Cartesian Low-Dissipation-and-Dispersion Schemes Combined with a Cut-Cell Method, Evaluated with Richardson Extrapolation

Master's Thesis

D.N. Vedder
Stadhoudershof 31
3481 HV Harmelen
The Netherlands

Graduation Committee
Prof. P. Wesseling
Dr. C. Vuik
Dr. W.T. van Horssen
Delft University of Technology

Prof. W. Shyy
University of Michigan

Contents

Abstract

Computational Aeroacoustics deals with the disciplines Computational Fluid Dynamics and Aeroacoustics. Acoustic propagation problems are governed by the linearized Euler equations. The properties of acoustic waves are encoded in the dispersion relation of these equations. Optimized high order numerical schemes are presented in this report. These schemes optimize the dispersion and dissipation errors by preserving the dispersion relation and are presented in the finite difference and the finite volume approach.

In many practical applications complex geometries have to be handled. This report presents a cut-cell method, which uses a Cartesian background grid. In the interior of the domain the numerical schemes can easily be implemented but in the vicinity of the boundaries special treatment for each cell is required.

Furthermore, many practical applications are exterior problems, which require boundary conditions for outgoing waves. These boundary conditions should not give any noticeable reflections of outgoing waves.

For an intensive investigation of this approach, a testcase has been studied. In this testcase a wave is reflected on a diagonal oblique wall. The investigation has been done by determining the space-dependent order of accuracy of the complete method by Richardson extrapolation. For applying Richardson extrapolation the numerical solutions have to be interpolated to a common grid, which should not affect the numerical solution. So, an accurate interpolation method has been developed that also deals with the complex geometry.

Because the numerical grid spacing and time step are coupled by the CFL-number, the Richardson extrapolation procedure can be reduced to one parameter, the order of accuracy.

Discontinuities in boundary and initial conditions have been investigated, which have a negative impact on the magnitude of the order of accuracy. Furthermore, the impact of the outflow conditions has been investigated for both the initial outgoing wave and the reflected outgoing wave. It turned out that the initial outgoing wave does not affect the order of accuracy, however, the reflected wave does. The outflow boundary conditions are suitable only for the initial outgoing wave. By comparing the order of accuracy of the initial wave and the reflected wave, the impact of the cut-cell method has been determined. In spite of the bad performance of the interpolation method, used in the cut-cell method, the cut-cell method performs well. However, the order of accuracy is lower for the reflected wave in part of the domain.

Richardson extrapolation can also be an effective tool to improve the numerical solution. For problems without any discontinuities Richardson extrapolation gives a smooth improved solution. However, for problems with a discontinuity in the initial or boundary condition, Richardson extrapolation cannot give a smooth improved solution, because this discontinuity leads to a low and unstructured order of accuracy.

Acknowledgement

This Master's thesis has been written for the degree of Master of Science in Applied Mathematics, faculty Electrical Engineering, Mathematics and Computer Science at Delft, University of Technology in the Netherlands. The graduation work has been done in the unit of Numerical Analysis, taking about 9 months of work.

First 3 months of the project has been carried out at Delft, University of Technology and the following 6 months has been carried out at the Department of Aerospace Engineering at the University of Michigan in Ann Arbor, USA.

I would like to thank my direct supervisors Prof. Shyy and Dr. Vuik for their excellent supervising. I would like to thank Jan de Vries, STIR-Fonds, Fundatie van de Vrijvrouwe van Renswoude and the Universiteitsfonds Delft for their financial support. Furthermore, I would like to thank my family for their support.

Rick Vedder
June, Ann Arbor

Nomenclature

λ	Wavelength
f	Frequency
c_0	Speed of sound
ω	Angular frequency
$\overline{\omega}$	Angular frequency of a scheme
α	Wavenumber
$\overline{\alpha}$	Wavenumber of a scheme
x, y	Coordinates in space
t	Time
Δx, Δy	Grid spacing
Δt	Time step size
CFL, ν	Courant-Friedrichs-Lewy number, $CFL = c_0 \frac{\Delta t}{\Delta x}$
D_i	Derivative at $x = x_i$
D_i^B, D_i^F	Backward and forward derivative operator at $x = x_i$
$\widetilde{f}(\alpha)$	Fourier transform of $f(x)$
u_i^e, u_i^w	u at east and west face of cell i
u_i^{Fe}, u_i^{Be}	Forward and backward operator at east face of cell i
u_i^{Fw}, u_i^{Bw}	Forward and backward operator at west face of cell i
K_i	Stage i for Runge-Kutta time integration
$L_{N,k}$	Lagrange k^{th} basis polynomial with degree N
$P(\Delta x)$	Numerical pressure solution obtained with grid spacing Δx
M^0	Common grid on which numerical solutions are interpolated
M^1	Grid of the numerical solution
f_i^{filt}	Filtered solution at $x = x_i$

List of Figures

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 9

1 Introduction

Computational AeroAcoustics (CAA) deals with the disciplines Computational Fluid Dynamics and AeroAcoustics. Sound generated by aerodynamic forces or a flow, fall in the category of aeroacoustics. Aeroacoustics should not be confused with classical acoustics, e.g. sound generated by loudspeakers is in the domain of classical acoustics, whereas sound generated by a turbulent flow is in the aeroacoustic domain.

An acoustic wave has a wavelength λ in space, and a frequency f in time. The speed of sound in a medium is c_0. These are coupled by the following relation: $\lambda f = c_0$. Wave propagation properties are encoded in the dispersion relation of the governing equations [1,2]. The dispersion relation is a relation between the angular frequency $\omega = 2\pi f$ and the wavenumber of the wave $\alpha = \frac{2\pi}{\lambda}$. This relation can easily be obtained by taking space and time Fourier transforms of the governing equations. For example, the dispersion relation of the one-dimensional wave equation is:

$$\frac{\partial^2 u}{\partial t^2} - c_0^2 \frac{\partial^2 u}{\partial x^2} = 0 \qquad \Rightarrow \qquad \omega^2 = c_0^2 \alpha^2 \qquad \Rightarrow \qquad \omega = \pm c_0 \alpha = W(\alpha),$$

with angular frequency ω and wavenumber α.

Dispersiveness (spreading), dissipativeness (damping), group velocity ($v_{gr} = \frac{dW}{d\alpha}$), phase velocity ($v_{ph} = \frac{W(\alpha)}{\alpha}$) etc. are determined by the dispersion relation, e.g. when v_{ph} depends on α the waves are dispersive and when $W(\alpha)$ has an imaginary part the waves are dissipative [1,2].

It is well-known that acoustic waves are non-dispersive. Although sound is dissipated by viscosity, it is dissipated after a very long travel distance and therefore acoustics is considered to be an inviscid fluid phenomenon. So, dissipative loss becomes important only for sound with high frequencies, which travels long distances. Acoustic waves travel with the speed of sound. In order to commit satisfactory CAA all wave properties have to be preserved in the numerical solution, which implies low dispersion and dissipation error. This can be achieved by preserving the dispersion relation as well as possible in the numerical scheme, which can be done by preserving the wavenumber and the angular frequency. So when a numerical scheme has the same dispersion relation as the governing equations, the waves in the numerical solution will have the same properties as those of the governing equation.

Many CFD schemes are dispersive and dissipative. Sometimes numerical dissipation is added to make the scheme stable. In CFD, schemes are usually assessed by the order of the local truncation error, higher order often means better local approximation, and by the Lax theorem for convergence. In acoustics this is not sufficient to ensure a good quality numerical solution, as appears from the previous discussion. A common approach in CFD is to use non-uniform grids, e.g. in boundary layers. In acoustics, stretched grids could cause undesirable phenomena, such as change of frequency or even reflection [3].

Another difficulty in CAA arises from the type of problems in practical applications. These problems often are exterior problems, which implies that these problems have to be solved in an unbounded domain. Thus for computational purposes, the domain has to be cut off to make the computational domain finite. Boundary conditions have to be proposed for the cut-off boundaries. Reflection of outgoing waves from the artificial cut-off boundaries is undesirable.

Also problems in acoustics often have complex geometries. So, good methods to handle these complexities are required.

As explained before, sound propagation is modelled as an inviscid fluid phenomenon. On the other hand, if sound generation by a flow is considered, viscosity (involving the Reynolds number) plays an important role. Sound is mainly generated in turbulent flows, e.g. by acceleration of vorticity.

In Chapter 2 the governing equations for acoustics are derived, the problem is formulated and important objectives of the thesis are described. In Chapter 3 numerical methods are discussed. Space discretization, time integration methods and a cut-cell method are presented in this chapter in order to solve the problem. Also, interpolation methods and Richardson extrapolation are presented in this chapter in order to handle the objectives. In Chapter 4 there is a description of how to apply these methods to the particular problem, formulated in Chapter 2. Chapter 5 describes how problems that appeared with the implementation have been solved. In Chapter 6 results are presented and the conclusions from these results can be found in Chapter 7.

2 Problem

In this chapter an introduction to the problem will be given. The linearized Euler equations are derived, which govern the acoustic problem. Furthermore, the geometry of the problem is defined with the corresponding boundary and initial conditions. After the problem formulation the main objectives of this thesis will be discussed.

2.1 Linearized Euler Equations

As mentioned in the introduction, acoustics is considered to be an inviscid fluid phenomenon. Therefore, the Euler equations are used to derive the governing linearized Euler equations for acoustics. The linearized Euler equations are derived for a homogeneous fluid with characteristics:

$\rho = \rho_0, \qquad p = p_0, \qquad \boldsymbol{u} = 0,$

where $\boldsymbol{u}$ represents a vector and ρ_0 and p_0 are constants.

The Euler equations (governing an inviscid flow) are given by:

$$\begin{aligned} &\rho_t + \boldsymbol{u} \cdot \nabla \rho + \rho \nabla \cdot \boldsymbol{u} = 0, \\ &\rho(\boldsymbol{u}_t + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u}) + \nabla p = 0. \end{aligned} \tag{1}$$

When sound disturbs the fluid, the fluid is characterized by:

$$\begin{aligned} &\rho = \rho_0 + \rho', && |\rho| \ll \rho_0, \\ &p = p_0 + p', && |p'| \ll \rho_0 c_0^2, \\ &\boldsymbol{u} = \boldsymbol{u}', && \|\boldsymbol{u}'\| \ll c_0, \end{aligned} \tag{2}$$

where c_0 is the propagation speed (speed of sound).

Furthermore, the equation of state is given by:

$$\frac{1}{c_0^2} p' = \rho'. \tag{3}$$

Substituting (2) into (1) leads to:

$$\begin{aligned} &(\rho_0 + \rho')_t + \boldsymbol{u}' \cdot \nabla(\rho_0 + \rho') + (\rho_0 + \rho')\nabla \cdot \boldsymbol{u}' = \\ &\rho'_t + \boldsymbol{u}' \cdot \nabla \rho' + \rho_0 \nabla \cdot \boldsymbol{u}' + \rho' \nabla \cdot \boldsymbol{u}' = 0, \end{aligned} \tag{4}$$

and

$$\begin{aligned} &(\rho_0 + \rho')\left[\boldsymbol{u}'_t + (\boldsymbol{u}' \cdot \nabla)\boldsymbol{u}'\right] + \nabla(p_0 + p') = \\ &\rho_0 \boldsymbol{u}'_t + \rho' \boldsymbol{u}'_t + \rho_0(\boldsymbol{u}' \cdot \nabla)\boldsymbol{u}' + \rho'(\boldsymbol{u}' \cdot \nabla)\boldsymbol{u}' + \nabla p' = 0. \end{aligned} \tag{5}$$

Since all perturbation are small, higher order terms can be neglected, which results in the following:

$$\rho'_t + \rho_0 \nabla \cdot \boldsymbol{u}' = 0, \tag{6}$$

$$\rho_0 \boldsymbol{u}'_t + \nabla p' = 0. \tag{7}$$

From (3) it follows that $\rho'_t = \frac{1}{c_0^2} p'_t$, which can be substituted in (6) and multiplied by c_0^2:

$$p'_t + \rho_0 c_0^2 \nabla \cdot \boldsymbol{u}' = 0, \tag{8}$$

$$\rho_0 c_0^2 \boldsymbol{u}'_t + c_0^2 \nabla p' = 0. \tag{9}$$

Now, (8) and (9) are the Euler equations linearized around a state of rest. Note that (8) and (9) satisfy the wave equation.

After non-dimensionalizing and taking $c_0 = 1$ the linearized Euler equations appear, which are used in the following section.

$$\begin{aligned} \frac{\partial p}{\partial t} + \nabla \cdot \boldsymbol{u} = 0, \\ \frac{\partial \boldsymbol{u}}{\partial t} + \nabla p = 0. \end{aligned} \tag{10}$$

The linearized Euler equations have been derived.

2.2 Problem Formulation

The following acoustic problem has been posed by Popescu, Shyy and Tai [4]. The problem is solved in 2-D and has a geometry as in Figure 2.1.

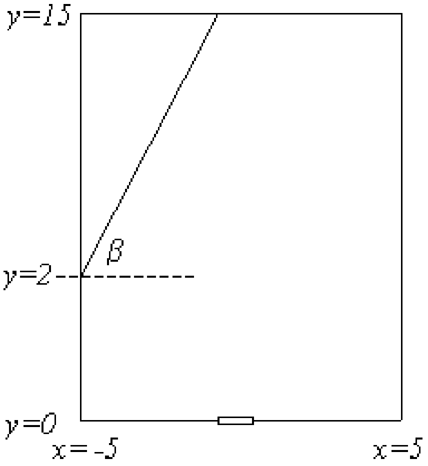

Figure 2.1: Geometry of the problem.

For this problem the computational domain is $(x, y) \in [-5, 5] \times [0, 15]$, which is cut off by a solid wall at $y = 2$, where the angle of the wall with the positive $x-$axis is equal to β. At $y = 0$ and around $x = 0$ a piston, that is baffled, is located. The problem is to find the sound field generated by this baffled piston. This field is reflected by the solid wall. The problem is modelled as follows:

1. The linearized Euler equations are used, which are non-dimensionalized with $c_0 = 1$:

$$\begin{aligned} \frac{\partial p}{\partial t} + \nabla \cdot \boldsymbol{u} = 0, \\ \frac{\partial \boldsymbol{u}}{\partial t} + \nabla p = 0, \end{aligned} \tag{11}$$

   where $\boldsymbol{u} = \binom{u}{v}$.

2. The initial conditions are as follows:

$$\begin{aligned} \boldsymbol{u} = 0, \\ p = 0. \end{aligned} \tag{12}$$

3. The boundary conditions are as follows:

$$v(x, 0, t) = \begin{cases} V_0 \cos(\omega t), & (x, 0) \in \mathit{piston} \\ 0, & \mathit{otherwise} \end{cases} \tag{13a}$$

   where $V_0 = 1$ and $\omega = 4$.

$$\left. \begin{aligned} \boldsymbol{u}_n = 0 \\ \frac{\partial p}{\partial n} = 0 \end{aligned} \right\} \text{ on the solid wall.} \tag{13b}$$

$$\text{Outflow conditions at } x = -5,\ x = 5 \text{ and } y = 15. \tag{13c}$$

The goal is to solve this problem numerically. Several methods are presented in the following chapters to obtain satisfactory numerical solutions. Results of Popescu, Shyy and Tai [4] are presented in Section 6.1.

2.3 Objectives of the Thesis

In this thesis an intensive investigation of the results of Popescu, Shyy and Tai [4] is presented. In the paper they present spatial discretization, time integration and a cut-cell method in order to handle the presented test problem. The results in Section 6.1 are quite satisfactory and give hope this approach to be effective for aeroacoustic problems. However, the results have not yet been investigated in detail. This thesis focuses on further investigation of the results that are obtained by Popescu, Shyy and Tai. An enumeration of the main objectives of this thesis is given.

1. Because there is no analytical solution of this problem, it is impossible to determine the order of accuracy explicitly. However, it is important to know what the order of accuracy is in practice. So, the first objective is to determine the order of accuracy.

2. For the complex boundary a special method has to be developed. This method is called cut-cell method and should preserve the order of accuracy that is derived for the methods used in the interior. A main objective is to investigate what the impact of this cut-cell method is on the order of accuracy of the complete simulation.

3. Finally, investigation of an improvement of the numerical solution is also an objective. Another way then simply decreasing stepsizes in both time and space, could lead to a smaller amount of work and memory in order to obtain a better numerical solution.

Methods and its applicabilities, that are used to deal with these objectives are presented in the next chapters.

3 Numerical Methods

In this chapter numerical method, that are used in this thesis, are presented. This includes spatial discretization, time integration, a cut-cell method, interpolation and extrapolation methods.

3.1 Spatial Discretization

For spatial discretization the Optimized-Prefactored-Compact (OPC) scheme is used. In this scheme the coefficients are determined such that it meets a required order of accuracy and a small dispersion error. In order to obtain a small dispersion error the dispersion relation has to be preserved. This can be done by preserving the wavenumber in the numerical scheme.

3.1.1 The Optimized-Prefactored-Compact Scheme

The OPC scheme is based on prefactored compact schemes, which require very small stencil support. The prefactor procedure splits the implicit central scheme into a forward and backward operator. The compact schemes have small stencils, which implies that the specification of the boundaries is simplified, because no additional conditions have to be proposed. Another advantage of the OPC scheme is that it has the same order of accuracy with a smaller stencil than explicit schemes.

3.1.2 Finite Difference approach

Let us introduce first the compact scheme:

$$\begin{gathered}\beta(D_{i-2}+D_{i+2})+\eta(D_{i-1}+D_{i+1})+D_i \simeq \\ \tfrac{a}{2\Delta x}(u_{i+1}-u_{i-1})+\tfrac{b}{4\Delta x}(u_{i+2}-u_{i-2})+\tfrac{c}{6\Delta x}(u_{i+3}-u_{i-3}).\end{gathered} \tag{14}$$

where D_i is the spatial derivative of u in the point x_i. Note that this implicit scheme uses 7 equally spaced nodes.

In order to obtain a certain order of the local truncation error a Taylor series expansion can be made.

The Taylor series expansion is shown below:

$$\begin{gathered}2\beta\left(\frac{\partial u_i}{\partial x}+\frac{(2\Delta x)^2}{2}\frac{\partial^3 u_i}{\partial x^3}+\frac{(2\Delta x)^4}{4!}\frac{\partial^5 u_i}{\partial x^5}+...\right)+ \\ 2\eta\left(\frac{\partial u_i}{\partial x}+\frac{\Delta x^2}{2}\frac{\partial^3 u_i}{\partial x^3}+\frac{\Delta x^4}{4!}\frac{\partial^5 u_i}{\partial x^5}+...\right)+\frac{\partial u_i}{\partial x}\simeq \\ 2\frac{a}{2\Delta x}\left(\frac{\partial u_i}{\partial x}\Delta x+\frac{\Delta x^3}{3!}\frac{\partial^3 u_i}{\partial x^3}+\frac{\Delta x^5}{5!}\frac{\partial^5 u_i}{\partial x^5}+...\right)+ \\ 2\frac{b}{4\Delta x}\left(\frac{\partial u_i}{\partial x}(2\Delta x)+\frac{(2\Delta x)^3}{3!}\frac{\partial^3 u_i}{\partial x^3}+\frac{(2\Delta x)^5}{5!}\frac{\partial^5 u_i}{\partial x^5}+...\right)+ \\ \frac{c}{6\Delta x}\left(\frac{\partial u_i}{\partial x}(3\Delta x)+\frac{(3\Delta x)^3}{3!}\frac{\partial^3 u_i}{\partial x^3}+\frac{(3\Delta x)^5}{5!}\frac{\partial^5 u_i}{\partial x^5}+...\right).\end{gathered} \tag{15}$$

After matching coefficients the following relations can be obtained [5]:

Second order:

$$a + b + c = 1 + 2\eta + 2\beta, \tag{16}$$

Fourth order:

$$a + 2^2 b + 3^2 c = 2\frac{3!}{2!}(\eta + 2^2\beta), \tag{17}$$

Sixth order:

$$a + 2^4 b + 3^4 c = 2\frac{5!}{4!}(\eta + 2^4\beta), \tag{18}$$

etc.

Only for tenth order all coefficients are unique.

Fourier transforms are used in order to derive the wavenumber of the scheme. Fourier transform with wavenumber $\alpha \in \mathbb{R}$, is defined as:

$$\widetilde{f}(\alpha) = \frac{1}{2\pi}\int_{-\infty}^{\infty} f(x)e^{-i\alpha x}dx, \tag{19}$$

and its inverse:

$$f(x) = \int_{-\infty}^{\infty} \widetilde{f}(\alpha)e^{i\alpha x}d\alpha. \tag{20}$$

The Fourier transform leads to the following derivative and shift theorems:

$$\begin{aligned} \widetilde{\tfrac{\partial f}{\partial x}(x)} &= i\alpha\widetilde{f}(\alpha), \\ \widetilde{f(x+\lambda)} &= e^{i\alpha\lambda}\widetilde{f}(\alpha). \end{aligned} \tag{21}$$

Then the (numerical) wavenumber of the scheme can be determined by applying Fourier transforms and its shift and derivative theorems to (14):

$$\begin{gathered} \beta(e^{-i\alpha 2\Delta x} + e^{i\alpha 2\Delta x})i\alpha\widetilde{u} + \eta(e^{-i\alpha\Delta x} + e^{i\alpha\Delta x})i\alpha\widetilde{u} + i\alpha\widetilde{u} \simeq \\ \tfrac{a}{2\Delta x}(e^{i\alpha\Delta x} - e^{-i\alpha\Delta x})\widetilde{u} + \tfrac{b}{4\Delta x}(e^{i\alpha 2\Delta x} - e^{-i\alpha 2\Delta x})\widetilde{u} + \\ \tfrac{c}{6\Delta x}(e^{i\alpha 3\Delta x} - e^{-i\alpha 3\Delta x})\widetilde{u}, \end{gathered} \tag{22}$$

$$\begin{gathered} (2\beta\cos(2\alpha\Delta x) + 2\eta\cos(\alpha\Delta x) + 1)i\alpha\widetilde{u} \simeq \\ (\tfrac{a}{\Delta x}\sin(\alpha\Delta x) + \tfrac{b}{2\Delta x}\sin(2\alpha\Delta x) + \tfrac{c}{3\Delta x}\sin(3\alpha\Delta x))i\widetilde{u}. \end{gathered} \tag{23}$$

Hence, the numerical wavenumber $\overline{\alpha}$ is given by:

$$\overline{\alpha}\Delta x = \frac{a\sin(\alpha\Delta x) + \frac{b}{2}\sin(2\alpha\Delta x) + \frac{c}{3}\sin(3\alpha\Delta x)}{2\beta\cos(2\alpha\Delta x) + 2\eta\cos(\alpha\Delta x) + 1}. \tag{24}$$

Note that this numerical wavenumber is real.

Now the prefactored compact scheme is defined, which uses a forward and backward operator [5]. The reason for defining a prefactored compact scheme is that it leads to reduced, upper and lower bidiagonal, matrices, which is advantageous when these have to be inverted (This procedure is described in the last part of this section). The derivative operator is split into a forward and a backward operator:

$$D_i = \frac{1}{2}(D_i^B + D_i^F), \tag{25}$$

which are defined by:

$$\beta_F D_i^F + \eta_F D_{i+1}^F \simeq \frac{1}{\Delta x}[a_F u_{i+2} + b_F u_{i+1} + c_F u_i + d_F u_{i-1} + e_F u_{i-2}], \tag{26}$$

$$\gamma_B D_{i-1}^B + \beta_B D_i^B \simeq \frac{1}{\Delta x}[a_B u_{i+2} + b_B u_{i+1} + c_B u_i + d_B u_{i-1} + e_B u_{i-2}]. \tag{27}$$

Again Fourier transforms determine the numerical wavenumer of these operators, which have a real and imaginary part.

The real and imaginary part of the numerical wavenumber of the forward operator is:

$$\begin{aligned}\mathrm{Re}(\overline{\alpha}_F \Delta x) = \; & \frac{(a_F\eta_F+b_F\beta_F-c_F\eta_F-d_F\beta_F)\sin(\alpha\Delta x)}{\eta_F^2+\beta_F^2+2\eta_F\beta_F\cos(\alpha\Delta x)} + \\ & \frac{(a_F\beta_F-d_F\eta_F-e_F\beta_F)\sin(2\alpha\Delta x)-e_F\eta_F\sin(3\alpha\Delta x)}{\eta_F^2+\beta_F^2+2\eta_F\beta_F\cos(\alpha\Delta x)},\end{aligned} \tag{28}$$

$$\begin{aligned}\mathrm{Im}(\overline{\alpha}_F \Delta x) = \; & \frac{-(b_F\eta_F+c_F\beta_F)-(a_F\eta_F+b_F\beta_F+c_F\eta_F+d_F\beta_F)\cos(\alpha\Delta x)}{\eta_F^2+\beta_F^2+2\eta_F\beta_F\cos(\alpha\Delta x)} + \\ & \frac{-(a_F\beta_F+d_F\eta_F+e_F\beta_F)\cos(2\alpha\Delta x)-e_F\eta_F\cos(3\alpha\Delta x)}{\eta_F^2+\beta_F^2+2\eta_F\beta_F\cos(\alpha\Delta x)}.\end{aligned} \tag{29}$$

The real and imaginary part of the numerical wavenumber of the backward operator is:

$$\begin{aligned}\mathrm{Re}(\overline{\alpha}_B \Delta x) = \; & \frac{(b_B\beta_B+c_B\gamma_B-d_B\beta_B-e_B\gamma_B)\sin(\alpha\Delta x)}{\gamma_B^2+\beta_B^2+2\gamma_B\beta_B\cos(\alpha\Delta x)} + \\ & \frac{(a_B\beta_B+b_B\gamma_B-e_B\beta_B)\sin(2\alpha\Delta x)+a_B\gamma_B\sin(3\alpha\Delta x)}{\gamma_B^2+\beta_B^2+2\gamma_B\beta_B\cos(\alpha\Delta x)},\end{aligned} \tag{30}$$

$$\begin{aligned}\mathrm{Im}(\overline{\alpha}_B \Delta x) = \; & \frac{-(c_B\beta_B+d_B\gamma_B)-(b_B\beta_B+c_B\gamma_B+d_B\beta_B+e_B\gamma_B)\cos(\alpha\Delta x)}{\gamma_B^2+\beta_B^2+2\gamma_B\beta_B\cos(\alpha\Delta x)} + \\ & \frac{-(a_B\beta_B+b_B\gamma_B+e_B\beta_B)\cos(2\alpha\Delta x)-a_B\gamma_B\cos(3\alpha\Delta x)}{\gamma_B^2+\beta_B^2+2\gamma_B\beta_B\cos(\alpha\Delta x)}.\end{aligned} \tag{31}$$

The real parts represent the dispersion relation and the imaginary parts the dissipation.

In order to be equivalent with the original compact scheme Hixon and Turkel [6] defined several conditions:

1. The imaginary parts of the forward and backward wavenumbers have to be equal but of opposite sign.

2. The real parts have to be equal and also equal to the wavenumber of the compact scheme.

To satisfy the first condition the following is required:

$$\begin{gathered}\beta_B = \beta_F, \qquad \gamma_B = \eta_F, \\ a_B = -e_F, \qquad b_B = -d_F, \qquad c_B = -c_F, \qquad d_B = -b_F, \qquad e_B = -a_F.\end{gathered} \tag{32}$$

To satisfy the second condition and supplementary relation $a_F + b_F + c_F + d_F + e_F = 0$, to ensure that when gradients are zero the computed gradients are also zero, the coefficients are known as a function of a, b, c, β and η , which are not completely determined yet in case of lower order accuracy than 10^{th} order.

Now the stencil has to be optimized. The optimize technique is applied to the original compact scheme. An errorfunction E is defined to measure the difference between the wavenumber and the numerical wavenumber:

$$E = \int_0^{r\pi} (\alpha\Delta x - \overline{\alpha}\Delta x)^2 W(\alpha\Delta x) d(\alpha\Delta x), \tag{33}$$

where $W(\alpha\Delta x)$ is a weigth function, which makes the expression analytically integrable, and r defines the optimized range. Kim and Lee [7] chose the function $W(\alpha\Delta x) = [2\beta\cos(2\alpha\Delta x) + 2\eta\cos(\alpha\Delta x) + 1]^2$. How much free variables are present depends on the choice of the order of accuracy. The errorfunction can be minimized by the free parameter(s) in the following way:

$$\min_{a_k} E \qquad \Leftrightarrow \qquad \frac{\partial E}{\partial a_k} = 0,$$

where a_k is a free parameter.

Now the problem is closed and all parameters of the compact scheme are known, which implies that also all parameters of the forward and backward schemes are known.

A 5-point compact stencil, which means $\beta = c = 0$, is equivalent with a 3-point OPC scheme. By requiring 4^{th} order of accuracy there is one parameter left for optimization. This 4^{th} order optimized 3-point OPC stencil is called the 6/4 OPC, because the scheme has an order of accuracy of 4 while the maximum order of accuracy is 6. The 8/4 OPC scheme is a 5-point stencil and uses two parameters for optimization.

Also boundary stencils have to be taken into account. For example, for a domain consisting of N nodes, a three-point stencil can be applied to the points $j = 2$ to $N - 1$. Therefore for the points 1 and N explicit boundary stencils have to be defined. These forward and backward boundary stencils are explicitly defined as:

$$\begin{aligned} D_1^B &= \tfrac{1}{\Delta x} \sum_{j=1}^{4} s_j u_j, \qquad & D_N^B &= \tfrac{1}{\Delta x} \sum_{j=N-3}^{N} e_j u_j, \\ D_1^F &= \tfrac{1}{\Delta x} \sum_{j=1}^{4} -e_{N+1-j} u_j, \qquad & D_N^F &= \tfrac{1}{\Delta x} \sum_{j=N-3}^{N} -s_{N+1-j} u_j, \end{aligned} \tag{34}$$

where the coefficients s_j and e_j are determined by matching coefficients from the Taylor series expansion.

Now, the system of equations following from (26) and (27) can be written as follows:

$$[A^F]D^F = [B^F]u, \tag{35}$$

$$[A^B]D^B = [B^B]u, \tag{36}$$

where [.] are matrices.

The system of equations following from (25) can be obtained using (35) and (36):

$$D = \frac{1}{2}([A^F]^{-1}[B^F] + [A^B]^{-1}[B^B])u. \tag{37}$$

The matrices $[A^F]$ and $[A^B]$ are an upper bidiagonal and a lower bidiagonal matrix, respectively.

Figure 3.1 [3] shows $\overline{\alpha}\Delta x$ as a function of $\alpha\Delta x$ for several schemes. Also the ideal situation $\overline{\alpha}\Delta x = \alpha\Delta x$ for all $\alpha\Delta x$ is shown.

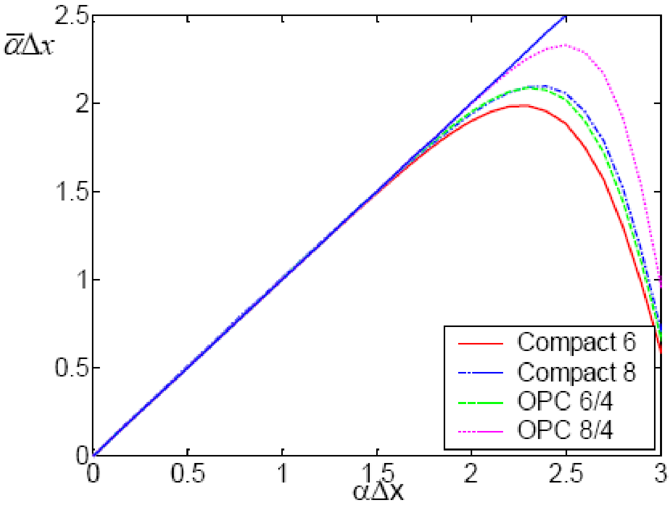

Figure 3.1: $\alpha\Delta x$ versus $\overline{\alpha}\Delta x$ for two OPC schemes and two compact schemes.

From Figure 3.1 it appears that $\overline{\alpha}\Delta x$ and $\alpha\Delta x$ are nearly the same up to $\alpha_c\Delta x$, which is easily determined by a simple condition, e.g. $|\alpha\Delta x - \overline{\alpha}\Delta x| < 0.001$. Also the group velocity is considered and compared with the other schemes in Figure 3.2 [3].

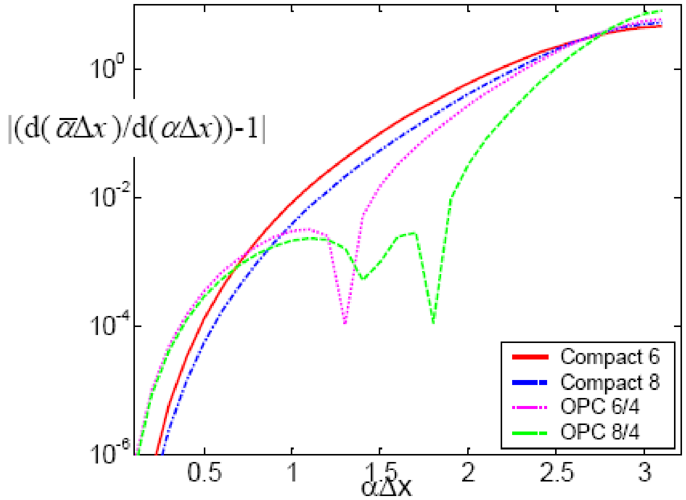

Figure 3.2: $\left|\frac{d(\overline{\alpha}\Delta x)}{d(\alpha\Delta x)} - 1\right|$ versus $\alpha\Delta x$ for two OPC schemes and two compact schemes.

It appears from the figures that the optimized schemes have a much larger range of wavenumbers, in which these perform well, than the unoptimized schemes.

3.1.3 Finite Volume approach

For the finite volume derivation of the OPC scheme Popescu [3] uses the one-dimensional linear wave equation (38) and the geometry as given in Figure 3.3.

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0. \tag{38}$$

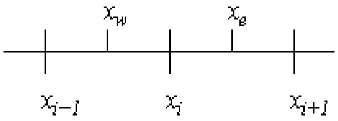

Figure 3.3: A cell centered grid with control volume $[x_w, x_e]$.

Integration over a cell gives:

$$\left(\overline{\frac{\partial u}{\partial t}}\right)_i \Delta x + u_i|_w^e = \left(\overline{\frac{\partial u}{\partial t}}\right)_i \Delta x + u_i^e - u_i^w = 0, \tag{39}$$

where $\left(\frac{\partial u}{\partial t}\right)_i$ is an average value over a control volume i.

Equations (25), (26) and (27) describe the OPC scheme and the approximations of u_e and u_w have similar forms:

$$\begin{aligned} u_i^e &= \tfrac{1}{2}(u_i^{Fe} + u_i^{Be}), \\ u_i^w &= \tfrac{1}{2}(u_i^{Fw} + u_i^{Bw}), \end{aligned} \tag{40}$$

where u_i^{Fe}, u_i^{Be}, u_i^{Fw} and u_i^{Bw} are determined from:

$$\begin{aligned} \eta u_{i+1}^{Fe} + \beta u_i^{Fe} &= b u_{i+1} - d u_i, \\ \eta u_{i+1}^{Fw} + \beta u_i^{Fw} &= b u_i - d u_{i-1}, \\ \beta u_i^{Be} + \eta u_{i-1}^{Be} &= b u_i - d u_{i+1}, \\ \beta u_i^{Bw} + \eta u_{i-1}^{Bw} &= b u_{i-1} - d u_i. \end{aligned} \tag{41}$$

Here the coefficients η, β, b and d are the same as in the finite difference approach: $\eta = \eta_F$, $\beta = \beta_F$, $b = b_F$ and $d = d_F$.

In order to solve this, again explicit boundary stencils are required like (34):

$$\begin{aligned} u_1^w &= \sum_{i=1}^{3} a_i u_i, & u_N^w &= \sum_{i=1}^{3} r_i u_{N-i}, \\ u_1^e &= \sum_{i=1}^{3} a_i u_{i+1}, & u_N^e &= \sum_{i=1}^{3} r_i u_{N-i+1}, \end{aligned} \tag{42}$$

all for forward and backward operators. Where the coefficients are:

$$\begin{aligned} a_1^B &= -s_1, & a_1^F &= e_N, \\ a_2^B &= -s_1 - s_2, & a_2^F &= e_N + e_{N-1}, \\ a_3^B &= -s_1 - s_2 - s_3, & a_3^F &= e_N + e_{N-1} - e_{N-2}, \\ \\ r_1^B &= e_N, & r_1^F &= -s_1, \\ r_2^B &= e_N + e_{N-1}, & r_2^F &= -s_1 - s_2, \\ r_3^B &= e_N + e_{N-1} - e_{N-2}, & r_3^F &= -s_1 - s_2 - s_3. \end{aligned} \tag{43}$$

Now, the systems of equations can be solved like in the finite differences approach.

3.2 Time Integration

For time integration a modified Runge-Kutta scheme is used. This scheme also has the goal to preserve the dispersion relation. This scheme is developed with Fourier transforms and Taylor series like the space discretization and is called the Low-Dispersion-and-Dissipation-Runge-Kutta scheme (LDDRK).

3.2.1 Low-Dispersion-and-Dissipation-Runge-Kutta scheme

For time integration the modified Runge-Kutta scheme, Low Dispersion and Dissipation Runge-Kutta (LDDRK), has been developed by Hu, Hussaini and Manthey [8]. In the original Runge-Kutta scheme the coefficients are chosen such that the Taylor series coefficients match up to a certain order of accuracy. The Runge-Kutta scheme is applied to the equation:

$$\frac{du}{dt} = F(u). \tag{44}$$

Consider the original p-stage explicit Runge-Kutta scheme in the $(n+1)^{th}$ iteration:

$$\begin{aligned} K_1 &= \Delta t F(u^n), \\ &\vdots \\ K_i &= \Delta t F(u^{(i-1)}), \\ u^{(i)} &= u^n + b_i K_i, \qquad i = 1...p, \\ &\vdots \\ u^{n+1} &= u^{(p)}, \end{aligned} \tag{45}$$

where $b_p = 1$.

A reason for using this notation is that it is easier to implement and it needs to store only two K_i's against p for the standard notation. See Appendix A to note the equivalence of this notation and standard notation for linear problems.

Another way to write u^{n+1} is:

$$u^{n+1} - u^n \simeq \sum_{j=1}^{p} \prod_{l=p-j+1}^{p} b_l \Delta t^j \frac{d^j u^n}{dt^j}. \tag{46}$$

When the left-hand side of this equation is expanded into a truncated Taylor series and the right hand side is written out the coefficients $\gamma_j = \prod_{l=p-j+1}^{p} b_l$ can be determined:

$$\begin{gathered} \Delta t \tfrac{du^n}{dt} + \tfrac{\Delta t^2}{2} \tfrac{d^2 u^n}{dt^2} + ... + \tfrac{\Delta t^p}{p!} \tfrac{d^p u^n}{dt^p} \simeq \\ b_p \Delta t \tfrac{du^n}{dt} + (b_p b_{p-1}) \Delta t^2 \tfrac{d^2 u^n}{dt^2} + ... + (b_p ... b_1) \Delta t^p \tfrac{d^p u^n}{dt^p}. \end{gathered} \tag{47}$$

Hence, $\gamma_j = \frac{1}{j!}$ for p^{th} order accuracy, like in the p-stage standard Runge-Kutta.

By applying Fourier transforms to (46) the numerical amplification factor r is obtained:

$$r = \frac{\widetilde{u}^{n+1}}{\widetilde{u}^{n}} = 1 + \sum_{j=1}^{p} \gamma_j (-i\overline{\omega}\Delta t)^j = 1 + \sum_{j=1}^{p} \gamma_j (-i\sigma)^j \,. \tag{48}$$

Also the exact amplification factor can be obtained:

$$r_{exact} = e^{-i\overline{\omega}\Delta t} = e^{-i\sigma}. \tag{49}$$

With $\gamma_j = \frac{1}{j!}$, it is easy to see that the numerical amplification factor is the truncated Taylor series of the exact amplification factor. To compare the exact and numerical amplification factor its ratio has to be obtained:

$$\frac{r}{r_{exact}} = \frac{1 + \sum_{j=1}^{p} \gamma_j (-i\sigma)^j}{e^{-i\sigma}}, \tag{50}$$

which can be rewritten as:

$$\frac{r}{r_{exact}} = |\eta| e^{-i\delta}. \tag{51}$$

In this expression $|\eta|$ represents the dissipation rate and δ represents the dispersion rate. Where the $|\eta|$ should be 1 and δ should be 0 for r and r_{exact} to be equal. Thus for accuracy's sake δ must be close to 0 and $|\eta|$ must be close to 1 and for stability's sake $|\eta| \leq 1$. To obtain this the Runge-Kutta method has to be modified to optimize the dispersion and dissipation rate. Hu, Hussiani and Manthey [8] showed that it is sufficient to minimize $|r - r_{exact}|^2$ as a function of $\overline{\omega}\Delta t$:

$$\min_{\gamma_j} \int_0^{\Gamma} \left| 1 + \sum_{j=1}^{p} \gamma_j (-i\overline{\omega}\Delta t)^j - e^{-i\overline{\omega}\Delta t} \right|^2 d(\overline{\omega}\Delta t)\,, \tag{52}$$

with a supplementary condition of $|\eta| \leq 1$ and $[0, \Gamma]$ the optimization range.

Hu, Hussiani and Manthey [8] showed that $|r - r_{exact}|^2$ is an approximation of the sum of the dispersion and dissipation errors. They also showed that minimization of this integral also preserves frequency. From Figure 3.4 [3] it appears that LDDRK has better dispersion and dissipation properties than the ordinary Runge-Kutta integration. Here L and R denote the accuracy limit and the stability limit, respectively.

Also the dissipation and dispersion error of an alternating scheme can be optimized. An example of the alternating scheme 4-6-LDDRK: in the odd time steps the four stage LDDRK and the even time steps the six stage LDDRK is used. In this procedure $|r_4 r_6 - r_{exact}^2|^2$ is minimized. For alternating schemes the dissipation and dispersion error can be further reduced with the minimization and another advantage is that higher order of accuracy can be maintained.

Hu, Hussiani and Manthey [8] present the coefficients of 4-6-LDDRK and 5-6-LDDRK, which are both fourth order accurate.

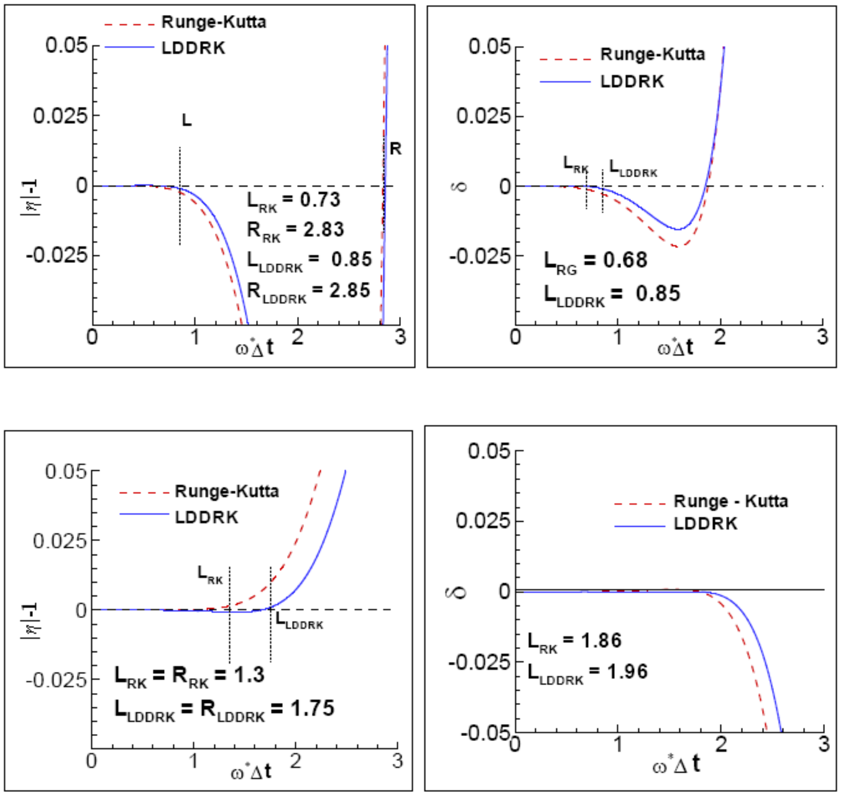

Figure 3.4: Dissipation and dispersion errors. Above 4 stages, below 6 stages.

Popescu [3] specifies the coefficients of the 4-6-LDDRK scheme:

1. Four stages:

$$b_1 = \tfrac{1}{4}, \qquad b_2 = \tfrac{1}{3}, \qquad b_3 = \tfrac{1}{2}.$$

2. Six stages:

$$b_1 = 0.17667, \qquad b_2 = 0.38904, \qquad b_3 = \tfrac{1}{4}, \qquad b_4 = \tfrac{1}{3}, \qquad b_5 = \tfrac{1}{2}.$$

3.3 Cut-Cell Method

In general there are two approaches to generate a grid.

1. A boundary conforming grid: the grid lines match with the boundary
2. A boundary non-conforming grid, the grid lines do not match or intersect the boundary

Often the first option results in curvilinear grids, for which numerical schemes are harder to implement due to the irregular cells. With the second option often a Cartesian grid is used, for which special treatment for each boundary cell is required. A great advantage of this Cartesian grid is the easy implementation for the interior domain. For the implementation of the boundaries the cut-cell technique is used. The standard second order cut-cell method is described below as an introduction for the cut-cell method for acoustic problems which is described later.

3.3.1 Ordinary Second Order Cut-Cell Method

The basic idea of the cut-cell method is to rearrange the control volumes, which are in vicinity of the boundaries, to create cells that conform the boundaries. Cells are cut off according to the boundary and become an independent cell or are merged with another (cut) cell. Cut cells are merged with others if the cell area is less than a minimum acceptable cell area $S_{\min}$. Otherwise, it could be independent. So, this procedure produces new irregular shaped boundary cells. In a finite volume approach the fluxes across the faces of the cells are approximated by:

$$\oint f \cdot n ds \approx \sum_{i=1}^{k} f_i n_i, \tag{53}$$

where flux f contains both the convective and the diffusive flux, resulting from a differential equation for ϕ. To compute the fluxes on the faces the midpoint rule is used which means that the fluxes are evaluated at the center of the faces.

For further explanation of the cut-cell method the geometry as in Figure 3.5 is used.

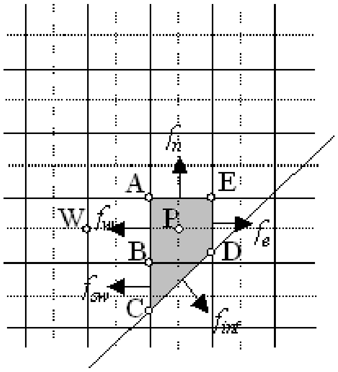

Figure 3.5: Two cut cells, $ABDE$ and BCD, are merged into a new cell $ABCDE$.

The approximation of the flux on a face is split into an approximation of the flux on a cut face and an approximation of the flux on a regular face. For example, the flux on face AC in Figure 3.5 is split into a flux f_w on regular face AB and a flux f_{sw} on cut face BC, which results in:

$$\int\limits_{AC} f\,dy = \int\limits_{AB} f\,dy + \int\limits_{BC} f\,dy. \tag{54}$$

This integral can be approximated by:

$$\int\limits_{AC} f\,dy \approx f_w(y_A - y_B) + f_{sw}(y_B - y_C). \tag{55}$$

In this case a second order approximation of f_w can be made just by linear interpolation of neighboring nodes P and W. For example if f_w requires a value for ϕ, this can be interpolated: $\phi_w = \lambda\phi_W + (1-\lambda)\phi_P$. If f_w requires the derivative of ϕ, this can be handled by a second order finite difference approximation.

This cannot be done for the fluxes f_{sw} and f_e, because of the absence of neighboring nodes due to the boundary. The approximation for f_{sw} is shown below, whereas f_e can be done in the same way. These fluxes can be approximated by making an interpolation polynomial. To make a second order approximation, an interpolation polynomial is made that is linear in x and quadratic in y:

$$\phi(x,y) = c_1xy^2 + c_2y^2 + c_3xy + c_4y + c_5x + c_6. \tag{56}$$

So,

$$\frac{\partial\phi}{\partial x} = c_1y^2 + c_3y + c_5. \tag{57}$$

This polynomial has six unknown coefficients. In order to solve these coefficients six neighboring points are needed. The trapezoidal region defined for this is shown in Figure 3.6.

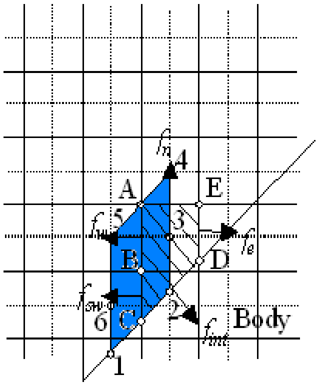

Figure 3.6: Cut cell $ABCDE$ (lined) and the trapezoidal region 123456 (blue) for the interpolation polynomial.

Substituting these six points into the interpolation polynomial leads to the following system:

$$\begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_6 \end{bmatrix} = \begin{bmatrix} x_1 y_1^2 & y_1^2 & x_1 y_1 & y_1 & x_1 & 1 \\ x_2 y_2^2 & y_2^2 & x_2 y_2 & y_2 & x_2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_6 y_6^2 & y_6^2 & x_6 y_6 & y_6 & x_6 & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_6 \end{bmatrix}. \tag{58}$$

This system is easily inverted, which gives c_1,..., c_6 in terms of ϕ_1,..., ϕ_6. Now, ϕ_{sw} and $\frac{\partial \phi_{sw}}{\partial x}$ are determined in terms of the neighboring nodes ϕ_1,..., ϕ_6 by substituting c_1,..., c_6, x_{sw} and y_{sw} into the interpolation polynomial.

In a similar way f_e can be determined. Also north and south faces, which are cut, are treated in this way. The only difference is that the interpolation polynomial is linear in y and quadratic in x.

Now, f_{int} on face CD has to be calculated. Since f_{int} is on the boundary, given boundary conditions could be implemented. When Dirichlet conditions are given this can directly be implemented. Then only $\frac{\partial \phi_{int}}{\partial n}$ is needed to approximate, when present in f_{int}. When Neumann conditions are given, only ϕ_{int} is needed to approximate. Below the description of the approximation for $\frac{\partial \phi_{int}}{\partial n}$:

$$\frac{\partial \phi}{\partial n} = \frac{\partial \phi}{\partial x} n_x + \frac{\partial \phi}{\partial y} n_y, \tag{59}$$

where n_x and n_y are the components of the unit normal vector of face CD, which are known. From Figure 3.6 it appears that f_{int} lies on node 2. This implies that node 3, 4 and "4_n" (north neighboring node of 4) are its neighboring nodes in y-direction. So for the approximation of $\frac{\partial \phi}{\partial y}$ the interpolation polynomial is only quadratic in y:

$$\phi(y) = c_1 y^2 + c_2 y + c_3, \tag{60}$$

for which the resulting system is easily solved. The approximation of $\frac{\partial \phi_{int}}{\partial y}$ is written as $\frac{\partial \phi_{int}}{\partial y} \approx 2c_1 y_{int} + c_2$.

To approximate $\frac{\partial \phi_{int}}{\partial x}$ the procedure is similar to the procedure used for the approximation of $\frac{\partial \phi_{sw}}{\partial x}$.

All fluxes have been determined now.

3.3.2 Cut-Cell Method for CAA Approach

In order to preserve the order of accuracy as developed in the DRP or OPC schemes the same order of accuracy is needed for the boundary implementation. To explain the adapted procedure for the CAA approach, again the geometry as in Figure 3.5 is used. Also the finite volume technique is used and gives the integral approximations as described in the ordinary cut-cell procedure. Only difference is the approximations of the fluxes.

Now f_w can be approximated by a given boundary stencil of the DRP or OPC scheme. For the OPC scheme this is for example (34) and for the DRP scheme ghostpoints have to be introduced. In the ordinary cut-cell procedure central differences and interpolation are used to approximate f_w.

To find a fourth order approximation of f_{sw} an interpolation polynomial is used. To make this approximation fourth order this polynomial is third order in x and fourth order in y:

$$\phi(x, y) = \sum_{i=0}^{4} \sum_{j=0}^{3} c_{ij} x^j y^i. \tag{61}$$

This polynomial has 20 unknown coefficients. Thus, 20 points are needed to determine these coefficients in terms of ϕ_1,..., ϕ_{20}. The trapezoidal region with points 1,..., 20 is shown in Figure 3.7. This results in a similar system as in the ordinary case. The only difference is that the system is now $\phi = [A]c$, with ϕ and c twenty-dimensional vectors and $[A]$ a 20×20 matrix, which is easily inverted.

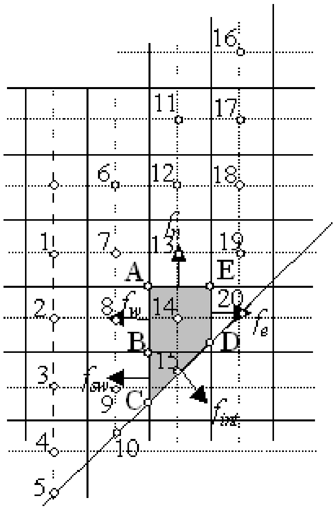

Figure 3.7: The cut cell $ABCDE$ and the trapezoidal $1 \cdots 20$ to approximate f_{sw}.

Also the approximation of f_e can be done with this procedure as well as north and south fluxes of this kind.

The approximation of f_{int} in this case is also similar to the approximation in the ordinary case. Again the approximation of the normal derivative is explained:

$$\frac{\partial \phi}{\partial n} = \frac{\partial \phi}{\partial x} n_x + \frac{\partial \phi}{\partial y} n_y, \tag{62}$$

where $\frac{\partial \phi_{int}}{\partial x}$ can be approximated by a interpolation polynomial that is third order in x and fourth order in y, like f_{sw} has been approximated, and $\frac{\partial \phi_{int}}{\partial y}$ can be approximated by a polynomial that is fourth order in y along the vertical line 14, 13, 12 , 11 and "11_n":

$$\phi(y) = c_1 y^4 + c_2 y^3 + c_3 y^2 + c_4 y + c_5, \tag{63}$$

with derivative:

$$\frac{\partial \phi}{\partial y} = 4c_1 y^3 + 3c_2 y^2 + 2c_3 y + c_4. \tag{64}$$

Substituting these 5 points in (63) leads to a small system, which is easily solved.

All fluxes have been determined with this cut-cell method. The procedure for acoustic problems is similar to that for ordinary problems, the only extension is higher order interpolation polynomials.

3.4 Interpolation Method

In order to apply Richardson extrapolation several numerical solutions have to be obtained. These solutions will be obtained with different grid spacings, which imply that, in general, the grids will not match. For Richardson extrapolation matching grids are needed. Therefore, the numerical solutions have to be interpolated to a common grid. In the next sections Lagrange interpolation is described for the one-dimensional case and its extension to the two-dimensional case.

3.4.1 One-dimensional Lagrange Interpolation

The Lagrange interpolation polynomial is the interpolation polynomial that equals the given values $f(x_i)$ at the given positions x_i.

Assume there is a set $\Lambda = \{(x_0, f(x_0)), (x_1, f(x_1)), ..., (x_N, f(x_N))\}$. For this set the unique Lagrange interpolation polynomial $P(x)$ of the N^{th} degree can be written as follows:

$$P(x) = \sum_{k=0}^{N} f(x_k) L_{N,k}(x), \tag{65}$$

where $L_{N,k}(x)$ must satisfy $L_{N,k}(x_i) = \left\{ \begin{array}{ll} 1, & \text{if } i = k \\ 0, & otherwise \end{array} \right.$.
With this condition the polynomial $P(x)$ intersects the set Λ.
All $L_{N,k}(x)$ are of degree N and are defined as follows:

$$\begin{aligned} L_{N,k}(x) &= \frac{(x-x_0)(x-x_1)\cdots(x-x_{k-1})(x-x_{k+1})\cdots(x-x_N)}{(x_k-x_0)(x_k-x_1)\cdots(x_k-x_{k-1})(x_k-x_{k+1})\cdots(x_k-x_N)} \\ &= \prod_{i=0,i\neq k}^{N} \frac{x-x_i}{x_k-x_i}. \end{aligned} \tag{66}$$

So, the Lagrange interpolation polynomial (65) can be written as follows:

$$P(x) = \sum_{k=0}^{N} f(x_k) \prod_{i=0,i\neq k}^{N} \frac{x-x_i}{x_k-x_i}. \tag{67}$$

Suppose that $x_0, x_1 ... x_N \in [a,b]$ and $f \in C^{N+1}[a,b]$. Burden and Faires [9] showed that for each $x \in [a,b]$ there exists a $\xi(x) \in [a,b]$, such that the following holds:

$$f(x) - P(x) = \frac{f^{(N+1)}(\xi(x))}{(N+1)!}(x-x_0)(x-x_1)\cdots(x-x_N). \tag{68}$$

Despite the fact that, in general, f is not known, this error formula gives an indication of the order of accuracy of Lagrange interpolation.

3.4.2 Two-dimensional Lagrange Interpolation

The procedure can easily be extended to the two-dimensional case.

Assume there is a set

$$\Lambda = \{(x_0, y_0, f(x_0, y_0)), (x_1, y_0, f(x_1, y_0)), ..., (x_N, y_M, f(x_N, y_M))\}.$$

This set consists of $(N+1) \times (M+1)$ pairs of $(x_i, y_j, f(x_i, y_j))$. Again, the Lagrange interpolation polynomial should equal the values $f(x_i, y_j)$ at the positions (x_i, y_j). The polynomial will be of N^{th} degree in x and of M^{th} degree in y.

Then the two-dimensional Lagrange interpolation polynomial can be written as follows [13]:

$$P_{N \times M}(x, y) = \sum_{i=0}^{N} \sum_{j=0}^{M} \left(\prod_{k=0, k \neq i}^{N} \frac{x - x_k}{x_i - x_k} \right) \left(\prod_{l=0, l \neq j}^{M} \frac{y - y_l}{y_j - y_l} \right) f(x_i, y_j). \tag{69}$$

In this case, the set is the rectangle $[x_0, x_N] \times [y_0, y_M]$.

In order to make interpolation applicable for more general domains, the Lagrange interpolation polynomial can also be determined with a more general set consisting of M pairs: $\Lambda = \{(x_1, y_1, f(x_1, y_1)), (x_2, y_2, f(x_2, y_2)), ..., (x_M, y_M, f(x_M, y_M))\}$.

Let us define:

- N_x is the degree of the polynomial in x,
- and N_y is the degree of the polynomial in y.

Then the interpolation polynomial is defined as follows:

$$c_1 x^{N_x} y^{N_y} + c_2 x^{N_x} y^{N_y - 1} + ... + c_{M-2} x + c_{M-1} y + c_M, \tag{70}$$

and the following must hold in order to determine an unique Lagrange polynomial:

$$M = (N_x + 1) \times (N_y + 1). \tag{71}$$

The coefficients c_1, c_2,..., c_M can be determined by solving the following system:

$$\begin{bmatrix} x_1^{N_x} y_1^{N_y} & x_1^{N_x} y_1^{N_y - 1} & \cdots & x_1 & y_1 & 1 \\ x_2^{N_x} y_2^{N_y} & x_2^{N_x} y_2^{N_y - 1} & \cdots & x_2 & y_2 & 1 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ x_{M-1}^{N_x} y_{M-1}^{N_y} & x_{M-1}^{N_x} y_{M-1}^{N_y - 1} & \cdots & x_{M-1} & y_{M-1} & 1 \\ x_M^{N_x} y_M^{N_y} & x_M^{N_x} y_M^{N_y - 1} & \cdots & x_M & y_M & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{M-1} \\ c_M \end{bmatrix} = \begin{bmatrix} f(x_1, y_1) \\ f(x_2, y_2) \\ \vdots \\ f(x_{M-1}, y_{M-1}) \\ f(x_M, y_M) \end{bmatrix} \tag{72}$$

3.5 Richardson Extrapolation

Let $P(\Delta x)$ be an approximation of p. Then the error of this approximation can be written as follows:

$$p - P(\Delta x) = C_1 \Delta x^{n_1} + C_2 \Delta x^{n_2} + ... \tag{73}$$

where $C_i \in \mathbb{R}$ and $n_i \in \mathbb{N}$ with $0 \leq n_1 < n_2 < ...$

When Δx is small, (73) can be approximated by:

$$p - P(\Delta x) = C \Delta x^n, \tag{74}$$

with $n = n_1$.

Now, this equation can be written for Δx, $\frac{\Delta x}{2}$ and $\frac{\Delta x}{4}$, which gives the following system:

$$\begin{array}{l} p - P(\Delta x) = C \Delta x^n, \\ p - P(\frac{\Delta x}{2}) = C \left(\frac{\Delta x}{2}\right)^n, \\ p - P(\frac{\Delta x}{4}) = C \left(\frac{\Delta x}{4}\right)^n. \end{array} \tag{75}$$

After subtracting the second equation from the first, the third from the second and dividing these expressions the following is obtained:

$$\frac{P(\frac{\Delta x}{2}) - P(\Delta x)}{P(\frac{\Delta x}{4}) - P(\frac{\Delta x}{2})} = 2^n. \tag{76}$$

From this equation the order of accuracy n can be determined.

By subtracting the second and third equation of (75), C can be determined:

$$C = \frac{P(\frac{\Delta x}{4}) - P(\frac{\Delta x}{2})}{(\frac{\Delta x}{2})^n (1 - (\frac{1}{2})^n)}. \tag{77}$$

From the third equation of (75) an improvement of the solution accuracy can be derived:

$$P_{impr} = P(\frac{\Delta x}{4}) + C \left(\frac{\Delta x}{4}\right)^n. \tag{78}$$

This can be written as follows:

$$P_{impr} = \frac{2^n P(\frac{\Delta x}{4}) - P(\frac{\Delta x}{2})}{(2^n - 1)}. \tag{79}$$

This procedure can easily be extended to higher dimensions, what could be done in order to make Richardson extrapolation applicable for an instationary problem.

4 Applicability

In this chapter it is described how the numerical methods are used and applied in order to solve the formulated problem and objectives.

4.1 Obtaining Numerical Solutions

4.1.1 Problem Modelling

Popescu, Shyy and Tai [4] solved the problem, posed in Section 2.1, with a cell-centered grid approach. However, there are cell centers on the north, east, south and west boundaries of the domain. So the domain is slightly extended by $\frac{\Delta x}{2}$ and $\frac{\Delta y}{2}$, in order to preserve the formal definition of a cell-centered approach.

For the spatial discretization of the linearized Euler equations (11) Popescu, Shyy and Tai [4] use the finite volume version of the 6/4 OPC scheme, as described in Section 3.1.1. So, the spatial discretization should have fourth order of accuracy. A uniform grid is used with $\Delta x = \Delta y = 0.05$ and $CFL = c_0 \frac{\Delta t}{\Delta x} = 0.5$ to determine the time step.

For time integration the 4-6-LDDRK method, as described in Section 3.2, with the time step, that just has been defined, is used. Also, the time integration should have fourth order of accuracy.

The initial condition (12) and boundary condition (13a) are easily implemented.

In vicinity of the solid wall the cut-cell procedure is used. For the implementation of boundary condition (13b) Figure 4.1 [4] is used to clarify the procedure.

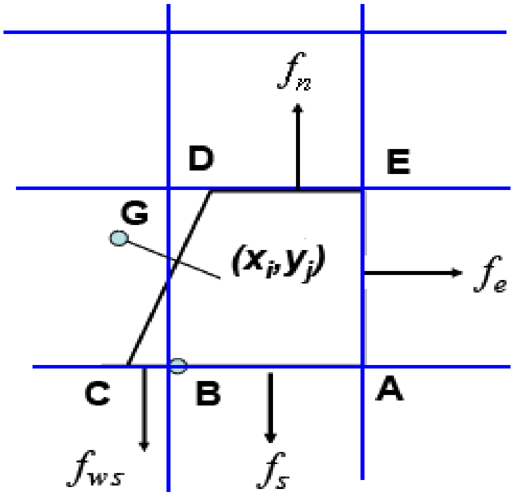

Figure 4.1: Geometry in vicinity of the solid wall with cut cell $ABCDE$.

The interpolation polynomials, as described in Section 3.3.2, are taken to be fourth order in x and y. The fluxes f_{ws} and f_n are approximated with these polynomials. For the fluxes f_s and f_e the boundary stencils of the OPC scheme can be used. For the flux on face CD a virtual point G is defined. Point (x_i, y_j) is the mass center of the boundary cell and G is the symmetrical opposite of this mass center. Then $\boldsymbol{u}_{CD}$ and p_{CD} are defined as follows:

$$\begin{array}{ll} \boldsymbol{u}_{CD} = \frac{\boldsymbol{u}_{ij}+\boldsymbol{u}_G}{2}, & \text{with } \boldsymbol{u}_G = \boldsymbol{u}_{ij} - 2(\boldsymbol{u}_{ij}\cdot\boldsymbol{n})\boldsymbol{n}. \\ p_{CD} = \frac{p_{ij}+p_G}{2}, & \text{with } p_G = p_{ij}. \end{array}$$

After some mathematical manipulation it appears that boundary condition (13b) holds.

The modelling of the outflow boundary conditions (13c) are based on the outflow boundary conditions of Tam [10]:

$$\begin{gathered} \frac{\partial p}{\partial t} + \frac{\partial p}{\partial x}\cos(\theta) + \frac{\partial p}{\partial y}\sin(\theta) + \frac{p}{2r} = 0, \\ \frac{\partial \boldsymbol{u}}{\partial t} + \nabla p = 0, \end{gathered} \tag{80}$$

where r is the distance from the boundary point to the center of the piston and θ the angular coordinate.

4.1.2 Stability Analysis

The stability of the linearized Euler equations is investigated by determining the dispersion relation. The linearized Euler equations are written as follows:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{E}}{\partial x} + \frac{\partial \boldsymbol{F}}{\partial y} = 0, \tag{81}$$

where $\boldsymbol{U} = \begin{bmatrix} p \\ u \\ v \end{bmatrix}, \qquad \boldsymbol{E} = \begin{bmatrix} u \\ p \\ 0 \end{bmatrix}, \qquad \boldsymbol{F} = \begin{bmatrix} v \\ 0 \\ p \end{bmatrix}.$

The Fourier transform of a function $f(x, y, t)$ is defined by:

$$\widetilde{f}(\alpha_1,\, \alpha_2,\, \omega) = \frac{1}{(2\pi)^3}\int_0^\infty \int_{-\infty}^\infty \int_{-\infty}^\infty f(x,y,t)e^{-i(\alpha_1 x+\alpha_2 y-\omega t)}dx\ dy\ dt, \tag{82}$$

where $\alpha_1,\, \alpha_2,\, \omega \in \mathbb{R}$,
and its inverse:

$$f(x,y,t) = \int_\Gamma \int_{-\infty}^\infty \int_{-\infty}^\infty \widetilde{f}(\alpha_1,\, \alpha_2,\, \omega)e^{i(\alpha_1 x+\alpha_2 y-\omega t)}d\alpha_1\ d\alpha_2\ d\omega, \tag{83}$$

where Γ is the contour in the complex ω-plane.
By applying Fourier transforms to (81) the following appears:

$$\boldsymbol{A}\widetilde{\boldsymbol{U}} = \widetilde{\boldsymbol{G}}, \tag{84}$$

where $\boldsymbol{A} = \begin{bmatrix} -i\omega & i\alpha_1 & i\alpha_2 \\ i\alpha_1 & -i\omega & 0 \\ i\alpha_2 & 0 & -i\omega \end{bmatrix}$ and $\widetilde{\boldsymbol{G}} = i\frac{\widetilde{\boldsymbol{U}}_{initial}}{2\pi}$.

Tam [10] showed that the dispersion relation can be obtained by setting the eigenvalues of $\boldsymbol{A}$ to zero.

The eigenvalues of $\boldsymbol{A}$ are:

$$\begin{aligned} \lambda_1 &= -i\omega + \sqrt{-\alpha_1^2 - \alpha_2^2}, \\ \lambda_2 &= -i\omega - \sqrt{-\alpha_1^2 - \alpha_2^2}, \\ \lambda_3 &= -i\omega. \end{aligned} \tag{85}$$

Here, the third eigenvalue is the dispersion relation of the vorticity wave. The first two eigenvalues are the eigenvalues, from which the dispersion relation of the acoustic wave can be determined. The dispersion relation is given by:

$$\lambda_1\lambda_2 = 0. \tag{86}$$

So, the dispersion relation is:

$$\omega = \sqrt{\alpha_1^2 + \alpha_2^2}. \tag{87}$$

Right-hand-side is the wavenumber's euclidean length α. For a stability analysis (87) is written as follows:

$$\omega = \alpha, \tag{88}$$

The schemes have been developed such that the wavenumber and angular frequency should equal the numerical wavenumber and numerical angular frequency, respectively. The stability condition of this problem can be obtained after some mathematical manipulation of the numerical dispersion relation:

$$\overline{\omega} = \overline{\alpha} \qquad \Leftrightarrow \qquad \overline{\omega}\Delta t = \overline{\alpha}\Delta x\frac{\Delta t}{\Delta x} \qquad \Leftrightarrow \qquad \overline{\omega}\Delta t = \overline{\alpha}\Delta x\nu,$$

where $\nu = \frac{\Delta t}{\Delta x}$ is the CFL-number.

Popescu [3] showed that $\overline{\alpha}\Delta x < 1.4$ is required for 6/4-scheme for low dispersion error. Criterion $\left|\frac{d(\overline{\alpha}\Delta x)}{d(\alpha\Delta x)} - 1\right| < 3\cdot10^{-3}$ is used to obtain these requirements. This is also illustrated in Figure 3.1 and Figure 3.2. Popescu [3] showed that for numerical stability of the 4-6-LDDRK $\overline{\omega}\Delta t < 2.52$ is required. For accuracy's sake this requirement is replaced by $\overline{\omega}\Delta t < 1.64$ which also satisfies the requirement of negligible numerical dissipation. These results lead to the following stability condition:

$$\overline{\omega}\Delta t = \frac{\Delta t}{\Delta x}1.4 = 1.4\nu \leq 1.64 \qquad \Rightarrow \qquad \nu \leq 1.17.$$

4.2 Application of Richardson Extrapolation

4.2.1 Interpolation

When numerical solutions are determined with different spacings, the grids will, in general, not match. In order to apply Richardson extrapolation several numerical solutions are needed on different grids. As said before, these grids will not match, in general. Richardson extrapolation can only be applied to grids that match. Therefore, an interpolation procedure is needed in order to transform the numerical solutions to one common grid. Shyy and Garbey [11] have addressed that the interpolation procedure could affect the accuracy of the original numerical solution, when it is interpolated to the common grid. So, this could lead to Richardson extrapolation results, which are not reliable. In order to maintain the accuracy of the numerical solution, the interpolation method must have an order of accuracy that is higher than the derived order of accuracy of the methods, which have been used to obtain the numerical solution.

The numerical solutions have an order of accuracy of four. Therefore, the Lagrange polynomials, described in Section 3.4 should be fifth order in x and fifth order in y or higher. For a fifth degree polynomial 36 nodes are needed.

In most of the domain, the rectangular interpolation area can be chosen. For example, for nodes in the interior of the domain and not in the vicinity of the boundaries this area is taken to be the square of the 6×6 surrounding nodes as in Figure 4.2. Here, the closed dot has to be interpolated and the open dots are the cell centers. Only part of the square $[i, j] = [1, 6] \times [1, 6]$ is shown, but it is obvious that the interpolated node is in the center of the square, which means 3 surrounding cell centers in each direction, north, east, south and west.

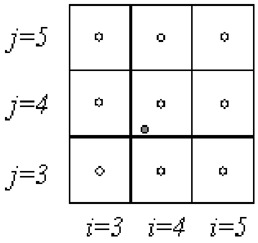

Figure 4.2: Interior interpolation area.

The area in the vicinity of the north, east, south or west is similar. It is also a square, but only difference is that the interpolated node is not in the center of the area. For example, an interpolated node in the vicinity of the east boundary has 2 cell centers east, 4 west, 3 north and 3 south. This is clarified in Figure 4.3. So, here $i = 6$ is on the boundary.

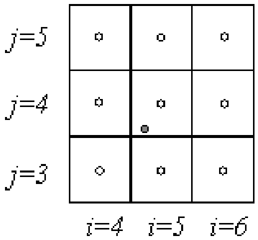

Figure 4.3: Interpolation area in the vicinity of east boundary.

In the vicinity of the wall, it is impossible to find a regularly shaped interpolation area. The procedure will be explained with Figure 4.4.

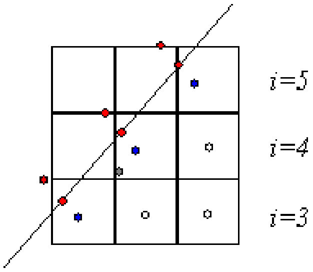

Figure 4.4: Interpolation area in the vicinity of the wall.

In Figure 4.4 the open dots are the cell centers, the blue dots are mass centers of the cut cells and the red dots are virtual points that are used for the finite volume integration procedure, explained in Section 4.1.1.

For the interpolation procedure in the vicinity of the boundary, the interpolation polynomial for general domains, described in Section 3.4.2, is used. There are two possibilities for obtaining 36 conditions in order to determine the interpolation polynomial.

1. Take for $i = 1..6$ the three cell centers east of each mass center and the red and blue nodes. This gives 36 conditions and the polynomial can uniquely be determined.

2. Take for $i = 1..6$ the three cell centers east of each mass center, the blue node, the red node on the wall and the condition $\frac{\partial p}{\partial n} = 0$ on the same red node. The last condition is part of the boundary condition (13b).

With each approach there are 36 conditions in order to determine the interpolation polynomial.

4.2.2 Extrapolation

In order to determine the order of accuracy of the complete method and the impact of the cut-cell method on it, Richardson extrapolation is used. Shyy and Garbey [11] showed that the order of convergence in CFD codes are often space-dependent. Therefore, a pointwise approach is chosen. With a pointwise approach also the impact of the cut-cell method can be detected easier.

Approach 1

Let $P_{ij}^n(\Delta x, \Delta t)$ be the numerical approximation of p_{ij}^n. Here, P is the numerical solution on a fine grid. So a numerical has been interpolated to the fine grid, like described in the previous section, and this solution is called P. It is assumed that $\Delta x = \Delta y$. Since the problem is nonstationary, the error can be written in the following way:

$$\begin{aligned} p_{ij}^n - P_{ij}^n(\Delta x, \Delta t) = \quad & c_1^x \Delta x^{p_1} + c_2^x \Delta x^{p_2} + c_3^x \Delta x^{p_3} + ... \\ & + c_1^t \Delta t^{q_1} + c_2^t \Delta t^{q_2} + c_3^t \Delta t^{q_3} + ... \end{aligned} \tag{89}$$

where:

1. $c_1^x,\, c_1^t,\, c_2^x,\, c_2^t,\, ... \in \mathbb{R}$,
2. $p_1,\, q_1,\, p_2,\, q_2,\, ... \in \mathbb{N}$,
3. $0 < p_1 < p_2 < p_3 < ....$ and
4. $0 < q_1 < q_2 < q_3 < ....$

When Δx and Δt are small, (89) can be approximated by:

$$p_{ij}^n - P_{ij}^n(\Delta x, \Delta t) = C_x \Delta x^p + C_t \Delta t^q. \tag{90}$$

In Section 3.5 is described how to use Richardson extrapolation for one-dimensional discretization. Now, there is spatial discretization and time integration. Therefore, the Richardson extrapolation procedure must be applied twice:

1. Once for decreasing Δx, in order to derive p and C_x.
2. Once for decreasing Δt, in order to derive q and C_t.

In order to obtain an improved solution the most accurate numerical solution must be extrapolated. Therefore, $\frac{\Delta t}{4}$ will be taken for simulations with decreasing Δx and $\frac{\Delta x}{4}$ will be taken for simulations with decreasing Δt. The procedure is described below and is similar to the one-dimensional case.

First, in order to derive p and C_x:

$$\begin{aligned} p_{ij}^n - P_{ij}^n(\Delta x, \tfrac{\Delta t}{4}) &= C_x \Delta x^p + C_t \left(\tfrac{\Delta t}{4}\right)^q, \\ p_{ij}^n - P_{ij}^n(\tfrac{\Delta x}{2}, \tfrac{\Delta t}{4}) &= C_x \left(\tfrac{\Delta x}{2}\right)^p + C_t \left(\tfrac{\Delta t}{4}\right)^q, \\ p_{ij}^n - P_{ij}^n(\tfrac{\Delta x}{4}, \tfrac{\Delta t}{4}) &= C_x \left(\tfrac{\Delta x}{4}\right)^p + C_t \left(\tfrac{\Delta t}{4}\right)^q. \end{aligned} \tag{91}$$

After subtracting the second from the first equation of (91), the third from the second and dividing these expressions, the spatial order of accuracy can be obtained:

$$\frac{P_{ij}^n(\frac{\Delta x}{2},\frac{\Delta t}{4}) - P_{ij}^n(\Delta x,\frac{\Delta t}{4})}{P_{ij}^n(\frac{\Delta x}{4},\frac{\Delta t}{4}) - P_{ij}^n(\frac{\Delta x}{2},\frac{\Delta t}{4})} = 2^p. \tag{92}$$

By subtracting the third from the second equation the coefficient C_x can be obtained:

$$C_x = \frac{P_{ij}^n(\frac{\Delta x}{4},\frac{\Delta t}{4}) - P_{ij}^n(\frac{\Delta x}{2},\frac{\Delta t}{4})}{\left(\frac{\Delta x}{2}\right)^p\left(1-\left(\frac{1}{2}\right)^p\right)}, \tag{93}$$

Second, in order to derive q and C_t:

$$\begin{aligned} p_{ij}^n - P_{ij}^n(\tfrac{\Delta x}{4},\Delta t) &= C_x\left(\tfrac{\Delta x}{4}\right)^p + C_t\Delta t^q,\\ p_{ij}^n - P_{ij}^n(\tfrac{\Delta x}{4},\tfrac{\Delta t}{2}) &= C_x\left(\tfrac{\Delta x}{4}\right)^p + C_t\left(\tfrac{\Delta t}{2}\right)^q,\\ p_{ij}^n - P_{ij}^n(\tfrac{\Delta x}{4},\tfrac{\Delta t}{4}) &= C_x\left(\tfrac{\Delta x}{4}\right)^p + C_t\left(\tfrac{\Delta t}{4}\right)^q. \end{aligned} \tag{94}$$

Note that the third simulation already has been done.
In a similar way the time integration order of accuracy can be obtained:

$$\frac{P_{ij}^n(\frac{\Delta x}{4},\frac{\Delta t}{2}) - P_{ij}^n(\frac{\Delta x}{4},\Delta t)}{P_{ij}^n(\frac{\Delta x}{4},\frac{\Delta t}{4}) - P_{ij}^n(\frac{\Delta x}{4},\frac{\Delta t}{2})} = 2^q, \tag{95}$$

and the coefficient C_t can be obtained:

$$C_t = \frac{P_{ij}^n(\frac{\Delta x}{4},\frac{\Delta t}{4}) - P_{ij}^n(\frac{\Delta x}{4},\frac{\Delta t}{2})}{\left(\frac{\Delta t}{2}\right)^q\left(1-\left(\frac{1}{2}\right)^q\right)}. \tag{96}$$

Now, the first and second step can be combined in order to obtain an improved solution from (90):

$$P_{impr} = P_{ij}^n(\frac{\Delta x}{4},\frac{\Delta t}{4}) + C_x\left(\frac{\Delta x}{4}\right)^p + C_t\left(\frac{\Delta t}{4}\right)^q. \tag{97}$$

Substituting the coefficients C_x and C_t leads to:

$$\begin{aligned} P_{impr} = \quad & P_{ij}^n(\tfrac{\Delta x}{4},\tfrac{\Delta t}{4}) + \frac{P_{ij}^n(\frac{\Delta x}{4},\frac{\Delta t}{4}) - P_{ij}^n(\frac{\Delta x}{2},\frac{\Delta t}{4})}{\left(\frac{\Delta x}{2}\right)^p\left(1-\left(\frac{1}{2}\right)^p\right)}\left(\tfrac{\Delta x}{4}\right)^p + \\ & + \frac{P_{ij}^n(\frac{\Delta x}{4},\frac{\Delta t}{4}) - P_{ij}^n(\frac{\Delta x}{4},\frac{\Delta t}{2})}{\left(\frac{\Delta t}{2}\right)^q\left(1-\left(\frac{1}{2}\right)^q\right)}\left(\tfrac{\Delta t}{4}\right)^q, \end{aligned} \tag{98}$$

which can be written as follows:

$$
\begin{aligned}
P_{impr} = \quad & P_{ij}^{n}(\tfrac{\Delta x}{4}, \tfrac{\Delta t}{4}) + \tfrac{P_{ij}^{n}(\frac{\Delta x}{4}, \frac{\Delta t}{4}) - P_{ij}^{n}(\frac{\Delta x}{2}, \frac{\Delta t}{4})}{(2^{p}-1)} + \\
& + \tfrac{P_{ij}^{n}(\frac{\Delta x}{4}, \frac{\Delta t}{4}) - P_{ij}^{n}(\frac{\Delta x}{4}, \frac{\Delta t}{2})}{(2^{q}-1)}.
\end{aligned} \tag{99}
$$

Here, 2^p and 2^q have already been determined in (92) and (95), respectively.

Approach 2

As in the first approach, let $P_{ij}^{n}(\Delta x, \Delta t)$ be the numerical approximation of p_{ij}^{n}. Here, P is the numerical solution on a fine grid. The error is written in (89). The time step for the time integration is determined by the CFL-number: $\nu = \frac{\Delta t}{\Delta x}$.

The second approach for Richardson extrapolation is based on decreasing the grid spacing and keeping the CFL-number constant, so the time step will automatically be decreased too.

This can be done by the substitution of $\Delta t = \nu \Delta x$ into (89):

$$
\begin{aligned}
p_{ij}^{n} - P_{ij}^{n}(\Delta x, \Delta t) = \quad & c_1^x \Delta x^{p_1} + c_2^x \Delta x^{p_2} + c_3^x \Delta x^{p_3} + ... \\
& + c_1^t (\nu \Delta x)^{q_1} + c_2^t (\nu \Delta x)^{q_2} + ...,
\end{aligned} \tag{100}
$$

which can be written as follows:

$$
\begin{aligned}
p_{ij}^{n} - P_{ij}^{n}(\Delta x, \Delta t) = \quad & c_1^x \Delta x^{p_1} + c_2^x \Delta x^{p_2} + c_3^x \Delta x^{p_3} + ... \\
& + \widehat{c}_1^t \Delta x^{q_1} + \widehat{c}_2^t \Delta x^{q_2} + ...
\end{aligned} \tag{101}
$$

The two parameters, Δx and Δt, in this expression have been reduced to one, Δx. Now, this can be rewritten as:

$$
p_{ij}^{n} - P_{ij}^{n}(\Delta x) = C_1 \Delta x^{m_1} + C_2 \Delta x^{m_2} + ..., \tag{102}
$$

where $C_i \in \mathbb{R}$ and $m_i \in \mathbb{N}$ with $0 \leq m_1 < m_2 < ...$ ($m_1 = \min(p_1, q_1)$, ...).

This is the same expression as (73) in Section 3.5. Now, the exact same procedure as in Section 3.5, can be followed in order to determine the order of accuracy (76) and the improved solution P_{impr} (79).

Validation

By substituting $\Delta t = \nu \Delta x$ expression (100) become ν-dependent. This could affect the determination of the order of accuracy and the improved solution. For example, if $q_1 < p_1$, then the CFL-number could make the term $\widehat{c}_1^t \Delta x^{q_1}$ smaller than or about equal to $c_1^x \Delta x^{p_1}$. In this case, different CFL-numbers lead to different orders of accuracy. So, validation for this approach is that the order of accuracy should be ν-independent.

5 Implementation

The implementation of the interpolation procedure from the simulation grid to a common grid will be explained and problems will be addressed in this section. Furthermore, an additional method and a description of adaptations made in the code are presented.

5.1 Interpolation

5.1.1 Structure

For every simulation the common grid M^0 has to be defined. The numerical solution on the computational grid M^1 has to be interpolated to the grid M^0. In order to keep the extra computational work of the interpolation procedure modest, a lot of information can be calculated once, before the simulation. The following can be calculated once:

1. For every point on grid M^0 the corresponding point on grid M^1 is determined. This corresponding point is the closest point in north-east direction. So, in Figure 4.2 this corresponding point is $(i, j) = (4, 4)$. It is obvious that this point determines the interpolation area, which is the following issue's objective. This subroutine also determines whether or not interpolation is needed. There is a possibility that the point of grids M^0 and M^1 match. So, in this case interpolation is not needed, because the value is known at that point.
2. For every point on grid M^1 the type of interpolation is determined. Since this point is the center of the interpolation area for the corresponding point in M^0, this subroutine will determine whether Lagrangian interpolation on a square (in the interior) or interpolation on an irregular domain (in the vicinity of the wall) is needed. If Lagrangian interpolation on a square is needed, the south-west corner of this square is determined. If interpolation on an irregular domain is needed, the interpolation matrix (72) and the coordinates of the right-hand side are determined. The procedure of finding the interpolation area is described in Section 4.1.2.
3. For every point that needs interpolation on a irregular domain, a matrix is determined. This matrix must be inverted in order to determine the coefficients of the interpolation polynomial. So, all matrices are inverted once.

After these issues the simulation can be started. When the pressure has been calculated on the computational grid M^1 it can be interpolated very easily to the common grid M^0 with the information that has been calculated once.

5.1.2 Adaptations

After the implementation of the procedure described in Section 4.2.1 the results were not satisfactory. Especially the interpolation in the vicinity of the wall performed unsatisfactorily. Several adaptations to the proposed procedure in Section 4.2.1 has been made in order to solve these problems.

Shifting the Interpolation Procedure

In order to determine the coefficients of the interpolation polynomial a system has to be solved. The matrices look like (72) with $N_x = N_y = 5$. Some examples of x- and y-coordinates in the vicinity of the wall are: $(x, y) = (-5, 2), (-2, 8), (1, 14)$. Because the interpolation polynomial is of degree 5 in x and y, the elements of the matrices can vary from $O(10^7)$ to $O(1)$, which leads to round-off errors. In order to overcome this problem two options of shifting the interpolation procedure are tried. The interpolation procedure is shifted to $(x, y) = (1, 1)$ or to $(x, y) = (0, 0)$. So, every (x_i, y_j), used for interpolation, was shifted around these coordinates. After shifting the interpolation procedure to $(x, y) = (1, 1)$ the results are better, but after shifting it to $(x, y) = (0, 0)$ the results are much better. However, it is still unsatisfactory.

The basic idea of shifting the interpolation procedure is that the elements of the matrices can be stored more exactly, which leads to smaller round-off errors. By shifting the procedure to the origin, the elements of the matrices contain much less digits than with the original coordinates and the elements are smaller than 1. So this has a big impact on the round-off errors.

Row Scaling

Actually, matrix (72) is a two-dimensional Vandermonde-matrix and it is well-known that this matrix can be ill-conditioned. In order to improve the conditioning of these matrices, row scaling will be applied to the matrices. Every element of a row is divided by the sum of the elements of that row. The results with row scaling are slightly better than without row scaling.

Note that row scaling can be done once, before the simulation.

Choosing Points

It appears that the choice of the interpolation points has the greatest influence on the results. In Section 4.2.1 two options are described. First option is to choose the mass center, two virtual points and three cell centers in each row of the domain and second option is to choose the virtual point on the wall, the wall condition $\frac{\partial p}{\partial n} = 0$, the mass center and three cell centers in each row of the domain in order to obtain 36 conditions to determine the coefficients of the interpolation polynomial. After implementation of both options it appeared that the results of option 2 were much better, but still unsatisfactory. So, the wall condition $\frac{\partial p}{\partial n} = 0$ has a big impact on the interpolation results.

In this approach the mass centers of the cut cells are chosen to be the cell centers of the former complete cells. So, with this approach the mass centers can be very close to the wall. This leads to ill-conditioned matrices for interpolation. Also taking the opposite (virtual) point of the wall in stead of the point on the wall when the mass center is close to the wall, does not give better results.

When the mass center is very close to the wall, the opposite point of the wall is also very close to the wall, which still leads to ill-conditioned matrices.

After taking the wall condition $\frac{\partial p}{\partial n} = 0$, the mass center and four cell centers east of the mass centers in each row of the domain, the result seems to be satisfactorily. The choice of the points is shown in Figure 5.1. So, the wall condition $\frac{\partial p}{\partial n} = 0$ on the red dots, the blue dots and four open dots are taken for the interpolation procedure. Results are shown in the next section.

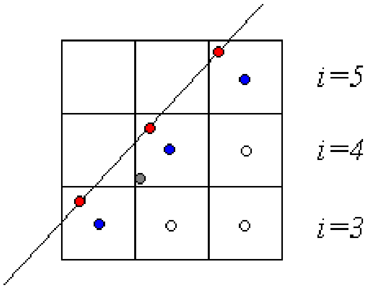

Figure 5.1: Final choice of the interpolation area in the vicinity of the wall.

5.1.3 Analytical Testcase With Results

In order to test this final approach, a given function will be interpolated on this domain. This function is known, so the quality of the interpolation procedure will be determined by comparing the interpolated and the analytical solution on the grids. The function must be chosen such that it satisfies the wall condition $\frac{\partial p}{\partial n} = 0$. A function that satisfies this condition is $f(x') = C_1 \cos(x')$, with x' the orthogonal distance to the wall and $C_1 \in \mathbb{R}$. The constant C_1 will be chosen such that it has similar amplitudes to the sound waves in the problem.

The function $f(x')$ with $C_1 = 10^{-3}$ is shown in Figure 5.2.

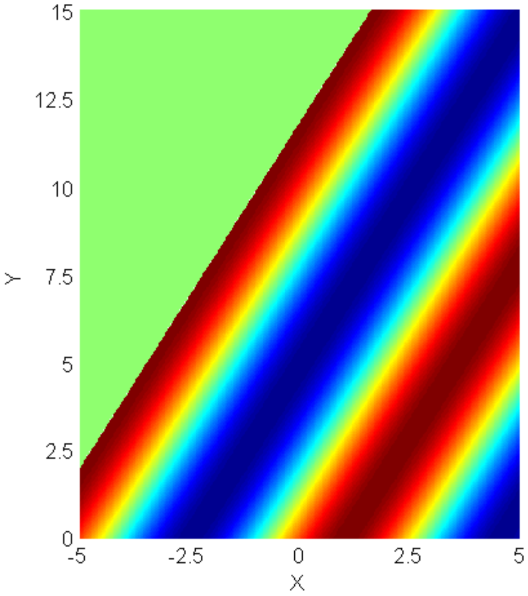

Figure 5.2: $f(x') = C_1 \cos(x')$ on a grid with $\Delta x = 0.025$.

In order to determine the order of accuracy of the interpolation method, this function must be interpolated from several grids with different grid spacings to common grid M^0. The common grid M^0 has been chosen with $\Delta x = 0.025$.

Figure 5.3 shows $f(x')$ on a grid with $\Delta x = 0.1$ and the interpolated function on grid M^0.

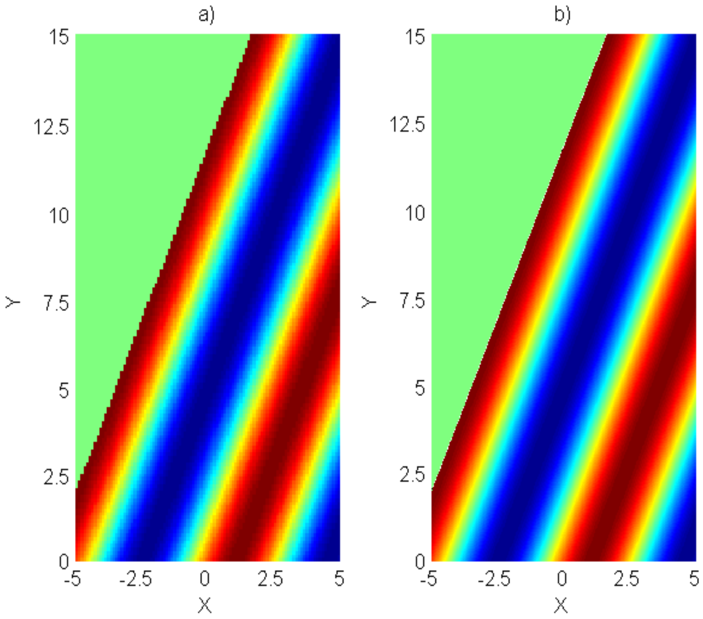

Figure 5.3: a) $f(x')$ on a grid with $\Delta x = 0.1$. b) Interpolated function on M^0.

The error will be determined by using the infinity norm. By repeating this procedure with different Δx, Figure 5.4 can be made.

Figure 5.4 shows the plot from which the order of accuracy of the complete final interpolation method can be derived.

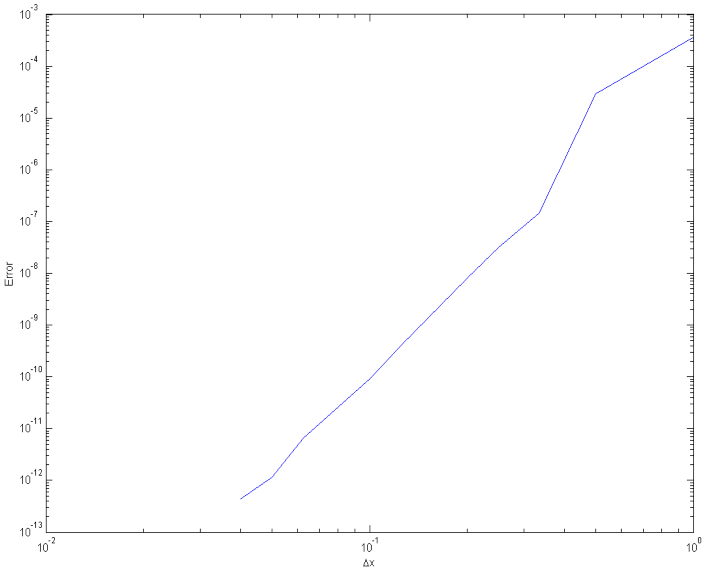

Figure 5.4: Plot of the error with logarithmic scale. The error has been determined by the infinity norm.

It appears that the order of accuracy is about 6. For further investigation of the performance of the complete interpolation method, the orders of accuracy of both used methods are determined seperately and extended to a larger range of Δx. The resulting plot is shown in Figure 5.5.

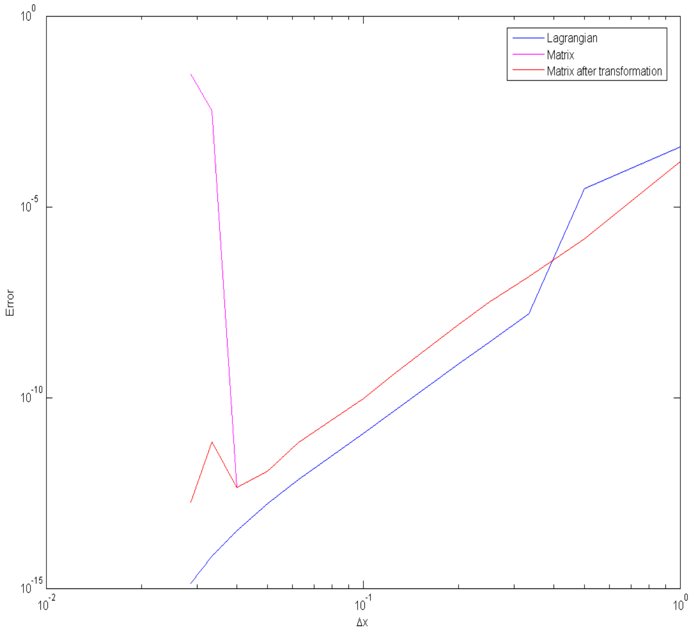

Figure 5.5: Plot of the error of both methods with logarithmic scale. The error has been determined by the infinity norm.

Taking the maximum of the two errors leads to Figure 5.4. It appears that both methods have an order of accuracy of about 6.

From Figure 5.5 it appears that when the grid becomes finer the matrix interpolation method will fail (purple). The points used for interpolation are too close to each other which leads to ill-conditioned matrices. Regarding Section 5.1.2, this has been expected. If interpolation from such a fine grid is needed, this problem can be solved by a simple transformation, which spreads the interpolation points. The results of this transformation are shown in Figure 5.5 (red).

5.2 Extrapolation

In order to obtain numerical solutions, a code has been made by Shyy, Popescu and Tai [4]. This code is used and extended for this thesis. In this section an additional method, the filter, and the main adaptations made in the code will be discussed. This is all necessary in order to be able to apply Richardson extrapolation.

5.2.1 Filter

In computational aeroacoustics usage of a filter is common in order to filter the numerical solution. The numerical solution can produce some high frequency noise, which is a numerical artefact. This high frequency noise can be handled and removed by a filter. The filter, used in this code, is written as follows:

$$f_i^{filt} = f_i + \frac{1}{5}\left[-\frac{1}{4}f_{i-2} + f_{i-1} - \frac{3}{2}f_i + f_{i+1} - \frac{1}{4}f_{i+2}\right]. \tag{103}$$

This one-dimensional filter is applied in both directions.

By applying Fourier transforms (19) and its shift and derivative theorems (21) the filtering proces can be shown. The Fourier transformation is given below:

$$\widetilde{f}_i^{filt} = \widetilde{f}_i + \frac{1}{5}\left[-\frac{1}{4}(e^{-2i\alpha\Delta x} + e^{2i\alpha\Delta x}) + (e^{-i\alpha\Delta x} + e^{i\alpha\Delta x}) - \frac{3}{2}\right]\widetilde{f}_i, \tag{104}$$

which can be written as follows:

$$\widetilde{f}_i^{filt} = \frac{1}{5}\left[-\frac{1}{2}\cos(2\alpha\Delta x) + 2\cos(\alpha\Delta x) + \frac{7}{2}\right]\widetilde{f}_i. \tag{105}$$

So, it can be seen that the Fourier transform of the filtered solution is the Fourier transform of the non-filtered solution multiplied by $\frac{1}{5}\left[-\frac{1}{2}\cos(2\alpha\Delta x) + 2\cos(\alpha\Delta x) + \frac{7}{2}\right]$. This multiplication factor is plotted in Figure 5.6.

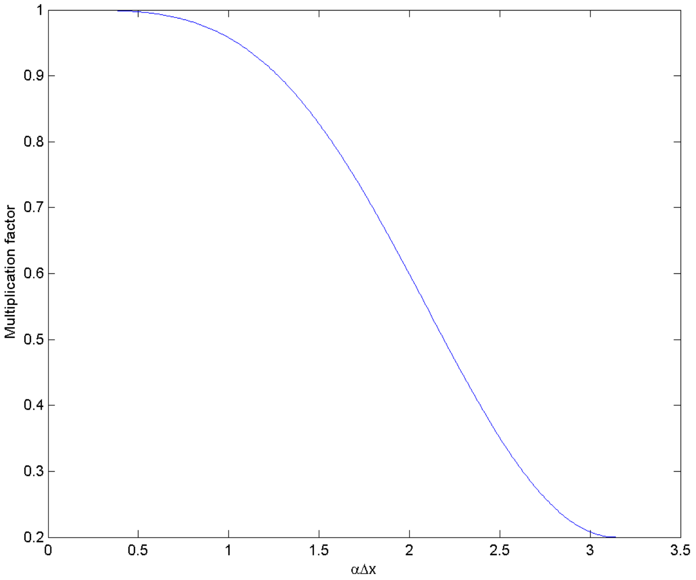

Figure 5.6: $\alpha\Delta x$ versus multiplication factor.

The multiplication factor is a function of the wavenumber. It is obvious that higher wavenumbers are filtered out of the solution, because they are damped by the filter. By the dispersion relation, it can be seen that high wavenumbers imply high frequencies. The lower frequencies are multiplied by 1, so the numerical solution does not contain the high frequency noise.

By expanding (103) in a Taylor series it appears that filtered solution still has an order of accuracy of four.

Obviously, (103) cannot be implemented near the boundaries. So, also for boundary cells filter expressions have been developed. The following boundary filters are used:

$$f_i^{filt} = f_i + \frac{1}{5}\left[f_{i-1} - f_i\right], \tag{106a}$$

$$f_i^{filt} = f_i + \frac{1}{5}\left[f_{i-1} - 2f_i + f_{i+1}\right]. \tag{106b}$$

Boundary filter (106a) is applied on the boundary points and therefore, it is an one-sided filter. Boundary filter (106b) is applied on the points next to the boundaries and still is a symmetrical filter. By expanding these filters in Taylor series, it appears that the boundary filters have first and second order of accuracy, respectively.

5.2.2 Code

The code written by Shyy, Popescu and Tai [4], has been used for this thesis. However, adaptations have been made in the code in order to make the code suitable for the application of interpolation and Richardson extrapolation. The geometry of the problem has been fixed and the filter has been implemented correctly. Main adaptations made in the code are described in the code document in Appendix B.

6 Results

In this chapter results obtained by Shyy, Popescu and Tai [4] are presented in the first section. In the next section results obtained in this project are presented. These results are structured as a process.

6.1 Previous Results

The results of Shyy, Popescu and Tai [4] are given in Figure 6.1. The angle β, which defines the solid wall, has been taken $\beta = 63^o$. Contourplots at different times have been plotted. At $t = 10$ can clearly be seen that the wall reflected the sound wave, which results in the local extrema.

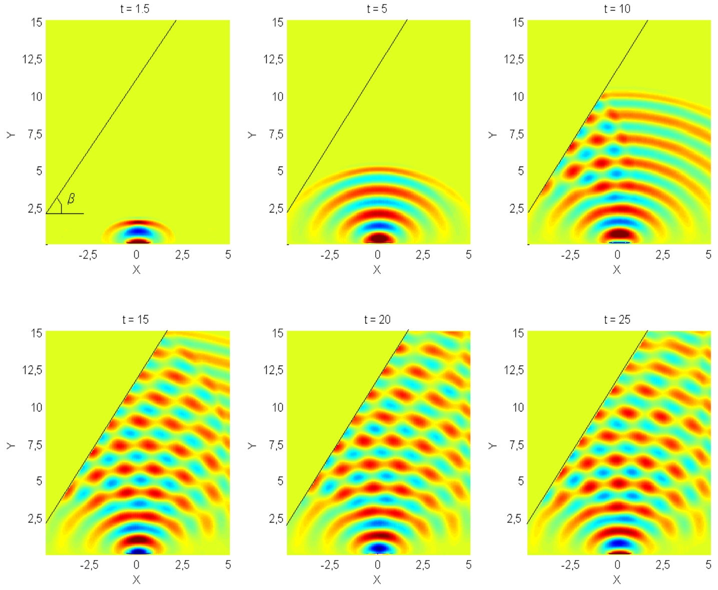

Figure 6.1: Contourplots of pressure. $CFL = 0.5$ and $\Delta x = 0.05$.

Popescu, Shyy and Tai [4] concluded that, based on this testcase, the presented approach can be effective for acoustic problems with complex geometry. Also they concluded that the computational overhead of the cut-cell method is modest, because many computations in the vicinity of the boundary have to be done once (because the geometry is fixed).

Results of further investigation of this testcase can be found in the next section.

6.2 New Results

In this section results obtained for this thesis are presented. First, the cause of the low order of accuracy is investigated. Second, a validation of the used approach is given. Third, some adaptations in the code and in the approach are described and explained and after this, the impact of the outflow boundary condition and cut-cell method is investigated. Finally, Richardson extrapolation is investigated.

6.2.1 Order of Accuracy

After adaptations made to the code, the results that are obtained with $\Delta x = \Delta y = 0.05$ and $CFL = 0.5$ in order to determine the time step, are shown in Figure 6.2.

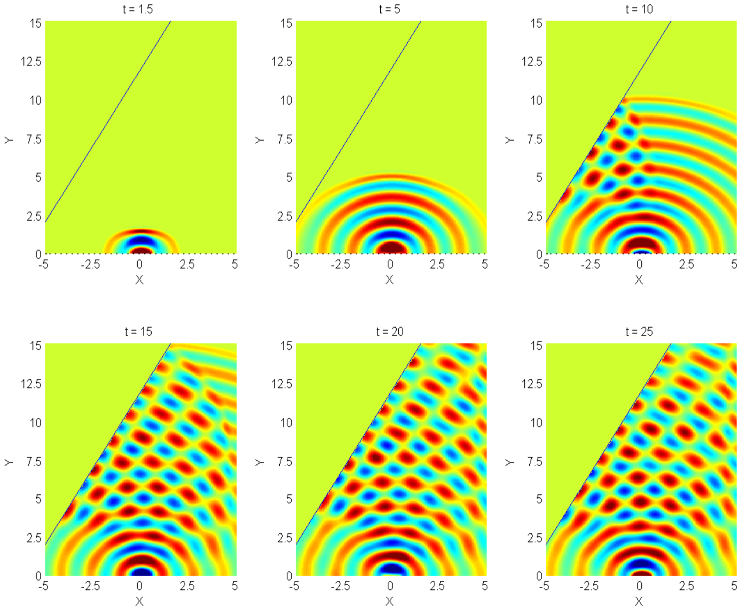

Figure 6.2: Contourplots of pressure. $CFL = 0.5$ and $\Delta x = 0.05$.

The differences in the solutions of Figure 6.1 and 6.2 have been discussed in Section 5.2.2 and Appendix B. These solutions converge for decreasing Δx with a CFL-number kept constant and for a decreasing CFL-number with Δx kept constant. This is shown in Figure 6.3, in which the slice $x = 0$ at $t = 4.2$ is plotted.

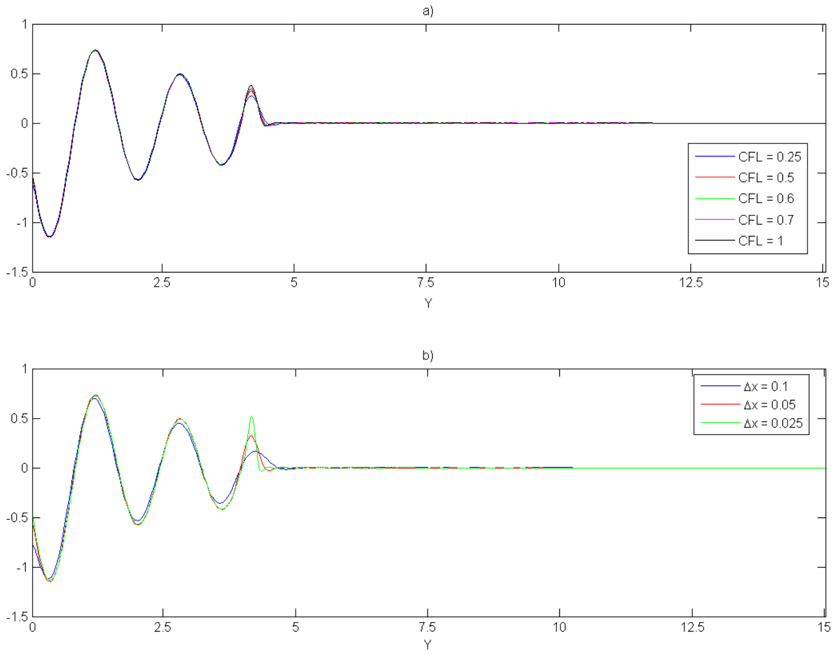

Figure 6.3: a) Solution slice $x = 0$ with decreasing CFL-numbers. b) Solution slice $x = 0$ with decreasing Δx. Both at $t = 4.2$.

In order to determine the order of accuracy Approach 2 from Section 4.2.2 is used. This approach has two main advantages: it reduces the number of parameters from two to one and therefore, it needs only three numerical solutions.

Three numerical solutions have to be obtained and interpolated to the common grid M^0. In order to determine the impact of the cut-cell procedure, first the order of accuracy is determined before the waves hit the wall. For the three simulations the CFL-number $= 0.5$ and the grid spacings are $\Delta x = 0.1$, $\frac{\Delta x}{2} = 0.05$ and $\frac{\Delta x}{4} = 0.025$. The common grid M^0 has grid spacing $\Delta x = 0.025$, so the finest grid solution does not need interpolation.

In order to reduce the impact of the filter on the solution, the filter is applied in the three simulations only on the coarsest time grid of the three simulations. So, each numerical solution will be filtered the same number of times.

Like Shyy and Garbey [11], the order of accuracy will not be determined at grid points where $P(\frac{\Delta x}{4}) = P(\frac{\Delta x}{2})$ within round-off error.

In Figure 6.4 a numerical solution and the order of accuracy is plotted.

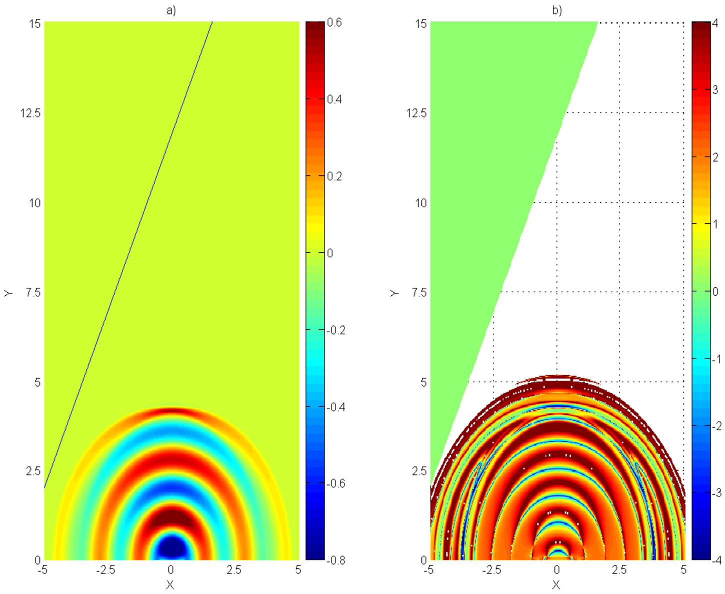

Figure 6.4: a) Numerical solution with grid spacing $\frac{\Delta x}{4}$ and $CFL = 0.5$. b) Order of accuracy.

It is obvious that the order of accuracy is space-dependent and that at some locations the order of accuracy is very poor.

Grid Refinement

In order to find out the cause of this poor convergence, grid refinement is applied. The order of accuracy is determined with three finer grid solutions. This grid refinement is applied twice. So, the order of accuracy is determined with the numerical solutions $P(\frac{\Delta x}{2})$, $P(\frac{\Delta x}{4})$ and $P(\frac{\Delta x}{8})$ and the order of accuracy is determined with even finer grid solutions $P(\frac{\Delta x}{4})$, $P(\frac{\Delta x}{8})$ and $P(\frac{\Delta x}{16})$. This refinement ensures that dispersion, dissipation and the multiplication factor of the filter do not affect the solution accuracy. Moreover, this refinement gives a better approximation of the asymptotic order of accuracy.

The common grid will have the grid spacing of the finest grid solution. Before these simulations are done, it is checked with the testcase, described in Section 5.1.3, that interpolation does not affect the numerical solution.

Results of the grid refinement are shown in Figure 6.5.

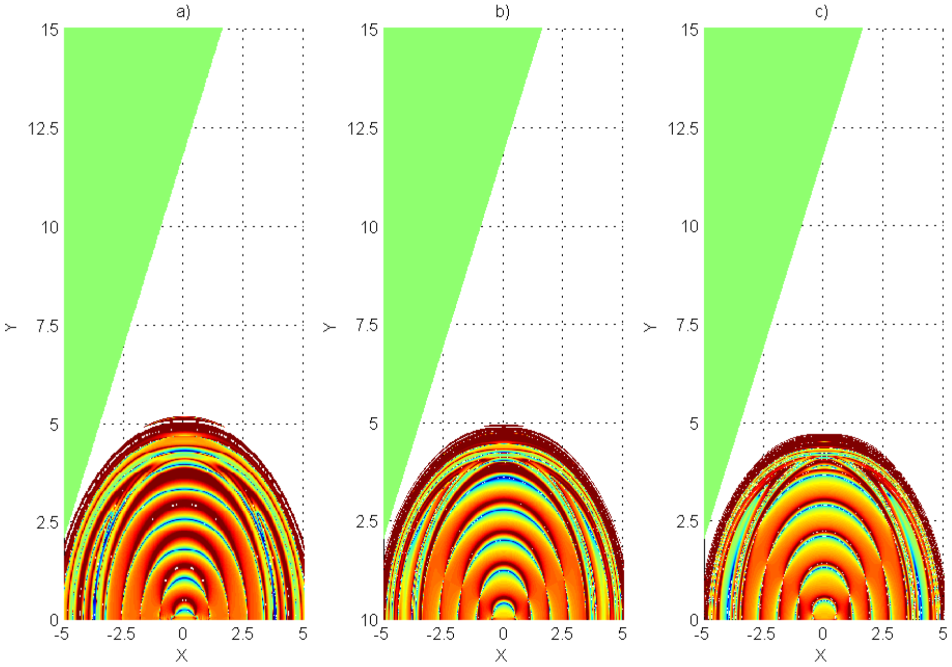

Figure 6.5: Order of accuracy obtained with a) $P(\Delta x)$, $P(\frac{\Delta x}{2})$ and $P(\frac{\Delta x}{4})$. b) $P(\frac{\Delta x}{2})$, $P(\frac{\Delta x}{4})$ and $P(\frac{\Delta x}{8})$. c) $P(\frac{\Delta x}{4})$, $P(\frac{\Delta x}{8})$ and $P(\frac{\Delta x}{16})$.

From Figure 6.5 it appears that the positive and negative peaks in the order of accuracy converge to singularities, which is illustrated more clearly in Figure 6.6.

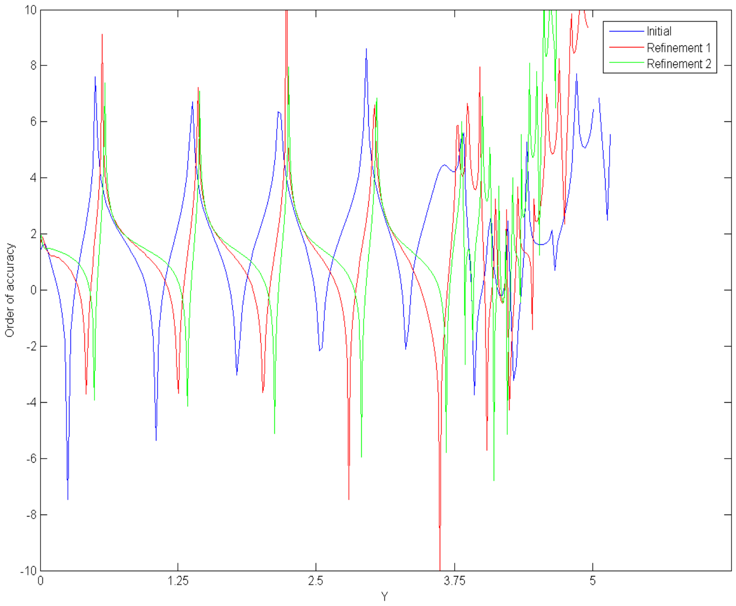

Figure 6.6: Slices $x = 0$ of the orders of accuracy in Figure 6.5.

Moreover, from Figure 6.6 it also appears that the order of accuracy converges to two, which is much less than the derived fourth order. The area between $y = 3.75$ and $y = 5$, in which the order of accuracy is very unstructured, is caused by the discontinuity in the boundary condition at $t = 0$.

In Figure 6.7 the slice $x = 0$ of the order of accuracy of the first refinement is plotted with the corresponding solutions that are used to determine the order of accuracy. The order of accuracy in Figure 6.7 b), is the slice that is plotted in Figure 6.6 (red).

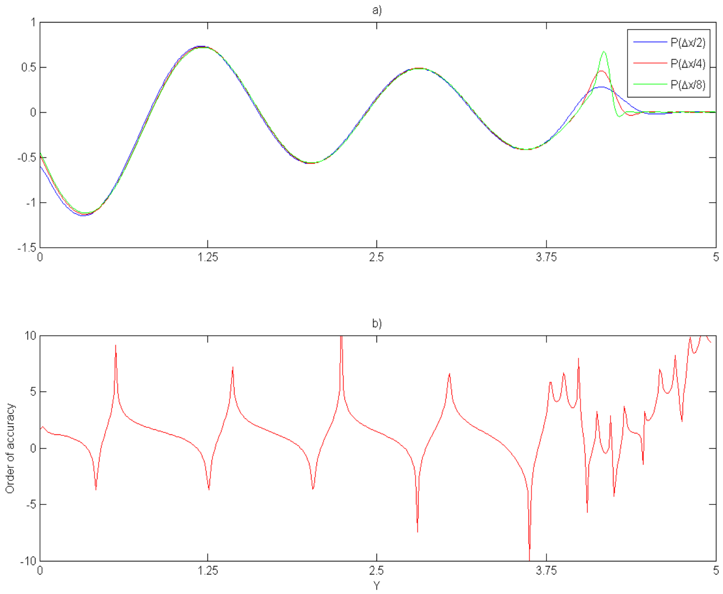

Figure 6.7: a) Slice $x = 0$ of solutions. b) Slice $x = 0$ of the order of accuracy.

In Figure 6.7 it can be seen at which locations in the solutions the singularities in the order of accuracy appear. It looks like when the solutions are close, there appears a singularity. Obviously, this phenomenon needs more investigation.

In order to find out more about the effect of the filter, the singularities, its locations and the low order of accuracy, some related problems will be investigated in the next sections.

Acoustic Pulse

The problem of the acoustic pulse is defined in a similar way. The geometry is almost the same, which means that there are outflow boundary conditions on all the boundaries, and the wall is at the same location. There is no baffled piston around the origin, but the initial condition is an acoustic pulse, defined by:

$$p(x, y, 0) = \exp\left(-\ln(2)\left[\left(\frac{x - x_0}{b}\right)^2 + \left(\frac{y - y_0}{b}\right)^2\right]\right), \qquad (107)$$

where (x_0, y_0) is the location of the pulse, which is chosen to be $(x_0, y_0) = (0, 5)$ and $b = \frac{1}{5}$.

Pulse without Filter

The problem of the pulse has been chosen because propagation of the pulse can be simulated without using the filter. The order of accuracy can be determined in the exact same way. Three numerical solutions have to be interpolated to the common grid M^0. The common grid is chosen with grid spacing $\frac{\Delta x}{8}$, where $\Delta x = 0.1$ as in previous sections. The three numerical solutions are $P(\frac{\Delta x}{2})$, $P(\frac{\Delta x}{4})$ and $P(\frac{\Delta x}{8})$. The order of accuracy at different times are given in Figure 6.8.

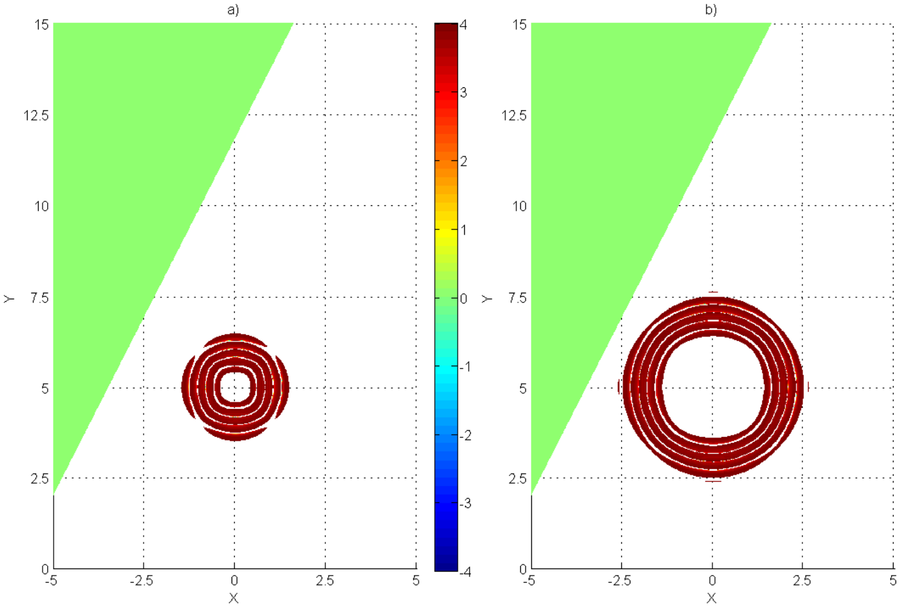

Figure 6.8: Order of accuracy of the propagation of an acoustic pulse at a) $t = 1$, b) $t = 2$.

The slices $x = 0$ are given in Figure 6.9.

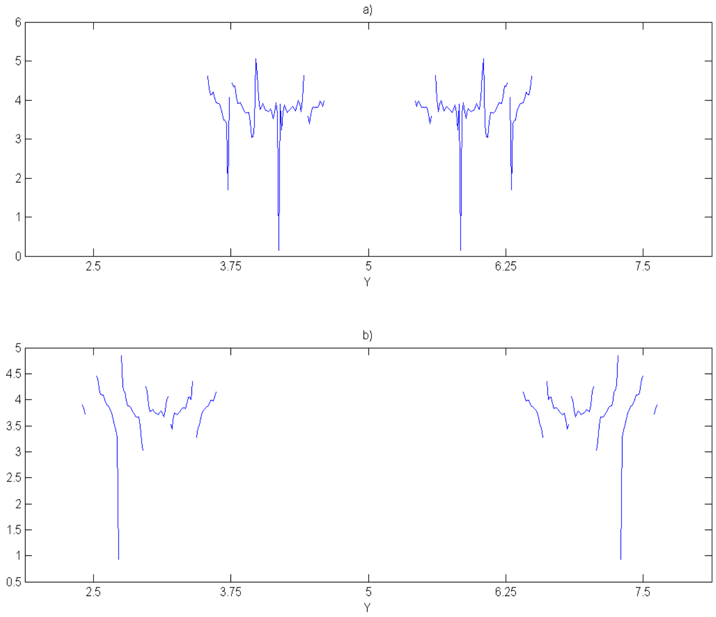

Figure 6.9: Slices $x = 0$ of order of accuracy at a) $t = 1$, b) $t = 2$.

The slices at both times show that the order of accuracy is about four and that there are also singularities, which are discontinuous. In the next section the results of simulations of the pulse with application of the filter, are shown.

Pulse with Filter

Now, the pulse is simulated and the filter is also applied. Again, the filter is applied on the coarsest time grid and the parameters Δx and CFL are the same. In Figure 6.10 the order of accuracy is shown.

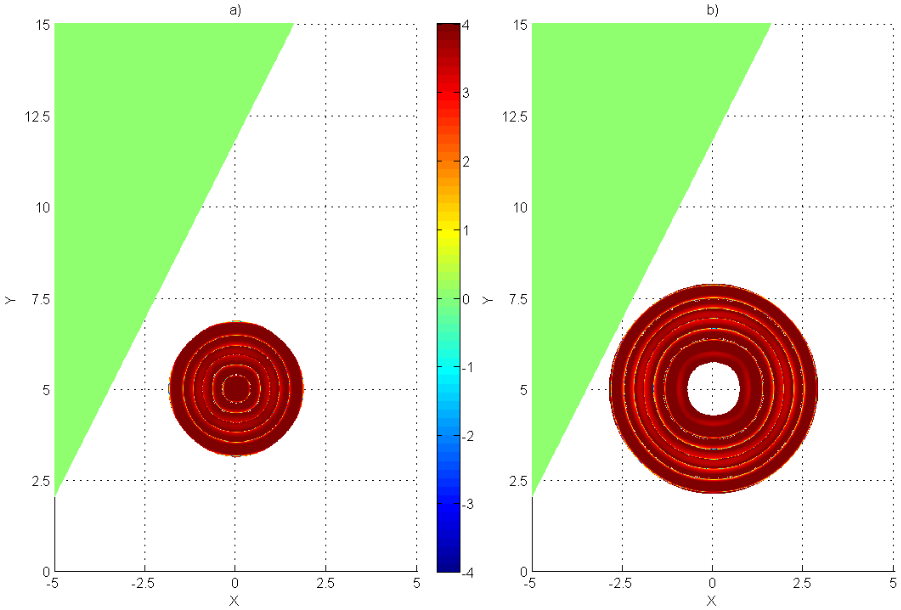

Figure 6.10: Order of accuracy of the propagation of an acoustic pulse with filter at a) $t = 1$, b) $t = 2$.

The slices $x = 0$ are given in Figure 6.11.

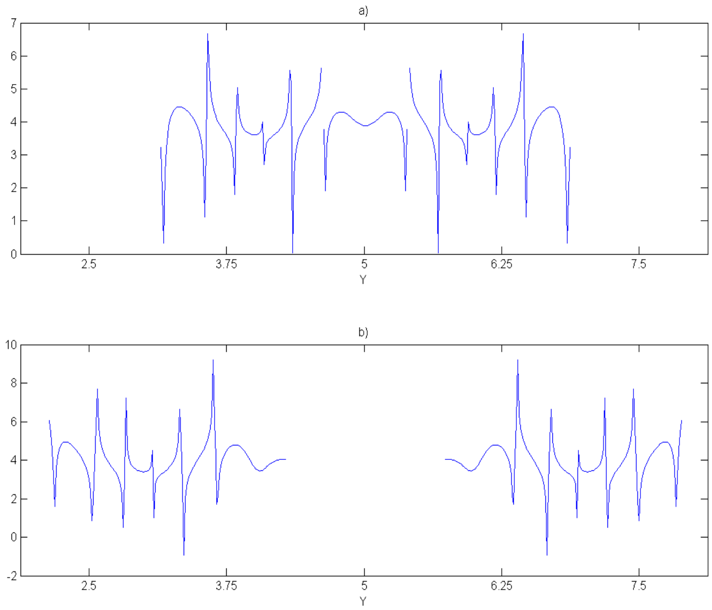

Figure 6.11: Slices $x = 0$ of order of accuracy at a) $t = 1$, b) $t = 2$.

From Figure 6.9 and 6.11 it appears that the use of the filter does not only smooth the solution but also the order of accuracy. Furthermore, it makes the singularities in the order of accuracy continuous. So, the filter makes the differences between the solutions larger, because when the filter is not applied, the order of accuracy will not be determined at the locations of the singularities. Thus, the (continuous) singularities are a result of the phenomenon that $P(\frac{\Delta x}{8}) = P(\frac{\Delta x}{4})$ within roundoff error. This is conform with Shyy and Garbey [11] and can also be confirmed by Figure 6.7.

Most important conclusion of this testcase is that the filter does not affect the magnitude of the order of accuracy. The order of accuracy is still about four.

Boundary Condition

The original problem, posed in Section 2.2, has boundary condition (13a) on the south boundary of the computational domain. The piston is defined by: $(x, y) = \big([-\frac{1}{2}, \frac{1}{2}], 0\big)$. This means that there are two discontinuities at $x = -\frac{1}{2}$ and $x = \frac{1}{2}$, because of the definition of the boundary condition. Moreover, at $t = 0$ the boundary condition leads to another discontinuity at $(x, y) = \big([-\frac{1}{2}, \frac{1}{2}], 0\big)$ in the initial condition. The boundary condition at $t = 0$ gives $v = V_0$ at the piston.

In order to determine the impact of these discontinuities on the order of accuracy, the problem is adapted to a problem without discontinuities. Therefore, boundary condition (13a) is replaced by the following boundary condition:

$$v(x,0,t) = \begin{cases} V_0 \sin(\omega t)\left[\frac{1}{2}\cos(2\pi x) + \frac{1}{2}\right], & (x,0) \in piston \\ 0, & otherwise \end{cases} \tag{108}$$

where time-dependent sine leads to continuity in the initial condition and the space-dependent cosine to continuity at $x = -\frac{1}{2}$ and $x = \frac{1}{2}$.

Again, these simulations are interpolated to the common grid M_0 with grid spacing $\frac{\Delta x}{8}$. The numerical solutions $P(\frac{\Delta x}{2})$, $P(\frac{\Delta x}{4})$ and $P(\frac{\Delta x}{8})$ are filtered on the coarsest time grid. The solution and order of accuracy are shown in Figure 6.12.

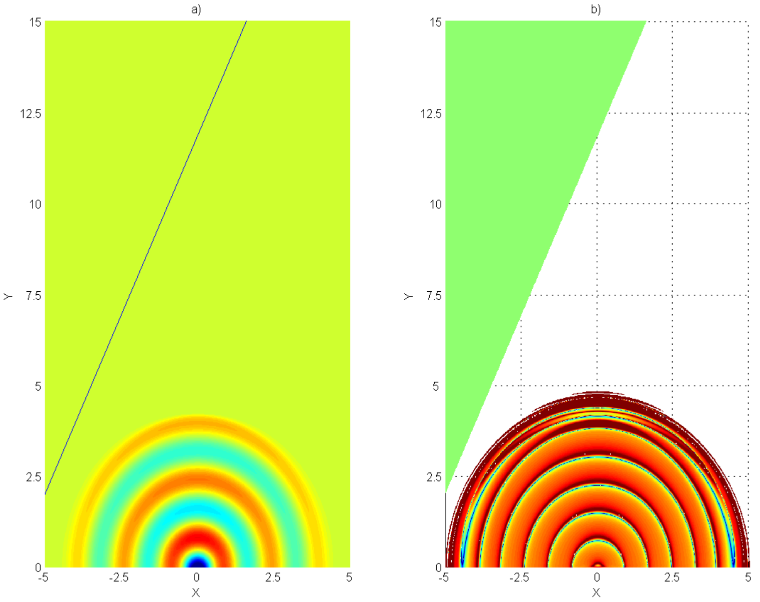

Figure 6.12: a) Solution at $t = 4.2$. b) Order of accuracy at $t = 4.2$.

From Figure 6.12 it appears that the singularities in the order of accuracy are smaller and shaped more circular than the singularities in Figure 6.5b, which is caused by the continuities at $x = -\frac{1}{2}$ and $x = \frac{1}{2}$. At the wave front the order of accuracy is much more structured, which is caused by the continuity at $t = 0$. The discontinuity at $t = 0$ leads to a peak at the wave front, which becomes higher when the grid spacing Δx become smaller. With the smoothing of the filter this leads to the unstructured order of accuracy at the wave front in the original problem.

However, the magnitude of the order of accuracy does not improve significantly.

Boundary Filter

The boundary filters are different than the filter in the interior and the filter (106a) that is applied on the boundary, is not symmetrical. Both boundary filters are lower order filters, which could affect the order of accuracy. This will be tested in the next sections.

Boundary Condition with Discontinuities

Boundaries of the numerical solutions of the original problem, posed in Section 2.2, are filtered with a lower order filter. However, this boundary filter is not needed to obtain a smooth numerical solution without high frequency noise. Therefore, the impact of the boundary filter can be tested. Simulations are interpolated to common grid M^0 with grid spacing $\frac{\Delta x}{8}$ and the filtering, without boundary filtering, is done on the coarsest time grid. The result of the order of accuracy, compared with the result with boundary filtering, is shown in Figure 6.13.

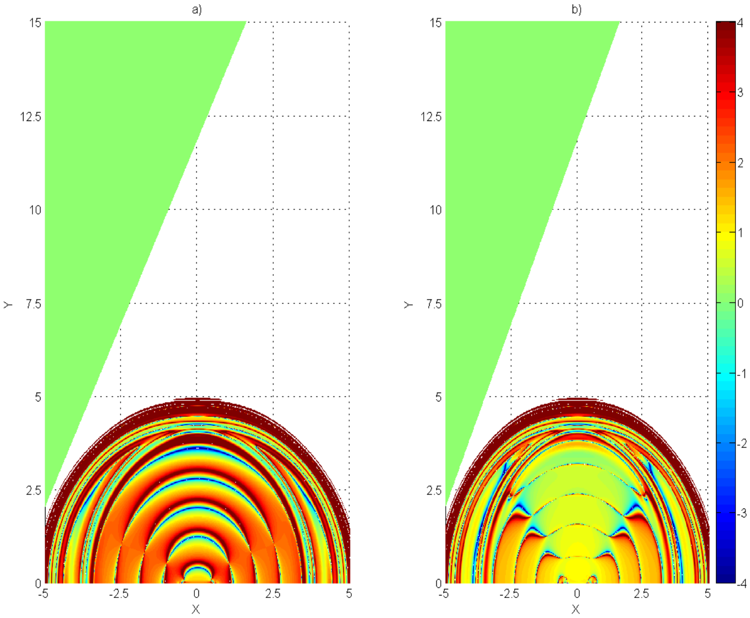

Figure 6.13: a) Order of accuracy with boundary filtering. b) Order of accuracy without boundary filtering.

Obviously, the result of not filtering the boundaries is even worse than the result of filtering the boundaries with a lower order filter. The magnitude of the order of accuracy is about one order lower. The singularities are also shaped differently.

So, filtering the boundaries of a numerical solution, defined by discontinuous boundary and initial conditions, influences the order of accuracy in a positive way. One may conclude that no boundary filtering in combination with discontinuous boundary and initial conditions will make the order of accuracy even lower.

Boundary Condition without Discontinuities

The same procedure can be done for the adapted problem without discontinuities in boundary and initial conditions. This means that boundary condition (13a) is replaced by (108). The same common grid is used and the filter, without boundary filtering, is applied on the coarsest time grid. The order of accuracy, compared with the order of accuracy with filtering the boundary, is shown in Figure 6.14.

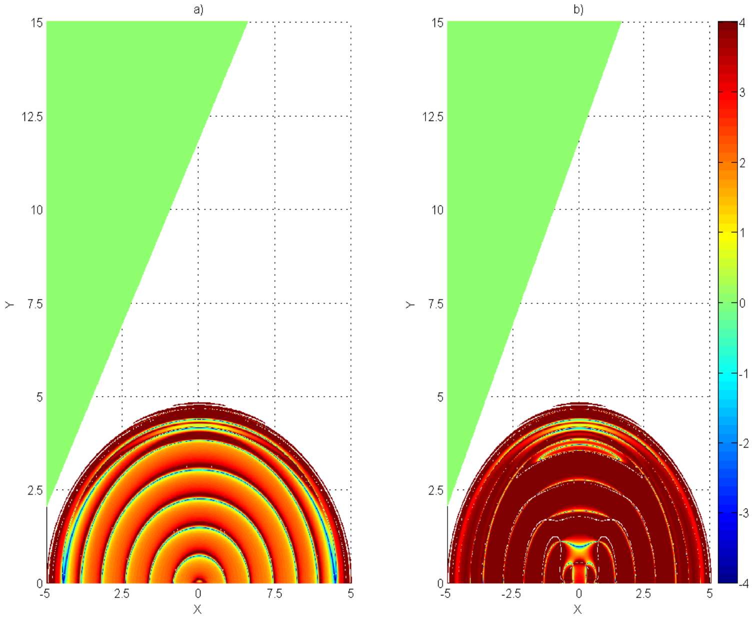

Figure 6.14: a) Order of accuracy with boundary filtering. b) Order of accuracy without boundary filtering.

From Figure 6.14 it appears that the boundary filtering affects the order of accuracy in a negative way. The magnitude of the order of accuracyof the adapted problem is about four, which is the derived order of accuracy.

So, lower order boundary filtering affects the order of accuracy badly in a problem without discontinuities. While lower order boundary filtering affects the order of accuracy in a positive way in a problem with discontinuities. Boundary filtering is not needed in order to obtain satisfactory results, when the problem is continuous.

6.2.2 Validation

In order to investigate the impact of the cut-cell method the adapted problem will be used. The continuous problem leads to a magnitude of the order of accuracy of about four.

For the validation of this approach the order of accuracy should be CFL-independent. In Figure 6.15 the order of accuracy is plotted with two different CFL-numbers.

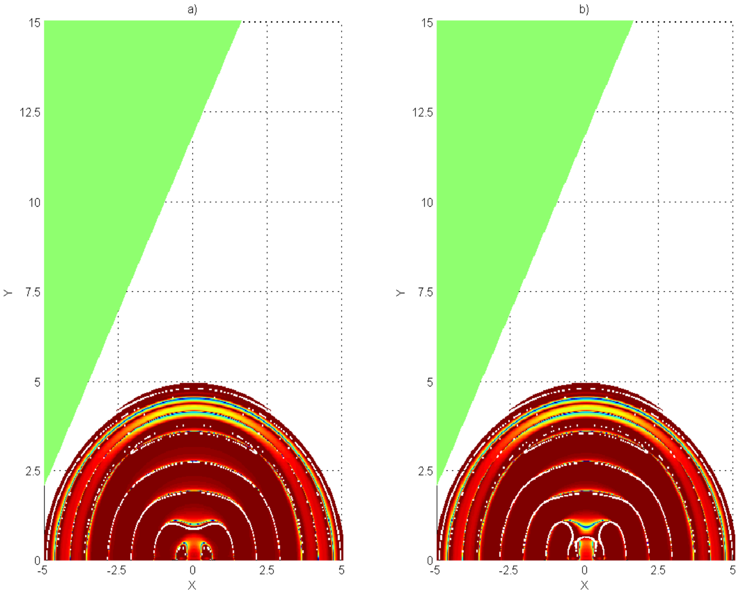

Figure 6.15: Order of accuracy at $t = 4.2$ with a) $CFL = \frac{1}{4}$. b) $CFL = \frac{1}{2}$.

From Figure 6.15 it appears that the order of accuracy has the same shape and values for different CFL-numbers. If it would not have been CFL-independent, the order of accuracy would have had different shape and different values. So, this approach is CFL-independent and can be taken.

6.2.3 Code

Some adaptations in the code and approach has been made in order to obtain better results. These adaptations are described in this section.

Coarser Grids

From simulations it appears that the solution $P(\frac{\Delta x}{4})$ does not give good results when the wave is reflected on the wall. This phenomenon will be discussed later in this section. So, a coarser grid must be used in order to obtain reliable results. The simulation with grid spacing $\frac{\Delta x}{4}$ is replaced by a simulation with grid spacing $\frac{\Delta x}{3}$. This leads to a nonlinear equation, which must be solved, in order to obtain the order of accuracy.

The procedure described in Section 3.5 can be followed with the following solutions: $P(\Delta x)$, $P(\frac{\Delta x}{2})$ and $P(\frac{\Delta x}{3})$.

$$\frac{P_{ij}^k(\frac{\Delta x}{2}) - P_{ij}^k(\Delta x)}{P_{ij}^k(\frac{\Delta x}{3}) - P_{ij}^k(\frac{\Delta x}{2})} = \frac{\Delta x^n - (\frac{\Delta x}{2})^n}{(\frac{\Delta x}{2})^n - (\frac{\Delta x}{3})^n}, \tag{109}$$

where n is de order of accuracy. By multiplying denominator and numinator by $\left(\frac{2}{\Delta x}\right)^n$, equation (109) can be rewritten as:

$$\frac{P_{ij}^k(\frac{\Delta x}{2}) - P_{ij}^k(\Delta x)}{P_{ij}^k(\frac{\Delta x}{3}) - P_{ij}^k(\frac{\Delta x}{2})} = \frac{2^n - 1}{1 - (\frac{2}{3})^n}. \tag{110}$$

This nonlinear equation can be solved for n, which will be done with the Newton-Raphson method [9].

The improved solution, based on (74), can be determined in a similar way as the case with Δx, $\frac{\Delta x}{2}$ and $\frac{\Delta x}{4}$. The improved solution can be written as follows:

$$P_{impr} = P_{ij}^k(\frac{\Delta x}{3}) + C\left(\frac{\Delta x}{3}\right)^n = \frac{P_{ij}^k(\frac{\Delta x}{2}) - \left(\frac{3}{2}\right)^n P_{ij}^k(\frac{\Delta x}{3})}{1 - \left(\frac{3}{2}\right)^n}. \tag{111}$$

The following results are obtained by the numerical solutions $P(\Delta x)$, $P(\frac{\Delta x}{2})$ and $P(\frac{\Delta x}{3})$ interpolated to the common grid M^0 with grid spacing $\frac{\Delta x}{4}$.

Order of Accuracy

With the three numerical solutions $P(\Delta x)$, $P(\frac{\Delta x}{2})$ and $P(\frac{\Delta x}{3})$, the order of accuracy can be determined by solving (110). The order of accuracy is plotted at two different times in Figure 6.16.

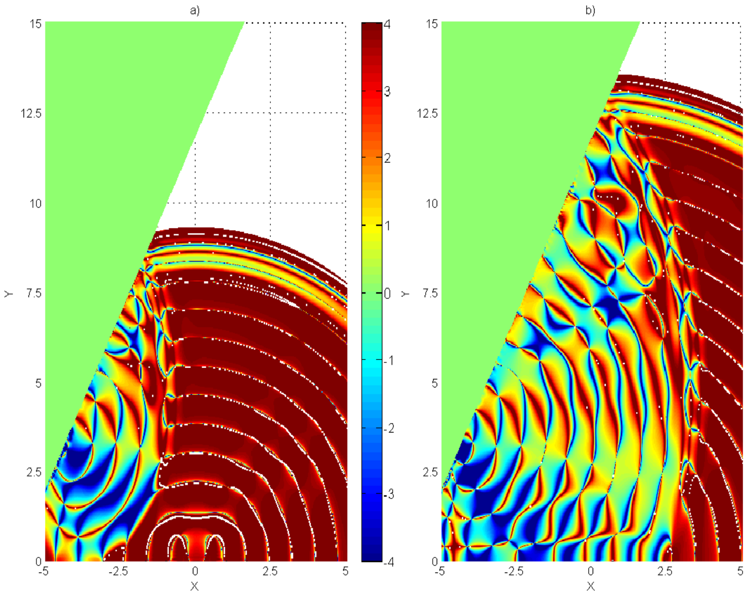

Figure 6.16: Order of accuracy at a) $t = 8.4$. b) $t = 12.6$.

From Figure 6.16 it appears that the results of the reflected wave are not satisfactory at all. The order of accuracy is minus four or less at some locations. Furthermore, it is very wild and not structured. After these results verification of the code is needed, which will be discussed in the next two subsections.

Cut-Cell Procedure

From the code it appears that the pure cut-cell procedure as proposed in [4], had not been implemented. The method that had been implemented is a kind of weighted average of the cut-cell procedure and a staircase approximation of the wall. In the staircase approximation the wall is approximated by complete cells. The implemented approach will be clarified:

If K_i is the i^{th} stage of the LDDRK method (45), this K_i is calculated in the following way:

$$K_i = \frac{1}{4}\left(cK_i^{cut} + (4-c)K_i^{stair}\right), \tag{112}$$

where $c \in [0, 4]$ and K_i^{cut} and K_i^{stair} are the stages calculated with the cut-cell approach and the staircase approach, respectively. By taking $c = 4$ the pure cut-cell approach should appear. However, by taking $c = 4$ the results were unsatisfactory and unstable.

In the cut-cell procedure cut cells can be independent or merged with another (cut) cell, if the surface is less than a defined minimum. In the code all cells are flagged as a complete cell or a cut cell, which could have been merged. If cut cells are merged, they are flagged as one cell. The wall is defined by angle β, Figure 2.1. Because $\beta > 45^o$, there is only one flagged cut cell on every j-level and there can be more on every i-level. The code takes into account only one cut cell for every i-level.

After the implementation of the pure cut-cell procedure with taking into account the possibility of more cut cells on every i-level, the result in Figure 6.17 is obtained.

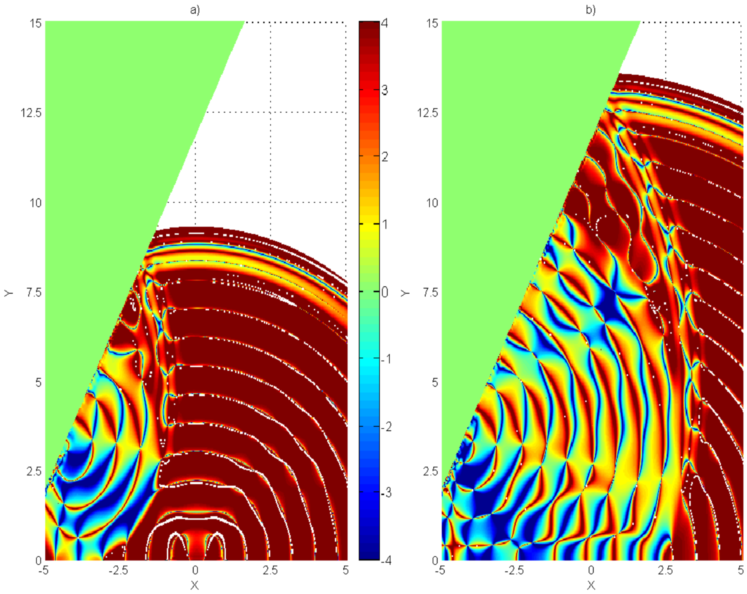

Figure 6.17: Order of accuracy at a) $t = 8.4$. b) $t = 12.6$.

From Figure 6.17 it appears that the results are still unsatisfactory. Moreover, the lower order boundary filter had to be used at the wall in order to obtain smooth solutions. In the next section the implementation of the outflow boundary condition (13c,80) will be discussed.

Outflow Boundary Condition

The outflow boundary condition is given by (80). So, on the outflow boundaries the finite volume treatment is applied to cells with (80). This boundary condition is based on waves that propagate from a certain direction, defined by θ and r. Only waves that propagate from that direction can be handled well by these cells.

This outflow boundary condition has been implemented by treating 15 boundary cells with (80). Thus, this is applied to the complete cells of $(i, j) = ([1, 15], [1, N_y])$, $(i, j) = ([N_x - 14, N_x], [1, N_y])$ and $(i, j) = ([1, N_x], [N_y -$

$14, N_y]$), which are the west, east and north outflow boundary, respectively. So, a kind of exit zone is created. However, waves which are reflected on the wall, have a different propagation direction. When these waves enter the exit zone, the cells cannot handle these waves. These waves will be partially reflected and dispersed.

By calculating the range of the exit zone, there can be determined in which regions it is expected that the solutions will give bad results. For the coarsest grid the exit zone is about $15\Delta x = 1.5$. From Figure 6.16 it appears that the origin of the low order of accuracy is near the wall in the range $x \in [-5, -3.5]$.

After this observation, the implementation of the outflow boundary condition in the code has been adapted. Based on Tam's [10] implementation of the outflow conditions, it appears that the exit zone can be chosen much smaller. Now, only the cells on the boundary, $(i, j) = (1, [1, N_y])$, $(i, j) = (N_x, [1, N_y])$ and $(i, j) = ([1, N_x], N_y)$ are treated with (80). The difference in the solutions can clearly be seen in Figure 6.18.

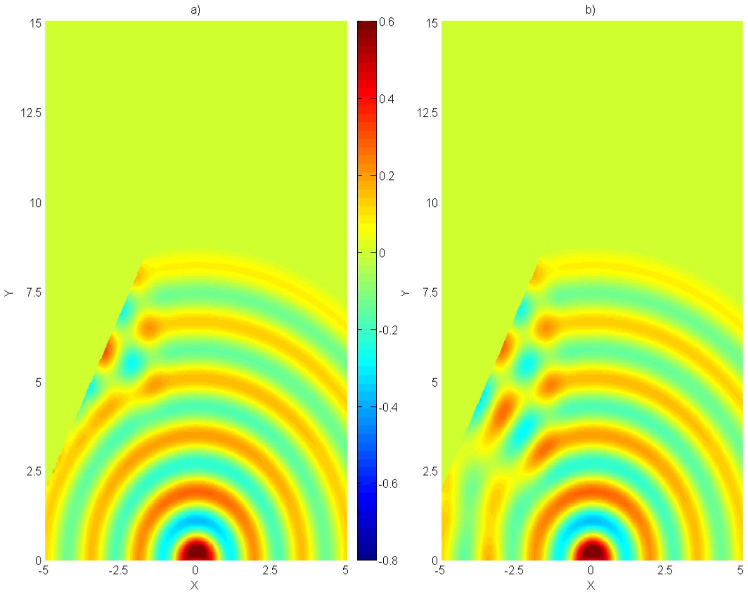

Figure 6.18: Numerical solution $P(\Delta x)$ at $t = 8.4$. a) With exit zone $15\Delta x$. b) With exit zone Δx.

Results and the impact of this adaptation and the boundary condition itself on the order of accuracy will be discussed in the next section.

6.2.4 Impact of the Outflow Boundary Condition

After the implementation, described in the previous section, the exit zone is only one cell and lower order boundary filter at the wall is not needed anymore. Now, the order of accuracy can be determined. Results are plotted in Figure 6.19.

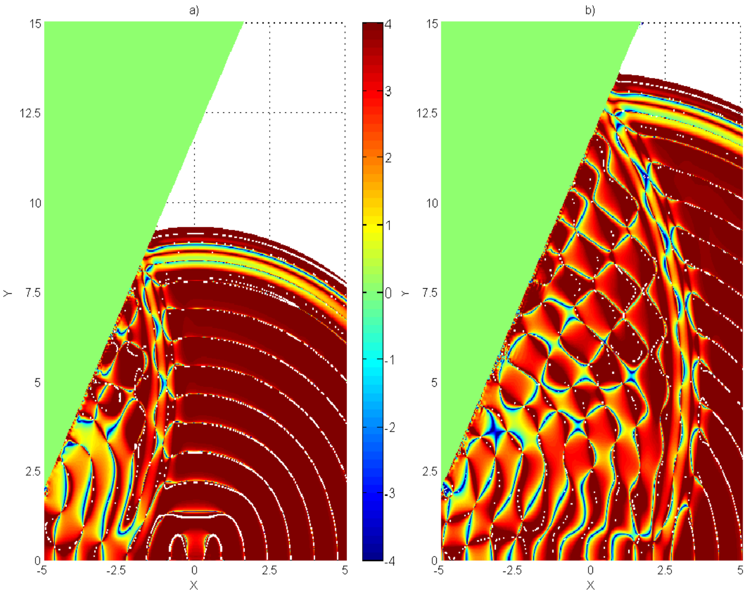

Figure 6.19: Order of accuracy at a) $t = 8.4$. b) $t = 12.6$.

It appears from Figure 6.19 and 6.17 that the order of accuracy has been improved by decreasing the exit zone. So, there can be concluded that the reflected waves could not be handled in the exit zone, which only "expects" waves with a given propagation direction.

Furthermore, from the east boundaries of the domain in Figure 6.18 it appears that an exit zone of only one cell is sufficient for the outflow boundaries. This can be confirmed by Figure 6.18. The outgoing waves are handled without any noticeable reflections.

It has already been addressed that the reflected waves have another propagation direction than the waves originating in the source. From the observation that the exit zone is specially designed for outgoing waves from the source, the reflected waves cannot leave the computational domain without noticeable reflections. In Figure 6.19 this can be noticed by areas with lower order of accuracy around the western outflow boundary.

So, in order to be able to determine the impact of the cut-cell method on the order of accuracy, the impact of this outflow boundary should be eliminated.

6.2.5 Impact of the Cut-Cell Procedure

In order to eliminate the impact of the outflow boundary the western outflow boundary is replaced by a solid wall. Thus, the boundary conditions at $x = -5$ are given by:

$$\begin{aligned} \frac{\partial p}{\partial n} &= 0, \\ \boldsymbol{u} \cdot \boldsymbol{n} &= 0, \end{aligned} \tag{113}$$

where $\boldsymbol{n} = \begin{pmatrix} 1 & 0 \end{pmatrix}^T$ the normal direction of the wall.

Now, the impact of the cut-cell method can be detected. The order of accuracy of this adapted problem is determined and plotted in Figure 6.20.

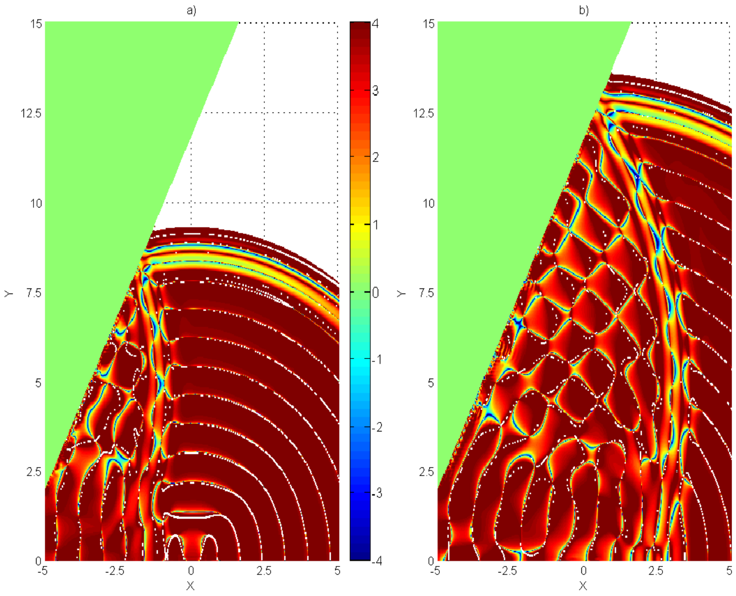

Figure 6.20: Order of accuracy at a) $t = 8.4$. b) $t = 12.6$.

From Figure 6.20 it appears that the order of accuracy of four is not maintained in the entire reflected wave. At some locations the order of accuracy is poor and even negative, at some locations it is still about four and at some locations it is about three. In Figure 6.21 the slice $x = 0$ at $t = 12.6$ is plotted to illustrate this.

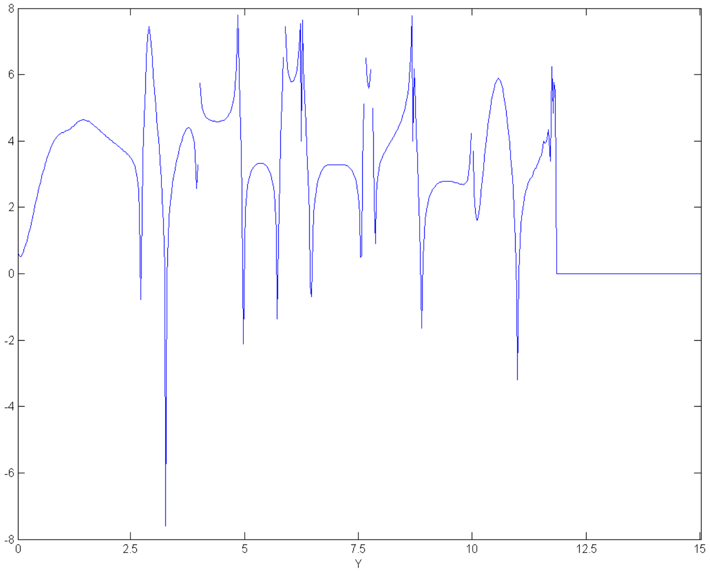

Figure 6.21: Slice $x = 0$ at $t = 12.6$ of order of accuracy from Figure 6.19 b).

In Figure 6.21 it can clearly be seen that the magnitude of the order of accuracy is three in some areas. However, there are also areas where the magnitude is still four. In the next section the interpolation method, that is used for the calculation of the fluxes on the cut faces, is investigated.

Interpolation

For the calculation of the flux on the cut faces an interpolation method is used, which is described in Section 3.3.2. This interpolation method is derived to have fourth order of accuracy. This will be tested with the testcase that is also used in Section 5.1.3.

For every grid the interpolated values at points that need interpolation are stored in a vector, say V_{int}. The analytical values of these points are also known and stored in a vector, say V_{an}. The error vector is defined as $E = V_{an} - V_{int}$. Three scaled norms are chosen to measure the order of accuracy: the L_1-norm, the L_2-norm and the L_∞-norm. The norms are scaled because when the grid spacing is decreasing, the number of points that need interpolation is increasing. So, the norm L_1-norm is divided by N, the number of cells in x-direction, the L_2-norm is divided by $\sqrt{N}$ and the L_∞-norm is not divided by anything. The scaled norms are given in Figure 6.22.

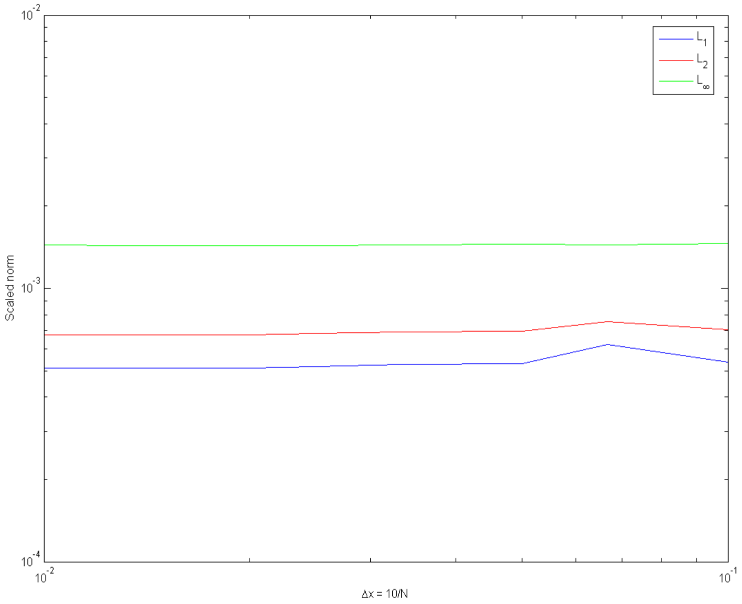

Figure 6.22: Scaled norm of the error vector.

The order of accuracy can be derived from Figure 6.22. The three norms all give similar results and, obviously, all three orders of accuracy are not satisfactory. From this figure it appears that the order of accuracy is about zero for all norms.

Based on the experiences from Section 5.1.2 it is obvious that it is hard to find an accurate interpolation method in the vicinity of the wall by not using the wall condition. In spite of the poor performance of the interpolation method the cut-cell method performs quite well for coarse grids. However, when the grid spacing decreases the interpolation method will have a bigger impact on the performance of the cut-cell method. So, the fact that the solution with grid spacing $\frac{\Delta x}{4}$ does not give satisfactory results can be clarified by the interpolation method.

6.2.6 Extrapolation

In this section there will be investigated whether Richardson extrapolation can be used to improve the numerical solution. Richardson extrapolation is designed such that it improves the solution by cancelling the leading truncation error term. However, for wave problems there is an additional condition for improving the solution: the extrapolated solution must be a smooth wave again.

Implementation

From Section 6.2.1 it appears that at locations, where singularities appear, the solutions are equal within round-off error. This implies that no extrapolation is needed at these locations. Furthermore, formulas (79) and (111) cannot be applied if the order of accuracy is 0. The implementation of Richardson extrapolation is made such that extrapolation is done at locations where the solutions are converging. This means that at locations where the order of accuracy is negative, there will be no extrapolation.

The nonlinear equation (110) for solving the order of accuracy must be solved accurately for not affecting the order of accuracy and the improved solution.

The extrapolation procedure is applied at $t = 4.2$ and the order of accuracy of the complete domain is in Figure 6.15 b). In Figure 6.23 the results of the extrapolation are shown in the slice $x = 0$.

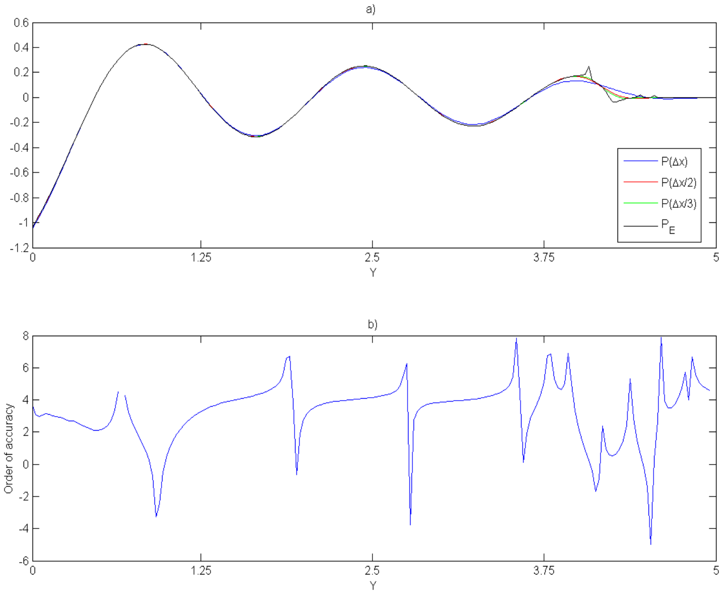

Figure 6.23: Slice $x = 0$ at $t = 4.2$ of a) solutions $P(\Delta x)$, $P(\frac{\Delta x}{2})$ and $P(\frac{\Delta x}{3})$ and extrapolated solution P_E. b) order of accuracy.

From Figure 6.23 it appears that the extrapolation works very well for the wave, but extrapolation in the wavefront gives problems. In the wavefront are local peaks, which are caused by the order of accuracy. So, the wave is not smooth anymore. From Figure 6.15 it appears that the order of accuracy in the wavefront is poor and is not very well-structured, which also appears from Figure 6.23 b). The poor order of accuracy in the wavefront is caused by the boundary and initial condition. When the time derivative of the initial condition is taken, it can be seen that it is discontinuous at $y = 0$.

In the next section Richardson extrapolation is applied to the acoustic pulse in order to investigate whether this gives better results.

Acoustic Pulse

The acoustic pulse without the use of the filter has already been investigated in Section 6.2.1. Results of the order of accuracy at $t = 1$ and $t = 2$ and its slices $x = 0$ can be found in Figure 6.8 and Figure 6.9, respectively. For investigating this problem, the standard formulas for Richardson extrapolation (76) and (79) are used again. Slices $x = 0$ of the results are given in Figure 6.24 and 6.25.

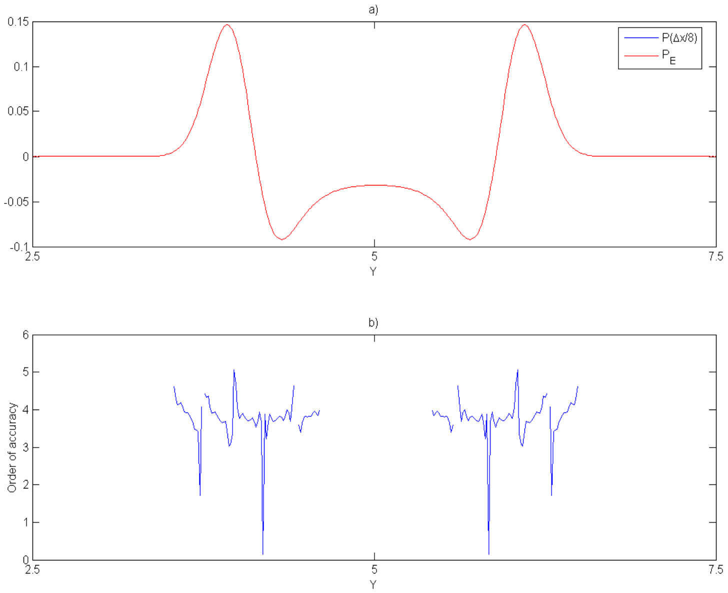

Figure 6.24: Slice $x = 0$ at $t = 1$ of a) solution $P(\frac{\Delta x}{8})$ and extrapolated solution P_E of pulse. b) order of accuracy of pulse.

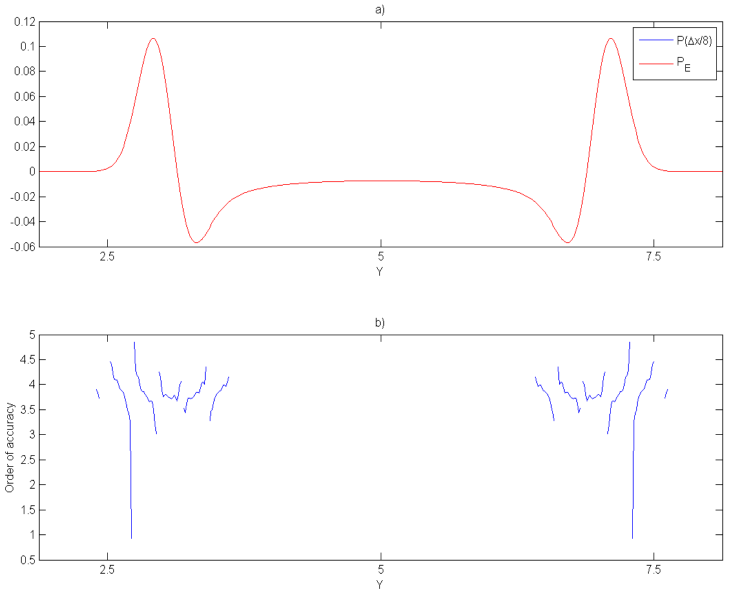

Figure 6.24: Slice $x = 0$ at $t = 2$ of a) solution $P(\frac{\Delta x}{8})$ and extrapolated solution P_E of pulse. b) order of accuracy of pulse.

From Figures 6.24 and 6.25 it appears that the extrapolation for the propagation of an acoustic pulse does not give problems. The wave after extrapolation is smooth and by the construction of Richardson extrapolation this gives an improved solution of the solved linearized Euler equations.

7 Conclusions

This thesis focused on an intensive investigation of the results obtained by Popescu, Shyy and Tai [4]. Schemes for spatial discretization and time integration have been developed which optimize the dispersion and dissipation errors. These schemes are suitable for cartesian grids. For a complex geometry a cut-cell method has been developed, which has to handle irregular cells near the complex boundary. The derived order of accuracy of the complete method is four. Popescu, Shyy and Tai [4] developed a testcase in order to test this approach and concluded that this approach could be effective for acoustic problems with complex geometry.

One of the main objectives of this thesis is to determine the order of accuracy of the complete numerical method. However, this problem does not have an analytical solution and therefore, Richardson extrapolation can be used to determine the order of accuracy. Three numerical solutions, which have different grid spacings, are needed to apply Richardson extrapolation. The numerical solutions have to be interpolated to a common grid, on which Richardson extrapolation can be applied.

The interpolation procedure should not affect the accuracy of the numerical solutions, because this could lead to results, which are not reliable. So, the interpolation of these numerical solutions should be done by a higher order method. In Section 4.2.1 and 5.1 a sixth order interpolation method has been developed that also deals with the complex geometry. For this interpolation method systems have to be solved, which could be ill-conditioned. To avoid the ill-conditioned systems the interpolation procedure was shifted to the origin. Furthermore, row scaling has been applied to the systems, which led to better condition numbers of the matrices. Also, taking the boundary condition $\frac{\partial p}{\partial n} = 0$ had a big positive impact on the interpolation method. This interpolation method has been tested with a testcase, which showed that the method is 6^{th} order.

With three numerical solutions on a common grid the order of accuracy can be determined. This has been done for the original problem formulated in Section 2.2 before the waves hit the wall. The order of accuracy was very poor at some locations and the magnitude was about two (Figure 6.4), which is much less than the derived fourth order. Discontinuities in the boundary and initial condition (13a) were the cause of the poor order of accuracy. However, by removing the discontinuities the order of accuracy was still not the derived fourth order.

In the implementation a filter has been used in order to obtain smooth solutions. At the boundaries a lower order filter has been used. By removing the lower order boundary filter the numerical results were still smooth and satisfactory. Moreover, the magnitude of the order of accuracy was four. However, by removing the lower order boundary filter in the problem with the discontinuities, the order of accuracy became worse.

It is recommended for further research to develop higher order boundary filters, which maintain the order of accuracy for both continuous and discontinuous problems.

Another phenomenon in the order of accuracy is the appearence of singularities. These singularities were caused by solutions that are equal within round-off error.

The outflow boundary condition is such that the outgoing wave should propagate out of the computational domain without any noticeable reflections. The outflow boundary condition can handle waves from only one direction, which are the waves from the baffled piston around the origin. This means that when the reflected wave arrives at the outflow boundary, it cannot propagate out of the domain without reflections. This can also been seen in the order of accuracy (Figure 6.19), which is lower near the outflow boundary.

It is recommended for further research to investigate outflow boundary conditions that can handle waves from all directions.

The outflow boundary condition at the west boundary has been replaced by an oblique wall in order to eliminate the impact of the outflow boundary condition and to investigate the impact of the cut-cell method. Figure 6.20 and 6.21 showed that the the cut-cell method has an impact on the order of accuracy. Before the waves hit the wall the order of accuracy was about four in the complete wave. After the waves hit the wall the order of accuracy decreased to three at some locations in the reflected wave and at some locations the order of accuracy is even less. The interpolation method used in the cut-cell method to determine the flux on the cut faces, caused this performance.

It is recommended for further research to develop an interpolation method that has a high order of accuracy. In Section 5.1.2 could already be seen that this is very hard, because the systems are ill-conditioned. Therefore, it is recommended to develop a different kind of interpolation method, for example spline interpolation for complex geometries.

For the investigation whether Richardson extrapolation can be a tool to improve numerical solutions, Richardson extrapolation was first applied on the waves before hitting the wall. This did not give smooth results, because of the poor and unstructured order of accuracy in the wavefront, which was caused by a discontinuity in the time derivative of the initial condition. By applying Richardson extrapolation to an acoustic pulse, which is completely continuous, it appeared that Richardson extrapolation gives a smooth improved solution.

So, Richardson extrapolation can be a tool to improve numerical solution if the problem does not have any discontinuities.

8 References

[1] G.B. Whitham, "Linear and nonlinear waves", Wiley, New York, 1974

[2] D.T. Blackstock, "Fundamentals of Physical Acoustics", Wiley, New York, 2000

[3] M. Popescu, "A Finite Volume, Cartesian Grid Method for Acoustic Problems with Complex Geometry, A Research Proposal", 2004

[4] M. Popescu, C. Tai and W. Shyy, "A Finite Volume-Based High Order Cartesian Cut-Cell Method for Computational Aeroacoustics", 11^{th} AIAA/CEAS Aeroacoustics Conference, Monterey, California, 2005

[5] G. Ashcroft and X. Zhang, "Optimized Prefactored Compact Schemes", Journal of Computational Physics, Vol. 190, p. 459-477, 2003

[6] R. Hixon and E. Turkel, "High-Accuracy Compact MacCormack-Type Schemes for Computational Aeroacoustics", NASA CR 1998-208672, 1998

[7] J.W. Kim and D.J. Lee, "Optimized Compact Finite Difference Schemes with Maximum Resolution", AIAA Journal, Vol. 34, p. 887, 1996

[8] F.Q. Hu, M.Y. Hussaini and J.L. Manthey, "Low Dissipation and Dispersion Runge-Kutta for Computational Acoustics", Journal of Computational Physics, Vol. 124, p. 177-191, 1996

[9] R.L. Burden and J.D. Faires, "Numerical Analysis, 8^{th} edition", Thomson, Belmont, 2005

[10] C.K.W. Tam and J.C. Webb, "Dispersion-Relation-Preserving Finite Difference Schemes for Computational Acoustics", Journal of Computational Acoustics, Vol. 107, p. 262-281, 1993

[11] W. Shyy and M. Garbey, "A least square extrapolation method for improving solution accuracy of PDE computations", Journal of Computational Physics, Vol. 186, p. 1-23, 2003

[12] M.E. Goldstein, "Aeroacoustics", McGraw-Hill, New York, 1976

[13] W. Shyy, M. Garbey, J. Wu and A. Appukuttan,"Evaluation of Richardson extrapolation in Computational Fluid Dynamics", Numerical Heat Transfer, Part B, Vol. 41, p. 139-164, 2002

9 Appendix

9.1 Appendix A

The equivalence is shown for the fourth order Runge-Kutta scheme and the following ordinary differential equation:

$\frac{du}{dt} = F(u)$,

where $F(u)$ is a linear operator.

1. Standard notation:

$k_1 = \Delta t F(u^n)$,
$k_2 = \Delta t F(u^n + \frac{1}{2}k_1)$,
$k_3 = \Delta t F(u^n + \frac{1}{2}k_2)$,
$k_4 = \Delta t F(u^n + k_3)$,
$u^{n+1} = u^n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$.

2. Notation used in this work:

$K_1 = \Delta t F(u^n)$,
$\vdots$
$K_i = \Delta t F(u^{(i-1)})$,
$u^{(i)} = u^n + b_i K_i, \qquad i = 1...p$,
$\vdots$
$u^{n+1} = u^{(p)}$,
where $b_p = 1$ and $p = 4$ for fourth order.

Writing out both:

1. $u^{n+1} = u^n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) =$
$u^n + \frac{1}{6}(\Delta t F(u^n) + 2\Delta t F(u^n + \frac{1}{2}k_1) + 2\Delta t F(u^n + \frac{1}{2}k_2) + \Delta t F(u^n + k_3)) =$
$u^n + \frac{1}{6}(6\Delta t F(u^n) + \Delta t F(k_1) + \Delta t F(k_2) + \Delta t F(k_3)) =$
$u^n + \frac{1}{6}(6\Delta t F(u^n) + \Delta t^2 F(F(u^n)) + \Delta t^2 F(F(u^n + \frac{1}{2}k_1)) +$
$\Delta t^2 F(F(u^n + \frac{1}{2}k_2))) =$
$u^n + \frac{1}{6}(6\Delta t F(u^n) + \Delta t^2 FF(u^n) + \Delta t^2 FF(u^n) + \Delta t^2 FF(\frac{1}{2}k_1) +$
$\Delta t^2 FF(u^n) + \Delta t^2 FF(\frac{1}{2}k_2)) =$
$u^n + \frac{1}{6}(6\Delta t F(u^n) + 3\Delta t^2 FF(u^n) + \frac{1}{2}\Delta t^3 FFF(u^n) +$
$\frac{1}{2}\Delta t^3 FFF(u^n + \frac{1}{2}k_1)) =$
$u^n + \frac{1}{6}(6\Delta t F(u^n) + 3\Delta t^2 FF(u^n) + \Delta t^3 FFF(u^n) + \frac{1}{4}\Delta t^4 FFFF(u^n)) =$
$u^n + \Delta t F(u^n) + \frac{1}{2}\Delta t^2 FF(u^n) + \frac{1}{6}\Delta t^3 FFF(u^n) + \frac{1}{24}\Delta t^4 FFFF(u^n)$.

2. $u^{n+1} = u^{(p)} = u^n + b_4 K_4 = u^n + b_4 \Delta t F(u^{(3)}) = u^n + b_4 \Delta t F(u^n + b_3 K_3) =$
$u^n + b_4 \Delta t F(u^n) + b_4 b_3 \Delta t F(\Delta t F(u^{(2)})) = u^n + b_4 \Delta t F(u^n) +$
$b_4 b_3 \Delta t^2 FF(u^n + b_2 K_2) =$
$u^n + b_4 \Delta t F(u^n) + b_4 b_3 \Delta t^2 FF(u^n) + b_4 b_3 b_2 \Delta t^2 FF(\Delta t F(u^{(1)})) =$
$u^n + b_4 \Delta t F(u^n) + b_4 b_3 \Delta t^2 FF(u^n) + b_4 b_3 b_2 \Delta t^3 FFF(u^n + b_1 K_1) =$
$u^n + b_4 \Delta t F(u^n) + b_4 b_3 \Delta t^2 FF(u^n) + b_4 b_3 b_2 \Delta t^3 FFF(u^n) + b_4 b_3 b_2 b_1 \Delta t^4 FFFF(u^n)$.

With:

$b_4 = 1$, $b_3 = \frac{1}{2}$, $b_2 = \frac{1}{3}$, $b_1 = \frac{1}{4}$,

this notation is equivalent with the standard Runge-Kutta fourth order notation.

9.2 Appendix B

Code document

This document describes the main adaptations made in the code. First, fixing the wall will be discussed. Second, the filter will be discussed. And last, some small adaptations to the code will be given.

The code has been made in one file: InterpCC.for.

Fixing the wall

The wall was defined by the starting point at the southwest corner of the cell close to $y = 2$. The statement used for this:

```
j_init=int(2/dy)+1,
```

So, `j_init` is the cell in y-direction where the wall starts. Obviously, this has been done because now there are only cells which need a cut-cell approach and cells with the outflow boundary condition. However, the wall is not fixed in this way, it is Δy- and Δx-dependent.

In cooperation with Tai, the wall starting point has been defined at $(x, y) = (-5 - \frac{\Delta x}{2}, 2 - \frac{\Delta x \tan(\alpha)}{2})$. (The wall starting point has been chosen like this, because the cell centers are chosen to be on the boundaries.) Therefor, a cell has been introduced with a cut face and an outflow face. Now, the wall always intersects the point $(x, y) = (-5, 2)$.

Filter

In the filtering procedure near the boundary the solutions of p, u and v were damped by a factor $\frac{1}{10}$ at $j = 1$ and a factor $\frac{1}{2}$ at $j = 2$:

```
j=1
    DO i=nnx(j),imax
        uu(i,j)= (ui(i,j)-filter_dom*(ui(i,j)-ui(i,j+1))) *.1
ENDDO
j=2
    DO i=nnx(j),imax
        uu(i,j)=(ui(i,j)+filter_dom*(ui(i,j-1)-
           2*ui(i,j)+ui(i,j+1))) *.5
    ENDDO
```

Solutions with damping are completely different than solutions without damping:

1. Obviously the amplitudes differ.
2. The shape of the wave is different. Without the damping the wave propagates much more spherical.
3. With damping the phases of the wave differ a lot with a different Δx and CFL.

Moreover, it makes the filtering procedure space-dependent.
Issues 1 and 2 are illustrated in Figure B.1:
The wall has not been plotted in the figures.

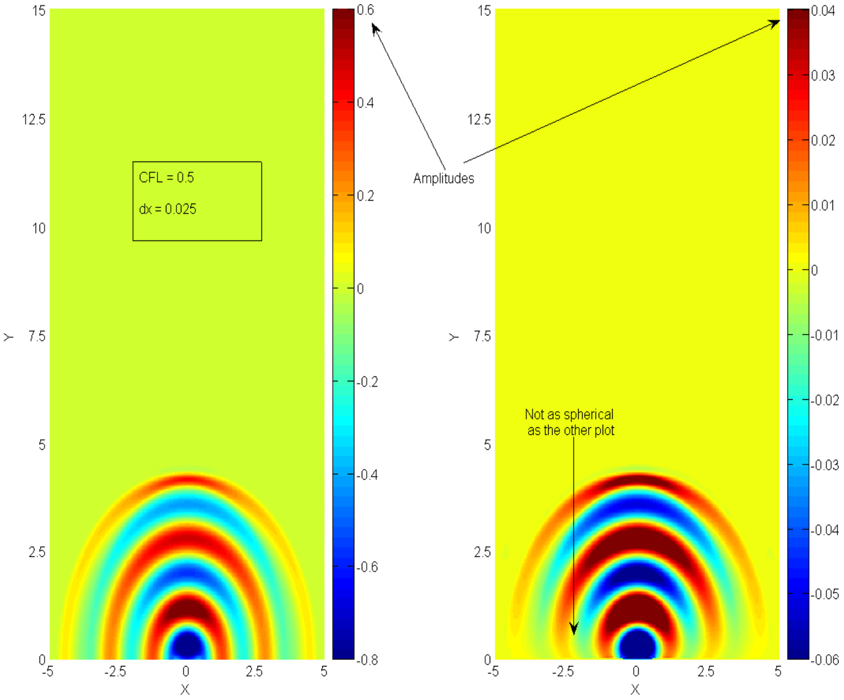

Figure B.1: Solutions at $t = 4.2$. a) Solution without damping. b) Solution with damping.

Issue 3 is clearly illustrated in Figure B.2:

Left is $\Delta x = 0.1$ and $CFL = \frac{1}{8}$ and right is $\Delta x = 0.025$ and $CFL = \frac{1}{2}$, both on $t = 4.2$ and with the described damping.

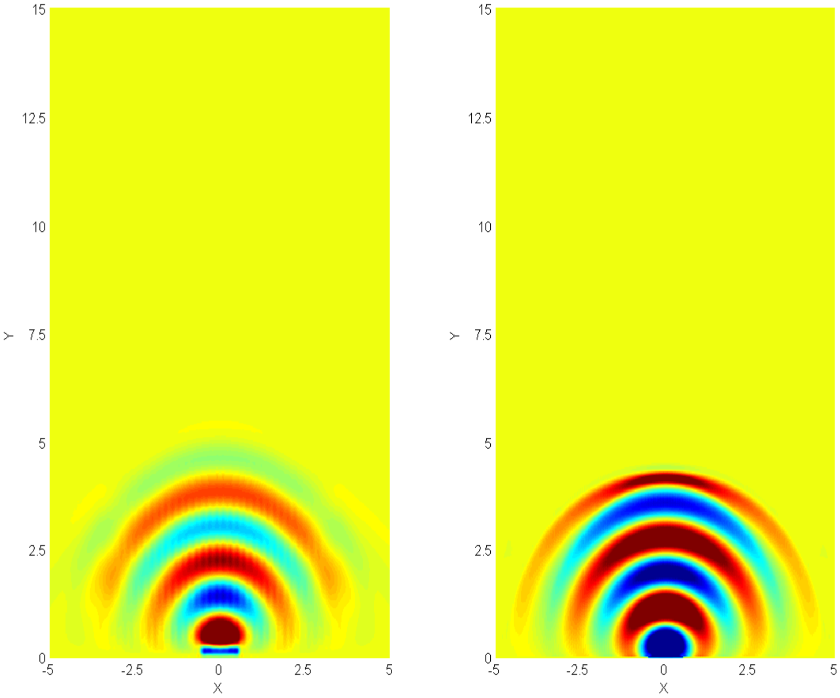

Figure B.2: Solutions at $t = 4.2$ and both with damping. a) Solution with $\Delta x = 0.1$ and $\nu = \frac{1}{8}$. b) Solution with $\Delta x = 0.025$ and $\nu = \frac{1}{2}$.

Furthermore, there was no filtering in $x-$direction. Filtering in $x-$direction was done in the domain above the wall, so that was implemented wrongly. Filtering in $x-$direction in the domain below the wall has been implemented now. The difference can be seen in Figure B.3. Here a $j-$slice is plotted, so $y =$*constant* (Obviously blue is after filtering):

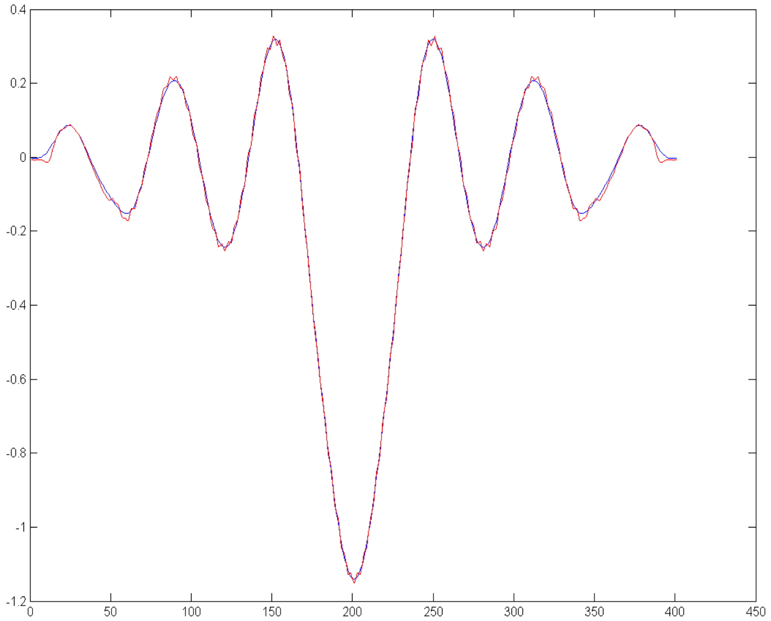

Figure B.3: j-Slice of the solution.

For filtering three different expressions have been used:

1. `uu(i,j)= ui(i,j)-filter_dom*(ui(i,j)-ui(i,j+1))`
2. `uu(i,j)=ui(i,j)+filter_dom*(ui(i,j-1)-2*ui(i,j)+ui(i,j+1))`
3. `uu(i,j)=ui(i,j)+filter_dom*(-0.25*ui(i,j-2)+ui(i,j-1)-1.5*ui(i,j)`

   `+ui(i,j+1)-0.25*ui(i,j+2))`

These expressions are for filtering in $y-$direction and similar expressions are used for filtering in $x-$direction. Expression 1 is used for filtering at the boundaries ($j = 1$, $j = N$, $i = 1$, $i = M$), expression 2 is used for filtering near the boundaries ($j = 2$, $j = N - 1$, $i = 2$, $i = M - 1$) and expression 3 is used for filtering the interior. However, in the code expression 3 has been used for filtering near the wall, so it uses also points that are not in the domain, but left of the wall. Now, expression 1 has been implemented at locations that are closest to the wall and expression 2 at locations next to that.

Small adaptations

1. The variable *pp* (characteristic of the piston) has been adapted in order to fix the geometry:

   `pp=0.5/dx+1` becomes `pp=0.5/dx`.

   The boundary condition is implemented with the following loop:

   `do i=cx-pp,cx+pp`

2. The boundary condition was "delayed" by dt. When $\cos(t)$ should be implemented, $\cos(t - dt)$ was implemented.

3. New subroutines has been made for writing solutions to files. The filename has been changed into, for example, dx4_cfl50_0001.txt. Here, 4 represents $\frac{1}{10\Delta x}$ (so $\Delta x = 0.025$) and 50 represents $100 * cfl$ (so $cfl = 0.5$).

4. The variable `writecontor` has been introduced. After every "writecontor" timesteps the solution will be written to a txt-file.

5. New subroutines has been made for writing $x-$ and $y-$coordinates to files.

6. For space discretization u^e and u^w (40) have to be calculated. Actually, $u_i^e = u_{i+1}^w$, so only the east faces or the west faces need to be calculated and one west or east boundary face, respectively. This saves work and storage. However, the storage that is saved is neglegible, because the procedure is done row by row (and column by column).