing of the shock and leading to the outflow boundary, all methods again deliver almost identical results as is to be expected so close to a prescribed boundary value.

In a subsequent test run with a time stepping of $dt = 0.1$ and 1000 time steps, thereby resulting in a simulation time of 100 seconds, no significant improvements on the solution are observed. However, further considering the respective plots in Figure 4.6, leads to conclusions about the robustness of greater time steppings of the implemented methods. While the solution of method C diverges severely in approach to the shock, methods A and B show no such behavior. Additional test runs, which are not represented by a figure, show that, for the given mesh, convergence is achieved for time steps of approximately 0.5 when applying A or B. Method C still converges for time steps of 0.1 but leads to the aforementioned non-sensible result.

4.2 Convergence Behavior

Aiming at gaining more insights on the functioning of the *compressible* flow solver in XNS, the evolution of a single nodal value is observed over time for each boundary value method.

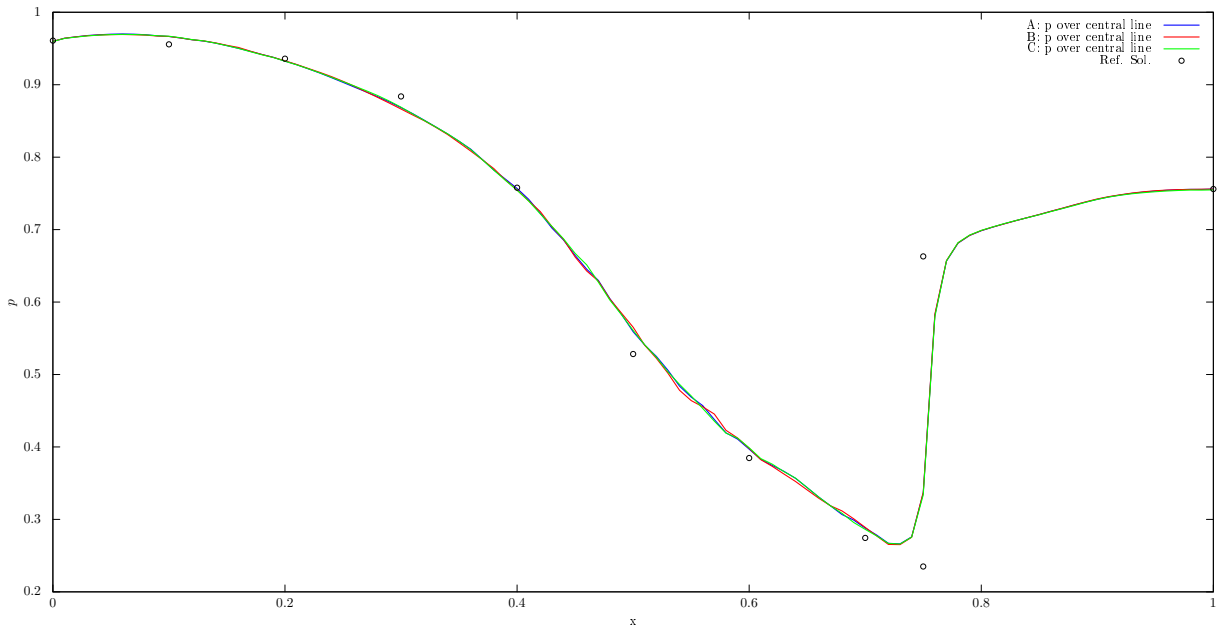

Figure 4.5: Pressure over central line for a *shocked* nozzle flow ($\Delta t = 0.01$)

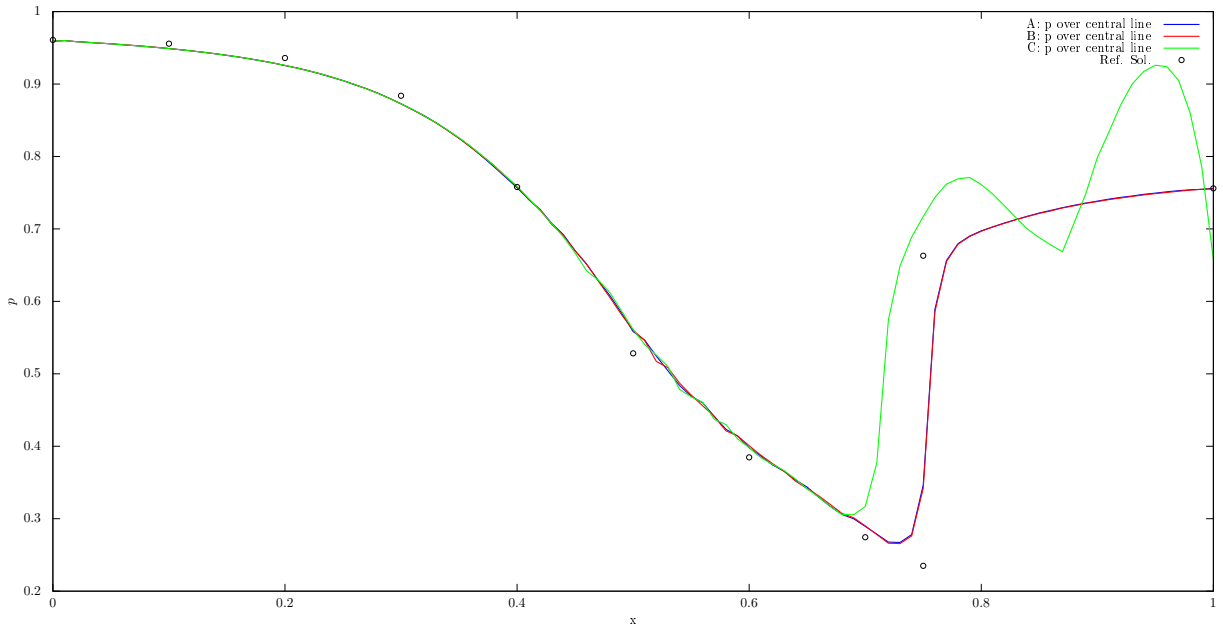

Figure 4.6: Pressure over central line, *shocked* nozzle flow ($\Delta t = 0.1$)

This is supposed to deliver information on the convergence behavior of the solver when invoking said methods. As mentioned above, this sampling node lies close to but *after* the shock in the *shocked* test case.

4.2.1 Supersonic Nozzle Flow

Comparable to the approach in section 4.1, the *supersonic* test case is used as a reference in order to check the overall functionality of the compressible flow solver measured by the observations of the *critical* test case. For this particular case, as is verifiable in Figure 4.7, convergence is achieved after around 120 time steps followed by potentially insignificant oscillations about a value of approximately 0.15. Without any arising shocks and none of the boundary value methods invoked, the *compressible* flow solver works as expected.

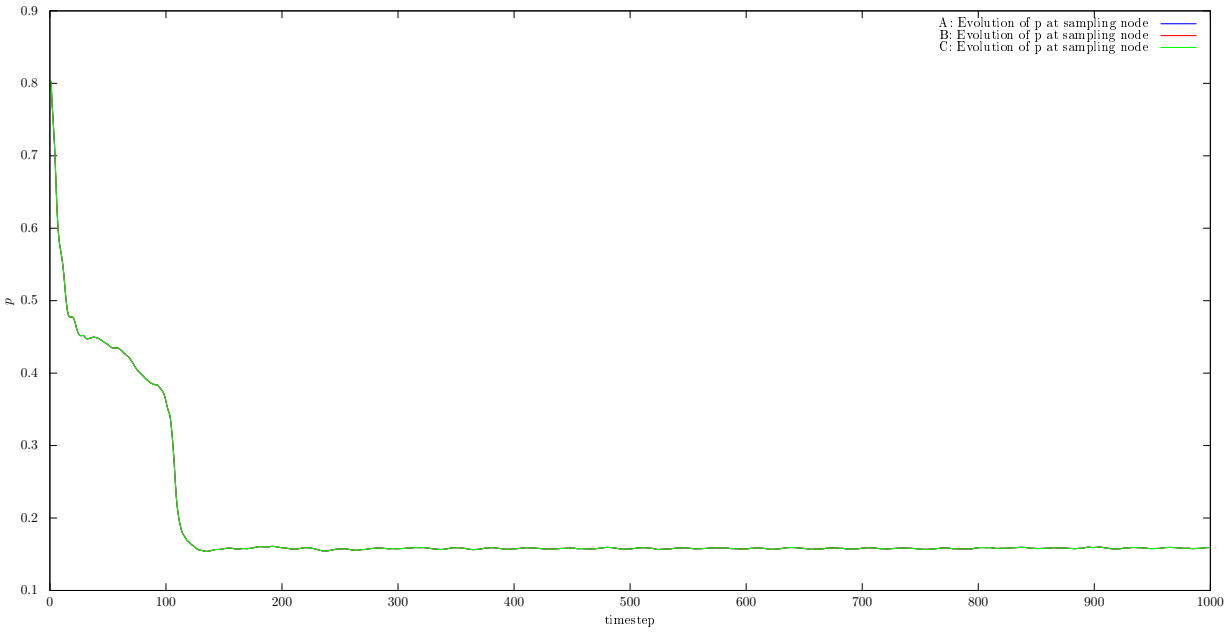

Figure 4.7: Solution evolution of *supersonic* nozzle flow at sampling point for methods A, B and C

4.2.2 Critical Nozzle Flow

The respective pressure value of the sampling node at each time step of the *critical* test case is plotted in Figure 4.8 for the methods A, B and C. Across the entire course of the plot, the solution oscillates extensively independent of the invoked method. Given by the virtue of the plot, it is reasonable to assume that the approximate solution of the sampling node is close to 0.875. Now assuming this value to be the exact solution, it is very closely approximated at around the 35th time step. Even though this solution is closely maintained in the next few time steps, a huge increase is observed ranging through the 50th followed by a decrease of roughly the same amplitude through around the 90th time step. Followed by some smaller oscillation the next few time steps, the solution again assumes a value of about 0.87 which is at least close to the assumed exact solution. The remaining course of

the plot shows the same characteristics while a similar accurate solution is only obtained at around the 300th time step. Even though the amplitudes of the oscillations only slightly decrease over time, all in all, a stable approximation is maintained by the solver which might be further improved by increasing the resolution of the grid.

This behavior is observed for each pressure boundary value method. Each method's time line looks exactly the same with the exception of minor differences at some of the peaks.

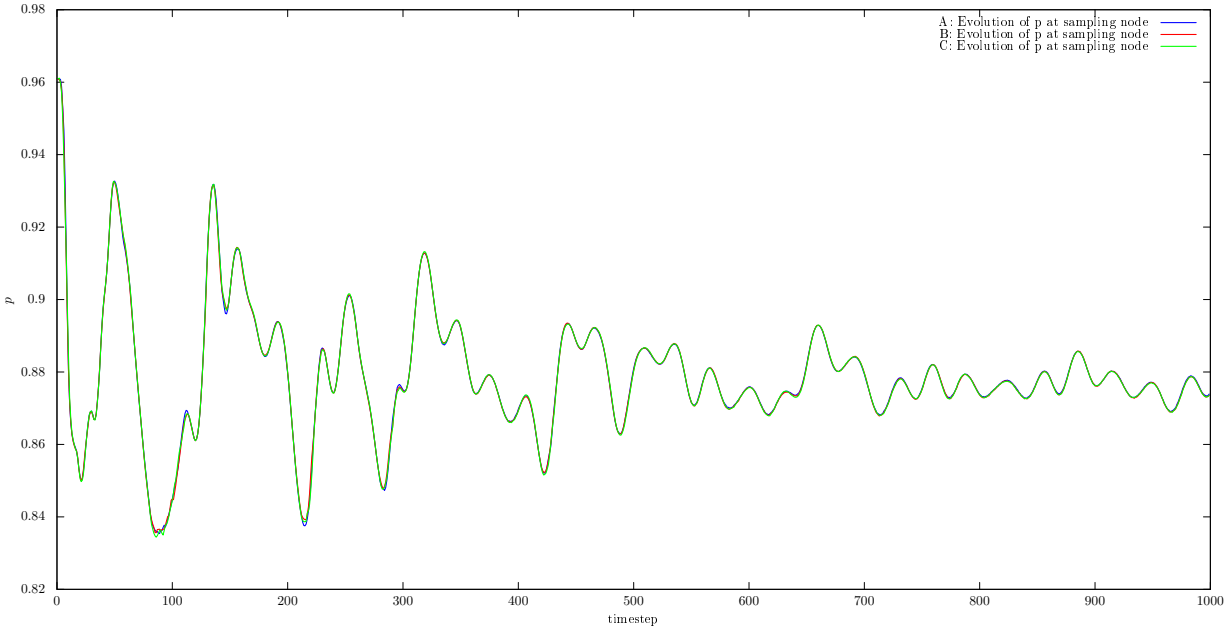

Figure 4.8: Solution evolution of *critical* nozzle flow at sampling point for methods A, B and C

4.2.3 Shocked Nozzle Flow

Analyzing the *shocked* test case in the same manner delivers a similar picture as the *critical* case but surprisingly, when considering Figure 4.9, the solution seems to stabilize significantly faster compared to the *critical* test case. The residual oscillations, however, remain approximately constant starting at about 200 time steps. Another similarity to the *critical* test case is given by the fact that the solution significantly improves between some time steps but then "swings" out again. This effect is represented by the oscillations with great amplitudes at the time steps 25, 50, 80 and 120.

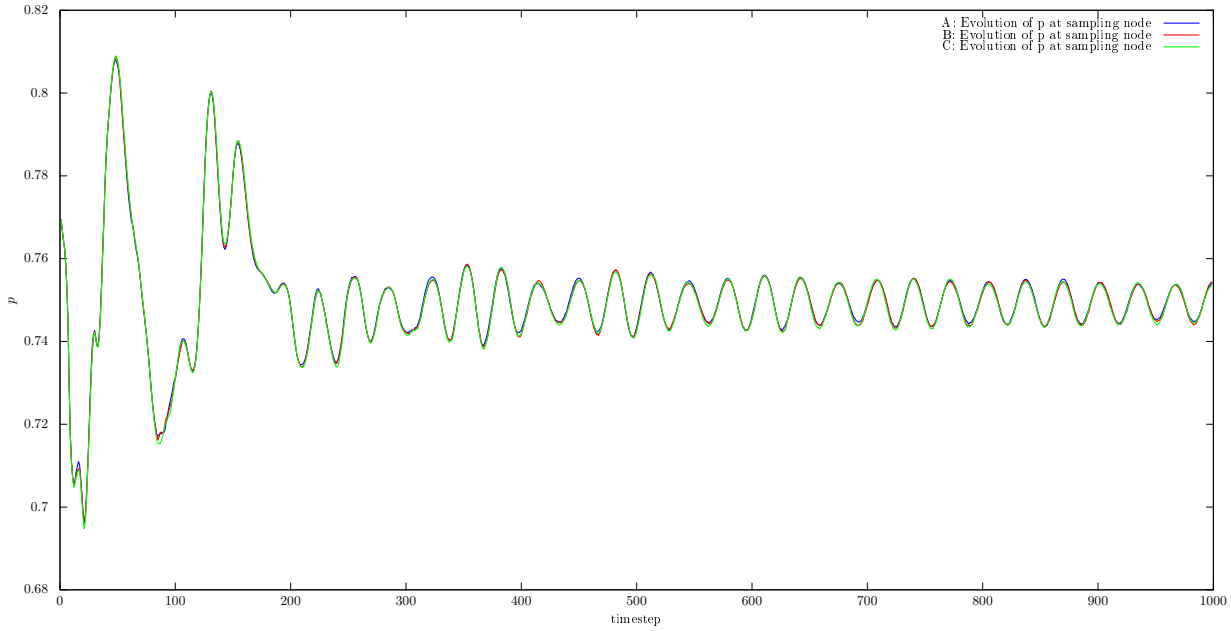

Figure 4.9: Solution evolution of *shocked* nozzle flow at sampling point for methods A, B and C

4.3 Conclusion

Without any view at the source code of the implemented pressure boundary value methods nor specifics on the solvers, this section attempts to draw conclusions from the observations made in the sections above in order to derive potentials for improvements and optimizations as well in the implementations of these methods as the underlying solver itself. Furthermore, as the primary focus of this document, a final statement about the observable differences between the methods is crafted in order to determine the best suitable method for each test case and overall.

In regard to the *shocked* test case in section 4.1.3 and the test case of the critical flow being close to the threshold of also generating a shock, it is obvious that the implemented discontinuity capturing in XNS performs satisfactory for the present test cases. However, the accuracy of the approximation has potential for improvement. A first assumption about the origin of this issue might be the implementation of insufficient stabilization. A look at the result for method C in Figure 4.6, although methods A and B work fine, partially confirms this assumption while it is difficult to rule out the impact of an error-prone implementation of the boundary value methods. This hypothesis is substantiated by the fact that even without invoking any of these methods, associated to the *supersonic* test case, a satisfactory accurate approximation to the corresponding exact solution can not be achieved (see Figure 4.3). Simultaneously, and contradicting to this exact conclusion,

Figure 4.7 shows a stable solution at the chosen sampling point after around 125 time steps. However, it remains reasonable to assume that this is not representative as on the one hand the approximation error is comparatively small at single points, as e.g. $x = 0.8$, which is close to the position of the chosen sampling point, and on the other hand Figure 4.8 clearly illustrates an immensely oscillating solution potentially caused by inadequate stabilization.

Apart from the factor of sufficient stabilization, the assumption of an inviscid flow as well as the choice of the grid must be discussed. The description of the test cases already hinted at the fact that resolving spatial discontinuities in the solution is of major dependence on the chosen grid resolution. Generally, in case the solver's consistency holds, the approximate solution on a grid of infinite resolution is equal to the exact solution. While this is a trivial statement, one can draw additional conclusions about the reasons behind the low accuracy of the given numerical solutions by computing identical test cases on a finer grid. In case an improvement arises form this investigations, the made statements on the stabilization in the paragraph above must be revisited.

Fortunately, as of the writing of this document, additional test cases independent of this work were executed. These showed that major improvements on the convergence behavior can be observed when adding viscosity properties to the flow as the lack of physical diffusivity leads to at least some of the issues illustrated and discussed above. As more detailed investigations of these issues are not sensible in the course of this work without any actual testing and therefore potentially unreasonably inaccurate, the remainder of this section solely refers to the comparison of the three pressure boundary value methods.

The following analysis is based on what can be derived from the given figures above and is extended by additional test runs which are not represented by their own plots. With the exception of method C in the *shocked* test case with a time stepping of $dt = 0.1$, compared to the otherwise smaller time stepping of $dt = 0.01$, there is no exceptionally pronounced characteristic in either of the methods. The generated results when invoking method A or B are approximately the same in the sense of how closely the approximation to the exact solution is, completely independent of the time stepping configurations. Slight differences, if any, are probably due to propagation of rounding errors through the different implementations. As a consequence, at least from the given results above, this leads to the final conclusion that there is no practical advantage of choosing B over A in XNS. A similar verdict is inevitable for C when choosing proper time stepping. Otherwise, C even produces non-sensible results.

The analysis of the executed tests do not favor any of the presented pressure boundary value methods over the other. However, due to their great similarity in terms of results, it is reasonable to assume that their implementation is correct. A more meaningful comparison might arise when applying the aforementioned changes as the grid resolution change.

A XNS Example Input Files

Listing A.1: XNS Input File, Method A, Shocked

```
title NozzleIannelliShocked

# Governing equations
compressible
steady off
space-time on

source /home/mk/CATS/My_Tests/Input/minf
mien /home/mk/CATS/My_Tests/Input/mien
mrng /home/mk/CATS/My_Tests/Input/mrng
mxyz /home/mk/CATS/My_Tests/Input/mxyz
mprm /home/mk/CATS/My_Tests/Input/mprm

nts 1000
dt 0.01
ntsbout 1

## Solver
# Preconditioner
rprecond ilut 100 1e-4
nprml rcm
# GMRES loop for fluid
ninner 80
nouter 4
epsilon 2 1e-8
# Newton-Raphson
nit 5
epsilon 1 1e-30

## Stabilization parameters
alpha 0.5
tau_dc_factor 1.0
tau_momentum_factor 1.0

## Physics
# (mu_0 = 0 makes the flow inviscid)
material 1 viscosity 0.0
# scales kappa / c_v
material 1 kappacv 0.0
```

```
## Boundary conditions (Here: Shocked)
#
# Inflow
rngdset 1 1 0 0 1
# rho:
rngdexp 1 1 0.972
# (rho e) = E:
rngdexp 1 4 2.441
# Nozzle walls
slip 2
slip 4
#
# Outflow
# "p" (Method A):
rngdexp 3 4 0.756/0.4+1/(2*r)*(ru*ru+rv*rv+rw*rw)


## Initial conditions
# Linear variation between inlet
# and outlet values
#
# rho:
initexp 1 0.972-x*(0.972-0.796)
#
# m:
initexp 2 0.274+x*(0.457-0.274)
#
initexp 3 0.0
#
# (rho e) = E: Uniform Initial Flow
initexp 4 2.441-x*(2.441-2.021)
```

Listing A.1: XNS Input File, Method A, Shocked

Bibliography

[1] Donea (2001). *Finite Element Method for Flow Problems*. John Wiley & Sons.

[2] Drela, P. M. (2006). Fluid mechanics. Online Lecture Notes.

[3] Gurris, D.-M. M. (2009). *Implicit Finite Element Schemes for Compressible Gas and Particle-Laden Gas Flows*. PhD thesis, TU Dortmund.

[4] Iannelli, J. (2006). *Characteristics finite element methods in computational fluid dynamics*. Springer.