Diss. ETH No. 14440

A Numerical Method for the Design of Internal Flow Configurations Based on the Inverse Euler Equations

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZÜRICH

for the degree of
Doktor der Mathematik

presented by
ANDREA SCASCIGHINI
Dipl. Math. ETH
born May 28, 1974
citizen of Minusio TI

accepted on the recommendation of
Prof. Dr. R. Jeltsch, examiner
Prof. Dr. C. Schwab, co-examiner
Dr. T. Sommer, co-examiner

2001

Seite Leer /
Blank leaf

Danksagung

Diese Arbeit enstand während meiner Assistenzzeit am Seminar für angewandte Mathematik an der ETH Zürich unter der Leitung von Prof. Rolf Jeltsch und im Rahmen des KTI Projektes KTS 4571.1 "Inverses Design von Turbinenkomponenten".

An dieser Stelle möchte ich mich bei meinem Doktorvater Prof. Rolf Jeltsch bedanken, der es mir ermöglicht hat, an einem interessanten, vielseitigen und sehr anwendungsorientierten Projekt mitzuwirken. Der offene und direkte Charakter von Prof. Rolf Jeltsch, sowie seine Bereitschaft, das unabhängige Denken und Handeln seiner Studenten zu fördern, habe ich von den Seminaren über gewöhnliche Differentialgleichungen über die Diplomarbeit bis hin zur Doktorarbeit immer geschätzt.

Prof. Christoph Schwab danke ich für die Übernahme des Korreferates und seine konstruktiven Bemerkungen.

Dr. Thomas Sommer (Alstom (Switzerland) Ltd.) ein sehr interessierter Anprechpartner gewesen. Für seine ständige Motivation und für sein Fokussieren auf die praxis–relevanten Probleme, sowie für die Übernahme des Korreferates bin ich ihm sehr dankbar.

Grosser Dank geht auch an Andreas Troxler, der alle Schritte erlebt hat, die zu dieser Dissertation geführt haben. Ohne die langen Diskussionen mit ihm, vorzugsweise am Morgen vor 8 Uhr oder am Freitag nach 5, und ohne seinen Humor wäre diese Arbeit nie zustandegekommen.

Mit Marina Savelieva und Manuel Torrilhon habe ich zwei gute Kollegen gefunden, sowohl tagsüber im Büro als auch bei gemütlichen Abendessen oder bei harten Velorundfahrten.

Ich freue mich weiterhin gute Kontakte mit meinen alten Kollegen Dr. Susanne Zimmermann und Dr. Jochen Maurer halten zu können, die mich ins Leben am SAM eingeführt haben.

Grosser Dank geht an Eleonora Ghertsos für das Korrekturlesen dieser Dissertation und für die Hinweise über die schönsten Inseln Griechenlands.

Mein Dank für das angenehme Arbeitsklima am SAM geht auch an Dr. Markus Melenk, Dr. Ana–Maria Matache, Simin Motamen, Dr. Markus Brun, Andreas Adelmann, Thomas Wihler, Gregor Schmidlin, Philip Frauenfelder, Damian Loher, Fabian Buchmann, Radu Todor und Andreas Rüegg.

Diese Arbeit wurde von Alstom Power (Switzerland) Ltd. und der Kommission für Innovation und Technologie (KTI) finanziert.

Zürich, im Dezember 2001 Andrea Scascighini

Abstract

In this thesis we develop a numerical method for the aerodynamic inverse design problem:

> find the shape of a flow device which generates a given target pressure along its walls.

The design method is based on the inverse Euler equations of J.J. Keller [ZAMP **49** (1998)]. The inverse Euler equations are a particular formulation of the Euler equations of gas dynamics. The Euler equations are rewritten in a flow aligned system of coordinates and the dependent and independent variables are exchanged. Because of their relevance in the turbomachinery industry the method is applied to the design of internal flow configurations, such as diffusers or nozzles.

The first part of this work deals with the two dimensional and axis–symmetric formulation of the inverse Euler equations. The derivation of the equations presented in this work shows that they are composed by a system of equations for the generation of the shape of the device and by equations describing the physics of the flow. This splitting of the system of partial differential equations allows to determine a set of physically relevant boundary conditions and to modify the equations for the shape generation in order to devise a more efficient numerical method.

The numerical method is based on a finite difference discretization and on a Newton solver for the resulting system of algebraic equations. The linear system is solved by the Bi–CGStab iterative solver preconditioned with an incomplete LU factorization. It is shown that the coupling of Newton's method with a nested iterations strategy gives a speedup by a factor 6 and more.

The three-dimensional extension of the inverse Euler equations due to J.J Keller is shown to be valid only for complex lamellar flows, i.e. flows in which the velocity vector is perpendicular to the vorticity.

In the second part of the thesis the axis–symmetric inverse Euler equations are extended to handle quasi–three–dimensional flows. New effects such as blade blockage and deflection can be taken into account in the novel inverse formulation of the quasi–three–dimensional flow equations. Viscous losses can be incorporated in the design by the inverse form of the distributed loss model or by means of a boundary layer method coupled to the inverse Euler equations.

A broad set of numerical examples shows the capabilities of our design tool.

Zusammenfassung

In dieser Arbeit wird ein numerisches Verfahren entwickelt für das aerodynamische inverse Entwurfsproblem:

> Finde die Form eines Strömungsgeräts, welches eine gegebene Ziel–Druckverteilung entlang seiner Wände erzeugt.

Die Methode basiert auf den inversen Eulergleichungen von J.J. Keller [ZAMP **49** (1998)]. Die inversen Eulergleichungen sind eine besondere Formulierung der Eulergleichungen der Gasdynamik. Dazu werden die Eulergleichungen in einem rechtwinkligem Stromlinien–Koordinatensystem geschrieben und die abhängigen und unabhängigen Variablen vertauscht. Wegen ihrer Relevanz in der Turbinenindustrie wird die Methode für das Design von Strömungsgeräten, wie Düsen und Diffusoren, angewandt.

Im ersten Teil dieser Arbeit geht es um die zwei–dimensionalen und axial–symmetrischen inversen Eulergleichungen. Die dort vorgestellte Herleitung zeigt, dass die inversen Eulergleichungen aus einem Gleichungssystem für die Erzeugung der Form des Gerätes und aus Gleichungen, die das Strömumgsfeld beschreiben, bestehen. Dieses Aufspalten des Systems von partiellen Differentialgleichungen ermöglicht es einen Satz physikalisch relevanter Randbedingungen herzuleiten und die Gleichungen für die Formgenerierung zu modifizieren, um ein effizienteres numerisches Verfahren zu entwickeln.

Das numerische Verfahren basiert auf einer Finite–Differenzen–Diskretisierung und auf einem Newton–Löser für das resultierende System von algebraischen Gleichungen. Das mit einer unvollständigen LU Zerlegung vorkonditionierte, lineare Gleichungssystem wird mit dem iterativen Löser Bi–CGStab gelöst. Es wird gezeigt, dass die Kopplung zwischen dem Newton–Vefahren und den geschachtelten Iterationen eine Beschleunigung um den Faktor 6 und mehr bringt.

Für die drei–dimensionale Erweiterung der inversen Eulergleichungen von J.J. Keller wird gezeigt, dass sie nur für komplex lammellare Strömungen gültig ist, d.h für Strömungen, in denen der Geschwindigkeitsvektor senkrecht zu der Wirbelstärke steht.

Im zweiten Teil dieser Arbeit werden die axial–symmetrischen inversen Eulergleichungen verallgemeinert, um quasi–drei–dimensionale Strömungen behandeln zu können. Neue Effekte, wie die Blockierung oder Umlenkung durch Schaufeln, können aufgenommen werden. Viskose Verluste werden mit einem "verteilte–Verluste–Modell" modelliert oder durch ein Grenzschichtenverfahren, das mit den inversen Eulergleichungen gekoppelt wird.

Mehrere numerische Beispiele zeigen die Einsatzmöglichkeiten des Entwurfswerkzeuges.

Contents

Danksagung **ii**

Abstract **v**

Zusammenfassung **vii**

1 Introduction **1**

2 An Introduction to the Inverse Euler Equations **7**

2.1 Aerodynamic Inverse Shape Design: A Motivation 7

2.2 Flow Equations 10

2.3 Inverse Coordinates 13

2.4 A Remark on the Inverse Coordinates in 3D 15

2.5 Transformation Rules 17

2.6 Keller's Inverse Formulation of the Euler Equations 19

2.7 The Scaled Laplace Equations 21

2.8 Interpretation 23

2.9 Definition of the Computational Domain 24

2.10 Flow Field–Geometry Relations 26

2.11 Boundary Conditions 27
2.12 Axis–Symmetric Flows 30
2.13 Conclusions 33

3 Discretization and Numerical Method 35
3.1 Discretization 36
3.1.1 Setup 36
3.1.2 Interior of the Domain 37
3.1.3 Boundary Conditions 37
3.2 The Solution Procedure 38
3.2.1 The Newton–Krylov Method 38
3.2.2 Finding the Initial Guess: The Nested Iterations 40
3.3 Validation of the Code 43
3.3.1 Numerical Study of the Order of Accuracy 43
3.3.2 The Recovery Problem 46
3.4 Performance of the Code 48
3.4.1 Inverse vs. Direct Computations 49
3.4.2 Comparison of Different Solution Strategies 49
3.5 Conclusions 52

4 Numerical Results 55
4.1 Numerical Results for Potential Flows 55
4.1.1 Laval Nozzle 56
4.1.2 Channel Flow 57
4.2 Numerical Results for Rotational Flows 58
4.2.1 Laval Nozzle 59

4.3 Axis–Symmetric Flows 60

4.3.1 Diffuser 61

4.3.2 Parametric Studies of an Elbow Channel 62

4.4 Conclusions 65

5 Advanced Flow Models in Inverse Coordinates 69

5.1 Quasi–Three–Dimensional Flow Model 70

5.1.1 Passage Averaged Equations 71

5.1.2 Inverse Coordinates and Transformation Rules 77

5.1.3 Q3D Inverse Euler Equations 79

5.1.4 Numerical Example 79

5.2 Distributed Loss Model 81

5.2.1 Flow Model 82

5.2.2 Distributed Forces 84

5.2.3 Numerical Example 85

5.3 Boundary Layer Computations 87

5.3.1 Integral Method 89

5.3.2 Viscous–Inviscid Coupling 91

5.3.3 Numerical Example 92

5.4 Concluding Remarks and Outlook 93

A Distributed Loss Model 97

Bibliography 99

Chapter 1

Introduction

The advent of fast computers is changing the strategy for the solution of complex industrial problems. A more and more pronounced shift from the necessity of conducting experiments towards the development of suitable numerical methods can be observed. A lot of effort has gone, and is still put, into the development of models of the physics of the problem. For example in fluid dynamics, models ranging from the simple potential flow to the viscous Navier–Stokes equations with turbulence models have been developed. There is a wide literature on the various flow models and on their validation by comparison with experiments. The development of numerical schemes for flow simulation is at an advanced stage too. Most of the research is concentrated in reproducing in a virtual environment the setting and the results of the classical experiment: the behavior of one configuration can be studied and predicted in an efficient and accurate way.

The ultimate goal of the industry is not to simulate one configuration but to produce better products, e.g. airplanes with a low level of fuel consumption because of reduced drag on the wings or a turbine with a higher efficiency index because of an improved combustion chamber, etc. There is a gap between the main flow of the research, which concentrates on simulation tools, and the high interest of the industry in design tools. In fact, although numerical methods (for simulation) have been introduced in the design cycle in order to speed it up, the design methods still rely heavily on the knowledge and on the intuition of the designer. The simplest, less efficient but more common way to design a new device is to notice the shortcoming of the existing design by analyzing the results from a simulation tool, to identify the possible causes and to propose a new design. The new configuration is analyzed with the computational tools at disposal and the loop is repeated until a satisfactory design point is reached. This "trial–and–error" approach is simple and does not need new computational tools, the improvements in the design however are usually limited.

We present a brief overview of some more complex strategies for the aerodynamic design and optimization of a flow device.

PDE-constrained optimization is the most general and flexible approach for the improvement of the aerodynamic properties of a device. Such methods consist in a functional to be minimized (or maximized), such as the drag or the pressure recovery, subject to the flow equations. Some manufacturing constraints such as a minimal thickness of the blade or the maximal length of the diffuser can complicate the problem. This approach is very natural: in many cases the definition of the functional is straightforward and the available flow solver can be coupled with a black–box optimization package. This approach is called deterministic optimization. The biggest drawback of the deterministic optimization is the high cost of the computation of the gradient of the functional. The straightforward approach of a finite difference approximation of the gradients is too costly since many solutions of the flow equations are needed: the cost of the simple–minded optimization is of many order of magnitude higher than the cost of a plain simulation. New ideas are used to reduce this shortcoming, for example by applying automatic differentiation techniques [41] [51]. A widespread alternative is the computation of the gradients or sensitivities via adjoint methods [33] [34]. An adjoint method reduces the computational cost of the design to about 5–10 flow simulations, by finding the gradient of the functional as the solution of a partial differential equation. A second point of concern of gradient based optimization tools is their local character. The solution is always a local minimizer (or maximizer) but there is no way to tell whether the solution is a global minimum (or maximum) too. The stochastic optimization overcomes this problem [55] [42] [28]. It consists in an evolutionary strategy for driving a population of configurations, maybe 50, towards an optimal one by applying the biological rules of evolution. By keeping many designs under consideration, local minima can be avoided and a global optimal solution can be found. The computational cost of solving many flow problems, one for each individual of the population, can be reduced by exploiting the perfect intrinsic parallelism of the method. To sum up, optimization methods are very flexible, can handle complex constraints and use state of the art flow solvers; unfortunately these methods are computationally very intensive.

An alternative approach is based on inverse methods. Under "inverse methods" we understand all the analytical and numerical methods which improve the aerodynamic properties of a device by controlling a physical quantity such as the pressure or the swirl along the walls of the device. The prototypical problem formulation is the "target–pressure problem":

> find the shape of a device which generates a prescribed target pressure along its walls.

Inverse methods are appealing since many kinds of losses, such as those arising from viscous effects and separation, can be controlled and reduced by a judicious choice of the pressure distribution, and in particular of the pressure gradient, along the side walls of the domain. Inverse methods have been successful in the re–design of turbine blades, airplane wings and channels in two and three dimensions.

Analytical work on the target–pressure problem is mainly based on conformal mappings. Lighthill in [38] [39] showed how to apply conformal mappings to the design of cascade blades. An incompressible, irrotational flow is assumed. In [38] conditions for the existence of a closed blade profile are given. The methods of complex variables are used by Daripa [12] [13] for generating airfoils for compressible irrotational flows. Both authors prescribe the modulus of the velocity vector on the whole boundary.

In the attempt to consider rotational, inviscid flow, various iterative inverse methods, based in general on existing Euler solvers, have been developed. An iterative design method is based on the coupling of an Euler solver with a wall modification procedure. An iteration consists in a geometry modification followed by a time step of the flow solver. The two parts of the iteration are basically independent and decoupled. Particularly appealing is the possibility of (re)using a known flow solver. Various strategies for the geometry update have been developed, for example the elastic membrane concept [27] [14], where the walls are considered as a membrane which moves according to the force exerted by the difference between actual and target pressure. An alternative is given by a transpiration model, which consists in assuming that the wall is porous and the mass can be fictitiously injected in such a way that the new wall satisfies the slip boundary condition [61] [16]. These iterative methods converge slowly, since the coupling between the flow and the geometry is very weak. Demeulenaere and van den Braembussche [16] first noticed that the convergence rate of the iterative scheme could be greatly improved by introducing "driving terms" expressing the coupling between geometry and flow. In [16], viscous and three dimensional flows are considered. Because of their modular structure, the iterative inverse methods can be applied on general configurations. Most of the focus has been put into design systems for turbine blades.

On the opposite side, some researchers try to solve the target–pressure problem in a fully coupled way. These approaches are based on the definition of an intrinsic or natural system of coordinates, in which the unknown domain is represented by a rectangle. For potential flows these systems of coordinates are based on the stream–function and on the potential. The first work in this direction is due to Stanitz [52]. Ahmed and Myring [1] follow the same ansatz for axis–symmetric flows. Zannetti [64] considers the two dimensional and axis–symmetric Euler equations in a fixed computational domain, image of the physical space with moving walls. The computational domain is based on the stream–function, which is constant along the side walls, independently of their shape. The time dependent Euler equations are solved and the steady state solution yields the desired design. The work of Stanitz [52] has been extended to two dimensional rotational flow by Dedoussis et al. [15]. Chaviaropoulos and coworkers [8] [9] extend their method to three dimensional potential flows. Some peculiarities of these works deserve an explanation. First of all the geometry of the device is found in one step, being the solution of a set of partial differential equations. For this reason they are called single–pass methods. Secondly, the configurations under investigation are *internal* configurations such as channels and nozzles. This striking difference to the iterative inverse methods described above is due to the fact the the intrinsic system of coordinates used in the single–pass methods is singular at stagnation points. The leading edge of a blade cannot be handled in a simple way. Last, the three dimensional equations and methods are limited to potential flows [8] [9]. We will show that this restriction is a direct consequence of the ansatz for the three dimensional natural coordinates. The strength of this class of methods lies in their speed and robust convergence of the solvers.

Single–pass methods are based on special systems of partial differential equations, which do not have a simple interpretation. Specifically, the equations do not look like the Euler or Navier–Stokes equations. One exception is the single–pass inverse method developed by Troxler [57], where the Euler equations and a standard grid generation problem [54] are solved in a fully coupled manner. This approach combines the intuitive flavor of the iterative inverse methods with the speed and robustness of the single–pass methods.

The inverse Euler equations of J.J. Keller [36], fall in the class of the single-pass methods. They are based on rewriting the Euler equations in a natural system of coordinates in the same way as [15] [8] [9] [64] [52] and on the interchange of dependent and independent variables. By means of this transformation new types of boundary conditions, in particular geometrical constraints, can be taken into account in the single–pass inverse method. Further, the standard inlet and outlet boundary conditions for internal flows [30] apply. The inverse Euler equations, not being based on purely geometric considerations, can be extended to handle quasi–three–dimensional flows.

We consider in this thesis the aerodynamic design of internal flow configurations and develop a numerical method for its solution based on the inverse Euler equations of J.J. Keller [36]. The strength of our method consists in physically relevant inlet and outlet boundary conditions as well as the possibility to prescribe on the side walls a pressure distribution and/or a geometrical constraint. Further our method adds to the standard axis–symmetric flow model (for which other single–pass inverse methods already exist) quasi–three dimensional effects such as blade blockage and deflection and viscous losses via a distributed loss model. The method accounts for losses generated by viscous effects near the walls by a boundary layer method too, thus allowing viscous design for unseparated flows.

This thesis is organized as follows:

In chapter 2 we first introduce the reader to the target–pressure problem from an industrial point of view and then derive the inverse formulation of the Euler equations due to Keller [36] for a two dimensional flow. This derivation of the inverse Euler equations shows the main points of the whole construction, in such a way that its validity in the three dimensional case can be analyzed. Chapter 2 is a reference for the whole thesis and in particular for chapter 5. We present the complete set of boundary conditions and the axis–symmetric inverse Euler equations too.

In chapter 3 we describe a numerical method for the solution of the inverse Euler equations. It is based on a finite difference discretization and on a Newton–Krylov–ILU solver. A nested iteration strategy for finding the initial guess on a fine grid is applied to accelerate the method and to enhance its robustness. The second part of chapter 3 is devoted to the validation of the code and to a comparison of different simplified versions of the solver.

A gallery of numerical results is presented in chapter 4. We stress the flexibility of our method by presenting results for potential and rotational flows, for inverse computations as well as computations with a mixture of inverse boundary conditions and geometrical constraints. Results for a technical relevant configuration of an axis–symmetric diffuser are presented; in particular we describe a simple parametric study which can be used as a guideline for a designer.

In the last chapter we describe some extensions of the inverse Euler equations. We show in chapter 2 that the three dimensional inverse formulation of the Euler equations is valid only for potential or complex lamellar flows. For relevant fully three dimensional duct flows, the inverse Euler equations described by Keller in [36] do not have a solution or the solution is unphysical. In chapter 5 we introduce therefore a quasi–three–dimensional flow model [35] [32] and rewrite it in a novel inverse form. The quasi–three–dimensional equations are augmented by a distributed loss model [30] for handling losses caused by shocks or viscosity. Furthermore an integral boundary layer method is described and coupled with the two dimensional inverse Euler equations. Examples elucidating the various effects are given.

Chapter 2

An Introduction to the Inverse Euler Equations

This first chapter gives an introduction to the inverse shape design problem and its solution based on the approach of Keller [36]. In the first section we introduce and motivate the inverse design problem by considering the industrial relevant problem of improving the performance of a diffuser. In the ten successive sections we derive in a detailed manner the inverse Euler equations. We will refer to these sections later on when rewriting some more advanced flow models in inverse coordinates. In the last section we give an overview of the axis–symmetric case.

2.1 Aerodynamic Inverse Shape Design: A Motivation

Diffusers are devices used for converting kinetic energy into pressure by slowing the flow [62]. They are relevant components of gas turbines and turbofans or wind tunnels. For example the exhaust diffuser in a gas turbine, used to bring the pressure of the flow after the turbine stages back to the atmospheric pressure, is known to have a significant impact on the performance of the whole unit.

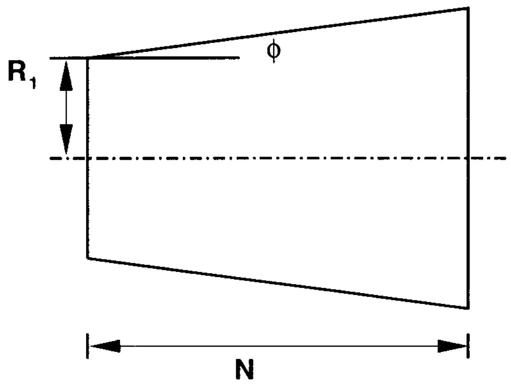

Figure 2.1: *Schematic view of the conical diffuser. The shape parameters are the length* N*, the inlet radius* R_1 *and the opening angle* ϕ *of the diffuser. Data on the efficiency of the diffuser in dependence of the shape parameters is given in Table 2.1.*

The underlying principles of a diffuser can be understood by considering an incompressible flow in a diverging channel, with uniform inlet and outlet flow conditions. Denoting by

$$\rho, p_1, u_1, A_1 \quad (p_2, u_2, A_2) \tag{2.1}$$

the inlet (resp. outlet) values of the density, pressure, velocity and area, we can write the continuity equation and conservation of momentum (Bernoulli 's equation) as

$$u_1 A_1 = u_2 A_2 \tag{2.2a}$$

$$\frac{1}{2}\rho u_1^2 + p_1 = \frac{1}{2}\rho u_2^2 + p_2. \tag{2.2b}$$

Solving for the pressure rise $\Delta p = p_2 - p_1$ gives

$$\Delta p = p_2 - p_1 = \frac{\rho}{2} u_1^2 \left(1 - \left(\frac{A_1}{A_2}\right)^2\right). \tag{2.3}$$

ϕ	R_1	$\frac{N}{R_1}$	AR	η
2.0	1.0	4.0	1.3	0.81
2.0	1.0	8.0	1.64	0.81
4.0	1.0	4.0	1.64	0.78
4.0	1.0	8.0	2.43	0.78
7.9	1.0	4.0	2.43	0.59
7.9	1.0	8.0	4.48	0.60

Table 2.1: *Experimental data on the efficiency η of a conical diffuser depending on the length N, the inlet radius R_1, the opening angle ϕ and the area ratio AR. Data from [40]. It is worth to notice that the efficiency sinks for constant area ratio and increasing opening angle .*

In the ideal case of an inviscid flow, the pressure rise depends on the area ratio $\mathrm{AR} = \frac{A_2}{A_1}$. By increasing the area ratio, more kinetic energy is converted into pressure. The dimensionless coefficient

$$C_{pr} = \frac{p_2 - p_1}{\frac{\rho}{2} u_1^2} \tag{2.4}$$

is known as the pressure recovery coefficient and in the ideal case is equal to the ideal pressure recovery coefficient

$$C_{pr.i} = 1 - \left(\frac{A_1}{A_2}\right)^2 = 1 - \frac{1}{\mathrm{AR}^2}. \tag{2.5}$$

Ideally the pressure recovery coefficient can be made as close as possible to one by increasing the area ratio, meaning that all the kinetic energy is transformed into static pressure. Unfortunately this ideal situation is not confirmed by experiments. For a conical diffuser we present in Table 2.1 the efficiency η

$$\eta = \frac{C_{pr}}{C_{pr.i}} \tag{2.6}$$

of the diffuser in dependence of the opening angle ϕ, the area ratio AR, and the ratio between the length of the diffuser N and the inlet radius R_1. The various shape parameters can be seen in Figure 2.1.

From Table 2.1 it can be deduced that diffusers with the same area ratio but different opening angles do not have the same efficiency. This discrepancy between theory and experiment, and hence the measurement of an efficiency η

smaller than one, is due to the presence of viscous effects in the real flow. The difficulty in the design of a diffuser is to minimize the losses due to viscous effects. From Table 2.1 we see that a large opening angle reduces the efficiency of the diffuser. The responsible phenomenon is the separation induced by a too strong adverse pressure gradient or by a too big curvature of the wall. The losses introduced by the separation bubble are relevant and the task of the designer is to avoid them as much as possible. There are analytical methods, such as the boundary layer methods of section 5.3, or empirical rules such as

- the opening angle of the diffuser should be less than 4 degrees and
- the diffuser should be bell–shaped,

as well as experimental data which allow the designer to determine an optimal pressure distribution along the walls. In this sense optimal means with a high pressure recovery and possibly low viscous losses. Related work on the specification of an optimal pressure distribution over airfoils can be found in [20].

The existence of methods for finding optimized pressure distributions naturally leads to the formulation of the target–pressure problem or aerodynamic inverse shape design problem:

> find the shape of a diffuser which generates a given target pressure along its walls.

This problem has been considered by various authors; some approaches have been described in the introduction. We follow in this thesis the ansatz of Keller [36], which consists in rewriting the flow equations in a streamline aligned system of coordinates and interchanging the dependent and independent coordinates. We present this approach for two dimensional flows in the sequel of this chapter.

2.2 Flow Equations

An easy–to follow derivation of the inverse Euler equations can be given provided that the flow equations are manipulated in the physical coordinates

(x, y) instead of working out the algebra in the transformed coordinates as in [36]. First, the Euler equations are manipulated in order to find an equivalent formulation which is well suited for a transformation to inverse coordinates.

We consider a steady, compressible, inviscid and two dimensional flow described by the (conservative) Euler equations

$$\begin{aligned} \operatorname{div}(\rho \mathbf{u}) &= 0 && (2.7)\\ \operatorname{div}\left(\rho \mathbf{u}\mathbf{u}^T + \mathbf{I}p\right) &= 0 && (2.8)\\ \operatorname{div}(\rho H \mathbf{u}) &= 0. && (2.9) \end{aligned}$$

The unknown variables are the density ρ, the pressure p, the temperature T the velocity vector $\mathbf{u} = (u, v)^T$ and the total enthalpy $H = c_pT + \frac{1}{2}\|\mathbf{u}\|^2$. The divergence acts on the rows of the matrix $\rho\mathbf{u}\mathbf{u}^T + \mathbf{I}p$, where $\mathbf{I}$ is the identity. The closure of the system is given by the state equation

$$p = \rho R T, \tag{2.10}$$

where R is the gas constant, which is related to the heat coefficients at constant pressure c_p and at constant volume c_v by $R = c_p - c_v$. The heat coefficients are related to the isentropic coefficient by

$$\gamma = \frac{c_p}{c_v}. \tag{2.11}$$

For air we set $\gamma = 1.4$ and $R = 287\frac{m^2}{s^2K}$.

For formal reasons we assume that the flow is smooth, and therefore we can expand the derivatives in (2.7)–(2.9) to get the non–conservative formulation of the Euler equations

$$\begin{aligned} \operatorname{div}(\rho \mathbf{u}) &= 0 && (2.12)\\ (\mathbf{u}\cdot\nabla)\,\mathbf{u} + \frac{1}{\rho}\nabla p &= 0 && (2.13)\\ (\mathbf{u}\cdot\nabla)\,H &= 0. && (2.14) \end{aligned}$$

Introducing the vorticity $\omega = v_x - u_y$ and by using the identity

$$(\mathbf{u}\cdot\nabla)\,\mathbf{u} = \nabla\left(\frac{1}{2}\|\mathbf{u}\|^2\right) - \mathbf{u}^{\perp}\omega \tag{2.15}$$

we can rewrite (2.13) as

$$\nabla\left(\frac{1}{2}\|\mathbf{u}\|^2\right) - \mathbf{u}^{\perp}\omega + \frac{1}{\rho}\nabla p = 0. \tag{2.16}$$

Through all the work we will use the notations $\mathbf{u}^\perp = (v, -u)^T$ and $\nabla^\perp = (\partial_y, -\partial_x)^T$.

From the first law of thermodynamics

$$Tds = c_v dT - \frac{p}{\rho}\frac{d\rho}{\rho} \tag{2.17}$$

and from the state equation

$$p = \rho(c_p - c_v)T = \rho RT \tag{2.18}$$

we deduce that

$$\frac{dp}{\rho} = c_p dT - Tds, \tag{2.19}$$

where s denotes the entropy. Again in (2.19) we assume that the flow is smooth and therefore rule the presence of shocks out.

Inserting (2.19) in the pressure gradient $\frac{1}{\rho}\nabla p$ we get Crocco's formulation of the momentum equations (2.8)

$$\nabla H - T\nabla s = \mathbf{u}^\perp \omega. \tag{2.20}$$

(2.20) describes the generation of vorticity because on non–vanishing total enthalpy or entropy gradients. The case of vorticity generation by a shock is ruled out by our smoothness assumption. The only possibility for a rotational flow is due to non–uniform inlet values such as those after a turbine stage.

Away from a stagnation point we are allowed to project (2.20) along the flow and along a direction perpendicular to it. In particular, by projecting (2.20) along the flow we get

$$\mathbf{u} \cdot (2.20) \Longleftrightarrow \mathbf{u} \cdot \nabla H - T\mathbf{u} \cdot \nabla s = 0 \underbrace{\Longleftrightarrow}_{(2.14)} \mathbf{u} \cdot \nabla s = 0. \tag{2.21}$$

Projection on the direction perpendicular to the flow gives

$$\mathbf{u}^\perp \cdot \nabla H - T\mathbf{u}^\perp \cdot \nabla s = \|\mathbf{u}\|^2 \omega. \tag{2.22}$$

To summarize, a smooth, isentropic planar flow can be described, away from a stagnation point, by the following set of equations

$$\begin{aligned} \operatorname{div}(\rho\mathbf{u}) &= 0 && (2.23)\\ \mathbf{u}^\perp \cdot (\nabla H - T\nabla s) &= \|\mathbf{u}\|^2\omega && (2.24)\\ \mathbf{u} \cdot \nabla s &= 0 && (2.25)\\ \mathbf{u} \cdot \nabla H &= 0. && (2.26) \end{aligned}$$

Neglecting the continuity equation (2.23), only transport operators of the form $\mathbf{u}^{\perp} \cdot \nabla$ and $\mathbf{u} \cdot \nabla$ occur. This property will be fully exploited by the inverse Euler equations.

2.3 Inverse Coordinates

In the target–pressure problem, the computational domain in the physical (x, y)-space is not known a priori. We are anyway able to define a fixed computational domain by introducing a body–fitted system of coordinates based on the stream–function ψ.

We define the stream–function ψ for the divergence free vector field $\rho\mathbf{u}$ by

$$\partial_y \psi = \rho u, \tag{2.27a}$$

$$-\partial_x \psi = \rho v. \tag{2.27b}$$

It follows that the continuity equation (2.23) is automatically satisfied and can be dropped. Further consequence of this definition and of the slip boundary condition $\mathbf{u} \cdot \mathbf{n} = 0$ is that along the side walls of the configuration the stream–function ψ is constant independently of their shape. In other words, the unknown wall coincides with a level set of ψ. This is easily seen from the following computation

$$\frac{d\psi(\mathbf{x}(s))}{ds} = \nabla\psi \cdot \dot{\mathbf{x}}(s) = \rho\mathbf{u} \cdot \dot{\mathbf{x}}^{\perp} = \rho\mathbf{u} \cdot \mathbf{n} = 0, \tag{2.28}$$

where $\mathbf{x}(s)$ is a parameterization of the boundary, $\mathbf{x}^{\perp} = (y, -x)^T$ and $\mathbf{n}$ is the normal to the boundary.

A further simplification of (2.24)–(2.25) can be obtained by introducing a coordinate σ along the flow. The natural coordinate σ is defined by

$$h\nabla\sigma = \frac{\mathbf{u}}{\|\mathbf{u}\|}. \tag{2.29}$$

The existence of the "potential" σ is assured by the presence of the integrating factor h (also called stretching factor).

In the case of an irrotational, isentropic flow, definition (2.29) can be reduced to the usual definition of the potential σ

$$\nabla\sigma = \mathbf{u}. \tag{2.30}$$

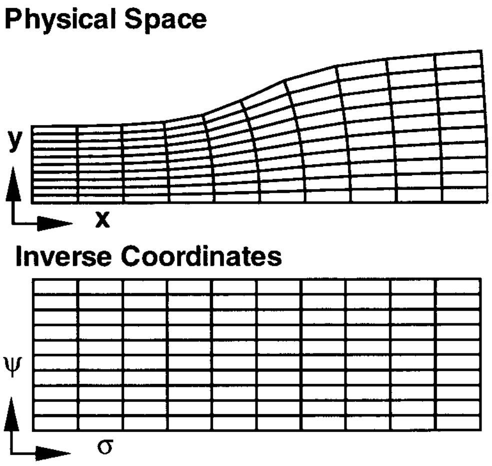

Figure 2.2: *Diffuser with streamline aligned orthogonal system of coordinates (top) and its representation in inverse coordinates (bottom). The iso–lines of σ and ψ generating an orthogonal grid are plotted*

We remark that the material derivative $\mathbf{u} \cdot \nabla$ is equivalent to a derivative in σ direction, see section 2.5.

The definitions of σ and ψ can be interpreted as the implicit definition of a mapping between the (x, y)–plane and the (σ, ψ)–plane or as the definition of a body fitted system of coordinates. The peculiarity of this mapping or new system of coordinates is that the side walls of the flow device are mapped, independently of their actual shape, on $\psi = const$ lines. The computational domain in the inverse coordinates (σ, ψ) is a known rectangle! A diffuser with the flow aligned orthogonal system of coordinates and its representation in inverse coordinates is shown in Figure 2.2. The grid lines consist in iso–line of σ and ψ.

We remark that the iso–lines of σ and ψ generate an orthogonal grid in the physical space (x, y) since

$$\nabla\sigma \cdot \nabla\psi = -\frac{1}{h\|\mathbf{u}\|}\mathbf{u} \cdot \rho\mathbf{u}^{\perp} = 0. \tag{2.31}$$

The orthogonal grid is generated automatically by the design method. It can, possibly after a refinement towards the side walls, be used as grid for a Navier–Stokes computation needed to quantify the losses in the diffuser in a late stage of the design process.

The main idea behind the inverse Euler equations is to rewrite the equations (2.23)–(2.26) in the (σ, ψ)–plane and then apply the hodograph transformation. For the details of the definition of the computational domain we refer to section 2.9.

2.4 A Remark on the Inverse Coordinates in 3D

In the "classical" fluid dynamics the potential and the stream–function are used in order to simplify the governing equations and/or to reduce the computational cost of a simulation. The interest in the natural coordinate σ and in the stream–function ψ in the field of the inverse design comes from the fact that ψ assumes two constant values along the walls of the diffuser; ψ can be used as an independent variable. Therefore the question arises, whether these ideas can be extended to the three dimensional case. We will show in this section that unfortunately the three dimensional inverse Euler equations hold only for a restricted class of flows.

The definition of the natural coordinate σ for a three dimensional velocity field is

$$h\nabla\sigma = \frac{\mathbf{u}}{\|\mathbf{u}\|}, \tag{2.32}$$

where now $\mathbf{u} \in \mathbb{R}^3$ is the velocity vector in cartesian coordinates. The simple computation

$$\mathbf{u} \cdot \nabla \times \mathbf{u} = (\|\mathbf{u}\|h)\nabla\sigma \cdot \left(\nabla(h\|\mathbf{u}\|) \times \nabla\sigma + (h\|\mathbf{u}\|)\nabla \times \nabla\sigma\right) = 0 \tag{2.33}$$

proves the

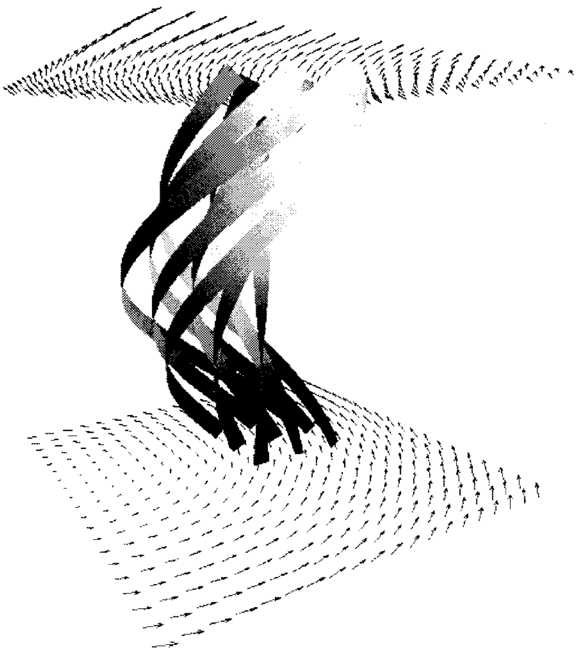

Figure 2.3: *Stream–ribbons of a flow for which the natural coordinate does not exist. The stream–ribbons follow the streamlines of the flow and their twist is proportional to the vorticity.*

Proposition 1 *The integrating factor h and the natural coordinate σ exist if and only if*

$$\mathbf{u} \cdot \nabla \times \mathbf{u} = 0. \tag{2.34}$$

In particular h and σ exist for two dimensional and axis–symmetric rotational flows, or for three dimensional potential flows.

Flows satisfying (2.34) are called complex lamellar [44]. From Proposition 1 it follows that we cannot expect the existence of the natural coordinate for a general flow, and therefore Keller' s construction of the inverse Euler equation cannot be applied. An example of a flow not satisfying (2.34) is a free vortex generated by the velocity vector

$$\mathbf{u} = (-y, x, 1)^T. \tag{2.35}$$

Stream ribbons of (2.35) can be seen in Figure 2.3.

Since this kind of flow is very common in turning ducts, the three dimensional inverse Euler equations lose their validity, resp. they have no solution or produce a non physical flow.

2.5 Transformation Rules

After the three–dimensional interlude, we return to the two dimensional case. The next step in the derivation of the inverse Euler equations is to rewrite (2.23)–(2.26) in the (σ, ψ) coordinates. We therefore recall the basic rules of the change of variables as well as the relations between the various derivatives.

We consider two systems of coordinates, the cartesian coordinates denoted by $\mathbf{x} = (x, y)^T$ and the inverse coordinates (σ, ψ). We present some relationships for the mapping $\mathbf{x} = \mathbf{x}(\sigma, \psi)$.

The regularity of the mapping

$$(x, y) \leftrightarrow (\sigma, \psi) \tag{2.36}$$

is controlled by the determinant of the Jacobi matrix of the transformation which is defined as

$$J = x_\sigma y_\psi - y_\sigma x_\psi . \tag{2.37}$$

It is assumed that J remains bounded, see section 2.10 for conditions ensuring the regularity of the mapping. J can be interpreted as the area element in (σ, ψ) coordinates.

The length element $\sqrt{x_\sigma^2 + y_\sigma^2}$ assumes a simple form because of the normalization $\frac{1}{\|\mathbf{u}\|}$ in (2.29):

$$h^2 = x_\sigma^2 + y_\sigma^2 = \|\mathbf{x}_\sigma\|^2. \tag{2.38}$$

(2.38) expresses the equivalence between h and the arc–length.

From the chain rule the following rules can be derived [37] [54]

$$\sigma_x = \frac{y_\psi}{J} \quad , \quad \sigma_y = \frac{-x_\psi}{J} \tag{2.39}$$

$$\psi_x = -\frac{y_\sigma}{J} \quad , \quad \psi_y = \frac{x_\sigma}{J} . \tag{2.40}$$

The first application of (2.39) and (2.40) is to express $\|\mathbf{u}\|$ as a function of x, y and h. It holds

$$\|\mathbf{u}\| = \frac{\|\rho\mathbf{u}\|}{\rho} \underset{(2.27)}{=} \frac{\|\nabla^{\perp}\psi\|}{\rho} \underset{(2.39),(2.40)}{=} \frac{\|\mathbf{x}_{\sigma}\|}{\rho J} \underset{(2.38)}{=} \frac{h}{\rho J}. \tag{2.41}$$

From (2.39) and (2.40) we find the representations

$$\mathbf{u} = \begin{pmatrix} u \\ v \end{pmatrix} \underset{(2.29)}{=} \|\mathbf{u}\| h \nabla\sigma \underset{(2.39)}{=} \|\mathbf{u}\| \frac{h}{J} \begin{pmatrix} y_{\psi} \\ -x_{\psi} \end{pmatrix} \underset{(2.41)}{=} \frac{h^2}{\rho J^2} \mathbf{x}_{\psi}^{\perp} \tag{2.42}$$

and

$$\mathbf{u}^{\perp} = \begin{pmatrix} v \\ -u \end{pmatrix} \underset{(2.27)}{=} -\frac{1}{\rho} \begin{pmatrix} \psi_x \\ \psi_y \end{pmatrix} \underset{(2.40)}{=} \frac{1}{\rho J} \begin{pmatrix} y_{\sigma} \\ -x_{\sigma} \end{pmatrix} \underset{(2.41)}{=} \frac{1}{\rho J} \mathbf{x}_{\sigma}^{\perp}. \tag{2.43}$$

Again the notation $\mathbf{x}^{\perp} = (y, -x)^T$ has been used.

The gradient transforms as follows

$$\nabla_{\mathbf{x}} f = \frac{1}{J} \begin{pmatrix} y_{\psi} & -y_{\sigma} \\ -x_{\psi} & x_{\sigma} \end{pmatrix} \cdot \nabla_{\sigma,\psi} f =: \mathbf{C} \cdot \nabla_{\sigma,\psi} f, \tag{2.44}$$

where $\nabla_{\mathbf{x}} = (\partial_x, \partial_y)^T$ and $\nabla_{\sigma,\psi} = (\partial_{\sigma}, \partial_{\psi})^T$.

From (2.44) we can derive some simple formulas for the differential operators arising in (2.24)–(2.26). For example the material derivative $\mathbf{u} \cdot \nabla_{\mathbf{x}}$ reduces to

$$\mathbf{u} \cdot \nabla_{\mathbf{x}} = \frac{h^2}{\rho J^2} \mathbf{x}_{\psi}^{\perp} \cdot \frac{1}{J} \mathbf{C} \nabla_{\sigma,\psi} = \frac{h^2}{\rho J^3} (\|\mathbf{x}_{\psi}\|^2, 0) \nabla_{\sigma,\psi} = \frac{1}{\rho J} \partial_{\sigma} \tag{2.45}$$

since $\nabla_{\mathbf{x}}\sigma$ is parallel to $\mathbf{u}$, while the cross–flow derivative obeys

$$\mathbf{u}^{\perp} \cdot \nabla_{\mathbf{x}} = \frac{1}{\rho J} \mathbf{x}_{\sigma}^{\perp} \cdot \frac{1}{J} \mathbf{C} \nabla_{\sigma,\psi} = \frac{1}{\rho J^2} (0, -h^2) \nabla_{\sigma,\psi} = -\frac{h^2}{\rho J^2} \partial_{\psi}, \tag{2.46}$$

since $\mathbf{u}^{\perp}$ is anti–parallel to $\nabla_{\mathbf{x}}\psi$. In both cases we use the orthogonality of the (σ, ψ) system of coordinates, $\mathbf{x}_{\sigma} \cdot \mathbf{x}_{\psi} = 0$.

The divergence of a vector field $\mathbf{f}$ transforms as follows

$$\mathrm{div}_{\mathbf{x}}\mathbf{f} = \frac{1}{J} \mathrm{div}_{\sigma,\psi} C^T \mathbf{f}, \tag{2.47}$$

where $\mathrm{div}_{\mathbf{x}}\mathbf{f} = \nabla_{\mathbf{x}} \cdot \mathbf{f}$ and $\mathrm{div}_{\sigma,\psi}\mathbf{f} = \nabla_{\sigma,\psi} \cdot \mathbf{f}$.

2.6 Keller's Inverse Formulation of the Euler Equations

In section 2.3 we implicitly defined a mapping between two system of coordinates

$$\mathbf{x}(\sigma, \psi) \leftrightarrow (\sigma(x, y), \psi(x, y)). \qquad (2.48)$$

The new system of coordinates (σ, ψ) has three valuable properties

- the unknown walls coincide with a $\psi = const$ line, thus we have a fixed, rectangular computational domain and
- the derivatives along the flow such as $\mathbf{u} \cdot \nabla$ reduce to derivatives in sigma direction. In particular we deduce that (2.25) and (2.26) can be written as $H = H(\psi), s = s(\psi)$. Since H, s are given at the inlet by the boundary conditions, we can in practice drop the equations (2.25) and (2.26) and just use the thermo-dynamical formula (2.72)–(2.74).
- the new system of coordinates is orthogonal. This is easily seen in (2.31).

To summarize, in the inverse coordinates σ, ψ the Euler equations reduce just to (2.24). Its inverse formulation can be found by rewriting (2.24) in the (σ, ψ) coordinates by applying the chain rule and by interchanging the dependent and independent coordinates in the same way as in the hodograph transformation. The inverse formulation of (2.24) is found to be

$$\partial_\psi H - T \partial_\psi s = \frac{1}{\rho J} \partial_\psi \left(\frac{h^2}{\rho J} \right), \qquad (2.49)$$

as one easily computes

$$\|\mathbf{u}\|^2 \omega = \frac{h^2}{\rho^2 J^2} \frac{1}{J} \mathrm{div}_{\sigma,\psi} \left(\frac{1}{J\rho} \mathbf{C}^T \mathbf{x}_\sigma \right) = -\frac{h^2}{\rho J^2} \frac{1}{J\rho} \partial_\psi \left(\frac{h^2}{\rho J} \right). \qquad (2.50)$$

We can now concentrate on the mapping between the system of coordinates. The mapping is described by the following set of equations

$$\|\rho \mathbf{u}\| h \nabla \sigma = \nabla^\perp \psi \qquad (2.51)$$

which are a direct consequence of the definitions $\nabla^{\perp}\psi = \rho\mathbf{u}$ and $h\nabla\sigma = \mathbf{u}/\|\mathbf{u}\|$. By applying (2.39) and (2.40) we find the inverse formulation

$$\mathbf{x}_\sigma = \frac{h^2}{J}\mathbf{x}_\psi^{\perp}, \tag{2.52}$$

where we used the identity $h^2 = \|\mathbf{x}_\sigma\|^2$. The system of partial differential equations (2.52) is called generalized Cauchy–Riemann equations.

The complete set of the inverse Euler equation for smooth, planar, compressible and rotational flows as derived in [36] is given below,

$$\begin{aligned}
x_\sigma &= \frac{h^2}{J}y_\psi && \text{(2.53a)}\\
y_\sigma &= -\frac{h^2}{J}x_\psi && \text{(2.53b)}\\
\partial_\psi H - T\partial_\psi s &= \frac{1}{\rho J}\partial_\psi\left(\frac{h^2}{\rho J}\right) && \text{(2.53c)}\\
\partial_\sigma H &= 0 && \text{(2.53d)}\\
\partial_\sigma s &= 0. && \text{(2.53e)}
\end{aligned}$$

The unknowns in the inverse Euler equations are x, y and h. The remaining variables can be expressed as a function of x, y and h in the way described in section 2.10. For sake of an easier implementation, in the numerical method developed in chapter 3 we substitute h by the auxiliary variable

$$\Phi = \frac{h^2}{J} \tag{2.54}$$

representing the ratio between the squared length element in σ direction h^2 and the local area element J.

Numerical methods for the Cauchy–Riemann equations The main difficulty in the development of a numerical method for the inverse Euler equations (2.53a)–(2.53e) lies in the discretization of the generalized Cauchy–Riemann equations (2.52). In particular (2.52) is not a quasi-linear system. A model problem can be found by setting the nonlinear factor $\frac{h^2}{J} = 1$. Physically this coincides with the assumption of an irrotational, incompressible flow. In the literature the model problem

$$\partial_x u + \partial_y v = 0 \text{ in } \Omega \tag{2.55a}$$

$$\partial_x v - \partial_y u = 0 \text{ in } \Omega \tag{2.55b}$$

$$\mathbf{n} \cdot \mathbf{u} = g \text{ on } \partial\Omega, \tag{2.55c}$$

is considered, where **n** is the normal to the domain Ω, $\mathbf{u} = (u, v)^T$ and g a given function. By integration by part (2.55) can only have a solution if

$$\int_{\partial\Omega} g ds = 0. \tag{2.56}$$

A numerical method for (2.55) must incorporate this constraint. In the literature [45] [26] [59] [6] the constraint is found in the least squares formulation of the discretization. Only if a discrete version of (2.55c) holds then the overdetermined system of algebraic equations reduces to a square system.

The inverse design problem is nonlinear. It would be very inefficient to solve it by a nonlinear least squares problem, not to speak about the difficulties in interpreting the results in case of non–vanishing residual. Therefore, we distance ourselves from the generalized Cauchy–Riemann based formulation of the inverse Euler equations (2.53) in the next section and decide to work with the corresponding second order system. Numerical experiments in section 3.3.1 show that the (possible) compatibility problems do not arise.

2.7 The Scaled Laplace Equations

The main difficulty in the numerical solution of the inverse Euler equations lies in the first order, nonlinear system (2.52). In fact there are a few numerical methods for (2.52), such as those described in [45] [26] [6] [59]. The common feature of these methods is that they rely on a least squares formulation of the discretized equations. This would lead to a computationally inefficient method and therefore we distance ourself from the original formulation of the inverse Euler equations due to Keller by rewriting the first order elliptical system in its corresponding second order system. The second order system is found by computing

$$\partial_\sigma \left(\frac{J}{h^2} \partial_\sigma \mathbf{x} \right) = \partial_\sigma \partial_\psi \mathbf{x}^\perp = \partial_\psi \partial_\sigma \mathbf{x}^\perp = -\partial_\psi \left(\frac{h^2}{J} \partial_\psi \mathbf{x} \right). \tag{2.57}$$

The resulting scaled Laplace equations

$$\partial_\sigma \left(\frac{J}{h^2} \partial_\sigma \mathbf{x} \right) + \partial_\psi \left(\frac{h^2}{J} \partial_\psi \mathbf{x} \right) = 0 \tag{2.58}$$

replace the generalized Cauchy–Riemann equations in the new formulation of the inverse Euler equations.

The scaled Laplace equations are a well known tool for planar grid generation [54] [37] and have been shown to be effective to construct orthogonal grids [53] [21] [22].

From now on the system of partial differential equations (2.58), (2.53c)–(2.53e) will be called inverse Euler equations.

By substituting the scaled Laplace equations (2.58) for the generalized Cauchy–Riemann equations (2.52) we lose a–priori the orthogonality of the grid (2.31). The orthogonality of the grid generated by the scaled Laplace equations holds in the case of

Proposition 2 *For each sufficiently smooth solution of the nonlinear elliptic problem*

$$\begin{aligned} \mathbf{x}_{\sigma.\sigma} + \mathbf{x}_{\psi.\psi} &= 0 \; in \; \Omega && (2.59) \\ \mathbf{x}_{\sigma} \cdot \mathbf{x}_{\psi} &= 0 \; on \; \partial\Omega && (2.60) \end{aligned}$$

it holds that

$$\mathbf{x}_{\sigma} \cdot \mathbf{x}_{\psi} = 0 \; in \; \Omega. \qquad (2.61)$$

In particular the grid generated from an isentropic, irrotational and incompressible flow, i.e. for $\frac{h^2}{J} = 1$, *is orthogonal.*

Proof

The idea of the proof is to show that the function $f = \mathbf{x}_{\sigma} \cdot \mathbf{x}_{\psi}$ satisfies a Laplace equation with zero boundary condition and therefore, invoking the maximum principle [24], must vanish identically in Ω. A simple computation reveals that

$$f_{\sigma.\sigma} + f_{\psi.\psi} = \mathbf{x}_{\psi} \cdot \partial_{\sigma}(\Delta\mathbf{x}) + 2\mathbf{x}_{\sigma.\psi} \cdot \Delta\mathbf{x} + \mathbf{x}_{\sigma} \cdot \partial_{\psi}(\Delta\mathbf{x}) = 0 \text{ in } \Omega. \qquad (2.62)$$

This completes the proof.

We remark that the last Proposition can be extended to handle the case of a given $\frac{h^2}{J} = p(\sigma)q(\psi)$ by making a change of variables. For general $\frac{h^2}{J}$, no proof of the orthogonality of the resulting grid is known, numerical computations in section 3.3.1 confirm the orthogonality of the grid.

Similarly the relation

$$h^2 = \|\mathbf{x}_\sigma\|^2 \tag{2.63}$$

is a–priori lost in the substitution of the scaled Laplace equations for the generalized Cauchy–Riemann equations. Numerical experiments show that (2.63) holds anyway.

2.8 Interpretation

The inverse Euler equations consist in an elliptic system (2.58) which is derived by purely geometric considerations and in a set of three ordinary differential equations (2.53c)–(2.53e) derived directly from the Euler equations of fluid dynamics. Depending on the viewpoint we can give the following interpretations:

The grid generation interpretation: The inverse Euler equations can be interpreted as a special type of grid generation procedure. In fact the scaled Laplace equations (2.58) are well known in the field of elliptic grid generation [37]. The elliptic grid generation procedures are characterized by the procedure they use to place the grid lines, i.e. where to refine. In the inverse Euler equations, the position of the grid lines, or of the streamlines, is determined by the factor $\Phi = \frac{h^2}{J}$. This geometrical factor is found by solving the momentum conservation (2.53c). We can therefore say that our method for inverse design consists in a grid generation procedure driven by physical considerations for placing the grid lines.

The flow problem interpretation: The inverse Euler equations can be thought of as being the Euler equations written in a special system of coordinates. In this case the main equation are (2.53c)–(2.53e). The remaining equations can be interpreted as a linear combination of the continuity equation written once for the natural coordinate σ and once for the stream–function ψ.

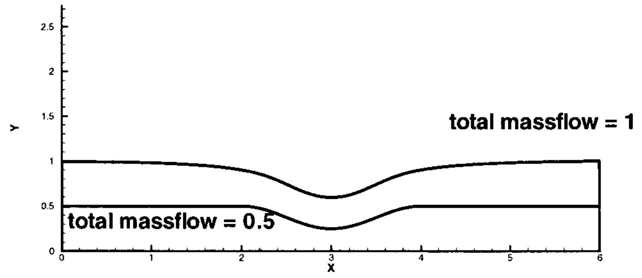

Figure 2.4: *Comparison of the shapes of two Laval nozzles for different total massflows in the case of an inverse computation. The total massflow determines the height of the channel.*

2.9 Definition of the Computational Domain

By construction the lower and upper wall of the unknown device are mapped onto iso–lines of ψ; without loss of generality we can assume that ψ ranges from 0 to a still to be specified value ψ_{max}. By definition the quantity ψ_{max} is equal to the total massflow through the inlet or, by the continuity equation, outlet surface. Thus

$$\psi_{max} = \int_{inlet} \rho \mathbf{u} \cdot \mathbf{n} dS. \tag{2.64}$$

Inverse Design In an inverse computation, the height of the nozzle or diffuser is not given a–priori since only a pressure distribution is known but no geometry. In this case, the total massflow through the device must be prescribed. The resulting geometry will adjust itself in order to produce the correct massflow through the device. In Figure 2.4 we compare two nozzles which are generated by the same pressure distribution along the upper wall but have different total massflows. In this case the inlet conditions consist of a uniform flow with density $\rho = 1$ and velocity $\mathbf{u} = (1, 0)^T$. We clearly see that in order to generate a total massflow of 0.5 the height of the channel must be 0.5 while for a total massflow of 1.0 the height must be 1.0.

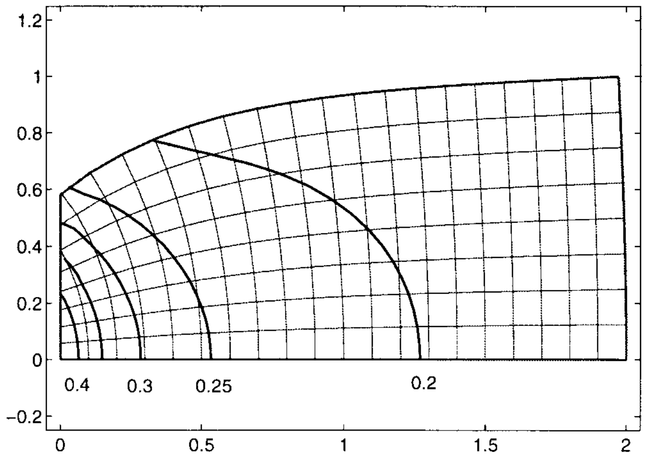

Figure 2.5: *Bell–shaped diffuser with non–isopotential inlet. The orthogonal grid and the Mach number contours are shown. The inlet does not coincide with a grid line. (Courtesy of A. Troxler)*

Direct computation The situation is slightly different in the case of a direct computation where the total massflow results from the difference between the inlet total pressure and the outlet static pressure. In the special case of uniform (irrotational) inlet flow conditions, ψ_{max} can be computed explicitly

$$\psi_{max} = \rho \mathbf{u} \cdot \mathbf{n} A, \tag{2.65}$$

where A is the area of the inlet surface. Using (2.65) we can define the computational domain in the case of a potential flow. In this case we can choose the outlet pressure to be compatible with the inlet total massflow.

In the case of a rotational flow ψ_{max} cannot be found a priori since the flow conditions at inlet and outlet are not fully known. The quantity ψ_{max} can be found iteratively by updating it after every iteration by integrating (2.64).

We assume further that inlet and outlet surfaces coincide with iso–lines of σ. This does not hold in general as seen in Figure 2.5. A procedure for handling this problem has been developed by Troxler [46]. A consequence of this simplifying assumption is that the inlet and outlet are perpendicular to the flow. Without loss of generality we assume that σ ranges form 0 to σ_{max}.

The computational domain in the (σ, ψ) plane is the rectangle $[0, \sigma_{max}] \times$

$[0, \psi_{max}]$. The value of σ_{max} is determined by the length of the reference streamline, e.g. at $\psi = 0$. For this length L it holds

$$L = \int_{\text{on streamline}} ds = \int_0^{\sigma_{max}} h d\sigma = \sigma_{max}, \tag{2.66}$$

where ds is the length element in the physical space (x, y) and under the assumption that $h^2 = 1$ along $\psi = 0$. This is imposed as a boundary condition in section 2.11. Recalling (2.38) we conclude that prescribing $h^2 = 1$ along one streamline is equivalent to set its length. In practice we assume that the wall at $\psi = 0$ is given, we compute its length L and we set $\sigma_{max} = L$.

In the case of two inverse walls, the length of a streamline can be freely chosen. It must be kept in mind that this length will determine the pressure gradient along the side wall.

2.10 Flow Field–Geometry Relations

The grid, or better the mapping

$$(x, y) \rightarrow (\sigma(x, y), \psi(x, y)) \tag{2.67}$$

is related to the flow velocity by means of the definitions (2.27) and (2.29). By exploiting the relations derived in section 2.5 similar relations for the inverse mapping

$$(\sigma, \psi) \rightarrow (x(\sigma, \psi), y(\sigma, \psi)) \tag{2.68}$$

can be found. This type of relations follows by considering the definition of the stream–function and the transformation rule (2.40)

$$\rho \mathbf{u} = \nabla^{\perp} \psi = \frac{1}{J} \mathbf{x}_\sigma \tag{2.69}$$

and thus by (2.38)

$$m := \|\rho \mathbf{u}\| = \rho \sqrt{u^2 + v^2} = \frac{h}{J}. \tag{2.70}$$

m is called specific massflow. An alternative formulation to (2.70) can be found by using the generalized Cauchy–Riemann equations (2.52)

$$\|\mathbf{x}_\psi\| = \frac{J}{h^2} \|\mathbf{x}_\sigma\| = \frac{J}{h} = \frac{1}{\|\rho \mathbf{u}\|}. \tag{2.71}$$

We notice that the mapping $\mathbf{x} = \mathbf{x}(\sigma, \psi)$ becomes singular in the case of a stagnation point since $\|\mathbf{u}\| \to 0$ and $J = \frac{h}{\|\rho\mathbf{u}\|} \to \infty$ under the reasonable assumption that h is $\mathcal{O}(1)$ in the whole domain.

The remaining flow variables in (2.53a)–(2.53e) can be found from m by means of the isentropic gas identities for an ideal gas:

$$m^2 = \rho_0^2 c_0^2 \frac{M^2}{\left(1 + \frac{\gamma-1}{2} M^2\right)^{\frac{\gamma+1}{\gamma-1}}} \tag{2.72}$$

$$T = T_0 \frac{1}{1 + \frac{\gamma-1}{2} M^2} \tag{2.73}$$

$$\rho = \rho_0 \left(\frac{T}{T_0}\right)^{\frac{1}{\gamma-1}} \tag{2.74}$$

$$p = \rho R T. \tag{2.75}$$

In (2.72)–(2.74) we assume that the stagnation quantities ρ_0, c_0, T_0 are known. As we will see in section 2.11, this is the case. T_0 can be thought as being the temperature of the fluid if it were brought to rest isentropically. Similar characterisations hold for ρ_0 and p_0. As explained in section 2.2 the rotational effects are given by non–vanishing gradients of H and s which are a consequence of non–uniform stagnation quantities.

In the case of a swirling axis–symmetric flow an additional term taking into account the angular momentum must added to (2.72)–(2.74) as described in section 2.12.

2.11 Boundary Conditions

The aim of this work is to develop a method for the design of internal flow configurations. In particular we would like to be able to keep the same boundary conditions as in the direct computation.

For a subsonic internal flow configuration such as a diffuser or a nozzle, the correct boundary conditions are [30]

inlet: the flow angle α, and the stagnation pressure p_0 and temperature T_0 are prescribed. In particular we can compute the total enthalpy $H = c_p T_0$

and integrate the first law of thermodynamics (2.17) to get the entropy distribution.

outlet: the static pressure distribution is prescribed. This is a signal of the elliptic character of the flow.

side walls (standard): the tangential flow condition $\mathbf{u} \cdot \mathbf{n} = \mathbf{0}$ is imposed. This coincides with the fact the the stream–function ψ is constant on the side walls.

side walls (inverse): a pressure distribution is imposed. At the outlet corners, the side wall pressure distribution must be compatible with the outlet pressure distribution.

We remark that these boundary conditions originate from the analysis of the characteristic relations for the Euler equations [30], and that they make physically sense as for example the outlet pressure of an exhaust diffuser is known to be the atmospheric pressure.

For the solution of the target-pressure problem we need to reformulate this set of boundary conditions in inverse coordinates. Since the system of the scaled Laplace equations (2.58) is of second order and elliptic, we need to impose two boundary conditions along each wall in order to have a unique solution. First of all we discuss the boundary condition arising from the flow equations.

flow angle BC: the flow angle α can be imposed by the equation

$$y_\sigma - \tan(\alpha) x_\sigma = 0. \tag{2.76}$$

We recall that the inlet wall is orthogonal to the flow.

stagnation quantities BC: the stagnation quantities are prescribed as a function of ψ. The resulting integrals of motions are used in the thermodynamical formulas (2.72)–(2.74).

outlet: in a preprocessing step the imposed static pressure is transformed to a specific massflow distribution m_{outlet} by (2.72)– (2.74). This is motivated by the fact that the simplest geometry — flow relation (2.70) is based on the specific massflow. The resulting specific massflow distribution is imposed by the equation

$$\Phi - J m_{outlet}^2 = 0, \tag{2.77}$$

where $\Phi = \frac{h^2}{J}$. (2.77) follows directly from (2.70) by inserting the definition of Φ.

side walls (standard): the slip boundary condition is automatically satisfied and the shape of the configuration is specified by an algebraic equation

$$f(x, y) = 0. \tag{2.78}$$

In [46] a method for handling geometric boundary conditions in parametric form is described.

side walls (inverse): the imposed static pressure is transformed to a specific massflow distribution m_{wall} which is imposed by the equation

$$\|\mathbf{x}_\psi\|^2 - \frac{1}{m_{wall}^2} = x_\psi^2 + y_\psi^2 - \frac{1}{m_{wall}^2} = 0 \tag{2.79}$$

which follows from (2.71). Again, the motivation for the transformation of the pressure to a specific massflow is given by the fact that the geometry — flow relations are simpler for the specific massflow than for the pressure.

These boundary conditions are all related to the state of the flow. The second boundary condition must be related to the grid generation, and in particular must enforce the grid orthogonality.

orthogonality BC: along all four boundaries we impose the orthogonality of the grid in the form

$$\mathbf{x}_\sigma \cdot \mathbf{x}_\psi = x_\sigma x_\psi + y_\sigma y_\psi = 0. \tag{2.80}$$

A justification of this boundary condition for irrotational, incompressible flows is given in section 2.7, Proposition 2.

In this way we set all the necessary BC for the scaled Laplace equations.

The inverse form of Crocco's theorem (2.53c) is an ordinary differential equation in ψ direction and as such needs an initial value along the $\psi = 0$ boundary.

length BC: We prescribe h^2 along the boundary $\psi = 0$, i.e. we impose a given length to the streamline along the lower wall. This BC is implemented by imposing the equation

$$\Phi J - 1 = 0, \tag{2.81}$$

on $\psi = 0$. By means of (2.81), the distance in flow direction between two successive σ iso–lines is set.

The uniqueness of the solution is not yet assured. To see this one should consider the target–pressure problem not as a fluid flow problem, but just as a geometrical problem. In particular a solution remains valid if it is displaced or rotated. To avoid the problem of the orientation we set the flow angle and to avoid the problem of the displacement we set the

displacement BC: the corner ($\sigma = 0, \psi = 0$) must be mapped onto $x = 0, y = 0$.

A schematic review of the boundary conditions can be found in Figure 2.6.

2.12 Axis–Symmetric Flows

The inverse Euler equations can be extended to handle axis–symmetric flows with swirl. Here we simply state the equations and present the differences with respect to the two dimensional case. Further details can be found in [36][47] and in chapter 5.

The definition of ψ is modified to

$$\rho u = \frac{1}{r}\partial_r \psi \tag{2.82a}$$

$$\rho v = -\frac{1}{r}\partial_x \psi, \tag{2.82b}$$

in order to take the radius changes into account, while the definition of σ does not change. We denote by $(u, v, w)^T$ the axial, radial and azimuthal component of the velocity vector.

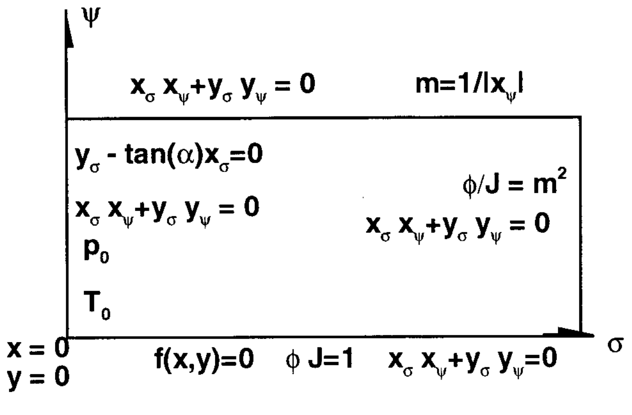

Figure 2.6: *Schematic view of the boundary conditions for the inverse Euler equations.*

The inverse Euler equations for smooth, inviscid, axis–symmetric flows are

$$\partial_\sigma\left(\frac{J}{h^2}\partial_\sigma \mathbf{x}\right)+\partial_\psi\left(\frac{h^2}{J}\partial_\psi \mathbf{x}\right)=0 \tag{2.83a}$$

$$\partial_\psi H - T\partial_\psi S - \frac{C}{r^2}\partial_\psi C = \frac{1}{\rho r J}\partial_\psi\left(\frac{h^2}{\rho r J}\right). \tag{2.83b}$$

$$\partial_\sigma s = 0 \tag{2.83c}$$

$$\partial_\sigma H = 0 \tag{2.83d}$$

$$\partial_\sigma C = 0. \tag{2.83e}$$

The unknowns are the axial and radial coordinates $\mathbf{x} = (x, r)^T$ and the integrating factor h. The determinant of the Jacobi matrix of the transformation can be expressed as $J = x_\sigma r_\psi - x_\psi r_\sigma$. There is an additional integral of motion, the angular momentum $C = rw$, as one easily deduces from the momentum conservation in azimuthal direction. The possibility of handling a non–

vanishing angular momentum provides an extension to the axis–symmetric inverse design method [1].

The geometry–flow field relations are based on the same gas dynamics identities. In order to take into account the azimuthal component of the velocity, we modify the definition of the total enthalpy H to

$$H = c_pT + \frac{1}{2}(u^2 + v^2 + w^2). \tag{2.84}$$

The density can be computed by

$$\rho = \rho_0 \left(1 - \frac{\frac{h^2}{r^2\rho^2J^2} + \frac{C^2}{r^2}}{2H} \right)^{\frac{1}{\gamma-1}} \tag{2.85}$$

since

$$(u^2 + v^2 + w^2) = \left(\frac{h}{r\rho J}\right)^2 + \left(\frac{C}{r}\right)^2. \tag{2.86}$$

The boundary conditions are modified as follows. On an inverse wall we prescribe a pressure distribution, which is first transformed into a Mach number distribution and then into a specific massflow distribution by the isentropic gas identities in order to apply the simplest flow — geometry relations. On an inverse side wall we impose the equation

$$\|\rho\mathbf{u}\|^2 = \rho^2(u^2 + v^2) = \frac{1}{r^2\|\mathbf{x}_\psi\|^2} \tag{2.87}$$

Notice that the differece with respect to the planar case (2.71) consists just in an additional r factor. The outlet boundary condition is implemented as

$$\frac{h^2}{r^2J^2} = m^2 = \rho^2\left(u^2 + v^2\right). \tag{2.88}$$

In both cases we consider only the specific massflow in the meridional (x, r) plane. It is worth to notice that the pressure depends via (2.84) from the full velocity vector, and in particular from the angular momentum C. There are no conceptual problems in prescribing the full three-dimensional specific massflow as a boundary condition. A code implementing this feature is presented in [56].

The angular momentum $C = rw$ is given at inlet as additional boundary condition, either directly or by prescribing $w(\psi)$.

2.13 Conclusions

In this chapter we presented the inverse Euler equations. By developing the algebra of Keller [36] in the physical space, the main ideas of the method were pointed out and we showed that they cannot be extended to general three dimensional flows. This is a strong limitation of all inverse methods based on a natural coordinate.

In the two dimensional and axis–symmetric case the strength of the inverse Euler equations lies in the boundary conditions. On one side, inlet and outlet allow exactly the same boundary conditions as in a standard CFD code, while all other single–pass inverse methods are based on ad hoc inlet and outlet boundary conditions. Flexibility is guaranteed on the side walls too. Unlike other single–pass methods we can prescribe any mixture of geometrical and inverse boundary conditions.

The main change with respect to the original paper of Keller [36] consists in substituting the generalized Cauchy–Riemann equations by the scaled Laplace equations. Even in its simplicity, this change has a great impact on the efficiency of the numerical method described in the next chapter.

Seite Leer /
Blank leaf

Chapter 3

Discretization and Numerical Method

The inverse Euler equations are a set of nonlinear partial differential equations for which no general analytical solution can be given. In order to find an approximated solution we need to formulate a suitable discretization and to implement an efficient solver.

Since the solutions of the inverse Euler equations are by assumption smooth, combining a discretization based on finite differences on a uniform grid and an inexact Newton–Krylov–ILU solver turns out to be a suitable choice for the numerical solution of the problem.

For sake of completeness we recall the system of partial differential equations to be solved

$$\partial_\sigma \left(\frac{J}{h^2} \partial_\sigma \mathbf{x} \right) + \partial_\psi \left(\frac{h^2}{J} \partial_\psi \mathbf{x} \right) = 0 \qquad (3.1a)$$

$$\partial_\psi H - T \partial_\psi s = \frac{1}{\rho J} \partial_\psi \left(\frac{h^2}{\rho J} \right). \qquad (3.1b)$$

Equations (3.1a)– (3.1b) are augmented by the boundary conditions described in section 2.11 and by the thermodynamic relations (2.72)–(2.74).

This chapter is structured as follows. First we present a second order discretization of the inverse Euler equations based on finite differences. In section 3.2 we describe the algorithm for the solution of the algebraic system of equations arising form the discretization of (3.1). Section 3.3 is devoted to the validation of the method by means of a numerical study of convergence and of the so called "recovery problem". In the last section we introduce some simplifications in the solution procedure and compare the execution times of the resulting methods. A comparison with a totally different solver for the inverse Euler equations is also included.

3.1 Discretization

3.1.1 Setup

The particular choice of the independent coordinates (σ, ψ) allows us to pose the target–pressure problem in the rectangle $\Omega = [0, \sigma_{\max}] \times [0, \psi_{max}]$. In Ω we introduce a uniform grid, the grid points are given by

$$(\sigma_i, \psi_j) = (i\Delta\sigma, j\Delta\psi) \text{ for } i = 0, \cdots, N_\sigma, j = 0, \cdots, N_\psi \tag{3.2}$$

where $\Delta\sigma = \frac{\sigma_{max}}{N_\sigma}$, $\Delta\psi = \frac{\psi_{max}}{N_\sigma}$. N_σ and N_ψ denote the number of cells in every spatial direction.
On this grid we introduce the standard difference operators

$$D_\sigma^0 f(\sigma_i, \psi_j) = \frac{f(\sigma_i + \Delta\sigma, \psi_j) - f(\sigma_i - \Delta\sigma, \psi_j)}{2\Delta\sigma} \tag{3.3}$$

$$D_\sigma^+ f(\sigma_i, \psi_j) = \frac{f(\sigma_i + \Delta\sigma, \psi_j) - f(\sigma_i, \psi_j)}{\Delta\sigma} \tag{3.4}$$

$$D_\sigma^- f(\sigma_i, \psi_j) = \frac{f(\sigma_i, \psi_j) - f(\sigma_i - \Delta\sigma, \psi_j)}{\Delta\sigma} \tag{3.5}$$

and their counterparts in ψ direction

$$D_\psi^0 f(\sigma_i, \psi_j) = \frac{f(\sigma_i, \psi_j + \Delta\psi) - f(\sigma_i, \psi_j - \Delta\psi)}{2\Delta\psi} \tag{3.6}$$

$$D_\psi^+ f(\sigma_i, \psi_j) = \frac{f(\sigma_i, \psi_j + \Delta\psi) - f(\sigma_i, \psi_j)}{\Delta\psi} \tag{3.7}$$

$$D_\psi^- f(\sigma_i, \psi_j) = \frac{f(\sigma_i, \psi_j) - f(\sigma_i, \psi_j - \Delta\psi)}{\Delta\psi}. \tag{3.8}$$

Strictly speaking, the unknowns in the inverse Euler equations are x, y, h. Anyway, for sake of an easier implementation we prefer to work with x, y, $\Phi := \frac{h^2}{J}$ and ρ as unknowns. Their nodal values are denoted by

$$x_{i.j} := x(i\Delta\sigma, j\Delta\psi), y_{i.j} := y(i\Delta\sigma, j\Delta\psi), \cdots \tag{3.9}$$

3.1.2 Interior of the Domain

The scaled Laplace equations can be rewritten as

$$\frac{J}{h^2}\mathbf{x}_{\sigma\sigma} + \frac{h^2}{J}\mathbf{x}_{\psi\psi} + \partial_\sigma\left(\frac{J}{h^2}\right)\mathbf{x}_\sigma + \partial_\psi\left(\frac{h^2}{J}\right)\mathbf{x}_\psi = 0 \tag{3.10}$$

since by assumption the solution is smooth. In the interior of the domain, where the scaled Laplace equations hold, we use central differences for the first order derivatives and the standard second order discretization for the Laplace operator. The discretized equations read

$$\frac{1}{\Phi_{i.j}} D_\sigma^+ D_\sigma^- \mathbf{x}_{i.j} + \Phi_{i.j} D_\psi^+ D_\psi^- \mathbf{x}_{i.j} + D_\sigma^0\left(\frac{1}{\Phi_{i.j}}\right) D_\sigma^0 \mathbf{x}_{i.j} + D_\psi^0 \Phi_{i.j} D_\psi^0 \mathbf{x}_{i.j} = 0. \tag{3.11}$$

Equations (3.11) hold for $i = 1, \cdots N_\sigma - 1$, $j = 1, \cdots N_\psi - 1$.

Crocco's theorem (3.1b) is discretized by the implicit trapezoidal rule [29]

$$\left(\frac{\Phi}{\rho}\right)_{i.j+1} = \left(\frac{\Phi}{\rho}\right)_{i.j} + \Delta\psi\left(F_{i.j+1} + F_{i.j}\right) \tag{3.12}$$

for $i = 0, \cdots, N_\sigma$ and $j = 0, \cdots, N_\psi - 1$. We set $F = \rho J\left(\partial_\psi H - T\partial_\psi s\right)$.

The determinant of the Jacobi matrix of the transformation J is discretized in (2.70) and (2.74), relating the density to the geometry, by central differences.

3.1.3 Boundary Conditions

The boundary conditions are discretized by finite differences in the same way as the equations in the interior of the domain. Nevertheless there is one critical point. In order to use a preconditioner based on the LU decomposition such as

ILU, or in order to use an iterative solver, such as Gauss–Seidel, for the linear system we need to be sure that all the elements on the diagonal of the Jacobi matrix are different from zero. The central difference operator D_σ^0 cannot be used on the boundary !

On the boundaries $\psi = 0, \psi = \psi_{max}$ we use the following third order finite difference approximation of the tangential derivative ∂_σ

$$\partial_\sigma \mathbf{x}_{i,j} \approx \left(\frac{1}{6}\mathbf{x}_{i-2,j} - \mathbf{x}_{i-1,j} + \frac{1}{2}\mathbf{x}_{i,j} + \frac{1}{3}\mathbf{x}_{i+1,j} \right) \frac{1}{\Delta\sigma}. \tag{3.13}$$

The remaining derivatives are discretized by one–sided finite differences. For example along $\psi = \psi_{max}$ we apply

$$\partial_\psi \mathbf{x}_{i,j} \approx \left(\frac{3}{2}\mathbf{x}_{i,N_\psi} - 2\mathbf{x}_{i,N_\psi-1} + \frac{1}{2}\mathbf{x}_{i,N_\psi-2} \right) \frac{1}{\Delta\psi}. \tag{3.14}$$

It can be easily seen that the discretization of the orthogonality boundary condition does not have any zeros on the diagonal of the Jacobi matrix.

By construction the discretization is of second order of accuracy. Numerical experiments of section 3.3.1 will confirm this theoretical result.

The discretization at every grid node of the two scaled Laplace equations, of the inverse form of Crocco's theorem and of (2.74) results in a nonlinear system of $4(N_\sigma + 1)(N_\psi + 1)$ algebraic equations which is solved by a Newton–Krylov method.

3.2 The Solution Procedure

The solutions of the inverse Euler equations have two properties which render appealing the use of a Newton based solution procedure. The solutions are by assumption smooth, and thus the Taylor expansion in the derivation of Newton's method is allowed. Further the geometrical character of the problem makes it easy to define a good starting value for the iterative scheme.

3.2.1 The Newton–Krylov Method

A Newton–Krylov method for nonlinear systems of algebraic equations is a Newton method where the linear system is solved by a Krylov subspace iterative method. Since the main components of the method are standard tools in computational mathematics, we just sketch the overall algorithm and give the references in the literature.

Newton's method for solving the system of equations

$$\mathbf{F}(\mathbf{x}^*) = \mathbf{0} \tag{3.15}$$

can be formulated as

$$\begin{aligned} D\mathbf{F}(\mathbf{x}^n)\Delta\mathbf{x}^n &= -\mathbf{F}(\mathbf{x}^n) && (3.16a) \\ \mathbf{x}^{n+1} &= \mathbf{x}^n + \Delta\mathbf{x}^n, && (3.16b) \end{aligned}$$

where the sequence $\mathbf{x}^n$ should approximate the exact solution $\mathbf{x}^*$. $D\mathbf{F}$ is the Jacobi matrix of the functional $\mathbf{F}$. A cubic backtracking method [17], which gives a globally convergent method (see Theorem 6.3.3 in [17]) and a monotone convergence behavior enhances the basic algorithm (3.16).

The linearized equations (3.16a) are solved by the BiConjugate Gradient Stabilized (Bi–CGSTAB) linear solver [58]. Bi–CGSTAB can be seen as an extension of the Conjugate Gradients method to non–symmetric matrices.

An incomplete LU factorization algorithm (ILU) is used as preconditioner for improving the convergence of the iterative solver for (3.16a). It is well known that even for a sparse matrix A, its LU factors are not sparse. The ILU overcomes this problem by applying the LU decomposition only for a set S of indices, such as the set of indices for which $a_{i,j} \neq 0$ [5].

It should be noted that the ILU preconditioner, if applied without pivoting, needs non–zero elements on the diagonal of the matrix $D\mathbf{F}$; for this reason a third order finite difference approximation of $\mathbf{x}_\sigma$ in the boundary conditions of the inverse Euler equations has been introduced, cf. section 3.1.3.

In the sequel we will call this solution algorithm "Newton solver", even though the precise denomination would be inexact Newton–Krylov–ILU method. In particular when analyzing the convergence curves of the solver one should keep in mind that the linear system is not solved exactly and therefore the quadratic convergence characteristic of the standard exact Newton method cannot be expected.

When not otherwise stated, all the numerical examples in this thesis are solved with a standard set of parameters, the most important being the relative termination tolerance of Newton's method which is set to 10^{-7}, of the linear solver set to $10^{-5}$and the number of fill in levels of the preconditioner which is set to 2.

All methods used in this work are implemented in a library of numerical schemes called Portable Extensible Toolkit for Scientific Computation (PETSc) [4]. For more informations on PETSc we refer to the web page [2] and to [3].

3.2.2 Finding the Initial Guess: The Nested Iterations

In order to exploit the fast local convergence of Newton's method, a good initial guess for the iteration must be provided. Our strategy is based on the nested iterations.

We assume that the fine grid on which we look for a solution consists of $2^k N_{\sigma,0}$ cells in σ direction and of $2^k N_{\psi,0}$ cells in ψ direction.

On the very coarse grid with $N_{\sigma,0} \times N_{\psi,0}$ cells an initial guess can be defined by means of the transfinite interpolation [37] between the two side walls of the configuration or by taking some old design. The transfinite interpolation is the simplest method for algebraic grid generation. It just needs a parameterization of the boundaries of the domain to generate a grid by an algebraic formula [37]. The idea is to solve the inverse Euler equations on the coarse grid and then to transfer the resulting solution on the next finer grid, found by doubling the number of cells in each coordinate direction. This interpolated solution is already very accurate and therefore Newton's method will converge very fast. We repeat this procedure until we find a solution on the desired grid.

The interpolation step is based on the standard bilinear interpolation.

In Figure 3.1 we plot, in a logarithmic scale, the residual of Newton's iteration on a sequence of nested grids versus the number of iterations. One clearly sees that the convergence is very fast on the fine level. Since the linear systems are not solved exactly, we do not expect the local quadratic convergence of Newton's method. In Figure 3.2 we present the convergence curve of Newton's method on the finest grid without computing the initial guess by nested iterations. We clearly recognize the long searching phase of the algorithm.

The qualitative behavior of the method alone, does not indicate if the use of the nested iterations introduces a gain in computing time. For the test case

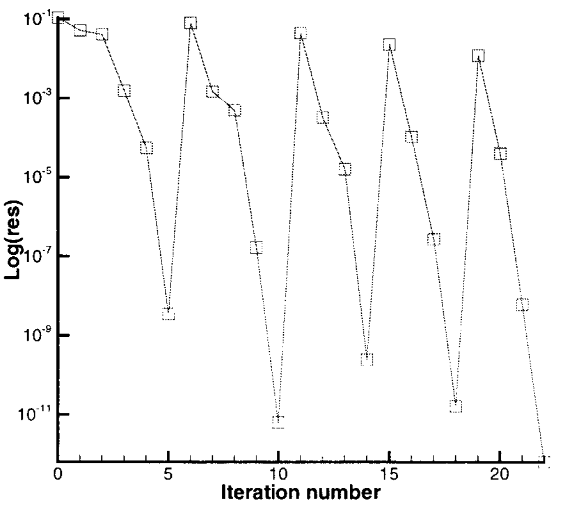

Figure 3.1: *Convergence curve of Newton's method coupled with the nested iteration. The evolution of the residual on five nested grids is shown.*

presented in section 4.1.1 we find the CPU times reported in Table 3.1. The starting value of the simple Newton method is found by a transfinite interpolation of the given geometry (4.2)–(4.3) which can be thought of as being an existing design to be improved. The computing times for the scheme accelerated by the nested iterations are cumulative, i.e. they are the sum of the execution times on all levels reported in Table 3.6. A speedup of a factor 6 can be obtained by simply coupling the standard Newton method to a nested iteration strategy. Further the speedup increases as the grid is refined.

More insight in this behavior can be obtained by looking at how much time is spent on every level with respect to the total execution time for example on the 384 $\times$ 32 grid. As seen in Table 3.2 the CPU time spent in the coarse levels is about 11 % of the total computing time, which is comparable with $\frac{1}{3}$ of an iteration on the finest level. The nested iterations produce a very good initial guess on the finest grid. The number of iterations on the grid where an iteration step is expensive are reduced at a cost which is almost negligible. The influence of the nested iterations on the overall method is not only restricted to

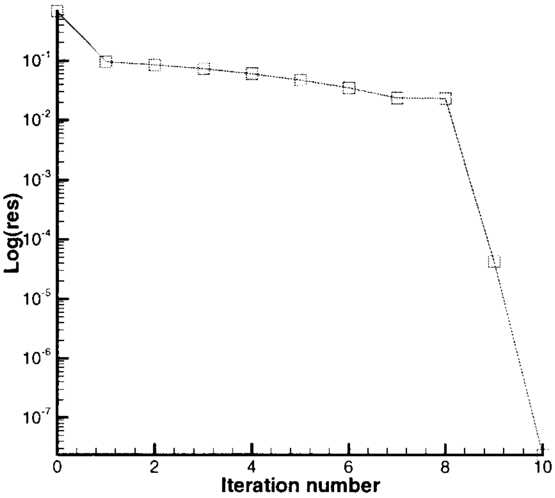

Figure 3.2: *Convergence curve of Newton's method without the nested iterations, i.e. on a fixed grid. Most of the computing time (up to 90%) is spent in the searching phase where the residual barely drops.*

Level	$N_\sigma \times N_\psi$	Newton	Newton + NI	Speedup
0	24 × 2	2.56e-01		
1	48 × 4	1.19e+00	5.43e-01	2.2
2	96 × 8	4.49e+00	1.71e+00	2.62
3	192 × 16	7.41e+01	1.41e+01	5.25
4	384 × 32	8.01e+02	1.30e+02	6.16

Table 3.1: *Comparison of the cumulative CPU times for the method with and without the nested iteration strategy (NI). By a simple procedure the computing time can be reduced significantly.*

a speedup. In fact finding a starting value on a very coarse grid is much easier than on a fine grid since the attempts are less CPU intensive and the system is easier to solve. Further, once a converged solution on the coarse grid is found, no user interaction is needed to compute the solution on the finer levels.

Level	$N_\sigma \times N_\psi$	% CPU Time
0	24×2	0.15
1	48×4	0.22
2	96×8	0.90
3	192×16	9.53
4	384×32	89.2

Table 3.2: *Percentage of the computing time spent on each level with respect to the total execution time. Only* 10% *of the total computing time is spent in the computation of the initial guess.*

3.3 Validation of the Code

We validate our code in three different ways. First we check that the predicted order of accuracy is obtained. Secondly, we analyze the error of the angle between grid lines and the arc–length with respect to consistent values of the solution of the Cauchy–Riemann equations (2.52), see section 2.7. In section 3.3.2 we present results for the third type of validation, the so called "recovery problem". In the recovery problem we solve the flow equations by means of a commercial flow solver on a geometry found by the inverse method. The validation consists in comparing the specific massflow distribution found by the flow solver with the one we imposed to the inverse design tool.

3.3.1 Numerical Study of the Order of Accuracy

The first step in the validation of the code is a numerical study of the convergence rate. We consider a rotational flow through a constant radius bend, for which an exact solution can be given

$$
\begin{aligned}
x(\sigma, \psi) &= \sin(\sigma)\exp(\frac{\psi}{2} + 1) && (3.17)\\
y(\sigma, \psi) &= \cos(\sigma)\exp(\frac{\psi}{2} + 1) - \exp(1) && (3.18)\\
\Phi &= 2 && (3.19)\\
\rho(\sigma, \psi) &= \frac{1}{\exp(\psi + 2)}. && (3.20)
\end{aligned}
$$

$N_\sigma \times N_\psi$	L^∞ norm	
	Discr. Error	Factor
5×5	4.38e-02	
10×10	1.12e-02	3.91
20×20	2.80e-03	4.00
40×40	7.02e-04	3.99
80×80	1.76e-04	3.99
160×160	4.40e-05	3.99
320×320	1.10e-05	3.99

Table 3.3: *Discretization error and convergence rate of the numerical method in the maximum norm. The second order of convergence is clearly recognizable.*

The inlet boundary conditions are

$$T_0(\psi) = (2\psi + \frac{14}{3})\exp(\psi + 2) \tag{3.21}$$

$$p_0(\psi) = 2\frac{(\psi + 7/3)^3}{(\psi + 2)^2} \tag{3.22}$$

$$\alpha(\psi) = 0, \tag{3.23}$$

while at the outlet and along the side walls we prescribe the specific massflow

$$m(\psi) = 2\exp(-(\frac{\psi}{2} + 1)). \tag{3.24}$$

On a $N_\sigma \times N_\psi$ grid we measure the discretization errors in the maximum norm

$$\|e\|_{L^\infty} = \max_{i,j} \|\mathbf{X}_{i,j} - \mathbf{X}(\sigma_i, \psi_j)\|_\infty \tag{3.25}$$

where $\mathbf{X} = (x, y, \rho, \Phi)$ is the vector of the unknowns and $\|\cdot\|_\infty$ is the usual maximum norm of a vector.

The results of the numerical study of convergence are summarized in Table 3.3. The second order of convergence can be easily seen. The theoretical predictions of section 3.1 are confirmed.

The choice of the scaled Laplace equations system (2.58) instead of the Cauchy–Riemann equation (2.52) is very favorable from the numerical point

	Orthogonality		Arc–length	
$N_\sigma \times N_\psi$	MDO	Factor	L^∞–Error	Factor
5×5	1.11e-03		3.62e-02	
10×10	1.97e-04	5.62	9.81e-03	3.68
20×20	2.81e-05	6.99	2.55e-03	3.85
40×40	4.18e-06	6.72	6.52e-04	3.91
80×80	6.29e-07	6.65	1.66e-04	3.94
160×160	8.84e-08	7.11	4.18e-05	3.96
320×320	1.19e-08	7.40	1.05e-05	3.98

Table 3.4: *Convergence rate of the MDO and of the arc–length.*

of view but unfortunately one loses *a–priori* the orthogonality of the grid and the equivalence of the arc–length with the stretching factor

$$h^2 = \Phi J = \|\mathbf{x}_\sigma\|^2. \tag{3.26}$$

In order to analyze these potential problems we introduce two standard measures of the quality of an orthogonal grid [22] [21] [53]. We define the maximum deviation from orthogonality MDO and the mean deviation from orthogonality ADO as

$$\begin{aligned} MDO &= \max_{i,j} |\frac{\pi}{2} - \theta_{i,j}|, & (3.27)\\ ADO &= \frac{1}{(N_\sigma + 1)(N_\psi + 1)} \sum_{i,j} |\frac{\pi}{2} - \theta_{i,j}| & (3.28) \end{aligned}$$

where

$$\cos(\theta) = \frac{\mathbf{x}_\sigma \cdot \mathbf{x}_\psi}{\|\mathbf{x}_\sigma\| \|\mathbf{x}_\psi\|}. \tag{3.29}$$

In Table 3.4 is it shown that the deviation from the orthogonality of the grid lines and the error in the definition of the arc–length are very small and they tend to zero as the grid is refined. Therefore in the limit, the solutions satisfy the original system of the Cauchy–Riemann equations (2.52). Notice that the convergence rate of the MDO is higher than the theoretical one of 2.

More significant results can be obtained by studying the MDO and ADO for the benchmark problem of section 4.1.1 and [43] [15]. The results are summarized in Table 3.5.

$N\sigma \times N_\psi$	ADO	Factor	MDO	Factor
48×4	6.93e-04		8.77e-03	
96×8	2.35e-04	2.95	3.13e-03	2.80
192×16	6.95e-05	3.38	9.55e-04	3.27
384×32	1.90e-05	3.65	3.15e-04	3.04
768×64	4.99e-06	3.81	1.05e-04	3.02
1536×128	1.28-6	3.90	3.27e-05	3.19

Table 3.5: *MDO and ADO convergence behavior for the Laval nozzle problem (cf. section 4.1.1).*

We notice that the ADO converges at almost the theoretical rate while the reduction factor for the MDO is of only about 3.

3.3.2 The Recovery Problem

The validation of a simulation code consists in a numerical study of the order of convergence, consisting in the comparison of the numerical solution with an exact one, and in the case of complex flows, in the comparison with experimental data. The validation of an inverse method has one more stage: the recovery problem.

The recovery problem addresses the following question: given a geometry found by and inverse method from a prescribed specific massflow distribution, will a simulation with an "external" flow solver recover the imposed specific massflow distribution?

We consider as a benchmark the Laval nozzle problem described in section 4.1.1, a slight modification of a test case of the EUROPT project on "Optimum Design Methods in Aerodynamics". In a first step, an inverse computation is carried out with a prescribed specific massflow distribution along the side wall. The resulting geometry (and grid) is used in a second step for a simulation of this configuration with the flow solver NSMB [23]. In Figure 3.3 we compare the specific massflow distribution along the upper wall of the nozzle found by NSMB with the one prescribed in the inverse computation. The agreement is good, as confirmed by the plot of the relative error shown in Figure 3.4.

The inverse design method is therefore validated.

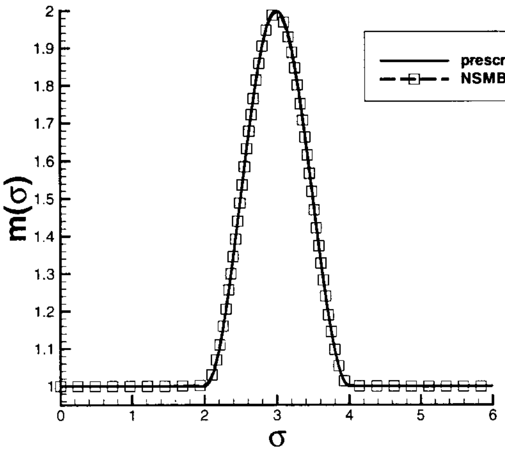

Figure 3.3: *Comparison of the specific massflow distribution computed by NSMB (dashed line and squares) with the prescribed (solid line). The computations are carried out on the grid found by the inverse design tool.*

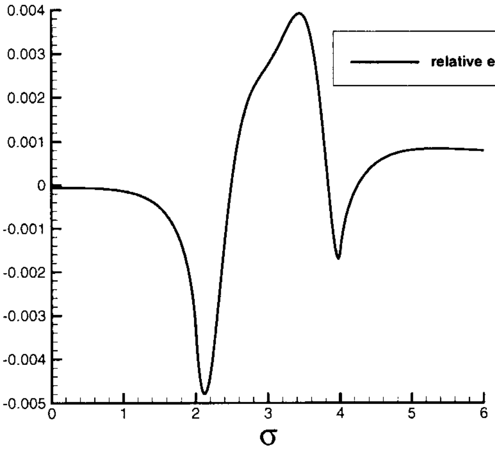

Figure 3.4: *Relative error in the specific massflow distribution along the upper wall of the Laval nozzle.*

Level	Direct	Inverse
1	4.35e-01	2.56e-01
2	1.67e+00	1.17e+00
3	1.40e+01	1.24e+01
4	1.01e+02	1.16e+02

Table 3.6: *Comparison of the CPU time spent on each level for a direct and an inverse computation. The differences are in the order of* 30% *and the method can be said to be single–pass.*

Level	$N_\sigma \times N_\psi$	Nr. BiCGStab Iter.		Nr. Newton Iter.	
		Direct	Inverse	Direct	Inverse
0	24×2	9	5	5	5
1	48×4	39	31	3	3
2	96×8	64	64	3	3
3	192×16	139	145	3	3
4	384×32	229	269	3	3
5	768×64	502	449	3	3

Table 3.7: *Detailed comparison of the execution of the code in a direct and an inverse computation. No significant differences exist in the performance.*

3.4 Performance of the Code

In this section we evaluate the performance of the code. The industrial background of this work defines the goals in terms of performance of the program. A designer is interested is assessing the quality of many designs in a short time. A very high accuracy is not required. The code must therefore run on a workstation and must yield results in about 1-2 minutes for typical grid sizes. Computationally intensive simulations are done only at advanced stages of the design.

The computations have been carried out on a Sun Ultra–Enterprise workstation with a peak performance of 10 MFlops/second.

3.4.1 Inverse vs. Direct Computations

We described in the introduction different approaches to the design problem. Since most of them are built upon a standard flow solver, the coupling to the geometry is weak and its update is done in an iterative way. This is very similar to a block Jacobi iteration and therefore converges very slowly. In practice an increase in computing time of a factor 5-10 in not uncommon. In this section we show the true single-pass character of our method, where single-pass means that the geometry is part of the solution and that there is almost no difference in the computing time for a simulation run and a design run.

The single-pass property is shown in Table 3.6, where we compare the execution times for the inverse and direct mode of the code. The difference in CPU time lies in the range of about 30% and thus our approach can be said to be truly single-pass. In Table 3.7 a comparison of the number of iterations of the linear solver and the number of Newton iterations is presented. We notice that the inverse mode of the code is, in some cases, a little faster.

Some recent research by Troxler [57] shows that every flow solver can be extended to be a single-pass inverse design tool provided that the flow equations and the grid generation equations are solved in an implicit and coupled way. The poor convergence behavior of many inverse methods in the literature is therefore due to the bad coupling between flow solver and geometry update.

3.4.2 Comparison of Different Solution Strategies

It is a common belief that the construction of the Jacobi matrix and of the preconditioner for the linear system are two of the most computationally expensive parts of a Newton–Krylov method. We present in this section two possible optimizations, and discuss whether they are really effective or not. In the first case we do not build a new preconditioner every step. The second attempt to optimize the code is done by freezing the Jacobian of the system, leading to a simplified Newton method. In the last part of this section we compare the execution times of our Newton solver with a completely different solver for the inverse Euler equations: the Recursive Projection method developed in [56].

Level	$N_\sigma \times N_\psi$	CPU(s)		Speedup	
		Direct	Inverse	Direct	Inverse
0	24 × 2	1.21e+00	1.76e-01		
1	48 × 4	2.46e-01	2.84e-01	1.76	0.91
2	96 × 8	1.03e+00	1.31e+00	1.62	0.89
3	192 × 16	1.091e+01	1.36e+01	1.28	0.91
4	384 × 32	9.86e+01	1.25e+02	1.03	0.93

Table 3.8: *Computing times for Newton method with frozen preconditioner. The speedup is to be understood with respect to the basic algorithm and its execution times given in Table 3.6.*

Frozen Preconditioner. Building the preconditioner is in general a time consuming operation. In this section we would like to understand the influence of this step in the overall efficiency of the code. To this end we built the preconditioner only in the first Newton step and we froze it for the remaining iterations. The timing results are presented in Table 3.8. In the direct case the code is marginally faster. Unfortunately, in the much more interesting case of an inverse computation the code slows down by about 10%. This amounts to the increased number of iterations needed for the linear solver.

Frozen Jacobi Matrix. A simplified Newton method is characterized by the fact that the the Jacobi matrix is not computed in every step. The drawback is that in general the convergence rate deteriorates to first order. To be precise, our implementation of the simplified Newton method consists in computing the Jacobian only in the first two iterations on a given level. This is motivated by the fact that the linear interpolation between levels will not represent correctly some of the derivatives. Comparing Table 3.9 with Table 3.6 we see that the simplified Newton formulation is 5% – 10% faster than the standard Newton formulation in the inverse case, while a speedup of about 10% – 30% can be obtained in the direct case. Further, by comparing the convergence curves in Figure 3.1 and 3.5 we see that the fast convergence is not destroyed by the frozen formulation.

The Recursive Projection Method. Troxler in his diploma thesis [56] [47] developed an iterative solver for the inverse Euler equations based on a nonlinear, collective Gauss–Seidel iterative solver accelerated by a deflation strategy

Level	$N_\sigma \times N_\psi$	CPU(s)		Speedup	
		Direct	Inverse	Direct	Inverse
0	24×2	1.58e-01	1.49e-01		
1	48×4	3.07e-01	2.37e-01	1.42	1.08
2	96×8	1.26e+00	1.05e+00	1.32	1.11
3	192×16	1.04e+01	1.20e+01	1.34	1.03
4	384×32	8.94e+01	1.11e+02	1.13	1.05

Table 3.9: *Computing times for Newton method with frozen Jacobian. The speedup is to be understood with respect to the basic algorithm and its execution times given in Table 3.6.*

Level	$N_\sigma \times N_\psi$	Newton	RPM	Speedup
0	24×2	2.56e-01	4.16e-02	0.02
1	48×4	2.87e-01	2.33e-02	0.08
2	96×8	1.17e+00	1.69e+00	1.44
3	192×16	1.24e+01	1.55e+01	1.25
4	384×32	1.26+e02	1.88e+02	1.49

Table 3.10: *Comparison of the execution times of RPM from [56] and of the Newton solver*

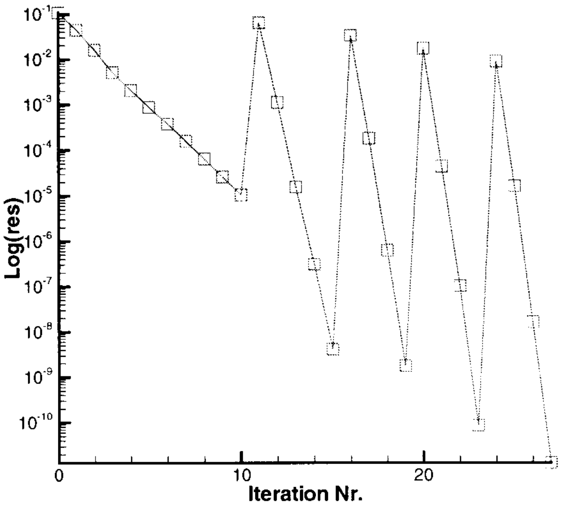

Figure 3.5: *Convergence curve for the simplified Newton method. The evolution of the logarithm of the residual on five nested grids is shown.*

called Recursive Projection Method (RPM) [50] [7]. In Table 3.10 we compare the execution times of the RPM solver and of the Newton method on each level of the grid hierarchy. The Newton scheme is about 1.5 times faster than the RPM code. This consideration, together with the fact that the components of our method are standard and hence of easier use for the non–expert, show that the proposed Newton solver is competitive and can be used in practice.

3.5 Conclusions

We presented a simple, yet effective and efficient numerical method for the solution of the inverse Euler equations. The limitation of the inverse Euler equations to smooth subsonic flows, drives the choice of the finite differences discretization. The Newton solver is greatly enhanced by the coupling with the nested iterations. There is a substantial gain in both speed and robustness. The code can be made marginally faster by reducing the computational cost associated with the construction of the Jacobian (and in some cases of the preconditioner); the robustness suffers marginally from these simplifications since it is more difficult to find a converged solution on the coarsest level.

Seite Leer /
Blank leaf

Chapter 4

Numerical Results

In this chapter we present a gallery of shapes found by our inverse design method. We focus on the different aspects of our method, i.e we show that it can handle a fixed geometry in the same way as a standard simulation tool, as well as a fixed wall and an inverse wall, where the fixed wall can be thought as being a constraint or that it can design both walls from a specific massflow distribution. We present the results in increasing order of difficulty. We begin with two dimensional potential flows, then step to rotational flows to end with the (three dimensional) axis–symmetric case. In the last section we present two parametric studies for a typical configuration of a diffuser in a gas turbine.

4.1 Numerical Results for Potential Flows

Potential flows represent the simplest model of a flow. The governing equations are particularly simple because the integrals of motion are constant not only along streamlines but also across the flow. They are interesting for a first, coarse design and in some cases are very useful as an initial guess for the corresponding rotational problem.

4.1.1 Laval Nozzle

We consider a potential flow through a Laval nozzle. This kind of nozzle is often used in wind tunnels. A possible application of our inverse method is to improve the quality of the flow in the testing part of the wind tunnel. This example is a benchmark test case of the EUROPT Project [43].

The reference (inlet) Mach number is 0.2. The uniform inlet conditions are normalized in such a way that

$$u = 1, T = 1, \rho = 1. \tag{4.1}$$

We consider two different classes of problems. In both cases the lower wall of the nozzle is given by the equation

$$y = 0. \tag{4.2}$$

The first test case consists in a direct computation where the upper wall is prescribed and the we look for the flow. The upper wall is given by

$$y_t(x) = \begin{cases} \frac{1}{2} & \text{for } x \leq 2 \text{ and } x \geq 4 \\ \frac{3}{8} + \frac{1}{8}\sin(\pi(x - \frac{7}{2})) & \text{for } 2 < x < 4 \end{cases}. \tag{4.3}$$

The resulting flow pattern is shown in the two lower pictures of Figure 4.1. We recall that the grid consists of iso–lines of σ and ψ and coincides with the direction of the flow.

The second case consist in an inverse computation where we impose a given specific massflow distribution along the upper wall instead of fixing the geometry. We choose the following specific massflow distribution

$$m_t(\sigma) = \begin{cases} 1 & \text{for } x \leq 2 \text{ and } \sigma \geq 4 \\ \frac{3}{2} - \frac{1}{2}\cos(\pi(\sigma - 2)) & \text{for } 2 \leq \sigma \leq 4 \end{cases}. \tag{4.4}$$

We recall that because of the isentropic relations (2.72)–(2.74), to prescribe a specific massflow distribution is equivalent to the prescription of a pressure distribution. In Figure 4.1 we present the shapes and density contours for both examples. We recognize that the shape arising from the inverse computation is smoother than the one given for the direct computation. This is due to the fact that the specific massflow is found by computing the derivative of the shape mapping $\mathbf{x}(\sigma, \psi)$, see (2.70), and therefore by integration we find a smoother, in this particular case C^2 shape.

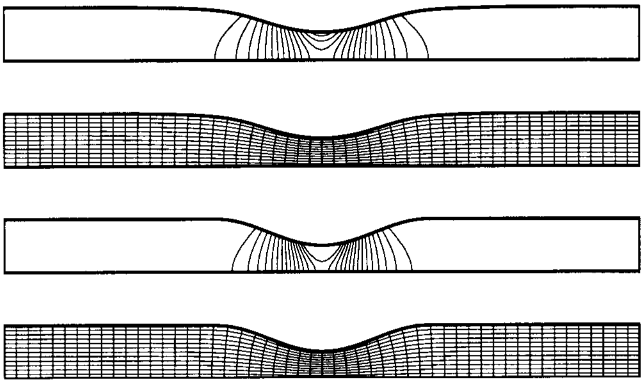

Figure 4.1: *Density contours and shape of the two Laval nozzles, on top in the case of the inverse computation, on the bottom in the direct one.*

4.1.2 Channel Flow

We present a two dimensional elbow channel, whose inlet Mach number is 0.4. The specific massflow distribution along the walls has a transition from 1 at inlet to 0.5 at outlet, therefore the area ratio of the diffuser will be 2. The specific massflow distribution along the walls has the form

$$m(\sigma) = \begin{cases} 1 & \text{for } \sigma \leq 2 \\ \frac{1}{2} & \text{for } \sigma \geq 4 \\ \frac{3}{4} + \frac{1}{4}\cos(\frac{\pi}{2}(\sigma - 2)) \pm \frac{1}{5}(\sigma - 2)(\sigma - 4) & \text{otherwise} \end{cases}, \tag{4.5}$$

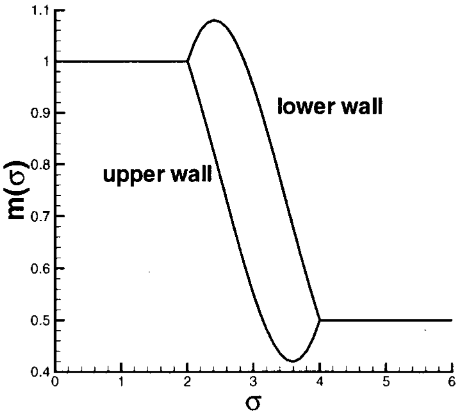

Figure 4.2: *Specific massflow distributions along the side walls for the elbow channel. The corresponding geometry can be found in Figure 4.3*

where the sign $+$ $(-)$ corresponds to the lower (upper) wall. The specific massflow distributions along the diffuser walls are presented in Figure 4.2. In Figure 4.3 we see the orthogonal grid generated by the iso–lines of σ and ψ. The shape of the specific massflow distributions is taken from [11] and [10], where similar specific massflow distributions are used to design blades without considering the leading edge or its stagnation point.

4.2 Numerical Results for Rotational Flows

Rotational flows are characterized by the presence of entropy or enthalpy gradients across the flow. In the inverse formulation of Crocco's theorem

$$\partial_\psi H - T\partial_\psi s = \frac{1}{\rho J}\partial_\psi\left(\frac{h^2}{\rho J}\right) \tag{4.6}$$

the left hand side cannot be dropped, nor (4.6) integrated explicitly as in the potential case.

Rotational flows are relevant since usually the flow after the compressor or

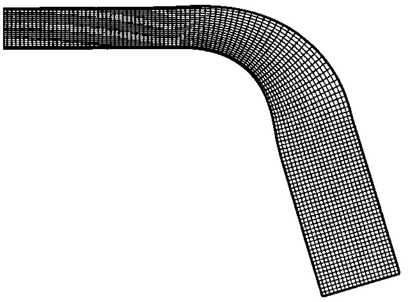

Figure 4.3: *Geometry of the two dimensional channel generated from the specific massflow distributions of Figure 4.2. The iso–lines of* σ *and* ψ *generate an orthogonal grid.*

turbine stages is not uniform and therefore the assumption of a potential flow is plainly wrong.

4.2.1 Laval Nozzle

We consider a rotational flow through a Laval nozzle. The rotational effects are given by the non–uniform total quantities at inlet, in particular we prescribe

$$p_0 = 18.3633 \tag{4.7a}$$

$$T_0 = 1.008 + 0.2(1-\psi)\psi \tag{4.7b}$$

$$\alpha = 0. \tag{4.7c}$$

The lower wall is kept fixed by $y = 0$ and the profile of the upper wall is implicitly defined by the specific massflow distribution

$$m_t(\sigma) = \begin{cases} 1 & \text{for } x \leq 2 \text{ and } \sigma \geq 4 \\ \frac{3}{2} - \frac{1}{2}\cos(\pi(\sigma - 2)) & \text{for } 2 \leq \sigma \leq 4 \end{cases} . \tag{4.8}$$

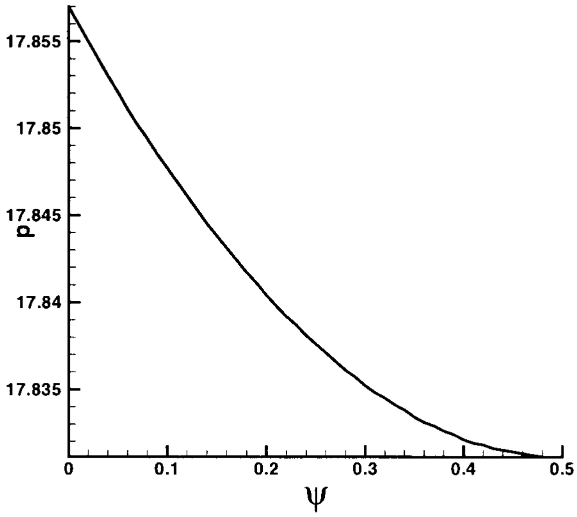

Figure 4.4: *Outlet pressure distribution for the rotational Laval nozzle.*

The total massflow ψ_{max} is 0.5. The outlet boundary condition is given by the pressure distribution of Figure 4.4. This pressure distribution corresponds to a constant outlet specific massflow $m_o(\psi) = 1$. The resulting shape and density contours are given in Figure 4.5.

4.3 Axis–Symmetric Flows

Axis–symmetric flows are of practical interest for the design of gas turbine. In this section we present first the results for a diffuser and then show how our inverse method can be used to evaluate different configurations of a turning diffuser such as the one connecting the compressor to the combustion chamber.

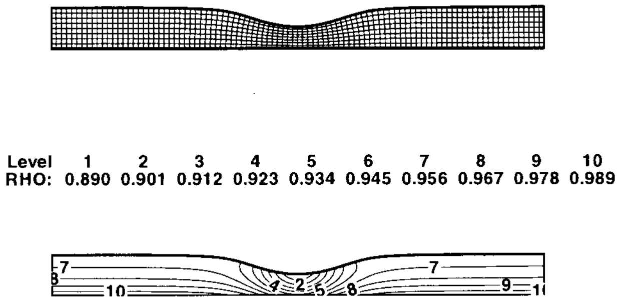

Figure 4.5: *Geometry of the Laval nozzle with prescribed specific massflow distribution along the upper boundary (rotational case). The orthogonal grid is shown in the top picture, while in the lower picture the density contours are plotted.*

4.3.1 Diffuser

We design an axis-symmetric diffuser. The flow is irrotational and the inlet Mach number is 0.8. By normalizing the field quantities by their inlet (reference) values we find that

$$T_0 = 1.128,\ p_0 = 1.70127. \tag{4.9}$$

Further at the inlet we set $\alpha = 0$, $C = 0$. The lower wall is defined by $r = 1$ while the upper wall should generate the massflow distribution

$$m_t(\sigma) = \begin{cases} 1 & \text{for } \sigma \le 2 \\ \frac{1}{4} & \text{for } \sigma \ge 4 \\ \frac{5}{8} + \frac{3}{8}\cos(\frac{\pi}{2}(\sigma - 2)) & \text{for } 2 \le \sigma \le 4 \end{cases} . \tag{4.10}$$

The outlet massflow is $\frac{1}{4}$. The computational domain is $[0, 6] \times [0, 1.5]$, thus the total massflow through the diffuser is 1.5. The grid size is 128×32. The

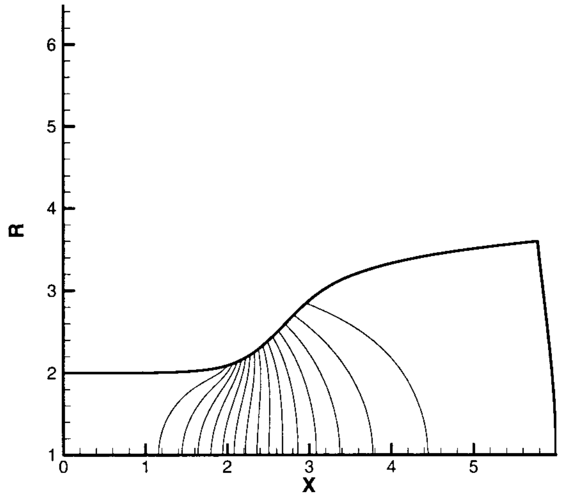

Figure 4.6: *Shape of the axis-symmetric diffuser. Plotted are the contour lines of the Mach number with* $\Delta M = 0.028$ *starting form* $M = 0.770$.

resulting diffuser can be seen in Figure 4.6.

We would like to point out that most of the examples are constructed in such a way that they show interesting features in the geometry, even if a configuration such as the one in Figure 4.6 surely generates a separation bubble in the opening part of the diffuser due to the strong curvature and the big opening angle.

4.3.2 Parametric Studies of an Elbow Channel

We close the gallery of examples by two parametric studies of an elbow channel. The device under consideration is an idealization of the diffuser connecting the compressor to the combustion chamber in a gas turbine. A three dimensional view of this, supposedly, axis–symmetric configuration is given in Figure 4.7 (actually this is the view of one of the diffusers found later on).

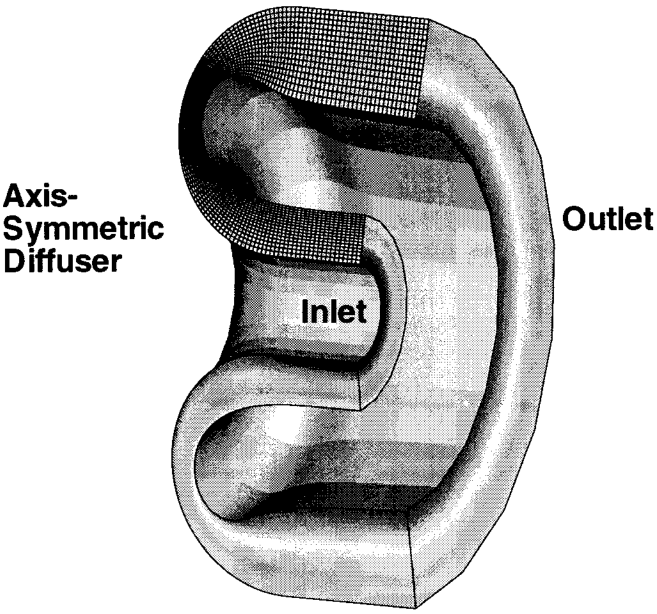

Figure 4.7: *3D view of the axis–symmetric (curved) diffuser. A similar configuration is used to connect the compressor to the combustion chamber is an axial gas turbine.*

We consider an axis–symmetric flow, whose inlet Mach number is 0.4. The external wall of the channel is kept fixed by the algebraic equation $f(x,r) = 0$, where

$$f(x,r) = \begin{cases} r-1 & \text{for } x \leq 2 \text{ and } r \leq 3 \\ (x-2)^2 + (r-3.5)^2 - 6.25 & \text{for } x \geq 2 \\ r-6 & \text{otherwise} \end{cases} . \tag{4.11}$$

The outlet specific massflow is set to 0.3.

In the first case, the distribution of the specific massflow along the internal wall is given by the function

$$m(\sigma) = \begin{cases} 1.0 & \text{for } \sigma \leq \sigma_0 \\ 0.3 & \text{for } \sigma \geq \sigma_0 + 5 \\ 0.65 + 0.35\cos(\frac{\pi}{5}(\sigma - \sigma_0)) & \text{otherwise} \end{cases} . \tag{4.12}$$

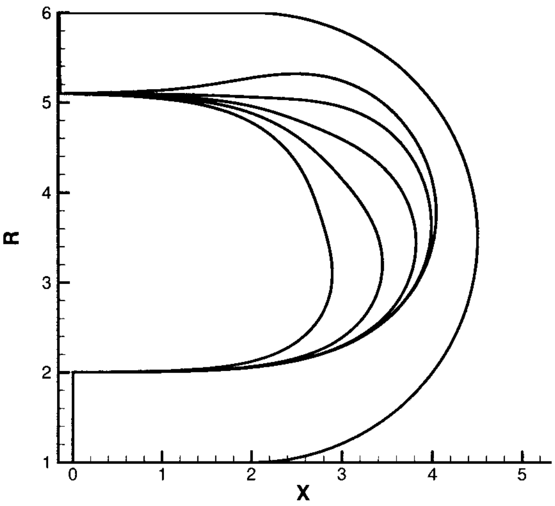

Figure 4.8: *Shapes of the elbow channel for* $\sigma_0 = 2, 3, 4, 5, 6$ *(from left to right).* σ_0 *gives the position of the transition between low and high pressure.*

σ_0 is the parameter defining the position of the transition between the two constant pressure (or specific massflow) states.

The first parametric study is based on changing the value of σ_0. The reasoning behind this is that for a given diffuser and given pressure rise there are different ways to achieve the pressure rise. Since the slope of the specific massflow distribution is limited by the occurrence of separation, we are left with one parameter which will determine the position of the transition between the low and high pressure. In Figure 4.8 we compare the shapes of the diffusers arising from the choices $\sigma_0 = 2, 3, \cdots, 6$, while in Figure 4.9 we present the Mach number contours in the different cases.

In the second parametric study we keep the position of the transition fixed but we allow for different pressure gradients, possibly in the real case ranging in a domain where no separation occurs. The specific massflow distribution

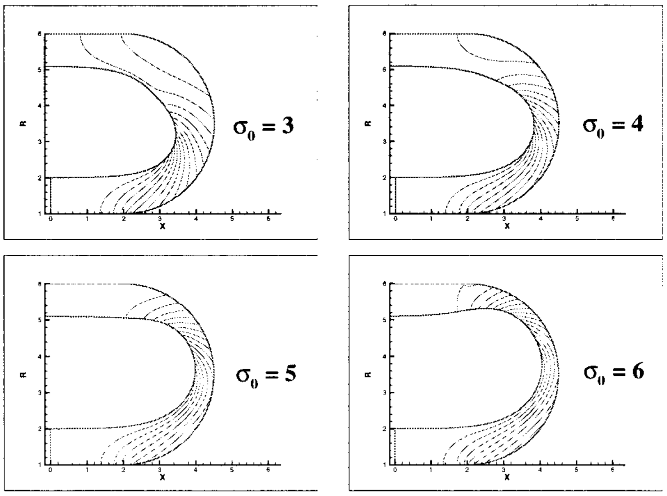

Figure 4.9: *Mach numbers iso–lines at different design points. The different positions of the transition between the two pressure levels are clearly recognizable.*

along the internal wall is given by

$$m(\sigma) = \begin{cases} 1.0 & \text{for } \sigma \leq 4 \\ 0.3 & \text{for } \sigma \geq 4 + \Delta\sigma \\ 0.65 + 0.35\ \cos(\frac{\pi}{\Delta\sigma}(\sigma - 4)) & \text{otherwise} \end{cases} , \qquad (4.13)$$

where $\Delta\sigma$ is the width of the non–constant part of the given specific massflow distribution. We notice that $\Delta\sigma$ drives the maximal magnitude of the gradient of the specific massflow distribution. A plot of $m(\sigma)$ for various values of $\Delta\sigma$ is given in Figure 4.11. We recall that the target–pressure problem was motivated by the fact that one would like to avoid separation in a viscous flow, it is therefore of great interest to know how the form of a diffuser changes in dependence of the specific massflow (or pressure) gradient. The results are presented in Figure 4.12 and 4.10.

4.4 Conclusions

The numerical examples of this chapter give a better feeling of the possible applications of the inverse method developed in this thesis. It seems to the

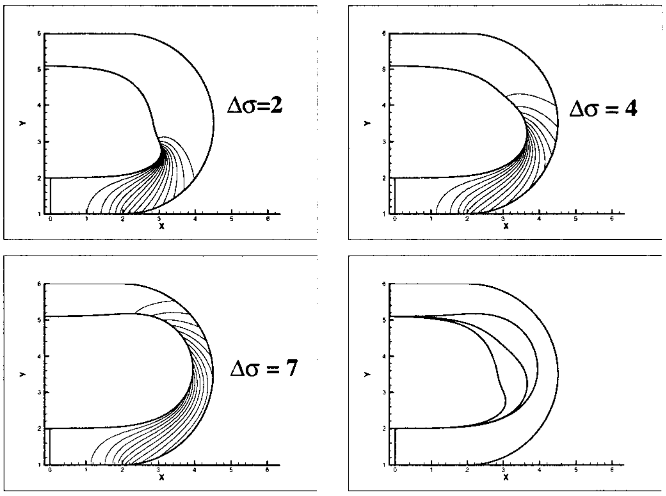

Figure 4.10: *Density contours for* $\Delta\sigma = 2, 4, 7$, *and comparison of the shapes for various pressure gradients. The contour lines show the variation of the width of the transition zone in dependence of* $\Delta\sigma$.

author that the parametric studies based on changing the main characteristic of the boundary pressure distribution can be very useful for devising new designs.

From a mathematical point of view, Figures 4.8 and 4.12 suggest that the shape of the domain depends in a continuous way from the boundary data. The inverse shape design problem is well–posed even though it is an *inverse* problem.

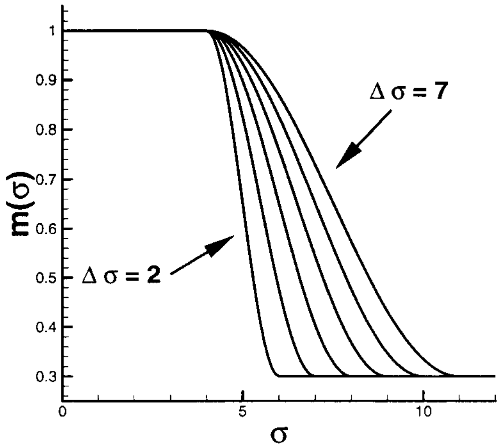

Figure 4.11: *Distribution of the specific massflow along the boundary parametrized by* $\Delta\sigma$ *(*$\Delta\sigma = 2, 3, 4, 5, 6, 7$*, from left to right). The resulting shapes are shown in Figure 4.12.*

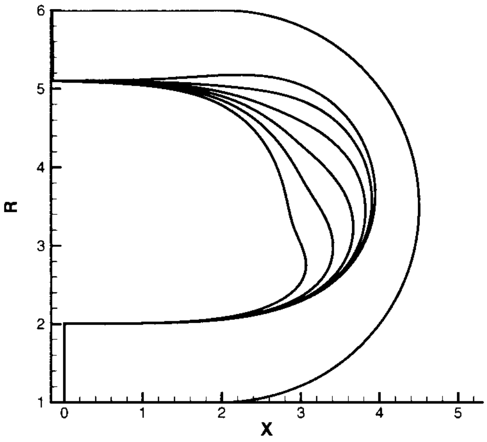

Figure 4.12: *Shapes of the elbow channels for different pressure gradients (*$\Delta\sigma = 2, 3, 4, 5, 6, 7$*, from left to right).*

Seite Leer /
Blank leaf

Chapter 5

Advanced Flow Models in Inverse Coordinates

The nature of turbomachinery flows is not as simple as the axis–symmetric flow model of chapters 2 and 3. Unsteady, viscous and fully three dimensional effects are present and strongly influence the behavior and the performance of the turbine. Anyway, an accurate simulation of all flow phenomena exceeds the needs and the resources of the designer, since in the design and optimization process the requirement in computing power increases by orders of magnitude with respect to plain simulation. The topic of this chapter is to present a more detailed flow model, which can handle some of these effects, but which retains the computational complexity, and the structure of a two dimensional method.

For the aerodynamic design only steady flows make sense, since it is not feasible to design the casing of a turbine which adapts itself to the underlying time dependent flow.

One of the main concerns of a designer is to reduce the losses of the turbine. These losses have many causes, such as frictional losses at the walls and losses due to the presence of shocks in a blade row. Further losses can be caused by secondary flows, leakage and wakes. The reliability of the predictions and the quality of the design is strongly affected by these phenomena, and therefore they need to be taken into account. Since a simulation resolving all the effects is not practicable, simplifications are made, for example the leakage effects are incorporated in the boundary layer. In this chapter we present two methods for handling the viscous losses, the distributed loss model and the boundary layer method. These two strategies are complementary, the distributed loss model can be used to simulate the losses due to a shock between the blades, while the boundary layer method is to be preferred for the evaluation of the boundary layer effects.

The interest in the reduction of the dimensionality of the problem comes from the fact that the full three dimensional character of the flow cannot be incorporated in the framework of the inverse Euler equations as seen in section 2.4. For an almost axis–symmetric flow an alternative, and in practice sufficiently accurate description is given by a quasi–three–dimensional (Q3D) model. The Q3D flow equations are widely used in the design of the components of a gas turbine. The Q3D flow model consists in a set of passage averaged equations with source terms to incorporate the 3D effects. These source terms are found either by a blade–to–blade computation or by measurements. This model is very attractive and widely used in the design process of diffusers since it gives accurate predictions at a low computational cost.

In this chapter we extend the range of applicability of the basic inverse design tool described in the chapters 2 and 3. In section 5.1 we extend the axis–symmetric inverse Euler equations (2.83) to incorporate the Q3D effects such as blade blockage and the deflection of the flow caused by the presence of blades. A novel inverse formulation of the Q3D equations is presented. In section 5.2 we describe the distributed loss model, which is the mechanism for considering viscous and shock–induced losses in the Q3D equations. A more detailed description of the boundary layer losses and development can be given by an integral boundary layer method. This is the topic of section 5.3.

5.1 Quasi–Three–Dimensional Flow Model

The quasi–three–dimensional (Q3D) flow model predicts the averaged flow in a device. It can be derived by averaging the flow equations in cylindrical coordinates over the angular variable θ. This results in a set of equations in the meridional plane or S_2–surface, where the deviation from the mean flow is taken into account by source terms which depend on the spatial fluctuations of the flow around its mean value. The spatial fluctuations express the three dimensional character of the flow and can be either found from a blade–to–blade computation in the tangential S_1–surface or derived from experiments or simply neglected. This model is widely used in the turbomachinery industry because the averaging reduces the number of independent variables by one, while the calibration of the source terms allows for very accurate prediction of the losses.

Since a fully three dimensional description of the flow is not possible in the framework of the inverse Euler equations (see section 2.4), we present a novel inverse formulation of the Q3D flow equations in the framework developed by Keller [36].

In the next sections, we summarize the three steps in the derivation of the inverse form of the Q3D equations. In section 5.1.1 we present the flow equations in the physical space, in section 5.1.2 we define the inverse coordinates while the inverse formulation is described in section 5.1.3. A numerical example follows in section 5.1.4.

5.1.1 Passage Averaged Equations

The passage averaged Euler equations are found by integrating the three dimensional Euler equations over the angular variable θ. The equations are then rewritten in terms of passage averaged and density weighted averaged variables. In this section we recall the basic definitions of the averaging procedure and present the passage averaged Euler equations.

In an (x, r, θ) cylindrical system of coordinates the three dimensional Euler equations are

$$\partial_x (r\rho u) + \partial_r (r\rho v) + \partial_\theta (\rho w) = 0 \tag{5.1a}$$

$$\partial_x \left(r\rho u^2\right) + \partial_r (r\rho u v) + \partial_\theta (\rho u w) = -r\partial_x p \tag{5.1b}$$

$$\partial_x (r\rho u v) + \partial_r \left(r\rho v^2\right) + \partial_\theta (\rho v w) - \rho w^2 = -r\partial_r p \tag{5.1c}$$

$$\partial_x (r\rho u w) + \partial_r (r\rho v w) + \partial_\theta \left(\rho w^2\right) + \rho v w = -\partial_\theta p \tag{5.1d}$$

$$u\partial_x H + v\partial_r H + w\partial_\theta H = 0. \tag{5.1e}$$

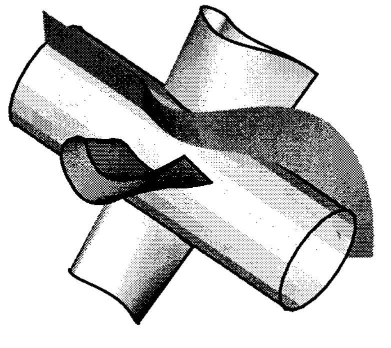

Figure 5.1: *Blade row and meridional surface on which the Q3D equations hold.*

These are the equations stating the conservation of mass (5.1a), of momentum (5.1b)–(5.1d) and of energy (5.1e). The unknowns are the axial, radial and azimuthal velocities u, v, w, the density ρ, the pressure p and the total enthalpy $H = c_pT + \frac{1}{2}(u^2 + v^2 + w^2)$ where T is the temperature and c_p is the heat coefficient at constant pressure. The equations are closed by the equation of state

$$p = \rho RT. \tag{5.2}$$

We consider a channel containing stationary blades such as those in Figure 5.1 or 5.2 and average (5.1) over θ from the pressure side $\theta_p(x, r)$ of the blade to the suction side $\theta_s(x, r)$. The integration is carried out on surfaces of revolution.

In order to write the passage averaged equations we need to introduce the passage average $\overline{A}$ of the quantity A with corresponding perturbation A' defined as

$$\overline{A} = \frac{1}{\theta_s - \theta_p} \int_{\theta_p}^{\theta_s} A d\theta \, , \, A = \overline{A} + A', \tag{5.3}$$

and the density weighted average $\widetilde{A}$

$$\widetilde{A} = \frac{\int_{\theta_p}^{\theta_s} \rho A d\theta}{\int_{\theta_p}^{\theta_s} \rho d\theta} \, , A = \widetilde{A} + A''. \tag{5.4}$$

The perturbations of a density weighted averaged quantity are denoted by A''.

The advantage of introducing two distinct averages is that in this way a passage averaged continuity equation without perturbation terms can be derived. In particular, the definition of a stream–function satisfying the passage averaged continuity equation is possible. A discussion on the various averaging strategies can be found in [32] and [31].

The passage averaging consists in integrating (5.1) over a surface of revolution from the pressure side to the suction side of the blade [35]. This results in the passage averaged equations

$$\partial_x \left(Br\overline{\rho}\widetilde{u} \right) + \partial_r \left(Br\overline{\rho}\widetilde{v} \right) = 0 \tag{5.5a}$$

$$\widetilde{u}\partial_x\widetilde{u} + \widetilde{v}\partial_r\widetilde{u} = -\frac{1}{\overline{\rho}}\overline{\partial_x p} + P_x \tag{5.5b}$$

$$\widetilde{u}\partial_x\widetilde{v} + \widetilde{v}\partial_r\widetilde{v} - \frac{\widetilde{w}^2}{r} = -\frac{1}{\overline{\rho}}\overline{\partial_r p} + P_r \tag{5.5c}$$

$$\widetilde{u}\partial_x \left(r\widetilde{w} \right) + \widetilde{v}\partial_r \left(r\widetilde{w} \right) = -\frac{1}{\overline{\rho}}\overline{r\partial_\theta p} + rP_\theta \tag{5.5d}$$

$$\widetilde{u}\partial_x\widetilde{H} + \widetilde{v}\partial_r\widetilde{H} = P_H \tag{5.5e}$$

The blade blockage term

$$B = \frac{N}{2\pi}\left(\theta_s - \theta_p\right) \tag{5.6}$$

is the non dimensionalized circumferential spacing between the blades. N is the number of blades. An example of the form of B, based on the blades depicted in Figure 5.1, can be seen in Figure 5.2. The averaging procedure produces source terms depending on the perturbations of the velocity field. Their explicit form is

$$P_x = -\frac{1}{Br\overline{\rho}}\left(\partial_x \left(Br\overline{\rho (u'')^2} \right) + \partial_r \left(Br\overline{\rho u'' v''} \right) \right) \tag{5.7a}$$

$$P_r = -\frac{1}{Br\overline{\rho}}\left(\partial_x\left(Br\overline{\rho u''v''}\right) + \partial_r\left(Br\overline{\rho(v'')^2}\right)\right) + \frac{\widetilde{(w'')^2}}{r} \tag{5.7b}$$

$$P_\theta = +\frac{1}{Br\overline{\rho}}\left(\partial_x\left(Br\overline{\rho u''w''}\right) + \partial_r\left(Br\overline{\rho v''w''}\right)\right) + \frac{\widetilde{v''w''}}{r}. \tag{5.7c}$$

$$P_H = -\frac{1}{Br\overline{\rho}}\left(\partial_x\left(Br\overline{\rho}\widetilde{H''u''}\right) + \partial_r\left(Br\overline{\rho}\widetilde{H''v''}\right)\right) \tag{5.7d}$$

The perturbation terms are usually computed from a blade–to–blade computation. Given their lower order effect and the difficulty of interpreting them in inverse coordinates, they are neglected in the subsequent algebra and in the numerical examples.

A closer look to (5.5) reveals that we did not work out the averaging of the pressure gradients terms such as

$$\overline{\partial_x p}. \tag{5.8}$$

In the framework of the inverse Euler equations, the pressure gradient is transformed by means of the first law of thermodynamics (2.17) to a combination of total enthalpy and entropy gradients. The algebra required for the derivation of Crocco's form of the passage averaged equations is more involved. Two possibilities to handle the pressure gradient term and to derive a Crocco's formulation are discussed.

Straightforward averaging of (5.8) as in [35] leads the following passage averaged momentum equations (the perturbation terms are omitted)

$$\widetilde{u}\partial_x\widetilde{u} + \widetilde{v}\partial_r\widetilde{u} = -\frac{1}{\overline{\rho}}\partial_x\overline{p} + F_{B.x} \tag{5.9a}$$

$$\widetilde{u}\partial_x\widetilde{v} + \widetilde{v}\partial_r\widetilde{v} - \frac{\widetilde{w}^2}{r} = -\frac{1}{\overline{\rho}}\partial_r\overline{p} + F_{B.r} \tag{5.9b}$$

$$\widetilde{u}\partial_x\left(r\widetilde{w}\right) + \widetilde{v}\partial_r\left(r\widetilde{w}\right) = rF_{B.\theta} \tag{5.9c}$$

where the blade forcing terms

$$F_{B.x} = \frac{N}{2\pi B\overline{\rho}}\left(p'_s\partial_x\theta_s - p'_p\partial_x\theta_p\right) \tag{5.10a}$$

$$F_{B.r} = \frac{N}{2\pi B\overline{\rho}}\left(p'_s\partial_r\theta_s - p'_p\partial_r\theta_p\right) \tag{5.10b}$$

$$F_{B,\theta} = \frac{N}{2\pi r B\overline{\rho}}\left(p'_p - p'_s\right), \tag{5.10c}$$

have been introduced. The blade forces are responsible for the deflection of the flow by changing the angular momentum $\widetilde{C} = r\widetilde{w}$.

This formulation of the Q3D equations can be rewritten in Crocco 's form by introducing an averaged first law of thermodynamics

$$\widetilde{T}d\hat{s} = c_v d\widetilde{T} + \frac{\overline{p}}{\overline{\rho}^2}d\overline{\rho}, \tag{5.11}$$

or equivalently by defining the entropy of the averaged flow by

$$\hat{s} = c_v \log\left(\frac{\overline{p}}{\overline{\rho}^\gamma}\right). \tag{5.12}$$

Perturbation terms occur in Crocco's form of the flow equations, since the entropy of the averaged flow $\hat{s}$ in (5.11) and (5.12) does not coincide with the density weighted average of the entropy $\widetilde{s}$. In particular $\hat{s}$ is not constant anymore along the averaged streamlines. Under the assumption that these perturbation terms can be neglected, Crocco's form of the momentum equations for the averaged flow (5.16) follows by the algebra presented in section 2.2

A more formal derivation [32] of the Crocco form of momentum conservation for the averaged flow can be found by the substitution

$$\nabla p = \rho\left(c_p\nabla T - T\nabla s\right), \tag{5.13}$$

where exceptionally the gradient is three dimensional, $\nabla = (\partial_x, \partial_r, \partial_\theta)^T$, and then applying the averaging procedure. It follows

$$\begin{aligned}\frac{1}{\overline{\rho}}\overline{\partial_x p} = {} & \left(c_p\partial_x\widetilde{T} - \widetilde{T}\partial_x\widetilde{s}\right) \\ & - \frac{N}{2\pi B\overline{\rho}}c_p\left(\rho_s T''_s\partial_x\theta_s - \rho_p T''_p\partial_x\theta_p\right) + \\ & + \frac{N}{2\pi B\overline{\rho}}\left(\rho_s\widetilde{T}_s s''_s\partial_x\theta_s - \rho_p\widetilde{T}_p s''_p\partial_x\theta_p\right) \\ & + Q_x\end{aligned} \tag{5.14}$$

where $\Delta\theta = \theta_s - \theta_p$ and Q_x is the perturbation term, which will be neglected. For the radial component a similar formula, with ∂_r instead of ∂_x, holds, while in the tangential direction we get

$$\frac{1}{\overline{\rho}}\overline{\partial_\theta p} = \frac{N}{2\pi B\overline{\rho}}\left(p'_s - p'_p\right). \tag{5.15}$$

Inserting (5.14) and (5.15)into (5.5) yields Crocco's form of the passage averaged equations

$$\nabla \widetilde{H} - \widetilde{T} \nabla \widetilde{s} - \frac{\widetilde{C}}{r^2} \nabla \widetilde{C} = \widetilde{\mathbf{u}}^{\perp} \widetilde{\omega} + \mathbf{F}, \tag{5.16}$$

Under the simplifying assumption that all perturbation terms can be neglected, and thus just keeping the blade forces arising from entropy and temperature differences between pressure and suction side, the forcing term $\mathbf{F} = \mathbf{F}_{\text{Blades}} = (F_{B.x}, F_{B.r})^T$ is found to be

$$F_{B.x} = \frac{N}{2\pi B \overline{\rho}} \left([\rho c_p T'' \partial_x \theta]_p^s - [\rho \widetilde{T} s'' \partial_x \theta]_p^s \right) \tag{5.17a}$$

$$F_{B.r} = \frac{N}{2\pi B \overline{\rho}} \left([\rho c_p T'' \partial_r \theta]_p^s - [\rho \widetilde{T} s'' \partial_r \theta]_p^s \right) \tag{5.17b}$$

$$F_{B.\theta} = \frac{N}{2\pi r B \overline{\rho}} \left(p'_p - p'_s \right), \tag{5.17c}$$

where $[A]_p^s = A_s - A_p$ denotes the jump across the blade.

The choice of neglecting the perturbation terms reflects the difficulty of prescribing the perturbations terms $\mathbf{Q} = \cdot (Q_x, Q_r, Q_\theta)^T$ in inverse coordinates and the fact that they induce lower order effects.

The two derivations of Crocco's formulation of the passage averaged equations lead formally to the same result (5.16), the difference consisting in the explicit form of the blade forces. The blade forces (5.10) are used in the industry, while (5.17) is to be preferred in the framework of the inverse Euler equations since it is directly related to the entropy and total enthalpy.

We consider now the energy equation (5.5e) and notice that the total enthalpy of the averaged flow, defined as

$$\hat{H} = c_p \widetilde{T} + \frac{1}{2} \left((\widetilde{u})^2 + (\widetilde{v})^2 + (\widetilde{w})^2 \right) \tag{5.18}$$

does not coincide with the average of the total enthalpy $\widetilde{H}$ appearing in (5.5e). The exact relation between $\widetilde{H}$ and $\hat{H}$ is

$$\widetilde{H} = \hat{H} + \frac{1}{2} \left(\widetilde{(u'')^2} + \widetilde{(v'')^2} + \widetilde{(v'')^2} \right). \tag{5.19}$$

These perturbation terms are neglected in the energy and Crocco's equation.

A transport equations for the entropy of the mean flow can be derived by the averaging procedure used above and then neglecting the perturbation terms. The entropy $\hat{s}$ is seen to satisfy

$$\widetilde{u}\partial_x\hat{s} + \widetilde{v}\partial_r\hat{s} = 0. \tag{5.20}$$

Again we substitute to the averaged entropy $\widetilde{s}$ the entropy of the averaged flow $\hat{s}$. The corresponding perturbation terms in Crocco's equation are discarded.

In the framework of the inverse Euler equation Crocco's equation (5.16) is dotted by the velocity in the meridional plane $\widetilde{\mathbf{u}} = (\widetilde{u}, \widetilde{v})^T$ and by $\widetilde{\mathbf{u}}^{\perp} = (v, -u)^T$ to get the final formulation of the passage averaged equations

$$\partial_x\left(Br\overline{\rho}\widetilde{u}\right) + \partial_r\left(Br\overline{\rho}\widetilde{v}\right) = 0 \tag{5.21a}$$

$$\widetilde{\mathbf{u}}^{\perp} \cdot \nabla\hat{H} - \widetilde{T}\widetilde{\mathbf{u}}^{\perp} \cdot \nabla\hat{s} - \frac{\widetilde{C}}{r^2}\widetilde{\mathbf{u}}^{\perp} \cdot \nabla\widetilde{C} = \|\widetilde{\mathbf{u}}\|^2\widetilde{\omega} + \widetilde{\mathbf{u}}^{\perp} \cdot \mathbf{F} \tag{5.21b}$$

$$\widetilde{u}\partial_x\widetilde{C} + \widetilde{v}\partial_r\widetilde{C} = rF_{B.\theta} \tag{5.21c}$$

$$\widetilde{u}\partial_x\hat{H} + \widetilde{v}\partial_r\hat{H} = \widetilde{\mathbf{u}} \cdot \mathbf{F} + \frac{\widetilde{C}}{r^2}\widetilde{\mathbf{u}} \cdot \nabla\widetilde{C} \tag{5.21d}$$

$$\widetilde{u}\partial_x\hat{s} + \widetilde{v}\partial_r\hat{s} = 0, \tag{5.21e}$$

where $\widetilde{\omega} = \partial_x\widetilde{v} - \partial_r\widetilde{u}$.

The right hand side of (5.21d) can be rewritten with the help of (5.21c) as

$$\widetilde{\mathbf{u}} \cdot \mathbf{F} + \frac{\widetilde{C}}{r^2}\widetilde{\mathbf{u}} \cdot \nabla\widetilde{C} = \widetilde{u}F_{B.x} + \widetilde{v}F_{B.r} + \widetilde{w}F_{B.\theta} = 0 \tag{5.22}$$

since the blade forces are perpendicular to the velocity vector.

To summarize, the passage averaged equations are derived by assuming that the integration is done on surfaces of revolution, by neglecting all perturbation terms and by introducing the entropy and the total enthalpy of the mean flow. The resulting set of equations is very similar to (2.83) and a framework for their transformation in an inverse form exists, see chapter 2.

5.1.2 Inverse Coordinates and Transformation Rules

The Q3D flow equations are similar to the axis–symmetric equations (2.83); the difference consists in additional source terms. For the derivation of the

Q3D inverse Euler equations we follow the steps outlined in chapter 2. Most of the algebra is skipped and only the differences outlined.

The definition of the stream–function needs to be modified to take into account the blockage term B. In order to automatically satisfy the equation of mass conservation (5.21a) we define

$$\nabla^{\perp}\psi = B r \overline{\rho} \widetilde{\mathbf{u}} \tag{5.23}$$

where $\nabla^{\perp} = (\partial_r, -\partial_x)^T$. We notice that the stream–function ψ defined in (5.23) is constant along the walls of the channel, independently of their shape even in the case of $B \neq 1$. We define the natural coordinate as being the coordinate along the streamlines of the averaged flow

$$h\nabla\sigma = \frac{\widetilde{\mathbf{u}}}{\|\widetilde{\mathbf{u}}\|}. \tag{5.24}$$

The inverse coordinates (σ, ψ) enjoy the same properties as in the planar and axis–symmetric case. In particular the grid generated by the mapping

$$x = x(\sigma, \psi) \text{ and } r = r(\sigma, \psi) \tag{5.25}$$

is orthogonal and streamline aligned. The algebra needed to derive the Q3D inverse Euler equations makes use of the transformation rules

$$\widetilde{\mathbf{u}} \cdot \nabla = \frac{1}{\overline{\rho} r B J} \partial_\sigma \tag{5.26a}$$

$$\widetilde{\mathbf{u}}^{\perp} \cdot \nabla = -\frac{h^2}{\overline{\rho} r B J^2} \partial_\psi \tag{5.26b}$$

and of the meridional specific massflow–geometry equation

$$m^2 := \|\overline{\rho}\widetilde{\mathbf{u}}\|^2 = \left(\frac{h}{rBJ}\right)^2. \tag{5.27}$$

The determinant of the Jacobi matrix of the transformation assumes the form

$$J = x_\sigma r_\psi - x_\psi r_\sigma. \tag{5.28}$$

5.1.3 Q3D Inverse Euler Equations

The formulas of section 5.1.2 allow us to state the inverse Q3D equations. The mass conservation is dropped and we get

$$\partial_\psi \hat{H} - \widetilde{T}\partial_\psi \hat{s} - \frac{\widetilde{C}}{r^2}\partial_\psi \widetilde{C} = \frac{1}{\overline{\rho} r B J}\partial_\psi \left(\frac{h^2}{\overline{\rho} r B J}\right) - \frac{J}{h^2}\mathbf{F}\cdot\mathbf{x}_\sigma^\perp \tag{5.29a}$$

$$\partial_\sigma \widetilde{C} = \overline{\rho} r^2 B J F_{B.\theta} \tag{5.29b}$$

$$\partial_\sigma \hat{H} = 0 \tag{5.29c}$$

$$\partial_\sigma \hat{s} = 0 \tag{5.29d}$$

It is remarkable that the equations for the geometry do not change

$$\partial_\sigma \left(\frac{J}{h^2}\partial_\sigma \mathbf{x}\right) + \partial_\psi \left(\frac{h^2}{J}\partial_\psi \mathbf{x}\right) = 0. \tag{5.30}$$

The system of partial differential equations (5.29)–(5.30) is called quasi–three–dimensional inverse Euler equations.

The structure of the Q3D inverse Euler equations is very similar to the axis–symmetric or 2D equations. In particular only the inverse form of Crocco's theorem is modified in order to take into account the blockage and the defection of the flow arising from the presence of blades. Surprisingly enough these effects can be handled with simple changes to the basic equations. In the same way as in a normal CFD code, where for Q3D effects the core solver must not be changed, in the inverse formulation, the "core" consisting in the scaled Laplace equations (5.30) does not change.

5.1.4 Numerical Example

The Q3D flow model introduces new effects which influence the shape of the diffuser. We consider a diffuser with inlet Mach number 0.4 whose lower wall is kept fixed at $r = 1$. Along the upper wall we impose the specific massflow distribution

$$m(\sigma) = \begin{cases} 1 & \text{for } \sigma \le 2 \\ \frac{1}{2} & \text{for } \sigma > 4 \\ \frac{3}{4} + \frac{1}{4}\cos(\frac{\pi}{2}(\sigma - 2)) & \text{elsewhere} \end{cases} . \tag{5.31}$$

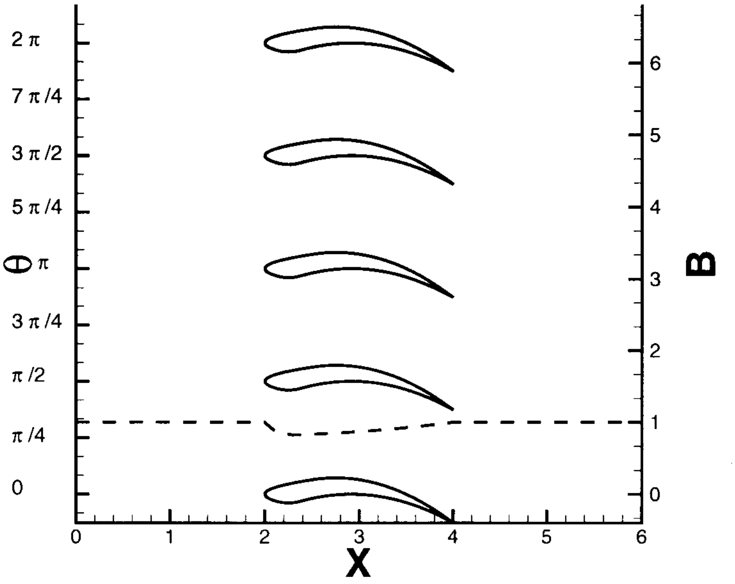

Figure 5.2: *Tangential view of the blade row and corresponding blockage function (dashed line). A three dimensional view of the configuration is given in Figure 5.1.*

In the region $2 \leq \sigma \leq 4$ we introduce four blades, which are represented in Figure 5.2. The corresponding blockage function is shown in Figure 5.2 too.

A comparison of the shape of the diffuser with and without blades is shown in Figure 5.3. Figure 5.3 allows to appreciate the strong dependence of the shape of the diffuser on the continuity equation. In the presence of blades, the distance in circumferential direction between the walls of the channel decreases. Therefore, in order to fulfill the continuity equation the channel designed with the Q3D model must be higher than the corresponding axis–symmetric duct. This effect is clearly recognizable in Figure 5.3.

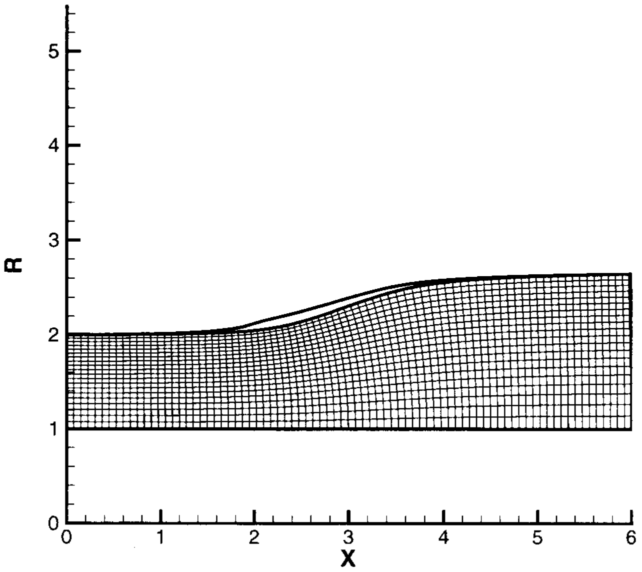

Figure 5.3: *Comparison of the shapes of a diffuser with and without blades. The upper line is the Q3D design, which takes into account the restriction of the channel in the region with blades.*

5.2 Distributed Loss Model

Our flow model can handle Q3D inviscid flows, but there is still no model for the losses occurring in the channel. A typical and widespread model for the losses in internal and channel flows, particularly in turbomachinery applied in conjunction with the averaged Q3D flow model, is given by a so called *distributed loss model*, which consists in assuming that the effects of the shear stresses are equivalent to the action of a distributed force. This force is mainly constructed from empirical data; correlations are available for many typical flows in turbomachinery. The motivation for the use of a distributed loss model comes from the fact that a flow in a turbine is so complicated that the simulation of all the effects (three dimensionality, unsteadiness, influence of the viscosity, leakage, etc.) is prohibitively costly and certainly not suited for the design stages where many configurations must be evaluated in a short time. In our Q3D model of the flow the losses are considered on a global basis. We could for example prescribe a drop, between inlet and outlet of the total pressure by 10 % without taking into account the exact mechanism and the location of these losses. At the Q3D level of approximation, there is therefore no point in considering the exact structure of the losses and so the losses can be modeled by a forcing term. Our flow model remains basically inviscid, but is not isentropic anymore, since the entropy variation along a streamline will be connected to the energy dissipation.

5.2.1 Flow Model

We present in this section the main idea of the distributed loss model by means of the two dimensional Euler equations, since this simplifies the notation. In Appendix A we present the complete distributed loss model for the Q3D approximation.

We consider Crocco's form of the momentum equations for a viscous flow [30]

$$\nabla H - T\nabla s = \mathbf{u}^{\perp}\omega + \frac{1}{\rho}\mathrm{div}(\tau) \tag{5.32}$$

where τ is the shear stress and the divergence acts on the rows of the tensor. The corresponding energy equation with heat diffusion and kinetic energy losses reads

$$\mathrm{div}(\rho\mathbf{u}H) = \mathrm{div}(k\nabla T) + \mathrm{div}(\tau\cdot\mathbf{u}), \tag{5.33}$$

where k is the thermal conductivity. By dotting (5.32) with $\mathbf{u}$ we derive the equation for transport and production of the entropy

$$T\rho\mathbf{u}\cdot\nabla s = \nabla\cdot(k\nabla T) + (\tau\cdot\nabla)\cdot\mathbf{u}. \tag{5.34}$$

The distributed loss model is based on the following two assumptions

- the heat diffusion compensates the kinetic energy losses

  $$\mathrm{div}(k\nabla T) + \mathrm{div}(\tau\cdot\mathbf{u}) = 0 \tag{5.35}$$

  This relation holds for flows at Prandl number of one. It is not valid in complex flows such as wakes but it remains still a useful approximation.

A more severe assumption would be to neglect the heat diffusion and to assume that

$$\operatorname{div}(\tau \cdot \mathbf{u}) = 0. \tag{5.36}$$

This assumption does not hold in regions near the walls. We describe in section 5.3, a boundary layer method that can be coupled to the distributed loss model in order to take into account the viscous losses near the walls.

- The shear stresses can be modeled by a distributed force $\mathbf{F}_\tau$ responsible of the overall entropy increase in the flow

  $$\operatorname{div}\tau = \rho \mathbf{F}_\tau. \tag{5.37}$$

  $\mathbf{F}_\tau$ is the only responsible of the entropy production.

Under these assumptions, the entropy equation (5.34) is modified to

$$T\mathbf{u} \cdot \nabla s = -\mathbf{u} \cdot \mathbf{F}_\tau. \tag{5.38}$$

We further simplify (5.38) by assuming that the distributed force acts only in the direction opposite to the local velocity vector $\mathbf{u}$. Thus there exists a positive scalar function F_τ such that

$$\mathbf{F}_\tau = -\frac{\mathbf{u}}{\|\mathbf{u}\|} F_\tau. \tag{5.39}$$

The final form of the entropy-production equation is found by inserting (5.39) into (5.38)

$$T\mathbf{u} \cdot \nabla s = \|\mathbf{u}\| F_\tau. \tag{5.40}$$

The complete flow equations under assumptions of the distributed loss model are

$$\partial_x(\rho u) + \partial_y(\rho v) = 0 \tag{5.41a}$$

$$\mathbf{u}^\perp \omega = \nabla H - T\nabla s + \frac{\mathbf{u}}{\|\mathbf{u}\|} F_\tau \tag{5.41b}$$

$$\mathbf{u} \cdot \nabla H = 0 \tag{5.41c}$$

$$T\mathbf{u} \cdot \nabla s = \|\mathbf{u}\| F_\tau. \tag{5.41d}$$

The flow equations under the the distributed loss model (5.41) are similar to the flow equations considered in section 2.2. The difference consists in the occurrence of an external force, responsible for the entropy generation, in Crocco's equations and in the entropy equation.

The inverse formulation of (5.41) is again found by taking $\mathbf{u}^{\perp} \cdot$ (5.41*b*) and applying the transformations of section 2.5. Since

$$\mathbf{u}^{\perp} \cdot \frac{\mathbf{u}}{\|\mathbf{u}\|} F_{\tau} = 0 \tag{5.42}$$

the inverse form of Crocco's equations, with and without the distributed loss model, coincide.

The inverse form of the entropy production equation is

$$T \partial_{\sigma} s = h F_{\tau}. \tag{5.43}$$

Since F_{τ} is positive , the entropy increases along the streamlines. In the case of a non–uniform friction force F_{τ} an irrotational flow at inlet becomes rotational because of the different levels of entropy on each streamline, see figure 5.5.

The inverse Euler equations with the distributed loss model have the same boundary conditions as the inverse Euler equations without the loss model, since only an external force has been added.

5.2.2 Distributed Forces

The distributed loss model is not closed until we specify the form of the frictional force F_{τ}. The accuracy of the results depends on how much experimental information is used for the derivation of the forcing term. There are different models for F_{τ}, such as the specified pressure loss method, the specified skin friction coefficient approach and the eddy viscosity model [18].

We describe a simple loss model which allows to take into account the losses generated in a shock and in a less accurate way by the boundary layers. For the latter an integral method is preferred, see section 5.3. The idea is to impose a static pressure loss along the streamlines. The static pressure is assumed to vary linearly with the meridional distance from the inlet. In order to have a linearly varying static pressure, F_{τ} must depend only on ψ and not on σ.

Further, in order to simulate the boundary layer effects, i.e. concentrate the losses in the near-wall regions, we use a loss term of the form

$$F_{\tau} = \Delta p_{\mathrm{mid}} + (\Delta p_{\mathrm{wall}} - \Delta p_{\mathrm{mid}}) |1 - \frac{2\psi}{\psi_{max}}|^{n}, \tag{5.44}$$

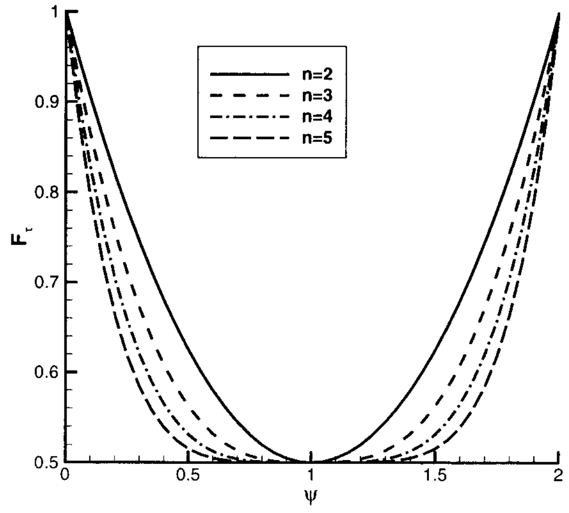

Figure 5.4: *Distributed forces are used to simulate viscous effects. For growing n, the losses are concentrated near the walls, in a way similar to a boundary layer.*

where Δp_{mid} and Δp_{wall} are the static pressure drop in the middle of the channel, respectively along the walls. If $\Delta p_{\text{wall}} > \Delta p_{\text{mid}}$ an increasing value of n concentrates the losses near the walls, as seen in figure 5.4.

5.2.3 Numerical Example

We consider again the exhaust diffuser of section 5.1.4. The viscous losses are modeled by a distributed force of the form

$$F_\tau = \Delta p_{\text{mid}} + (\Delta p_{\text{wall}} - \Delta p_{\text{mid}})|1 - \frac{2\psi}{\psi_{max}}|^4, \tag{5.45}$$

with the the numerical values

$$\Delta p_{\text{mid}} = 0.08\ ,\ \Delta p_{\text{wall}} = 0.15. \tag{5.46}$$

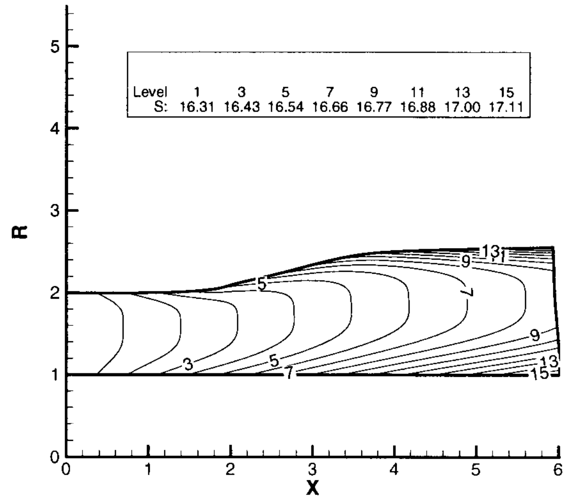

Figure 5.5: *Entropy contours in the exhaust diffuser. An increase of 5% of the entropy has been reached by adding a frictional force to the entropy transport equation (5.43).*

The outlet specific massflow distribution has been modified to

$$m_{\text{outlet}}(\psi) = 0.83(\psi - 0.85)^2 + 0.58 \tag{5.47}$$

to take into account that the outlet, atmospheric, pressure is constant.

The resulting geometry with the entropy contours is given in figure 5.5. The computation has been carried out on a 160 × 40 grid. The concentration of the viscous losses near the walls is clearly recognizable. At first it might be strange that the outlet wall is not vertical. Because of the orthogonality of the grid, we are not allowed to prescribe the position and shape of the outlet surface. It automatically adjusts in order to combine the requirements of a prescribed specific massflow and of the orthogonality of the grid.

5.3 Boundary Layer Computations

Inviscid flow models are important and useful in the first stages of the design process, where big changes in the geometry are expected. The ultimative goal of the design, i.e. the reduction of the losses and an increased performance of the device, cannot be reached nor assessed without considering the viscous dissipation. In many cases a complete description of the flow given by the solution of the full Navier–Stokes equation is not necessary. When the recirculation regions are small or absent, simpler and computationally less expensive methods can be applied. These methods consists in a zonal approach, i.e. the domain is divided into an inner and an outer region. In the outer region, the viscous effects are neglected and the inviscid Euler equations used. In the inner, boundary layer region, a simplified version of the Navier–Stokes equations that still retains some viscous terms, is used.

The boundary layer method adds the following properties to our inverse design tool.

fictitious walls: because of the no–slip boundary condition, there is less massflow passing through the boundary layer for the real viscous profile than there would if an inviscid profile existed. By considering a modified, "displaced" inviscid wall, the same massflow near the wall as in the real viscous case can be generated. From the definition of the displacement thickness δ^*

$$\delta^* = \int_0^\delta \left(1 - \frac{\rho u}{\rho_e u_e}\right) d\eta, \tag{5.48}$$

where the subscript $_e$ denotes the variables on the boundary layer edge, follows that the two shaded patches in figure 5.6 have the same area. Therefore an inviscid profile

$$\rho u = \begin{cases} 0 & \text{for } \eta \leq \delta^* \\ \rho_e u_e & \text{for } \eta > \delta^* \end{cases} \tag{5.49}$$

has the same massflow near the wall as the viscous profile. δ^* is the distance, viscous effects appear to displace the inviscid flow from the wall. When the displacement thickness becomes large, the effective shape of the walls is quite different from the physical shape of the walls and the flow pattern must be recomputed. In the case of the inverse design we assume that the prescribed specific massflow distribution determines the shape of the inviscid domain, and that the displacement induced by the

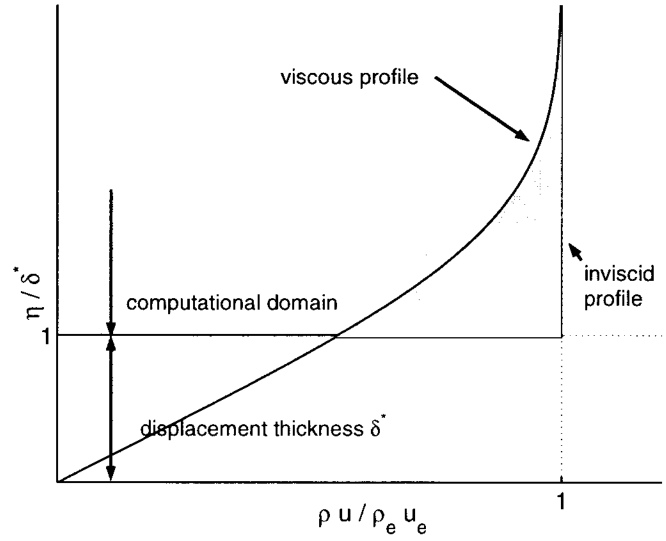

Figure 5.6: *Schematic view of the definition of the displacement thickness of a boundary layer. The two shaded areas are the same. The actual inviscid computational domain is displaced by a distance δ^* in the direction perpendicular to the wall denoted by the coordinate η.*

boundary layer can be added to it in a postprocessing step, see section 5.3.2. It is a very favorable property of the inverse design method that the inviscid computation and the boundary layer computation are totally decoupled. In the case of a fixed wall an implicit treatment is needed. The details can be found in section 5.3.2.

control of separation: we motivated the target–pressure–problem by the attempt to avoid a separating flow, but we never discussed how such pressure distributions can be obtained. The boundary layer method can answer this question as the ordinary differential equations of the boundary layer method become singular at a stagnation point, and hence at the beginning of the recirculation zone.

evaluation of the diffuser: after a new design is proposed, it must be evaluated and in particular the losses must be predicted. A boundary layer method gives, for non separated flow, accurate predictions of the viscous losses at a reduced computational cost.

Neither the solution of the full Navier–Stokes equations nor of Prandl's boundary layer equations is necessary to address these three points. We present in the next section a two equation integral method for boundary layers. The next section is a mere collection of formulas and can be skipped by the reader interested more in the ideas than in the details. In this case, the main point is that the integral method is just a black–box for finding the displacement thickness and hence for finding the effective inviscid walls.

5.3.1 Integral Method

The global features of the boundary layer, such as the displacement thickness or the viscous losses, can be described in an economical way by an integral method.

An integral method is derived by integrating Prandl's boundary layer equations over the boundary layer [49] [48]. Following [19] we use a two equation integral method which consists in the integral momentum equation

$$\frac{d\theta}{d\sigma} + (2 + H - M_e^2)\frac{\theta}{u_e}\frac{du_e}{d\sigma} = \frac{C_f}{2} \tag{5.50}$$

and in the kinetic shape parameter equation

$$\theta\frac{dH^*}{d\sigma} + \left(2H^{**} + H^*(1 - H)\right)\frac{\theta}{u_e}\frac{du_e}{d\sigma} = 2C_D - H^*\frac{C_f}{2}. \tag{5.51}$$

The coordinate σ is a coordinate aligned with the wall. The definitions of the symbols arising in (5.50) and (5.51) are shown in table 5.1. In the boundary layer method the flow variables at the edge of the boundary layer ρ_e, M_e, u_e are assumed to be known from the computation of the inviscid flow field. In particular the velocity gradient $\frac{du_e}{d\sigma}$ drives the development of the boundary layer and the generation of separation. Equations (5.50) and (5.51) are solved for θ and H^*.

Many of the variable introduced in (5.50) and (5.51) and in table 5.1 depend on the density and on the velocity field inside the boundary layer. It is therefore necessary to assume that the velocity has a special form, such as the Falken–Skan profile [49]. Closure relations and correlations for a Falken-Skan profile can be expressed as

$$H_k = \frac{H - 0.290M_e^2}{1 + 0.113M_e^2} \tag{5.52a}$$

ρ_e		density on the boundary layer edge
u_e		velocity on the boundary layer edge
M_e		Mach number on the boundary layer edge
τ_{wall}		wall shear stress
μ_e		viscosity on the boundary layer edge
Re_θ	$= \frac{\rho_e u_e \theta}{\mu_e}$	momentum thickness Reynolds number
δ^*	$= \int_0^\delta \left(1 - \frac{\rho u}{\rho_e u_e}\right) d\eta$	displacement thickness
θ	$= \int \frac{\rho u}{\rho_e u_e} \left(1 - \frac{u}{u_e}\right) d\eta$	momentum thickness
δ^{**}	$= \int \frac{u}{u_e} \left(1 - \frac{\rho}{\rho_e}\right) d\eta$	density thickness
θ^*	$= \int \frac{\rho u}{\rho_e u_e} \left(1 - \frac{u^2}{u_e^2}\right) d\eta$	kinetic energy thickness
H	$= \frac{\delta^*}{\theta}$	shape parameter
H^*	$= \frac{\theta^*}{\theta}$	kinetic shape parameter
H^{**}	$= \frac{\delta^{**}}{\theta}$	density shape parameter
C_f	$= 2\frac{\tau_{wall}}{\rho_e u_e^2}$	skin–friction coefficient
C_D	$= \frac{1}{\rho_e u_e^3} \int \tau \frac{\partial u}{\partial \eta} d\eta$	dissipation coefficient

Table 5.1: *Quantities arising in the two equation integral method (5.50) and (5.51).*

$$H^* = \begin{cases} 1.515 + 0.076\frac{(4-H_k)^2}{H_k} & \text{for } H_k < 4 \\ 1.515 + 0.040\frac{(4-H_k)^2}{H_k} & \text{for } H_k > 4 \end{cases} \tag{5.52b}$$

$$Re_\theta \frac{C_f}{2} = \begin{cases} -0.067 + 0.01977\frac{7.4-H_k}{H_k-1} & \text{for } H_k < 7.4 \\ -0.067 + 0.022\left(1 - \frac{1.4}{H_k-6}\right)^2 & \text{for } H_k > 7.4 \end{cases} \tag{5.52c}$$

$$Re_\theta \frac{2C_D}{H^*} = \begin{cases} 0.207 + 0.00205(4 - H_k)^{5.5} & \text{for } H_k < 4 \\ 0.207 - 0.003\frac{(H_k-4)^2}{1+0.02(H_k-4)^2} & \text{for } H_k > 4 \end{cases} \tag{5.52d}$$

$$H^{**} = \left(\frac{0.064}{H_k - 0.8} + 0.251\right) M_e^2. \tag{5.52e}$$

The closure relations (5.52) hold for laminar flows. The corresponding relations for turbulent flows can be found in [19]. H_k is the kinetic shape parameter defined with the density taken constant across the boundary layer. Therefore (5.52a) represents the correction needed to handle compressible flows.

At $H_k = 4$ the relation between H^* and H_k becomes singular. In this case separation occurs and the equations cannot be integrated further.

5.3.2 Viscous–Inviscid Coupling

In this section we address the problem of coupling the inviscid flow field computed by the inverse method of chapters 2 and 3 with the fictitious inviscid walls found by the boundary layer method of section 5.3.1.

In the literature, two ways of coupling the viscous and the inviscid flow model have been devised.

direct coupling: In the direct coupling the inviscid flow equations are solved and the pressure at the wall is used as input quantity for computing the boundary layer development. From the solution of the boundary layer equations the displacement effects (see figure 5.6) can be computed and the effective shape of the domain (for the inviscid core of the flow) can be found. Since the effective shape does not coincide with the original, the procedure is repeated until convergence is reached. The iterative procedure is known to converge slowly [60]. A full coupling of the two set of equations is more effective [63] and is simple to implement in our framework. We present the details of this approach later in this section.

inverse coupling: In the inverse boundary layer method the thickness of the boundary layer is specified and the velocity distribution along the boundary layer edge is found by solving the boundary layer equations (5.50) and (5.51). An inviscid inverse design tool can be applied to find the shape of the inviscid core. The overall shape is found by adding the given displacement to the inviscid shape. In this case the boundary layer computation is totally decoupled (!) from the inviscid computation, therefore only a preprocessing step is needed to find the correct velocity distribution. For this reason this method has not been explicitly implemented.

In the specific case of our inverse flow solver, the viscous–inviscid coupling is treated in two different ways according to the type of boundary condition imposed.

inverse wall: the thickness displacement δ, computed by the integral method, is added to the shape resulting from the inverse computation

$$\mathbf{x}_{\text{viscous wall}} = \mathbf{x}_{\text{inviscid}} + \delta \frac{\mathbf{x}_{\psi}}{\|\mathbf{x}_{\psi}\|}, \tag{5.53}$$

where $\frac{\mathbf{x}_\psi}{\|\mathbf{x}_\psi\|}$ is the normal to the wall. As already pointed out in this case the two flow models are independent. The coupling consists in the pressure distribution (or in the flow speed) along the edge of the boundary layer. Given this quantity, the two problems can be solved independently.

fixed wall: the pressure distribution along the wall is in this case *a–priori* not known, and depends on the whole flow field. The streamline along the edge of the boundary layer is found by adding the displacement thickness to the shape of the wall. Therefore it holds

$$f(\mathbf{x}_{\text{first streamline}} - \delta \frac{\mathbf{x}_\psi}{\|\mathbf{x}_\psi\|}) = 0, \tag{5.54}$$

where $f(x, y)$ is the algebraic equation defining the shape of the wall.

The integral momentum equation (5.50) and the kinetic shape parameter equation (5.51) are discretized by the trapezoidal rule, which is A–stable and hence does not have any problems with the stiffness of the integral equations (5.50) and (5.51) for high Reynolds numbers.

In order to avoid the convergence problems related with an iterative procedure continuously switching between flow solver and boundary layer method, we solve the equations for the core flow and for the boundary layer in a coupled way with the Newton solver described in section 3.2.

5.3.3 Numerical Example

We demonstrate the influence of a thick inlet boundary layer on the shape of a planar diffuser. We consider a Mach number 0.2 flow at a Reynolds number of $Re = 9.7 \cdot 10^5$. The inlet boundary layer thickness is about 10% of the channel height. This is reasonable for an exhaust diffuser in a gas turbine, when the actual boundary layer and the leakage effect are taken into account. The lower wall is kept fixed by the boundary condition $y = 0$, while on the upper wall we prescribe the specific massflow distribution

$$m(\sigma) = \begin{cases} 1, & \text{for } \sigma \leq 2 \\ \frac{1}{2} \frac{1}{1+(\sigma-2)^2/10} & \text{otherwise} \end{cases}. \tag{5.55}$$

The specific massflow distribution is plotted in figure 5.7, while in figure 5.8 we compare the inviscid solution (lower picture) and the solution from the viscous–inviscid method.

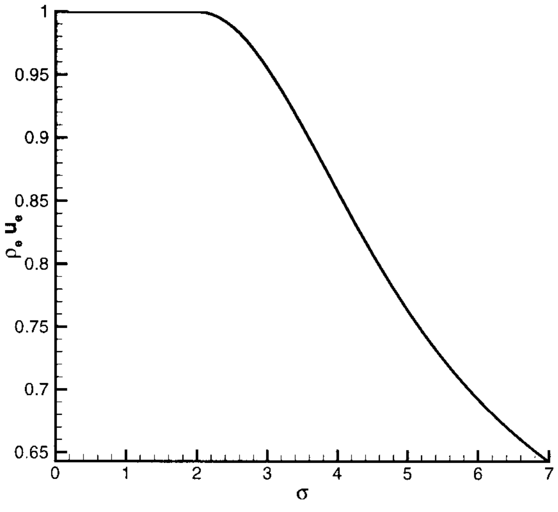

Figure 5.7: *Specific massflow distribution along the boundary layer edge on the upper wall of the diffuser. The resulting geometry is shown in figure 5.8.*

5.4 Concluding Remarks and Outlook

In this chapter we presented some flow models in inverse coordinates to overcome, at least in part, the impossibility to have a fully three dimensional design tool based on the inverse coordinates (σ, ψ, η).

The quasi–three–dimensional equations usually used for the simulation and design in industry have been rewritten in an inverse form and applied to the design of a diffuser with stationary blades. The most evident changes imposed by the quasi–three–dimensional equations regard the continuity equations. Changes in the width of the channel can be taken into account in the

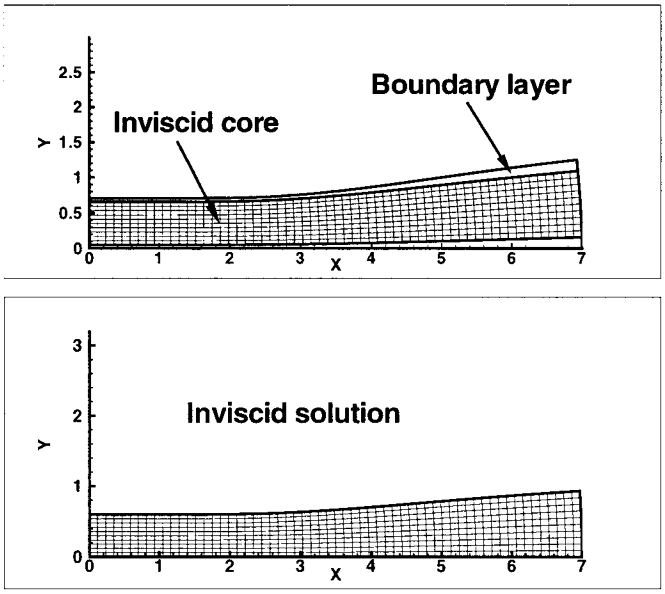

Figure 5.8: *Two dimensional laminar boundary layer in a diffuser. The specific massflow distribution along the edge of the boundary layer of the upper wall is plotted in figure 5.7.*

shape of the diffuser. The deflection of the flow due to the blades is with less influence on the shape, but still important for the accurate prediction of the flow. The deflection manifests itself in a change of the angular momentum. The influence of the perturbation terms is neglected because of the difficulties in prescribing them in inverse coordinates: one does not know a–priori how to prescribe the perturbation in function of σ and ψ.

The inverse formulation of the quasi–three dimensional flow equations extends all previously available single–pass methods based on natural (or inverse or intrinsic) system of coordinates.

Since the viscosity plays an important role, its effects are introduced in our inverse equations by means of a distributed force, responsible for the entropy production and by means of a boundary layer method. These two strategies are complementary since the distributed loss model can be used to simulate shock–induced losses between blades, while the boundary layer method is better suited for the evaluation of the effects of viscosity in the near–wall regions.

The coupling of an inverse method with a boundary layer method is already present in the literature [1], while the inverse formulation of the distributed loss model is basically new. Our inverse design tool presents more features than the previously available design tools of the same type.

Stressing "of the same type" in the last sentence is not casual. The inverse method based on standard Euler or Navier–Stokes solvers can handle much more complex flows, in particular three dimensional turbulent flows in a blade row. The trade–off between flow complexity and execution speed must be carefully assessed in every specific design problem, but it seems a promising direction in the research to work towards the extension of existing flow solvers to single–pass inverse methods.

Flow design optimization [25], i.e. optimization of the flow based on an inverse method, as opposed to shape optimization , is further an interesting topic in the situation where a flow of "good quality" is sought. In this case, the reduced computational requirements of our method are interesting for coupling an optimization procedure with the inverse method and its adjoint formulation to find the sensitivities. More complex constraints such as a prescribed outlet surface or only bounds on the pressure (opposed to a given pressure distribution), could then be handled.

Appendix A

Distributed Loss Model

The distributed loss model presented in section 5.2 can be extended to quasi–three–dimensional flows by incorporating a component in θ direction of the frictional force $\mathbf{G}_\tau$.

We consider a frictional force $\mathbf{G}_\tau = (F_{\tau.x}, F_{\tau.r}, F_{\tau.\theta})^T$ responsible for the entropy generation in the flow. The three dimensional Euler equations in cylindrical coordinates are augmented by the external force $\mathbf{G}_\tau$

$$\partial_x (r\rho u) + \partial_r (r\rho v) + \partial_\theta (\rho w) = 0 \tag{A.1a}$$

$$\partial_x \left(r\rho u^2\right) + \partial_r (r\rho u v) + \partial_\theta (\rho u w) = -r\partial_x p + r\rho F_{\tau.x} \tag{A.1b}$$

$$\partial_x (r\rho u v) + \partial_r \left(r\rho v^2\right) + \partial_\theta (\rho v w) - \rho w^2 = -r\partial_r p + \rho r F_{\tau.r} \tag{A.1c}$$

$$\partial_x (r\rho u w) + \partial_r (r\rho v w) + \partial_\theta \left(\rho w^2\right) + \rho v w = -\partial_\theta p + \rho r F_{\tau.\theta} \tag{A.1d}$$

$$u\partial_x H + v\partial_r H + w\partial_\theta H = 0 \tag{A.1e}$$

Passage averaging and transformation to Crocco's formulation yields

$$\partial_x (Br\overline{\rho}\widetilde{u}) + \partial_r (Br\overline{\rho}\widetilde{v}) = 0 \tag{A.2a}$$

$$\nabla \hat{H} - \widetilde{T}\nabla\hat{s} - \frac{\widetilde{C}}{r^2}\nabla\widetilde{C} = \widetilde{\mathbf{u}}^\perp\widetilde{\omega} + \mathbf{F}_B + \widetilde{\mathbf{F}_\tau} \tag{A.2b}$$

$$\widetilde{u}\partial_x \widetilde{C} + \widetilde{v}\partial_r \widetilde{C} = r\left(F_{B.\theta} + \widetilde{F_{\tau.\theta}}\right) \tag{A.2c}$$

$$\widetilde{u}\partial_x \hat{H} + \widetilde{v}\partial_r \hat{H} = 0, \tag{A.2d}$$

where $\widetilde{\mathbf{F}_\tau} = (\widetilde{F_{\tau.x}}, \widetilde{F_{\tau,r}})^T$ represents the components of $\widetilde{\mathbf{G}_\tau}$ in the meridional plane, and $\mathbf{F}_B$ has the same meaning as in section 5.1.1.

The entropy transport equation is derived by dotting (A.2b) by $\widetilde{\mathbf{u}} = (\widetilde{u}, \widetilde{v})^T$ and inserting (A.2c) and (A.2d)

$$-\widetilde{T}\widetilde{\mathbf{u}} \cdot \nabla\hat{s} = \widetilde{\mathbf{u}} \cdot \widetilde{\mathbf{F}_\tau} + \widetilde{w}\widetilde{F_{\tau.\theta}}. \tag{A.3}$$

Under the distributed loss model assumption, $\mathbf{G}_\tau$ is a vector anti–parallel to the velocity vector $\mathbf{v} = (u, v, w)^T$. Thus there is a positive function F_τ such that

$$\mathbf{G}_\tau = -\frac{\mathbf{v}}{\|\mathbf{v}\|}F_\tau. \tag{A.4}$$

The entropy production equation (A.3) can therefore be rewritten as

$$\widetilde{T}\widetilde{\mathbf{u}} \cdot \nabla\hat{s} = \|\widetilde{\mathbf{v}}\|\widetilde{F_\tau}. \tag{A.5}$$

The transformation to inverse coordinates is straightforward and gives

$$\partial_\sigma\left(\frac{J}{h^2}\partial_\sigma \mathbf{x}\right) + \partial_\psi\left(\frac{h^2}{J}\partial_\psi \mathbf{x}\right) = 0 \tag{A.6a}$$

$$\partial_\psi \hat{H} - \widetilde{T}\partial_\psi \hat{s} - \frac{1}{\overline{\rho}rBJ}\partial_\psi\left(\frac{h^2}{\overline{\rho}rBJ}\right) - \frac{\widetilde{C}}{r^2}\partial_\psi \widetilde{C} = -\frac{J}{h^2}(\mathbf{F} + \widetilde{\mathbf{F}_\tau}) \cdot \mathbf{x}_\sigma^\perp \tag{A.6b}$$

$$\partial_\sigma \widetilde{C} = \overline{\rho}r^2BJ(F_{B.\theta} + \widetilde{F_{\tau.\theta}}) \tag{A.6c}$$

$$\partial_\sigma \hat{H} = 0 \tag{A.6d}$$

$$\frac{\widetilde{T}}{\overline{\rho}rBJ}\partial_\sigma \hat{s} = \|\widetilde{\mathbf{v}}\|\widetilde{\mathbf{F}_\tau} \tag{A.6e}$$

The right hand side of (A.6e) can be rewritten in inverse coordinates with the help of the expression

$$\|\widetilde{\mathbf{v}}\|^2 = \left(\frac{h}{\overline{\rho}rBJ}\right)^2 + \left(\frac{\widetilde{C}}{r}\right)^2. \tag{A.7}$$

The description of the distributed loss model in the case of a quasi–three–dimensional flow is complete.

Bibliography

[1] N. Ahmed and D. Myring. An inverse method for the design of axis–symmetric optimal diffusers. *International Journal for Numerical Methods in Engineering*, 22:377–394, 1986.

[2] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, L. C. McInnes, and B. F. Smith. PETSc home page. http://www.mcs.anl.gov/petsc, 2001.

[3] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhauser Press, 1997.

[4] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.0, Argonne National Laboratory, 2001.

[5] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. V. der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.* SIAM, 1994.

[6] A. Borzi, K. Morton, E. Süli, and M. Vanmaele. Multilevel solution of cell vertex Cauchy–Riemann equations. *SIAM Journal on Scientific Computing*, 18(2):444–459, 1997.

[7] K. Burrage, J. Erhel, B. Pohl, and A. Williams. A deflation technique for linear system of equations. *SIAM Journal of Scientific Computing*, 19(4):1245–1260, 1998.

[8] P. Chaviaropoulos, V. Dedoussis, and K. Papailliou. On the 3–D inverse potential target pressure problem. Part 1. Theoretical aspects and method formulation. *Journal of Fluid Mechanics*, 282:121–146, 1995.

[9] P. Chaviaropoulos, V. Dedoussis, and K. Papailliou. On the 3–D inverse potential target pressure problem. Part 2. Numerical aspects and application to duct design. *Journal of Fluid Mechanics*, 282:147–162, 1995.

[10] T. Dang and V. Isgro. Inverse method for turbomachine blades using existing time–marching techniques. Technical report, The American Society of Mechanical Engineers, 1994.

[11] T. Dang and V. Isgro. Euler–based inverse method for turbomachine blades Part 1: Two dimensional cascades. *AIAA Journal*, 33(12):2308–2315, 1995.

[12] P. Daripa. On applications of a complex variable method in compressible flows. *Journal of Computational Physics*, pages 337–361, 1989.

[13] P. Daripa and L. Sirovich. An inverse method for subcritical flows. *Journal of Computational Physics*, 63:311–328, 1986.

[14] L. de C. Santos. Convergence acceleration strategies for an inverse design method. Technical report, AIAA Paper, 1999.

[15] V. Dedoussis, P. Chaviaropoulos, and K. Papailliou. Rotational compressible design method for two-dimensional internal flows. *AIAA Journal*, 31:551–558, 1993.

[16] A. Demeulenaere and R. van den Braembussche. Three–dimensional inverse method for turbomachinery blading design. *ASME Journal of Turbomachinery*, 120(2):247–255, 1998.

[17] J. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization*. Prentice–Hall, Englewood Cliffs, NJ, 1983.

[18] J. D. Denton. *Operating Instruction for the Program LOSS3D*.

[19] M. Drela and M. B. Giles. Viscous–inviscid analysis of transonic and low Reynolds number airfoils. *AIAA Journal*, 26(10):1347–1355, 1987.

[20] G. Dulikravich. A criteria for surface pressure specification in aerodynamic shape design. Technical report, AIAA Paper, 1990.

[21] R. Duraiswami and A. Prosperetti. Orthogonal mapping in two dimensions. *Journal of Computational Physics*, (98):254–268, 1992.

[22] L. Eca. 2D orthogonal grid generation with boundary point distribution control. *Journal of Computational Physics*, (125):440–453, 1996.

[23] J. V. et al. *NSMB Handbook 4.5*, May 2001.

[24] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, 1998.

[25] M. Ferlauto, A. Iollo, and L. Zannetti. Optimal inverse method for turbomachinery design. In *Proceedings of the ECCOMAS 2000*, 2000.

[26] G. J. Fix and M. E. Rose. A comparative study of finite element and finite difference method for Cauchy–Riemann type equations. *SIAM Journal on Numerical Analysis*, 22(2):250–261, 1985.

[27] P. Garabedian and G. McFadden. Design of supercritical swept wings. *AIAA Journal*, 30(3), 1982.

[28] K. Giannakoglou, T. I. Pappou, A. Giotis, and D. Koubogiannis. A parallel inverse–design algorithm in aeronautics based on genetic algorithms and the adjoint method. In *Proceedings of the ECCOMAS 2000*, 2000.

[29] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. Springer–Verlag, 1991.

[30] C. Hirsch. *Numerical Computation of Internal and External Flows*, volume 1. John Wiley & Sons, Chichester, 1988.

[31] C. Hirsch and H. Deconinck. Through flow models for turbomachines: Stream surface and passage averaged representations. In S. Ucer, P. Stow, and C. Hirsch, editors, *Thermodynamics and Fluid Mechanics of Turbomachinery*, volume 1 of *NATO ASI Series*. Martinus Nijhoff, 1985.

[32] C. Hirsch and G. Warzee. An integrated quasi–3D finite element calculation program for turbomachinery flows. *Journal of Engineering Power*, 101, 1979.

[33] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, 1988.

[34] A. Jameson. Essential elements of computational algorithms for aerodynamic analysis and design. Technical Report 97-68, ICASE, NASA Langley Research Center, Hampton VA, December 1997.

[35] I. Jennions and P. Stow. A quasi–three–dimensional turbomachinery blade design system, Part 1 — Through–flow analysis. Technical report, Rolls–Royce Ltd., 1984.

[36] J. J. Keller. Inverse Euler equations. *Zeitschrift für Angewandte Mathematik und Physik*, 49(3):363–383, 1998.

[37] P. Knupp and S. Steinberg. *Fundamentals of Grid Generation.* CRC Press, Boca Raton, 1994.

[38] M. J. Lighthill. A mathematical method of cascade design. *Reports and Memoranda*, (2104), 1945.

[39] M. J. Lighthill. A new method for two–dimensional aerodynamic design. *Reports and Memoranda*, (2112), 1945.

[40] A. McDonald and R. Fox. An experimental investigation of incompressible flow in conical diffusers. *Int. J. Mech. Sci.*, 8:125–139, 1966.

[41] B. Mohammadi. Practical applications to fluid flows of automatic differentiation for design problems. In *Inverse Design and Optimization Methods*, Lecture Series 1997–05. von Karman Institute for Fluid Dynamics, 1997.

[42] A. Oyama, S. Obayashi, and K. Nakahashi. Fractional factorial design of genetic coding for aerodynamic optimization. Technical report, AIAA Paper, 1999.

[43] J. Périaux, G. Bugeda, P. K. Chaviaropoulos, T. Labrujère, and B. Stoufflet, editors. *EUROPT–A European Initiative on Optimum Design Methods in Aerodynamics*, volume 55 of *Notes on Numerical Fluid Mechanics.* Vieweg, 1997.

[44] R. Panton. *Incompressible Flows.* John Wiley and Sons, Chichester, 1984.

[45] M. E. Rose. A unified numerical treatment of the wave equation and the Cauchy–Riemann equations. *SIAM Journal on Numerical Analysis*, 18(2):372–376, 1981.

[46] A. Scascighini and A. Troxler. Inverse design for turbine components: Annual report of the KTI project, January 2000.

[47] A. Scascighini, A. Troxler, and R. Jeltsch. Inverse design of an axis–symmetric diffuser. In *Proceedings of the ECMI 2000*, Springer Series in Industrial Mathematics, 2000.

[48] J. A. Schetz. *Boundary Layer Analysis*. Prentice Hall, 1993.

[49] H. Schlichting and K. Gersten. *Grenzschicht–Theorie*. Springer, 1997.

[50] G. Shroff and B. Keller. Stabilization of unstable procedures: the Recursive Projection Method. *SIAM Journal on Numerical Analysis*, 30(4):1099–1260, 1993.

[51] M. Stanciu, B. Mohammadi, and S. Moreau. Turbulent flow simulation and shape optimization in turbomachinery. In *Proceedings of the ECCOMAS 2000*, 2000.

[52] J. D. Stanitz. A review of certain inverse methods for the design of ducts with 2– or 3– dimensional potential flows. In G. Dulikravich, editor, *Proceedings of the Second International Conference on Inverse Design Concepts and Optimization in Engineereing Sciences*, 1987.

[53] P. Tamamidis and D. N. Assanis. Generation of orthogonal grids with control of spacings. *Journal of Computational Physics*, (94):437–453, 1991.

[54] C. Thompson, Z. Warsi, and C. Mastin. *Numerical Grid Generation, Foundations and Applications*. North Holland, 1985.

[55] M. Trigg, G. Tubby, and A. Sheard. Automatic genetic optimization approach for two–dimensional blade profile design for steam turbines. *Journal of Turbomachinery*, 121(11), 1999.

[56] A. Troxler. Ein Deflationsverfahren für die Lösung der Inversen Eulergleichungen. Master's thesis, ETH Zurich, 1999.

[57] A. Troxler and A. Scascighini. Inverse design for 2D internal flows. In *Proceedings of the 5th ISAIF Conference*, Gdansk, September 2001.

[58] H. A. van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of non-symmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 13:631–644, 1992.

[59] M. Vanmaele, K. Morton, E. Süli, and A. Borzi. Analysis of the cell vertex finite volume method for the Cauchy–Riemann equations. *SIAM Journal on Numerical Analysis*, 34(5):2043–2062, 1997.

[60] A. E. P. Veldmann. A numerical view on strong viscous–inviscid interaction. In W. Habashi, editor, *Computational Methods in Viscous Flows, Recent Advances in Numerical Methods in Fluids*, volume 3. Pineridge Press, 1984.

[61] Z. Wang and G. Dulkravich. Inverse shape design of turbomachinery airfoils using Navier–Stokes equations. Technical report, AIAA Paper, 1995.

[62] D. G. Wilson. *The design of high efficiency turbomachinery and gas turbines*. MIT Press, 1984.

[63] K. F. C. Yiu and M. Giles. Simultaneous viscous–inviscid coupling via transpiration. *Journal of Computational Physics*, 120:157–170, 1995.

[64] L. Zannetti. A natural formulation for the solution of two–dimensional or axis–symmetric inverse problems. *Int. J. Numer. Meth. Eng.*, 22:451–463, 1986.

Curriculum Vitae

28th May 1974	Born in Sorengo, Switzerland
1980–1985	Scuola elementare, Canobbio
1985–1989	Scuola media, Canobbio and Viganello
1989–1993	Liceo Lugano I, Matura Typus C
1993–1998	Study of mathematics at ETH Zurich
April 1998	Diploma in mathematics with honors
April 1998–present	Research and teaching assistant at the Seminar for Applied Mathematics, ETH Zurich