Drukcorrectiealgoritmen voor willekeurige fluïda bij lage snelheden, toegepast op simulaties van niet-voorgemengde vlammen

Pressure-Correction Algorithms for General Fluids at Low Speeds, Applied to Non-Premixed Flame Simulations

Pieter Rauwoens

Promotoren: prof. dr. ir. B. Merci, prof. dr. ir. J. Vierendeels
Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen: Werktuigkunde-Elektrotechniek

Vakgroep Mechanica van Stroming, Warmte en Verbranding
Voorzitter: prof. dr. ir. R. Sierens
Faculteit Ingenieurswetenschappen
Academiejaar 2007 - 2008

UNIVERSITEIT GENT

ISBN 978-90-8578-209-4
NUR 978, 928
Wettelijk depot: D/2008/10.500/28

Promotoren: prof. B. Merci,
prof. J. Vierendeels

Vakgroep Mechanica van Stroming, Warmte en Verbranding
Sint-Pietersnieuwstraat 41
B-9000 Gent
België

Preface

During the three and a half years as a PhD-student, I have been in touch with various facets of research. There is the initial enthousiasm, the first confrontation with its own lack of knowledge, the exploration of the undiscovered research area, getting lost in the enormous amount of literature on the topic and not knowing what still is to be done in the field. But suddenly, one has the feeling that in one tiny area, something is missing and that one can make a difference. Then there is the joy of the discovery of new things. And if these discoveries arouse others' interest, knowing that, what one has discovered, will not disappear in vanity after the PhD is finished, but might be useful to other researchers, one can only be happy with what has been achieved.

I wish to thank my promotor prof. Bart Merci, for offering me the opportunity to work in his research group and get in touch with all these experiences. Because of his policy, I have met many fellow researchers from all over the world. Thanks for making it possible to share my discoveries with other people at many conferences and in journal publications.

Also my second promotor, prof. Jan Vierendeels deserves my gratitude. He has been a great help in the numerical and programming work, necessary to achieve the results of this thesis. Thanks for the advice, that was always to the point.

I wish to thank prof. Erik Dick for his valuable insights, helping me in the last months to order my findings. I am sure that, because of his recommendations, value is added to this work.

I am also thankful to my collegues, especially Dieter Fauconnier, who, at the same moment, started his PhD. Many frustrations flew off over one or two beers, where, at the same time, new ideas occured. With the other collegues, I stuck to coffees over noon, but sure I considered these equally imperative.

I also thank my parents and family. Not knowing what exactly I was working on, they were always interested. I hope that, at my public defense, at last they will have a clue of what I did, so they can finally answer people's questions.

Not at least I thank my companion in life, Mieke. During my PhD, she has been

both my inspiration and consolation. What started as love, has recently been sealed with marriage. I am grateful for the support she gave, especially in the last few months, when more words were spoken between me and my computer, than between the two of us. I hope to make amends for this behaviour as soon as possible.

Pieter Rauwoens
March 10th, 2008

Nomenclature

Latin Symbols (I)

A	Arrhenius constant	$(K.s)^{-1}$
A	fluid A	
B	fluid B	
Cond	conduction term	
C_s	Smagorinsky constant	
D	diffusion coefficient	m^2/s
D	discrete divergence operator	m^{-1}
D	Lagrangian derivative	
D^T	thermal diffusion coefficient	$Pa.s$
E	specific total internal energy	J/kg
E_A	activation energy	J/mol
H	specific total enthalpy	J/kg
F	flux	kg/s
F	fuel	
F	LES-filter	
G	discrete gradient operator	m^{-1}
H	Heaviside function	
J	species diffusion flux	$kg/(m^2.s)$
L	discrete Laplacian operator	m^{-2}
L	reference lenght	m
M	amount of reactions	
N	amount of scalars	
N	amount of species	
N_x	amount of gridpoints	
O	oxidizer	
P	pressure vector	Pa
P	products	

Latin Symbols (II)

R	gas constant per mass unit	$J/(kg.K)$
S	surface	m^2
T	temperature	K
U	contravariant velocity	s^{-1}
U	diffusion velocity	m/s
V	contravariant flux	m^3/s
V	volume	m^3
W	molecular weight	$kg/mole$
X	mole fraction	
Y	species mass fraction	
Z	element mass fraction	
a	contravariant base vector	m^{-1}
c_p	specific heat at constant pressure	$J/(kg.K)$
c_p^m	mean specific heat at constant pressure	$J/(kg.K)$
c_v	specific heat at constant volume	$J/(kg.K)$
det	determinant	
d	total derivative	
e	specific internal energy	J/kg
f	external force	m/s^2
f	fuel elements mass	kg/m^3
g	gravitational constant	m/s^2
g	metric tensor	m^6
h	specific static enthalpy	J/kg
k	reaction rate coefficient	s^{-1}
$\dot{m}$	mass flux	kg/s
n	reaction order	$mole$
n	outward pointing normal	m
p	(kinematic) pressure	Pa
p_0	thermodynamic pressure	Pa
p_2	kinematic pressure	Pa
q	energy flux	W/m^2
q^0	heat release	J/kg
s	stoichiometric ratio	
u	velocity	m/s
$\dot{v}$	face velocity flux	m^3/s
t	time	s
x	space direction	m
y	generic scalar	
y	second space direction	m
z	element mass fraction per species	

Calligraphic Symbols

$\mathcal{D}$	diffusion term	$Pa.s$
$\mathcal{G}$	generic equation of state expression	
$\mathcal{H}$	generic equation of state expression	
$\mathcal{H}_C$	chemical operator	
$\mathcal{M}$	generic molecule	
$\mathcal{O}$	order	
$\mathcal{P}$	generalised pressure gradient	$Pa.m$
$\dot{\mathcal{Q}}$	energy source term	W/s
$\mathcal{R}$	universal gas constant per mole unit	$J/(mole.K)$

Greek Symbols (I)

Δ	difference	
Δ	filter width	
Δt	time step	s
Δx	grid spacing in x-direction	m
Δy	grid spacing in y-direction	m
Δh_f^0	specific enthalpy of formation	J/kg
Γ	gamma-function	
Γ	rectangular mapped control volume	
Π	product	
Σ	sum	
Ω	control volume	m^3
α	multistage coefficient	
α	rescaling factor	
β	Arrhenius constant	
γ	specific heat ratio	
δ	discrete derivative	
δ_{ij}	Kronecker delta	
ε	cell-face velocity correction	m/s
$\hat{\varepsilon}$	cell-face velocity correction	m/s
ϵ	non-dimensional temperature difference	
ζ	damping factor	
λ	thermal conductivity	$W/(m.K)$
μ	dynamic viscosity	$Pa.s$
ν	kinematic viscosity	m^2/s
ν	reaction constant	$mole$
ξ	Cartesian mapped space direction	
ξ	mixture fraction	

Greek Symbols (II)

ρ	density	kg/m^3
τ	time-scale	s
τ	viscous stress tensor	Pa
χ	scalar dissipation rate	s^{-1}
ϕ	generic variable	
$\dot{\omega}$	chemical source term	$kg/(m^3.s)$

Symbols

∂	boundary surface of control volume
∂	partial derivative
∇	divergence/gradient operator
$\mathbb{R}$	real space

Dimensionless Groups

Da	Damköhler number
Du	Dufour number
Ex	external forces dimensionless group
Fr	Froude number
Le	Lewis number
M	Mach number
Nu	Nusselt number
Pr	Prandtl number
Ra	Rayleigh number
Re	Reynolds number
Sc	Schmidt number

Subscripts (I)

∞	reference value
$\overline{\ }$	corrected interpolation
A	fluid A
B	fluid B
C	convective
D	diffusive
E	East
F	fuel

Subscripts (II)

H	hydrodynamic
L	left
N	North
O	oxidizer
R	right
S	South
W	West
b	backward
c	cold
c	convective
d	diffusive
f	forward
h	hot
i	node index in 1D
i, j	node index in 2D
$\max$	maximum value in the field
st	stoichiometric
t	turbulent
π	spurious mode

Superscripts

$\hat{}$	non-dimensional variable
$\bar{}$	arithmetic mean of face values
$\bar{}$	Reynolds average
$\tilde{}$	arithmetic mean of node values
$\tilde{}$	Favre average
$\check{}$	known value
$'$	correction value
$'$	Reynolds fluctuation
$''$	Favre fluctuation
$*$	predictor value
$\diamond$	operator-split time derivative
i	coordinate index in mapping theory
n	time level
T	transposed
(ν)	multistage level

Abbreviations

1D	one-dimensional
2D	two-dimensional
CFD	computational fluid dynamics
CFL	Courant-Friedrich-Levy number
CMC	conditional moment closure
CSP	computational single perturbation
DNS	direct numerical simulation
FGM	flamelet generated manifold
FPI	flame prolongation of intrinsic low-dimensional manifold
ICE-PIC	invariant constrained-equilibrium edge pre-image curve
ILDM	intrinsic low-dimensional manifold
LES	large-eddy simulation
MAC	marker and cell
PDF	probability density function
PWI	pressure-weighted interpolation
QSSA	quasi steady-state assumption
RANS	Reynolds-averaged Navier-Stokes
RCCE	rate-controlled constrained equilibrium
REDIM	reaction-diffusion manifolds
RHS	right hand side
SIMPLE	semi-implicit method for pressure-linked equations
TNF	turbulent non-premixed flames
TVD	total variation diminishing

Summary

Computational fluid dynamics (CFD) becomes increasingly important in the design process of modern systems involving reacting flows. Design and optimisation of, for instance, industrial combustion devices, is intensely guided by numerical simulations nowadays. The complexity of the processes occurring in such systems demands for accurate models and advanced numerical methods. Direct numerical simulation (DNS) and large-eddy simulation (LES) are very powerful high-fidelity CFD tools that have the potential to meet these demands. Unfortunately, these tools can only predict quantitative results if the underlying algorithms are capable of dealing with time-accurate simulations of reacting flows.

Many algorithms used by researchers, although performing well for constant density (non-reacting) flows, give rise to instabilities in the solution when adopted in variable density flows, where density can strongly vary from cell to cell. These algorithms are part of the class denoted here as *continuity-constraint pressure-correction schemes*. Other algorithms perform better with respect to stability, but predict solutions that are far from physically possible: the predicted states do not correspond to the realistic equation of state. These algorithms are denoted here as *analytical compatibility-constraint pressure-correction schemes*. Because of these shortcomings, I develop an algorithm that (i) is stable and robust, (ii) conserves mass and scalars, such as energy and fuel elements mass, (iii) predicts states that match exactly with the equation of state, (iv) can be efficiently implemented and (v) allows time-accurate solutions. Using this type of algorithm should provide a consistent code, that can serve as a basis for quantitative predictions and further model development in LES, without fearing unexpected instabilities. The algorithm's applicability contains low-Mach number flows of general fluids, described by an unlimited amount of scalar transport equations and an arbitrary equation of state. Its primary area of application, however, is in these cases where a non-linear equation of state exists. Turbulent non-premixed flames are a key example for this area of interest.

The developed algorithm is situated in the class of pressure-correction algorithms. In this segregated approach, the equations are solved sequentially. At the end of every timestep, a global correction step is needed to account for the pressure influence. The pressure follows from an elliptic equation, derived from a constraint on the velocity field. I construct the constraint from a combination of the discrete equations of continuity and scalars, imposing that the newly predicted state should be compatible, in agreement

with the equation of state. This leads to the *discrete compatibility-constraint pressure-correction algorithm*. It is different from the standard pressure-correction schemes, where the constraint is formulated either solely based on the continuity equation (continuity-constraint pressure-correction scheme) or from an analytical combination of the material derivative of the equation of state and the continuity and scalar equations (analytical compatibility-constraint pressure-correction scheme).

Specifically when the equation of state expresses a non-linear relationship between the state variables, the developed algorithm reveals its superior qualities. The extreme case of a reacting flow where chemistry is assumed infinitely fast (the Burke-Schumann chemistry model) imposes a highly non-linear, non-differentiable relationship between density and fuel elements mass. In this case, all standard algorithms fail, whereas the newly developed algorithm predicts a correct result. This is clearly shown in a set of one-dimensional test cases, involving convection and diffusion of sharp initial scalar gradients. In these test-cases, three fluid types are investigated: a single-fluid ideal gas at different temperatures, a two-fluid non-reacting flow and a two-fluid combusting flow. The continuity-constraint pressure-correction scheme reveals instabilities in the solutions in nearly all cases and can therefore not be adopted in variable density flows with sharp density gradients. The analytical compatibility-constraint pressure-correction scheme yields stable results, but predicts states that deviate strongly from the equation of state in case of a non-linear equation of state (combusting flow) and is therefore inaccurate. The discrete compatibility-constraint pressure-correction scheme does yield stable and accurate results in all cases.

The algorithm still proves reliable in more-dimensional configurations, albeit that the issue of odd-even decoupling needs to be resolved first. Indeed: if a collocated grid arrangement is used, a spurious mode for the pressure can appear. I construct a velocity-interpolation formula for variable density flows, that can be applied in combination with the developed algorithm, to suppress the spurious mode and still guarantee a solution, even in enclosed systems.

The resulting collocated algorithm is validated on two-dimensional test cases, including a thermally driven cavity with large horizontal temperature differences. For a broad range of Rayleigh numbers, good solutions are obtained, even on relatively coarse meshes. Moreover, the spurious pressure mode is suppressed. In a second test case, the stability and convergence of the method are demonstrated by means of a reacting and a non-reacting mixing layer. I conclude from this test case that, especially in combusting flows, the continuity-constraint pressure-correction algorithm cannot be stabilised, unless measures are taken that corrupt time-accuracy. The analytical compatibility-constraint pressure-correction algorithm is stable, but predicts a converged solution that differs noticeably from the exact result. The reason for this stems from the uncontrollable drift from the non-linear equation of state. On the other hand, the here presented algorithm yields a stable result and predicts states that exactly match the equation of state. The benefits of higher robustness and greater accuracy are acquired with only a minimal additional computational effort, as compared to the other algorithms, yielding an algorithm that is

not only stable and accurate, but also efficient.

In conclusion, I show in this work that, because of the non-linearity of e.g. the combustion process, standard algorithms are bound to fail and novel algorithms, based on discrete compatibility-constraint formulations are to be adopted, yielding stable and more accurate time-dependent solutions.

Samenvatting

Numerieke stromingsmechanica (CFD) wordt alsmaar belangrijker in de ontwerpfase van hedendaagse systemen waar reagerende stromingen deel van uitmaken. Numerieke simulaties worden dan ook veelvuldig gebruikt bij het ontwerp en optimalisatie van bijvoorbeeld industriële branders. Een goede voorspelling van de ingewikkelde processen die zich voordoen in dergelijke systemen is enkel mogelijk indien men beschikt over nauwkeurige modellen en geavanceerde numerieke methoden. Voorbeelden van dergelijke krachtige en betrouwbare CFD-technieken omvatten *directe numerieke simulaties* (DNS) en *large-eddy simulaties* (LES). Helaas zijn deze technieken enkel in staat kwantitatieve voorspellingen te maken als de onderliggende algoritmen geschikt zijn voor tijdsnauwkeurige simulaties van reagerende stromingen.

De algoritmen die veelal gebruikt worden in de onderzoekswereld, blijken aanleiding te geven tot onstabiele oplossingen als deze toegepast worden op stromingen met sterk variabele dichtheid. Nochtans bleken deze algoritmen goed te functioneren in stromingen met constante dichtheid. De desbetreffende algoritmen maken deel uit van een klasse die we hier aanduiden onder de noemer *massabehoud gebonden drukcorrectiealgoritmen*. Andere gangbare algoritmen blijken wel stabiel, maar voorspellen oplossingen die fysisch niet mogelijk zijn. De voorspelde toestanden voldoen immers niet aan de toestandsvergelijking. Deze algoritmen vallen onder de klasse *analytische-compatibiliteit gebonden drukcorrectiealgoritmen*. Omwille van deze tekortkomingen, ontwikkel ik hier een nieuw algoritme met volgende eigenschappen: (i) het is stabiel en robuust, (ii) het behoudt massa en andere scalaire grootheden, zoals energie en atomaire brandstofmassa, (iii) het voorspelt toestanden die exact voldoen aan de toestandsvergelijking, (iv) het kan doeltreffend worden geïmplementeerd en (v) het maakt tijdsnauwkeurige simulaties mogelijk. Het gebruik van dergelijk algoritme kan bijgevolg aanleiding geven tot een consistent simulatieprogramma, dat op zijn beurt kwantitatieve voorspellingen kan maken. Zodoende kan een basis gelegd worden voor verdere modellering op gebied van LES van reagerende stromingen, zonder te vrezen voor onverwacht onstabiel gedrag. Het toepassingsgebied van dit algoritme bevat stromingen van willekeurige fluïda bij lage Machgetallen. Een willekeurig fluïdum wordt beschreven door een onbegrensd aantal scalaire grootheden, transportvergelijkingen voor die grootheden en een willekeurige toestandsvergelijking die deze grootheden koppelt. Het algoritme heeft voornamelijk een toegevoegde waarde wanneer de toestandsvergelijking niet-lineair is, zoals in het geval van turbulente niet-voorgemengde vlammen.

Het ontwikkelde algoritme bevindt zich in de klasse van de drukcorrectiealgoritmen. In deze gesegregeerde oplossingsmethode worden de vergelijkingen sequentieel opgelost. Op het einde van elke tijdstap volgt een globale correctiestap, die de druk in rekening brengt. De druk wordt berekend uit een elliptische vergelijking, die afgeleid wordt uit een vergelijking die een beperking oplegt op het snelheidsveld. In het nieuwe algoritme wordt deze beperkende vergelijking opgebouwd uit een combinatie van de discrete massabehoudswet en de discrete transportvergelijking voor de scalaire grootheden, op een dusdanige manier dat de nieuw voorspelde toestand voldoet aan de toestandsvergelijking. Het aldus gevormde algoritme heet *discrete-compatibiliteit gebonden drukcorrectiealgoritme*. Het verschilt van de standaard drukcorrectiealgoritmen, waar de beperkende vergelijking enkel gebaseerd is op massabehoud (massabehoud gebonden drukcorrectiealgoritme) of op een analytische combinatie van de materiële afgeleide van de toestandsvergelijking enerzijds en de massabehoudswet en de transportvergelijkingen van de scalaire grootheden anderzijds (analytische-compatibiliteit gebonden drukcorrectiealgoritme). In het bijzonder wanneer de toestandsvergelijking een niet-linear verband beschrijft tussen de toestandsgrootheden, is het gebruik van het discrete-compatibiliteit gebonden drukcorrectiealgoritme behept met voortreffelijke kwaliteiten. Het extreme geval van reagerende stromingen waar de scheikundige reacties oneindig snel verlopen (het scheikundig model van Burke en Schumann) legt een sterk niet-lineair en niet-differentieerbaar verband op tussen dichtheid en atomaire brandstofmassa. In tegenstelling tot het nieuw ontwikkelde algoritme, slaagt een standaardalgoritme niet in een correcte voorspelling van dit geval.

Deze bevinding wordt duidelijk aangetoond in enkele ééndimensionale testgevallen, waarbij een initieel scherpe gradiënt in een scalaire grootheid wordt geconvecteerd of gediffundeerd. Hierbij worden drie types fluïda onderzocht: de stroming van een ideaal gas, dat zich op verschillende temperatuur bevindt; de stroming van twee fluïda die mengen doch niet reageren; en een stroming van twee reagerende fluïda. Het massabehoud gebonden drukcorrectiealgoritme geeft aanleiding tot onstabiel gedrag in bijna alle onderzochte gevallen en is bijgevolg niet toepasbaar in stromingen met scherpe gradiënten in het dichtheidsveld. Het analytische-compatibiliteit gebonden drukcorrectiealgoritme leidt wel tot stabiele oplossingen, doch voorspelt toestandsgrootheden die sterk afwijken van de fysische toestandsvergelijking indien deze laatste niet-linear is in de toestandsvariabelen, zoals het geval is in reagerende stromingen. Het discrete-compatibiliteit gebonden drukcorrectiealgoritme daarentegen levert wel stabiele oplossingen die bovendien in alle gevallen voldoen aan de toestandsvergelijking.

Ook in meerdere dimensies levert het algoritme betrouwbare resultaten, op voorwaarde dat een oplossing geboden wordt aan het probleem van de ontkoppeling tussen de even en oneven knooppunten. Indien de simulatie immers uitgevoerd wordt op een rekenrooster waar alle variabelen op dezelfde plaats gestockeerd worden, verschijnt er een niet-fysische drukgolf in de oplossing. Om deze niet-fysische golf te onderdrukken, ontwerp ik een bijzondere snelheidsinterpolatie, specifiek voor stromingen met variabele dichtheid. Ook wanneer de stroming zich in een ingesloten ruimte bevindt, garandeert het gebruik van deze interpolatie nog steeds de oplosbaarheid van de elliptische drukcorrectievergelijk-

ing. De combinatie van de snelheidsinterpolatie met het discrete-compatibiliteit gebonden drukcorrectiealgoritme, levert een algoritme dat gemakkelijk kan aangewend worden in meerdimensionale stromingen.

Een eerste validatie van het algoritme in twee dimensies betreft een thermisch gedreven caviteit die onderworpen wordt aan grote horizontale temperatuursverschillen. Voor een grote waaier aan Rayleigh getallen worden goede oplossingen bekomen, zelfs indien de simulatie uitgevoerd wordt op relatief grove rekenroosters. Bovendien wordt de niet-fysische drukgolf onderdrukt.

De stabiliteit en de convergentie van het algoritme worden onderzocht met behulp van een tweede testgeval. Het betreft hier een tweedimensionale menglaag, die de menging beschrijft van twee inerte of reagerende fluïda. Uit de verkregen resultaten besluit ik dat het massabehoud gebonden drukcorrectiealgoritme, in het bijzonder bij toepassing op reagerende stromingen, geen stabiele resultaten kan leveren, tenzij maatregelen getroffen worden die de tijdsnauwkeurigheid van het algoritme schaden. Het analytische-compatibiliteit gebonden drukcorrectiealgoritme is wel stabiel, maar voorspelt een oplossing die merkbaar verschilt van de juiste oplossing. Deze foute voorspelling vindt zijn oorsprong in de onbeheersbare afwijking van de niet-lineaire toestandsvergelijking. Het discrete-compatibiliteit gebonden drukcorrectiealgoritme daarentegen levert een stabiele oplossing, wier toestandsgrootheden exact voldoen aan de toestandsvergelijking. Deze grotere robuustheid en nauwkeurigheid kunnen verkregen worden tegen een kleine meerkost in termen van rekentijd. Het uiteindelijke algoritme is dus niet enkel stabiel en nauwkeurig, maar tevens efficiënt.

Samenvattend kunnen we stellen dat dit werk het bewijs levert dat, omwille van de niet-lineaire toestandsvergelijking in, onder meer, het verbrandingsproces, standaardalgoritmes niet leiden tot het gewenste resultaat. Bijgevolg dienen nieuwe algoritmes aangewend te worden, gebaseerd op de grondbeginselen van het hier ontwikkelde discrete-compatibiliteit gebonden drukcorrectiealgoritme. Op die manier kunnen stabiele en nauwkeurige oplossingen bekomen worden in tijdsnauwkeurige simulaties.

Contents

List of figures

List of tables

Chapter 1

Introduction

Combustion has always played an important role in life of mankind. One can simply not imagine living without devices, ranging from simple domestic tools, such as cooking and heating furnaces, over internal combustion engines used in transport vehicles, up to the burning of fossil fuels in conventional power plants. It is therefore not surprising that a lot of resources are spent on research in that area.

Combustion research has traditionally been driven by four major technological and societal concerns [37], namely energy and fuels, environment and health, hazards, and defense and space. These four areas command a large share of the modern human activities. Among all the technology drivers, energy sufficiency and environmental quality unquestionably demand the highest concerns in maintaining the prosperity and progress of nowadays society. Concerning energy sufficiency, 85% of the world's energy supply comes from the burning of fossil fuels. Combustion is therefore one of the crucial technologies that power the modern society. Furthermore, since about 30% of the fossil fuels are used for transportation, mainly in the form of petroleum products, the diminishing petroleum reserve and the uncertainty of its supply due to geopolitical considerations exert an increasing stress on the global economy and harmony. Regarding environmental quality, it is recognised that while the decrease of conventional air pollutant emissions, especially those of NO_x and particulate soot, continues to be a major concern in combustion, the revealed consequences of global warming through the emission of anthropogenic sources of combustion-generated greenhouse gases, particularly CO_2, are becoming equally worrisome.

In the light of these concerns, one can ask the question: "If sustainable energy supply and climatic catastrophy control require the phasing out of fossil fuels in favour of renewables, why continue to spend money on combustion research?" The answer to that question is twofold. A major challenge is indeed hidden in new technologies, such as the production and utilisation of hydrogen and bio-fuels, the sequestration of CO_2 and the large

scale deployment of fuel cells. It will, however, take 25-30 years until these technologies are proven to be mature enough to produce hydrogen at large scales either through nuclear fusion or fossil fuel conversion with the generated CO_2 properly sequestered. Insufficiently resolved combustion-related issues in the use of hydrogen are its unique combustion characteristics and its danger in explosion hazards upon accidental leakage. In addition to hydrogen, combustion of biomass is also considered to be an important source of renewable energy in the years to come. Combustion science still has a large role to play in technology improvement in this area [8].

Even with the greenhouse gas constraints, it is likely that combustion of fossil fuels will continue to be the major source for energy well into the 21st century [8]. Consequently, in anticipation of the maturation of other technologies, the primary impact that combustion can make on energy and climate is to further improve the combustion efficiency, which leads to the simultaneous reduction in fuel consumption and CO_2 emission.

As to increase efficiency and decrease pollutant formation, combustor developers have traditionally resorted to empirical methods as engineering practice. Thirty years ago, such empirical methods were all that were used by combustion engineers, because of the extreme complexity of the (turbulent) combustion phenomenon. Today, combustion engineers heavily rely on predictions from Computational Fluid Dynamics (CFD) codes, incorporating models for turbulence, for combustion processes and their interaction. If thirty years ago, empirical models were sufficient for obtaining timely development of combustors using premium fuels and subject to moderate pollutant emmision limits, today, the market demands rapid development of an increasingly broad range of products, subject to even tighter pollutant emission controls. CFD has become a very important component of the engineer's toolkit. Empirical development is still necessary, but this is greatly aided by the insights gained from CFD studies, not only of the particular device of practical interest, but also of the simpler model problems that have been studied in detail by combustion scientists [8].

As an example, I highlight the case of homogeneous charge, compression ignition (HCCI) combustion, where the existence of a mature simulation capability has revolutionised the overall research plan for an industry. The HCCI may or may not have a significant impact on power production, but it illustrates the complete integration of computer modelling into research and development [82].

Such advanced computer-aided development is only feasible if CFD has a quantitative predictability for combusting flows. This drive towards quantitative predictability has been accelerating with the rapid advance in computing power and algorithms. At present, if one can afford the investment in computer power, computational capabilities are such that serious large-scale simulations of fairly complex combustion phenomena, such as those involving turbulent flows and within engines, can be performed with moderate simplifications [65].

The quantitative predictability of the computational simulations is still not reached for general practical applications, but strongly depends on the situation considered. For problems of moderate complexity, existing models are sufficiently accurate and quantitative results can be obtained. The more complex the problem in terms of chemistry, turbulence or geometry, the more complex the simulation, making use of more sophisticated models. The coverage of the entire field of practical combustion applications forms a major challenge for the combustion community. Progress in combustion science can be accelerated by international collaboration among experimental and computational researchers in the field of combustion.

In recent years, outstanding advances have been made in turbulent non-premixed combustion. These advances have been closely associated with a series of international workshops on turbulent non-premixed flames (TNF) [1]. The aim of these workshops has been to encourage close interaction between experimentalists, modellers, and theoreticians, with a focus on clarifying the physics and computational modelling of the chemistry-turbulence interactions that occur, and on improving measurement techniques. It is generally agreed that advances here have been more rapid than in premixed turbulent flames, and that this is partly because there exists a well-developed and widely accepted hierarchy of experiments that is well aligned with the capabilities of the modelling and simulation [9].

The focus of the TNF workshops has been on simple laboratory-scale jet flames, including bluff-body stabilised and swirling flames. Especially this latter configuration, shows large-scale unsteady behaviour of three-dimensional nature. It appears that it may be necessary to move to Large-Eddy Simulation (LES) to get sufficient accuracy in the predictions for the flow and the mixing. By using this LES-technique, large structures are time-accurately followed and the large-scale unsteadiness is resolved, yielding a better solution. There are, however, numerical issues involved. In the conclusion of the latest TNF-workshop, held in Heidelberg (2006), we read:

> *LES is a promising technique to accurately predict turbulent flows. However, many fundamental problems are unsolved, and no single LES procedure has emerged as a standard. Within the context of the TNF Workshop, validation of combustion LES against experimental results, particularly in flames with strong effects of turbulence-chemistry interaction, cannot be carried forward effectively until issues of code verification and LES quality assessment are addressed.*

This strong statement motivates the subject of the present thesis. Since LES inherently is a time-accurate simulation technique, time accuracy has to be retained by the algorithm used to perform the simulation. The use of standard algorithmic techniques in a transient simulation code for non-premixed combustion, reveals serious stability problems, as observed by many researchers [16, 33, 46, 48]. Attempts in stabilizing the codes have been applied, but these jeopardise the results' quality. I believe that, because of the

non-linearity of the combustion process, standard algorithms are bound to fail and novel algorithms with greater ingenuity are to be adopted. The development of such algorithms, specifically designed for transient reacting flow simulations, forms the major innovation of the present work.

Note that, although a time-dependent formulation of non-premixed reacting flows forms the motivation and first application for the algorithm developed by the author, it is not restricted to such flows. On the contrary, the developed algorithm is found to be the only consistent algorithm of its kind for the general class of low-Mach number flows, obeying an arbitrary equation of state in an exact manner. Of course, because of the high non-linearity of combusting flows, the primary area of application is to be found in this field.

The work performed by the author is fundamental in nature, but can be quickly adopted to perform a LES of swirling flames, leading to better computer codes, so that reliable progress can be made in further turbulence and chemistry model development.

Overview

After this introduction, the governing equations that mathematically describe a reacting flow are derived (chapter 2). Emphasis is put on the low-Mach number limit of the equations.

In chapter 3, the number of equations is drastically reduced by introducing physically founded model assumptions. The computational cost is diminished by using models that describe unresolved chemistry, turbulence and their interaction.

Chapter 4 transforms the continuous equations into discrete ones, that can be handled by a calculating machine. Issues concerning stability, accuracy and monotonicity are discussed for a simple model equation.

Using the ingredients of chapters 2, 3 and 4, a novel algorithm for low-Mach number flows of general fluids is constructed in chapter 5. The algorithm is characterised by features of efficiency, accuracy, stability, robustness, conservation and concistency.

The novel algorithm is compared to other existing algorithms in the class of pressure-correction algorithms in chapter 6, proving its superior qualities, e.g. in reacting flows. This is illustrated by means of one-dimensional test cases in chapter 7.

If more-dimensional simulations are to be performed on collocated grids, the issue of odd-even decoupling needs to be resolved. A special cure for variable-density flows, emphasizing the solvability condition, is derived in chapter 8. Two-dimensional validation of the ultimate collocated algorithm is performed in chapter 9.

The major findings by the author are summarised in chapter 10.

Finally, prospects of possible future work, inspired by the author's findings, are listed in chapter 11.

Chapter 2

Governing Equations

In order to perform simulations, a suitable mathematical description of the reality must be provided. Several scientists from the ancient and more recent past have built the foundations of what is now known as the Navier-Stokes equations for multi-component fluids: a set of equations, describing in a sufficiently accurate manner the physical processes that appear in a flow, consisting of several chemical species. Two kinds of equations can be distinguished: differential (or integral) and algebraic equations. The (partial) differential equations describe essentially the properties adherent to the motion of the flow. As they essentially express conservation properties of the flow, they are called *conservation equations*. The algabraic equations describe the properties of the fluid itself. They are called *constitutive laws*.

In this chapter, both kinds of equations are thoroughly described for the general case. The general equations in certain cases describe physical properties that can be neglected. Therefore, simplified, still sufficiently accurate, forms of the equations for three common cases are derived.

2.1 Conservation Equations

Conservation equations can be written in two ways: as an integral or a differential expression. The integral form expresses the physics behind the maths. The differential form is merely a short notation formulation, but requires all variables to be differentiable. In this work, both formulations will be used, depending on which formulation suits best. In the integral formulation, the conservation properties are expressed for an arbitrary volume of fluid (control volume, Ω), bounded by a surface ($\partial\Omega$) with outward pointing normal $\boldsymbol{n}$.

2.1.1 Conservation of Mass

Mass can neither be created, nor destroyed. Consequently, the rate of accumulation of mass inside the control volume Ω has to be balanced by the net outflow of mass through the boundaries:

$$\frac{\partial}{\partial t}\int_{\Omega}\rho\,\mathrm{d}V + \oint_{\partial\Omega}\rho\boldsymbol{u}\cdot\boldsymbol{n}\,\mathrm{d}S \;=\; 0. \tag{2.1}$$

For an infinitely small control volume, the differential version of the above integral expression is obtained:

$$\frac{\partial\rho}{\partial t} + \frac{\partial\rho u_i}{\partial x_i} \;=\; 0. \tag{2.2}$$

The mass conservation equation is also known as the continuity equation.

2.1.2 Conservation of Species Mass Fraction

The conservation of mass can also be applied to each chemical species, keeping in mind that species can be created or destroyed through chemical reactions. The conservation equation for species k then reads:

$$\frac{\partial}{\partial t}\int_{\Omega}\rho Y_k\,\mathrm{d}V + \oint_{\partial\Omega}\rho Y_k\boldsymbol{u_k}\cdot\boldsymbol{n}\,\mathrm{d}S \;=\; \int_{\Omega}\dot{\omega}_k\,\mathrm{d}V.$$

$\boldsymbol{u}_k$ denotes the velocity of species k. $\dot{\omega}_k$ is the source term of species k due to chemical reactions. The velocity of the flow $\boldsymbol{u}$ is defined as a mass average of the N species velocities:

$$\boldsymbol{u} = \sum_{k=1}^{N} Y_k\boldsymbol{u_k}.$$

The difference between the species velocity and the flow velocity is the diffusion velocity:

$$\boldsymbol{U_k} = \boldsymbol{u_k} - \boldsymbol{u}.$$

The species conservation equation can be formulated in such a way that a diffusion term is more obvious:

$$\frac{\partial}{\partial t}\int_\Omega \rho Y_k \,\mathrm{d}V + \oint_{\partial\Omega} \rho Y_k \boldsymbol{u}\cdot\boldsymbol{n}\,\mathrm{d}S \;=\; -\oint_{\partial\Omega} \boldsymbol{J_k}\cdot\boldsymbol{n}\,\mathrm{d}S + \int_\Omega \dot{\omega}_k \,\mathrm{d}V, \tag{2.3}$$

with J_k the diffusion flux of species k:

$$\boldsymbol{J_k} = \rho Y_k \boldsymbol{U_k}.$$

The differential form of equation (2.3) reads:

$$\frac{\partial \rho Y_k}{\partial t} + \frac{\partial \rho Y_k u_i}{\partial x_i} \;=\; -\frac{\partial J_{k,i}}{\partial x_i} + \dot{\omega}_k. \tag{2.4}$$

Since, by definition,

$$\sum_{k=1}^{N} Y_k = 1$$

and

$$\sum_{k=1}^{N} \dot{\omega}_k = 0,$$

the summation of all N species equations (2.4) must yield the continuity equation (2.2), so that

$$\sum_{k=1}^{N} \boldsymbol{J_k} = 0.$$

2.1.3 Conservation of Momentum

This conservation law is based on Newton's second law, stating that the change in momentum in a control volume is due to the forces that act on that volume:

$$\frac{\partial}{\partial t}\int_\Omega \rho \boldsymbol{u}\,\mathrm{d}V + \oint_{\partial\Omega} \rho \boldsymbol{u}\boldsymbol{u}\cdot\boldsymbol{n}\,\mathrm{d}S = -\oint_{\partial\Omega} p\boldsymbol{n}\,\mathrm{d}S + \oint_{\partial\Omega} \bar{\bar{\tau}}\cdot\boldsymbol{n}\,\mathrm{d}S + \int_\Omega \rho\boldsymbol{g}\,\mathrm{d}V + \int_\Omega \rho\sum_{k=1}^{N} Y_k \boldsymbol{f_k}\,\mathrm{d}V, \tag{2.5}$$

where we included the gravity as an external force, acting on the control volume. $\boldsymbol{f_k}$ is the volume force acting on species k. The normal (pressure) and tangential (viscous shear) forces, acting on the boundary surface, are internal forces from the surrounding fluid. The differential equation reads:

$$\frac{\partial \rho u_j}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_i} = -\frac{\partial p}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_i} + \rho g \delta_{j3} + \rho\sum_{k=1}^{N} Y_k f_{k,j}. \tag{2.6}$$

Without lack of generality, we assume that the gravity is aligned with the downward pointing third axis.

2.1.4 Conservation of Energy

According to the first law of thermodynamics, energy cannot be created, nor destroyed. Consequently, a balance exists between the rate of accumulation of total energy in a control volume and heat added to the volume and work exerted on the volume. Heat is added by convection, conduction and thermal radiation; work is exerted by the pressure, the viscous and body forces:

$$\frac{\partial}{\partial t}\int_\Omega \rho E\,\mathrm{d}V + \oint_{\partial\Omega} \rho E\boldsymbol{u}\cdot\boldsymbol{n}\,\mathrm{d}S = -\oint_{\partial\Omega} p\boldsymbol{u}\cdot\boldsymbol{n}\,\mathrm{d}S + \oint_{\partial\Omega} (\bar{\bar{\tau}}\cdot\boldsymbol{u})\cdot\boldsymbol{n}\,\mathrm{d}S - \oint_{\partial\Omega} \boldsymbol{q}\cdot\boldsymbol{n}\,\mathrm{d}S + \int_\Omega \dot{\mathcal{Q}}\,\mathrm{d}V + \int_\Omega \rho\boldsymbol{g}\cdot\boldsymbol{u}\,\mathrm{d}V + \int_\Omega \rho\sum_{k=1}^{N} Y_k f_{k,j}\left(u_j + U_{k,j}\right)\,\mathrm{d}V, \tag{2.7}$$

or, in differential form with the pressure lumped into the total enthalpy:

$$\frac{\partial \rho E}{\partial t} + \frac{\partial \rho u_i H}{\partial x_i} = \frac{\partial\left(\tau_{ij}u_j\right)}{\partial x_i} - \frac{\partial q_i}{\partial x_i} + \dot{\mathcal{Q}} + \rho g u_3 + \rho\sum_{k=1}^{N} Y_k f_{k,j}\left(u_j + U_{k,j}\right). \tag{2.8}$$

$\dot{\mathcal{Q}}$ is a heat source and $\boldsymbol{q}$ is an energy flux containing heat diffusion, species enthalpy diffusion and the so-called *Dufour* effect.

For a good understanding of energy and enthalpy in multi-component fluids, we recall the basic thermodynamic definitions:

$$\begin{aligned} H &= E + \frac{p}{\rho}, \\ H &= h + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}, \\ E &= e + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}, \end{aligned}$$

with E the total specific energy, H the total specific enthalpy, e the specific internal energy and h the specific (static) enthalpy. Both energy and enthalpy consist of two parts: a sensible part, related to the temperature of the fluid mixture, and a chemical part, related to the formation energy or enthalpy of the chemical species of which the mixture consists. We come back to this point in the next section.

2.2 Constitutive Laws

In the above section, we determined $N + 4$ independent equations. However, more unknown variables appear in the equations. In this section, we give additional relationships, depending on the fluid itself, so that the system becomes closed, i.e. there are as many equations as unknowns. Note that the relationships expressed in this section are specific for commonly encountered fluids in combusting flows and closely mimmic reality. They, however, are not restringent for the applicability of the algorithm developed in this thesis, such that the use of other constitutive laws is possible.

2.2.1 Ideal Gas Law

For an ideal gas, density and temperature are related through the equation of state:

$$\rho = \frac{p}{RT}.$$

In this equation, R is the gas constant of the mixture, given by

$$R = \sum_{k=1}^{N} Y_k R_k,$$

with

$$R_k = \frac{\mathcal{R}}{W_k},$$

where $\mathcal{R}$ is the universal gas constant and W_k the molecular weight of species k. According to the equation of state, the density of a mixture is then function of pressure, temperature and the composition of the mixture:

$$\rho = \frac{p}{\mathcal{R}T \sum_{k=1}^{N} \frac{Y_k}{W_k}}. \tag{2.9}$$

2.2.2 Caloric Equation of State

The static enthalpy per unit of mass is defined as a mass average of the individual species specific enthalpies:

$$h = \sum_{k=1}^{N} Y_k h_k.$$

As mentioned above, in multi-component flows, the static enthalpy of a chemical species consists of two parts: sensible or thermal enthalpy and latent or chemical enthalpy:

$$h_k = \int_{T_{\mathrm{ref}}}^{T} c_{p,k}\,\mathrm{d}T + \Delta h^0_{f,k}, \tag{2.10}$$

with $\Delta h^0_{f,k}$ the specific enthalpy of formation of species k at temperature T_{ref}. The specific heat and the specific enthalpy of formation of the mixture are then given by:

$$c_p = \sum_{k=1}^{N} Y_k c_{p,k} \tag{2.11}$$

and

$$\Delta h^0_f = \sum_{k=1}^{N} Y_k \Delta h^0_{f,k}. \tag{2.12}$$

2.2.3 Diffusion Flux

The diffusion flux is related to the diffusion velocities through:

$$\boldsymbol{J_k} = \rho Y_k \boldsymbol{U_k}.$$

In their most general formulation, the diffusion velocities are the solution of the system [35]

$$\begin{aligned}\frac{\partial X_k}{\partial x_i} &= \sum_{l=1}^{N} \frac{X_k X_l}{D_{kl}} \left(U_{l,i} - U_{k,i}\right) \\ &+ \left(Y_k - X_k\right) \frac{1}{p} \frac{\partial p}{\partial x_i} \\ &+ \frac{\rho}{p} \sum_{l=1}^{N} Y_k Y_l \left(f_{k,i} - f_{l,i}\right) \\ &+ \sum_{l=1}^{N} \frac{X_k X_l}{\rho D_{kl}} \left(\frac{D_l^T}{Y_l} - \frac{D_k^T}{Y_k}\right) \frac{1}{T} \frac{\partial T}{\partial x_i},\end{aligned} \tag{2.13}$$

where $D_{kl} = D_{lk}$ is the binary diffusion coefficient of species k into species l, D_k^T is the thermal diffusion coefficient of species k and X_k is the mole fraction of species k:

$$X_k = \frac{Y_k / W_k}{\sum_{l=1}^{N} Y_l / W_l}.$$

Eq. (2.13) expresses that concentration gradients are due to differences in diffusion velocities, pressure gradients, differences in body forces and temperature gradients (the *Soret* effect). Solving the system is a tedious task and mostly a simplified approach based on Fick's law is used:

$$J_{k,i} = -\rho D_k \frac{\partial Y_k}{\partial x_i}, \tag{2.14}$$

with D_k the molecular diffusion coefficient of species k in the mixture, which can be approximated to zeroth order by the Hirschfelder-Curtiss approximation:

$$D_k = \frac{1 - Y_k}{\sum_{l=1; l \neq k}^{N} X_l / D_{kl}}.$$

2.2.4 Chemical Source Term

The chemical source term (reaction rate) $\dot{\omega}_k$ has contributions from every reaction that can appear

$$\dot{\omega}_k = \sum_{j=1}^{M} \dot{\omega}_{k,j},$$

if there are M possible reactions:

$$\sum_{l=1}^{N} \nu'_{l,j} \mathcal{M}_l \rightleftharpoons \sum_{l=1}^{N} \nu''_{l,j} \mathcal{M}_l.$$

The source term of species k for reaction j is given as a function of species concentrations and the forward and backward reaction rate coefficients $k_{f,j}$ and $k_{b,j}$:

$$\dot{\omega}_{k,j} = W_k \left(\nu''_{k,j} - \nu'_{k,j}\right) \left(k_{f,j} \prod_{l=1}^{N} \left(\frac{\rho Y_l}{W_l}\right)^{\nu'_{l,j}} - k_{b,j} \prod_{l=1}^{N} \left(\frac{\rho Y_l}{W_l}\right)^{\nu''_{l,j}} \right). \tag{2.15}$$

The reaction rate coefficients are Arrhenius expressions, with activation energy $E_{A,j}$ and pre-exponential factor $A_j T^{\beta_j}$:

$$k_j = A_j T^{\beta_j} \exp\left(-\frac{E_{A,j}}{\mathcal{R}T}\right). \tag{2.16}$$

2.2.5 Viscous Stress Tensor

For a Newtonian fluid, the shear stresses are given by the relationship:

$$\tau_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu \frac{\partial u_k}{\partial x_k} \delta_{ij}, \tag{2.17}$$

with μ the dynamic viscosity.

2.2.6 Energy Flux

The energy flux contains heat diffusion, species enthalpy diffusion and an additional heat flux originating from species concentration gradients (the *Dufour* effect):

$$q_i = -\lambda \frac{\partial T}{\partial x_i} + \sum_{k=1}^{N} \rho Y_k h_k U_{k,i} + \mathcal{R} T \sum_{k=1}^{N} \sum_{l=1}^{N} \frac{X_l D_k^T}{W_k D_{kl}} \left(U_{k,i} - U_{l,i} \right), \tag{2.18}$$

with λ the thermal conductivity.

2.3 Non-Dimensional Equations

In the field of Fluid Mechanics, the equations are normally written in dimensionless formulation. As such, all variables are expressed relatively to a reference value, such that all dimensionless variables have the same order of magnitude. As a result of the non-dimensional formulation, several dimensionless groups appear. In certain cases, some groups can be extremely large or infinitely small. As a result certain terms in the equations can be neglected.

The major goal of this section is to derive the low Mach number equations. Therefore, the non-dimensional formulation is not carried out to the level of the chemical reaction terms, but up to the level that the low-Mach number equations can be derived.

2.3.1 Dimensionless Variables

As reference variables, we choose a reference length L, density ρ_∞, velocity u_∞, pressure p_∞, viscosity μ_∞, thermal conductivity λ_∞, diffusivity $D_{k,\infty}$, binary diffusion coefficient $D_{kl,\infty}$, thermal diffusion coefficient $D^T_{k,\infty}$, molecular weight W_∞, specific heat at constant pressure $c_{p,\infty}$, external force f_∞ and the reference chemical reaction rate coefficients $k_{f,j,\infty}$ and $k_{b,j,\infty}$. Using these values, all variables can be non-dimensionalised:

$$\begin{aligned} \hat{x}_i &= \frac{x_i}{L}, \\ \hat{t} &= \frac{t}{L/u_\infty}, \\ \hat{W}_k &= \frac{W_k}{W_\infty}, \end{aligned}$$

$$
\begin{aligned}
\hat{\rho} &= \frac{\rho}{\rho_\infty}, \\
\hat{u}_i &= \frac{u_i}{u_\infty}, \\
\hat{U}_{k,i} &= \frac{U_{k,i}}{D_{k,\infty}/L}, \\
\hat{p} &= \frac{p}{p_\infty}, \\
\hat{T} &= \frac{T}{p_\infty W_\infty/\mathcal{R}\rho_\infty}, \\
\hat{E} &= \frac{E}{p_\infty/\rho_\infty}, \\
\hat{H} &= \frac{H}{p_\infty/\rho_\infty}, \\
\hat{c}_p &= \frac{c_p}{c_{p,\infty}}, \\
\hat{\mu} &= \frac{\mu}{\mu_\infty}, \\
\hat{\lambda} &= \frac{\lambda}{\lambda_\infty}, \\
\hat{D}_k &= \frac{D_k}{D_{k,\infty}}, \\
\hat{D}_{kl} &= \frac{D_{kl}}{D_{kl,\infty}}, \\
\hat{D}_k^T &= \frac{D_k^T}{D_{k,\infty}^T}, \\
\hat{k}_{f,j} &= \frac{k_{f,j}}{k_{f,j,\infty}}, \\
\hat{k}_{b,j} &= \frac{k_{b,j}}{k_{b,j,\infty}}, \\
\hat{f}_i &= \frac{f_i}{f_\infty}.
\end{aligned}
$$

2.3.2 Dimensionless Groups

The reference variables can be combined in a way that a non-dimensional variable appears. Smart combinations reveal important dimensionless groups, which express how important certain physical effects are with respect to others.

2.3.2a Specific Heat Ratio

The specific heat ratio expresses the ratio of specific heat at constant pressure to specific heat at constant volume. It is also called the adiabatic exponent and is defined as:

$$\gamma = \frac{c_{p,\infty}}{c_{v,\infty}} = \frac{c_{p,\infty}}{c_{p,\infty} - \mathcal{R}/W_\infty}. \tag{2.19}$$

2.3.2b Mach Number

The Mach number is defined as the flow velocity, divided by the speed of sound in that flow. It is named after Austrian physicist and philosopher Ernst Mach (1838-1916). It is defined as:

$$\mathrm{M} = \frac{u_\infty}{\sqrt{\gamma p_\infty / \rho_\infty}}. \tag{2.20}$$

2.3.2c Froude Number

The Froude number is a dimensionless number comparing inertial and gravitational forces. It is named after William Froude (1810-1879), a British engineer, hydrodynamicist and naval architect. It is defined here as:

$$\mathrm{Fr} = \frac{u_\infty}{\sqrt{gL}}. \tag{2.21}$$

2.3.2d Reynolds Number

The Reynolds number is the ratio of inertial forces to viscous forces and consequently it quantifies the relative importance of these two types of forces for given flow conditions. It is named after Osborne Reynolds (1842-1912), who proposed it in 1883. It is defined as:

$$\mathrm{Re} = \frac{\rho_\infty u_\infty L}{\mu_\infty}. \tag{2.22}$$

2.3.2e Prandtl Number

The Prandtl number is a dimensionless number approximating the ratio of momentum diffusivity (viscosity) and thermal diffusivity. It is named after the German physicist Ludwig Prandtl (1875-1953). It is defined as:

$$\mathrm{Pr} = \frac{\mu_\infty c_{p,\infty}}{\lambda_\infty}. \tag{2.23}$$

2.3.2f Schmidt Number

The Schmidt number is a dimensionless number defined as the ratio of momentum diffusivity (viscosity) and mass diffusivity, and is used to characterise fluid flows in which there are simultaneous momentum and mass diffusion convection processes. It was named after the German engineer Ernst Heinrich Wilhelm Schmidt (1892-1975). It is defined as:

$$\mathrm{Sc}_k = \frac{\mu_\infty}{\rho_\infty D_{k,\infty}}. \tag{2.24}$$

2.3.2g Lewis Number

The Lewis number is a dimensionless number defined as the ratio of thermal diffusivity to mass diffusivity. It is used to characterise fluid flows where there is simultaneous heat and mass transfer by convection. It is named after Warren Kendall Lewis (1882-1975), who was the first head of the Chemical Engineering Department at MIT. The Lewis number of species k is defined as:

$$\mathrm{Le}_k = \frac{\lambda_\infty}{c_{p,\infty}\rho_\infty D_{k,\infty}}. \tag{2.25}$$

The Lewis number can also be expressed in terms of the Schmidt number and the Prandtl number :

$$\mathrm{Le}_k = \frac{\mathrm{Sc}_k}{\mathrm{Pr}}.$$

In the following, the Lewis number of species k into species l will also be needed:

$$\mathrm{Le}_{kl} = \frac{\lambda_\infty}{c_{p,\infty}\rho_\infty D_{kl,\infty}}. \tag{2.26}$$

2.3.2h Dufour Number

The Dufour number is a dimensionless number expressing the relative importance of the thermal diffusion, compared to mass diffusion due to temperature gradients. Its primary field of interest is situated in the domain of convection through porous media, from which we adopted the formulation:

$$\mathrm{Du}_k = \frac{\gamma - 1}{\gamma} \frac{\mu_\infty}{D^T_{k,\infty}}. \tag{2.27}$$

2.3.2i Damköhler Number

The Damköhler numbers are dimensionless numbers used in chemical engineering to relate the chemical reaction timescales to other phenomena occurring in a system, such as the timescale of the flow motion. It is named after German chemist Gerhard Damköhler (1908-1944). The Damköhler Number for the j-th forward reaction is defined as:

$$\mathrm{Da}_{f,j} = \frac{k_{f,j,\infty} L}{u_\infty} \left(\frac{\rho_\infty}{W_\infty} \right)^{n'_j - 1}, \tag{2.28}$$

with n'_j the order of the reaction, defined as:

$$n'_j = \sum_{l=1}^{N} \nu'_{l,j}.$$

We can do the same for the backward reaction:

$$\mathrm{Da}_{b,j} = \frac{k_{b,j,\infty} L}{u_\infty} \left(\frac{\rho_\infty}{W_\infty} \right)^{n''_j - 1}, \tag{2.29}$$

with

$$n''_j = \sum_{l=1}^{N} \nu''_{l,j}.$$

2.3.2j External Forces Non-Dimensional Group

This dimensionless number is comparable to the Froude number and compares inertial and external forces. It is defined as:

$$\mathrm{Ex} = \frac{u_\infty}{\sqrt{f_\infty L}}. \tag{2.30}$$

2.3.3 Non-Dimensional Equations

A combination of all previous equations, gives us the governing equations in their most global formulation:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} &= 0; \\
\frac{\partial \rho Y_k}{\partial t} + \frac{\partial \rho Y_k u_i}{\partial x_i} &= \frac{\partial}{\partial x_i}\left(\rho D_k \frac{\partial Y_k}{\partial x_i}\right) \\
&\quad + \sum_{j=1}^{M} W_k \left(\nu''_{k,j} - \nu'_{k,j}\right) \left(k_{f,j} \prod_{l=1}^{N} \left(\frac{\rho Y_l}{W_l}\right)^{\nu'_{l,j}} \right. \\
&\qquad \left. - k_{b,j} \prod_{l=1}^{N} \left(\frac{\rho Y_l}{W_l}\right)^{\nu''_{l,j}} \right); \\
\frac{\partial \rho u_j}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_i} &= -\frac{\partial p}{\partial x_j} \\
&\quad + \frac{\partial}{\partial x_i}\left[\mu\left(\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right)\right] \\
&\quad + \rho g \delta_{j3} \\
&\quad + \rho \sum_{k=1}^{N} Y_k f_{k,j};
\end{aligned}$$

$$\begin{aligned}\frac{\partial(\rho E)}{\partial t}+\frac{\partial(\rho u_i H)}{\partial x_i} &= \frac{\partial}{\partial x_i}\left[\mu\left(\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)-\frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right)u_j\right]\\ &+\frac{\partial}{\partial x_i}\left(\lambda\frac{\partial T}{\partial x_i}\right)\\ &-\frac{\partial}{\partial x_i}\left(\sum_{k=1}^{N}\rho Y_k h_k U_{k,i}\right)\\ &-\frac{\partial}{\partial x_i}\left(\mathcal{R}T\sum_{k=1}^{N}\sum_{l=1}^{N}\frac{X_l D_k^T}{W_k D_{kl}}\left(U_{k,i}-U_{l,i}\right)\right)\\ &+\dot{\mathcal{Q}}\\ &+\rho g u_3\\ &+\rho\sum_{k=1}^{N}Y_k f_{k,j}\left(u_j+U_{k,j}\right).\end{aligned}$$

Replacing all the variables by the non-dimensional ones, the following non-dimensional equations are obtained:

$$\frac{\partial\hat{\rho}}{\partial\hat{t}}+\frac{\partial\hat{\rho}\hat{u}_i}{\partial\hat{x}_i} = 0; \tag{2.31}$$

$$\begin{aligned}\frac{\partial\hat{\rho}Y_k}{\partial\hat{t}}+\frac{\partial\hat{\rho}Y_k\hat{u}_i}{\partial\hat{x}_i} &= \frac{1}{\mathrm{Le}_k\mathrm{PrRe}}\frac{\partial}{\partial\hat{x}_i}\left(\hat{\rho}\hat{D}_k\frac{\partial Y_k}{\partial\hat{x}_i}\right)\\ &+\sum_{j=1}^{M}\hat{W}_k\left(\nu''_{k,j}-\nu'_{k,j}\right)\left(\mathrm{Da}_{f,j}\hat{k}_{f,j}\prod_{l=1}^{N}\left(\frac{\hat{\rho}Y_l}{\hat{W}_l}\right)^{\nu'_{l,j}}\right.\\ &\left.-\mathrm{Da}_{b,j}\hat{k}_{b,j}\prod_{l=1}^{N}\left(\frac{\hat{\rho}Y_l}{\hat{W}_l}\right)^{\nu''_{l,j}}\right);\end{aligned} \tag{2.32}$$

$$\begin{aligned}\frac{\partial\hat{\rho}\hat{u}_j}{\partial\hat{t}}+\frac{\partial\hat{\rho}\hat{u}_i\hat{u}_j}{\partial\hat{x}_i} &= -\frac{1}{\gamma\mathrm{M}^2}\frac{\partial\hat{p}}{\partial\hat{x}_j}\\ &+\frac{1}{\mathrm{Re}}\frac{\partial}{\partial\hat{x}_i}\left[\hat{\mu}\left(\left(\frac{\partial\hat{u}_i}{\partial\hat{x}_j}+\frac{\partial\hat{u}_j}{\partial\hat{x}_i}\right)-\frac{2}{3}\frac{\partial\hat{u}_k}{\partial\hat{x}_k}\delta_{ij}\right)\right]\\ &+\frac{1}{\mathrm{Fr}^2}\hat{\rho}\delta_{j3}\\ &+\frac{1}{\mathrm{Ex}^2}\hat{\rho}\sum_{k=1}^{N}Y_k\hat{f}_{k,j};\end{aligned} \tag{2.33}$$

$$\begin{aligned}\frac{\partial\hat{\rho}\hat{E}}{\partial\hat{t}}+\frac{\partial\hat{\rho}\hat{u}_i\hat{H}}{\partial\hat{x}_i} &= \frac{\gamma\mathrm{M}^2}{\mathrm{Re}}\frac{\partial}{\partial\hat{x}_i}\left[\hat{\mu}\left(\left(\frac{\partial\hat{u}_i}{\partial\hat{x}_j}+\frac{\partial\hat{u}_j}{\partial\hat{x}_i}\right)-\frac{2}{3}\frac{\partial\hat{u}_k}{\partial\hat{x}_k}\delta_{ij}\right)\hat{u}_j\right]\\ &+\frac{\gamma}{\gamma-1}\frac{1}{\mathrm{PrRe}}\frac{\partial}{\partial\hat{x}_i}\left(\hat{\lambda}\frac{\partial\hat{T}}{\partial\hat{x}_i}\right)\\ &-\sum_{k=1}^{N}\frac{1}{\mathrm{Le}_k\mathrm{PrRe}}\frac{\partial}{\partial\hat{x}_i}\left(\hat{\rho}\hat{Y}_k\hat{h}_k\hat{U}_{k,i}\right)\\ &-\hat{T}\sum_{k=1}^{N}\sum_{l=1}^{N}\frac{\gamma/(\gamma-1)}{\mathrm{Du}_k\mathrm{Le}_{kl}\mathrm{Re}}\frac{\partial}{\partial\hat{x}_i}\left(\frac{X_l\hat{D}_k^T}{\hat{W}_k\hat{D}_{kl}}\left(\mathrm{Le}_k\hat{U}_{k,i}-\mathrm{Le}_l\hat{U}_{l,i}\right)\right)\\ &+\hat{\dot{\mathcal{Q}}}\\ &+\frac{\gamma\mathrm{M}^2}{\mathrm{Fr}^2}\hat{\rho}\hat{u}_3\\ &+\frac{\gamma\mathrm{M}^2}{\mathrm{Ex}^2}\hat{\rho}\sum_{k=1}^{N}Y_k\hat{f}_{k,j}\left(\hat{u}_j+\frac{1}{\mathrm{Le}_k\mathrm{PrRe}}\hat{U}_{k,j}\right),\end{aligned}\tag{2.34}$$

if the heat source is made non-dimensional with $\dot{\mathcal{Q}}_\infty = u_\infty p_\infty/L$. Following equations complete the system:

$$\hat{H} = \hat{h}+\frac{1}{2}\frac{1}{\gamma\mathrm{M}^2}u_k u_k;\tag{2.35}$$

$$\hat{E} = \hat{H}-\frac{\hat{p}}{\hat{\rho}};\tag{2.36}$$

$$\hat{\rho}=\frac{\hat{p}}{\hat{T}\sum_{k=1}^{N}\frac{Y_k}{\hat{W}_k}};\tag{2.37}$$

$$\hat{h}=\frac{\gamma}{\gamma-1}\int_{\hat{T}_{\mathrm{ref}}}^{\hat{T}}\hat{c}_p\,\mathrm{d}\hat{T}+\Delta\hat{h}_f^0.\tag{2.38}$$

From now on, we work with these equations, but skip the ˆ notation.

2.4 Simplified Flow Cases

2.4.1 Low Mach Number Flow

Various flow regimes encountered in industrial devices are of low speed nature: the arising velocities are much smaller than the speed of sound. Such flows are characterised by a Mach number that is at least an order of magnitude smaller than unity. Also in many combustion applications the Mach number remains low (exceptions are e.g. detonation waves). In this case, the set of equations (2.31)-(2.34) can be simplified [44, 50]. Every variable is expanded in a power series of $\sqrt{\gamma}\mathrm{M}$:

$$\phi = \phi_0 + \sqrt{\gamma}\mathrm{M}\phi_1 + \gamma\mathrm{M}^2\phi_2 + ...$$

and filled out into the equations. The asymptotic limit for $\sqrt{\gamma}\mathrm{M}$ going to zero is taken. For every variable the lowest order term remains in the equations, except for the pressure, which has two parts: a thermodynamic part p_0 and a kinematic part p_2. The second order term of the pressure remains because the momentum equation has two parts: the zero-th order equation, expressing that p_0 is uniform over the domain and the second order equation, that is similar to the original momentum equation. The complete low-Mach number equations for reacting flows read:

$$p_0 = p_0(t);$$

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0; \tag{2.39}$$

$$\begin{aligned}\frac{\partial \rho Y_k}{\partial t} + \frac{\partial \rho Y_k u_i}{\partial x_i} = & \frac{1}{\mathrm{Le}_k \mathrm{PrRe}} \frac{\partial}{\partial x_i}\left(\rho D_k \frac{\partial Y_k}{\partial x_i}\right) \\ & + \sum_{j=1}^{M} W_k \left(\nu''_{k,j} - \nu'_{k,j}\right)\left(\mathrm{Da}_{f,j} k_{f,j} \prod_{l=1}^{N}\left(\frac{\rho Y_l}{W_l}\right)^{\nu'_{l,j}}\right. \\ & \left. - \mathrm{Da}_{b,j} k_{b,j} \prod_{l=1}^{N}\left(\frac{\rho Y_l}{W_l}\right)^{\nu''_{l,j}}\right);\end{aligned} \tag{2.40}$$

$$\begin{aligned}\frac{\partial \rho u_j}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_i} &= -\frac{\partial p_2}{\partial x_j} \\ &+ \frac{1}{\mathrm{Re}} \frac{\partial}{\partial x_i} \left[\mu \left(\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \right] \\ &+ \frac{1}{\mathrm{Fr}^2} \rho \delta_{j3} \\ &+ \frac{1}{\mathrm{Ex}^2} \rho \sum_{k=1}^{N} Y_k f_{k,j}; \end{aligned} \tag{2.41}$$

$$\begin{aligned}\frac{\partial \rho h}{\partial t} + \frac{\partial \rho u_i h}{\partial x_i} &= \frac{\mathrm{d}p_0}{\mathrm{d}t} \\ &+ \frac{\gamma}{\gamma - 1} \frac{1}{\mathrm{PrRe}} \frac{\partial}{\partial x_i} \left(\lambda \frac{\partial T}{\partial x_i} \right) \\ &- \sum_{k=1}^{N} \frac{1}{\mathrm{Le}_k \mathrm{PrRe}} \frac{\partial}{\partial x_i} \left(\rho Y_k h_k U_{k,i} \right) \\ &- T \sum_{k=1}^{N} \sum_{l=1}^{N} \frac{\gamma/(\gamma-1)}{\mathrm{Du}_k \mathrm{Le}_{kl} \mathrm{Re}} \frac{\partial}{\partial x_i} \left(\frac{X_l D_k^T}{W_k D_{kl}} \left(\mathrm{Le}_k U_{k,i} - \mathrm{Le}_l U_{l,i} \right) \right) \\ &+ \dot{\mathcal{Q}}, \end{aligned} \tag{2.42}$$

with the zeroth-order equations of state:

$$\rho = \frac{p_0}{T \sum_{k=1}^{N} \frac{Y_k}{W_k}} \tag{2.43}$$

and

$$h = \sum_{k=1}^{N} Y_k h_k, \tag{2.44}$$

with

$$\begin{aligned} h_k &= \frac{\gamma}{\gamma - 1} \int_{T_{\mathrm{ref}}}^{T} c_{p,k} \, \mathrm{d}T + \Delta h_{f,k}^0 \\ &= \frac{\gamma}{\gamma - 1} c_{p,k}^m \left(T - T_{\mathrm{ref}} \right) + \Delta h_{f,k}^0, \end{aligned}$$

with $c_{p,k}^m$ the mean specific heat at constant pressure for species k in the temperature interval $[T_{\text{ref}},\ T]$. Note that the subscript 0 is dropped for all variables, except for the pressure to distinguish between thermodynamic and kinematic pressure.

2.4.2 Two Fluid Flow: Conserved Scalars

Although the Low Mach number equations are a simplification of the original equations, still, there are terms that require complicated calculations and detailed information about e.g. the chemistry. Therefore, the method of conserved scalars is introduced. Conserved scalars are scalars that are not affected by the chemical reactions. In contrast with chemical species (molecules) mass fraction, chemical elements mass fraction is not altered by the chemical reaction.

2.4.2a Chemical Elements Transport Equation

In the mixture, N species are considered, consisting of N' chemical elements. The mass fraction of chemical element k' in species k is denoted as $z_{k'k}$. The mass fraction of element k' in the mixture is then given by:

$$Z_{k'} = \sum_{k=1}^{N} Y_k z_{k'k},$$

so that a transport equation for elements mass fraction can be formed from a combination of the species elements conservation equations:

$$\begin{aligned}\sum_{k=1}^{N} z_{k'k}\left[\frac{\partial \rho Y_k}{\partial t} + \frac{\partial \rho Y_k u_i}{\partial x_i}\right] &= \sum_{k=1}^{N} z_{k'k}\left[\frac{1}{\text{Le}_k \text{PrRe}}\frac{\partial}{\partial x_i}\left(\rho D_k \frac{\partial Y_k}{\partial x_i}\right)\right] \\ \Leftrightarrow \frac{\partial \rho Z_{k'}}{\partial t} + \frac{\partial \rho Z_{k'} u_i}{\partial x_i} &= \frac{1}{\text{PrRe}}\sum_{k=1}^{N} z_{k'k}\left[\frac{1}{\text{Le}_k}\frac{\partial}{\partial x_i}\left(\rho D_k \frac{\partial Y_k}{\partial x_i}\right)\right]. \end{aligned} \tag{2.45}$$

Because $Z_{k'}$ is a conserved scalar, no chemical source term appears in eq. (2.45):

$$\sum_{k=1}^{N} z_{k'k}\sum_{j=1}^{M} W_k\left(\nu''_{k,j} - \nu'_{k,j}\right)\left(\text{Da}_{f,j} k_{f,j}\prod_{l=1}^{N}\left(\frac{\rho Y_l}{W_l}\right)^{\nu'_{l,j}} - \text{Da}_{b,j} k_{b,j}\prod_{l=1}^{N}\left(\frac{\rho Y_l}{W_l}\right)^{\nu''_{l,j}}\right) = 0.$$

In general, the right hand side of eq. (2.45) cannot be written as a function of $Z_{k'}$ only. If, however, Fick's law (2.14) is valid with equal diffusivities for all species[1], i.e.

$$D_k = D, \quad k = 1, \ ... \ N,$$

it does become possible:

$$\frac{\partial \rho Z_{k'}}{\partial t} + \frac{\partial \rho Z_{k'} u_i}{\partial x_i} = \frac{1}{\mathrm{LePrRe}} \frac{\partial}{\partial x_i} \left(\rho D \frac{\partial Z_{k'}}{\partial x_i} \right). \tag{2.46}$$

2.4.2b Mixture Fraction Transport Equation

Consider now the case where fuel and oxidizer are two separate flows, as is typical for non-premixed combustion. Any element with a different mass fraction in fuel and oxidizer can then be used as an indicator for the degree of mixing of the two flows. The fuel is denoted by F, the oxidizer by O. Consider element i, which has a different mass fraction $Z_{i,F}$ in the fuel and $Z_{i,O}$ in the oxidizer. Then the normalised conserved scalar ξ_i expresses the fraction of mass at a certain position originating from the fuel:

$$\xi_i \ = \ \frac{Z_i - Z_{i,O}}{Z_{i,F} - Z_{i,O}}. \tag{2.47}$$

Note that, by definition, $\xi_{k'}$ equals zero in the oxidizer stream and 1 in the fuel stream. The transport equation for $\xi_{k'}$ is immediately derived from eqs. (2.46) and (2.47):

$$\frac{\partial \rho \xi_{k'}}{\partial t} + \frac{\partial \rho \xi_{k'} u_i}{\partial x_i} = \frac{1}{\mathrm{LePrRe}} \frac{\partial}{\partial x_i} \left(\rho D \frac{\partial \xi_{k'}}{\partial x_i} \right).$$

Since all $\xi_{k'}$ obey the same transport equation, with the same boundary and initial conditions, there is only one transport equation left:

$$\frac{\partial \rho \xi}{\partial t} + \frac{\partial \rho \xi u_i}{\partial x_i} = \frac{1}{\mathrm{LePrRe}} \frac{\partial}{\partial x_i} \left(\rho D \frac{\partial \xi}{\partial x_i} \right). \tag{2.48}$$

[1] The assumption of equal diffusivities is common practice in the field of turbulent combustion, where also contributions appear from the unresolved part (see chapter 3). If hydrogen flames are considered, this assumption has to be used with care, because species like H_2 or H diffuse much faster than heavier molecules.

The variable ξ is called the mixture fraction.

The main assumptions, leading to eq. (2.48), are the equality of the diffusivities and the existence of only two streams with different mixture compositions at the inlet.

2.4.2c Static Enthalpy as a Conserved Scalar

The energy flux in the low Mach number energy conservation equation can be expressed in terms of static enthalpy. Therefore, from eq. (2.44), we express the gradient of static enthalpy as:

$$\begin{aligned}\frac{\partial h}{\partial x_i} &= \sum_{k=1}^{N} \frac{\partial Y_k h_k}{\partial x_i} \\ &= \sum_{k=1}^{N} \frac{\partial Y_k \Delta h^0_{f,k}}{\partial x_i} + \frac{\gamma}{\gamma-1} \sum_{k=1}^{N} \frac{\partial c^m_{p,k} Y_k T}{\partial x_i} \\ &= \sum_{k=1}^{N} \Delta h^0_{f,k} \frac{\partial Y_k}{\partial x_i} + \frac{\gamma}{\gamma-1} \frac{\partial c^m_p T}{\partial x_i},\end{aligned}$$

so that the enthalpy equation, assuming constant c^m_p, becomes:

$$\begin{aligned}\frac{\partial \rho h}{\partial t} + \frac{\partial \rho u_i h}{\partial x_i} &= \frac{\mathrm{d}p_0}{\mathrm{d}t} \\ &+ \frac{1}{\mathrm{PrRe}} \frac{\partial}{\partial x_i} \left[\frac{\lambda}{c^m_p} \left(\frac{\partial h}{\partial x_i} - \sum_{k=1}^{N} \left(1 - \frac{\rho D_k c^m_p}{\lambda \mathrm{Le}_k} \right) \Delta h^0_{f,k} \frac{\partial Y_k}{\partial x_i} \right) \right] \\ &+ \frac{1}{\mathrm{PrRe}} \frac{\partial}{\partial x_i} \left[\sum_{k=1}^{N} \frac{\rho c^m_{p,k}}{Le_k} \left(T - T_{\mathrm{ref}} \right) Y_k U_{k,i} \right] \\ &- \hat{T} \sum_{k=1}^{N} \sum_{l=1}^{N} \frac{\gamma/(\gamma-1)}{\mathrm{Du}_k \mathrm{Le}_{kl} \mathrm{Re}} \frac{\partial}{\partial \hat{x}_i} \left(\frac{X_l \hat{D}^T_k}{\hat{W}_k \hat{D}_{kl}} \left(\mathrm{Le}_k \hat{U}_{k,i} - \mathrm{Le}_l \hat{U}_{l,i} \right) \right) \\ &+ \dot{\mathcal{Q}}.\end{aligned}$$

Under the following assumptions, the above low Mach equation can be further simplified:

1. constant thermodynamic pressure in time;
2. negligable Dufour-effect ($\mathrm{Du}_k \to \infty$);

3. equal diffusivities ($D_k = D$);
4. equal and constant specific heats ($c_{p,k} = c_p$);
5. unity Lewis number ($\mathrm{Le}_k = \mathrm{Le} = 1$);
6. absense of radiation and other heat sources ($\dot{\mathcal{Q}} = 0$)
7. variation of specific heat, diffusivity, density and conductivity is such that the combination $\lambda/c_p^m \rho D$ is always unity;

and the static enthalpy transport equation becomes:

$$\frac{\partial \rho h}{\partial t} + \frac{\partial \rho u_i h}{\partial x_i} = \frac{1}{\mathrm{PrRe}} \frac{\partial}{\partial x_i} \left(\frac{\lambda}{c_p^m} \frac{\partial h}{\partial x_i} \right). \tag{2.49}$$

Because of assumption 7, $\lambda/c_p^m = \rho D$, equation (2.49) is exactly the same as equation (2.48). If the inlet streams have a fixed enthalpy and walls are adiabatic, the boundary conditions are identical to the ones for the mixture fraction, if the normalised quantity

$$h' = \frac{h - h_O}{h_F - h_O}$$

is used as variable. This implies that

$$h' = \xi.$$

Thus, the energy equation need not be resolved anymore. The static enthalpy can be calculated from the mixture fraction:

$$h = h_F \xi + h_O (1 - \xi).$$

2.4.3 Single Fluid Flow

In case of an inert single fluid flow, the equations extremely reduce in complexity. The species conservation equations become superfluous, and the energy equation simplifies into:

$$\frac{\partial \rho h}{\partial t} + \frac{\partial \rho u_i h}{\partial x_i} = \frac{\mathrm{d}p_0}{\mathrm{d}t} + \frac{\gamma}{\gamma - 1} \frac{1}{\mathrm{PrRe}} \frac{\partial}{\partial x_i} \left(\lambda \frac{\partial T}{\partial x_i} \right) + \dot{\mathcal{Q}},$$

with

$$\rho = \frac{p_0}{T}$$

and (with constant c_p)

$$h = \frac{\gamma}{\gamma - 1} T.$$

In combination with the continuity equation, a transport equation for temperature can be derived:

$$\frac{\gamma}{\gamma - 1}\rho\left[\frac{\partial T}{\partial t} + u_i\frac{\partial T}{\partial x_i}\right] - \frac{dp_0}{dt} = \frac{\gamma}{(\gamma - 1)\,\mathrm{RePr}}\frac{\partial}{\partial x_i}\left(\lambda\frac{\partial T}{\partial x_i}\right), \tag{2.50}$$

where the heat source has been omitted. The equation for temperature can also be written as an energy equation in conservative form for the thermodynamic pressure:

$$\frac{dp_0}{dt} + \gamma p_0\frac{\partial u_i}{\partial x_i} = \frac{\gamma}{\mathrm{RePr}}\frac{\partial}{\partial x_i}\left(\lambda\frac{\partial T}{\partial x_i}\right). \tag{2.51}$$

Unless we are dealing with enclosed systems, the thermodynamic pressure is assumed constant in space and time.

2.5 Summary

In the remainder of this thesis, we will work with the following governing equations.

In case of an inert or reacting two fluid flow, the equations we solve are:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} &= 0 \\
\frac{\partial \rho u_j}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_i} &= -\frac{\partial p_2}{\partial x_j} \\
&\quad + \frac{1}{\mathrm{Re}}\frac{\partial}{\partial x_i}\left[\mu\left(\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right)\right] \\
&\quad + \frac{1}{\mathrm{Fr}^2}\rho\delta_{j3} \\
\frac{\partial \rho\xi}{\partial t} + \frac{\partial \rho\xi u_i}{\partial x_i} &= \frac{1}{\mathrm{LePrRe}}\frac{\partial}{\partial x_i}\left(\rho D\frac{\partial \xi}{\partial x_i}\right),
\end{aligned} \tag{2.52}$$

with

$$\rho = \rho\left(\xi\right) \tag{2.53}$$

following from a chemistry model (see next chapter).

In case of an inert single fluid flow, the equations are

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} &= 0 \\
\frac{\partial \rho u_j}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_i} &= -\frac{\partial p_2}{\partial x_j} \\
&\quad + \frac{1}{\mathrm{Re}} \frac{\partial}{\partial x_i} \left[\mu \left(\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) \right] \\
&\quad + \frac{1}{\mathrm{Fr}^2} \rho \delta_{j3} \\
\frac{\gamma}{\gamma - 1} \rho \left[\frac{\partial T}{\partial t} + u_i \frac{\partial T}{\partial x_i} \right] - \frac{dp_0}{dt} &= \frac{\gamma}{(\gamma - 1)\,\mathrm{RePr}} \frac{\partial}{\partial x_i} \left(\lambda \frac{\partial T}{\partial x_i} \right),
\end{aligned} \tag{2.54}$$

with

$$\rho = \frac{p_0}{T}. \tag{2.55}$$

Chapter 3

Model Assumptions

In the previous chapter, we derived the equations that govern the flow. In certain cases, the amount of equations to be actually solved, could drastically be reduced, by assuming for instance all diffusivities to be the same. However, in doing so, we lose information to determine e.g. the chemical composition of a mixture, characterised by a certain value of mixture fraction. A chemical model will provide this information, as described in the following section.

Not only the amount of equations determines the practical ability to simulate a flow. Also the number of grid points needed to discretise the system is of major importance. If the instantaneous equations of chapter 2 were used in a turbulent flow, too many grid points would be needed. The amount of grid points can be reduced by averaging or filtering the equations and adding a turbulence model to close the unknown terms.

Because of the non-linearity of the chemistry, the chemistry and turbulence models cannot be combined by simply putting them together. A special procedure, making use of probability density functions (PDF), is needed to cope with this turbulence-chemistry interaction. This will be addressed in the last section of this chapter.

3.1 Chemistry Modelling

For numerical simulations of reacting flows, the chemical scheme has to be defined. This means that the knowledge of all species, reactions and values of the Arrhenius parameters must be determined before the computation can be carried out. In the combustion community, the CHEMKIN format [32] has become a practical standard. In this formulation, reactions are listed using a prescribed format, along with the values for A_j , β_j and $E_{A,j}$ (eq. (2.16)).

As an indication, the use of such a complete kinetic scheme to describe the chemistry requires about eight species and 40 irreversible elementary reactions for hydrogen/oxygen combustion, typically around 50 species and a few hundred chemical reactions for methane, while more complex fuels like n-decane or cetane require several hundreds of species and several thousands of elementary reactions [27].

As stated before, a transport equation should ideally be solved for each of these species in order to accurately describe the processes occurring during combustion. It is thus clear why very few complete mechanisms are used in practical simulations. The induced computing costs and memory requirements are huge and it is almost impossible to use such complete mechanisms for multi-dimensional simulations. Even in case of hydrogen combustion, the complete hydrogen reaction scheme is hardly ever used, although it is the simplest reaction system ('only' eight species, plus possibly supplementary species and reactions to describe NO_x production).

Researchers dealing with practical applications would in fact like to work at least with methane, more often with natural gas or n-decane, chemicals of high complexity involving roughly 100-1000 chemical intermediate species. In this case complete reaction mechanisms are impossible to use in order to investigate turbulent configurations, since even simple one-dimensional simulations become extremely demanding. Several techniques have therefore been developed to reduce complete mechanisms to a simpler sub-set.

The ultimate goal is to obtain a chemical mechanism that contains fewer species and fewer reactions, but still yields sufficiently accurate results. Chemical mechanisms can be reduced by means of a sensitivity analysis based on reaction rates, by which one investigates which species can be removed from the mechanism without substantial effect on the species that are considered as relevant. This can be followed by a further reduction, based on the quasi-steady state assumption (QSSA). This approach relies on the fact that very reactive species equilibrate with respect to slower species. These rates generally involve different elementary reactions. Using the quasi-steady state assumption, the corresponding intermediate radicals are removed from the original complete mechanism and replaced by implicit relations as functions of the other species.

However, for complex reaction systems, these methods often become cumbersome and demand a detailed understanding of the relevant chemistry. As a consequence new complementary approaches have been derived recently.

The most widely used technique is the ILDM-method, proposed by Maas and Pope [41]. Extensions of this approach are either the FGM-method, proposed by De Goey and Van Oijen [73] or the FPI-method [25]. In these methods intrinsic low-dimensional manifolds (ILDM), flamelet generated manifolds (FGM), or flame prolongation of ILDM (FPI) are identified in the composition space. These manifolds correspond to a description of the complete reaction system by a much smaller number of coordinates (chemical species, mixture fraction, static enthalpy). These manifolds are used in the reacting flow

simulation, where only the coordinates and not all the chemical species have to be solved for. But all the intermediate species are still available throughout the computation by means of algebraic expressions in terms of the coordinates.

For completeness, we add other related dimension-reduction techniques, such as computational single perturbation (CSP) [36], where chemical time scales are separated into fast and slow subspaces; the rate-controlled constrained equilibrium (RCCE) [31]; and the invariant constrained-equilibrium edge pre-image curve method (ICE-PIC) [61], which constructs a trajectory-generated manifold that preserves continuity. One issue of reduction methods or reduced kinetics models is the fact that the reduced chemistry modelling is typically based purely on the analysis of the chemical source terms, neglecting the potential impact of the transport processes on the so-called manifold. The reaction-diffusion manifold approach (REDIM) [11] allows to incorporate the effect of coupling of reaction and diffusion processes. In general, chemistry reduction for turbulent flames is an active research area, that is beyond the subject of the present thesis.

In the extreme case, only the mixture fraction is retained as a coordinate. All species, including the intermediate species, are then derived from the mixture fraction. By this approximation, only one scalar transport equation is solved: the mixture fraction equation, which contains no chemical source term. As a result, finite rate chemistry effects can only be included on a mixing-controlled basis[1]. If ignition and extinction need to be modelled, as is the case in e.g. internal combustion engines, a finite rate chemistry model is needed and the single-coordinate approximation cannot be used. In some cases, extinction and ignition are not important and the use of an infinitely fast chemistry model provides good results. In the following subsections, we look deeper into these models.

3.1.1 Mixing-Controlled Chemistry: Flamelet Approximation

The flamelet equations are unsteady balance equations for temperature and species in mixture fraction space [53]. They result from a coordinate transformation of the Cartesian equations into a coordinate system attached to the stoichiometric surface, with the ξ-direction perpendicular to the stoichiometric plane. If the flame is infinitely thin[2], the species equations (2.4) up to leading order then transform into:

$$\rho\frac{\partial Y_k}{\partial t} = \rho\frac{\chi}{2}\frac{\partial^2 Y_k}{\partial \xi^2} + \dot{\omega}_k,$$

[1] Depending on the mixing to reaction time scale ratio, the chemistry varies from infinitely fast chemistry to inert mixing (frozen chemistry).

[2] In some cases, extinction effects are governed by advection parallel to the stoichiometric surface. The flamelet equation cannot describe such effects.

with χ the scalar dissipation rate, defined by:

$$\chi = \frac{2D}{\mathrm{LePrRe}} \frac{\partial \xi}{\partial x_i} \frac{\partial \xi}{\partial x_i}.$$

A similar equation for the energy equation can be derived. The solution of these equations is defined by its initial and boundary conditions, and by the instantaneous scalar dissipation rate distribution in mixture fraction space. This results in the observation that if unsteady effects are neglected, the diffusion flame is fully determined by the mixture fraction ξ and the scalar dissipation rate χ, yielding

$$Y_i = Y_i(\xi, \chi), \quad T = T(\xi, \chi),$$

which is the solution of the system of the $N + 1$ steady flamelet equations[3]

$$\rho \frac{\chi}{2} \frac{\partial^2 Y_k}{\partial \xi^2} + \dot{\omega}_k = 0$$

and a similar equation for the temperature.

In most implementations, the flamelet approach is used in such a way that the chemistry (flamelet equations) can be solved seperately from the hydrodynamics. The scalar dissipation rate, which is a function of mixture fraction, is then replaced by its value at the stoichiometric plane $\chi_{st} = \chi(\xi_{st})$. These flamelet equations are solved prior to the hydrodynamic simulation. As a function of the two parameters χ_{st} and ξ, a representative look-up table is built.

3.1.2 Mixing-Controlled Chemistry: Infinitely Fast Chemistry

For the purpose of this thesis, namely algorithm development, the mixing-controlled finite rate chemistry, is not considered[4]. Therefore, the still widely used infinitely fast chemistry models are discussed here. Moreover, because of their discontinuities in gradients in mixture fraction space, these models are even more challenging from the algorithmic point of view. If an algorithm is able to handle the discontinuities of idealised models, we expect it to behave at least as good for the other models, that have a smoother behaviour.

[3] Remark that in this equation, some finite rate chemistry is still included because of the appearance of two time scales: the mixing time scale (χ^{-1}) and the chemical time scale (hidden in the chemical source term $\dot{\omega}_k$).

[4] although the developed algorithm can handle these type of models easily.

The infinitely fast chemistry model assumption is a valid approach if the reactions have enough time to reach the model state. This is true when the chemical time scale is much smaller than the characteristic time of the flow, i.e. a high Damköhler number is required. If this is the case, essentially two approaches are possible, depending on whether we consider the reactions to be reversible, so that the chemical equilibrium can be obtained, or that the reactions are irreversible, so that fuel and oxidizer immediately react with instantaneous formation of combustion products.

If infinitely fast chemistry is assumed, the chemical time scale τ_c is much smaller than the diffusive time scale $\tau_\chi = \chi^{-1}$. In this case of high Damköhler numbers, the steady flamelet equations reduce to

$$\dot{\omega}_k = 0, \tag{3.1}$$

so that the influence of the scalar dissipation rate χ is eliminated and species mass fractions and temperature are a function of mixture fraction only:

$$Y_i = Y_i(\xi), \quad T = T(\xi).$$

3.1.2a Equilibrium Chemistry

If equilibrium chemistry is assumed, the reactions are considered to be reversible and reach an equilibrium state, according to the local composition. As a consequence, the equilibrium state is function of the reaction equilibrium constants and species concentration, which are both function of mixture fraction. The equilibrium condition is the solution of the flamelet equation (3.1), and is determined numerically through the equilibrium constants of the appearing reactions. Another manner to derive the chemical equilibrium is through the minimisation of the Gibbs free energy. The reader is referred to [35, 56] for more details.

A property of the equilibrium assumption is that also intermediate species are incorporated. These species will naturally appear in the solution obtained by a simulation.

3.1.2b Burke-Schumann Flame Sheet

In the Flame-Sheet chemistry model, originally developed by Burke and Schumann [10], infinitely fast irreversible chemistry is assumed. By this assumption, the chemical reaction mechanism reduces to a one-step chemistry model, given by the reaction, involving fuel

(F), oxidizer (O) and products (P):

$$\nu_F F + \nu_O O \rightharpoonup \nu_P P. \tag{3.2}$$

Since the reaction is assumed irreversible, (3.2) can proceed only from left to right. As a result, fuel and oxidizer cannot be found together at the same location. In case of a one-step reaction, the expression for mixture fraction can be reformulated as

$$\xi = \frac{sY_F - Y_O + Y_O^0}{sY_F^0 + Y_O^0}.$$

Y_F^0 and Y_O^0 are fuel and oxidizer mass fractions in the fuel and oxidizer inlet streams respectively and

$$s = \frac{\nu_O W_O}{\nu_F W_F}$$

is the stoichiometric ratio. The mixture fraction at stoichometric conditions is then defined by

$$\xi_{st} = \frac{Y_O^0}{sY_F^0 + Y_O^0}.$$

The chemical properties as a function of ξ, are on the fuel side ($\xi > \xi_{st}$),

$$\begin{aligned} Y_F(\xi) &= Y_F^0 \frac{\xi - \xi_{st}}{1 - \xi_{st}} \\ Y_O(\xi) &= 0 \\ Y_P(\xi) &= 1 - Y_F - Y_O \end{aligned} \tag{3.3}$$

and on the oxidizer side ($\xi < \xi_{st}$),

$$\begin{aligned} Y_F(\xi) &= 0 \\ Y_O(\xi) &= Y_O^0 \frac{\xi_{st} - \xi}{\xi_{st}} \\ Y_P(\xi) &= 1 - Y_F - Y_O. \end{aligned} \tag{3.4}$$

If the static enthalpy is also a conserved scalar and under the conditions of section 2.4.2c, the temperature can be written in terms of mixture fraction:

$$\begin{aligned} T(\xi) = &\quad \xi T_F^0 + (1-\xi)\, T_O^0 + \Delta T \tfrac{\xi}{\xi_{st}} \quad \text{for} \;\; \xi \leq \xi_{st} \\ T(\xi) = &\quad \xi T_F^0 + (1-\xi)\, T_O^0 + \Delta T \tfrac{1-\xi}{1-\xi_{st}} \quad \text{for} \;\; \xi \geq \xi_{st}, \end{aligned} \tag{3.5}$$

with

$$\Delta T = \frac{q^0}{c_p} \xi_{st}$$

and q^0 the heat release due to reaction.

The above expressions can now be used to determine the density:

$$\rho(\xi) = p_0 \frac{W(\xi)}{T(\xi)}, \tag{3.6}$$

with $W(\xi)$ the mean molecular weight:

$$\frac{1}{W(\xi)} = \sum_{i=1}^{N} \frac{Y_i(\xi)}{W_i}.$$

For the one-step chemistry (3.2), (3.6) yields:

$$\begin{aligned} \rho(\xi) &= p_0 \left[\sum_{i=1}^{N} \frac{Y_i(\xi)}{W_i} \right]^{-1} \frac{1}{T(\xi)} \\ &= p_0 \left[\frac{Y_F(\xi)}{W_F} + \frac{Y_O(\xi)}{W_O} + \frac{Y_P(\xi)}{W_P} \right]^{-1} \frac{1}{T(\xi)} \\ \Leftrightarrow \rho(\xi) &= p_0 \left[\frac{Y_O(\xi)}{W_O} + \frac{1-Y_O(\xi)}{W_P} \right]^{-1} \frac{1}{T(\xi)} \quad \text{for} \;\; \xi \leq \xi_{st} \\ \rho(\xi) &= p_0 \left[\frac{Y_F(\xi)}{W_F} + \frac{1-Y_F(\xi)}{W_P} \right]^{-1} \frac{1}{T(\xi)} \quad \text{for} \;\; \xi \geq \xi_{st}. \end{aligned} \tag{3.7}$$

This can be formulated as:

$$\rho(\xi) = \frac{p_0}{a\,(\xi/\xi_{st})^2 + b\,(\xi/\xi_{st}) + c} \tag{3.8}$$

$\xi \leq \xi_{st}$	a	$-\left(\frac{1}{W_O}-\frac{1}{W_P}\right) Y_F^0 \left[\Delta T + \xi_{st}\left(T_F^0 - T_O^0\right)\right]$
	b	$\left[\left(\frac{1}{W_O}-\frac{1}{W_P}\right) Y_F^0 + \frac{1}{W_P}\right] \left[\Delta T + \xi_{st}\left(T_F^0 - T_O^0\right)\right]$ $-\left(\frac{1}{W_O}-\frac{1}{W_P}\right) Y_F^0 T_O^0$
	c	$\frac{T_O^0}{W_P}$
$\xi \geq \xi_{st}$	a	$\left[\left(\frac{1}{W_F}-\frac{1}{W_P}\right) Y_F^0 \frac{\xi_{st}}{1-\xi_{st}}\right] \left[\xi_{st}\left(T_F^0 - T_O^0\right) - \Delta T \frac{\xi_{st}}{1-\xi_{st}}\right]$
	b	$\left[\left(\frac{1}{W_F}-\frac{1}{W_P}\right) Y_F^0 \frac{\xi_{st}}{1-\xi_{st}}\right] \left(T_O^0 + \frac{\Delta T}{1-\xi_{st}}\right) +$ $\left[\frac{1}{W_P} - \left(\frac{1}{W_F}-\frac{1}{W_P}\right) Y_F^0 \frac{\xi_{st}}{1-\xi_{st}}\right] \left[\xi_{st}\left(T_F^0 - T_O^0\right) - \Delta T \frac{\xi_{st}}{1-\xi_{st}}\right]$
	c	$\left[\frac{1}{W_P} - \left(\frac{1}{W_F}-\frac{1}{W_P}\right) Y_F^0 \frac{\xi_{st}}{1-\xi_{st}}\right] \left(T_O^0 + \frac{\Delta T}{1-\xi_{st}}\right)$

Table 3.1: Constants for the determination of the density.

with a, b and c constants from table 3.1.

As an example, a pure methane-oxygen flame is considered:

$$CH_4 + 2O_2 \rightharpoonup CO_2 + 2H_2O.$$

The following dimensional values apply: $W_P = (0.080/3)kg/mole$; $W_O = 0.032kg/mole$; $W_F = 0.016kg/mole$; $\xi_{st} = 0.2$; $T_0 = 273K$; $\Delta T = 10000K$; $p_0 = 101300Pa$; $\mathcal{R} = 8.31J/moleK$. The density is then given in fig. 3.1. For states close to stoichiometry, the density is not uniquely determined as a function of $(\rho\xi)$. This is shown in fig. 3.2.

3.1.2c Further Simplification

For the discussion of the algorithm, a further simplification is made: the quadratic term $(a(\xi/\xi_{st})^2)$ in the denominator is zero, if we assume that all molecular weights W_F, W_O and W_P are the same. In that case, the change in density is only due to the release of heat in the reaction zone, not because of the reaction itself (through the formation of

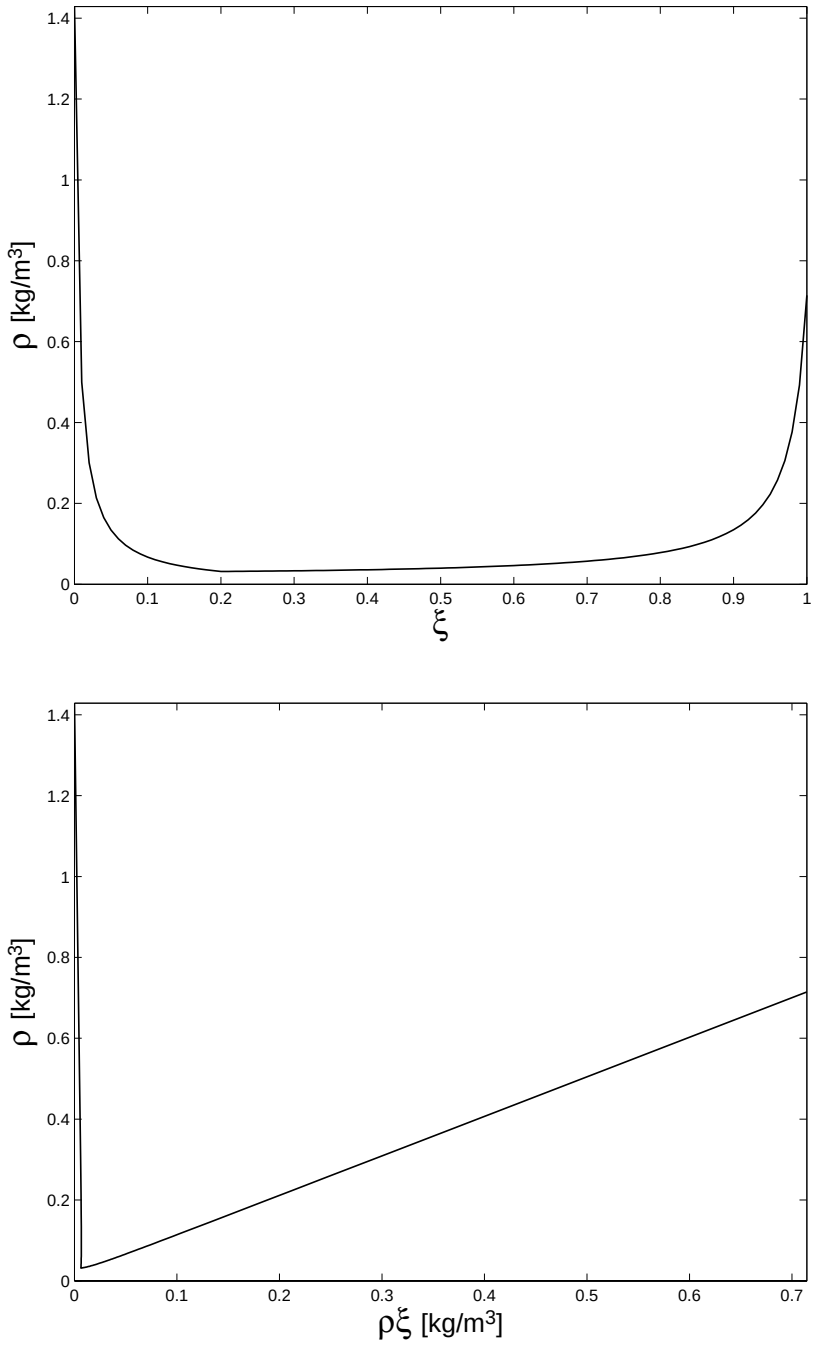

Figure 3.1: Density as a function of mixture fraction (ξ) and fuel elements mass ($\rho\xi$) for pure methane-oxygen combustion.

other species). We obtain formally

$$\rho(\xi) \quad = \quad p_0 \frac{1}{b\,(\xi/\xi_{st}) + c}$$

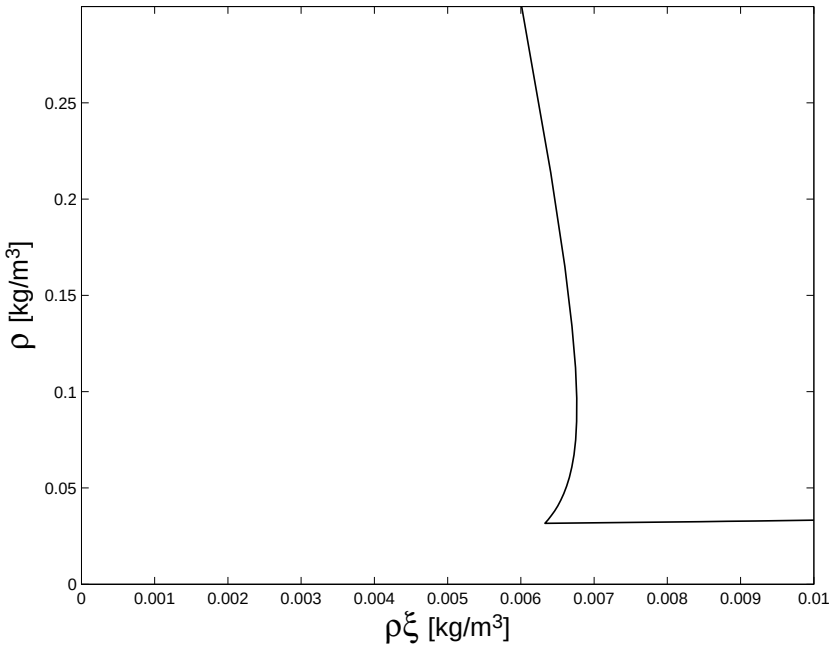

Figure 3.2: Density as a function of fuel elements mass ($\rho\xi$) for pure methane-oxygen combustion, zoom.

$$\begin{aligned} \Leftrightarrow \rho &= p_0 \frac{\rho}{b\,(\rho\xi/\xi_{st}) + c\rho} \\ \Leftrightarrow b\,(\rho\xi/\xi_{st}) + c\rho &= p_0 \\ \Leftrightarrow \rho &= p_0 - \frac{b}{c\xi_{st}}\rho\xi = \alpha\rho\xi + \beta, \end{aligned} \tag{3.9}$$

so that ρ is piecewise linearly dependent on $\rho\xi$. Expression (3.9) is illustrated in Fig. 3.3, using the dimensional values $W_P = W_O = W_F = 0.028kg/mole$, $\xi_{st} = 0.1$, $T_0 = 273K$, $\Delta T = 1000K$, $p_0 = 101300Pa$ and $\mathcal{R} = 8.31J/moleK$.

3.1.3 Inert Mixing

In the case of inert mixing, two species A and B, with mass fraction Y_A , resp. Y_B, are mixed. Since no other species are involved, $Y_A + Y_B = 1$. The mixture fraction is here defined as the mass fraction of species A, and equations (2.52) are valid. $\xi = 1$ in case of pure A, $\xi = 0$ in case of pure B. The density is obtained from the ideal gas law

$$\rho = \frac{p_0 W}{T}, \tag{3.10}$$

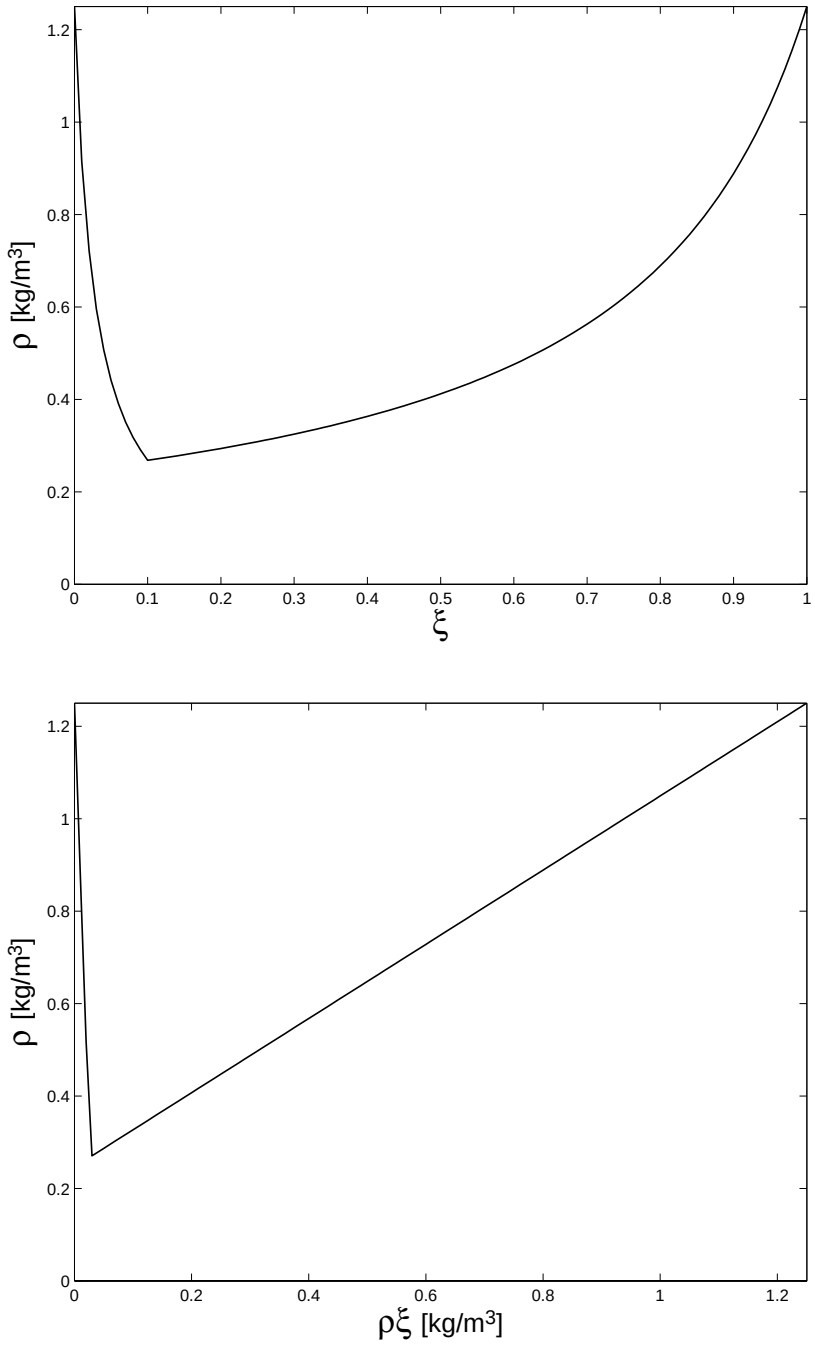

Figure 3.3: Density as a function of mixture fraction (ξ) and fuel elements mass ($\rho\xi$), if we assume all molecular weights to be equal and the same inlet temperature for fuel and oxidizer.

with W the mean molecular weight, defined by

$$\frac{1}{W} = \frac{Y_A}{W_A} + \frac{Y_B}{W_B}. \tag{3.11}$$

The density is then given as a function of mixture fraction by

$$\rho = \frac{p_0}{T} \frac{1}{\left(\frac{1}{W_A} - \frac{1}{W_B}\right)\xi + \frac{1}{W_B}} \tag{3.12}$$

or, after rearrangement, as a function of 'fuel' elements mass $\rho\xi$ by

$$\rho = \rho_B + \left(1 - \frac{\rho_B}{\rho_A}\right)\rho\xi, \tag{3.13}$$

with

$$\rho_A = \frac{p_0 W_A}{T}$$

and

$$\rho_B = \frac{p_0 W_B}{T}.$$

Expressions (3.12) and (3.13) are illustrated in Fig. 3.4, where the dimensional values $\rho_A = 1kg/m^3$, $\rho_B = 0.1kg/m^3$ are used.

3.2 Turbulence Modelling

The Navier-Stokes equations are intrinsically non-linear equations. An important non-linear term is the convective transport term in the momentum equations. This term has been the subject of many studies, since in many practical applications, this term plays an important role. Depending on the Reynolds number, two flow regimes can be distinguished. If the Reynolds number is low[5], the flow is called *laminar*. The non-linear term is of minor importance, and the flow is governed by viscous effects. In this case, a stable and smooth solution for the Navier-Stokes equations can be obtained. In high Reynolds-number flows, convection plays a dominant role and the flow is called *turbulent*. Because of the non-linearity, the flow becomes unstable and large flow structures break up in smaller and smaller eddies, until dissipated into heat by viscous effects. Smaller

[5]The exact value of the Reynolds number where transition from the laminar to the turbulent regime takes place, depends on the geometry of the configuration.

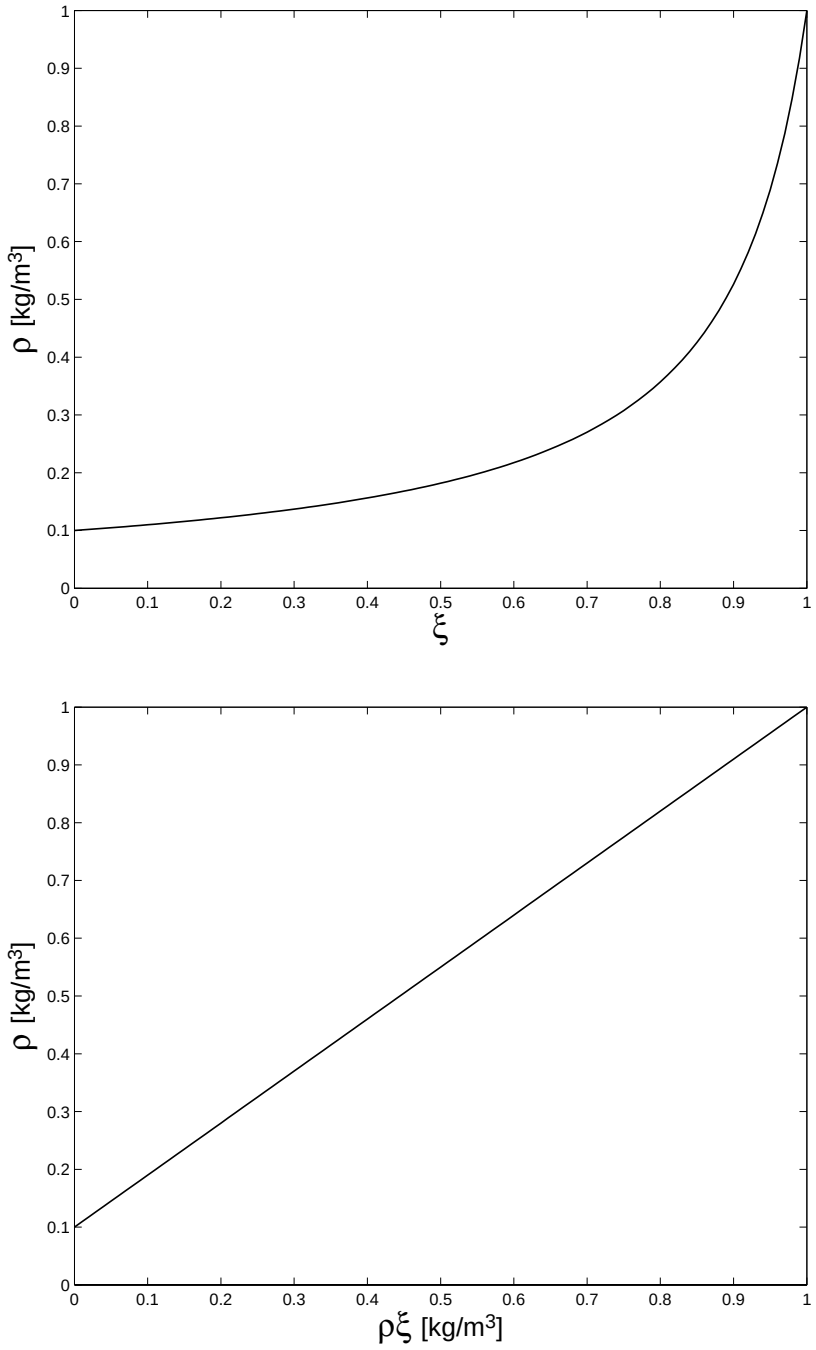

Figure 3.4: Density as a function of mixture fraction (ξ) and fuel elements mass ($\rho\xi$) for non-reacting flows.

flow structures exist than in the laminar regime. If all these structures would need to be resolved by a computational mesh, a very fine discretisation of the domain would be needed, resulting in an extremely high amount of grid points and a consequential growth

of computational time needed to simulate these flows.

To decrease the computing time of turbulent flows, turbulence models have been developed. These models allow for a coarser grid resolution since the effects of the small, unresolved structures are accounted for by the model.

Only a concise introduction to the existing turbulence models is given here. The interested reader is referred to [56, 76] for a full overview of the existing methods for simulating turbulent reacting flows.

3.2.1 Direct Numerical Simulation (DNS)

If one does not want to apply a turbulence model, all details of the flow must be resolved. The instantaneous equations of mass, species mass fraction, momentum and energy (2.39-2.42) apply and must be solved on a grid that captures all involved length and time scales. Such simulations are called Direct Numerical Simulation (DNS). This does not necessarily imply that all the features of combustion are resolved. DNS studies often use simplified physical properties and kinetic rates, or neglect density changes by heat release. Since the complexity of the flow increases (and the appearing time and length scales become smaller) with the Reynolds and Damköhler numbers, only flows with sufficiently low Re and Da can be calculated using DNS. In the extreme case, also sufficiently low Re and infinitely high Da can be calculated using DNS.

It is clear that no realistic flow configurations can be simulated using DNS. Therefore, the full numerical simulation of the instantaneous balance equations is limited to very simplified cases, where the number of time and length scales present in the flow is not too large [54, 55, 75]. Nevertheless, DNS is a useful tool for understanding and also to produce data sets that can be used for the validation of the models.

3.2.2 Large-Eddy Simulation (LES)

In LES, the evolution in time and space of a turbulent flow is calculated in such a way that all flow details that are larger in space than a certain filter size are resolved. All smaller scales are only implicitly taken into account. The objective of large-eddy simulation is thus to explicitly compute the large scales (typically the structures larger than the computational mesh size) while the effects of the smaller ones are modelled through the *subgrid model*. This separation of scales is possible since the large structures generally depend on the geometry of the system, whereas the small scales feature more universal properties. This generally valid assumption, makes the use of LES attractive and more and more feasible in the combustion community with the increase in available computer power. According

to the criterion of Pope [57], approximately 80% of the kinetic energy in the flow must be resolved in a high-quality LES. This means that still a high grid resolution is needed. To allow the large flow structures to break up into smaller ones, the flow has to be simulated in three dimensions and time-accuracy has to be retained, imposing requirements to the algorithm. Also, since the modelled terms are of the same order of magnitude as the convective terms, severe requirements are put to the discretisation[6].

In LES, the relevant quantities ϕ are filtered in spectral or physical space. The filtering operation is defined by:

$$\bar{\phi}(\boldsymbol{x}) = \int \phi(\boldsymbol{x}^*) F(\boldsymbol{x} - \boldsymbol{x}^*) \, \mathrm{d}\boldsymbol{x}^*,$$

where F is the LES-filter. In combusting flows, a mass weighted, Favre filtering is introduced as:

$$\bar{\rho}\tilde{\phi} = \overline{\rho\phi} = \int \rho\phi(\boldsymbol{x}^*) F(\boldsymbol{x} - \boldsymbol{x}^*) \, \mathrm{d}\boldsymbol{x}^*,$$

Filtering the instantaneous balance equations, leads to the governing equations used in the LES, e.g. for inert or reacting two fluid flow:

$$\begin{aligned}
\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial \bar{\rho}\tilde{u}_i}{\partial x_i} &= 0 \\
\frac{\partial \bar{\rho}\tilde{u}_j}{\partial t} + \frac{\partial \bar{\rho}\tilde{u}_i\tilde{u}_j}{\partial x_i} &= -\frac{\partial \bar{p}_2}{\partial x_j} + \frac{1}{\mathrm{Re}}\frac{\partial \bar{\tau}_{ij}}{\partial x_i} + \frac{1}{\mathrm{Fr}^2}\bar{\rho}\delta_{j3} - \frac{\partial}{\partial x_i}\left[\bar{\rho}\left(\widetilde{u_i u_j} - \tilde{u}_i\tilde{u}_j\right)\right] \\
\frac{\partial \bar{\rho}\tilde{\xi}}{\partial t} + \frac{\partial \bar{\rho}\tilde{\xi}\tilde{u}_i}{\partial x_i} &= \frac{1}{\mathrm{PrRe}}\frac{\partial \bar{J}_i}{\partial x_i} - \frac{\partial}{\partial x_i}\left[\bar{\rho}\left(\widetilde{u_i\xi} - \tilde{u}_i\tilde{\xi}\right)\right].
\end{aligned}$$

The last terms in the momentum and mixture fraction equation, are unclosed and require a model. The simplest model for the subgrid Reynolds stresses is given by Smagorinsky [67]:

$$\widetilde{u_i u_j} - \tilde{u}_i\tilde{u}_j = \nu_t \left(\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} \right) - \frac{2}{3}\frac{\partial \tilde{u}_k}{\partial x_k}\delta_{ij} \right), \tag{3.14}$$

using a turbulent viscosity, determined by the LES-filter width Δ and the Smagorinsky

[6] Convective terms should be discretised with a minimal amount of numerical dissipation. To stabilise the simulation, an energy conserving discretisation of the momentum convection is mostly needed.

constant C_s:

$$\nu_t = \frac{1}{\mathrm{Re}} \left(C_s \Delta\right)^2 \sqrt{\frac{1}{2} \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right) \left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i}\right)}.$$

The subgrid scale mixture fraction flux is modelled as

$$\widetilde{u_i \xi} - \tilde{u}_i \tilde{\xi} = \frac{1}{\mathrm{RePr}} D_t \frac{\partial \tilde{\xi}}{\partial x_i},$$

with

$$D_t = \frac{\nu_t}{\mathrm{Sc}_t}, \tag{3.15}$$

mostly using a constant turbulent Schmidt number.

3.2.3 Reynolds-Averaged Navier-Stokes (RANS)

Traditional turbulence models do not calculate large scale structures, but are developed to obtain a set of mean values. In Reynolds-Averaged Navier-Stokes (RANS) studies, one consideres the turbulent flow from a statistical point of view and then restricts the description to a subset of all statistical properties, e.g. the mean value and (co-)variance of velocity components and concentrations. The mean values that are considered can be time-averages or ensemble averages, depending on the situation.

In RANS, the relevant quantities ϕ are split into a mean value $\bar{\phi}$ and a deviation from the mean ϕ':

$$\phi = \bar{\phi} + \phi'.$$

If Favre averaging is considered, the splitting is formally the same

$$\phi = \tilde{\phi} + \phi''.$$

Averaging the instantaneous balance equations, leads to the governing equations used in RANS simulations, e.g. for inert or reacting two fluid flow:

$$\begin{aligned}
\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial \bar{\rho}\tilde{u}_i}{\partial x_i} &= 0 \\
\frac{\partial \bar{\rho}\tilde{u}_j}{\partial t} + \frac{\partial \bar{\rho}\tilde{u}_i\tilde{u}_j}{\partial x_i} &= -\frac{\partial \bar{p}_2}{\partial x_j} + \frac{1}{\mathrm{Re}}\frac{\partial \bar{\tau}_{ij}}{\partial x_i} + \frac{1}{\mathrm{Fr}^2}\bar{\rho}\delta_{j3} - \frac{\partial}{\partial x_i}\left(\widetilde{\bar{\rho}u_i''u_j''}\right) \\
\frac{\partial \bar{\rho}\tilde{\xi}}{\partial t} + \frac{\partial \bar{\rho}\tilde{\xi}\tilde{u}_i}{\partial x_i} &= \frac{1}{\mathrm{PrRe}}\frac{\partial \bar{J}_i}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\widetilde{\bar{\rho}u_i''\xi''}\right).
\end{aligned}$$

Since the turbulent fluxes in RANS have a less universal character than the subgrid terms in LES, the last terms in the momentum (Reynolds stresses) and mixture fraction (mixture fraction turbulent flux) equation require a better model. Since in RANS, the breaking of large structures into smaller ones is not simulated, a 3D grid is not strictly necessary, so that one can rely on the symmetry of the problem. Also, a time dependent formulation is not needed[7]. In general, the requirements for accuracy of the discretisation are also less restringent, because of the relatively large importance of the turbulent model terms.

3.3 Turbulence-Chemistry Interaction Modelling

If a DNS is performed, the instantaneous balance equations are solved and the equation of state, in the formulation

$$\rho = \rho\left(\xi\right)$$

applies. If a turbulence model is incorporated, and some kind of averaging is introduced, the above relationship is still true, but cannot be simply applied to averaged values, because of the highly non-linear character of the function $\rho(\xi)$:

$$\bar{\rho} = \overline{\rho\left(\xi\right)} \neq \rho\left(\bar{\xi}\right).$$

To overcome this problem, the probability density functions come into play.

[7] An exception is Unsteady RANS (URANS), where a transient problem is solved.

3.3.1 Probability Density Function (PDF)

If the distribution of a variable (e.g. mixture fraction) is known at every spatial coordinate at every time, the mean density can be derived, by integrating the distribution:

$$\bar{\rho}\left(\boldsymbol{x},t\right)=\int_{0}^{1}\rho\left(\xi\right)P\left(\xi;\boldsymbol{x},t\right)\,\mathrm{d}\xi. \tag{3.16}$$

The distribution P is called the probability density function. Since the real distribution of the mixture fraction is not known, it has to be estimated. Only using the first moment (the mean value), is not sufficient because of the non-linearity. Therefore, mostly two moments are used, being the mean and the variance of mixture fraction, and the shape of the PDF is assumed to be a β-shape, defined by:

$$P\left(\xi;\boldsymbol{x},t\right)=\frac{\xi^{\alpha-1}\left(1-\xi\right)^{\beta-1}}{\Gamma\left(\alpha\right)\Gamma\left(\beta\right)}\Gamma\left(\alpha+\beta\right).$$

Γ is the gamma-function and the two parameters $\alpha=\tilde{\xi}\gamma$ and $\beta=\left(1-\tilde{\xi}\right)\gamma$, with γ given by

$$\gamma=\frac{\tilde{\xi}\left(1-\tilde{\xi}\right)}{\widetilde{\xi''^{2}}}-1,$$

are related to the mean $\tilde{\xi}\left(\boldsymbol{x},t\right)$ and the variance $\widetilde{\xi''^{2}}\left(\boldsymbol{x},t\right)$. The shape of the β-function is chosen because it can mimic various functions. The mixture fraction variance can be obtained from integration of an extra transport equation or from an algebraic expression, requiring the knowledge of the mean mixture fraction field.

Given the mean mixture fraction $\tilde{\xi}$ and the mixture fraction variance $\widetilde{\xi''^{2}}$ in a certain point, the density in that point can be determined, by integrating eq. (3.16). The integration can be carried out before the simulation starts, and the results can be stored in look-up tables. Density is then function of two variables, by the adjusted equation of state for turbulent flows:

$$\bar{\rho}=\bar{\rho}\left(\tilde{\xi},\widetilde{\xi''^{2}}\right). \tag{3.17}$$

3.3.2 Other Approaches

Instead of pre-assuming the shape of the PDF, it can also be explicitly calculated. In that case, statistical information is gathered by a combination of fluid particles in the control volume. These particles are followed, typically using a Lagrangian formulation. Since a high amount of particles is required, such simulations become very costly. Apart from a few exceptions, this research area has therefore mostly been limited to the RANS turbulence modelling concept.

Another way to incorporate chemistry is not to use flamelet tables, but instead to solve balance equations for the mean value of species mass fractions, conditional on a given value of mixture fraction, also called *conditional averages*. This approach is known as the Conditional Moment Closure (CMC) model [7, 34]. By using conditional averages, the non-linearity in the relationship between species mass fraction and mixture fraction is dealt with. On the other hand, the number of transport equations increases a lot. Since the integration of all transport equations is a very tedious task, the conditional equations are solved on a mesh that is much coarser than the CFD-mesh to solve the flow equations.

Many other models exist that are beyond the scope of this work. For a comprehensive review, the reader is again referred to [76]. We only wish to point out that the equation of state in all of these cases can be brought under the formulation $\mathcal{G}\left(\rho, y_1, ..., y_N\right) = 0$, where the amount of scalars y_α depends on the model used.

Chapter 4

Discretised Equations

In the previous chapters, we set a practical amount of equations that can be used as a starting point for (non-)reacting flow simulations. The task is now to transform these analytical equations in a format a calculating machine can deal with. Doing so, we obtain the discretised equations.

In the governing equations, two types of derivatives appear. There are derivatives that express the variation in time ($\partial/\partial t$) and derivatives that express a variation in space ($\partial/\partial x_i$). These derivatives have a different character and will therefore be discretised in a different manner. Both types are separately discussed in the next sections. Combining the two leads to the fully discretised equation, with conditions to obtain a stable solution.

For the sake of clarity, the mixture fraction transport equation will be used as an example throughout this chapter. In this equation, several terms appear that are representative for any general transport equation. We make distinction between the time derivative (I), the convective (II) and the diffusive (III) term:

$$\underbrace{\frac{\partial \rho \xi}{\partial t}}_{\text{I}} + \underbrace{\frac{\partial \rho \xi u_i}{\partial x_i}}_{\text{II}} = \underbrace{\frac{1}{\mathrm{PrRe}} \frac{\partial}{\partial x_i} \left(\rho D \frac{\partial \xi}{\partial x_i} \right)}_{\text{III}}.$$

4.1 Finite Volume Discretisation

In general, three approaches exist to discretise differential equations: finite differences, finite elements and finite volumes. In the field of computational fluid dynamics (CFD), the finite volumes discretisation is most common. It is indeed a natural choice, because

the differential equations themselves are derived by expressing conservation properties over a control volume:

$$\frac{\partial}{\partial t}\int_{\Omega_{i,j}} \rho\xi \,\mathrm{d}V + \oint_{\partial\Omega_{i,j}} \rho\xi u_k n_k \,\mathrm{d}S \;=\; \frac{1}{\mathrm{PrRe}}\oint_{\partial\Omega_{i,j}} \rho D\frac{\partial\xi}{\partial x_k} n_k \,\mathrm{d}S. \tag{4.1}$$

The aim of the finite volume approach is to divide the domain into a number of sub-volumes and to express the integral conservation laws over these volumes. In that way, the conservation properties are exactly fulfilled, also in the discretised version. Numerical errors have no impact on this property.

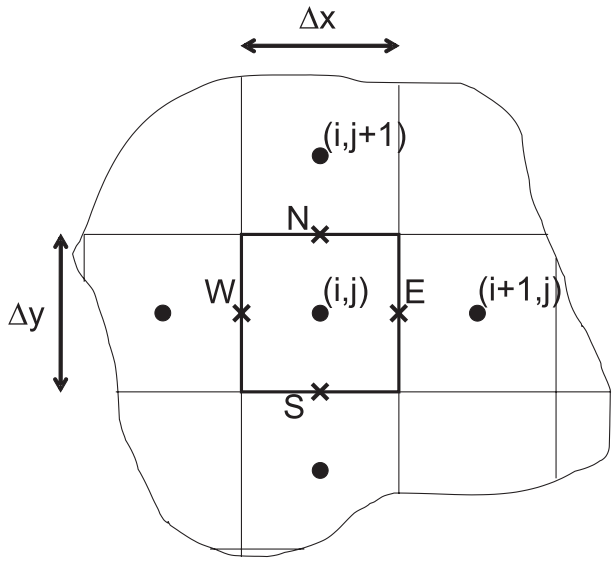

Figure 4.1: Finite volume discretisation in two dimensions on a Cartesian mesh.

If a rectangular and uniform grid is used in two dimensions, with grid spacings Δx and Δy, the discretisation reads:

$$\begin{aligned}\frac{\partial}{\partial t}\left(\rho\xi\right)_{i,j}\Delta x\Delta y \;=\;& F_{C,E}+F_{C,W}+F_{C,N}+F_{C,S}\\ &+F_{D,E}+F_{D,W}+F_{D,N}+F_{D,S},\end{aligned}$$

where F_C and F_D are the convective, resp. diffusive fluxes at the East, West, North and South faces.

4.2 Time Discretisation

The simplest and least accurate way to advance the equations in time is by a first order approximation:

$$\frac{\partial}{\partial t}\left(\rho\xi\right)_{i,j} \approx \frac{\left(\rho\xi\right)_{i,j}^{n+1} - \left(\rho\xi\right)_{i,j}^{n}}{\Delta t}.$$

By this discretisation, the time of interest is subdivided into a series of smaller time steps Δt. A series of subsequent discretisations in time is then able to predict the value at the time level of interest, starting from a known initial condition at time level n. Depending on the time level at which the spatial terms are discretised, we call this the forward Eulerian (time level n), or backward Eulerian (time level $n+1$) discretisation. The resulting scheme is then explicit, resp. implicit, where the implicit version has the disadvantage that it requires a matrix inversion. On the other hand, it does not suffer from a stability criterion. In this work, we restrict ourselves to explicit schemes, to obtain the time discretisation

$$\begin{aligned}\frac{\left(\rho\xi\right)_{i,j}^{n+1} - \left(\rho\xi\right)_{i,j}^{n}}{\Delta t}\Delta x\Delta y &= F_{C,E}^{n} - F_{C,W}^{n} + F_{C,N}^{n} - F_{C,S}^{n} \\ &\quad + F_{D,E}^{n} - F_{D,W}^{n} + F_{D,N}^{n} - F_{D,S}^{n}.\end{aligned}$$

Higher order accuracy in time can easily be achieved by performing several Euler-explicit steps during one time-step. The resulting *Multi-Stage* algorithm then reads, e.g. for a 4-stage low storage Runge-Kutta scheme:

$$\begin{aligned}\left(\rho\xi\right)^{(0)} &= \left(\rho\xi\right)^{n} \\ \left(\rho\xi\right)^{(1)} &= \left(\rho\xi\right)^{(0)} + \alpha_1 \frac{\Delta t}{\Delta x\Delta y}\sum F^{(0)} \\ \left(\rho\xi\right)^{(2)} &= \left(\rho\xi\right)^{(0)} + \alpha_2 \frac{\Delta t}{\Delta x\Delta y}\sum F^{(1)} \\ \left(\rho\xi\right)^{(3)} &= \left(\rho\xi\right)^{(0)} + \alpha_3 \frac{\Delta t}{\Delta x\Delta y}\sum F^{(2)} \\ \left(\rho\xi\right)^{(4)} &= \left(\rho\xi\right)^{(0)} + \alpha_4 \frac{\Delta t}{\Delta x\Delta y}\sum F^{(3)} \\ \left(\rho\xi\right)^{n+1} &= \left(\rho\xi\right)^{(4)},\end{aligned}$$

where 3 intermediate stages are involved. The appearing flux terms $F^{(\nu)}$ are evaluated using the known state values at the intermediate time level ν. Standard coefficients are $\alpha_1 = 1/4$, $\alpha_2 = 1/3$, $\alpha_3 = 1/2$ and $\alpha_4 = 1$.

4.3 Space Discretisation

4.3.1 Discretisation of the Convective Fluxes

The convective fluxes at the East, West, North and South faces are given by the following expressions:

$$\begin{aligned} F_{C,E} &= - \Delta y \left(\rho u_x \xi\right)_E, & F_{C,W} &= \Delta y \left(\rho u_x \xi\right)_W, \\ F_{C,N} &= - \Delta x \left(\rho u_y \xi\right)_N, & F_{C,S} &= \Delta x \left(\rho u_y \xi\right)_S. \end{aligned}$$

Depending on the actual implementation, either density and velocity can be considered together to yield:

$$\begin{aligned} F_{C,E} &= - \dot{m}_E \xi_E, & F_{C,W} &= \dot{m}_W \xi_W, \\ F_{C,S} &= - \dot{m}_N \xi_N, & F_{C,N} &= \dot{m}_S \xi_S, \end{aligned}$$

with the mass flux at the face

$$\begin{aligned} \dot{m}_E &= \Delta y \left(\rho u_x\right)_E, & \dot{m}_W &= \Delta y \left(\rho u_x\right)_W, \\ \dot{m}_S &= \Delta x \left(\rho u_y\right)_N, & \dot{m}_N &= \Delta x \left(\rho u_y\right)_S, \end{aligned}$$

or density and mixture fraction can be considered together, yielding

$$\begin{aligned} F_{C,E} &= - \dot{v}_E \left(\rho\xi\right)_E, & F_{C,W} &= \dot{v}_W \left(\rho\xi\right)_W, \\ F_{C,N} &= - \dot{v}_N \left(\rho\xi\right)_N, & F_{C,S} &= \dot{v}_S \left(\rho\xi\right)_S, \end{aligned} \tag{4.2}$$

with the face velocity flux

$$\begin{aligned} \dot{v}_E &= \Delta y u_{xE}, & \dot{v}_W &= \Delta y u_{xW}, \\ \dot{v}_N &= \Delta x u_{y_N}, & \dot{v}_S &= \Delta x u_{y_S}. \end{aligned}$$

Continuing with eq. (4.2), the value for $(\rho\xi)$ at the face can be estimated from the known node values as a simple average:

$$\begin{aligned} \left(\rho\xi\right)_E &= \tfrac{(\rho\xi)_{i,j}+(\rho\xi)_{i+1,j}}{2}, & \left(\rho\xi\right)_W &= \tfrac{(\rho\xi)_{i,j}+(\rho\xi)_{i-1,j}}{2}, \\ \left(\rho\xi\right)_N &= \tfrac{(\rho\xi)_{i,j}+(\rho\xi)_{i,j+1}}{2}, & \left(\rho\xi\right)_S &= \tfrac{(\rho\xi)_{i,j}+(\rho\xi)_{i,j-1}}{2}. \end{aligned}$$

However, such a central discretisation gives rise to wiggles in the solution and, moreover, is not stable in combination with a forward Euler discretisation in time. For that reason, the concept of *upwinding* has been introduced. In upwind schemes, the face values are estimated using essentially only these node values, where historical information is situated. In the simple case of a constant velocity going from West to East, fluid particles located at face E were previously located upstream (*upwind*), i.e. on the western side of face E. In its simplest formulation, the first order upwind scheme discretises the face value as:

$$
\begin{array}{rclcrcl}
(\rho\xi)_E & = & \left\{ \begin{array}{ll} (\rho\xi)_{i,j} & \text{for } \dot{v}_E > 0 \\ (\rho\xi)_{i+1,j} & \text{for } \dot{v}_E < 0 \end{array} \right. , & (\rho\xi)_W & = & \left\{ \begin{array}{ll} (\rho\xi)_{i-1,j} & \text{for } \dot{v}_W > 0 \\ (\rho\xi)_{i,j} & \text{for } \dot{v}_W < 0 \end{array} \right. , \\
(\rho\xi)_N & = & \left\{ \begin{array}{ll} (\rho\xi)_{i,j} & \text{for } \dot{v}_N > 0 \\ (\rho\xi)_{i,j+1} & \text{for } \dot{v}_N < 0 \end{array} \right. , & (\rho\xi)_S & = & \left\{ \begin{array}{ll} (\rho\xi)_{i,j-1} & \text{for } \dot{v}_S > 0 \\ (\rho\xi)_{i,j} & \text{for } \dot{v}_S < 0 \end{array} \right. .
\end{array}
$$

Application of the first order upwind scheme yields stable results and has the property of monotonicity, i.e. no new local extrema are created and the value of a local minimum/maximum is non-decreasing/non-increasing in time. This property is very useful for bounded scalars, such as density and mixture fraction. A disadvantage of the upwind scheme is its high level of numerical dissipation that causes sharp gradients in a scalar field to be smoothed out in subsequent time levels. Higher accuracy, with less numerical dissipation but still monotonicity preserving properties, can be achieved, making use of non-linear limiter functions. The according discretisation schemes are called *Total Variation Diminishing* (TVD) schemes.

4.3.2 Discretisation of the Diffusive Fluxes

The diffusive fluxes have a different character than the convective fluxes in the sense that all directions are equivalent. For a 2D configuration, the diffusive fluxes are given by the following expressions:

$$
\begin{array}{rclcrcl}
F_{D,E} & = & \frac{1}{\mathrm{PrRe}} \Delta y \left(\rho D \frac{\partial \xi}{\partial x} \right)_E , & & F_{D,W} & = - & \frac{1}{\mathrm{PrRe}} \Delta y \left(\rho D \frac{\partial \xi}{\partial x} \right)_W , \\
F_{D,N} & = & \frac{1}{\mathrm{PrRe}} \Delta x \left(\rho D \frac{\partial \xi}{\partial y} \right)_N , & & F_{D,S} & = - & \frac{1}{\mathrm{PrRe}} \Delta x \left(\rho D \frac{\partial \xi}{\partial y} \right)_S ,
\end{array}
$$

which can be discretised, using central differences, as

$$
\begin{array}{rclcrcl}
F_{D,E} & = & \frac{1}{\mathrm{PrRe}} \Delta y \, (\rho D)_E \, \frac{\xi_{i+1,j} - \xi_{i,j}}{\Delta x}, & & F_{D,W} & = & \frac{1}{\mathrm{PrRe}} \Delta y \, (\rho D)_E \, \frac{\xi_{i-1,j} - \xi_{i,j}}{\Delta x}, \\
F_{D,N} & = & \frac{1}{\mathrm{PrRe}} \Delta x \, (\rho D)_E \, \frac{\xi_{i,j+1} - \xi_{i,j}}{\Delta y}, & & F_{D,S} & = & \frac{1}{\mathrm{PrRe}} \Delta x \, (\rho D)_E \, \frac{\xi_{i,j-1} - \xi_{i,j}}{\Delta y}.
\end{array}
$$

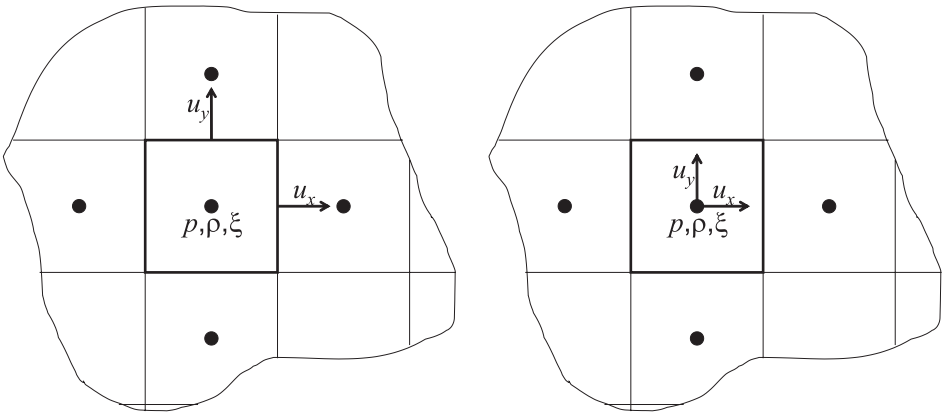

Figure 4.2: Storage of variables in the staggered (left) and collocated (right) grid topology.

4.3.3 Staggered vs. Collocated Grid Topology

From eq. (4.2) it is seen that the 'best' way to store the cell face velocities is at the cell face itself. This way, different variables are stored at different places and the *staggered* grid topology of fig. 4.2 is used. The convective flux (4.2) can then immediately be discretised as

$$\begin{aligned} F_{C,E} &= -\Delta y u_{x_{i+\frac{1}{2},j}} (\rho\xi)_E , \quad F_{C,W} = \Delta y u_{x_{i-\frac{1}{2},j}} (\rho\xi)_W , \\ F_{C,N} &= -\Delta x u_{y_{i,j+\frac{1}{2}}} (\rho\xi)_N , \quad F_{C,S} = \Delta x u_{y_{i,j-\frac{1}{2}}} (\rho\xi)_S . \end{aligned}$$

The face velocities are calculated from the discretised momentum equations, using shifted control volumes, centred around the velocity storage points. The elegant this approach appears to be in uniform, Cartesian grids, the cumbersome it appears in more general three-dimensional environments making use of body-fitted grids. In these more general cases there are practical advantages to use grids with *collocated* arrangements, that store all variables at the same place per cell. A simple central discretisation of the cell face velocity is however not a good option, because then spurious modes appear in the solution. Special measures have to be taken to interpolate the cell face velocity. For the solution to this problem, we refer to chapter 8.

4.4 Stability Limit

If the Euler-explicit scheme is used, special care is needed to obtain a stable solution. In practice, this means that the time step Δt must not be too large. From a linear stability analysis, a maximum time step can be calculated for the system to be stable. In appendix

A, this is done for a pure convection-diffusion equation. In reality, the system of discretised governing equations is non-linear, but still a maximum time step following from the linear analysis can provide a good estimate for stability. The CFL-number of the simulation is then defined by the ratio of the actual time step to the maximum time step for stability:

$$CFL = \frac{\Delta t}{(\Delta t)_{\max}}. \tag{4.3}$$

In most of the computations, a CFL-number as high as 0.9 will be used in order to prove that the stability of the schemes investigated is only dependent on the linear time step limit and not on any density ratio.

4.5 Summary

Unless stated otherwise, we will use in this work[1] the following finite-volume discretisations:

- first order upwind differencing for the convective terms[2];
- second order central differencing for the diffusive terms;
- second order central differencing for the pressure term;
- explicit Euler forward differencing in time.

One-dimensional test cases will be performed with a staggered grid arrangement, whereas the more-dimensional test cases will be simulated on a collocated mesh, using the pressure stabilisation term from chapter 8.

[1] without restriction for the applicability of the findings of the present work for other types of discretisations.

[2] We recall that first order upwind differencing introduces a high amount of numerical diffusion and is therefore not suited for LES.

Chapter 5

Algorithm Development

In the previous sections, we derived the governing equations that describe the physics of the reacting flow field (chapter 2). In chapter 3, the number of equations is drastically reduced by means of theoretically founded model assumptions. Still, the equations are too complicated for an analytical solution. In chapter 4, tools are provided to represent the equations in discrete points, so that a numerical solution can be obtained. In this section, a step by step recipe is given to predict the state at a next time level, based on the knowledge of the present state and making use of the discretised forms of the governing equations. The step by step recipe is called an 'algorithm'.

Although some researchers prefer an algorithm that solves for all the equations at once ('coupled' solution strategy, see chapter 6), the most natural way to solve the low Mach number equations is the use of a segregated solution strategy, by means of a pressure-correction[1] algorithm. Using the latter class of algorithms, the equations are solved sequentially, one at a time, after which a global correction step is taken to account for the pressure influence.

In the literature, two different approaches exist to construct this pressure correction equation. Either the continuity equation is used, by which a constant coefficient Poisson equation for pressure is obtained [16, 33, 39, 43, 46], or another equation, based on the equation of state, is used, resulting in a density weighted (variable coefficient) Poisson equation [5, 20, 48, 69]. The former is easier to solve and saves computing time, but is unstable for high density ratios (approximately a factor of 3, according to literature). The latter makes the simulation much more stable (even up to density ratios of 1:800 [5]), at the expense of more computational effort to solve the variable coefficient Poisson equation and the loss of conservation properties [39] or the use of a modified equation of state [5, 20]. Moreover, to the best of the author's knowledge, a variable coefficient Poisson equation

[1] We use the term 'pressure-correction' to mark the general class of algorithms, literature refers to as 'pressure-correction', 'pressure-projection' and 'pressure-based' algorithms.

for reacting flows, making use of the mixture fraction as a conserved scalar, has not been reported yet. The reason for this, stems from the fact that chemistry is mostly tabulated, so that a right hand side of the density weighted pressure equation cannot be determined. Even when it can be done analytically, as is the case in the Burke-Schumann chemistry model, used in the present work, a huge error is made near stoichiometry, so that large errors would appear in the conservation of mass or fuel elements mass, or, depending on the equations actually solved, in the density-mixture fraction-relationship, in comparison with the physics, according to the equation of state.

In this chapter, a new pressure-correction algorithm is developed, that does not suffer from the shortcomings of the algorithms provided by literature. The starting point of this new algorithm is neither the one nor the other approach, described above. Instead, given a number of prerequisites, we construct a variable coefficient Poisson equation for pressure, albeit not in the standard $\nabla \cdot \left(\frac{1}{\rho}\nabla\right)$-formulation, that provides us a tool for stable time-accurate flow calculations of a fluid with a general equation of state, in a low Mach number environment. The applicability to non-premixed flames, making use of the mixture fraction as a conserved variable, will serve as illustrative example for the general case throughout the rest of this thesis.

5.1 Prerequisites

5.1.1 Stability, Robustness, Consistency

The basic property of a good algorithm is, of course, that a result can be obtained. Therefore, an algorithm needs to be stable. Stability can be defined as:

Stability In order to obtain a numerical solution of a discrete equation, it is necessary that a perturbation on the discrete solution is not amplified without bound.

An algorithm is called robust if, even for large perturbations, stability is guaranteed. In the context of the present work, the robustness is defined as:

Robustness The stability of the algorithm is not restricted to small perturbations, or other restrictions, such as the magnitude of the appearing density ratios.

An algorithm not only has to be stable and robust. The numerical solution, provided by the algorithm, needs to represent the reality. This property is called consistency:

Consistency The discretised equations should tend to the differential equations to which they are related when Δt and Δx tend to zero.

Remark that consistency has two variants: consistency in time and in space.

5.1.2 Time Accuracy

The goal is to construct an algorithm that behaves well for transient calculations. This property is important for the state-of-the-art simulations, where time-accurate results provide important data. Examples include large-eddy simulations used to get deeper insight in combustion instabilities.

5.1.3 Conservation

This property is directly related to the derivation of the governing equations and expresses the physics of the flow. Mass can not be created, nor destroyed. The same is true for momentum, energy and fuel elements mass, unless source terms appear in the equations. The conservation property is automatically fulfilled if the equations are discretised based on the integral expressions in a finite volume context.

In the development of the algorithm, mass, fuel elements mass and energy are ought to be exactly conserved. The conservation of momentum is considered to be of minor importance since, e.g. in LES, the convective terms are mostly discretised in a skew-symmetric form. This is done to retain stability in high Reynolds number flows, when the convective terms are discretised with a minimal amount of numerical dissipation. In this case, kinetic energy is conserved, at the expense of momentum conservation.

5.1.4 Fulfillment of Equation of State

Once a new state at a time level is obtained, we expect it to be a physically possible result. In other words, we expect the variables to behave according to the equation of state. In case of an inert or reacting two fluid flow, density and mixture fraction must satisfy eq. (2.53). In case of a single fluid ideal gas, density, temperature and thermodynamic pressure must satisfy eq. (2.55).

5.2 The Pressure-Correction Formalism

The pressure-correction formalism, that has its origin in early papers [13, 14, 26], was originally developed for constant density flows. The general algorithm has proven to be accurate in these flows and no substantial problems are encountered there. The basic idea behind the pressure-correction strategy, is to advance momentum in two steps. In the first step, all influences are accounted for, except for the pressure, whose influence is reserved for the second step. Assuming we know the variables at time n, the state at time level $n+1$ is calculated in the way described hereafter.

5.2.1 Velocity Prediction

The velocity (or, equally, momentum) field is determined in two steps. First a prediction is made, using the momentum equations with the pressure term evaluated at time level n:

$$(\rho u_j)^* = (\rho u_j)^n + \Delta t \left[-\frac{\delta \left(\rho u_i u_j\right)^n}{\delta x_i} - \frac{\delta p^n}{\delta x_j} + \frac{1}{\mathrm{Re}} \frac{\delta \tau_{ij}^n}{\delta x_i} + \frac{1}{\mathrm{Fr}^2} \rho^n \delta_{j3} \right], \tag{5.1}$$

where the subscript 2 for the kinematic pressure is dropped for simplicity of the notation. The δ-notation is introduced to stress the fact that the derivatives are discrete.

5.2.2 Pressure Correction

Ultimately, we do not wish to satisfy eq. (5.1), but the following equation, with the pressure, being an acoustic term, evaluated implicitly at time level $n+1$:

$$(\rho u_j)^{n+1} = (\rho u_j)^n + \Delta t \left[-\frac{\delta \left(\rho u_i u_j\right)^n}{\delta x_i} - \frac{\delta p^{n+1}}{\delta x_j} + \frac{1}{\mathrm{Re}} \frac{\delta \tau_{ij}^n}{\delta x_i} + \frac{1}{\mathrm{Fr}^2} \rho^n \delta_{j3} \right].$$

The predicted field is corrected to give the velocity at the new time level

$$(\rho u_j)^{n+1} = (\rho u_j)^* + (\rho u_j)', \tag{5.2}$$

where the correction for the momentum $(\rho u_j)'$ is related to the correction for the pressure

$p' = p^{n+1} - p^n$:

$$(\rho u_j)' = -\Delta t \frac{\delta p'}{\delta x_j}. \tag{5.3}$$

The correction for the pressure follows from inversion of an elliptic equation, based on a constraining equation for the velocity field at time level $n+1$. In constant density flows, the continuity equation naturally imposes a constraint on the velocity field:

$$\frac{\delta \left(\rho u_i\right)^{n+1}}{\delta x_i} = 0, \tag{5.4}$$

so that the pressure equation in this case is:

$$\frac{\delta^2 p'}{\delta x_i \delta x_i} = \frac{1}{\Delta t} \frac{\delta \left(\rho u_i\right)^*}{\delta x_i}, \tag{5.5}$$

In variable density flow, the constraining equation can be built in multiple ways and forms the major difference between several pressure-correction formalisms. The key to obtain a pressure-correction algorithm that satisfies all prerequisites, is to build a consistent constraining equation for the velocity (or momentum), as is done in the next section.

5.3 Algorithmic Strategy for General Incompressible Fluid

Let us consider the general case of a flow, governed by the equations of continuity (2.39), momentum (2.41), and a set of conservation equations for N scalars:

$$\frac{\partial \rho y_\alpha}{\partial t} + \frac{\partial \left(\rho u_i y_\alpha\right)}{\partial x_i} = RHS_\alpha, \qquad \alpha = 1, \ \ldots \ , \ N. \tag{5.6}$$

We assume the right hand side of the transport equations, containing diffusive and source terms, to be discretised conservatively. If a first order time stepping is used, the continuity and scalar equations are discretised in time as:

$$\rho^{n+1} = \rho^n - \Delta t \frac{\delta \left(\rho u_i\right)^n}{\delta x_i} \tag{5.7}$$

$$\left(\rho y_\alpha\right)^{n+1} = \left(\rho y_\alpha\right)^n - \Delta t \frac{\delta \left(\rho u_i y_\alpha\right)^n}{\delta x_i} + \Delta t RHS_\alpha^n, \tag{5.8}$$

which provides us the values of density and scalars at the new time level. As explained in section 5.2 the velocity is determined in two steps. The predicted value follows from eq. (5.1), whereas the ultimate value at time level $n+1$ follows from a constraining equation. A constraint for $\boldsymbol{u}^{n+1}$ is found by combination of (5.7) and (5.8), shifted at the next time level:

$$\rho^{n+2} = \rho^{n+1} - \Delta t \frac{\delta \left(\rho u_i\right)^{n+1}}{\delta x_i} \tag{5.9}$$

$$\left(\rho y_\alpha\right)^{n+2} = \left(\rho y_\alpha\right)^{n+1} - \Delta t \frac{\delta \left(\rho u_i y_\alpha\right)^{n+1}}{\delta x_i} + \Delta t RHS_\alpha^{n+1}. \tag{5.10}$$

For a general fluid at low Mach number, a general equation of state can be formulated, expressing that the state variables ρ and y_α are not independent:

$$\begin{aligned} &\mathcal{G}\left(\rho, y_1, ..., y_N\right) = 0 \\ \Leftrightarrow\ &\mathcal{H}\left(\rho, \rho y_1, ..., \rho y_N\right) = 0 \end{aligned} \tag{5.11}$$

The constraint is now formulated by requiring the equation of state to be fulfilled at every time level, in particular at time level $n+2$:

$$\mathcal{H}\left(\rho^{n+2}, \left(\rho y_1\right)^{n+2}, ..., \left(\rho y_N\right)^{n+2}\right) = 0, \tag{5.12}$$

which yields, after inserting (5.9) and (5.10), a non-linear equation in $\boldsymbol{u}^{n+1}$ and ultimately in p'.

To the best of the author's knowledge this strategy, though simple, has never been followed before. Researchers have always searched for methods to construct constraining equations, built on the analytical differential equations, yielding an algorithm that does not satisfy all the requirements we impose. A reason for this might be historical: the first algorithms were developed to obtain a steady state solution, where no benefit is to be found in a discrete construction of the constraint. When time accuracy becomes important, as it is nowadays, better algorithms are required, superior to what exists now.

In the following section, the general strategy described here will be applied to the cases of ideal gas single fluids and (non-)reacting two fluid mixtures.

5.4 Application to Simplified Fluid Flow

The above principles are now specified for common cases. For comprehensiveness, we describe the algorithms in 1D discrete space. Extension to more dimensions is straightforward. In all cases, we assume that the variables at time n are known, and describe how the state at time level $n+1$ is calculated.

5.4.1 Single Fluid Flow: Ideal Gas

5.4.1a Constraining Equation

The governing equations for an ideal gas, are the continuity, momentum and energy equations (2.54). Because of the importance of the conservation property, the energy equation in conservative form (2.51) is used. Given these equations, we can apply the general framework, considering one scalar equation with $y_1 = T$. The constraining equation is then formed, using the equation of state:

$$\begin{aligned} \mathcal{G}\left(\rho_i^{n+2}, T_i^{n+2}\right) &= \rho_i^{n+2} T_i^{n+2} - \check{p}_0^{n+2} = 0 \\ \Leftrightarrow \mathcal{H}\left(\rho_i^{n+2}, p_{0,i}^{n+2}\right) &= p_{0,i}^{n+2} - \check{p}_0^{n+2} = 0, \end{aligned} \tag{5.13}$$

where $\check{p}_0^{n+2}$ is assumed to be known. In open domains, $\check{p}_0^{n+2} = p_0^{n+1}$, in enclosures, $\check{p}_0^{n+2}$ follows from an integral equation over the whole domain (see chapter 9).

5.4.1b Algorithm Overview

The algorithm is now as follows. First, the density at the new time is determined from the continuity equation. Doing so, mass is conserved. This reads:

$$\rho_i^{n+1} = \rho_i^n - \frac{\Delta t}{\Delta x}\left(\rho_R^n u_{i+\frac{1}{2}}^n - \rho_L^n u_{i-\frac{1}{2}}^n\right) \tag{5.14}$$

where the L and R subscripts indicate extrapolated values at the left and right face of the control volume. For a first order upwind scheme, with positive values for the velocity, $\rho_L = \rho_{i-1}$, $\rho_R = \rho_i$.

A predicted value for the velocity is obtained, using the momentum equation,

$$(\rho u)_{i+\frac{1}{2}}^* = (\rho u)_{i+\frac{1}{2}}^n \quad -\tfrac{\Delta t}{\Delta x}\left((\rho u)_R^n\, u_{i+1}^n - (\rho u)_L^n\, u_i^n\right) - \tfrac{\Delta t}{\Delta x}\left(p_{i+1}^n - p_i^n\right) + \tfrac{\Delta t}{\mathrm{Re}}\tfrac{\delta \tau^n}{\delta x},$$

with

$$u^*_{i+\frac{1}{2}} = \frac{(\rho u)^*_{i+\frac{1}{2}}}{\rho^{n+1}_{i+\frac{1}{2}}}. \tag{5.15}$$

For first order upwinding and positive values of the velocity, the extrapolation is done, using $(\rho u)_L = \rho_{i-1/2} u_{i-1/2}$ and $(\rho u)_R = \rho_{i+1/2} u_{i+1/2}$, with averaged face density values: $\rho_{i+1/2} = (\rho_i + \rho_{i+1})/2$. The node velocities are calculated by averaging: $u_i = \left(u_{i-1/2} + u_{i+1/2}\right)/2$. The values for velocity,

$$u^{n+1}_{i+\frac{1}{2}} = u^*_{i+\frac{1}{2}} + u'_{i+\frac{1}{2}} \tag{5.16}$$

and pressure

$$p^{n+1}_i = p^n_i + p'_i$$

are related through

$$u'_{i+\frac{1}{2}} = -\Delta t \frac{1}{\rho^{n+1}_{i+\frac{1}{2}}} \frac{p'_{i+1} - p'_i}{\Delta x}. \tag{5.17}$$

Obeying the constraint (5.13), requires a discrete evaluation of the thermodynamic pressure, eq. (2.51):

$$p^{n+2}_{0,i} = p^{n+1}_0 - \frac{\Delta t}{\Delta x} \gamma p^{n+1}_0 \left(u^{n+1}_{i+\frac{1}{2}} - u^{n+1}_{i-\frac{1}{2}}\right) - \frac{\Delta t}{\Delta x} \frac{\gamma}{\mathrm{RePr}} \left(q^{n+1}_{i+\frac{1}{2}} - q^{n+1}_{i-\frac{1}{2}}\right), \tag{5.18}$$

with

$$q^{n+1}_{i+\frac{1}{2}} = -\lambda^{n+1}_{i+\frac{1}{2}} \frac{T^{n+1}_{i+1} - T^{n+1}_i}{\Delta x}$$

and

$$T^{n+1}_i = \frac{p^{n+1}_0}{\rho^{n+1}_i}.$$

Combination of (5.13) and (5.18), provides us the constraining equation for u^{n+1}:

$$\frac{u_{i+\frac{1}{2}}^{n+1} - u_{i-\frac{1}{2}}^{n+1}}{\Delta x} = RHS_i,$$

with

$$RHS_i = -\frac{1}{\gamma p_0^{n+1}} \frac{\check{p}_0^{n+2} - p_0^{n+1}}{\Delta t} - \frac{1}{p_0^{n+1}} \frac{1}{\mathrm{RePr}} \frac{q_{i+\frac{1}{2}}^{n+1} - q_{i-\frac{1}{2}}^{n+1}}{\Delta x}.$$

In combination with (5.16) and (5.17), this results in a variable coefficient Poisson equation for the pressure:

$$-\frac{\Delta t}{\Delta x^2} \left(\frac{p'_{i+1} - p'_i}{\rho_{i+\frac{1}{2}}^{n+1}} - \frac{p'_i - p'_{i-1}}{\rho_{i-\frac{1}{2}}^{n+1}} \right) = RHS_i - \frac{u_{i+\frac{1}{2}}^{*} - u_{i-\frac{1}{2}}^{*}}{\Delta x}. \tag{5.19}$$

5.4.2 Two-Fluid Flow: Inert Mixing

5.4.2a Constraining Equation

In case of inert mixing, again one scalar, namely the mixture fraction, is present. The constraining equation can now be written, using a chemical operator $\mathcal{H}_C$, defined as $\rho = \mathcal{H}_C(\rho\xi)$ according to (3.13):

$$\begin{aligned} \mathcal{G}\left(\rho_i^{n+2}, \xi_i^{n+2}\right) &= 0 \\ \Leftrightarrow \mathcal{H}\left(\rho_i^{n+2}, (\rho\xi)_i^{n+2}\right) &= 0, \\ \Leftrightarrow \rho_i^{n+2} &= \mathcal{H}_C\left((\rho\xi)_i^{n+2}\right) = \rho_B + \left(1 - \frac{\rho_B}{\rho_A}\right)(\rho\xi)_i^{n+2}. \end{aligned} \tag{5.20}$$

For the ease of notation, we introduce a new variable, fuel elements mass f, defined as

$$f = \rho\xi.$$

5.4.2b Algorithm Overview

First, fuel elements mass at the new time is determined from the mixture fraction equation (2.48):

$$f_i^{n+1} = f_i^n - \frac{\Delta t}{\Delta x}\left(f_R^n u_{i+\frac{1}{2}}^n - f_L^n u_{i-\frac{1}{2}}^n\right) - \frac{\Delta t}{\Delta x}\frac{1}{\mathrm{LePrRe}}\left(J_{i+\frac{1}{2}}^n - J_{i-\frac{1}{2}}^n\right), \quad (5.21)$$

with

$$J_{i+\frac{1}{2}}^n = -\left(\rho D\right)_{i+\frac{1}{2}}^n \frac{\xi_{i+1}^n - \xi_i^n}{\Delta x}.$$

The prediction of velocity is equal to the single fluid ideal gas case. The correction step, however, is different. This step is constructed by combination of the discrete continuity and mixture fraction equation according to (5.20). The two discrete equations are:

$$\begin{aligned}
\rho_i^{n+2} &= \rho_i^{n+1} - \frac{\Delta t}{\Delta x}\left(\rho_R^{n+1} u_{i+\frac{1}{2}}^{n+1} - \rho_L^{n+1} u_{i-\frac{1}{2}}^{n+1}\right) \\
f_i^{n+2} &= f_i^{n+1} - \frac{\Delta t}{\Delta x}\left(f_R^{n+1} u_{i+\frac{1}{2}}^{n+1} - f_L^{n+1} u_{i-\frac{1}{2}}^{n+1}\right) - \frac{\Delta t}{\Delta x}\frac{1}{\mathrm{RePrLe}}\left(J_{i+\frac{1}{2}}^{n+1} - J_{i-\frac{1}{2}}^{n+1}\right).
\end{aligned}$$

Combination yields the constraining equation for u^{n+1}:

$$\frac{u_{i+\frac{1}{2}}^{n+1} - u_{i-\frac{1}{2}}^{n+1}}{\Delta x} = RHS_i,$$

with

$$RHS_i = -\frac{1}{\mathrm{RePrLe}}\left(\frac{1}{\rho_A} - \frac{1}{\rho_B}\right)\frac{J_{i+\frac{1}{2}}^{n+1} - J_{i-\frac{1}{2}}^{n+1}}{\Delta x}.$$

This results in a variable coefficient Poisson equation for the pressure that is formally the same as in the case of a single fluid ideal gas, eq. (5.19):

$$-\frac{\Delta t}{\Delta x^2}\left(\frac{p'_{i+1} - p'_i}{\rho_{i+\frac{1}{2}}^{n+1}} - \frac{p'_i - p'_{i-1}}{\rho_{i-\frac{1}{2}}^{n+1}}\right) = RHS_i - \frac{u_{i+\frac{1}{2}}^* - u_{i-\frac{1}{2}}^*}{\Delta x}. \quad (5.22)$$

5.4.3 Two-Fluid Flow: Non-Premixed Combustion

5.4.3a Constraining Equation

In case of reacting flows, the same chemical operator is used, but no analytical expression is provided. So, the general constraint for reacting flows takes the form:

$$\rho_i^{n+2} = \mathcal{H}_C\left((f)_i^{n+2}\right). \tag{5.23}$$

Notice that, in contrast to inert mixing, the chemical operator is highly non-linear.

5.4.3b Algorithm Overview

The algorithm[2] is exactly the same as in the case of inert mixing, except for the correction step, following from constraint (5.23):

$$\begin{aligned}&\rho_i^{n+1} - \Delta t\frac{\rho_R^{n+1}u_{i+\frac{1}{2}}^{n+1} - \rho_L^{n+1}u_{i-\frac{1}{2}}^{n+1}}{\Delta x} = \\ &\mathcal{H}_C\left(f_i^{n+1} - \Delta t\frac{f_R^{n+1}u_{i+\frac{1}{2}}^{n+1} - f_L^{n+1}u_{i-\frac{1}{2}}^{n+1}}{\Delta x} - \frac{\Delta t}{\mathrm{RePrLe}}\frac{J_{i+\frac{1}{2}}^{n+1} - J_{i-\frac{1}{2}}^{n+1}}{\Delta x}\right).\end{aligned} \tag{5.24}$$

Inserting $u^{n+1} = u^* + u'$, yields

$$\rho_i^* + \rho_i' = \mathcal{H}_C\left(f_i^* + f_i'\right), \tag{5.25}$$

with

$$\rho_i^* = \rho_i^{n+1} - \Delta t\frac{\rho_R^{n+1}u_{i+\frac{1}{2}}^* - \rho_L^{n+1}u_{i-\frac{1}{2}}^*}{\Delta x} \tag{5.26}$$

$$f_i^* = f_i^{n+1} - \Delta t\frac{f_R^{n+1}u_{i+\frac{1}{2}}^* - f_L^{n+1}u_{i-\frac{1}{2}}^*}{\Delta x} - \frac{\Delta t}{\mathrm{RePrLe}}\frac{J_{i+\frac{1}{2}}^{n+1} - J_{i-\frac{1}{2}}^{n+1}}{\Delta x} \tag{5.27}$$

$$\rho_i' = -\Delta t\frac{\rho_R^{n+1}u_{i+\frac{1}{2}}' - \rho_L^{n+1}u_{i-\frac{1}{2}}'}{\Delta x} \tag{5.28}$$

$$f_i' = -\Delta t\frac{f_R^{n+1}u_{i+\frac{1}{2}}' - f_L^{n+1}u_{i-\frac{1}{2}}'}{\Delta x}. \tag{5.29}$$

[2]The algorithm for reacting flows, with the introduction of the chemical operator, was firstly published in brief form in [59].

Remark that we obtain a non-linear equation in velocity (or pressure), so that an iteration procedure is necessary. Therefore, (5.25) can be linearised around f^*, resulting in

$$\rho_i^* + \rho_i' = \mathcal{H}_C\left(f_i^*\right) + \frac{\mathrm{d}\mathcal{H}_C}{\mathrm{d}f}\left(f_i^*\right) f_i'. \tag{5.30}$$

(5.30) can be written in system notation, with matrices A and B, pressure-correction vector $\vec{p'}$:

$$\left(A - \frac{\mathrm{d}\mathcal{H}_C}{\mathrm{d}f}\left(\vec{f^*}\right) B\right) \vec{p'} = RHS, \tag{5.31}$$

with $RHS = \mathcal{H}_C\left(\vec{f^*}\right) - \vec{\rho}^*$. All vectors have the dimension of the number of grid nodes. The system is solvable at low cost if the matrices A and B do not change during iteration. Unfortunately, this is not true, since the matrices are composed of extrapolated values of density and fuel mass, which depend on the sign of the unknown velocity u^{n+1}. The same holds for the RHS, which value also depends on the sign of u^{n+1}. Since this influence is only secondary, a minor assumption could be introduced at this level, still preserving the consistency of the algorithm: if the extrapolated values of ρ and f are calculated, based on the sign of u^*, instead of u^{n+1}, matrices A and B and vector RHS only need one calculation per timestep, saving computing time. Since, strictly speaking, the monotonicity of the spatial discretisation is then no longer guaranteed, this assumption is not used in the simulation results.

For an internal node, eq. (5.31) yields:

$$\begin{aligned}\frac{(\Delta t)^2}{(\Delta x)^2}\Bigg\{&\left[\rho_R^{n+1} - \frac{\mathrm{d}\mathcal{H}_C}{\mathrm{d}f}\left(f_i^*\right) f_R^{n+1}\right]\frac{p'_{i+1} - p'_i}{\rho_{i+\frac{1}{2}}^{n+1}} \\ &- \left[\rho_L^{n+1} - \frac{\mathrm{d}\mathcal{H}_C}{\mathrm{d}f}\left(f_i^*\right) f_L^{n+1}\right]\frac{p'_i - p'_{i-1}}{\rho_{i-\frac{1}{2}}^{n+1}}\Bigg\} \;=\; \mathcal{H}_C\left(f_i^*\right) - \rho_i^*.\end{aligned} \tag{5.32}$$

5.4.3c Remarks

We first make some remarks for 1D flows:

- The momentum equation becomes of no interest, since the velocity field follows from the constraining equation: the velocity field is chosen in such a way that the new state (ρ, f) is physically correct. This means that we require (ρ, f) to be on the curve, defined by the chemical properties.

- It is instructive to note that the ultimate algorithm can be illustrated in a geometrical way. For each cell, one of the two face-velocities ($u_{i-\frac{1}{2}}$ or $u_{i+\frac{1}{2}}$) can be treated as known (because of a velocity inlet somewhere in the domain), such that the other one is essentially the only unknown in the constraining equation. The unknown velocity can be determined geometrically, in the (ρ, f)-plane (Fig. 5.1). Therefore the constraint is rewritten as a combination of 2 equations:

$$\begin{pmatrix} \rho \\ f \end{pmatrix}_i^{n+1} = \begin{pmatrix} \rho \\ f \end{pmatrix}_i^n - \frac{\Delta t}{\Delta x} u_{i+\frac{1}{2}}^n \begin{pmatrix} \rho \\ f \end{pmatrix}_R^n + \frac{\Delta t}{\Delta x} u_{i-\frac{1}{2}}^n \begin{pmatrix} \rho \\ f \end{pmatrix}_L^n + \begin{pmatrix} 0 \\ \mathcal{D} \end{pmatrix}, \quad (5.33)$$

  which has, in case of the simplified Flame-Sheet Model, eq. (3.9), for a given state (ρ^n, f^n), and one given velocity, only one physically possible solution.

$$\mathcal{D} = -\frac{\Delta t}{\mathrm{RePrLe}} \frac{J_{i+\frac{1}{2}}^n - J_{i-\frac{1}{2}}^n}{\Delta x}$$

  denotes the diffusive part.

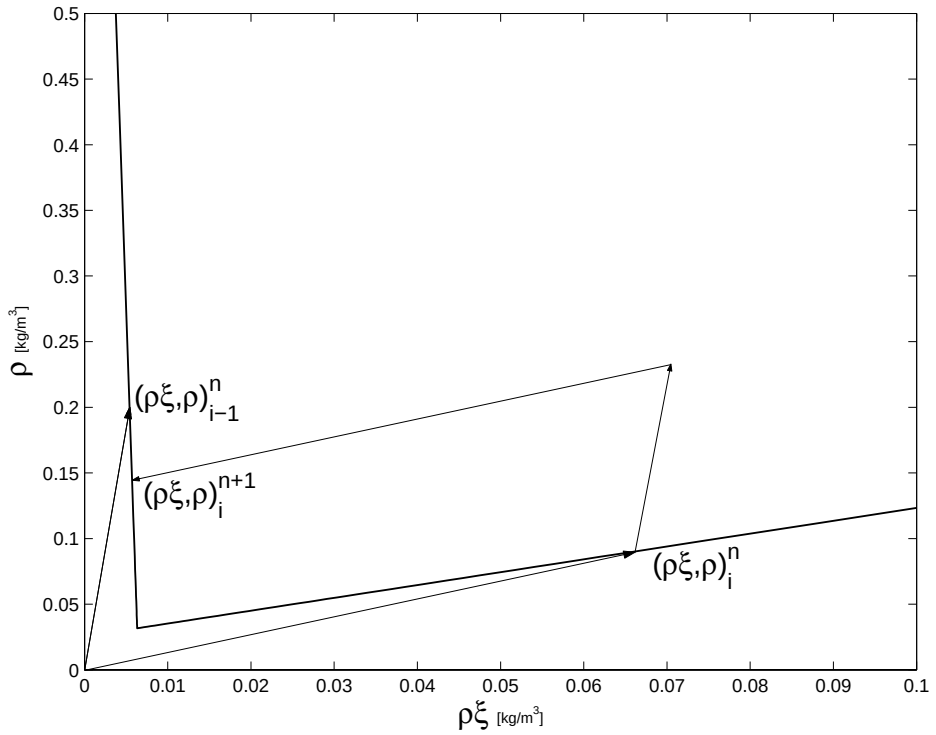

Figure 5.1: Geometrical solution, in case of $\mathcal{D} = 0$ and a first order upwind scheme, with positive velocity u, so that $L = i - 1$ and $R = i$. If $u_{i-\frac{1}{2}}$ is known, the magnitude of $u_{i+\frac{1}{2}}$ can be derived from the vector that intersects the chemical $(\rho, \rho\xi)$ dependency.

A few general remarks must be made as well:

- A problem occurs when the general Flame-Sheet model is implemented. We recall the example of the pure methane-oxygen flame of section 3.1.2b. For states close to stoichiometry, the density is not uniquely determined as a function of $(\rho\xi)$. This is a problem, since the algorithm requires the evaluation of $\mathcal{H}_C(f^*)$. It is possible, however, to perform a coordinate transformation, such that the uniqueness is guaranteed. The transformation is done from (ρ, f) to (ρ, ϕ), where ϕ is a linear combination of ρ and f: $\phi = \rho - f/\xi_{st}$, with governing equation:

$$\frac{\partial \phi}{\partial t} + \frac{\partial \phi u_i}{\partial x_i} = \frac{1}{\mathrm{RePrLe}} \frac{\partial}{\partial x_i} \left(\rho D \frac{\partial \xi}{\partial x_i} \right). \tag{5.34}$$

The resulting diagram is depicted in fig. 5.2. Another possibility is to use the

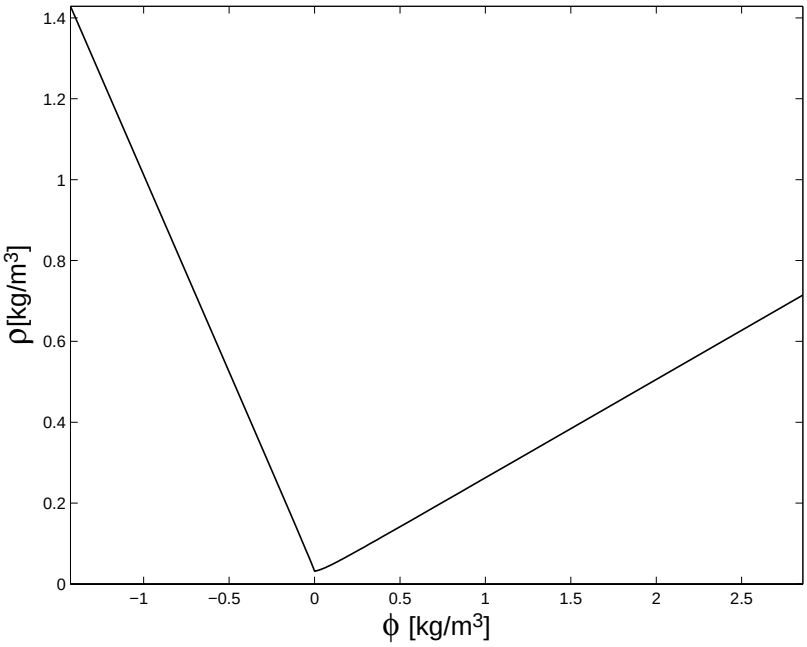

Figure 5.2: Density as a function of $\phi = \rho - f/\xi_{st}$ for pure methane-oxygen combustion: the density is uniquely determined.

functional $\mathcal{H}$, rather than $\mathcal{H}_C$. In that case, a solution can always be obtained, following the general strategy of section 5.3. It is then not required that the density is a function[3] of the other scalars (here: fuel elements mass); the knowledge of the general functional $\mathcal{H}(\rho, \rho\xi)$ is sufficient to obtain a solution.

- In absence of diffusion, the velocity field can be determined, independent of the size of time step $\Delta t_{n+1 \to n+2}$. However, if $\mathcal{D} \neq 0$, the value of the next time step is required to determine the velocity. Since, for the sake of stability, the maximum

[3]The term *function* indicates that there exists a one-to-one relationship.

allowable time step $\Delta t_{n+1\to n+2}$ depends on the magnitude of the velocity u^{n+1}, we do not know it a priori. In many simulations, the allowable time step does not change much during the computation, so that it can be fixed to a value, without great danger for instability, corresponding to the velocity at time level n.

- In the present approach, all conservation properties are exactly fulfilled, together with the equation of state (in contrast to what is stated in [20]). However, the equation of state is only exact, up to the convergence criterion for the Poisson solver, which is normally small compared to the discretisation errors.
- The present algorithm is not restricted to first order time accuracy. A multistage version of the algorithm is given in Appendix B.

5.5 Recapitulation

The pressure-correction scheme is summarised in the following flow chart.

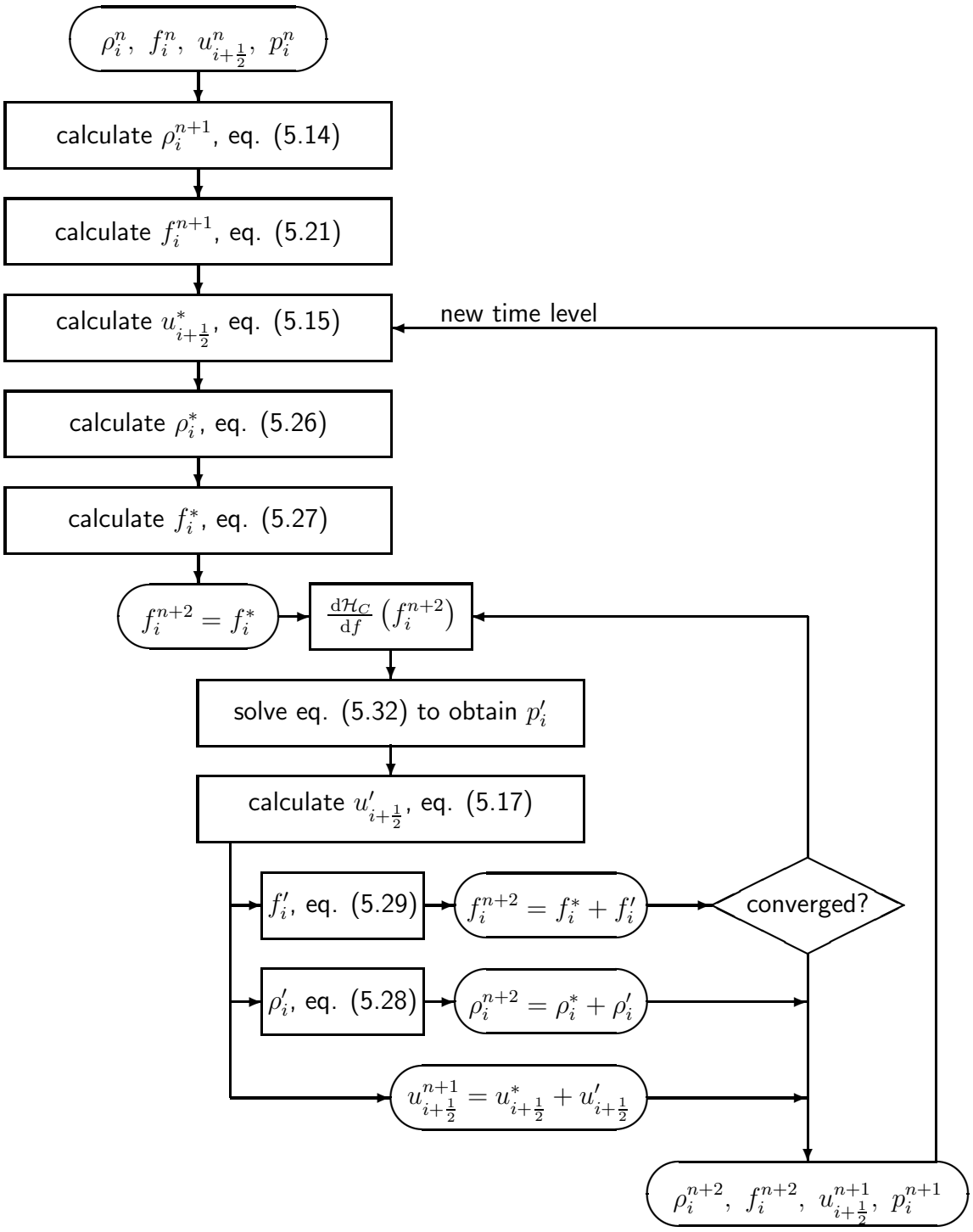

Figure 5.3: Summary of the discrete compatibility-constraint pressure-correction algorithm for two-fluid flow.

Chapter 6

Comparison to Existing Algorithms

Time accurate simulations of variable density flows are not new. Many papers have been published, concerning algorithmic development for transient flow calculations, even in the field of combustion. In this chapter, we try to divide the most important papers into various classes.

The division into classes is not a simple task, because of the following reasons:

- A lot of papers exist on the development of unified methods for flows at all speeds. The basic equations in this case are the Navier-Stokes equations in their most general formulation. Since the pressure, that in this case does not fall into two parts, appears also in the energy equation, a stronger coupling exists between the separate equations and a more careful correction step is needed. If we reduce the existing algorithms to low Mach number flow, they basically reduce to the same algorithmic class.
- A lot of effort is spent on efficient construction of higher order pressure-correction algorithms. As a result, the number of substeps and pressure equations to solve increases dramatically. If we reduce all algorithms to their lowest order formulation, mostly they all end up the same.
- No algorithm is really the same. But the difference is mostly subtle. We tried, however, to make abstraction of the details of implementation and tried to look at the conceptual interpretation, mostly hidden behind the algorithm.

In the end, we distinguish between two general strategies, being the 'coupled'[1] and the 'segregated' approach. Since the contents of this work is on segregated algorithms, the segregated approach is further explored and distinction is made basically based on the way the constraining equation for the velocity is derived. We conclude that all available

[1]The coupled approach is also named in literature 'density-based' algorithm.

methods are based on the analytical differential equations, whereas the algorithm described in this work takes as a starting point the equations in discretised form, resulting in a pressure-correction method with better properties.

6.1 Coupled Solution Methods

If a low Mach number flow is simulated, using the compressible formulation of the flow equations, a basic difficulty stems from the acoustic waves. As acoustic waves act at a substantially different timescale than the convective phenomena in low Mach number flows, the acoustic modes do not significantly alter the solution and may be regarded as superfluous. The use of larger time steps, corresponding to the convective scales, can therefore strongly improve a method's efficiency without loss of interesting information. Since, on the other hand, the increase of the time step may jeopardise stability, some special precautions must be taken.

Numerical efficiency can be increased, modifying the acoustic time step limit by means of preconditioning techniques. However, if time-accuracy has to be respected, classical preconditioning methods do not apply. In [81], an artificial acoustic stiffness reduction is suggested through the introduction of an additional term in the energy equation, so that the speed of sound is effectively reduced and pressure gradients remain well controlled. Another manner is to solve the system implicitly, so that the time step limit is avoided. Intensive dual time-stepping methods [28] are indeed successfully applied: the implicit system is solved in pseudo-time, using all kinds of acceleration techniques (multistage, preconditioning, multigrid, etc.).

Although we are convinced that a segregated solution procedure is more efficient in low Mach number flows [47], there are researchers that use coupled solvers with the argument of its simplicity of implementation [17, 74]. When trying to use these approaches in combination with a flamelet library for turbulent non-premixed combustion, there are, however, issues that require specific caution, e.g. with respect to the boundary conditions [72].

6.2 Segregated Solution Methods

The segregated algorithms are all extensions of Chorin's constant density pressure-correction formalism [13, 14]. Although the extension seems straightforward, special attention is needed, specifically on the construction of a constraint for the velocity field. A first way of deriving the constraint is based solely on the continuity equation. The corresponding scheme is called here the *continuity-constraint pressure-correction* algorithm. It can easily

be adopted to any kind of flow type, no matter how many scalar transport equations are solved. It suffers, however, from instability problems when density ratios are too high. A second scheme, the *analytical compatibility-constraint pressure-correction* scheme, derives the constraint essentially from the equation of state, which is combined with the transport equations for mass and scalars, to yield a constraint on the velocity. Even for high density ratios, the scheme provides stable results. However, in its original formulation, the equation of state cannot be satisfied in an exact manner. The third and last scheme, the *discrete compatibility-constraint pressure-correction* scheme, is newly developed in the previous chapter.

6.2.1 Continuity-Constraint Pressure-Correction

6.2.1a Algorithm Overview

In the standard formulation, momentum, rather than velocity, is used as a primitive variable. First, a predictor step for the momentum is taken, with the value of the pressure obtained from the previous time step:

$$\begin{aligned}(\rho u)^{*}_{i+\frac{1}{2}} &= (\rho u)^{n}_{i+\frac{1}{2}} \quad -\frac{\Delta t}{\Delta x}\left((\rho u)^{n}_{R}\, u^{n}_{i+\frac{1}{2}} - (\rho u)^{n}_{L}\, u^{n}_{i-\frac{1}{2}}\right) - \frac{\Delta t}{\Delta x}\left(p^{n}_{i+1} - p^{n}_{i}\right) \\ &\qquad\qquad +\Delta t\frac{\delta \tau^{n}}{\delta x}. \end{aligned} \tag{6.1}$$

The equation to be ultimately solved, includes the pressure at the new time level:

$$\begin{aligned}(\rho u)^{n+1}_{i+\frac{1}{2}} &= (\rho u)^{n}_{i+\frac{1}{2}} \quad -\frac{\Delta t}{\Delta x}\left((\rho u)^{n}_{R}\, u^{n}_{i+\frac{1}{2}} - (\rho u)^{n}_{L}\, u^{n}_{i-\frac{1}{2}}\right) - \frac{\Delta t}{\Delta x}\left(p^{n+1}_{i+1} - p^{n+1}_{i}\right) \\ &\qquad\qquad +\Delta t\frac{\delta \tau^{n}}{\delta x}, \end{aligned}$$

so that momentum and pressure corrections are related through:

$$(\rho u)'_{i+\frac{1}{2}} = -\Delta t\frac{p'_{i+1} - p'_{i}}{\Delta x}, \tag{6.2}$$

with $(\rho u)' = (\rho u)^{n+1} - (\rho u)^{*}$ and $p' = p^{n+1} - p^{n}$.

The constraint on the momentum is based on the continuity equation, with the spatial

derivatives discretised at time level $n+1$:

$$\frac{\rho_i^{n+1}-\rho_i^n}{\Delta t} = -\frac{(\rho u)_{i+\frac{1}{2}}^{n+1}-(\rho u)_{i-\frac{1}{2}}^{n+1}}{\Delta x}, \tag{6.3}$$

or, in terms of momentum corrections:

$$\frac{(\rho u)'_{i+\frac{1}{2}}-(\rho u)'_{i-\frac{1}{2}}}{\Delta x} = -\frac{\rho_i^{n+1}-\rho_i^n}{\Delta t} - \frac{(\rho u)^*_{i+\frac{1}{2}}-(\rho u)^*_{i-\frac{1}{2}}}{\Delta x}.$$

This results in a constant coefficient Poisson equation for the pressure:

$$-\frac{\Delta t}{\Delta x^2}\left(p'_{i+1}-2p'_i+p'_{i-1}\right) = -\frac{\rho_i^{n+1}-\rho_i^n}{\Delta t} - \frac{(\rho u)^*_{i+\frac{1}{2}}-(\rho u)^*_{i-\frac{1}{2}}}{\Delta x} \tag{6.4}$$

The unknown value ρ_i^{n+1} in eq. (6.4) follows from stepping of the scalar equations, e.g. temperature:

$$T_i^{n+1} = T_i^n - \Delta t u_i^n \frac{T^n_{i+\frac{1}{2}}-T^n_{i-\frac{1}{2}}}{\Delta x} - \frac{\Delta t}{\rho_i^n}\frac{1}{\mathrm{RePr}}\frac{q^n_{i+\frac{1}{2}}-q^n_{i-\frac{1}{2}}}{\Delta x} \tag{6.5}$$

or mixture fraction:

$$f_i^{n+1} \;=\; f_i^n - \frac{\Delta t}{\Delta x}\left((\rho u)^n_{i+\frac{1}{2}}\,\xi_R^n - (\rho u)^n_{i-\frac{1}{2}}\,\xi_L^n\right) - \frac{\Delta t}{\Delta x}\frac{1}{\mathrm{LePrRe}}\left(J^n_{i+\frac{1}{2}} - J^n_{i-\frac{1}{2}}\right) \tag{6.6}$$

and the equation of state, e.g.

$$\rho_i^{n+1} = \frac{p_0^{n+1}}{T_1^{n+1}} \tag{6.7}$$

or

$$\rho_i^{n+1} = \mathcal{H}_C\left(f_i^{n+1}\right). \tag{6.8}$$

The sequence of steps in the continuity-constraint pressure-correction algorithm is now as follows:

1. advance the scalar equations, e.g. (6.5), to obtain the scalars at the new time level;
2. use the equation of state, e.g. (6.7), to determine the density at the new time level;
3. advance the momentum equation (6.1) for momentum prediction;
4. solve the pressure Poisson equation (6.4) to obtain the pressure corrections;
5. correct the momentum with the pressure corrections (6.2).

6.2.1b Literature Review

The numerically attractive continuity-constraint pressure-correction algorithm has been widely used in literature and still forms the basis of many simulation codes. Apparently, it is reported in literature that this algorithm behaves unstable if density ratios exceed a factor of about 3, depending on the actual implementation details. One may thus find it surprising that, precisely in the combustion community, where sharp gradients in density are common[2], this algorithm is seen as the standard method for solving low Mach number reacting flows. The use of nested predictor-corrector methods or rescaling factors is needed to stabilise the solution, up to a higher density ratio, as is discussed next. These cures, however, cannot guarantee stability in all cases and harm the consistency of the discretised equations.

Cook et al. [16] found that a second-order approximation to the time derivative of the density in the constraining continuity equation was significantly more stable than a third order approximation. A third order Adams-Bashforth time stepping algorithm was reported to be stable for maximum density changes up to about a factor of 3. For higher order density ratios, they propose the use of a predictor-corrector method such as a second or third order Runge Kutta.

Najm et al. [46] propose a predictor-corrector scheme, requiring two inversions of the constant coefficient Poisson equation each timestep. If only the predictor step would have been involved, the scheme is found to be conditionally stable for density ratios up to a factor of 2. Adding the corrector step, allows for higher density ratios.

The above conclusions are summarised in [48]. They serve as the author's reason for using a variable coefficient Poisson equation, which can handle density ratios much larger than 3. Furthermore, it is shown that the use of a constant coefficient Poisson operator, as approximation to the (correct) variable coefficient equation for the pressure, results in errors that increase with increasing density ratio and ultimately destabilises the solution.

[2] Especially near the flame front, a density ratio of 8 is not unusual for e.g. atmospheric methane-air combustion.

In [24], the constant coefficient Poisson solver is also found to become unstable for large density ratios. Artificial stabilisation is introduced in the pressure equation by means of an (unphysical) rescaling of the density time derivative. Note that the rescaling factor has to be taken two orders of magnitude smaller than 1. Improvements are succesfully proposed in [33] to reduce the impact of the artificial rescaling terms on the consistency of the algorithm, but no really satisfactory cure is found.

A constant coefficient Poisson equation in a predictor-corrector formalism is proposed in [39]. The authors claim stable results, which are due to the predictor-corrector formalism and the use of the energy equation (however, in non-conservative form) together with the equation of state for evaluation of the density. Results for (global) density ratios up to 8 are shown.

A method for premixed combustion simulations can be found in [43], using a ghost-fluid method, resulting in a constant coefficient Poisson equation. The resulting scheme is only stable, however, because a good approximation for the jump is found[3]. This clever algorithm is only applicable in premixed combustion, and is (as far as we can think of) not readily extendable to non-premixed combustion simulations.

In [33], another issue related to the continuity-constraint pressure-correction algorithm is addressed. The algorithm requires an evaluation of the density, based on the scalars through the equation of state. It is therefore necessary that the equation of state can be written under the formulation

$$\rho^{n+1} = \mathcal{H}_C\left((\rho\xi)^{n+1}\right).$$

In certain cases, the relationship between ρ and $\rho\xi$ is not unique (see fig. 3.2) and a special *splitting*-procedure is to be adopted, which makes the continuity-constraint algorithm less reliable for these cases, unless extra substeps are included in the overall algorithm.

The method can be extended to compressible flows. Early work [30] solves the steady compressible Navier-Stokes equations, using a continuity-based constraint. Transient variants can be found in [6, 80]. They all use the pressure to correct the predicted momentum field, eq. (6.2). In high speed flows, the use of the SIMPLE[4] algorithm [51], which is able to treat convective terms implicity, can increase the efficiency. It is indeed the CFL-number, based on the speed of sound, that imposes a restriction on the time step. An implicit treatment of the convective terms thus allows for a larger time step. In the SIMPLE-algorithm it is more convenient to use the pressure to correct the predicted velocity field, eq. (5.17), rather than the momentum field. Variants using velocity correction are

[3] The jump is equivalent to the difference between a variable and a constant coefficient Poisson equation, and the behaviour of this jump at the next time step can be approximated in a good manner because the turbulent flame speed is known.

[4] Semi-Implicit Method for Pressure Linked Equations

[40, 42, 66]. When dealing with low Mach number flows, the use of a velocity correction has no advantages; on the contrary, in that case a variable coefficient Poisson equation has to be inverted, requiring more computational time.

6.2.1c Summary

In the continuity-constraint pressure-correction algorithm, a constraint for the momentum field, originating from the continuity equation, is used to construct a pressure Poisson equation. This method is regularly applied in the combustion community because of its properties of

- (fuel elements) mass conservation;
- minimal computational effort for inverting the constant coefficient Poisson equation;
- ease to implement in the flamelet assumption strategy.

There are however issues concerning

- stability in the presence of high density ratios;
- determination of density in terms of other scalars (splitting).

The stability issues are ameliorated (not solved!) by using

- predictor-corrector formalisms;
- inconsistent rescaling factors,

but no really satisfactory cure is found.

6.2.2 Analytical Compatibility-Constraint Pressure-Correction

6.2.2a Algorithm Overview

In the analytical compatibility-constraint pressure-correction scheme, a constraint is formulated, stating that the variables, associated with a fluid particle, should change according

to the equation of state. In practice, this constraint is formulated in a differential formulation, using the Lagrangian[5] derivative, defined as

$$\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + u_i \frac{\partial \phi}{\partial x_i}.$$

For a fluid particle, the equation of state should be fulfilled, thus also its material derivative:

$$\begin{aligned} \frac{D}{Dt}\mathcal{G}\left(\rho, y_\alpha\right) &= 0 \\ \Leftrightarrow \frac{D\rho}{Dt}\frac{\partial \mathcal{G}}{\partial \rho} + \frac{Dy_\alpha}{Dt}\frac{\partial \mathcal{G}}{\partial y_\alpha} &= 0. \end{aligned}$$

The Lagrangian derivatives of the density and scalars follow from their differental transport equations, reformulated as:

$$\begin{aligned} \frac{D\rho}{Dt} &= -\rho \frac{\partial u_i}{\partial x_i} \\ \frac{Dy_\alpha}{Dt} &= \frac{RHS_\alpha}{\rho}, \end{aligned}$$

so that a constraining equation for the velocity yields:

$$\frac{\partial \mathcal{G}}{\partial \rho}\frac{\partial u_i}{\partial x_i} = \frac{RHS_\alpha}{\rho^2}\frac{\partial \mathcal{G}}{\partial y_\alpha}. \tag{6.9}$$

We can use this constraint to obtain the pressure correction equation:

$$\begin{aligned} -\frac{\Delta t}{\Delta x^2}\left(\frac{p'_{i+1}-p'_i}{\rho^{n+1}_{i+\frac{1}{2}}} - \frac{p'_i - p'_{i-1}}{\rho^{n+1}_{i-\frac{1}{2}}}\right) &= \left[\frac{RHS_\alpha}{\rho^2}\left(\frac{\partial \mathcal{G}}{\partial y_\alpha}\right)\left(\frac{\partial \mathcal{G}}{\partial \rho}\right)^{-1}\right]_i \\ &\quad - \frac{u^*_{i+\frac{1}{2}} - u^*_{i-\frac{1}{2}}}{\Delta x}, \end{aligned} \tag{6.10}$$

where the first term in the right hand side of (6.10) yields, for a single fluid ideal gas:

$$-\frac{1}{\gamma}\frac{1}{p_0^{n+1}}\frac{p_0^{n+2}-p_0^{n+1}}{\Delta t} + \frac{1}{p_0^{n+1}}\frac{1}{\mathrm{RePr}}\frac{q^{n+1}_{i+\frac{1}{2}} - q^{n+1}_{i-\frac{1}{2}}}{\Delta x};$$

[5] According to a Lagrangian description of the fluid, where the equations are derived in a coordinate system associated with a fluid particle. The Lagrangian derivative is therefore also called *material* derivative.

for two-fluid inert mixing:

$$-\frac{1}{\mathrm{RePrLe}}\left(\frac{1}{\rho_A}-\frac{1}{\rho_B}\right)\frac{J^{n+1}_{i+\frac{1}{2}}-J^{n+1}_{i-\frac{1}{2}}}{\Delta x};$$

and for two-fluid non-premixed combustion:

$$-\frac{1}{\mathrm{RePrLe}}\frac{\frac{\mathrm{d}\mathcal{H}_C}{\mathrm{d}f}\left(\rho_i^{n+1}\xi_i^{n+1}\right)}{\rho_i^{n+1}-\rho_i^{n+1}\xi_i^{n+1}\frac{\mathrm{d}\mathcal{H}_C}{\mathrm{d}f}\left(\rho_i^{n+1}\xi_i^{n+1}\right)}\frac{J^{n+1}_{i+\frac{1}{2}}-J^{n+1}_{i-\frac{1}{2}}}{\Delta x}.$$

For the derivation of these expressions, we refer to appendix C.

In the construction of the algorithm itself, there are several variants, depending on which equations we want to satisfy. We follow here the approach of Bell [20], where conservation properties are considered to be more valuable than the fulfillment of the equation of state. In that case, we end up with the following substeps:

1. advance the continuity equation (5.7) to obtain the density at the new time level;
2. advance the scalar equations (5.8) to obtain the scalars at the new time level;
3. advance the momentum equation (6.1) for momentum prediction;
4. determine the predicted velocity (5.15);
5. solve the variable coefficient pressure Poisson equation (6.10) to obtain the pressure corrections;
6. correct the predicted velocity with the pressure correction (5.17).

6.2.2b Literature Review

For variable density flows, the compatibility-constraint pressure-correction method was introduced by Bell et al. in [5] as an extension of the constant density algorithm [3]. [3] is merely a second order extension of Chorin's original pressure projection formalism [14]. In the early versions [2, 5], not many physics were incorporated, resulting in the simple constraint that the velocity field must be solenoidal:

$$\frac{\partial u_i}{\partial x_i}=0.$$

As more challenging flows are examined [4, 20], the constraint becomes more complicated, eq. (6.9). The use of a variable coefficient Poisson equation, provides a stable scheme, even for high density ratios[6]. As it is the authors' intention to obey the conservation laws (and thus conserve mass, enthalpy, species mass fractions exactly), they need to slightly modify the equation of state. Because of that, the solution drifts away from the equation of state, but the drift is controlled by means of a defect correction term [20].

In [69, 70] a variable coefficient Poisson equation is used, together with the equations of temperature, species mass fractions and velocity in advective form. As a result, the quantities are not conserved, but they claim an error in mass conservation, which is $\mathcal{O}(\Delta t^{J+1}/\mathrm{Re})$ for a J-th order time integration.

In [29, 63], the analytical compatibility-constraint method is adopted for time-accurate predictions, using implicit discretisations. In the proposed method, two elliptic pressure-equations must be solved each time step: one for the stepping over a half of the time step and another for a stepping over the total time step. A staggering procedure in space and in time is introduced.

The compatibility-constraint based approach can also be extended to flows at higher Mach number in the subsonic range [15, 45] or for flows at all speeds [71].

6.2.2c Summary

In the analytical compatibility-constraint pressure-correction algorithm, a constraint for the velocity field, originating from the Lagrangian derivative of the equation of state, in combination with the equations of continuity and scalar transport, is used to construct a pressure Poisson-like equation. This method is applied in certain research areas because of its properties of

- species mass fraction and energy conservation;
- stability in the presence of high density ratios.

There are however issues concerning

- the computational effort to invert the variable coefficient Poisson equation;
- the approximate fulfillment of the equation of state;
- the implementation strategy in case of a flamelet model assumption.

[6] Density ratios of 1:800 are simulated with success.

The fulfillment of the equation of state is ameliorated (not solved!) by using

- a defect correction term,

but this approach is not satisfactory for simulations using the flamelet model assumption, as will be illustrated in the next chapter.

6.2.3 Three Pressure-Correction Algorithms: Comparison

Comparison of the properties of the different pressure-correction schemes, yields following result:

	continuity-constr.	anal. compat.-constr.	discr. compat.-constr.
stable/consistent results		x	x
mass conservation	(x)	x	x
exact fulfillment of state eq.	x		x
constant coefficient Poisson eq.	x		

Table 6.1: Summary of the algorithms' properties. In case of the continuity-constraint pressure-correction, mass conservation is only guaranteed in open domains, hence the notation between brackets.

As shown in table 6.1, the discrete compatibility constraint pressure-correction scheme (chapter 5) yields stable results and provides a consistent solution, which is the most vital property missing in the continuity-constraint pressure-correction (section 6.2.1). Furthermore, all conservation properties are fulfilled, together with the equation of state. The latter is only approximately fulfilled in case of the analytical compatibility-constraint pressure-correction scheme (section 6.2.2). The only drawback of the discrete compatibility-constraint pressure-correction scheme is that the resulting pressure equation has variable coefficients that require several recalculations every time step. The extra cost, however, is marginal compared to the benefits associated with this algorithm: the higher robustness and the greater accuracy in terms of state prediction.

Chapter 7

One-Dimensional Test Cases

The properties of the pressure-correction algorithms from chapters 5 and 6 are now demonstrated for a few well-chosen and simple examples. In this chapter, we restrict ourselves to one-dimensional tests. Although some may find it dangerous to use a 1D configuration in order to qualify/disqualify numerical schemes, many valuable insights are given by these flow configurations. Especially these idealised test cases can give an honest answer to the question why common pressure-correction algorithms fail to give consistent or, at least, stable results[1]. We are also aware of the fact that algorithms might behave better in more realistic flows in more dimensions, but consider a succesful simulaton in 1D as a necessary condition for the eventual algorithm to yield stable and consistent solutions in any general variable-density problem.

The algorithm is put to two different kinds of tests, being:

- a purely convective transport of a sharp (density) gradient in a channel, and
- a purely diffusive transport in the appearance of the same gradient.

A combination of these two tests, collects all possibilities in a 1D flow. The origin of the appearing gradients, depends on the fluid properties. Three kinds of fluids are considered for each of the tests:

- single fluid ideal gas;
- two fluid inert mixing;
- two fluid non-premixed combustion.

[1] It was in fact the study of these 1D flows that led to the development of the discrete compatibility-constraint pressure-correction algorithm.

The 3 major pressure correction algorithms are used in the simulations:

- continuity-constraint pressure-correction;
- analytical compatibility-constraint pressure-correction;
- discrete compatibility-constraint pressure-correction.

If every possible combination is considered, 2x3x3=18 simulations have to be performed.

The test cases in this chapter use a staggered grid topology to cope with possible odd-even decoupling. The transient calculations are performed using a time step of 0.9 times the maximum allowable time step for stability, according to a linear stability analysis (see appendix A). For each test, results after 1 and 10 time steps are displayed to illustrate the time-accurate property of the different algorithms.

7.1 One-Dimensional Pure Convection

In a first test, pure convection is considered: all diffusive terms (conduction and species diffusion) are set to zero. In the resulting test case, a step in the scalar variable ϕ (temperature or mixture fraction) is convected in a straight channel. The initial step[2] is defined in space as the piecewise constant function

$$\phi_i = \begin{cases} \phi_1 & \text{for} \quad i \in [1, i_1[\\ \phi_2 & \text{for} \quad i \in [i_1, i_2] \\ \phi_1 & \text{for} \quad i \in]i_2, N_x] \end{cases} \tag{7.1}$$

In this case, an analytical solution exists: the velocity in every section of the channel should remain equal to the imposed inlet velocity and the scalar field is shifted in space, over a distance u/t, with $u = 1$ the inlet velocity and t the simulated time. In the problem considered, $N_x = 50$ grid points were used and the step in the scalar field is situated in the grid node interval $[10, 30]$. The grid spacing is set to 1, so that the time step is calculated as:

$$\Delta t = 0.9/u_{max}. \tag{7.2}$$

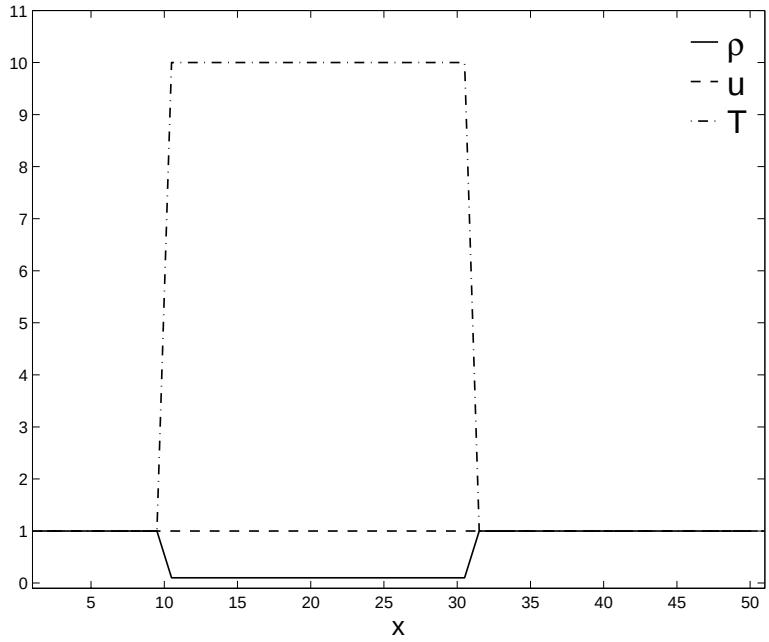

Figure 7.1: Purely convective transport of a density jump in a straight channel, filled with an ideal gas at different temperatures: initial condition.

7.1.1 Single Fluid Flow: Ideal Gas

The initial velocity, temperature and density field is depicted in fig. 7.1. A temperature step with a step factor of 10 is used, resulting in a density field, having the same initial ratio.

7.1.1a Continuity-Constraint Pressure-Correction

Fig. 7.2 shows that the continuity-constraint pressure-correction scheme gives inaccurate predictions for the velocity field (dashed line), even in regions far away from the density jump ($x >> 30$). This observation follows from the corrector step (6.3), imposing mass conservation:

$$\frac{\rho_i^{n+1} - \rho_i^n}{\Delta t} = -\frac{(\rho u)_{i+\frac{1}{2}}^{n+1} - (\rho u)_{i-\frac{1}{2}}^{n+1}}{\Delta x}.$$

[2]Note that the initial step becomes smoother after a few time steps because of numerical diffusion, so that the convection of a smooth profile is embedded in this test case.

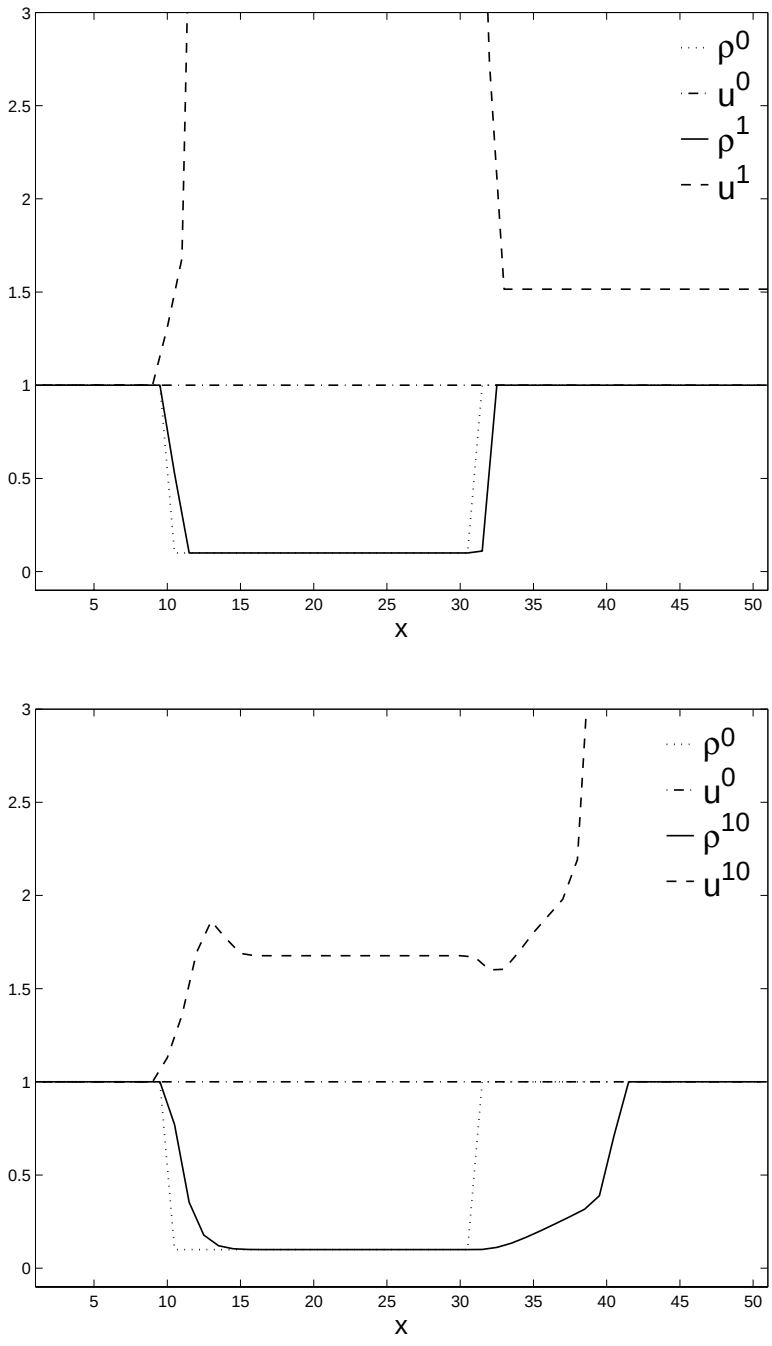

Figure 7.2: Density and velocity fields, obtained with the continuity-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure convection of an ideal gas at different temperatures. Results for the velocity field are inaccurate, even in the region far from the density jump ($x >> 30$).

Since the density field at the new time level follows from the non-conservative discretisation (6.5), mass is conserved through the adjustment of the outlet velocity. As a consequence, errors near sharp gradients do not only have a local impact, but also have a major influence in the entire domain.

The algorithm is not only inaccurate, it is also inconsistent, since a grid refinement is not able to overcome the inaccuracy. The difference with the exact solution is even so extreme, that the simulation does not remain stable, if continued in time.

Since the CFL-number is kept constant during the time stepping, with a large maximum value for the velocity in the domain, the actual time evolved is small, which explains why the first density gradient is still around node 10, even after 10 time steps.

7.1.1b Analytical Compatibility-Constraint Pressure-Correction

With the analytical compatibility-constraint pressure-correction algorithm, the velocity field does not differ from the exact solution and remains constant throughout the iterations (fig. 7.3). This is not surprising, since the compatibility constraint for non-diffusive flows simplifies into $\nabla \cdot \boldsymbol{u} = 0$, for a 1D problem resulting in a constant velocity field. The density field does show some deviation from the exact solution, but this observation is merely due to the use of a first order upwind scheme for the convective fluxes.

Note that, in contrast to the previous simulation, the time step does not vary significantly, so that the first step in temperature is already convected further downstream.

7.1.1c Discrete Compatibility-Constraint Pressure-Correction

Since in the case of an ideal gas, the constraining equation for the compatibility-constraint pressure-correction does not differ whether the analytical or the discrete derivation is used, the same results as fig. 7.3 are obtained.

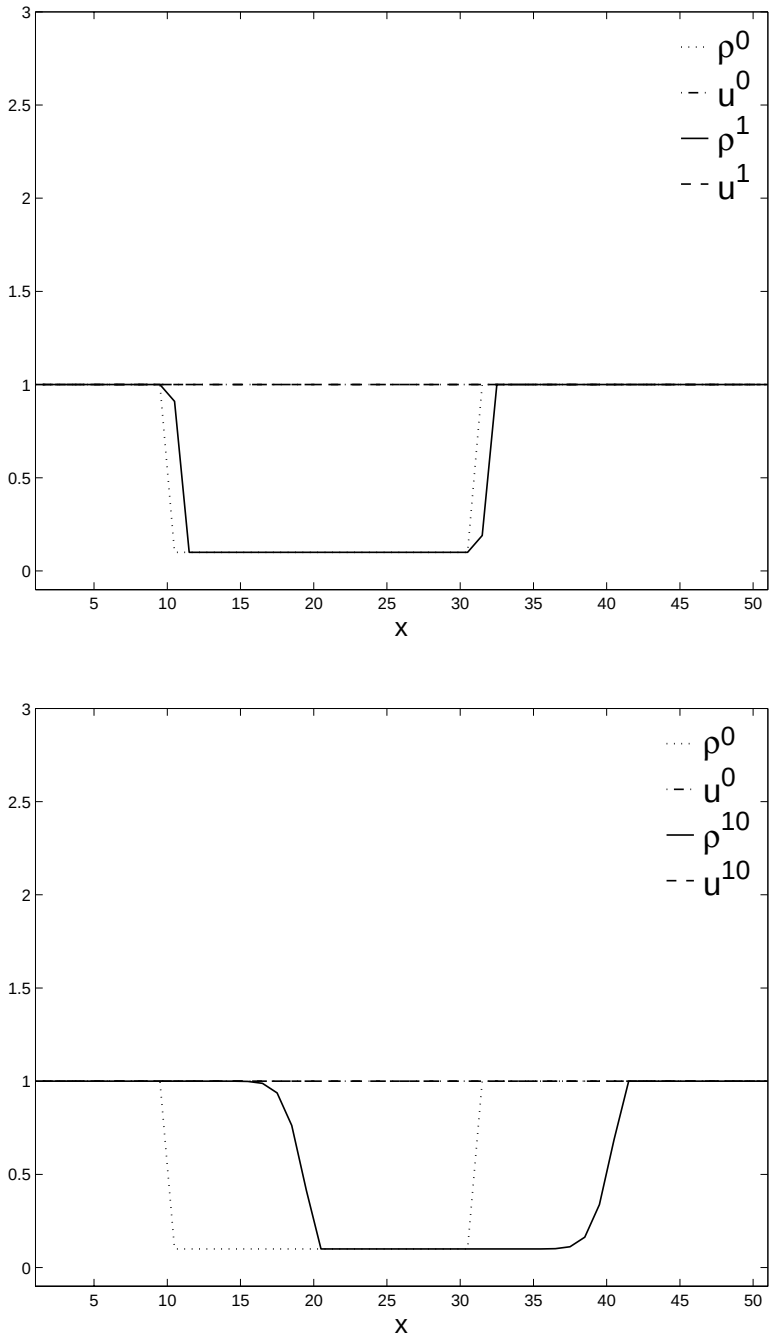

Figure 7.3: Density and velocity fields, obtained with the compatibility-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure convection of an ideal gas at different temperatures. The obtained results are equal for the analytical and the discrete version of the algorithm. A consistent prediction of the velocity is obtained.

7.1.2 Two Fluid Flow: Inert Mixing

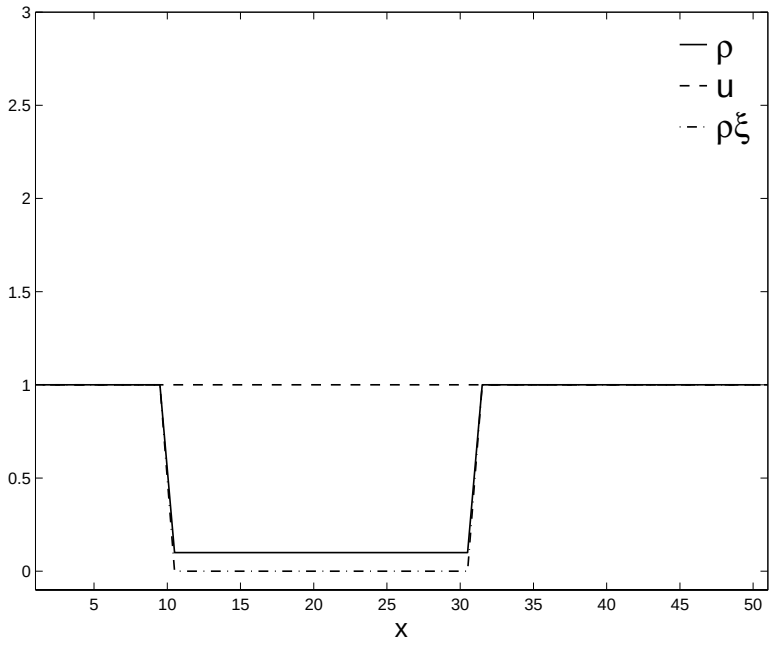

Figure 7.4: Purely convective transport of two inert mixing fluids with different density in a straight channel: initial condition.

The initial velocity, fuel elements mass and density field is depicted in fig. 7.4. Initially, two fluids A and B are placed next to each other. The two fluids are characterised by densities, that differ with a ratio of 10:1.

7.1.2a Continuity-Constraint Pressure-Correction

Again, the continuity-constraint pressure-correction algorithm predicts the wrong result (fig. 7.5). However, since fuel elements mass is predicted in a conservative way (6.6) and the density is a linear function of fuel elements mass (3.13), the predictor step is mass conserving. As a consequence, possible errors induced by the requirement of mass conservation by the corrector step (6.3), remain localised near sharp gradients, and will not interfere with results far away from them, as was the case when a single fluid ideal gas was considered.

Notice that physically impossible values are reached for the density ($\rho > \rho_{\max}$). This is a result of the fuel elements mass prediction, where in this case only the mixture fraction is upwinded. The mass flux ρu, that, as a whole, is considered to be a face value, is not

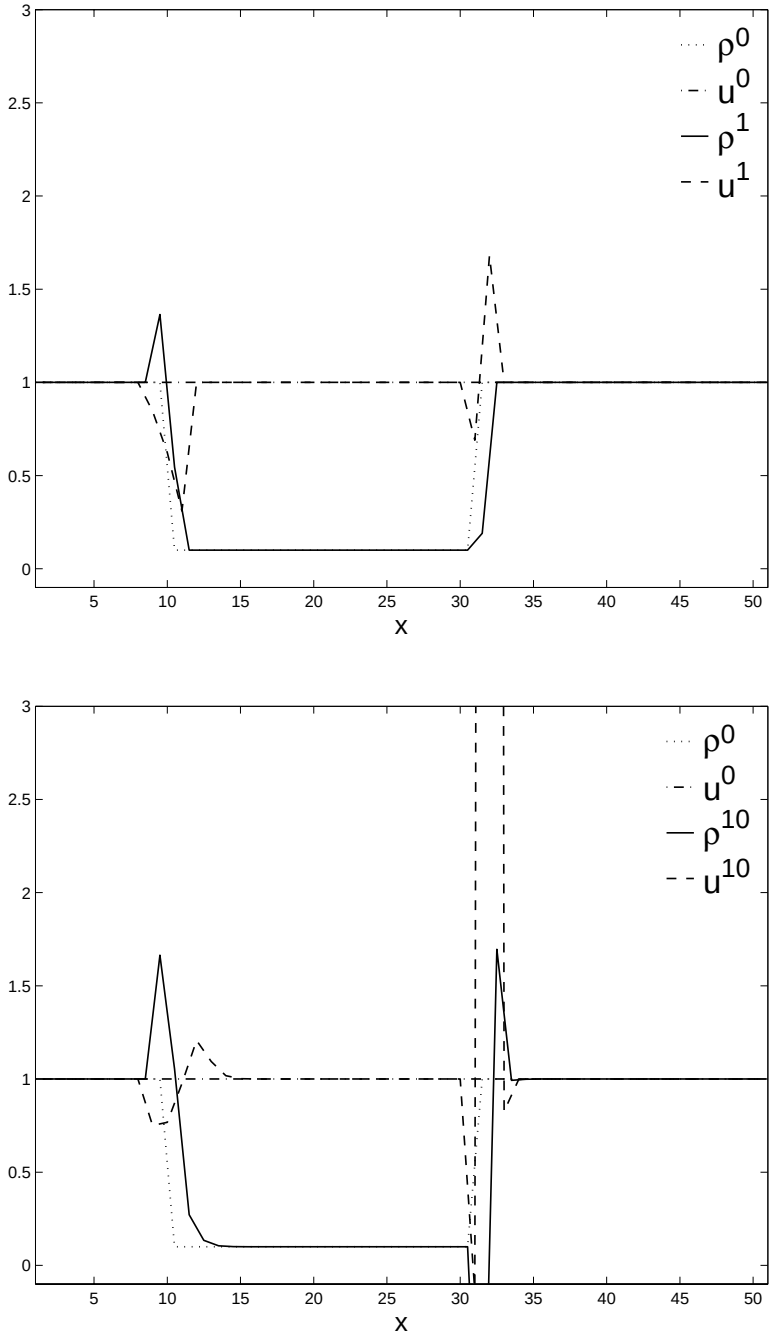

Figure 7.5: Density and velocity fields, obtained with the continuity-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure convection of inert mixing fluids. Results for the velocity field are inaccurate, close to density jumps.

constant in space, and results in local compression or expansion of the flow, from which the impossible values are obtained.

With this algorithm, the equation of state is fulfilled at every time step. This is done automatically, because the new density field is determined from the mixture fraction prediction, precisely using the equation of state.

Also in this case, a longer simulation will result in unstable solutions.

7.1.2b Analytical Compatibility-Constraint Pressure-Correction

Because of the same reason as the ideal gas case, the exact velocity field is also obtained in case of inert mixing (fig 7.6).

The fulfillment of the state equation can easily be checked by plotting density against fuel elements mass. Fig. 7.7 shows that at every time step, density and fuel elements mass are predicted according to the equation of state. This result comes as no surprise, since, also in case of inert mixing, the constraining equation is the same whether the analytical or the discrete compatibility is expressed. Results obtained with the analytical scheme will therefore not differ from the ones obtained with the discrete scheme, which is especially designed to obtain results that exactly obey the equation of state.

7.1.2c Discrete Compatibility-Constraint Pressure-Correction

As discussed above, figs. 7.6 and 7.7 also apply for this algorithm.

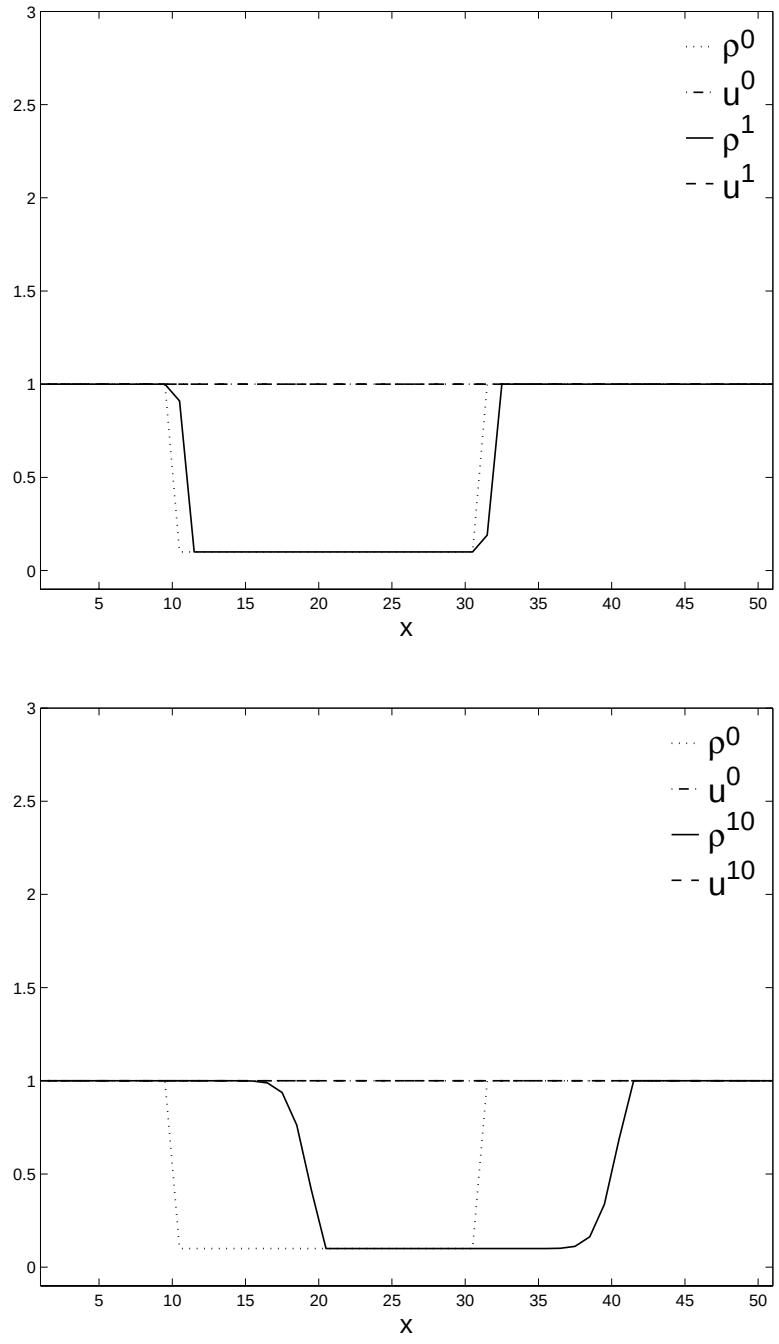

Figure 7.6: Density and velocity fields, obtained with the compatibility-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure convection of inert mixing fluids. The obtained results are equal for the analytical and the discrete version of the algorithm. A consistent prediction of the velocity is obtained.

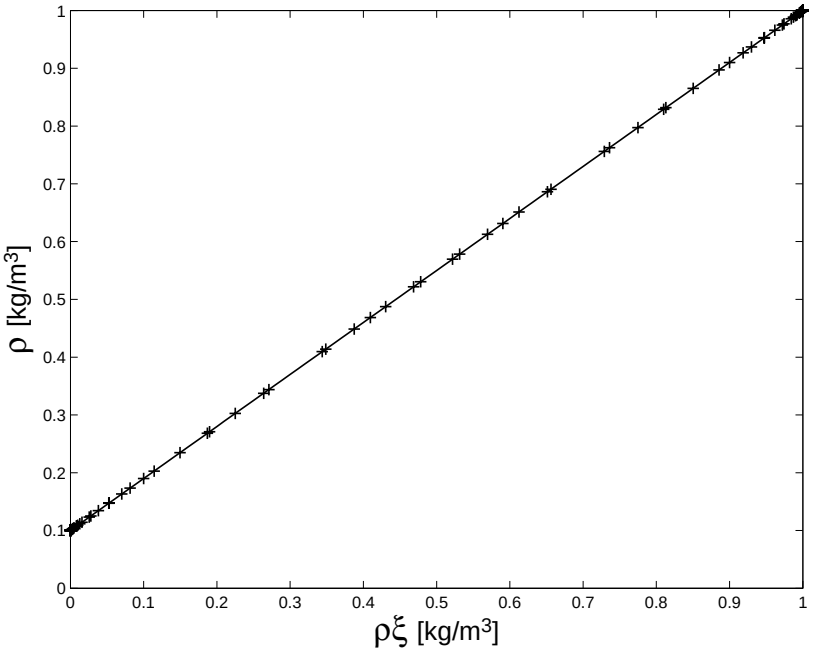

Figure 7.7: Scatter plot of the obtained density and fuel mass elements predictions during 10 time steps in the simulation of pure convection of inert mixing fluids with the compatibility-constraint pressure-correction algorithm. Density and fuel elements mass are predicted according to the equation of state (full line).

7.1.3 Two-Fluid Flow: Non-Premixed Combustion

The initial velocity, fuel elements mass and density field is depicted in fig. 7.8. Initially, fuel and oxidizer, having the same density are placed next to each other. The properties of fuel and oxidizer and its mixtures are obtained from fig. 3.3.

7.1.3a Continuity-Constraint Pressure-Correction

This time, the first time step is relatively well predicted (fig. 7.9 top), because of the constant initial density and velocity in the entire domain. At the interfaces between fuel and oxidizer, the density deviates from its constant value, due to numerical diffusion in the mixture fraction prediction. Because of that, numerical mixing occurs between fuel and oxidizer, yielding a lower density due to reaction. Because mass has to be conserved in the corrector step (6.3), a locally lower density requires the flow to accelerate towards the outlet, as can be seen in the figure (dashed line).

In the next time steps (fig. 7.9 bottom), the velocity and density fields are no longer

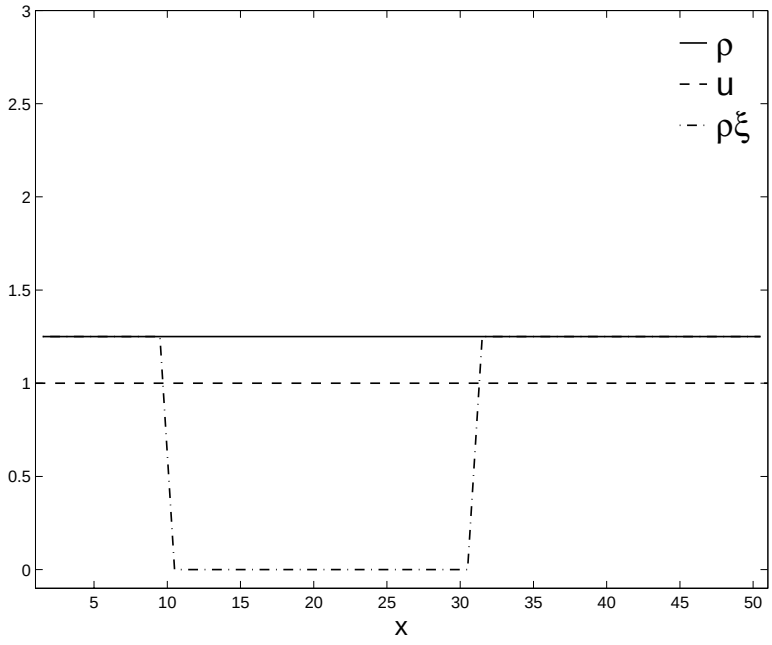

Figure 7.8: Purely convective transport of two reacting fluids in a straight channel: initial condition.

constant, and the solution gets worse. Velocity fields that differ several orders of magnitude from the exact solution can be noticed (dashed line) as well as unphysical values (higher than the initial value) of the density field (solid line). A longer simulation returns unstable solutions.

7.1.3b Analytical Compatibility-Constraint Pressure-Correction

Based on the simulation results of fig. 7.10, we would judge this scheme to be ideal, because the density and velocity fields exactly correspond to the analytical solution. However, if we consider a scatter plot of the obtained states, fig. 7.11, big discrepancies exist between the predicted density and fuel elements mass fields and the equation of state. Indeed, because of numerical diffusion, the step in mixture fraction is smoothed during convection, so that intermediate states can be obtained, however, with non-corresponding density. In this case, an analytical expression for the constraint is not sufficient to incorporate the effects, related to reacting flows, or to more general fluids.

The dicrepancies between the predicted states and the equation of state can be controlled by incorporating a defect correction in the constraining equation. Results for this test case, using a damping factor $\zeta = 0.5$, eq. (C.24), are shown in figs. 7.12 and 7.13.

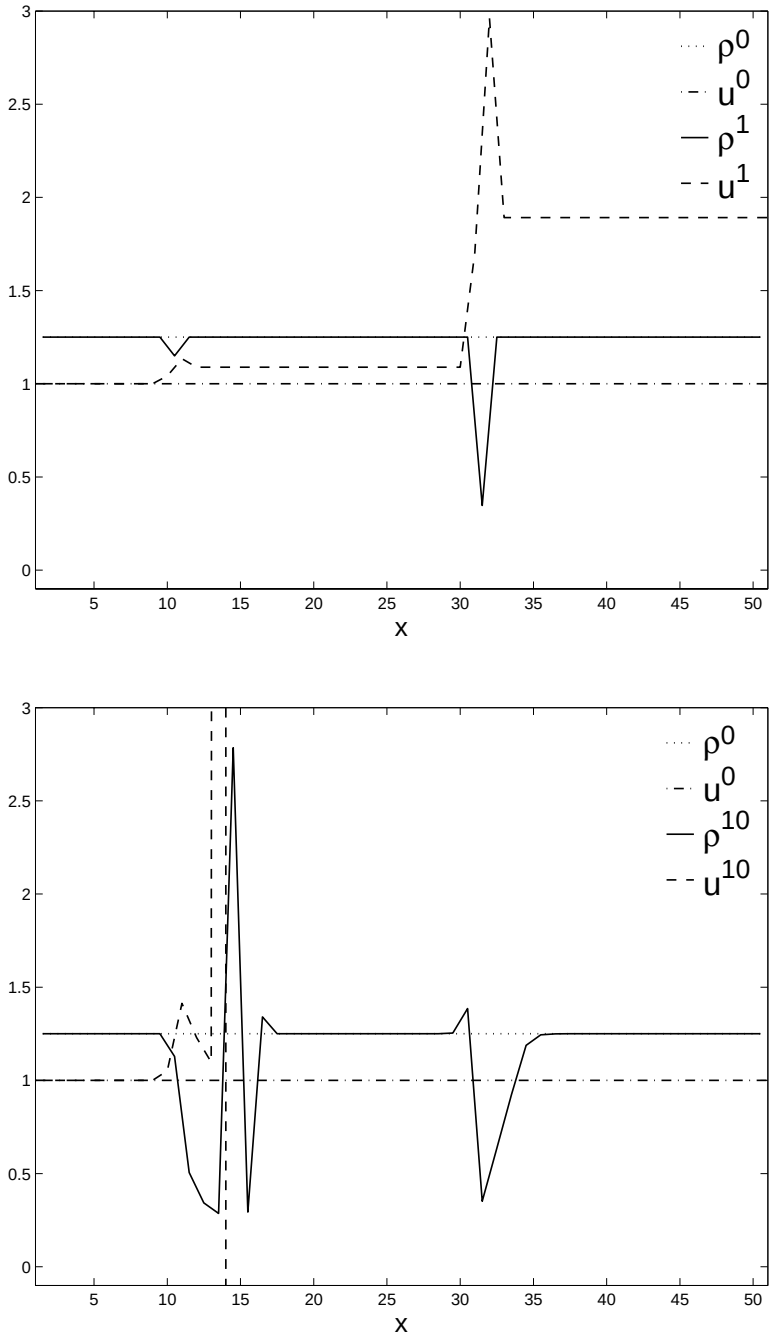

Figure 7.9: Density and velocity fields, obtained with the continuity-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure convection of reacting fluids. Results for the velocity field differ several orders of magnitude from the exact solution.

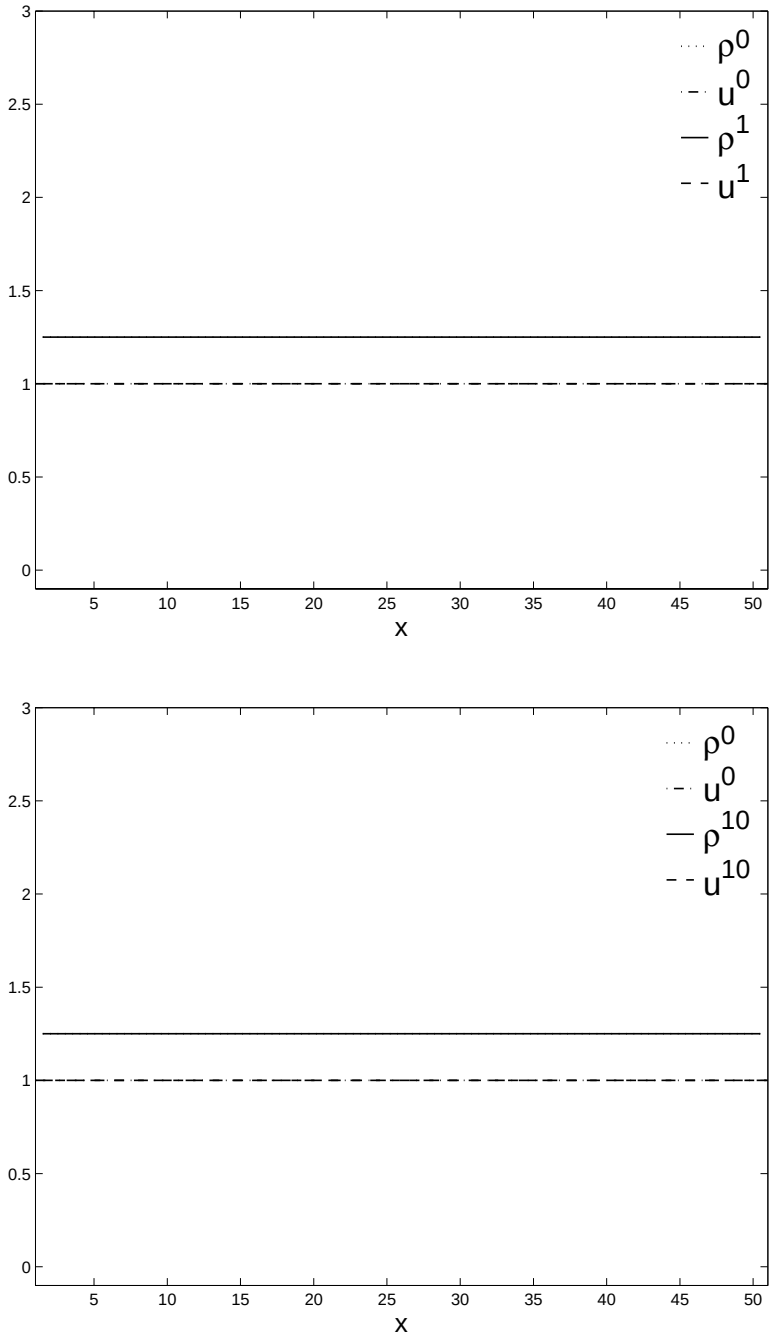

Figure 7.10: Density and velocity fields, obtained with the analytical compatibility-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure convection of reacting fluids. Solutions for the velocity field do not differ from the exact solution.

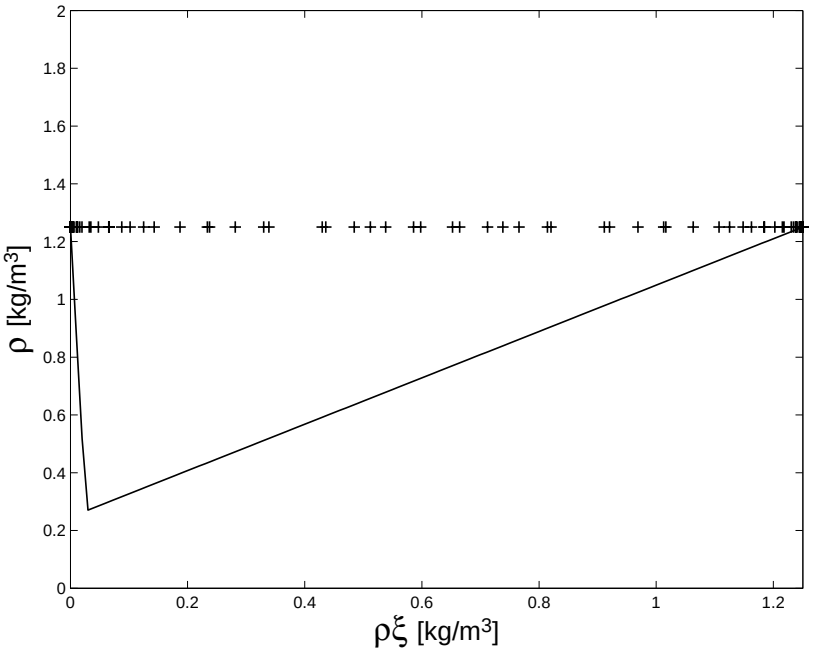

Figure 7.11: Scatter plot of the obtained density and fuel mass elements predictions during 10 time steps in the simulation of pure convection of reacting fluids with the analytical compatibility-constraint pressure-correction algorithm. There are big discrepancies between the predicted density and fuel elements mass fields and the equation of state (full line).

From the scatter plot, it can be seen that the predicted states are indeed closer to the equation of state. Density and velocity fields now are no longer exact, but incorporate the effect of numerical diffusion due to upwinding in the convective fluxes, as already explained.

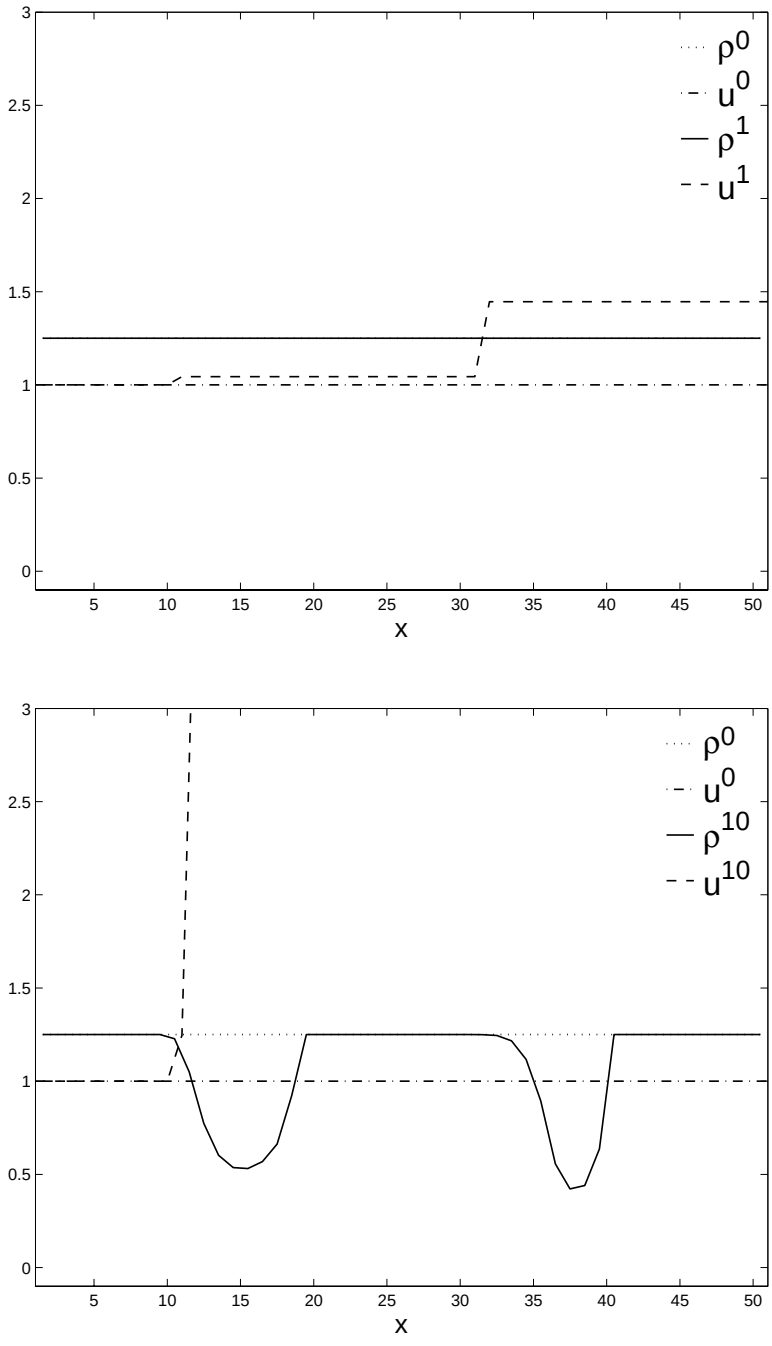

Figure 7.12: Density and velocity fields, obtained with the analytical compatibility-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure convection of reacting fluids. The drift from the equation of state is controlled using a damping factor $\zeta = 0.5$ in the defect correction term.

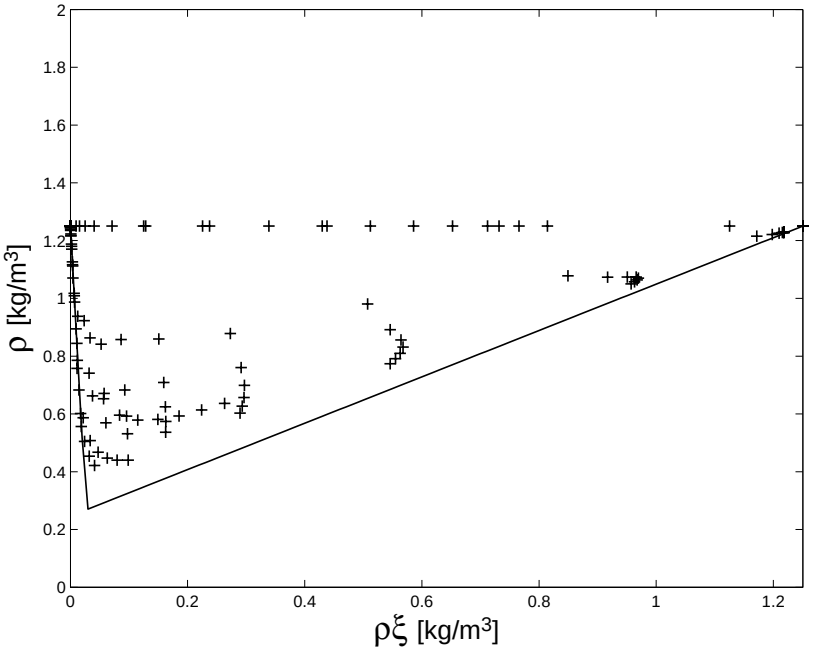

Figure 7.13: Scatter plot of the obtained density and fuel mass elements predictions during 10 time steps in the simulation of pure convection of reacting fluids with the analytical compatibility-constraint pressure-correction algorithm. The drift from the equation of state is controlled using a damping factor $\zeta = 0.5$ in the defect correction term, but still big discrepancies exist between the predicted density and fuel elements mass fields and the equation of state (full line).

7.1.3c Discrete Compatibility-Constraint Pressure-Correction

In case of non-premixed combustion, the discrete formulation of the constraint differs from the analytical one, resulting in different results (fig. 7.14). Again, a reaction zone, characterised by a lower density, can be noticed at the interfaces between fuel and oxidizer. As a result, the flow accelerates towards the outlet. The reaction is caused by numerical diffusion: due to the first order upwinding of the convective terms, the initial step in the mixture fraction field is smoothed, resulting in intermediate mixture fraction values, whose corresponding density is lower, according to the equation of state.

In contrast to the continuity-constraint pressure-correction scheme, the compatibility-constraint pressure correction yields stable results, that are physically possible. In the discrete version, the predicted states correspond exactly to the equation of state (fig. 7.15).

To obtain this exact correspondence, a price is to be paid. The constraining equation

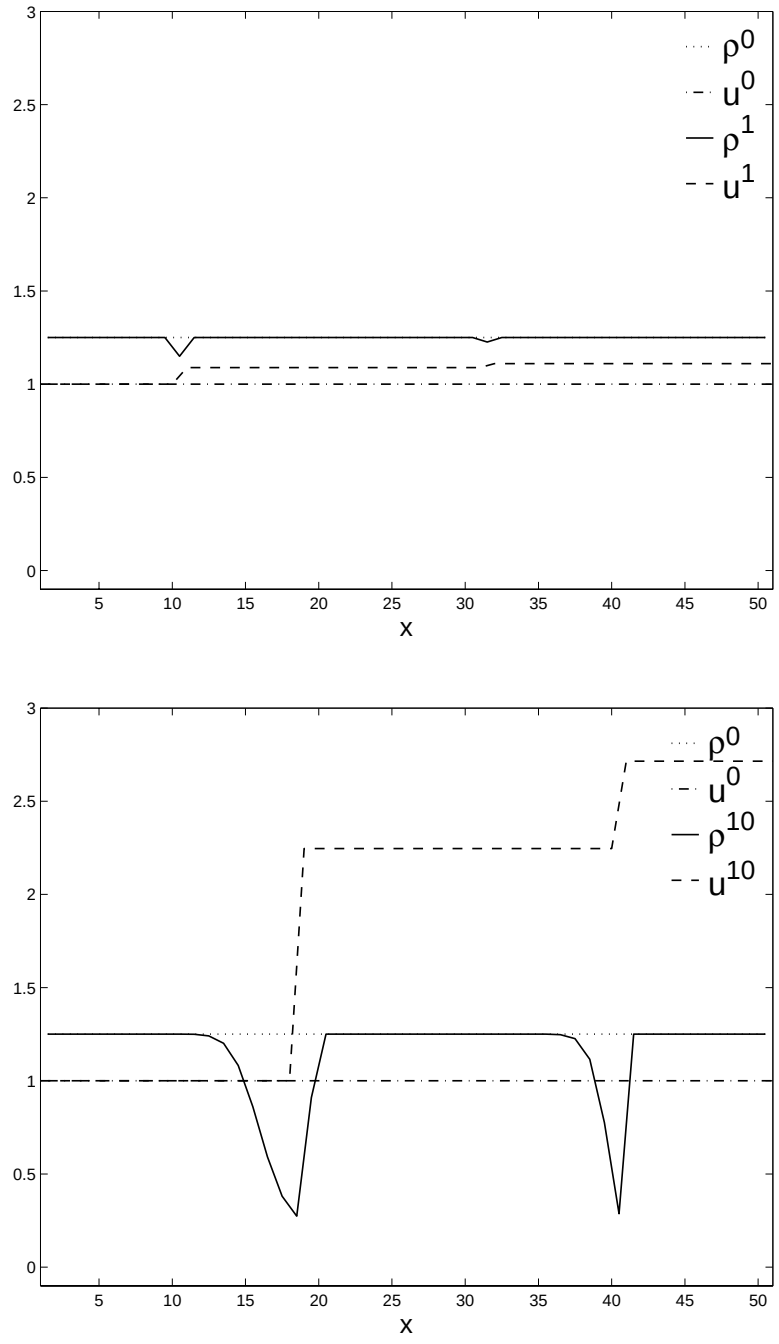

Figure 7.14: Density and velocity fields, obtained with the discrete compatibility-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure convection of reacting fluids. The difference between the results and the exact solution is due to upwinding.

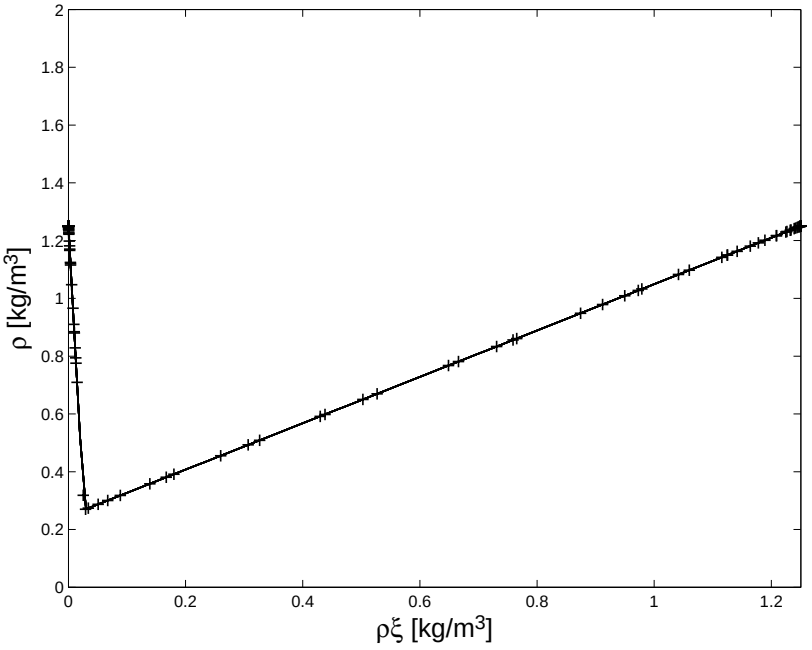

Figure 7.15: Scatter plot of the obtained density and fuel mass elements predictions during 10 time steps in the simulation of pure convection of reacting fluids with the discrete compatibility-constraint pressure-correction algorithm. The predicted density and fuel elements mass behave exactly according to the equation of state (full line).

(5.25) is now a non-linear equation in u', so that several iterations are needed to obtain the solution, performing linearisation (5.30). For this problem only one re-linearisation was needed to solve the non-linear system. In more general flows, more iterations will be needed. Still, if the re-linearisation is done in a smart way, as discussed in section 5.4.3, only a minimal additional effort is spent, since the elliptic pressure equation already requires an iterative solution procedure for realistic problems.

7.2 One-Dimensional Pure Diffusion/Conduction

As a second test, pure diffusion is considered, meaning that initially the velocity is zero everywhere in the domain but diffusive transport is admitted through conduction or species diffusion, using values for the diffusive constants $\lambda = 1$ in case of a single fluid ideal gas and $\rho D = 1$ in case of two fluid flow. It should be noted that non-zero velocities will arise during the simulation, giving rise to a convective part. Since this is only a secondary effect, the term 'pure diffusion' is used to capture this series of test cases.

In the resulting test case, diffusion takes place at a step in the scalar variable ϕ (temperature or mixture fraction) in a straight channel. The initial step is defined in space as the piecewise constant function

$$\phi_i = \begin{cases} \phi_1 & \text{for} \quad i \in [1, i_1[\\ \phi_2 & \text{for} \quad i \in]i_1, N_x] \end{cases} \tag{7.3}$$

In this case, no analytical solution exists. In the problem considered, $N_x = 50$ grid points were used and the step in the scalar field is situated at grid node 20. The grid spacing is set to 1. At the left boundary a velocity of 0 is imposed, whereas the velocity is let free at the right boundary.

It will become clear in this section that results for pure diffusion show the same tendencies as in the pure convection case. In general, it is seen that a diffusive flow behaves more robust than a purely convective flow.

7.2.1 Single Fluid Flow: Ideal Gas

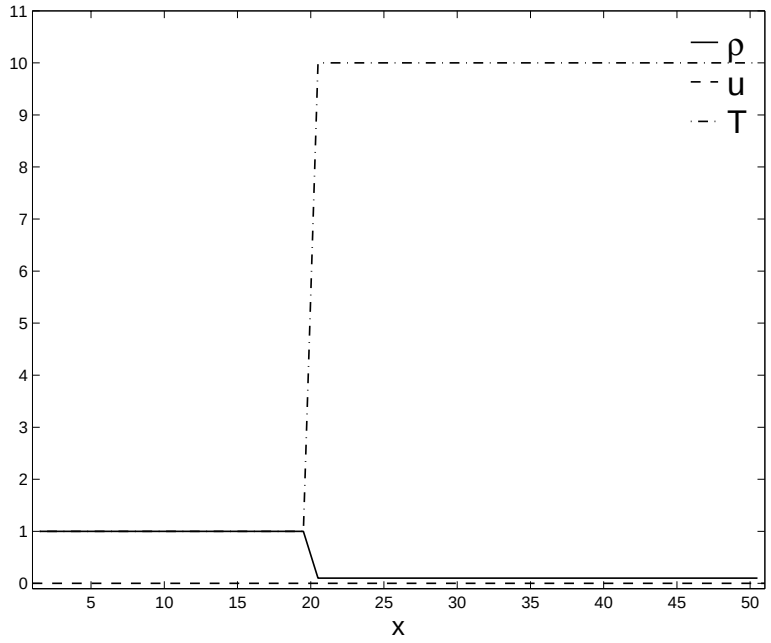

Figure 7.16: Purely conductive transport of a density jump in a straight channel, filled with an ideal gas at different temperatures: initial condition.

The initial velocity, temperature and density field is depicted in fig. 7.16. A temperature step with a step factor of 10 is used, resulting in a density field, having the same initial ratio.

7.2.1a Continuity-Constraint Pressure-Correction

For this test case, the continuity-constraint pressure-correction algorithm remains stable (fig. 7.17). However, because of the non-mass conserving prediction of temperature (and density), inaccurate velocity fields are predicted, even in regions far from the diffusion layer.

7.2.1b Analytical Compatibility-Constraint Pressure-Correction

This algorithm shows consistent predictions for conductive flows (fig. 7.18). A positive velocity field near the diffusion zone ensures mass transport from the high-density region ($x < 20$) to the low-density region ($x > 20$). At the outlet, the velocity remains zero, as it should: since there is no reaction, the flow should not display overall acceleration.

7.2.1c Discrete Compatibility-Constraint Pressure-Correction

Since this algorithm does not differ from the analytical one, fig. 7.18 also applies.

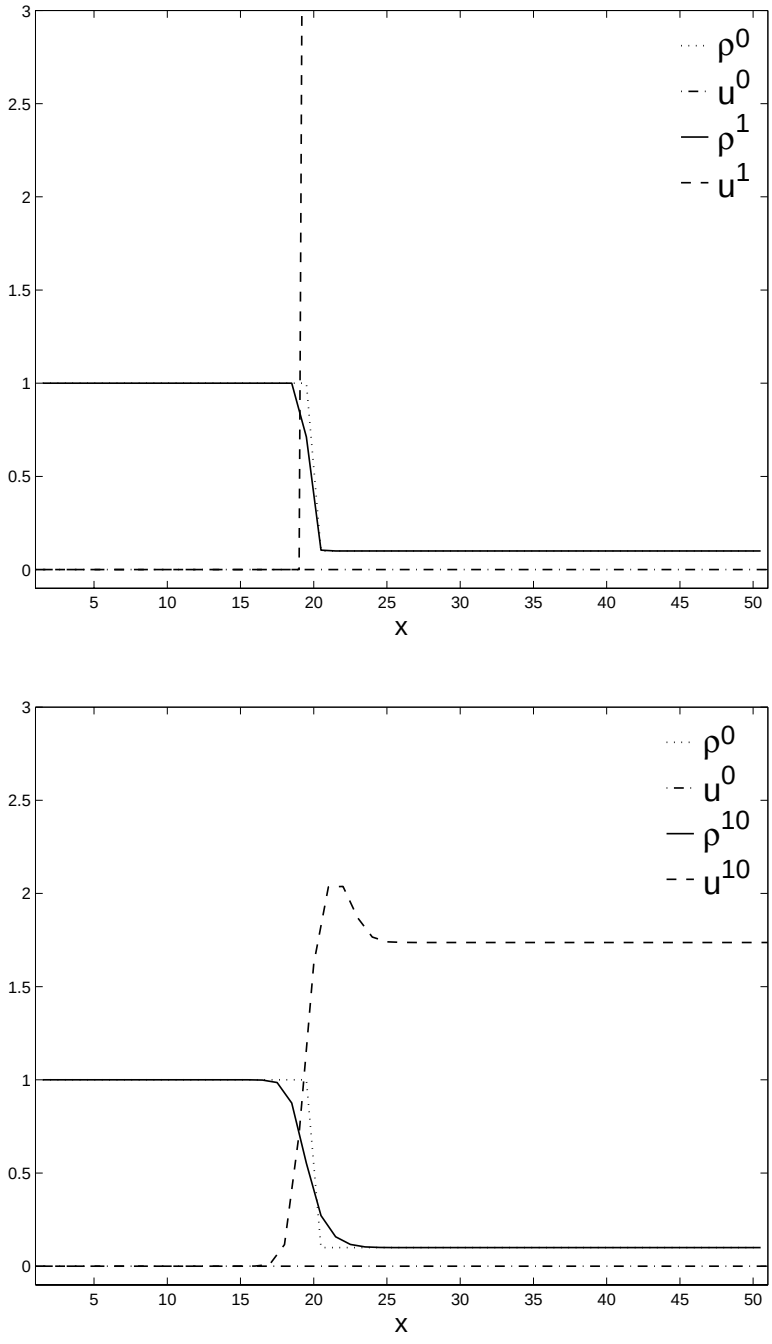

Figure 7.17: Density and velocity fields, obtained with the continuity-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure conduction of an ideal gas at different temperatures. Results for the velocity field are inaccurate, even in the region far from the density jump ($x >> 20$).

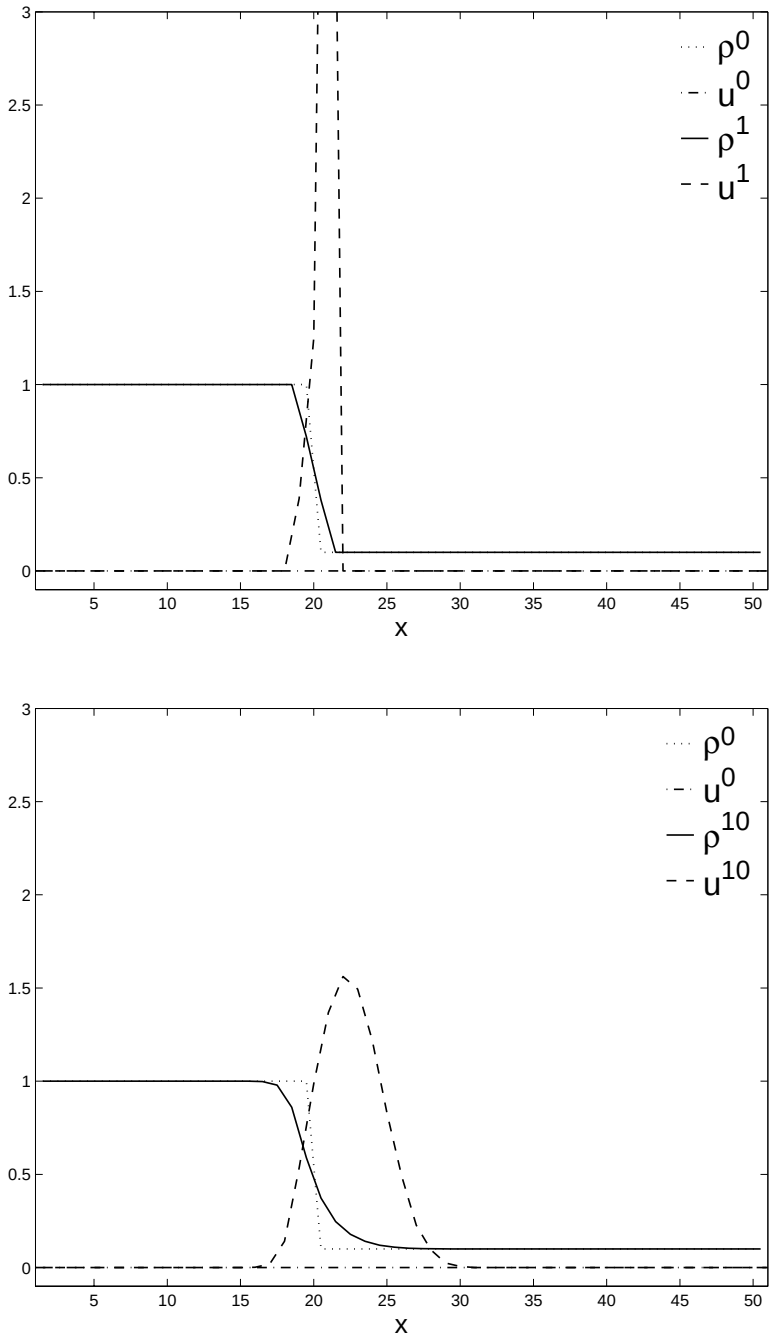

Figure 7.18: Density and velocity fields, obtained with the compatibility-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure conduction of an ideal gas at different temperatures. The obtained results are equal for the analytical and the discrete version of the algorithm. A consistent prediction of the velocity is obtained.

7.2.2 Two Fluid Flow: Inert Mixing

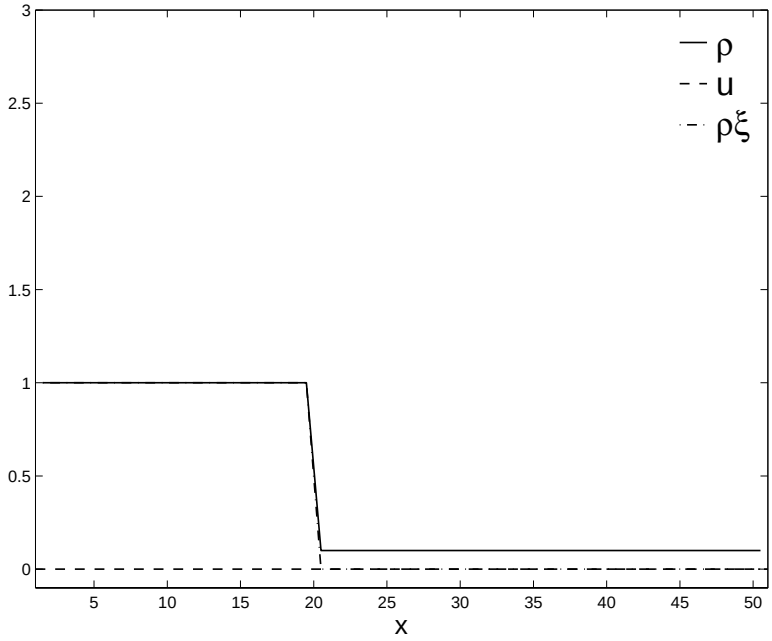

Figure 7.19: Purely diffusive transport of two inert mixing fluids with different density in a straight channel: initial condition.

The initial velocity, fuel elements mass and density field is depicted in fig. 7.19. Initially, two fluids A and B are placed next to each other. The two fluids are characterised by densities, that differ with a ratio of 10:1.

7.2.2a Continuity-Constraint Pressure-Correction

The flow shows a good behaviour (fig. 7.20). Especially the zero velocity at the right boundary is noticable, originating from a mass conserving predictor step.

7.2.2b Analytical Compatibility-Constraint Pressure-Correction

A good prediction of density and velocity fields is obtained (fig. 7.21), resulting in predicted states that behave exactly according to the equation of state (fig. 7.22).

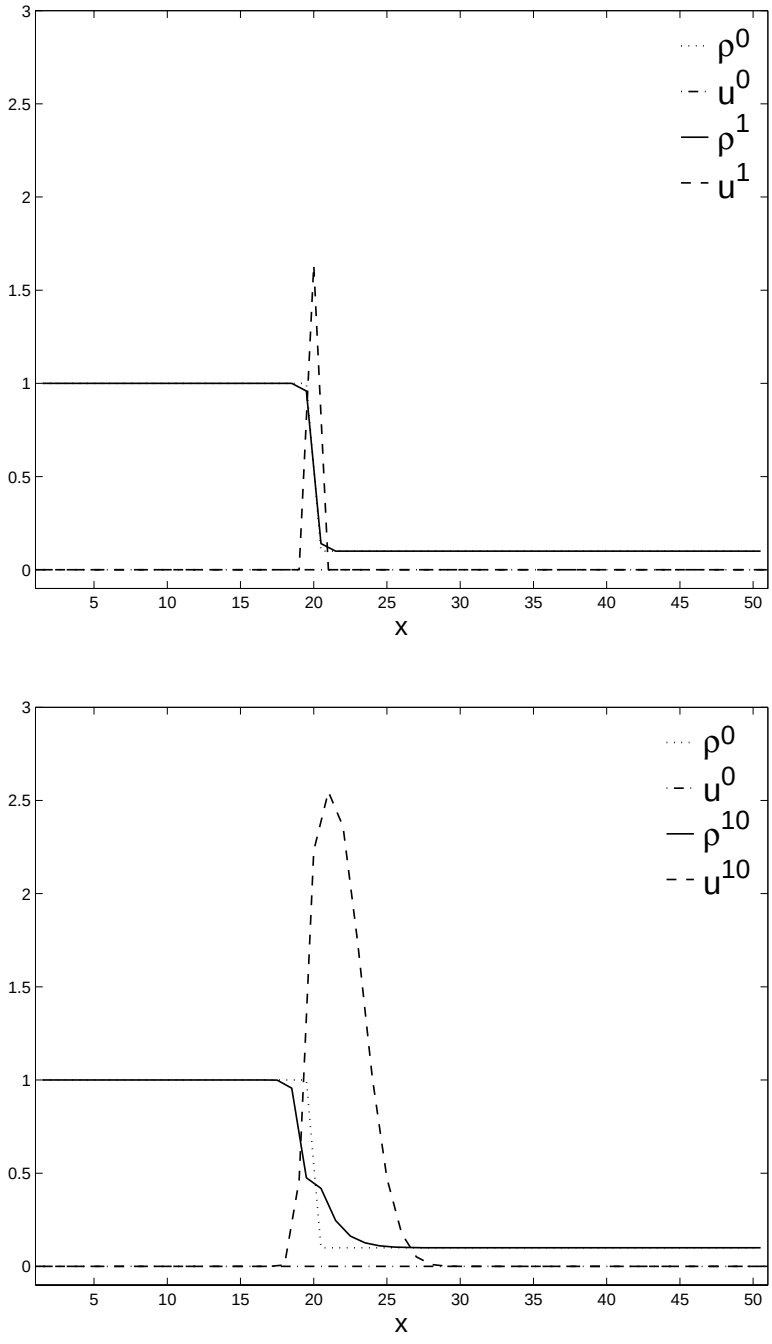

Figure 7.20: Density and velocity fields, obtained with the continuity-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure diffusion of inert mixing fluids. Results for the velocity field are inaccurate, close to density jumps.

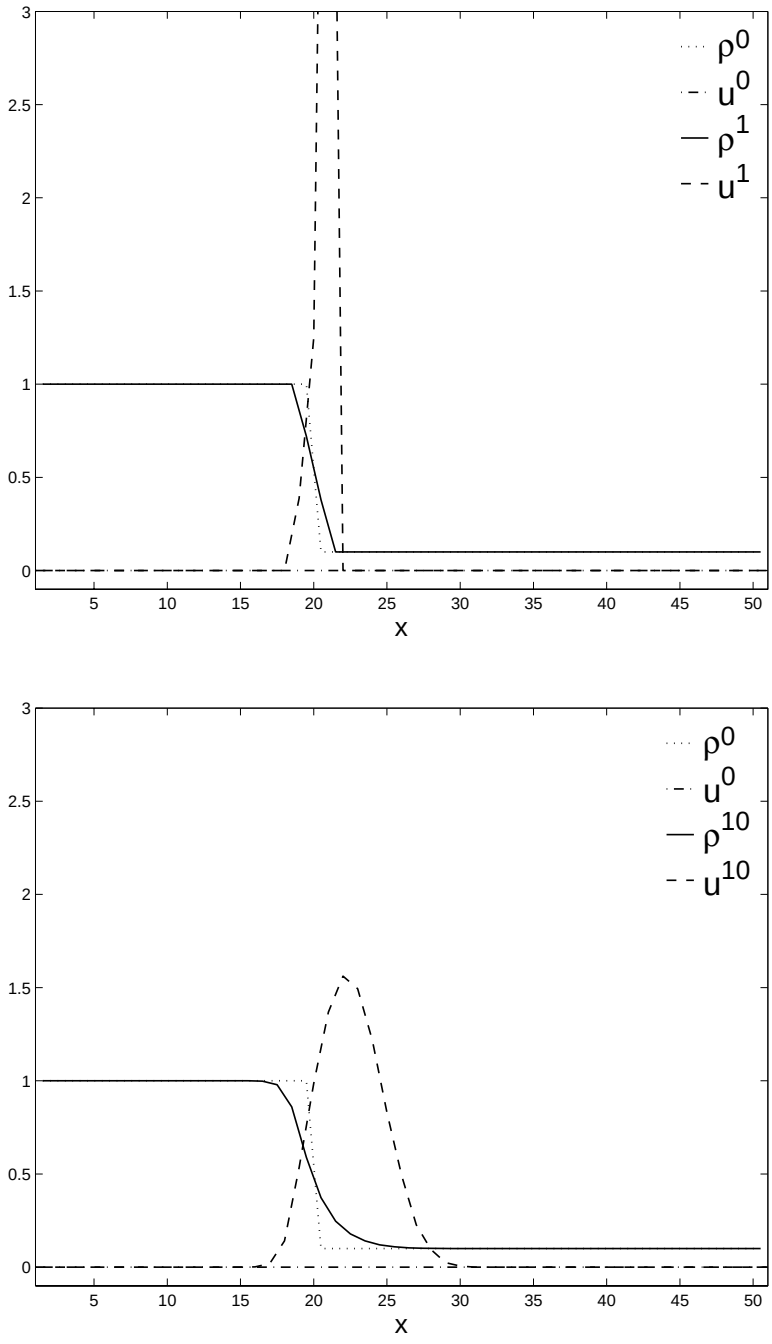

Figure 7.21: Density and velocity fields, obtained with the compatibility-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure diffusion of inert mixing fluids. The obtained results are equal for the analytical and the discrete version of the algorithm. A consistent prediction of the velocity is obtained.

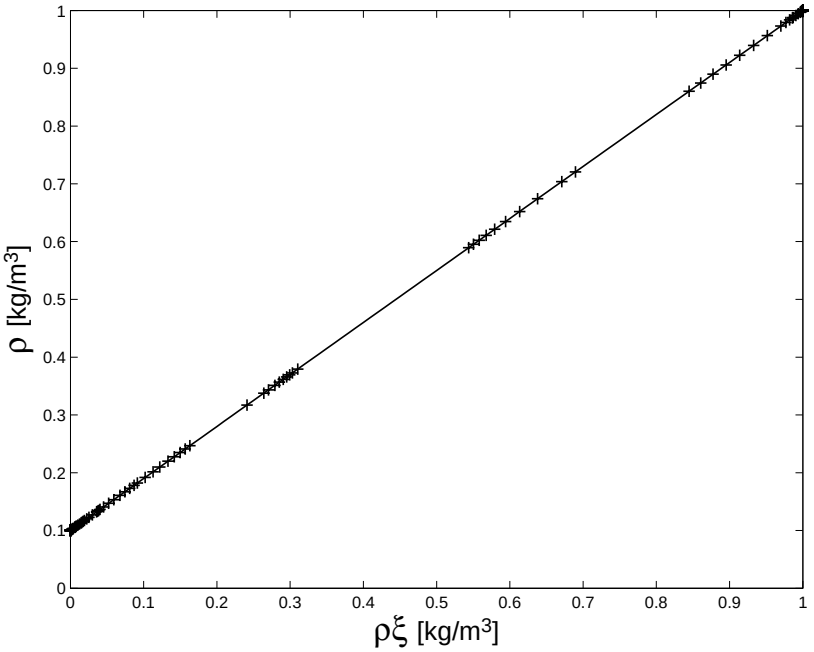

Figure 7.22: Scatter plot of the obtained density and fuel mass elements predictions during 10 time steps in the simulation of pure diffusion of inert mixing fluids with the compatibility-constraint pressure-correction algorithm. Density and fuel elements mass are predicted according to the equation of state (full line).

7.2.2c Discrete Compatibility-Constraint Pressure-Correction

We refer to figs. 7.21 and 7.22 for the simulation results.

7.2.3 Two-Fluid Flow: Non-Premixed Combustion

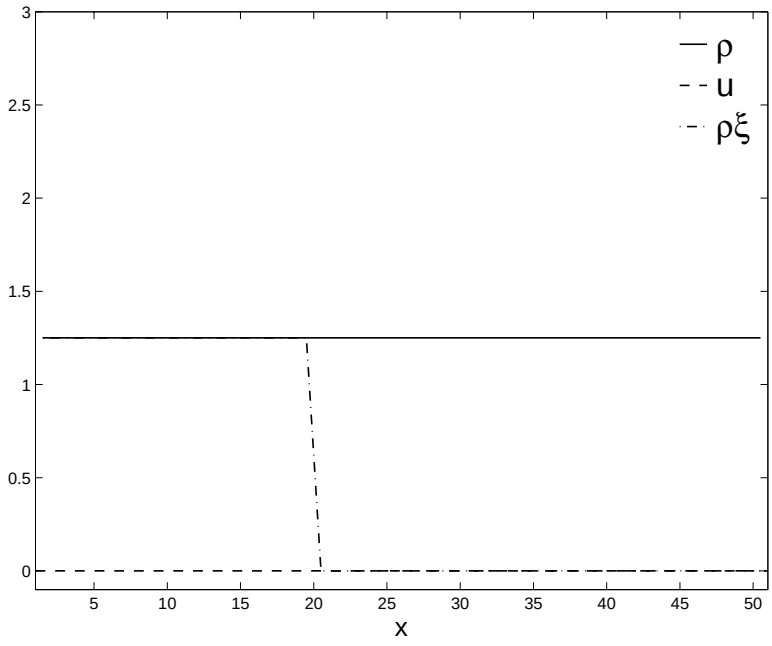

Figure 7.23: Purely diffusive transport of two reacting fluids in a straight channel: initial condition.

The initial velocity, fuel elements mass and density field is depicted in fig. 7.23. Initially, fuel and oxidizer, having the same density are placed next to each other. The properties of fuel and oxidizer and its mixtures are obtained from fig. 3.3.

7.2.3a Continuity-Constraint Pressure-Correction

Fig. 7.24 shows the results for the continuity-constraint pressure-correction. Wiggles appear in the solution for the velocity. Already after 10 time steps, the velocity field deviates several orders of magnitude from the exact solution, yielding unstable solutions.

7.2.3b Analytical Compatibility-Constraint Pressure-Correction

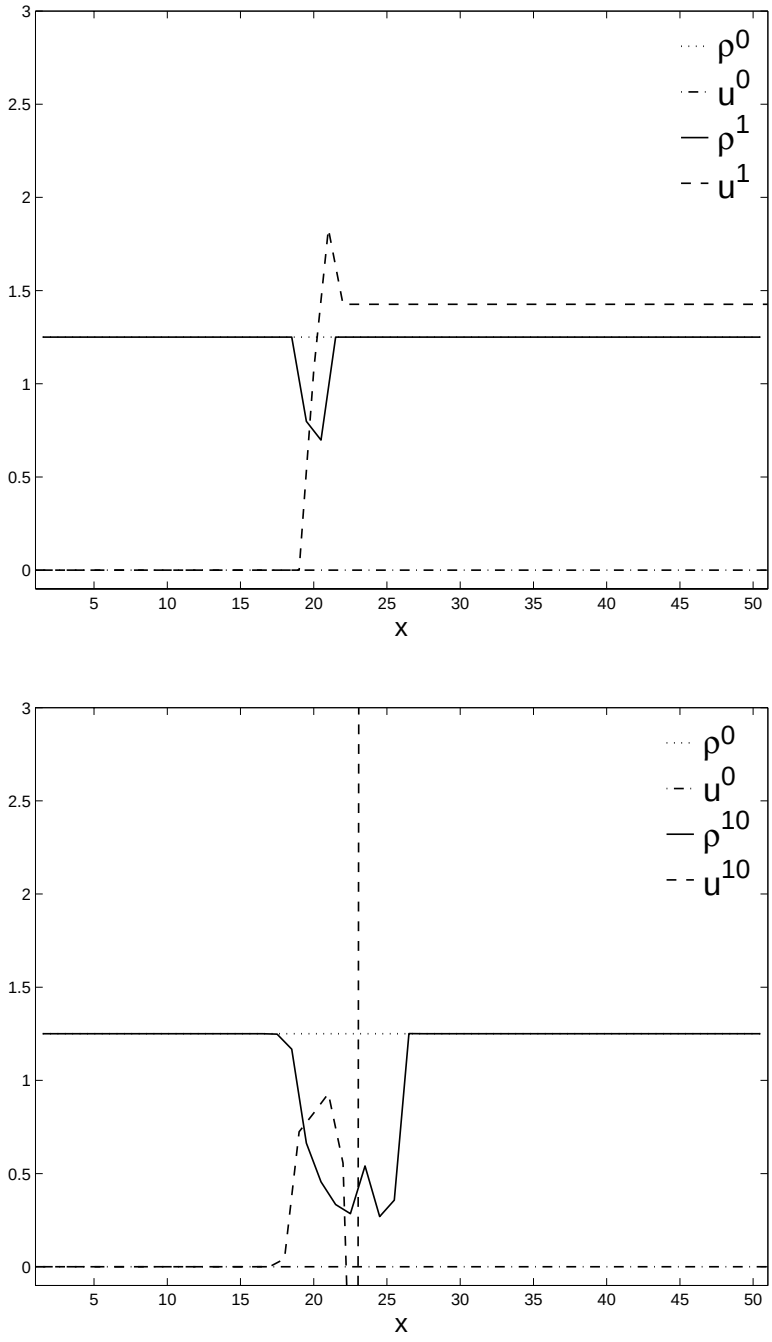

Figure 7.24: Density and velocity fields, obtained with the continuity-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure diffusion of reacting fluids. Results for the velocity field differ several orders of magnitude from the exact solution.

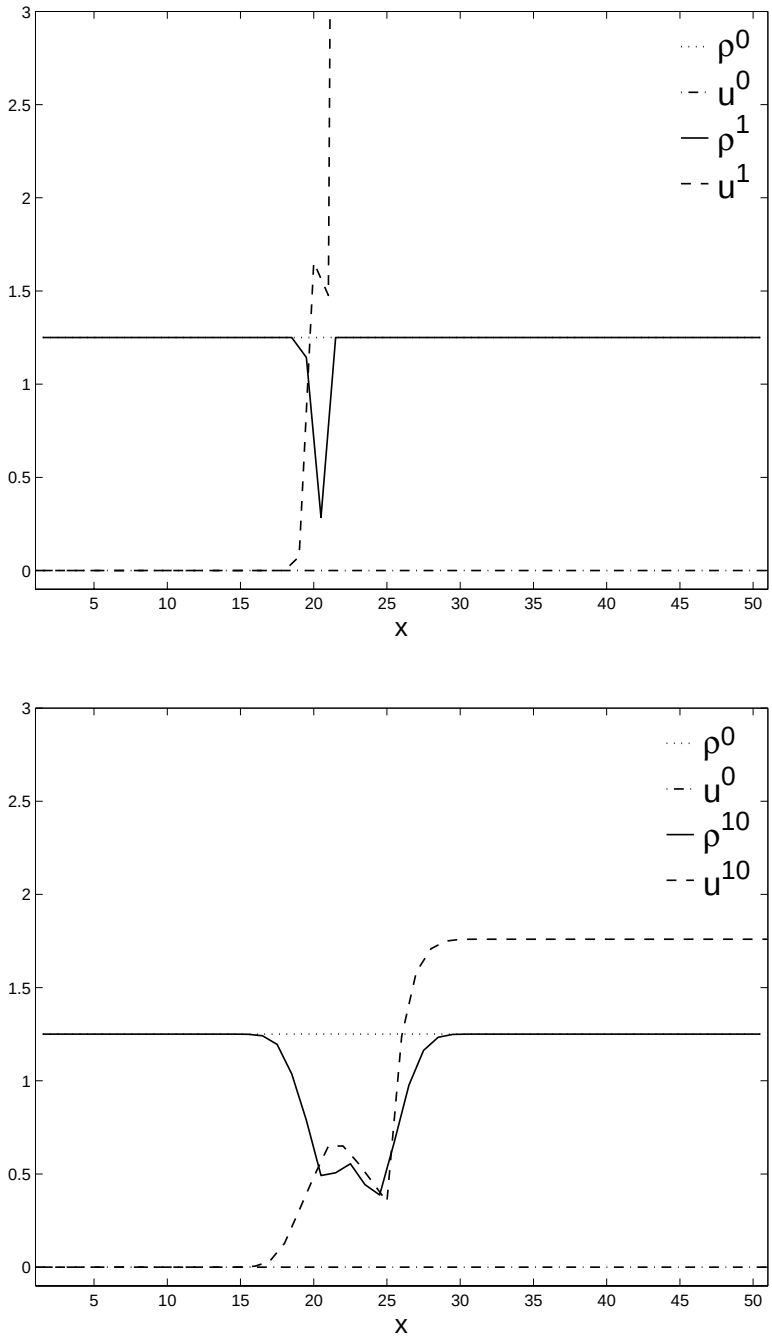

Figure 7.25: Density and velocity fields, obtained with the analytical compatibility-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure diffusion of reacting fluids. Inaccurate results are obtained due to the drift from the equation of state.

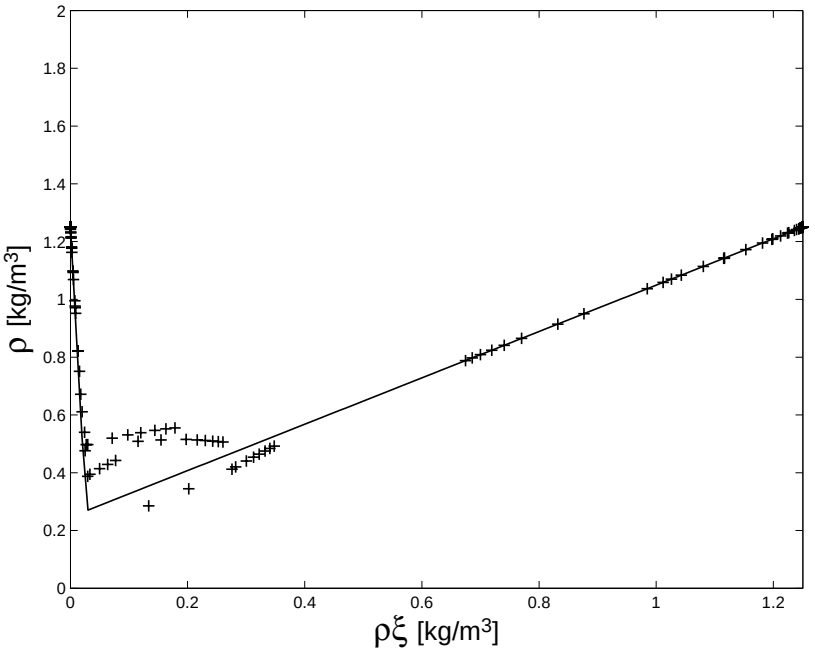

Figure 7.26: Scatter plot of the obtained density and fuel mass elements predictions during 10 time steps in the simulation of pure diffusion of reacting fluids with the analytical compatibility-constraint pressure-correction algorithm. There are big discrepancies between the predicted density and fuel elements mass fields and the equation of state (full line).

Using the analytical compatibility-constraint pressure-correction method, a stable simulation can be performed, yielding at first sight (fig. 7.25) acceptably accurate results. Remarkable is the bump in the density field after 10 time steps. A look at the scatter plot (fig. 7.26) shows that this is due to discrepancies between the predicted density and fuel elements mass fields and the equation of state.

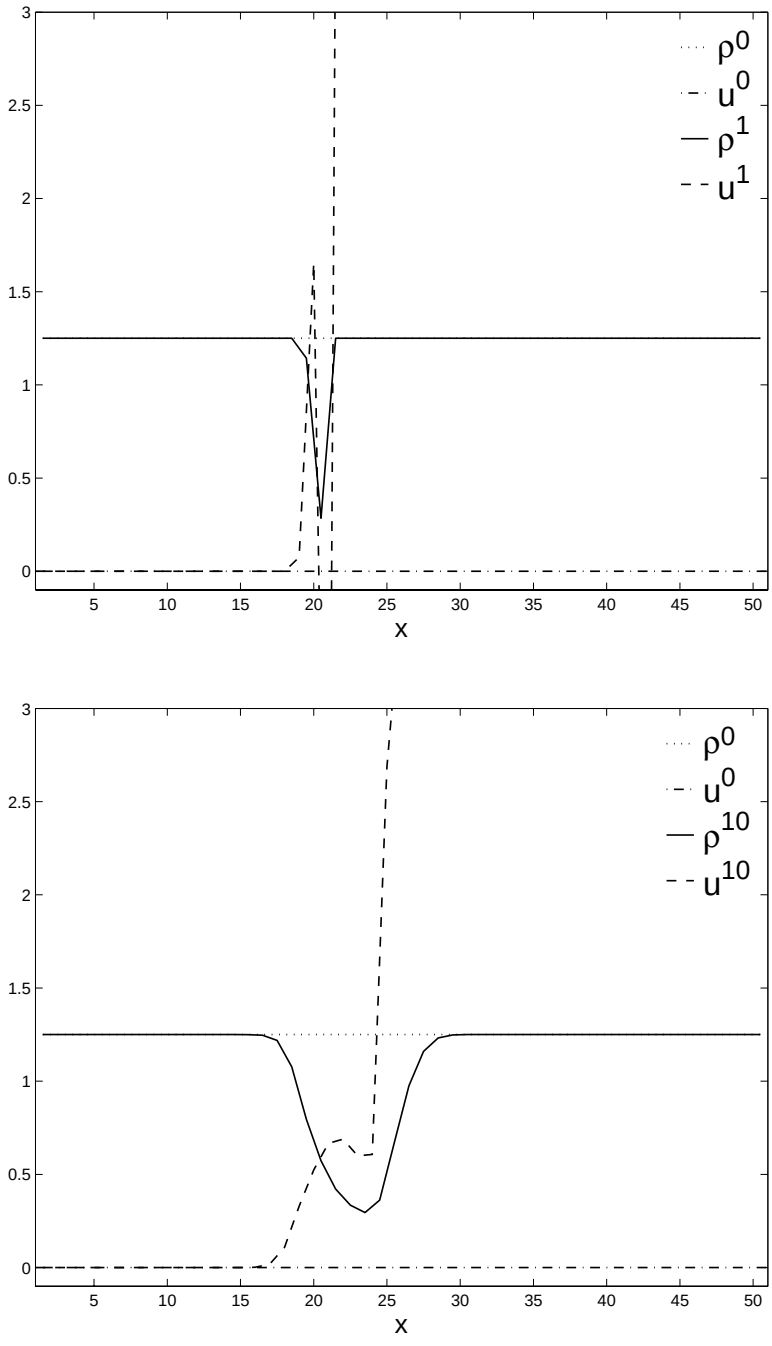

Figure 7.27: Density and velocity fields, obtained with the analytical compatibility-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure diffusion of reacting fluids. The drift from the equation of state is controlled using a damping factor $\zeta = 0.5$ in the defect correction term.

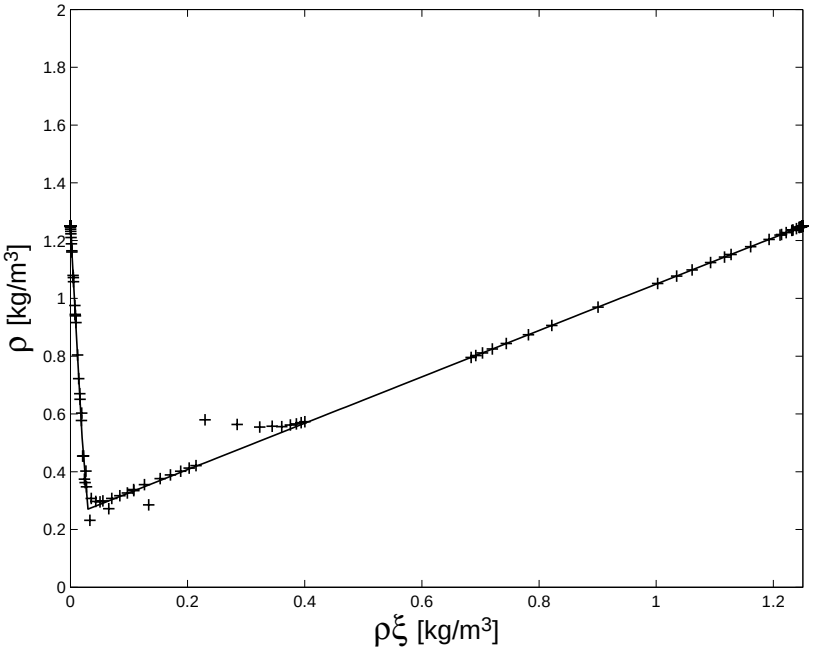

Figure 7.28: Scatter plot of the obtained density and fuel mass elements predictions during 10 time steps in the simulation of pure diffusion of reacting fluids with the analytical compatibility-constraint pressure-correction algorithm. The drift from the equation of state is controlled using a damping factor $\zeta = 0.5$ in the defect correction term, but still big discrepancies exist between the predicted density and fuel elements mass fields and the equation of state (full line).

Inclusion of a defect correction term in the corrector step, with $\zeta = 0.5$, eq. (C.24), alleviates the difference between the predicted state and the state equation (fig. 7.28) and is able to remove the bump in the density field (fig. 7.27 bottom), yielding a more accurate prediction.

7.2.3c Discrete Compatibility-Constraint Pressure-Correction

The use of the discrete compatibility-constraint pressure-correction algoritm is able to predict good results for all state variables (fig. 7.29), that behave exactly according to the equation of state (fig. 7.30). To achieve this exact correspondence, again mostly one, at maximum two re-linearisations were needed during the inversion of the pressure Poisson-like equation.

The deceleration in the velocity field after 10 time steps (fig. 7.29 bottom, around $x = 24$) can be explained at this stage. The velocity field shows a combination of two physical phenomena, being reaction, imposing a flow acceleration towards the outlet, and

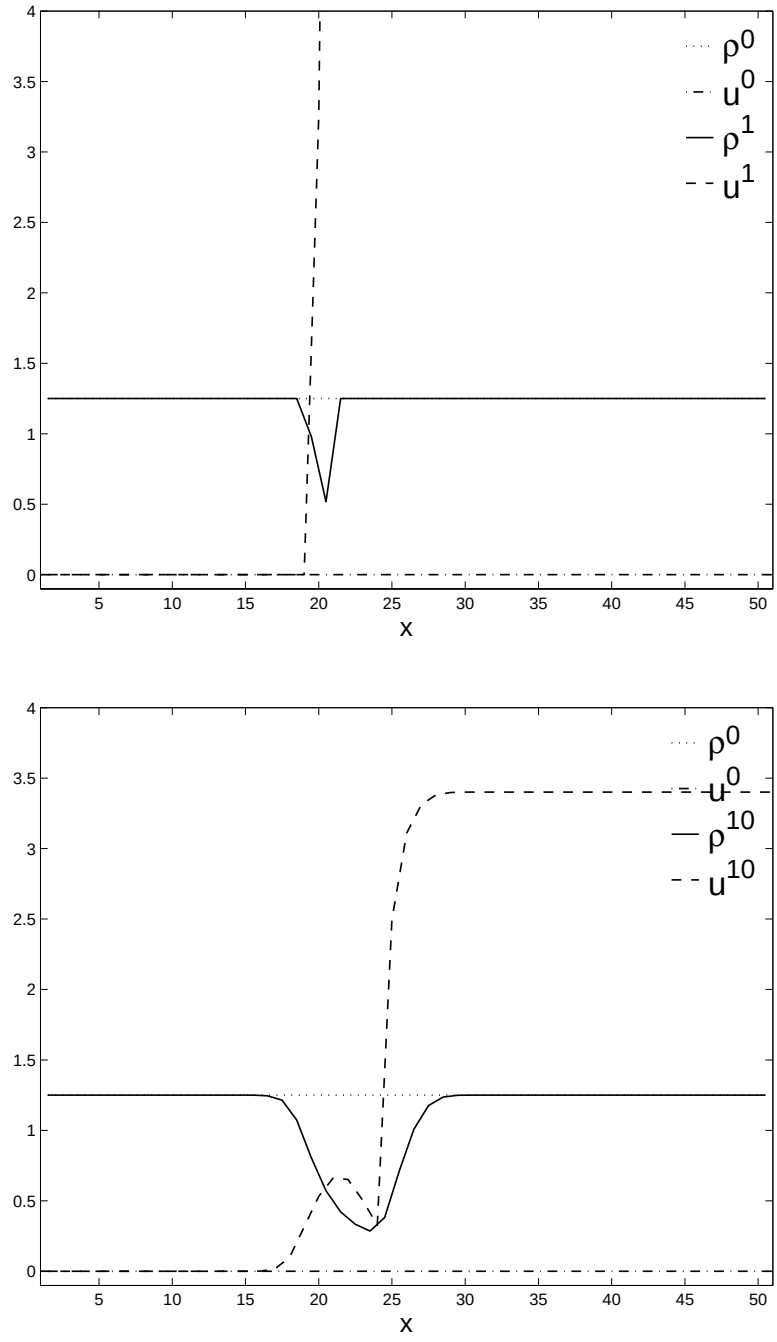

Figure 7.29: Density and velocity fields, obtained with the discrete compatibility-constraint pressure-correction algorithm after 1 (top) and 10 (bottom) time steps for pure diffusion of reacting fluids. Apart from discretisation errors, a satisfactory solution is obtained.

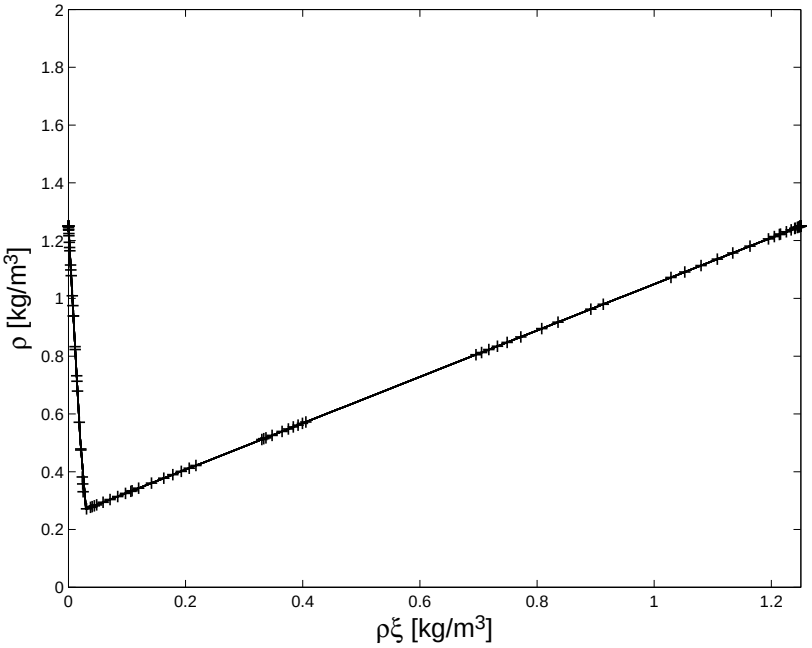

Figure 7.30: Scatter plot of the obtained density and fuel mass elements predictions during 10 time steps in the simulation of pure diffusion of reacting fluids with the discrete compatibility-constraint pressure-correction algorithm. The predicted density and fuel elements mass behave exactly according to the equation of state (full line).

mixing between reaction products and fuel, imposing mass transfer due to the difference in density, similar to the inert mixing case (see e.g. fig. 7.21).

7.3 Conclusion

In this chapter, three pressure-correction algorithms were used to simulate one-dimensional test cases, involving convection and diffusion of sharp initial scalar gradients. Three fluid types were investigated: a single-fluid ideal gas at different temperatures, a two-fluid non-reacting flow and a two-fluid combusting flow. The continuity-constraint pressure-correction scheme revealed instabilities in nearly all cases and is therefore not suited to simulate variable density flows with sharp density gradients. The analytical compatibility-constraint pressure-correction scheme yielded stable results, but predicted states that strongly deviate from the equation of state in case of a non-linear equation of state, as demonstrated in the combusting flow case. With the incorporation of a defect correction term, better results could be obtained, but still, the predict states did not exactly match the equation of state. Finally, the discrete compatibility-constraint pressure-correction

algorithm was able to yield stable and accurate results in all cases.

Chapter 8

Odd-Even Decoupling

It is well-known [52] that discretisation of the partial differential flow equations on a mesh with collocated variables, which means that all local state and flow variables are stored at the same position, can give rise to a *spurious mode* (a π-wave) for the pressure, when the cell-face velocities are linearly interpolated between the neighbouring nodes without pressure stabilisation and when the pressure term in the momentum equations is approximated by central differencing. This mode is not seen by the discretised equations and leads to a solution without physical meaning.

A solution for this problem is a *staggered* treatment of the variables [52]. This solution was adopted in the 1D test cases of chapter 7. Variants of the pressure-correction scheme of this type are the MAC (Marker-And-Cell) [26] and the SIMPLE (Semi-Implicit Method for Pressure-Linked Equations) [51] methods. If rectangular grids are used, the choice of a staggered grid arrangement is the most natural choice for a straightforward discretisation of the governing equations. However, this approach is not comfortable, especially in a three-dimensional environment making use of body-fitted grids [49]. In these more general cases there are practical advantages to use grids with collocated arrangements. Although convenient, a major drawback is found in the complication of the algorithms to filter out spurious modes. This problem can then be resolved by using special flux-splitting schemes [22] or pressure weighted interpolation (PWI) methods, as first introduced in [62]. Other propositions, concerning pressure weighted velocity interpolation or pressure gradient interpolation were made in [18, 19, 79] for low Mach number flows or in [23, 47] for flows at all speeds. Unfortunately, in time-accurate solutions of variable density flows, special requirements of system solvability are not unconditionally fulfilled by these propositions.

In this chapter we rigorously derive a solution formalism for solving the odd-even decoupling problem in the framework of pressure-correction algorithms for variable density flows[1]. The strategy is described for a single fluid ideal gas flow in the presence of

[1]The contents of this chapter has been published in [58].

conduction[2], but is immediately applicable for general fluids, using the algorithm of chapter 5. The solution formalism will be applied in the simulations of the 2D test-cases on collocated grids in chapter 9.

In the next section the pressure correction scheme applied to the low Mach continuity, momentum and energy equations, is given without a remedy for odd-even decoupling. Emphasis is put on the solvability condition of the resulting Poisson equation for the pressure and the problem of the spurious mode is further elaborated. Section 8.2 describes the cure for the odd-even decoupling including variable density. It consists of two remedies: a correction term for the cell-face velocity, similar to [62], is introduced and the stencil for the discrete Laplacian in the equation for pressure is compacted. Section 8.3 shows the applicability of the method on general curvilinear coordinate systems in three dimensions. Section 8.4 compares the present cure with other remedies for odd-even decoupling in the literature. In section 8.5, finally, the most important findings are summarised.

8.1 Problem Setting

8.1.1 Pressure-Correction Algorithm

The pressure-correction algorithm is almost identical to the described one of section 5.4.1. We briefly repeat the most important features for a 1D case in order to reveal the odd-even decoupling and its solution. Here, a modification is made on the temperature equation: equation (2.42) is rewritten in terms of density, using equation of state (2.55) and continuity equation (2.39):

$$\frac{\partial \rho}{\partial t} + u_i \frac{\partial \rho}{\partial x_i} = \frac{\rho}{\gamma p_0} \frac{dp_0}{dt} - \frac{\rho}{\mathrm{RePr}} \frac{\partial q_i}{\partial x_i}, \tag{8.1}$$

with $q_i = \lambda \frac{\partial (1/\rho)}{\partial x_i}$.

The constraint on the velocity field is then derived from a combination of (2.39) and (8.1) by elimination of the density time derivative[3].

$$-\frac{\partial \rho u}{\partial x} + u \frac{\partial \rho}{\partial x} = -\frac{\rho}{\mathrm{RePr}} \frac{\partial q}{\partial x}. \tag{8.2}$$

[2] In this chapter, a slightly modified pressure-correction algorithm, comparable to the described one of section 5.4.1, is used to stress the importance of solvability. This modification, however, does not harm the conclusions and has no further impact on the presented strategy.

[3] This is not strictly necessary. One might as well use constraint (C.9).

The discrete algorithm consists of the different substeps, given below.

8.1.1a Density Stepping

$$\frac{\rho_i^{n+1} - \rho_i^n}{\Delta t} = -\frac{u_{i+\frac{1}{2}}^n \rho_R^n - u_{i-\frac{1}{2}}^n \rho_L^n}{\Delta x}. \tag{8.3}$$

The L and R subscripts indicate extrapolated values at the left and right surface of the control volume respectively. The subscript $i + \frac{1}{2}$ refers to a simple arithmetic mean: $u_{i+\frac{1}{2}} = \frac{u_i + u_{i+1}}{2}$.

8.1.1b Velocity Predictor

The equations are solved here by means of a projection method, i.e. the intermediate state for the velocity is determined by removing the pressure from equation (2.41). The prediction of the velocity u^* is then found from:

$$\frac{\left(\rho^{n+1}u^*\right)_i - (\rho u)_i^n}{\Delta t} = -\frac{u_{i+\frac{1}{2}}^n (\rho u)_R^n - u_{i-\frac{1}{2}}^n (\rho u)_L^n}{\Delta x} + \frac{1}{\mathrm{Re}}\left(\frac{\delta\tau}{\delta x}\right)^n + \frac{\rho_i}{\mathrm{Fr}^2}. \tag{8.4}$$

The specific discretisation of the viscous fluxes is of no importance and the gravitational force is assumed to be aligned with the considered direction of the 1D problem.

8.1.1c Velocity Constraint

In discrete formulation, eq. (8.2) reads

$$\begin{aligned} -\frac{\rho_R^{n+1} u_{i+\frac{1}{2}}^{n+1} - \rho_L^{n+1} u_{i-\frac{1}{2}}^{n+1}}{\Delta x} &+ u_i^{n+1} \frac{\rho_R^{n+1} - \rho_L^{n+1}}{\Delta x} = \\ &- \frac{\rho^{n+1}}{\mathrm{RePr}} \frac{\partial}{\partial x}\left(\lambda \frac{\partial}{\partial x}\left(\frac{1}{\rho^{n+1}}\right)\right) + \frac{\rho}{\gamma p_0}\frac{dp_0}{dt}^{n+1}. \end{aligned} \tag{8.5}$$

Inserting $u_i^{n+1} = u_i^* - \Delta t \frac{p_{i+1}^{n+1} - p_{i-1}^{n+1}}{2\rho_i \Delta x}$ in the last equation results in a Poisson-like equation for the pressure. The discretisation of the conductive fluxes will be defined below.

8.1.2 Example: Conduction in a One-Dimensional Adiabatic Channel

Consider the conduction of a density jump in an adiabatic 1D channel with constant cross-section. The channel is closed at both ends, so that wall boundary conditions apply. For this case, the non-dimensional flow equations (2.39), (2.41) and (8.1) become:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} = 0, \tag{8.6}$$

$$\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u^2}{\partial x} + \frac{\partial p_2}{\partial x} = 0, \tag{8.7}$$

$$\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} = -\frac{\rho}{\mathrm{RePr}}\frac{\partial q}{\partial x}. \tag{8.8}$$

Note that the thermodynamic pressure p_0 does not change in time in an adiabatic environment.

The initial conditions are

$$\begin{aligned} \rho\,(t_0) &= \rho_0 + \rho_1 H(x - x_0), \\ u\,(t_0) &= 0, \end{aligned} \tag{8.9}$$

with $H(x)$ the Heaviside function:

$$\begin{aligned} H(x) &= 0 \quad \text{for} \quad x < 0 \\ H(x) &= 1 \quad \text{for} \quad x \geq 0. \end{aligned}$$

The equations are discretised using first order velocity upwinding for the convective terms and second order central discretisation for the conductive and pressure term. For every node a Poisson-like equation for the pressure can be derived, resulting formally in the matrix expression

$$LP = B, \tag{8.10}$$

with L the discrete Laplacian-like operator, $P = [p_1 \;\; p_2 \;\; \ldots \;\; p_N]^T$ the pressure vector and B the right hand side, containing the predicted velocity values and the conductive terms. Note that in the case of a pressure-correction method, in (8.10) we would have the vector of pressure corrections P' instead of P.

The system is singular and contains a nullspace of dimension 2, for which $LP = 0$. Indeed, two spurious modes exist: the hydrostatic pressure field $P_H = [1 \;\; 1 \;\; ... \;\; 1]^T$ and the π-wave $P_\pi = \left[1 \;\; -1 \;\; ... \;\; (-1)^{N+1}\right]^T$.

We can perform the same analysis for the transpose of the operator L, resulting again in a nullspace of dimension 2, based on 2 vectors R_H and R_π, for which $R^T L = 0$. The exact expressions for these vectors cannot easily be determined. For the set of discretised equations in the pressure-correction step to be solvable, the RHS of the equation has to fulfill certain conditions:

$$\begin{aligned} R_H^T LP &= R_H^T B = 0, \\ R_\pi^T LP &= R_\pi^T B = 0. \end{aligned} \tag{8.11}$$

It is more instructive to consider these restrictions at the level of the constraining equation for the velocity, from which the Poisson-like equation is derived. For an internal node, we can write (8.2) in a semi-discretised manner:

$$-\frac{\rho_R u_{i+\frac{1}{2}} - \rho_L u_{i-\frac{1}{2}}}{\Delta x} + u_i \frac{\rho_R - \rho_L}{\Delta x} = -\frac{\rho}{\mathrm{RePr}} \frac{\partial}{\partial x} \left(\lambda \frac{\partial}{\partial x} \left(\frac{1}{\rho} \right) \right), \tag{8.12}$$

where all variables are evaluated at time level $n+1$. If the RHS is discretised in the same manner as the LHS, system (8.10) is solvable (see also Appendix D) and the fully discretised equation becomes:

$$\begin{aligned} -\frac{\rho_R u_{i+\frac{1}{2}} - \rho_L u_{i-\frac{1}{2}}}{\Delta x} + u_i \frac{\rho_R - \rho_L}{\Delta x} &= -\frac{1}{\mathrm{RePr}} \\ &\left[\frac{\rho_R q_{i+\frac{1}{2}} - \rho_L q_{i-\frac{1}{2}}}{\Delta x} + q_i \frac{\rho_R - \rho_L}{\Delta x} \right], \end{aligned} \tag{8.13}$$

with

$$q_i = \lambda_i \frac{\frac{1}{\rho_{i+1}} - \frac{1}{\rho_{i-1}}}{2\Delta x} \tag{8.14}$$

and

$$q_{i+\frac{1}{2}} = \frac{q_i + q_{i+1}}{2} \tag{8.15}$$

For the system to be solvable, the important observation is thus that the conductive fluxes must be calculated at the nodes and interpolated towards the cell faces. Since these terms are evaluated centrally, a π-wave for the density is not noticed and can consequently increase without limitation. As a result, the spurious pressure wave gives rise to unphysical results not only for the pressure, but also for other variables, such as the density, as shown in Fig. 8.1.

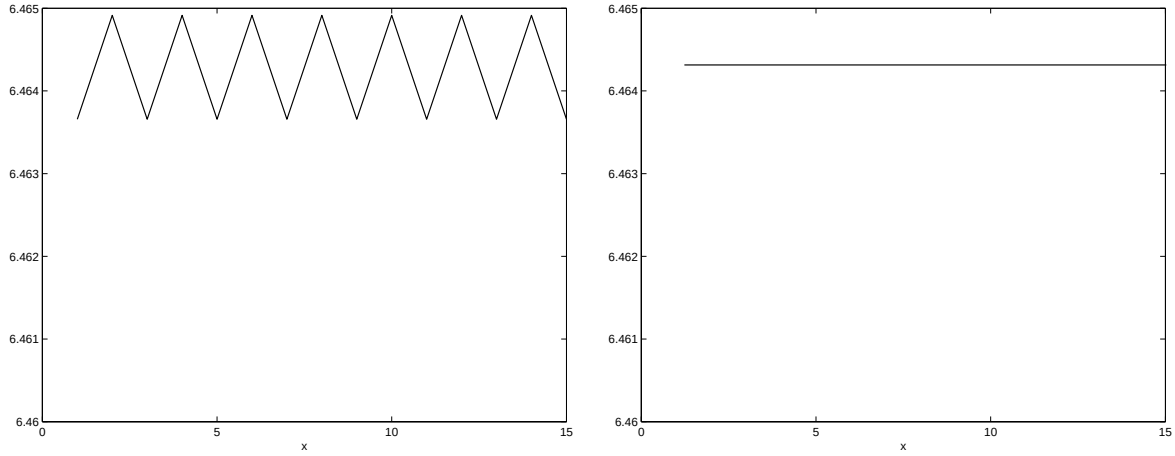

Figure 8.1: Density plot of the converged solution for the conductive heat transfer in a 1D adiabatic channel, discretised in 15 points. Initial conditions (8.9) apply with $\rho_0 = 1,\ \rho_1 = 10,\ x_0 = 7$. A spurious mode for the density appears (left), compared to the exact solution (right).

8.1.3 Solvability Condition

From the above example, it is clear that any cure for the odd-even decoupling problem, should comply with the solvability conditions of the Poisson-like pressure equation. So, a good remedy for the odd-even decoupling problem does not only remove the spurious mode P_π from the solution, but also guarantees that the resulting system is solvable. Indeed, even if the spurious mode P_π were to be removed from the solution of the above example, the resulting system would still be singular with a nullspace of dimension 1, and one condition for the RHS remains from (8.11):

$$R_H^T L P \;=\; R_H^T B = 0. \tag{8.16}$$

In general, this condition is not fulfilled if one does not apply the propositions made by e.g. [18, 19, 62] in a rigorous way, i.e. by considering the first principles of these propositions in the special case of a variable-density pressure-correction or pressure-projection method.

In addition to the solvability requirement, we prefer a discrete Laplacian of the pressure field that is easily solvable by an iterative method, i.e. a Laplacian with a compact stencil.

8.2 Adjusted Algorithm

We consider equations (2.39), (2.41) and (8.1). The basic algorithm is the same as described in section 8.1.1, discretised on a collocated mesh. However, in order to cure the odd-even decoupling, certain equations are assumed to be solved in a staggered way. Note that we use the assumption only to derive the algorithm, suitable for a collocated mesh approach. As a result of the staggered approach assumption, extra terms appear in the discretisation. In this section, we restrict ourselves to a one dimensional uniform mesh with grid spacing Δx. In section 8.3, the reasoning is generalised to general curvilinear meshes in three dimensions.

We introduce notations $\widetilde{\phi}$ and $\overline{\phi}$. The former is defined on the cell face and is calculated as the arithmetic mean of the neighbouring node values; the latter is defined in the node and is calculated as the mean of the neighbouring face values[4]:

$$\widetilde{\phi}_{i+\frac{1}{2}} = \frac{\phi_i + \phi_{i+1}}{2}, \quad \overline{\phi}_i = \frac{\phi_{i-\frac{1}{2}} + \phi_{i+\frac{1}{2}}}{2}. \tag{8.17}$$

The algorithm is first presented as such, after which the subsequent substeps, including the derivation of the correction term, are explained in greater detail, starting from known values at time level n.

8.2.1 Summary

Given an initial density field ρ_i^0, thermodynamic pressure p_0^0 and an initial velocity field u_i^0 (and kinematic pressure field p_i^0), satisfying the velocity constraint, the cured pressure-correction scheme consists of the substeps shown in the flowchart of Fig. 8.2. The correction for the cell-face velocities $\hat{\varepsilon}$ considers $\frac{\nabla p}{\rho}$ as an entity and is used in the convective terms of continuity and momentum equation. The pressure follows from a Poisson-like equation, originating from a combination of the equations of continuity and temperature.

[4]The notation must not be confused with Reynolds-averaging and Favre-averaging.

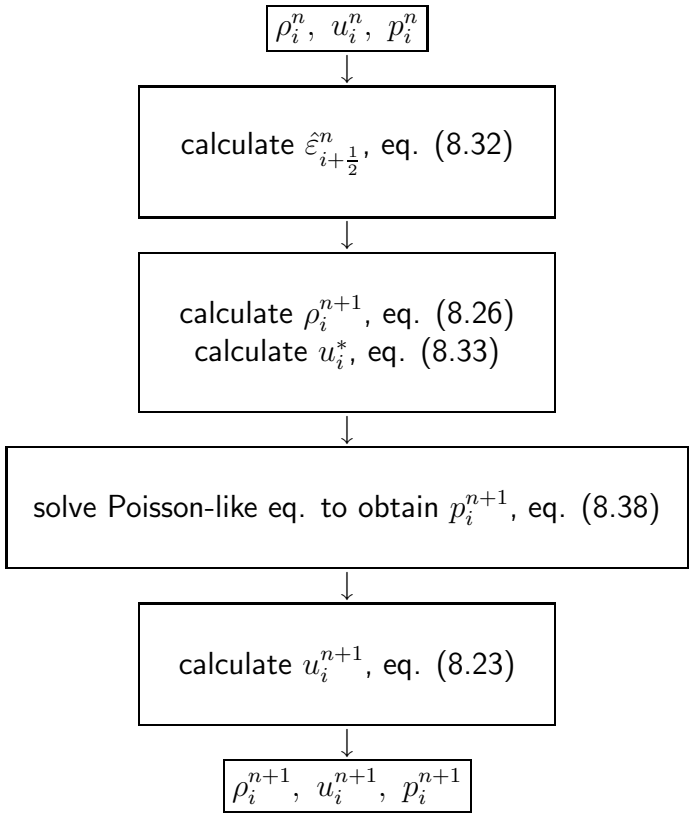

Figure 8.2: Summary of the pressure-corection algorithm, cured for odd-even decoupling.

8.2.2 Density Stepping

For now, we do not yet specify how to determine ρ_i^{n+1} and consider this quantity as known. The exact way of calculating ρ_i^{n+1} can be found under section 8.2.6.

8.2.3 Velocity Predictor

The prediction of the velocity u_i^* is done in the same way as (8.4), now using the interpolation notation:

$$\frac{\left(\rho^{n+1}u^*\right)_i - (\rho u)_i^n}{\Delta t} = -\frac{\widetilde{u}^n_{i+\frac{1}{2}}\,(\rho u)^n_R - \widetilde{u}^n_{i-\frac{1}{2}}\,(\rho u)^n_L}{\Delta x} + \frac{1}{\mathrm{Re}}\left(\frac{\delta\tau}{\delta x}\right)^n. \tag{8.18}$$

Again, the discretisation details of the viscous term are of no importance.

8.2.4 Velocity Corrector

The velocity at the new time level is now calculated using the predicted velocity field, and the correction from the pressure term. Since the pressure in collocated formulation gives rise to spurious modes in the solution, the relationship between pressure and velocity is expressed here as if the corrector step were solved on a staggered mesh. Hence, the following staggered momentum equations are thought to be solved:

$$\frac{u_{i+\frac{1}{2}}^{n+1} - \widetilde{u}_{i+\frac{1}{2}}^{*}}{\Delta t} = -\frac{1}{\widetilde{\rho}_{i+\frac{1}{2}}^{n+1}} \frac{p_{i+1}^{n+1} - p_{i}^{n+1}}{\Delta x} \tag{8.19}$$

$$\frac{u_{i-\frac{1}{2}}^{n+1} - \widetilde{u}_{i-\frac{1}{2}}^{*}}{\Delta t} = -\frac{1}{\widetilde{\rho}_{i-\frac{1}{2}}^{n+1}} \frac{p_{i}^{n+1} - p_{i-1}^{n+1}}{\Delta x}. \tag{8.20}$$

Combination of (8.19) and (8.20) gives a collocated formulation:

$$\overline{u}_{i}^{n+1} - \overline{\widetilde{u}}_{i}^{*} = -\frac{\Delta t}{2\Delta x} \left[\frac{p_{i+1}^{n+1} - p_{i}^{n+1}}{\widetilde{\rho}_{i+\frac{1}{2}}^{n+1}} + \frac{p_{i}^{n+1} - p_{i-1}^{n+1}}{\widetilde{\rho}_{i-\frac{1}{2}}^{n+1}} \right]. \tag{8.21}$$

We write the last equation in a more compact form:

$$\overline{u}_{i}^{n+1} - \overline{\widetilde{u}}_{i}^{*} = -\frac{\Delta t}{2} \left[\left. \frac{\nabla p}{\widetilde{\rho}} \right|_{i+\frac{1}{2}}^{n+1} + \left. \frac{\nabla p}{\widetilde{\rho}} \right|_{i-\frac{1}{2}}^{n+1} \right] = -\Delta t \left. \overline{\frac{\nabla p}{\widetilde{\rho}}} \right|_{i}^{n+1}. \tag{8.22}$$

However, a problem arises if we use this expression each time step. The calculated velocity field would then become too smooth because of the averaging of u^*. Indeed, on a collocated mesh, the velocity is normally calculated from

$$u_{i}^{n+1} - u_{i}^{*} = -\Delta t \left. \overline{\frac{\nabla p}{\widetilde{\rho}}} \right|_{i}^{n+1}, \tag{8.23}$$

with no averaging in the LHS. The relationship between u_i^{n+1} and $\overline{u}_i^{n+1}$ is determined from eqs. (8.23) and (8.22):

$$u_{i}^{n+1} = \overline{u}_{i}^{n+1} + \left(u_{i}^{*} - \overline{\widetilde{u}}_{i}^{*} \right). \tag{8.24}$$

8.2.5 Pressure Correction Equation

The corrected velocity u_i^{n+1} can be determined from (8.23) if the pressure field is known. The pressure is calculated from the pressure correction equation, following from a constraining equation on the velocity field. The velocity field has to satisfy the continuity equation (8.6), with (provisionally) an imposed change in density $\frac{\partial \rho}{\partial t} = \frac{\rho^{n+2}-\rho^{n+1}}{\Delta t}$:

$$\begin{aligned}
\left.\frac{\partial \rho}{\partial t}\right|_i &= -\frac{\rho_R^{n+1} u_{i+\frac{1}{2}}^{n+1} - \rho_L^{n+1} u_{i-\frac{1}{2}}^{n+1}}{\Delta x} \\
&= -\frac{\rho_R^{n+1} \widetilde{u}_{i+\frac{1}{2}}^{*} - \rho_L^{n+1} \widetilde{u}_{i-\frac{1}{2}}^{*}}{\Delta x} \\
&\quad + \frac{\Delta t}{\Delta x}\left[\rho_R^{n+1}\left.\frac{\nabla p}{\widetilde{\rho}}\right|_{i+\frac{1}{2}}^{n+1} - \rho_L^{n+1}\left.\frac{\nabla p}{\widetilde{\rho}}\right|_{i-\frac{1}{2}}^{n+1}\right],
\end{aligned} \tag{8.25}$$

where the last step uses the staggered momentum equations (8.19) and (8.20). Equation (8.25) is a Poisson equation for the pressure, where the spurious pressure mode P_π is no longer part of the solution of the system. Using the staggered equations thus eliminates the spurious mode.

8.2.6 Density Stepping (Completed)

In case of the pressure-correction algorithm in section 8.1.1, the density is determined from the conservation equation of mass. The value obtained for $\frac{\partial \rho}{\partial t} = \frac{\rho^{n+2}-\rho^{n+1}}{\Delta t}$ must be identical to (8.25) but is calculated using collocated variables:

$$\left.\frac{\partial \rho}{\partial t}\right|_i = -\frac{\rho_R^{n+1} \underline{u}_{i+\frac{1}{2}}^{n+1} - \rho_L^{n+1} \underline{u}_{i-\frac{1}{2}}^{n+1}}{\Delta x}, \tag{8.26}$$

with $\underline{u}$ a corrected interpolation, given by

$$\underline{u}_{i+\frac{1}{2}} = \widetilde{\overline{u}}_{i+\frac{1}{2}} + \varepsilon_{i+\frac{1}{2}}. \tag{8.27}$$

Equalisation of (8.25) and (8.26) yields, for the flux at the right cell face:

$$\rho_R^{n+1} \widetilde{u}_{i+\frac{1}{2}}^{*} - \Delta t \rho_R^{n+1}\left.\frac{\nabla p}{\widetilde{\rho}}\right|_{i+\frac{1}{2}}^{n+1} = \rho_R^{n+1} \underline{u}_{i+\frac{1}{2}}^{n+1}. \tag{8.28}$$

For the corrector equation (8.21), we can further elaborate the previous expression:

$$\begin{aligned}
\Leftrightarrow \widetilde{u}^*_{i+\frac{1}{2}} - \Delta t \left.\frac{\nabla p}{\widetilde{\rho}}\right|^{n+1}_{i+\frac{1}{2}} &= \widetilde{\overline{u}}^{n+1}_{i+\frac{1}{2}} + \varepsilon^{n+1}_{i+\frac{1}{2}} \\
&= \frac{\overline{u}^{n+1}_i + \overline{u}^{n+1}_{i+1}}{2} + \varepsilon^{n+1}_{i+\frac{1}{2}} \\
&= \frac{\overline{\widetilde{u}}^*_i + \overline{\widetilde{u}}^*_{i+1}}{2} - \frac{\Delta t}{2}\left[\left.\frac{\overline{\nabla p}}{\widetilde{\rho}}\right|^{n+1}_{i} + \left.\frac{\overline{\nabla p}}{\widetilde{\rho}}\right|^{n+1}_{i+1}\right] + \varepsilon^{n+1}_{i+\frac{1}{2}} \\
&= \widetilde{\overline{u}}^*_{i+\frac{1}{2}} - \Delta t \left.\frac{\widetilde{\overline{\nabla p}}}{\widetilde{\rho}}\right|^{n+1}_{i+\frac{1}{2}} + \varepsilon^{n+1}_{i+\frac{1}{2}}, \qquad (8.29)
\end{aligned}$$

from which the correction factor follows:

$$\varepsilon^{n+1}_{i+\frac{1}{2}} = \left[\widetilde{u}^*_{i+\frac{1}{2}} - \widetilde{\overline{u}}^*_{i+\frac{1}{2}}\right] - \Delta t \left[\left.\frac{\nabla p}{\widetilde{\rho}}\right|^{n+1}_{i+\frac{1}{2}} - \left.\frac{\widetilde{\overline{\nabla p}}}{\widetilde{\rho}}\right|^{n+1}_{i+\frac{1}{2}}\right]. \qquad (8.30)$$

If we redefine the corrected interpolation (8.27) as

$$\underline{\phi}_{i+\frac{1}{2}} = \widetilde{\phi}_{i+\frac{1}{2}} + \hat{\varepsilon}_{i+\frac{1}{2}}, \qquad (8.31)$$

for the corrector equation (8.23), the new correction factor $\hat{\varepsilon}$ becomes[5]

$$\hat{\varepsilon}^{n+1}_{i+\frac{1}{2}} = -\Delta t \left[\left.\frac{\nabla p}{\widetilde{\rho}}\right|^{n+1}_{i+\frac{1}{2}} - \left.\frac{\widetilde{\overline{\nabla p}}}{\widetilde{\rho}}\right|^{n+1}_{i+\frac{1}{2}}\right], \qquad (8.32)$$

which is the analogon of expression (27) in [62].

For consistency, the same cell-face velocity interpolation is used in the velocity predictor step, so that (8.18) now becomes:

$$\frac{\left(\rho^{n+1}u^*\right)_i - (\rho u)^n_i}{\Delta t} = -\frac{\underline{u}^n_{i+\frac{1}{2}}\,(\rho u)^n_R - \underline{u}^n_{i-\frac{1}{2}}\,(\rho u)^n_L}{\Delta x} + \frac{1}{\mathrm{Re}}\left(\frac{\delta\tau}{\delta x}\right)^n. \qquad (8.33)$$

[5]The correction factor does not depend on the actual pressure-correction scheme used. This way, the slight modification of the algorithm, used throughout this chapter to stress the solvability condition, has no impact on the final result, displayed here.

8.2.7 Correction Equation (Completed)

Equation (8.25) is now completed for the pressure field to be calculated. To fix thoughts, the above-mentioned pressure-correction algorithm is considered, although other alternatives are possible as well [58]. The variation in time of density $\frac{\partial \rho}{\partial t}$ is obtained from the equation of temperature (8.8), discretised as:

$$\begin{aligned}\left.\frac{\partial \rho}{\partial t}\right|_i &= -\overline{u}_i^{n+1}\frac{\rho_R^{n+1}-\rho_L^{n+1}}{\Delta x} - \rho\frac{\partial q}{\partial x} \\ &= \left[\overline{\widetilde{u}}_i^* - \Delta t\left.\overline{\frac{\nabla p}{\widetilde{\rho}}}\right|_i^{n+1}\right]\frac{\rho_R^{n+1}-\rho_L^{n+1}}{\Delta x} - \frac{\rho}{\mathrm{RePr}}\frac{\partial q}{\partial x}.\end{aligned} \tag{8.34}$$

Elimination of $\frac{\partial \rho}{\partial t}$ from (8.34) and (8.25), results in the pressure Poisson-like equation

$$\begin{aligned}&\frac{\rho_R^{n+1}+\rho_L^{n+1}}{2}\frac{\Delta t}{\Delta x}\left[\left.\frac{\nabla p}{\widetilde{\rho}}\right|_{i+\frac{1}{2}}^{n+1} - \left.\frac{\nabla p}{\widetilde{\rho}}\right|_{i-\frac{1}{2}}^{n+1}\right] = \\ &\qquad\frac{\rho_R^{n+1}+\rho_L^{n+1}}{2}\frac{\widetilde{u}_{i+\frac{1}{2}}^* - \widetilde{u}_{i-\frac{1}{2}}^*}{\Delta x} - \frac{\rho}{\mathrm{RePr}}\frac{\partial q}{\partial x}.\end{aligned} \tag{8.35}$$

This set of equations can again formally be written as (8.10), with L now a matrix with a nullspace of dimension 1 (the hydrostatic pressure field). We can perform the same analysis for the system to be solvable as in example 8.1.2, resulting in a discretisation of the conductive term under the form:

$$\rho\frac{\partial q}{\partial x} = \frac{\rho_R^{n+1}+\rho_L^{n+1}}{2}\frac{q_{i+\frac{1}{2}}^{n+1} - q_{i-\frac{1}{2}}^{n+1}}{\Delta x}, \tag{8.36}$$

with

$$q_{i+\frac{1}{2}}^{n+1} = \lambda_i\frac{\frac{1}{\rho_{i+1}^{n+1}} - \frac{1}{\rho_i^{n+1}}}{\Delta x}. \tag{8.37}$$

Using (8.36), (8.35) is simplified to

$$\frac{\Delta t}{\Delta x}\left[\left.\frac{\nabla p}{\widetilde{\rho}}\right|_{i+\frac{1}{2}}^{n+1} - \left.\frac{\nabla p}{\widetilde{\rho}}\right|_{i-\frac{1}{2}}^{n+1}\right] = \frac{\widetilde{u}_{i+\frac{1}{2}}^* - \widetilde{u}_{i-\frac{1}{2}}^*}{\Delta x} - \frac{1}{\mathrm{RePr}}\frac{q_{i+\frac{1}{2}}^{n+1} - q_{i-\frac{1}{2}}^{n+1}}{\Delta x}. \tag{8.38}$$

Note that correction term (8.30), although apparent in the calculation of the density at the new time level, does not appear in the Poisson-like equation for the pressure. Also note that eq. (8.38) does not contain any upwind values. This is an advantage in terms of efficiency, since in general the sign of the velocity components at the new time level is not known. Thus, if upwind values appeared, the Laplacian operator would need to be evaluated every iteration per time step. Here, only one evaluation per time step is required.

8.3 Extension to General Curvilinear Coordinate Systems in Three Dimensions

The extension of the previous approach towards three-dimensional Cartesian equidistant meshes is straightforward. Application of the method on general structured grids is still possible by means of a rigourous deduction, based on the invariant formulation of the Navier-Stokes equations. In this formulation the physical domain is mapped onto a rectangular block. Hence the curvilinear grid is mapped onto a Cartesian equidistant grid, on which the above solution method can be applied. From now onwards, we use the notations corresponding to the mapping theory [49, 64]. This implies that the coordinate indices are written in superscript.

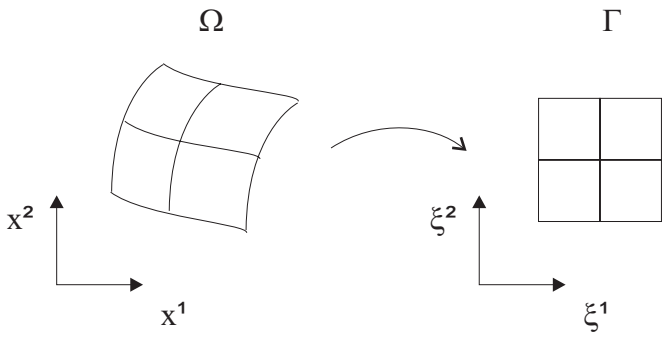

Figure 8.3: Mapping of a physical domain Ω onto a rectangular block Γ.

8.3.1 Finite Volume Formulation in General Coordinates

We note the Navier-Stokes equations in an invariant formulation, according to [49, 64]. The equations on a Cartesian grid are the following [58]:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x^i} &= 0 \\ \left(\frac{\partial \rho u_j}{\partial t}\right)^{\diamond} + \frac{\partial p}{\partial x^j} &= 0 \\ \frac{\partial \rho}{\partial t} + u_i \frac{\partial \rho}{\partial x^i} &= \mathrm{Cond}(\rho), \end{aligned} \tag{8.39}$$

with

$$\mathrm{Cond}(\rho) = -\frac{\rho}{\mathrm{RePr}} \frac{\partial}{\partial x^i} \left(\lambda \frac{\partial}{\partial x^i} \left(\frac{1}{\rho} \right) \right) \tag{8.40}$$

The diamond ($^{\diamond}$) is introduced because the convective and diffusive transport in the momentum equation are not written, so that the update in time is done from state $(\rho u)^*$ to state $(\rho u)^{n+1}$:

$$\begin{aligned} \left(\frac{\partial \rho u_j}{\partial t}\right) + \frac{\partial \rho u_i u_j}{\partial x^i} - \frac{1}{\mathrm{Re}} \frac{\partial \tau_{ij}}{\partial x^i} + \frac{\partial p}{\partial x^j} &= 0 \\ \Leftrightarrow \left(\frac{\partial \rho u_j}{\partial t}\right)^{*} + \left(\frac{\partial \rho u_j}{\partial t}\right)^{\diamond} + \frac{\partial \rho u_i u_j}{\partial x^i} - \frac{1}{\mathrm{Re}} \frac{\partial \tau_{ij}}{\partial x^i} + \frac{\partial p}{\partial x^j} &= 0, \end{aligned}$$

with $\left(\frac{\partial \rho u_j}{\partial t}\right)^{*}$ following from the predictor step

$$\left(\frac{\partial \rho u_j}{\partial t}\right)^{*} + \frac{\partial \rho u_i u_j}{\partial x^i} - \frac{1}{\mathrm{Re}} \frac{\partial \tau_{ij}}{\partial x^i} = 0.$$

Written in integral formulation the equations become:

$$\begin{aligned} \iiint_{\Omega} \frac{\partial \rho}{\partial t} \,\mathrm{d}V + \iint_{\partial\Omega} \rho \boldsymbol{u} \cdot \boldsymbol{n} \,\mathrm{d}S &= 0 \\ \iiint_{\Omega} \left(\frac{\partial \rho \boldsymbol{u}}{\partial t}\right)^{\diamond} \mathrm{d}V + \iint_{\partial\Omega} p \boldsymbol{n} \,\mathrm{d}S &= 0 \\ \iiint_{\Omega} \frac{\partial \rho}{\partial t} \,\mathrm{d}V + \iiint_{\Omega} \boldsymbol{u} \cdot \boldsymbol{\nabla} \rho \,\mathrm{d}V &= \iiint_{\Omega} \mathrm{Cond}(\rho) \,\mathrm{d}V. \end{aligned} \tag{8.41}$$

The above equations are transformed to the coordinate system (ξ^1, ξ^2, ξ^3), with metric tensor

$$\begin{aligned} g_{\alpha\beta} &= \frac{\partial x^k}{\partial \xi^\alpha}\frac{\partial x^k}{\partial \xi^\beta} \\ g &= det\,(g_{\alpha\beta}) \end{aligned} \tag{8.42}$$

and the following properties:

$$\begin{aligned} \mathrm{d}V &= \sqrt{g}\,\mathrm{d}\xi^1\,\mathrm{d}\xi^2\,\mathrm{d}\xi^3 \\ \boldsymbol{n}_{(\alpha)}\,\mathrm{d}S &= \boldsymbol{a}^{(\alpha)}\sqrt{g}\,\mathrm{d}\xi^\beta\,\mathrm{d}\xi^\gamma. \end{aligned} \tag{8.43}$$

with $\boldsymbol{a}^{(\alpha)}$ the contravariant basevector, perpendicular to the ξ^β and ξ^γ coordinate lines:

$$\boldsymbol{a}_k^{(\alpha)} = \frac{\partial \xi^\alpha}{\partial x^k}. \tag{8.44}$$

Hence (8.41) becomes

$$\iiint_\Gamma \frac{\partial \rho}{\partial t}\sqrt{g}\,\mathrm{d}\xi^1\,\mathrm{d}\xi^2\,\mathrm{d}\xi^3 + \iint_{\partial\Gamma} \rho\boldsymbol{u}\cdot\boldsymbol{a}^{(\alpha)}\sqrt{g}\,\mathrm{d}\xi^\beta\,\mathrm{d}\xi^\gamma = 0 \tag{8.45}$$

$$\iiint_\Gamma \left(\frac{\partial \rho\boldsymbol{u}}{\partial t}\right)^\diamond \sqrt{g}\,\mathrm{d}\xi^1\,\mathrm{d}\xi^2\,\mathrm{d}\xi^3 + \iint_{\partial\Gamma} p\boldsymbol{a}^{(\alpha)}\sqrt{g}\,\mathrm{d}\xi^\beta\,\mathrm{d}\xi^\gamma = 0 \tag{8.46}$$

$$\iiint_\Gamma \frac{\partial \rho}{\partial t}\sqrt{g}\,\mathrm{d}\xi^1\,\mathrm{d}\xi^2\,\mathrm{d}\xi^3 + \iiint_\Gamma \boldsymbol{u}\cdot\boldsymbol{\nabla}\rho\sqrt{g}\,\mathrm{d}\xi^1\,\mathrm{d}\xi^2\,\mathrm{d}\xi^3 = \iiint_\Omega \mathrm{Cond}(\rho)\,\mathrm{d}V. \tag{8.47}$$

The gradient is transformed into

$$\begin{aligned} \boldsymbol{\nabla}\phi &= \frac{\partial \phi}{\partial \xi^\alpha}\boldsymbol{a}_\beta^{(\alpha)}\boldsymbol{a}_\beta^{(\gamma)}\boldsymbol{a}_{(\gamma)} \\ &= \frac{\partial \phi}{\partial \xi^\alpha}g^{\alpha\gamma}\boldsymbol{a}_{(\gamma)} \\ &= \frac{\partial \phi}{\partial \xi^\alpha}\boldsymbol{a}^{(\alpha)} \end{aligned} \tag{8.48}$$

Thus, equation (8.47) becomes:

$$\iiint_\Gamma \frac{\partial \rho}{\partial t}\sqrt{g}\,\mathrm{d}\xi^1\,\mathrm{d}\xi^2\,\mathrm{d}\xi^3 + \iiint_\Gamma \boldsymbol{u}\cdot\frac{\partial \rho}{\partial \xi^\alpha}\boldsymbol{a}^{(\alpha)}\sqrt{g}\,\mathrm{d}\xi^1\,\mathrm{d}\xi^2\,\mathrm{d}\xi^3 = \iiint_\Omega \mathrm{Cond}(\rho)\,\mathrm{d}V. \tag{8.49}$$

The inner product $\boldsymbol{u} \cdot \boldsymbol{a}^{(\alpha)}$ equals, by definition, the contravariant component U^α of the velocity vector $\boldsymbol{u}$:

$$U^\alpha = \boldsymbol{u} \cdot \boldsymbol{a}^{(\alpha)} \tag{8.50}$$

It is known [49, 64] that it is better to consider the contravariant fluxes V^α since they are continuous in the entire domain:

$$V^\alpha = U^\alpha \sqrt{g} \tag{8.51}$$

Equation (8.45) then becomes

$$\iiint_\Gamma \frac{\partial \rho}{\partial t} \sqrt{g}\, \mathrm{d}\xi^1\, \mathrm{d}\xi^2\, \mathrm{d}\xi^3 + \iint_{\partial\Gamma} \rho V^\alpha\, \mathrm{d}\xi^\beta\, \mathrm{d}\xi^\gamma \;=\; 0, \tag{8.52}$$

while equation (8.49) reads

$$\iiint_\Gamma \frac{\partial \rho}{\partial t} \sqrt{g}\, \mathrm{d}\xi^1\, \mathrm{d}\xi^2\, \mathrm{d}\xi^3 + \iiint_\Gamma \frac{\partial \rho}{\partial \xi^\alpha} V^\alpha\, \mathrm{d}\xi^1\, \mathrm{d}\xi^2\, \mathrm{d}\xi^3 \;=\; \iiint_\Omega \mathrm{Cond}(\rho)\, \mathrm{d}V. \tag{8.53}$$

The finite volume formulation for the control volume around the node with index θ immediately follows:

$$|\Omega_\theta| \left(\frac{\partial \rho}{\partial t}\right)_\theta + \sum_\alpha \left[\rho V^\alpha\right]_{\partial\Gamma_{\theta_{\alpha-}}}^{\partial\Gamma_{\theta_{\alpha+}}} \;=\; 0 \tag{8.54}$$

$$|\Omega_\theta| \left(\frac{\partial \rho}{\partial t}\right)_\theta + \sum_\alpha \left(V^\alpha\right)_\theta \left[\rho\right]_{\partial\Gamma_{\theta_{\alpha-}}}^{\partial\Gamma_{\theta_{\alpha+}}} \;=\; \iiint_\Omega Cond(\rho)\, \mathrm{d}V. \tag{8.55}$$

$\partial\Gamma_{\theta_{\alpha+}}$ indicates the part in the positive α-direction of the boundary face of the control volume around the node with index θ, so that the summation sums over all 6 faces of the control volume.

The momentum equations (8.46) can also be written in terms of the contravariant fluxes. Therefore the inner product of (8.46) and the contravariant basevector, averaged over the control volume, $\bar{\boldsymbol{a}}_\theta^{(\alpha)}$ is taken:

$$\iiint_\Gamma \bar{\boldsymbol{a}}_\theta^{(\alpha)} \cdot \left(\frac{\partial \rho \boldsymbol{u}}{\partial t}\right)^\diamond \sqrt{g}\, \mathrm{d}\xi^1\, \mathrm{d}\xi^2\, \mathrm{d}\xi^3 + \bar{\boldsymbol{a}}_\theta^{(\alpha)} \cdot \iint_{\partial\Gamma} p \boldsymbol{a}^{(\delta)} \sqrt{g}\, \mathrm{d}\xi^\beta\, \mathrm{d}\xi^\gamma \;=\; 0, \tag{8.56}$$

which gives in finite volume formulation

$$\left(\frac{\partial \rho V^\alpha}{\partial t}\right)_\theta^\diamond + \bar{\boldsymbol{a}}_\theta^{(\alpha)} \cdot \sum_\delta \left[p\boldsymbol{a}^{(\delta)}\sqrt{g}\right]_{\partial\Gamma_{\theta_{\delta-}}}^{\partial\Gamma_{\theta_{\delta+}}} = 0. \tag{8.57}$$

8.3.2 Algorithm in General Curvilinear Coordinates

Since eqs. (8.54), (8.55) and (8.57) are very similar to the original transport equation in one dimension, a flux correction term can be derived in the same fashion, provided that the derivation is done on the Cartesian coordinate system (ξ^1, ξ^2, ξ^3) with the contravariant flux vectors V^α as velocity unknowns. The only difficulty appears in the pressure term, which is now more complicated, as can be seen in (8.57). We make abstraction of this pressure term by the definition

$$\mathcal{P}_\theta = \bar{\boldsymbol{a}}_\theta^{(\alpha)} \cdot \sum_\delta \left[p\boldsymbol{a}^{(\delta)}\sqrt{g}\right]_{\partial\Gamma_{\theta_{\delta-}}}^{\partial\Gamma_{\theta_{\delta+}}}. \tag{8.58}$$

Equations (8.19), (8.22), (8.25), (8.30), (8.32) and (8.38) become respectively

$$\frac{V^{\alpha n+1}_{\theta_{\alpha+}} - \widetilde{V^\alpha}^*_{\theta_{\alpha+}}}{\Delta t} = -\frac{1}{\widetilde{\rho}^{n+1}_{\theta_{\alpha+}}}\mathcal{P}_{\theta_{\alpha+}}; \tag{8.59}$$

$$\overline{V^\alpha}^{n+1}_\theta - \overline{\widetilde{V^\alpha}}^*_\theta = -\Delta t \left.\frac{\overline{\mathcal{P}}}{\widetilde{\rho}}\right|_\theta^{n+1}; \tag{8.60}$$

$$\begin{aligned}
|\Omega_\theta|\left(\frac{\partial \rho}{\partial t}\right)_\theta &= -\sum_\alpha \left[\rho V^\alpha\right]_{\partial\Gamma_{\theta_{\alpha-}}}^{\partial\Gamma_{\theta_{\alpha+}}} \\
&= -\sum_\alpha \left[\rho^{n+1}_{\alpha R} V^{\alpha n+1}_{\theta_{\alpha+}} - \rho^{n+1}_{\alpha L} V^{\alpha n+1}_{\theta_{\alpha-}}\right] \\
&= -\sum_\alpha \left[\rho^{n+1}_{\alpha R} \widetilde{V^\alpha}^*_{\theta_{\alpha+}} - \rho^{n+1}_{\alpha L} \widetilde{V^\alpha}^*_{\theta_{\alpha-}}\right] \\
&\quad +\Delta t \sum_\alpha \left[\rho^{n+1}_{\alpha R} \left.\frac{\mathcal{P}}{\widetilde{\rho}}\right|_{\theta_{\alpha+}}^{n+1} - \rho^{n+1}_{\alpha L} \left.\frac{\mathcal{P}}{\widetilde{\rho}}\right|_{\theta_{\alpha+}}^{n+1}\right];
\end{aligned} \tag{8.61}$$

$$\varepsilon^{\alpha}_{\theta_{\alpha+}} = \left[\widetilde{V^{\alpha}}^{*}_{\theta_{\alpha+}} - \widetilde{\overline{V^{\alpha}}}^{*}_{\theta_{\alpha+}} \right] - \Delta t \left[\left. \frac{\mathcal{P}}{\widetilde{\rho}} \right|^{n+1}_{\theta_{\alpha+}} - \left. \frac{\widetilde{\overline{\mathcal{P}}}}{\widetilde{\rho}} \right|^{n+1}_{\theta_{\alpha+}} \right]; \tag{8.62}$$

$$\hat{\varepsilon}^{\alpha}_{\theta_{\alpha+}} = -\Delta t \left[\left. \frac{\mathcal{P}}{\widetilde{\rho}} \right|^{n+1}_{\theta_{\alpha+}} - \left. \frac{\widetilde{\overline{\mathcal{P}}}}{\widetilde{\rho}} \right|^{n+1}_{\theta_{\alpha+}} \right]. \tag{8.63}$$

$$\Delta t \left[\left. \frac{\mathcal{P}}{\widetilde{\rho}} \right|^{n+1}_{\theta_{\alpha+}} - \left. \frac{\mathcal{P}}{\widetilde{\rho}} \right|^{n+1}_{\theta_{\alpha-}} \right] = \left[\widetilde{V^{\alpha}}^{*}_{\theta_{\alpha+}} - \widetilde{V^{\alpha}}^{*}_{\theta_{\alpha-}} \right] - \left[Q^{\alpha n+1}_{\theta_{\alpha+}} - Q^{\alpha n+1}_{\theta_{\alpha-}} \right]; \tag{8.64}$$

In (8.64), Q denotes the discretised version of the heat flux, written in terms of ρ and evaluated at the cell faces. Using these equations, the algorithm is very similar to the one for Cartesian coordinate systems (Fig. 8.2).

8.3.3 Remarks

8.3.3a Flux-Velocity Relation

The above equations consider the contravariant flux as primary variable. Since we are interested in the velocity itself in physical space, we apply the following conversion formulas:

$$\begin{aligned} V^{\alpha}{}_{\theta} &= \iiint_{\Gamma} V^{\alpha} \,\mathrm{d}\xi^1 \,\mathrm{d}\xi^2 \,\mathrm{d}\xi^3 \\ &= \iiint_{\Gamma} \boldsymbol{a}^{(\alpha)} \cdot \boldsymbol{u} \sqrt{g} \,\mathrm{d}\xi^1 \,\mathrm{d}\xi^2 \,\mathrm{d}\xi^3 \\ &= \iiint_{\Omega} \boldsymbol{a}^{(\alpha)} \cdot \boldsymbol{u} \,\mathrm{d}V \\ &= \iiint_{\Omega} \boldsymbol{a}^{(\alpha)} \,\mathrm{d}V \cdot \boldsymbol{u}_{\theta}. \end{aligned} \tag{8.65}$$

The inverse formula yields

$$\begin{aligned} {u^\alpha}_\theta &= \frac{1}{\Omega}\iiint_\Omega u^\alpha \,\mathrm{d}V \\ &= \frac{1}{\Omega}\iiint_\Omega \frac{\boldsymbol{a}_{(\alpha)}\cdot\boldsymbol{V}}{\sqrt{g}}\,\mathrm{d}V \\ &= \frac{1}{\Omega}\iiint_\Gamma \boldsymbol{a}_{(\alpha)}\cdot\boldsymbol{V}\,\mathrm{d}\xi^1\,\mathrm{d}\xi^2\,\mathrm{d}\xi^3 \\ &= \frac{1}{\Omega}\iiint_\Gamma \boldsymbol{a}_{(\alpha)}\,\mathrm{d}\xi^1\,\mathrm{d}\xi^2\,\mathrm{d}\xi^3\cdot\boldsymbol{V}_\theta. \end{aligned} \tag{8.66}$$

The integrals only contain geometrical quantities and can be exactly calculated. Remark that the flux at the face is calculated as the average of the fluxes at the neighbouring nodes, which is not the same as calculating the flux from the face velocity, which would be the average of the neighbouring node velocities:

$$\begin{aligned} {V^\alpha}_{\theta_{\alpha+}} &= \frac{{V^\alpha}_{\theta_\alpha}+{V^\alpha}_{\theta_{\alpha++}}}{2} \\ &= \frac{1}{2}\left[\iiint_{\Omega_\theta}\boldsymbol{a}^{(\alpha)}\,\mathrm{d}V\cdot\boldsymbol{u}_\theta+\iiint_{\Omega_{\theta_{\alpha++}}}\boldsymbol{a}^{(\alpha)}\,\mathrm{d}V\cdot\boldsymbol{u}_{\theta_{\alpha++}}\right] \\ &\neq \iiint_{\Omega_{\theta_{\alpha+}}}\boldsymbol{a}^{(\alpha)}\,\mathrm{d}V\cdot\frac{\boldsymbol{u}_{\theta_\alpha}+\boldsymbol{u}_{\theta_{\alpha++}}}{2} \end{aligned} \tag{8.67}$$

8.3.3b Evaluation of $\mathcal{P}$

The evaluation of $\mathcal{P}$ is complicated but can be done in a straightforward manner. It is important to note that the evaluation at the surfaces of the term $p\boldsymbol{a}^{(\delta)}\sqrt{g}$ is well defined since the expression is continuous over the surface. Remark that here the exact geometrical quantity $\boldsymbol{a}^{(\delta)}\sqrt{g}$ can be used, integrated over the surface area, consistent with (8.56), from which this term originates.

8.4 Discussion: The Failure of Other Cell-Face Velocity Interpolations

8.4.1 The Cure, Seen from a Different Perspective

Let us reconsider the example of a 1D adiabatic channel (section 8.1.2). The uncured algorithm results in a Laplacian-like operator with a stencil, clearly indicating the odd-even decoupling:

$$\left[\begin{array}{cccccc} \frac{1}{\rho_{i-1}} & 0 & -\frac{1}{\rho_{i-1}}-\frac{1}{\rho_{i+1}} & 0 & \frac{1}{\rho_{i+1}} \end{array}\right] \tag{8.68}$$

The adjusted algorithm, however, eliminates the spurious mode, resulting in a compact stencil:

$$\left[\begin{array}{cccccc} 0 & \frac{1}{\rho_{i-\frac{1}{2}}} & -\frac{1}{\rho_{i-\frac{1}{2}}}-\frac{1}{\rho_{i+\frac{1}{2}}} & \frac{1}{\rho_{i+\frac{1}{2}}} & 0 \end{array}\right] \tag{8.69}$$

In constant density flows, the effect of the correction term $\hat{\varepsilon}$ can be visualised in stencil notation: by adding the correction term, the wide stencil is compacted [62, 68]:

$$\begin{array}{rl} & [\ 1\ \ 0\ \ -2\ \ 0\ \ 1\ ]_{\text{wide stencil}} \\ +& [\ -1\ \ 3\ \ -3\ \ 1\ \ 0\ ]_{\hat{\varepsilon}\text{ on the left face}} \\ +& [\ 0\ \ 1\ \ -3\ \ 3\ \ -1\ ]_{\hat{\varepsilon}\text{ on the right face}} \\ \hline =& [\ 0\ \ 4\ \ -8\ \ 4\ \ 0\ ]_{\text{compact stencil}} \end{array} \tag{8.70}$$

Consequently, adding the correction term to the non-cured constraining equation, results in the correct constraining equation. This observation does not hold in variable density flow. Indeed:

$$\begin{array}{rl} & \left[\begin{array}{ccccc} \frac{1}{\rho_{i-1}} & 0 & -\frac{1}{\rho_{i-1}}-\frac{1}{\rho_{i+1}} & 0 & \frac{1}{\rho_{i+1}} \end{array}\right] \\ +& \left[\begin{array}{ccccc} -\frac{1}{\rho_{i-\frac{3}{2}}} & \frac{1}{\rho_{i-\frac{3}{2}}}+\frac{2}{\rho_{i-\frac{1}{2}}} & -\frac{2}{\rho_{i-\frac{1}{2}}}-\frac{1}{\rho_{i+\frac{1}{2}}} & \frac{1}{\rho_{i+\frac{1}{2}}} & 0 \end{array}\right] \\ +& \left[\begin{array}{ccccc} 0 & \frac{1}{\rho_{i-\frac{1}{2}}} & -\frac{1}{\rho_{i-\frac{1}{2}}}-\frac{2}{\rho_{i+\frac{1}{2}}} & \frac{2}{\rho_{i+\frac{1}{2}}}+\frac{1}{\rho_{i+\frac{3}{2}}} & -\frac{1}{\rho_{i+\frac{3}{2}}} \end{array}\right] \\ \hline \neq& \left[\begin{array}{ccccc} 0 & \frac{4}{\rho_{i-\frac{1}{2}}} & -\frac{4}{\rho_{i-\frac{1}{2}}}-\frac{4}{\rho_{i+\frac{1}{2}}} & \frac{4}{\rho_{i+\frac{1}{2}}} & 0 \end{array}\right] \end{array} \tag{8.71}$$

The wide stencil, for which the above reasoning does hold, is:

$$\left[\begin{array}{cccccc} \frac{1}{\rho_{i-\frac{3}{2}}} & -\frac{1}{\rho_{i-\frac{3}{2}}}+\frac{1}{\rho_{i-\frac{1}{2}}} & -\frac{1}{\rho_{i-\frac{1}{2}}}-\frac{1}{\rho_{i+\frac{1}{2}}} & \frac{1}{\rho_{i+\frac{1}{2}}}-\frac{1}{\rho_{i+\frac{3}{2}}} & \frac{1}{\rho_{i-\frac{3}{2}}} \end{array}\right] \tag{8.72}$$

which is hard to see a priori.

8.4.2 Demand for Solvability

As explained in [68] for constant-density flows, the choice for the compacted Laplacian-like operator is arbitrary, as long as it is consistent. In variable density flows, where conductive effects enter, an extra condition (the solvability condition) appears. We have shown above that, using the present approach, we end up with a pressure equation which can be solved. To that purpose (as explained in section 8.1.2) the conductive term requires a special discretisation.

8.4.3 How to Make Other Approaches Work

From section 8.4.1, we saw that (8.72) can be seen as the basic wide stencil. If we use the correction term $\hat{\varepsilon}$, as defined in (8.32), a regular compact stencil is obtained. If other cell-face velocity interpolations would have been used, a modified stencil for the Poisson equation would be the result. In general, the correction terms take the following stencil, for the left and right face respectively:

$$\begin{matrix} [& a_1 & a_2 & a_3 & a_4 & 0 &] \\ [& 0 & -b_1 & -b_2 & -b_3 & -b_4 &], \end{matrix} \tag{8.73}$$

where the coefficients must be chosen in such a way that the resulting expression for $\hat{\varepsilon}$ takes the form

$$\hat{\varepsilon}_L = f\left(\left.\frac{\nabla p}{\rho}\right|_{i-\frac{3}{2}}, \left.\frac{\nabla p}{\rho}\right|_{i-\frac{1}{2}}, \left.\frac{\nabla p}{\rho}\right|_{i+\frac{1}{2}} \right). \tag{8.74}$$

The resulting *'compact'* stencil takes the form:

$$\begin{matrix} [& \frac{1}{\rho_{i-\frac{3}{2}}} + a_1 & -\frac{1}{\rho_{i-\frac{3}{2}}} + \frac{1}{\rho_{i-\frac{1}{2}}} + a_2 + b_1 & -\frac{1}{\rho_{i-\frac{1}{2}}} - \frac{1}{\rho_{i+\frac{1}{2}}} + a_3 + b_2 & \dots \\ & \dots & \frac{1}{\rho_{i+\frac{1}{2}}} - \frac{1}{\rho_{i+\frac{3}{2}}} + a_4 + b_3 & \frac{1}{\rho_{i-\frac{3}{2}}} + b_4 &], \end{matrix} \tag{8.75}$$

which can be written again as

$$\nabla u' = \mathcal{F}\left(\left.\frac{\nabla p}{\rho}\right|_{i-\frac{3}{2}}, \left.\frac{\nabla p}{\rho}\right|_{i-\frac{1}{2}}, \left.\frac{\nabla p}{\rho}\right|_{i+\frac{1}{2}}, \left.\frac{\nabla p}{\rho}\right|_{i+\frac{3}{2}} \right). \tag{8.76}$$

This defines the discretisation of the conductive term:

$$\nabla q = \mathcal{F}\left(q_{i-\frac{3}{2}}, q_{i-\frac{1}{2}}, q_{i+\frac{1}{2}}, q_{i+\frac{3}{2}}\right). \tag{8.77}$$

8.4.4 Traps with Other Approaches

If other approaches are used, special requirements are needed for the discretisation. In general, these requirements are not incorporated in the codes and the correction term for the cell-face velocity is seen merely as an ad-hoc adjustment. There are however consequences involved. Some traps are listed:

- The correction term cannot be written as (8.74). Indeed: many used correction terms originate from constant density flow calculations, and even in variable density equations, the pressure Poisson equation is generally written as a constant coefficient Poisson equation, such that the constant density behaviour is retained. If this condition is not fulfilled, it is not clear what a solvability condition should look like and an a priori discretisation of the conductive terms cannot be determined.

- The conductive terms are not discretised properly. Since (8.77) can become complicated, it is tempting to take a simpler discretisation for the conductive term. Unfortunately, the system becomes unsolvable, and elimination of one equation (to fix the pressure level) results in different solutions, depending on the equation eliminated.

- The compact stencil (8.69) is used, with a general correction term. Hence, the system is solvable (with normal discretisation of the conductive terms), but the density stepping is no longer directly related to the constraining equation. As a result, properties of monotonicity or TVD, intrinsic to the choice of the spatial discretisation, are no longer guaranteed (if the continuity equation is used to calculate the density field), or mass is no longer conserved (if the density equation is used).

8.5 Conclusions

In this chapter, we presented a solution for the odd-even decoupling of the pressure field in the framework of pressure-correction algorithms for variable density flow. It consists of two remedies. First, a correction term for the cell-face velocity, similar to the proposition in [62] is introduced (eq. (8.30)). Secondly, the wide stencil for the discrete Laplacian is compacted, involving only the immediate neighbours of the node and the node itself. It is

important that the correction term does not appear in the pressure Poisson-like equation (eq. (8.38)).

There is an additional advantage in the pressure equation (8.38) in terms of efficiency due to the absence of upwind density values. As a result, the Laplacian-like operator L needs to be evaluated only once each time step.

Finally, the method has been extended for three-dimensional curvilinear grids. The presented method will be used in the next chapter to simulate two-dimensional test cases on a collocated mesh.

Chapter 9

Two-Dimensional Test Cases

In chapter 7, the three different pressure-correction algorithms have been systematically compared for one-dimensional problems with different fluid properties. In this chapter, we show that the use of the algorithms is not limited to 1D cases, by means of two test cases.

In a first test case, the compatibility-constraint pressure-correction algorithm[1] is used to simulate the ideal gas flow inside a cavity. Because, in this case, the solvability condition has to be fulfilled (section 8.1.3), the use of the continuity-constraint pressure-correction is no option. Indeed: because the density field is determined in a non-mass conserving way, the mass unbalance in the cavity has to be adjusted using an overall rescaling of the density field. Such methods have been used with success [12], but are restricted to steady-state simulations.

In the second test case, the stability limits of the continuity-constraint pressure-correction algorithm are investigated. This is done by means of both inert and reacting two-dimensional mixing layers. It is confirmed that stability cannot be guaranteed if the appearing density ratios are too high. The use of the discrete compatibility-constraint pressure-correction is found to be stable in all cases, if the CFL time step limit is respected.

9.1 Thermally Driven Cavity

9.1.1 Problem Description

We consider the flow in a differentially heated square cavity in which a temperature difference is applied to two opposite vertical walls, while the other sides of the square are

[1] in its slightly modified formulation of chapter 8

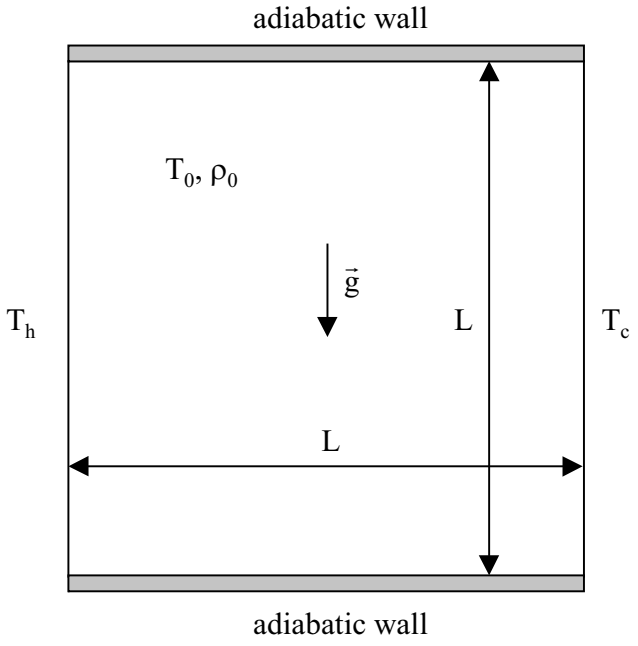

Figure 9.1: Geometry, initial and boundary conditions for the thermally driven cavity problem.

perfectly thermally insulated (Fig. 9.1). Furthermore, large temperature differences are considered. For cases involving natural convection, the reference velocity is chosen as $u_\infty = \lambda/\rho_\infty L c_p$ [12], so that $\mathrm{RePr} = 1$. Hence the non-dimensional momentum equation takes the form

$$\frac{\partial(\rho u_j)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_i} = -\frac{\partial p_2}{\partial x_j} + \mathrm{Pr}\frac{\partial \tau_{ij}}{\partial x_i} + \frac{\mathrm{RaPr}}{2\epsilon}\rho\delta_{j3},$$

with $\mathrm{Ra} = 2\epsilon \mathrm{Re}^2\mathrm{Pr}/\mathrm{Fr}^2$ and ϵ a non-dimensional temperature difference. Application to the test case of the thermally given cavity, yields the definition for the Rayleigh number:

$$\mathrm{Ra} = \frac{g\rho_\infty^2 (T_h - T_c) L^3 \mathrm{Pr}}{T_\infty \mu_\infty^2}, \tag{9.1}$$

where L is the characteristic dimension of the cavity, T_h and T_c respectively the hot and cold temperatures applied to the vertical walls, T_∞ a reference temperature equal to $(T_h + T_c)/2$, μ_∞ a reference viscosity coefficient and ρ_∞ a reference density, both corresponding to T_∞. The gravitational constant is set to $g = 9.81 m/s^2$. The temperature difference can be presented by a non-dimensional parameter $\epsilon = (T_h - T_c)/(T_h + T_c)$.

The heat transfer through the wall is represented by local Nusselt number:

$$\mathrm{Nu}\,(y) \;=\; \frac{L\lambda\,\frac{\partial T}{\partial x}\big|_{wall}}{\lambda_\infty(T_h - T_c)}, \tag{9.2}$$

and average Nusselt number:

$$\overline{\mathrm{Nu}} \;=\; \frac{1}{L}\int_{y=0}^{y=L} \mathrm{Nu}\,(y)\;\mathrm{d}y. \tag{9.3}$$

In the above expressions $\lambda_\infty = \lambda\,(T_\infty)$. $\lambda\,(T)$ is the heat conduction coefficient $\lambda\,(T) = \mu\,(T)\,c_p/\mathrm{Pr}$. In the test cases considered here, the Prandtl number is assumed to remain constant, equal to $\mathrm{Pr} = 0.71$, and the viscosity is given by Sutherland's law:

$$\mu\,(T)\,/\mu^* = (T/T^*)^{3/2}\,(T^* + S)\,/\,(T + S)$$

with $T^* = 273\,\mathrm{K}$, $S = 110.5\,\mathrm{K}$, $\mu^* = 1.68 \cdot 10^{-5}\,\mathrm{kg/m/s}$, $c_p = \gamma R/\,(\gamma - 1)$, $\gamma = 1.4$ and $R = 287.0\,\mathrm{J/kg/K}$. The influence of the temperature on c_p is neglected. The problem is completely defined by the Rayleigh number, the value of ϵ, a reference state (here $p_0^0 = 101325\,\mathrm{Pa}$, $T_\infty = 600\,\mathrm{K}$, $\rho_\infty = p_0^0/\,(RT_\infty)$), the previously mentioned fluid properties and the initial conditions.

Further details and benchmark studies can be found in [12, 21, 38, 60, 77, 78, 79].

9.1.2 Implementation Details

The general algorithm, as described in section 8.3, is implemented on a cell-vertex collocated grid. A third order Van Leer-κ method discretisation is used for the convective fluxes and a second order central discretisation for the diffusive fluxes and the pressure term. Because of the projection method, a special treatment of the boundaries is needed, and specific care must be taken of the temperature boundary condition at the isothermal walls. Since we are dealing with an enclosure, the total pressure p_0 is variable in time. In order to increase temporal accuracy, the pressure-correction algorithm is put in a multistage loop. Time stepping is done with the explicit form of a 4 stage Runge-Kutta scheme of a system $\frac{\partial \psi}{\partial t} = G(\psi)$:

$$\begin{aligned} \psi^{(0)} &= \psi^n \\ \psi^{(1)*} &= \psi^{(0)} + \alpha_1 \Delta t G\left(\psi^{(0)}\right) \end{aligned}$$

$$\begin{aligned}
\psi^{(2)*} &= \psi^{(0)} + \alpha_2 \Delta t G\left(\psi^{(1)}\right) \\
\psi^{(3)*} &= \psi^{(0)} + \alpha_3 \Delta t G\left(\psi^{(2)}\right) \\
\psi^{(4)*} &= \psi^{(0)} + \alpha_4 \Delta t G\left(\psi^{(3)}\right) \\
\psi^{n+1} &= \psi^{(4)}, \qquad (9.4)
\end{aligned}$$

with coefficients $\{\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1\}$. Between every stage, the velocity field is corrected in order to obey the velocity constraint.

9.1.2a Boundary Conditions

In general, zero flux boundary conditions are applied at walls: zero mass flux at all walls, and zero conductive flux at the adiabatic walls. Special care is needed for the boundary conditions of the predicted velocities at all walls and the temperature (read: density) at the isothermal walls. The correction term $\hat{\varepsilon}$ also has a special formulation at the boundaries.

Boundary conditions for $\boldsymbol{u}^*$ We distinguish between the normal and tangential components of the predicted velocity vector $\boldsymbol{u}^* = \boldsymbol{u}_n^* + \boldsymbol{u}_t^*$. Because of the viscous forces, the tangential component $\boldsymbol{u}_t^* = 0$. The normal component $\boldsymbol{u}_n^*$, however, cannot be set to zero at the wall, since the *blocking* effect of the wall on the flow field appears in the normal momentum equation under the form of a pressure force acting on the flow. Since in the operator splitting approach, the pressure terms are removed from the predictor momentum equations, the normal wall effect is removed as well, and no normal boundary conditions apply. For the corrected velocity, the complementary wall boundary conditions apply: the normal velocity component is set to zero, whereas the tangential component is left free.

Special treatment of the correction term at the boundaries We recall expression (8.32) for $\hat{\varepsilon}$ for a 1D flow and evaluate this at the boundary, for node $i = 0$, starting from (8.28):

$$\begin{aligned}
\rho_R^{n+1} \widetilde{u}_{\frac{1}{2}}^* - \Delta t \rho_R^{n+1} \left.\frac{\nabla p}{\widetilde{\rho}}\right|_{\frac{1}{2}}^{n+1} &= \rho_R^{n+1} \underline{u}_{\frac{1}{2}}^{n+1} \\
\overset{(8.31)\ \text{with}\ u_0 = 0}{\Leftrightarrow} \quad \frac{u_0^* + u_1^*}{2} - \Delta t \left.\frac{\nabla p}{\widetilde{\rho}}\right|_{\frac{1}{2}}^{n+1} &= \frac{1}{2} u_1^{n+1} + \hat{\varepsilon}_{\frac{1}{2}} \\
\overset{(8.23)}{\Leftrightarrow} \quad \frac{u_0^* + u_1^*}{2} - \Delta t \left.\frac{\nabla p}{\widetilde{\rho}}\right|_{\frac{1}{2}}^{n+1} &= \frac{1}{2} u_1^* - \Delta t \frac{1}{2} \left.\frac{\overline{\nabla p}}{\widetilde{\rho}}\right|_{1}^{n+1} + \hat{\varepsilon}_{\frac{1}{2}},
\end{aligned}$$

so that the correction term at the boundaries becomes:

$$\hat{\varepsilon}_{\frac{1}{2}} = -\Delta t \left[\left. \frac{\nabla p}{\widetilde{\rho}} \right|_{\frac{1}{2}}^{n+1} - \frac{1}{2} \left. \frac{\overline{\nabla p}}{\widetilde{\rho}} \right|_{1}^{n+1} \right] + \frac{u_0^*}{2}.$$

Boundary conditions at isothermal walls A peculiarity of the present pressure-correction formulation is that a density stepping is used instead of the commonly applied temperature stepping. As a consequence, the Dirichlet boundary condition for temperature cannot directly be enforced. This problem is solved in the following way. The requirement $\frac{\partial T}{\partial t} = 0$ gives

$$\begin{aligned} \frac{T^{n+1} - T^n}{\Delta t} &= 0 \\ \Leftrightarrow \frac{1}{\Delta t} \left(\frac{p_0^{n+1}}{\rho^{n+1}} - \frac{p_0^n}{\rho^n} \right) &= 0 \\ \Leftrightarrow \frac{p_0^n}{\rho^n \rho^{n+1}} \frac{\rho^n - \rho^{n+1}}{\Delta t} + \frac{1}{\rho^{n+1}} \frac{p_0^{n+1} - p_0^n}{\Delta t} &= 0. \end{aligned} \tag{9.5}$$

At the boundary, $\frac{\partial \rho}{\partial t}$ can be expressed by equation (2.39), with $u_i = 0$:

$$\begin{aligned} \frac{\rho^{n+1} - \rho^n}{\Delta t} &= -\frac{\rho_R^n u^n_{i+\frac{1}{2}}}{\Delta x} \quad \text{for } i = 0 \\ \frac{\rho^{n+1} - \rho^n}{\Delta t} &= \frac{\rho_L^n u^n_{i-\frac{1}{2}}}{\Delta x} \quad \text{for } i = N. \end{aligned} \tag{9.6}$$

The last expression can be used to formulate the constraint for the velocity field at these nodes. For an internal node, the constraint follows from elimination of $\frac{\partial \rho}{\partial t}$ from (2.39) and (8.1), yielding

$$\frac{\partial u_i^n}{\partial x_i} = -\frac{1}{\gamma p_0^n} \frac{dp_0}{dt} + \frac{\partial q_i^n}{\partial x_i}. \tag{9.7}$$

Inserting (9.6) into (9.5), yields the constraint at the isothermal boundary nodes (e.g. for node $i = 0$):

$$\frac{\partial u_i^n}{\partial x_i} = -\frac{\rho^n}{\rho_R^n} \frac{1}{p_0^n} \frac{dp_0}{dt}, \tag{9.8}$$

from which the pressure equation follows. There is, however, a small difficulty with the previous expression: since we do not know the direction of the velocity u^n, the exact extrapolations ρ_R^n and ρ_L^n cannot yet be determined. The solution is found in the following identity for enclosures:

$$\int_V \frac{\partial u_i^n}{\partial x_i} \mathrm{d}V \equiv \oint_{\partial V} \boldsymbol{u} \cdot \boldsymbol{n} \, \mathrm{d}S = 0.$$

In finite volume formulation, we then obtain:

$$\begin{aligned} \int_V \frac{\partial u_i^n}{\partial x_i} \mathrm{d}V &= \sum_{internal\ \ nodes} \frac{\partial u_k^n}{\partial x_k} V_i + \sum_{isothermal\ \ wall} \frac{\partial u_k^n}{\partial x_k} V_i \\ \Leftrightarrow 0 &= -\frac{1}{\gamma p_0^n} \frac{dp_0}{dt} \underbrace{\left[\sum_{int\ \ nds} V_i + \sum_{iso\ \ wall} \frac{\gamma \rho^n}{\rho_{R/L}^n} V_i \right]}_{\mathcal{V} \geq 0} + \sum_{int\ \ nds} \frac{\partial q_k^n}{\partial x_k} V_i, \end{aligned} \tag{9.9}$$

from which the sign of $\frac{dp_0}{dt}$ can be determined, and, because of (9.8), the sign of $\frac{\partial u_i^n}{\partial x_i}$, such that the exact extrapolation for ρ is known. Because of that, term $\mathcal{V}$ in expression (9.9) is known, and $\frac{dp_0}{dt} = \frac{p_0^{n+1} - p_0^n}{\Delta t}$ can be calculated.

The above 1D reasoning is easily extended to higher dimensions, if we assume that no transport occurs parallel to the wall at the isothermal boundary nodes.

9.1.2b Time Step Restriction

For a one dimensional problem, using a forward Euler scheme, the time step restriction is given by the semi-empirical stability condition (see appendix A):

$$\Delta t \leq \frac{1}{\frac{1}{(\Delta t)_c} + \frac{1}{(\Delta t)_d}}, \tag{9.10}$$

with

$$(\Delta t)_c \leq \frac{\Delta x}{\frac{1}{2}\left(\underline{u}_{i-\frac{1}{2}} + \underline{u}_{i+\frac{1}{2}} \right)},$$

and

$$(\Delta t)_d \leq \frac{2\Delta x^2}{(\rho_L + \rho_R)\left(\tilde{k}_{i-\frac{1}{2}} + \tilde{k}_{i+\frac{1}{2}}\right)},$$

with

$$\tilde{k}_{i+\frac{1}{2}} = \frac{\lambda_{i+\frac{1}{2}}}{\mathrm{RePr}\rho_i\rho_{i+1}}.$$

When $\mathrm{Pr} < 1$, the viscous time step limit is less restrictive than the conductive time step limit, so that stability for the momentum equation is also ensured.

9.1.3 Results

Ra	$\overline{\mathrm{Nu}}$	$\overline{p}/p_0^0$	$\overline{\mathrm{Nu}}$[77]	$\overline{p}/p_0^0$[77]
10^3	1.1061	0.9381	1.1077	0.93805
10^4	2.2115	0.9166	2.218	0.91463
10^5	4.4333	0.9293	4.480	0.92196
10^6	8.3747	0.9487	8.687	0.92449

Table 9.1: Nusselt number at the midplane and mean pressure for different Rayleigh numbers on a 64x64 mesh.

Cartesian		stretched	
$\overline{\mathrm{Nu}}_{cold}$	$\overline{p}/p_0^0$	$\overline{\mathrm{Nu}}_{cold}$	$\overline{p}/p_0^0$
1.1080	0.9383	1.0814	0.9406

Table 9.2: Nusselt number at the cold wall and mean pressure for $\mathrm{Ra} = 10^3$. Comparison between Cartesian and stretched grids of 32x32 nodes.

For a non-dimensional temperature difference $\epsilon = 0.6$, cases with four different Rayleigh numbers ($\mathrm{Ra} = 10^3$, 10^4, 10^5 and 10^6) are calculated. A uniform grid with square control volumes on a 64x64 mesh is used. According to the stability domain of the multistage algorithm (9.4), with coefficients $\{\frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1\}$, the time step is set to 1.5 times the maximum timestep as defined in eq. (9.10).

$\mathrm{Ra} = 10^3$ $\mathrm{Ra} = 10^4$

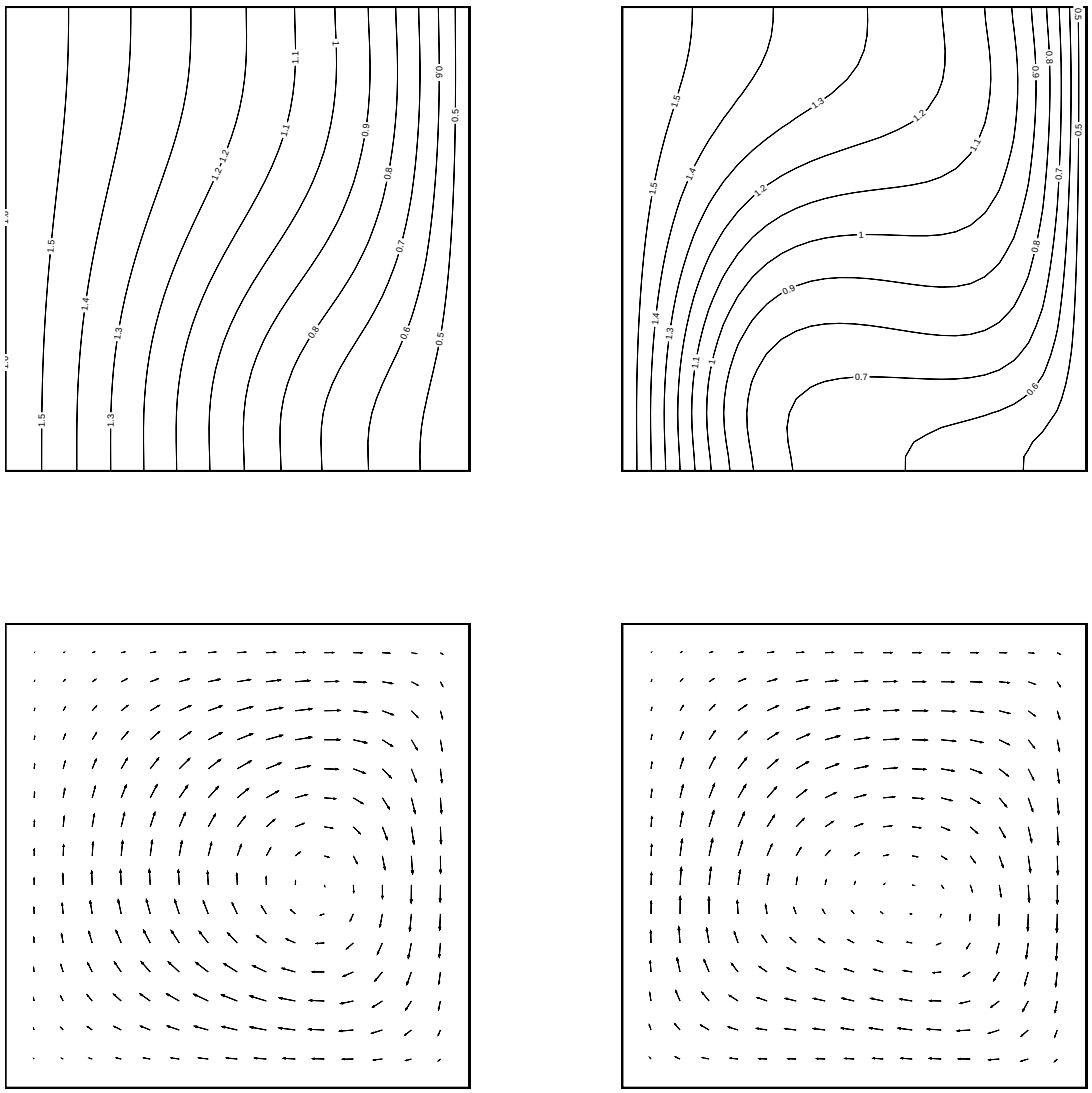

Figure 9.2: Isolines for the dimensionless temperature ϵ (top) and velocity vector-fields (bottom) for $\mathrm{Ra} = 10^3$(left) and 10^4(right). Isolines for the temperature range from $\epsilon = -0.6$ to $\epsilon = 0.6$ with intervals of 0.1. Velocity vectors are scaled with the maximum velocity in the domain.

The results are summarised in table 9.1. In low Mach number flows, the mean pressure is given by $\bar{p} = p_0$. The Nusselt number is evaluated at the midplane between the two isothermal walls. Comparison with the data obtained in [77] for the full Navier-Stokes equations shows good agreement for lower Rayleigh numbers. For the higher

$\mathrm{Ra} = 10^5$ $\mathrm{Ra} = 10^6$

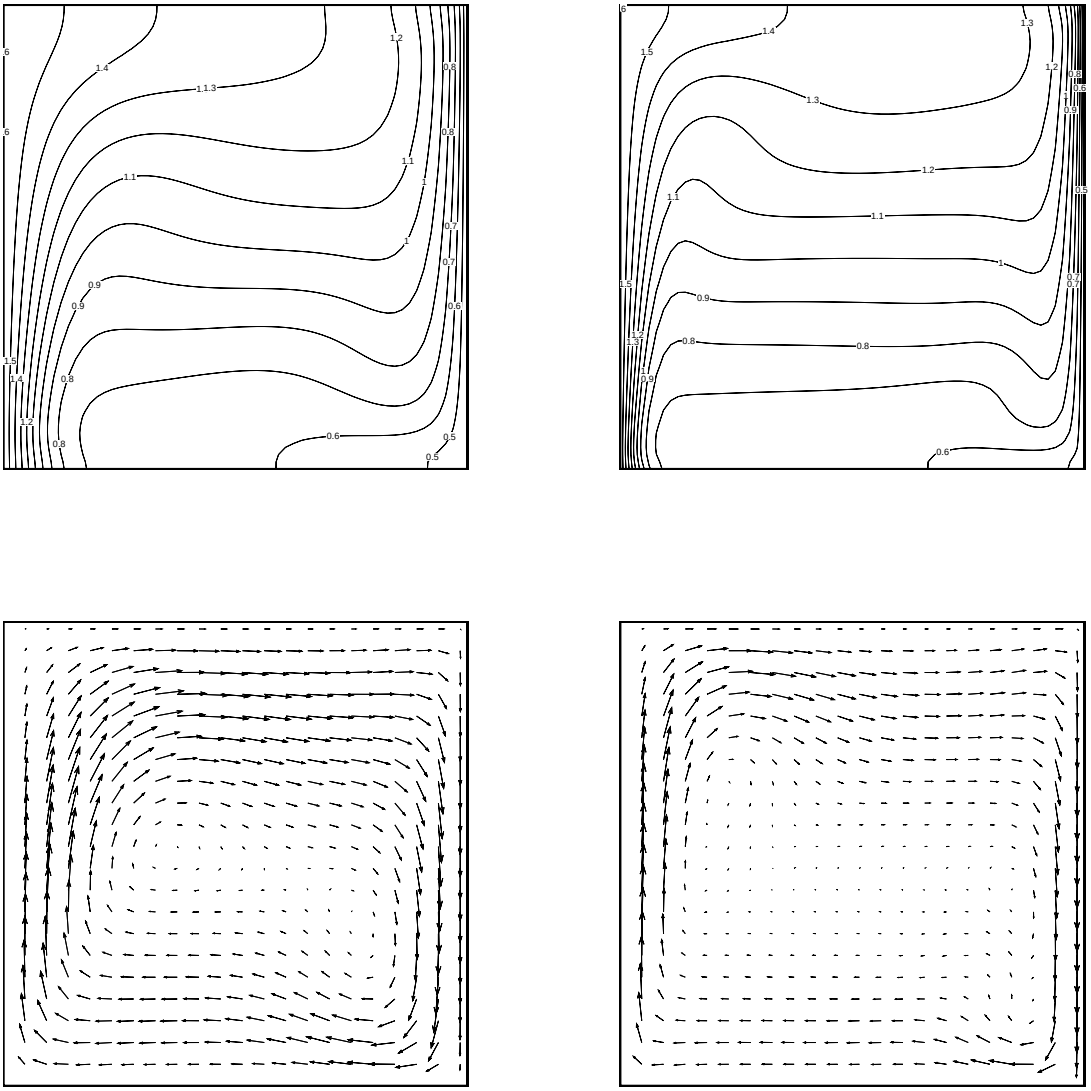

Figure 9.3: Isolines for the dimensionless temperature ϵ (top) and velocity vector-fields (bottom) for $\mathrm{Ra} = 10^5$(left) and 10^6(right). Isolines for the temperature range from $\epsilon = -0.6$ to $\epsilon = 0.6$ with intervals of 0.1. Velocity vectors are scaled with the maximum velocity in the domain.

Rayleigh numbers, there is a larger error, which is due to the relatively coarse grid used for these calculations (for comparison a calculation of $\mathrm{Ra} = 10^6$ on a 128x128 mesh yields $\overline{Nu} = 8.6880$ and $\overline{p}/p_0^0 = 0.9307$). Qualitatively, good results were obtained for all Rayleigh numbers, as can be seen in the isotemperature lines and velocity fields (figs.

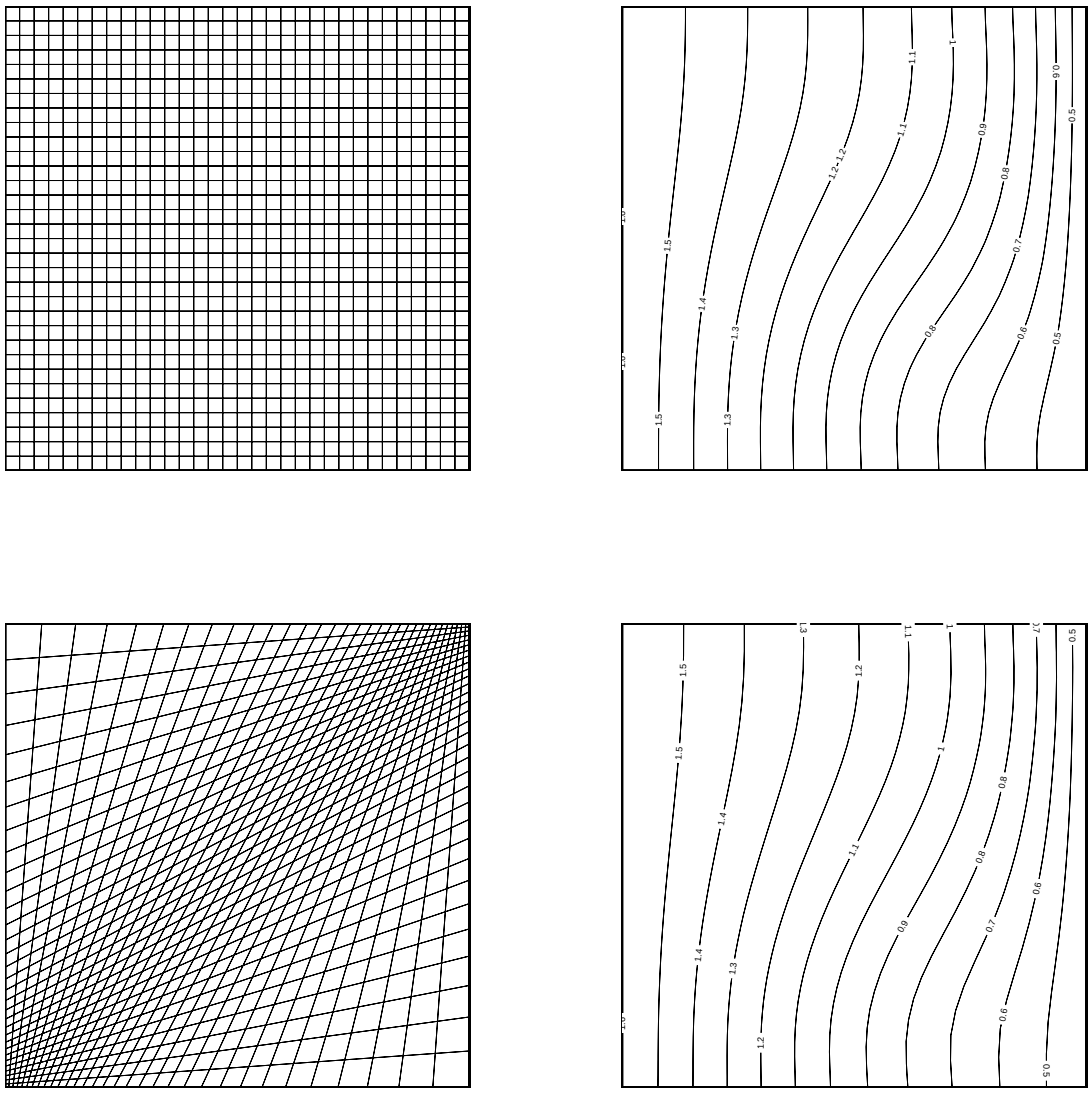

Figure 9.4: 32x32 stretched and skewed grid (bottom left), resulting in smooth isolines for the dimensionless temperature ϵ for $Ra = 10^3$(bottom right). Results for the Cartesian grid (top) are included for comparison.

9.2 and 9.3). The temperature and velocity fields are smooth, even when relative coarse meshes are used, where the odd-even decoupling is expected to be more pronounced [19].

The behaviour of the algorithm on stretched and skewed grids is checked as well. As an example, the resulting temperature field for $\mathrm{Ra} = 10^3$ on a 32x32 non uniform grid is shown in fig. 9.4, resulting in practically the same solution as on the Cartesian grid (table

9.2), without odd-even decoupling.

9.1.4 Conclusion

In this section, we used a two dimensional thermally driven cavity to validate the single fluid ideal gas algorithm, using the cure from chapter 8 to cope with odd-even decoupling. We explained in great detail the calculation of thermodynamic pressure in enclosures and the correct implementation of boundary conditions.

The obtained results are found to be comparable to previously performed calculations. Even for highly stretched and skewed meshes, good results are obtained, showing the large applicability of the presented cure for odd-even decoupling.

9.2 Mixing Layer

9.2.1 Problem Description

A second test case is a laminar mixing layer. Two different fluids enter separately at the same speed into a straight channel, where mixing takes place. Note that no experimental data are available for this test case. It is merely a hypothetical situation, allowing for code validation and performance monitoring. For the benefit of this thesis, this test case is used to compare different algorithms in a physically more realistic set up. Thus, it indicates the behaviour in realistic flow configurations.

For the mixing layer, a computational domain of $1\,m \times 0.5\,m$ has been used. The upper and lower boundary are walls. The two fluids enter at the left boundary and leave the domain on the right. The mixing layer is calculated on a 32x16 collocated vertex-centered mesh, using square control volumes. The entering fluids are fluid A and B in case of inert mixing, or fuel and oxidizer, representing non-premixed combustion (see fig. 9.5). The mixing process is described by the mixture fraction equation. The equation of state, imposing the density mixture-fraction relationship, is the previously introduced equation of state for inert mixing (fig. 3.4) or the simplified Burke-Schumann chemistry model (fig. 3.3) for non-premixed combustion.

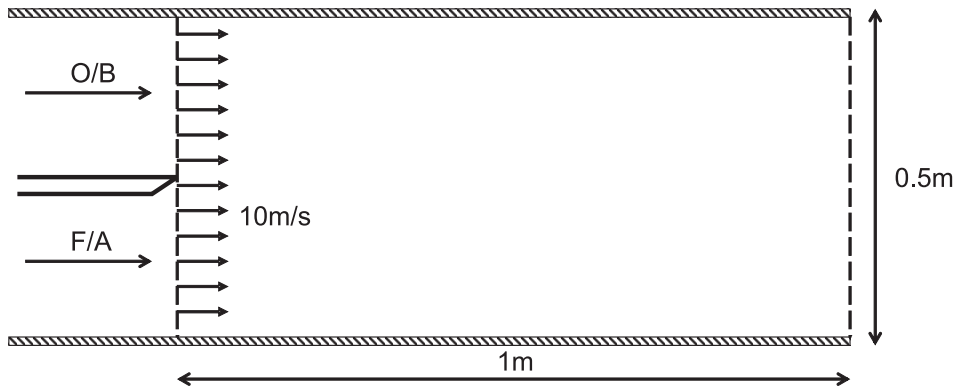

Figure 9.5: Set-up for the simulation of a laminar mixing layer, with mixing of two inert fluids (A and B) or with mixing and reaction of fuel and oxidizer.

The fluid's viscosity is set to 0, but species diffusion is admitted with a species diffusivity equal to $\rho D = 0.015625\,Pa.s$. Note that, because of the hypothetical nature of the test case, this value is arbitrary and is chosen in a way that a nicely diverging mixing layer is obtained in the domain. Due to the absence of viscosity, the upper and lower walls are treated as slip walls.

The test case is calculated in a time-accurate manner, using a constant time step Δt, depending on the specific test case, until a converged solution is obtained. As initial

condition, the computational domain is filled with pure fluid B or oxidizer. Initially, the flow velocity is set to $10\,m/s$ in the entire domain. All convective terms are discretised with a first order upwind scheme, whilst a second order central discretisation is applied for the diffusive and pressure terms. A first order explicit Euler time stepping procedure is used for the transient calculation.

9.2.2 Implementation Details

Besides the slip wall boundary conditions at the top and bottom boundary, imposing a zero normal velocity component, also in- and outlet boundary conditions are to be prescribed in this test case. At the outlet, all variables are evaluated from the interior of the domain, except for the pressure, whose value follows from a zero normal stress boundary condition. At the inlet, the incoming fluxes are prescribed, i.e. a uniform velocity of $10\,m/s$ and a mixture fraction $\xi = 1$ for $y_i < 0.25m$ and $\xi = 0$ for $y_i \geq 0.25m$, where y_i is the center of the control volume. Because of this piecewise constant prescription of the fluxes, the computational geometry is not symmetrical, but the scalar and density gradients do appear sharper. The density values are reported below.

In order to avoid the spurious mode, the correction term for the cell-face velocity (8.32) was also used in the present test case, when simulated using the compatibility-constraint pressure-correction algorithm. When the continuity-constraint pressure-correction algorithm is used, a similar correction is added to the mass flux at the cell faces [62].

9.2.3 Results

Different tests were performed. Both reacting and non-reacting mixing layers are calculated, using eq. (3.13) for the non-reacting case and eq. (3.9) for the reacting case. Furthermore, not only the newly developed discrete compatibility-constraint pressure-correction scheme was used (chapter 5), but also the analytical compatibility-constraint (section 6.2.2) and the continuity-constraint pressure-correction scheme (section 6.2.1). The emphasis in this section is not on an accurate prediction of the test case, but on the stability of the algorithms when applied in different situations. Therefore, different time-accurate calculations of the test case were performed with increasing time steps, kept constant during the simulation.

9.2.3a Inert Mixing Layer

A density ratio of 10 is applied at the inlet of the domain ($\rho_A = 1\,kg/m^3$; $\rho_B = 0.1\,kg/m^3$). The density ratio is maximal at the the inlet of the domain. The maxi-

continuity-constraint	compatibility-constraint
0.00014	0.0001

Table 9.3: Maximum allowable timestep (in seconds) for stability during the simulation of the 2D inert mixing layer. The different pressure-correction schemes are comparable with respect to stability.

mum time step that yields a stable simulation in the inert mixing case is reported in table 9.3. We conclude that all pressure-correction algorithms[2] yield a stability limit that is of the same order of magnitude. The reader should not be surprised by this observation. Also in the 1D tests (chapter 7) involving inert mixing, a relatively good prediction was obtained, especially in case of purely diffusive transport (fig. 7.20). In case of pure convection (fig. 7.5), the simulation results became unstable, but inaccuracies were restricted to the region close to sharp gradients. Including an amount of diffusion (here of the same order of magnitude, because of the choice of the species diffusivity), seems able to alleviate the convective inaccuracy.

From table 9.3, one must be careful not to draw the conclusion that the continuity-constraint pressure-correction scheme is more stable than the compatibility-constraint pressure-correction because the maximum allowable time step is bigger. Indeed, because a different algorithm is used, different results for e.g. velocity are obtained during timestepping. Since the magnitude of velocity has a direct impact on the maximum time step for stability through the CFL-number (eq. 4.3), the reported maximum time step is also influenced by this.

The converged solution is independent of the pressure-correction scheme and is depicted in fig. 9.6.

9.2.3b Reacting Mixing Layer

Since the densities of pure fuel and pure oxidizer are assumed equal ($\rho_F = \rho_O \approx 1.25$), there is no density ratio between the incoming fluxes at the inlet. Due to chemical reaction, large density ratios appear inside the domain. It was found that no stable results could be obtained for the reacting test case, using the continuity-constrained scheme, unless the density time derivative in the continuity equation is rescaled by a factor $\alpha < 1$:

$$\frac{\partial \rho u'_i}{\partial x_i} = -\frac{\partial \rho u_i^*}{\partial x_i} - \alpha \frac{\partial \rho}{\partial t}$$

[2]Note that in case of inert mixing, the discrete and analytical compatibility-constraint pressure-correction schemes do not differ.

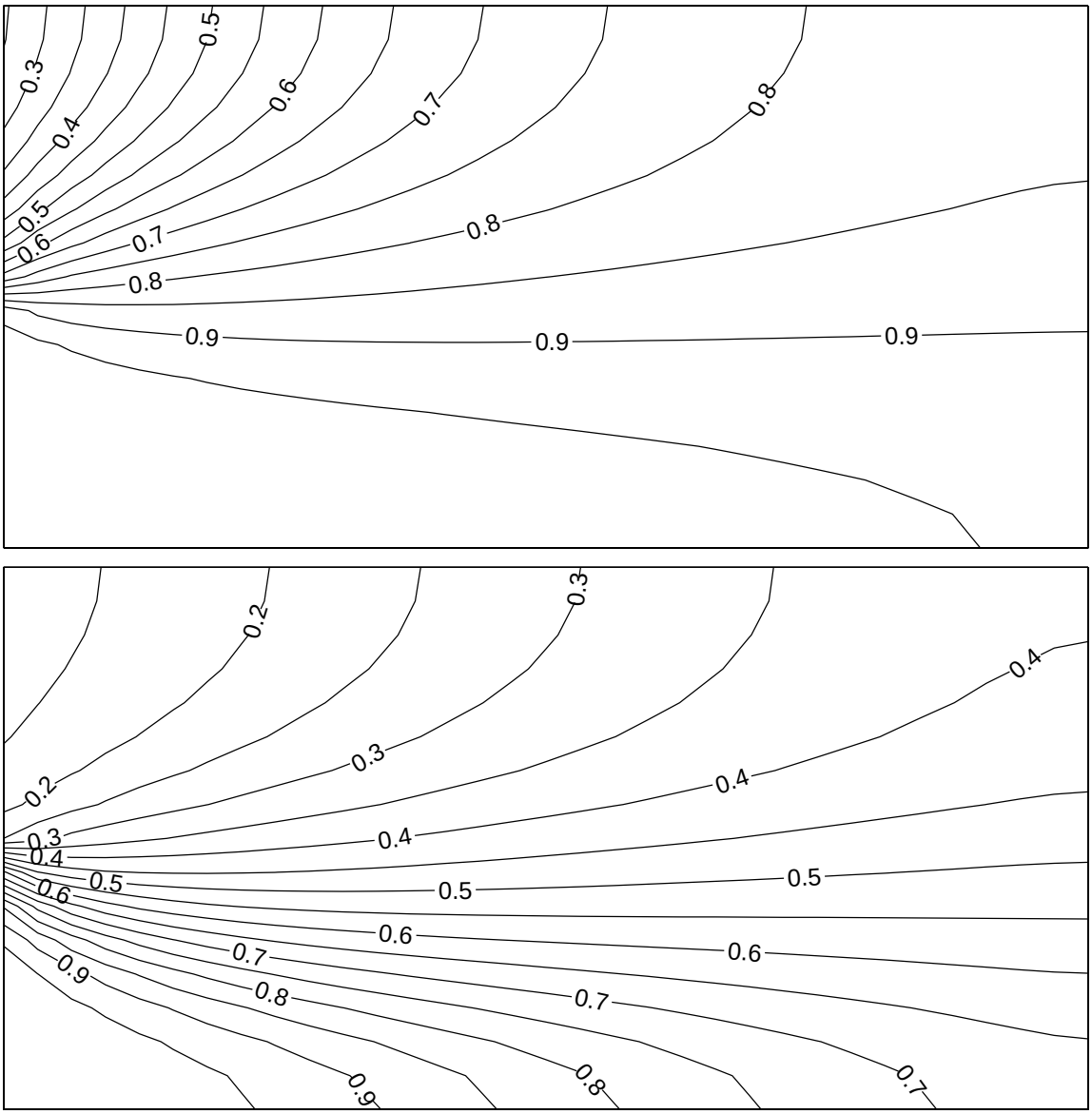

Figure 9.6: Converged solution of the 2D inert mixing layer: contourlines of mixture fraction (top) and density (bottom).

The maximum timestep is given as a function of α in table 9.4. The maximum time step in reacting flows is smaller, compared to the non-reacting case, due to the higher velocity magnitudes appearing due to flow acceleration. For the continuity-constraint scheme to be stable in reacting flows, not only the use of a rescaling factor is neccesary, but also a remarkably smaller time step has to be used, compared to the discrete compatibility-constraint scheme.

We observed that for the time steps, listed in table 9.4, the same converged solution was

continuity-constr.		anal. compat.-constr.		discr. compat.-constr.
Δt	α	Δt	ζ	Δt
-	1	0.00016	0	0.00022
0.00004	0.1	0.00017	0.1	
0.00006	0.01			

Table 9.4: Maximum allowable timestep (in seconds) to obtain a stable steady-state solution during the simulation of the 2D reacting mixing layer. The continuity-constraint pressure-correction scheme does not yield stable results, unless a rescaling of the density time derivative is used with rescaling factor α. The drift from the equation of state is controlled in the analytical continuity-constraint scheme, using a damping factor ζ.

obtained with the continuity-constraint pressure-correction scheme and the discrete compatibility-constraint pressure-correction scheme. A contourplot of the converged solution is depicted in fig. 9.7.

When the timestep is sufficiently low, using the continuity-constraint pressure-correction scheme with e.g. $\alpha = 0$, the simulation remains stable, but does not converge to a steady-state solution. An oscillating behaviour was noticed, especially at the stoichiometric line ($\xi = 0.1$). As an example for this observation, two subsequent time levels of the 'converged' solution are plotted on the same graph in fig. 9.8 for two values of the time step. To our experience, the higher the timestep, the more pronounced the oscillations, until the solution diverges for a time step that is sufficiently large. Note that, in LES calulations, it would be very hard to identify such oscillations. It remains to be investigated whether this would affect the results of such simulations substantially.

It is remarkable that a different steady-state is obtained when the analytical compatibility-constraint pressure-correction scheme is used. Converged solutions with the maximum time step are shown in fig. 9.9. The fact that a different solution is obtained with this algorithm, stems from the uncontrolled drift from the equation of state. The equation of state is indeed not imposed in a hard manner. On the contrary, the equation of state is imposed in a weak sense, through the material derivative. This explains the different solution, as can also be concluded from the scatter plot of the converged states (fig 9.10).

The drift from the equation of state can be controlled, using a defect correction term. Here, the value for the damping factor was taken as $\zeta = 0.1$. This results yet in a different converged solution, as depicted in fig. 9.11 (which looks like the contour plot of fig. 9.7, but is not identical), since the obtained states are closer to the equation of state (fig. 9.12). Still, in some nodes, large errors remain. Also note that, in practice, it is not straightforward to choose ζ.

Using the discrete compatibility-constraint pressure-correction-scheme yields, by definition, density and fuel elements mass fields that correspond exactly to the equation of

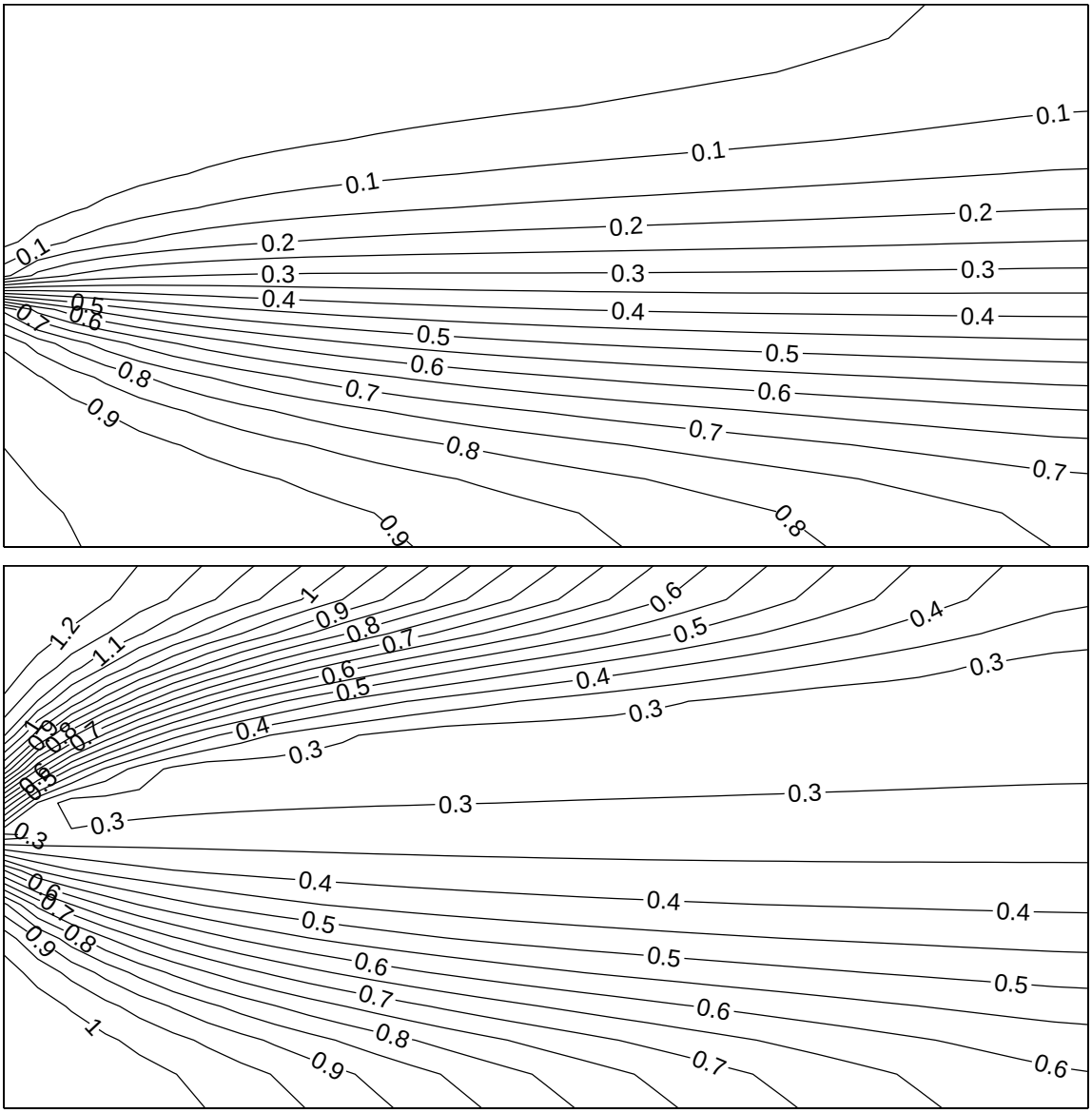

Figure 9.7: Converged solution of the 2D reacting mixing layer, obtained with the continuity-constraint and the discrete compatibility-constraint pressure-correction scheme: contourlines of mixture fraction (top) and density (bottom).

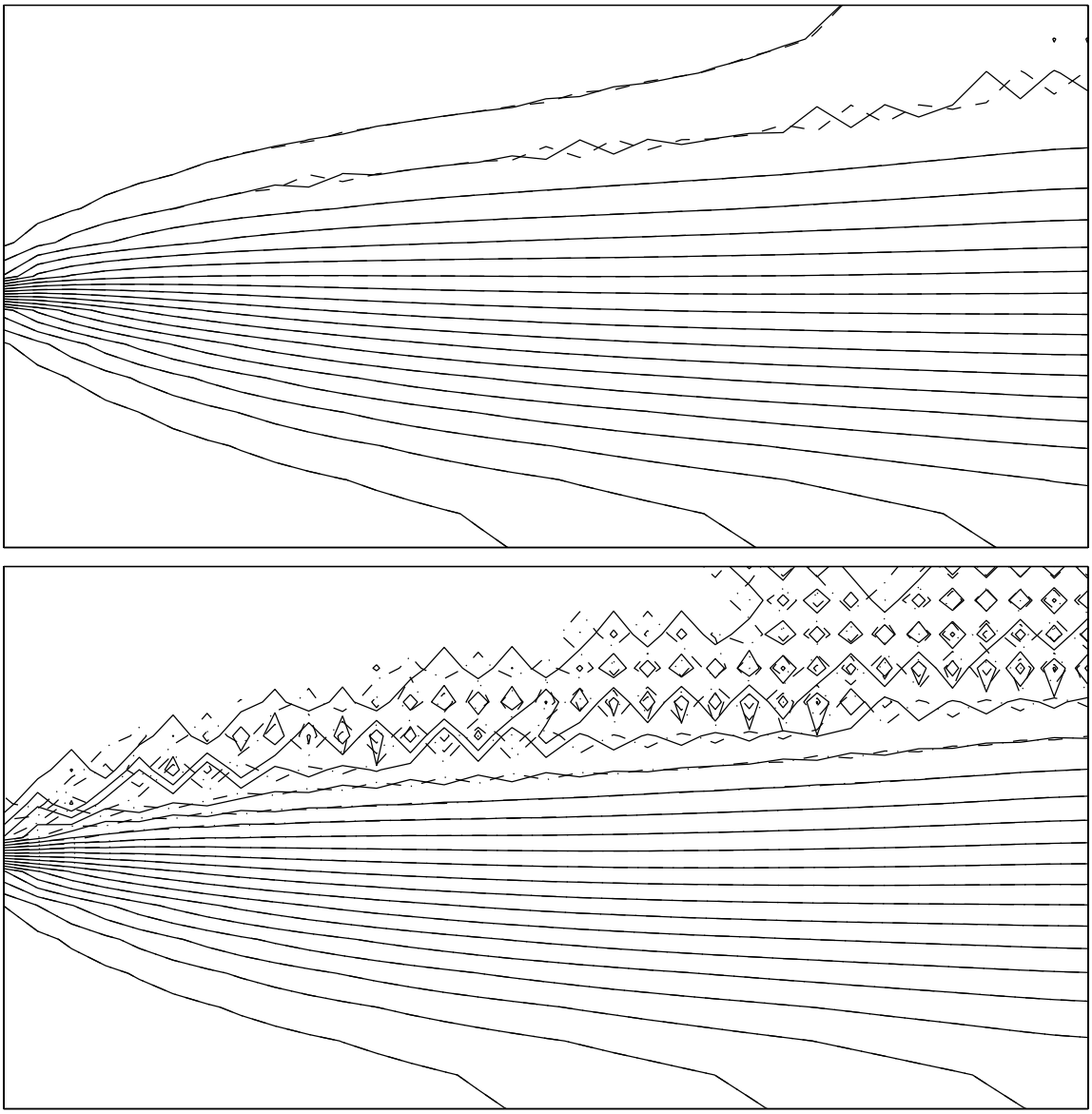

Figure 9.8: Oscillating 'converged' solution of the 2D reacting mixing layer, obtained with the continuity-constraint pressure-correction scheme with rescaling factor $\alpha = 0$, using a time step $\Delta t = 0.00008\,s$ (top) and $\Delta t = 0.0002\,s$ (bottom): contourlines of mixture fraction at two subsequent time levels.

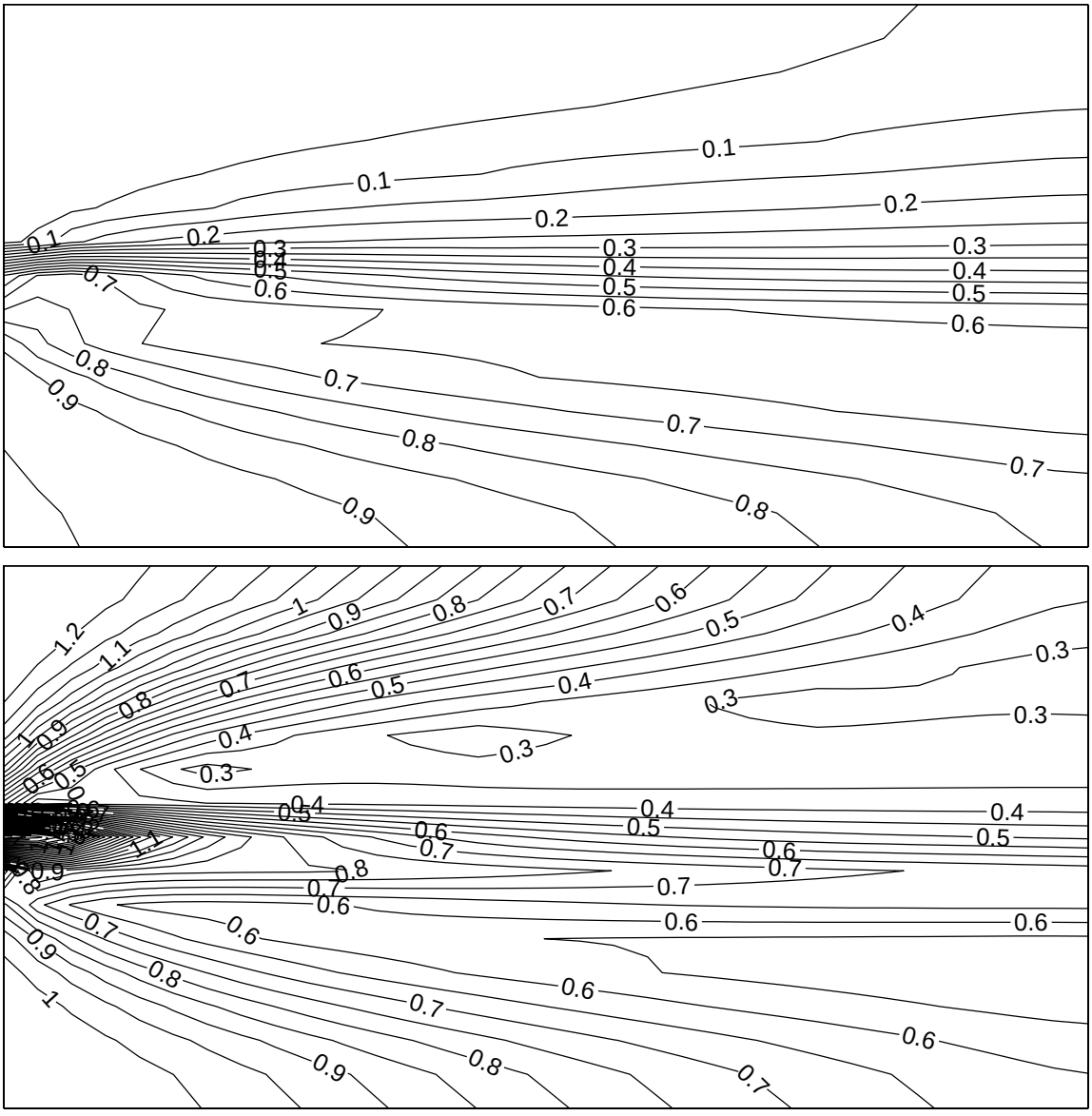

Figure 9.9: Converged solution of the 2D reacting mixing layer, obtained with the analytical compatibility-constraint pressure-correction scheme: contourlines of mixture fraction (top) and density (bottom).

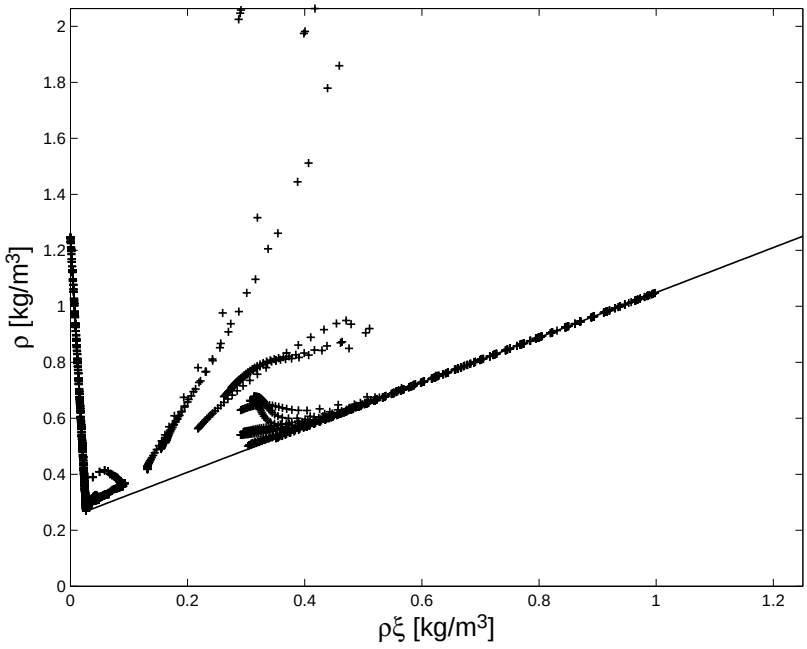

Figure 9.10: Scatter plot of the obtained density and fuel elements mass predictions in the converged solution of the 2D reacting mixing layer, obtained with the analytical compatibility-constraint pressure-correction scheme: even in the converged solution, the equation of state is not fulfilled.

state (fig. 9.13). There is, as mentioned earlier, a price to pay: the equation for pressure is a non-linear equation and will require several re-linearisations per time step, inducing an extra computational cost. From the 1D test cases, we concluded that at most 2 linearisations were needed per time step. In two dimensions, this statement no longer holds. However, if we monitor the actual re-linearisations needed during the time stepping of this test case (table 9.5), we can conclude that the extra cost involved, remains very low for more general 2D test cases. It was found that mostly not even one single re-linearisation is needed and that the maximum number of re-linearisations (6) was only encountered in two time steps out of thousand. The extra cost is thus marginal compared to the benefits associated with this algorithm: the higher time step limit and the greater accuracy.

9.2.4 Conclusion

In this section, the stability and convergence of the discrete-compatibility pressure-correction method in two dimensions were demonstrated by means of a reacting and a non-reacting mixing layer. Again, the two other pressure-correction methods were used for comparison.

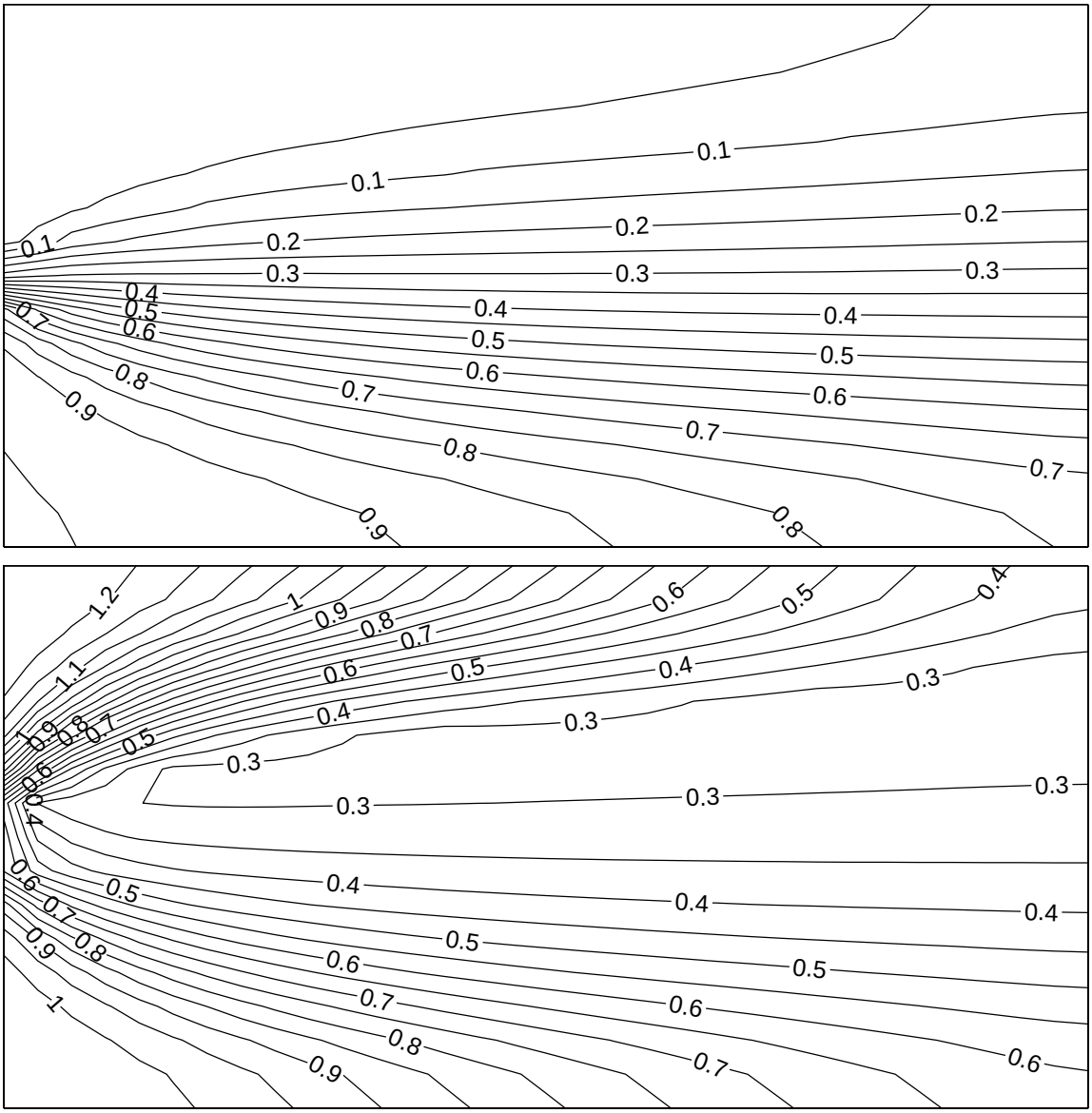

Figure 9.11: Converged solution of the 2D reacting mixing layer, obtained with the analytical compatibility-constraint pressure-correction scheme with defect correction ($\zeta = 0.1$): contourlines of mixture fraction (top) and density (bottom).

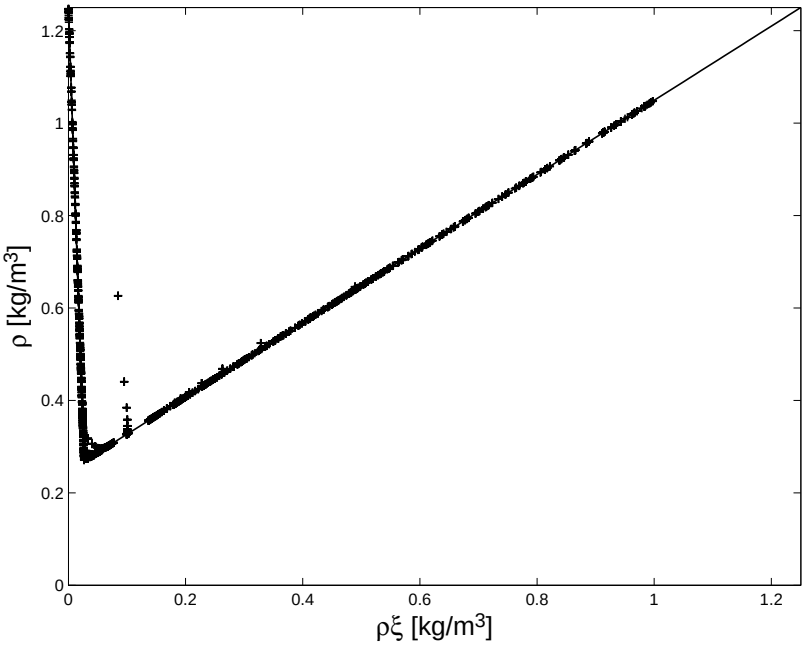

Figure 9.12: Scatter plot of the obtained density and fuel elements mass predictions in the converged solution of the 2D reacting mixing layer, obtained with the analytical compatibility-constraint pressure-correction scheme with defect correction ($\zeta = 0.1$): the equation of state is better predicted, but in some places, large errors remain.

No substantial differences appear between the different pressure-correction schemes in case of inert mixing, both in terms of stability and accuracy.

In combusting flows, however, the continuity-constraint pressure-correction algorithm could not be stabilised, unless measures were taken that corrupt time-accuracy. The analytical compatibility-constraint pressure-correction algorithm was stable, but predicted a converged solution that differs noticeably from the exact result. The reason for this stems from the uncontrollable drift from the non-linear equation of state. On the other hand, the here presented algorithm was able to yield a stable result and to predict states that match exactly according to the equation of state. The benefits of higher robustness and greater accuracy were acquired with only a minimal additional computational effort, as compared to the other algorithms, so that the algorithm is not only stable and accurate, but also efficient.

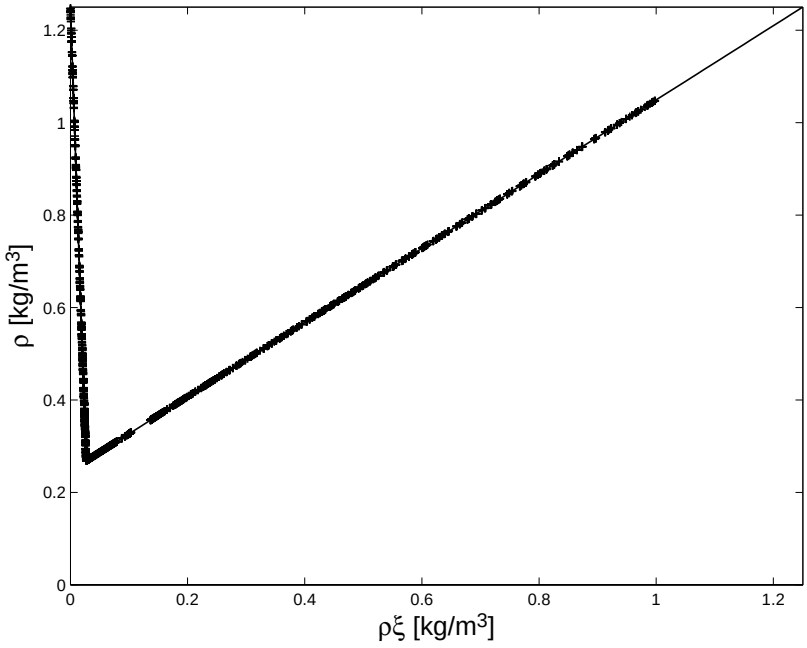

Figure 9.13: Scatter plot of the obtained density and fuel elements mass predictions in the converged solution of the 2D reacting mixing layer, obtained with the discrete compatibility-constraint pressure-correction scheme: the equation of state is exactly predicted.

number of re-linearisations	occurences	frequency(%)
6	2	0.2
5	5	0.5
4	5	0.5
3	5	0.5
2	27	2.7
1	123	12.3
0	833	83.3

Table 9.5: Number of re-linearisations needed during the first 1000 iterations of the discrete compatibility-constraint pressure-correction scheme with $\Delta t = 0.0002\,s$: in most cases, not a single re-linearisation is needed.

Chapter 10

Conclusions

In this thesis, I developed the discrete compatibility-constraint pressure-correction algorithm for transient simulations of general fluids at low-Mach numbers on collocated grids. The algorithm is new, because the constraint for the velocity field is constructed from a combination of the discrete equations of continuity and scalar transport, imposing that the newly predicted state must be compatible, in agreement with the equation of state. This way, mass and scalar conservation are guaranteed and the equation of state is exactly fulfilled at every time step.

Furthermore, a specially developed cell-face velocity interpolation formula suppresses the spurious pressure mode, still guaranteeing the solvability of the elliptic pressure equation in case of enclosed flow simulation.

I thoroughly described the algorithm for three example fluids: a single-fluid ideal gas, a two-fluid inert mixture and a two-fluid infinitely fast combusting mixture. The latter is special in the sense that the equation of state is highly non-linear and non-differentiable.

For comparison, two standard pressure-correction algorithms were used. A first algorithm, denoted as continuity-constraint pressure-correction, is based on a constraint for the velocity field that is derived solely from the continuity equation. The second algorithm, denoted as analytical compatibility-constraint pressure-correction, constructs the constraint from an analytical combination of the material derivative of the equation of state and the continuity and scalar equations.

The three algorithms were compared on several one and two-dimensional test cases, involving sharp density gradients. From these test cases, I conclude that the continuity-constraint pressure-correction algorithm must not be applied when sharp density gradients appear, because of its lack of stability. If the test case involves a considerable amount of diffusion and if the equation of state is linear, a stable result might be obtained. However, in case of reacting flows, with a non-linear equation of state, the simulation cannot be

stabilised. Possible measures to cure the stability corrupt time-accuracy, which jeopardises their use in transient simulations. The analytical compatibility-constraint pressure-correction algorithm performs better with respect to stability. In case of combusting flow simulation, however, inaccurate solutions are obtained and states are predicted that deviate strongly from the non-linear equation of state.

The newly developed algorithm performs well on all test cases. Both stable and reliable results are obtained, that exactly match the equation of state. Furthermore, the benefits of higher robustness and greater accuracy are acquired with only a minimal additional computational effort. Indeed, most of the times, not even one single re-linearization is needed to solve the non-linear pressure equation.

The general conclusion of this work is that, because of the non-linearity of e.g. a combustion process, standard algorithms are bound to fail and novel algorithms are to be adopted. I developed the discrete compatibility-constraint pressure-correction algorithm to provide reliable predictions in those cases where standard algorithms fail. The algorithm's properties are:

i. it is stable and robust;

ii. mass and scalars, such as energy and fuel elements mass are conserved;

iii. the predicted states exactly match the equation of state;

iv. the additional computational effort is minimal;

v. it allows time-accurate solutions.

Chapter 11

Future Work

In this work, I spent a lot of effort on the development of a new algorithm. I showed that it has great potential for flows with a highly non-linear equation of state (general fluid algorithm).

A first route of future work lies in the application to flows with combustion and fire. To that purpose, the discrete compatibility-constraint based house code will serve as a basis for future research in the research group. So far, simulations were only performed on systems involving one scalar equation. The algorithm, however, is not restricted to such cases. In the future, more complicated cases will be investigated. The flames from the TNF-workshop [1] will serve as primary target cases. Several of these turbulent (swirling) flames require LES to predict good solutions, so that the time-accuracy of the algorithm is important. Even if we apply the simplest flamelet model, in turbulent flows, the density is a function of mixture fraction and mixture fraction variance, whose value follows from a transport equation. There are thus two scalar variables that determine the density. If more complicated chemistry effects, such as local extinction, are important, the use of mixture fraction alone as a coordinate in chemical space is not sufficient. More progress variables are needed to describe the chemistry, each with an associated transport equation. Regardless of the amount of scalars that determine the density field, the algorithm remains applicable. There are however still issues of convergence and computing time that require further investigation.

A second topic of investigation is the extension of the algorithm towards flows at all speeds. Without going into details, this extension can serve valuable for situations where a broad range of Mach numbers appears in the flow field. Since such an extension has already been performed earlier for ideal gases, using other pressure-correction algorithms as a starting point [15, 45, 71], a combination of these methods with the present algorithm should yield an algorithm for general fluids at all speeds.

Appendix A

Linear Stability Analysis

A.1 Introduction

A stability analysis is normally done by a transformation to Fourier space. The set of equations is linearised around an equilibrium solution, and a von Neumann stability analysis is performed. For a standard convection-diffusion equation

$$\frac{\partial \psi}{\partial t} = -u\frac{\partial \psi}{\partial x} + \alpha \frac{\partial^2 \psi}{\partial x^2}, \tag{A.1}$$

the variable ψ is expanded around its equilibrium value:

$$\psi \;\; = \;\; \psi_{eq} + \delta\psi. \tag{A.2}$$

The error $\delta\psi$ is then decomposed into a Fourier series with harmonic components of the form

$$\delta\psi_i = \widetilde{\psi} e^{jik\pi/N}, \quad k = -N..N,$$

with $j = \sqrt{-1}$. Normally this is represented as a function of the phase angle $\phi = k\pi/N$, giving

$$\delta\psi_i = \widetilde{\psi} e^{ji\phi}, \quad \phi = -\pi..\pi.$$

The convection-diffusion equation can be discretised centrally for the diffusive part and with a first order upwind scheme for the convective part:

$$\psi_i^{n+1} = \psi_i^n - \frac{u\Delta t}{\Delta x}\left(\psi_i^n - \psi_{i-1}^n\right) + \frac{\alpha\Delta t}{\Delta x^2}\left(\psi_{i+1}^n - 2\psi_i^n + \psi_{i-1}^n\right).$$

Making use of (A.2), we get

$$\widetilde{\psi}^{n+1} = \widetilde{\psi}^n\left[1 - u\frac{\Delta t}{\Delta x}\left(1 - e^{-j\phi}\right) + \frac{\alpha\Delta t}{\Delta x^2}\left(e^{j\phi} - 2 + e^{-j\phi}\right)\right].$$

The stability region can be examined by requiring that the amplification factor G is within the unity circle.

$$G = 1 - u\frac{\Delta t}{\Delta x}\left(1 - e^{-j\phi}\right) + \frac{\alpha\Delta t}{\Delta x^2}\left(e^{j\phi} - 2 + e^{-j\phi}\right).$$

Requiring $||G|| \leq 1$ gives the following stability condition:

$$\Delta t \leq \frac{1}{\frac{u}{\Delta x} + 2\frac{\alpha}{\Delta x^2}}. \tag{A.3}$$

A.2 Operator Splitting

The convection-diffusion equation (A.1) can now be split in two parts:

$$\frac{\partial\psi}{\partial t} = \left(\frac{\partial\psi}{\partial t}\right)_c + \left(\frac{\partial\psi}{\partial t}\right)_d,$$

with

$$\left(\frac{\partial\psi}{\partial t}\right)_c = -u\frac{\partial\psi}{\partial x} \tag{A.4}$$

$$\left(\frac{\partial\psi}{\partial t}\right)_d = \alpha\frac{\partial^2\psi}{\partial x^2}. \tag{A.5}$$

With the same discretisations as before, the stability conditions for (A.4) and (A.5) are respectively:

$$(\Delta t)_c \leq \frac{\Delta x}{u};$$
$$(\Delta t)_d \leq \frac{\Delta x^2}{2\alpha}.$$

Because of linearity, the characteristic velocity of the entire system (A.1) is given by summation of the characteristic velocities of the two subsystems (A.4) and (A.5). Since the characteristic velocity is inversely proportional to the maximum time step, the global time step restriction follows from

$$\frac{1}{\Delta t} = \frac{1}{(\Delta t)_c} + \frac{1}{(\Delta t)_d} \tag{A.6}$$

A.3 Stability Criterion for Conservative Transport Equation

The previous reasoning can be used to build up a stability condition for a transport equation in conservative form. As an example we consider the continuity equation

$$\frac{\partial \rho}{\partial t} = -\frac{\partial (\rho u)}{\partial x}. \tag{A.7}$$

This equation is discretised as

$$\begin{aligned}\frac{\rho_i^{n+1} - \rho_i^n}{\Delta t} &= -\frac{\rho_R u_{i+\frac{1}{2}} - \rho_L u_{i-\frac{1}{2}}}{\Delta x} \\ &= -\frac{1}{2\Delta x}\left(u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}}\right)(\rho_R - \rho_L) \\ &\quad -\frac{1}{2\Delta x}(\rho_L + \rho_R)\left(u_{i+\frac{1}{2}} - u_{i-\frac{1}{2}}\right).\end{aligned} \tag{A.8}$$

The operator splitting technique can be used to obtain the convective and diffusive part of the equation

$$\begin{aligned}\left(\frac{\partial \rho}{\partial t}\right)_c &= -\frac{1}{2\Delta x}\left(u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}}\right)(\rho_R - \rho_L) \\ \left(\frac{\partial \rho}{\partial t}\right)_d &= -\frac{1}{2\Delta x}(\rho_L + \rho_R)\left(u_{i+\frac{1}{2}} - u_{i-\frac{1}{2}}\right).\end{aligned} \tag{A.9}$$

The convective time step restriction follows immediately:

$$(\Delta t)_c \leq \frac{\Delta x}{\frac{1}{2}\left(u_{i-\frac{1}{2}} + u_{i+\frac{1}{2}}\right)}, \tag{A.10}$$

whereas the diffusive time step restriction follows from the reasoning below: expression (A.9) is further elaborated, making use of the constraining equation for the velocity:

$$\frac{\partial u}{\partial x} = \frac{\partial}{\partial x}\left(k\frac{\partial}{\partial x}\left(\frac{1}{\rho}\right)\right),$$

with $k = \lambda/\mathrm{RePr}$ the heat conduction coefficient, which is discretised as

$$\begin{aligned}\Delta x\left(u_{i+\frac{1}{2}} - u_{i-\frac{1}{2}}\right) &= k_{i-\frac{1}{2}}\frac{1}{\rho_{i-1}} - \left(k_{i-\frac{1}{2}} + k_{i+\frac{1}{2}}\right)\frac{1}{\rho_i} + k_{i+\frac{1}{2}}\frac{1}{\rho_{i+1}} \\ &= -\frac{k_{i-\frac{1}{2}}}{\rho_{i-1}\rho_i}\rho_{i-1} + \left(\frac{k_{i-\frac{1}{2}}}{\rho_{i-1}\rho_i} + \frac{k_{i+\frac{1}{2}}}{\rho_i\rho_{i+1}}\right)\rho_i - \frac{k_{i+\frac{1}{2}}}{\rho_i\rho_{i+1}}\rho_{i+1}.\end{aligned} \tag{A.11}$$

If we define a mass diffusion coefficient $\tilde{k}_{i+\frac{1}{2}} = \frac{k_{i+\frac{1}{2}}}{\rho_i\rho_{i+1}}$, we can introduce (A.11) in (A.9):

$$\begin{aligned}\left(\frac{\partial\rho}{\partial t}\right)_d &= \frac{(\rho_L + \rho_R)}{2\Delta x^2}\left(\tilde{k}_{i-\frac{1}{2}}\rho_{i-1} - \left(\tilde{k}_{i-\frac{1}{2}} + \tilde{k}_{i+\frac{1}{2}}\right)\rho_i + \tilde{k}_{i+\frac{1}{2}}\rho_{i+1}\right) \\ \Leftrightarrow \rho_i^{n+1} &= \rho_i + \frac{\Delta t\,(\rho_L + \rho_R)}{2\Delta x^2}\left(\tilde{k}_{i-\frac{1}{2}}\rho_{i-1} - \left(\tilde{k}_{i-\frac{1}{2}} + \tilde{k}_{i+\frac{1}{2}}\right)\rho_i + \tilde{k}_{i+\frac{1}{2}}\rho_{i+1}\right).\end{aligned} \tag{A.12}$$

For stability, we require all coefficients to be positive, in particular the coefficient proceeding ρ_i^n:

$$1 - \frac{\Delta t}{2\Delta x^2}(\rho_L + \rho_R)\left(\tilde{k}_{i-\frac{1}{2}} + \tilde{k}_{i+\frac{1}{2}}\right) \geq 0, \tag{A.13}$$

so that the time step restriction becomes:

$$(\Delta t)_d \leq \frac{2\Delta x^2}{(\rho_L + \rho_R)\left(\tilde{k}_{i-\frac{1}{2}} + \tilde{k}_{i+\frac{1}{2}}\right)}. \tag{A.14}$$

Appendix B

Multistage Pressure Correction for Reacting Flows

B.1 Single Stage Pressure Correction Algorithm

The following equations govern the system:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 \tag{B.1}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} + \frac{\partial p}{\partial x_i} = \frac{\partial \tau_{ij}}{\partial x_j} \tag{B.2}$$

$$\frac{\partial \rho \xi}{\partial t} + \frac{\partial \rho \xi u_i}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\rho D \frac{\partial \xi}{\partial x_i}\right). \tag{B.3}$$

They express the conservation of mass, momentum and fuel mass. A temporal discretisation is given under the form:

$$\frac{\rho^{n+1} - \rho^n}{\Delta t} = -\frac{\partial \rho u_i^n}{\partial x_i} \tag{B.4}$$

$$\frac{\rho u_i^{n+1} - \rho u_i^n}{\Delta t} = -\frac{\partial \rho u_i u_j^n}{\partial x_j} - \frac{\partial p^n}{\partial x_i} + \frac{\partial \tau_{ij}^n}{\partial x_j} \tag{B.5}$$

$$\frac{\rho \xi^{n+1} - \rho \xi^n}{\Delta t} = -\frac{\partial \rho \xi u_i^n}{\partial x_i} + \frac{\partial}{\partial x_i}\left(\rho D \frac{\partial \xi^n}{\partial x_i}\right). \tag{B.6}$$

The algorithm for a simple forward Euler scheme in time is then:

1. determination of the fuel mass $f^{n+1} = \rho\xi^{n+1}$:

$$f^{n+1} = f^n + \Delta t_n^{n+1}\left[-\frac{\partial (fu_i)^n}{\partial x_i} + \frac{\partial}{\partial x_i}\left(\rho D \frac{\partial \xi^n}{\partial x_i}\right)\right]; \tag{B.7}$$

2. prediction of the velocity field $u_i^{*,n+1}$:

$$(\rho u_i)^{*,n+1} = (\rho u_i)^n + \Delta t_n^{n+1}\left[-\frac{\partial \rho u_i u_j^n}{\partial x_j} + \frac{\partial \tau_{ij}^n}{\partial x_j}\right]; \tag{B.8}$$

3. pressure correction:

$$\begin{aligned}
\rho^{n+2} &= \mathcal{H}_C\left(f^{n+2}\right) \\
\Leftrightarrow \rho^{n+1} - \Delta t_{n+1}^{n+2}\left[\frac{\partial \rho u_i^{n+1}}{\partial x_i}\right] &= \\
\mathcal{H}_C\left\{f^{n+1} + \Delta t_{n+1}^{n+2}\left[-\frac{\partial (fu_i)^{n+1}}{\partial x_i} + \frac{\partial}{\partial x_i}\left(\rho D \frac{\partial \xi^{n+1}}{\partial x_i}\right)\right]\right\} & \\
\Leftrightarrow \rho^{*,n+2} + \rho'^{,n+2} &= \mathcal{H}_C\left(f^{*,n+2} + f'^{,n+2}\right),
\end{aligned} \tag{B.9}$$

with the following definitions:

$$\begin{aligned}
\rho^{*,n+2} &= \rho^{n+1} - \Delta t_{n+1}^{n+2}\left[\frac{\partial \rho^{n+1} u_i^{*,n+1}}{\partial x_i}\right] \\
f^{*,n+2} &= f^{n+1} - \Delta t_{n+1}^{n+2}\left[\frac{\partial f^{n+1} u_i^{*,n+1}}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\rho D \frac{\partial \xi^{n+1}}{\partial x_i}\right)\right] \\
\rho'^{,n+2} &= -\Delta t_{n+1}^{n+2}\left[\frac{\partial \rho^{n+1} u_i'^{,n+1}}{\partial x_i}\right] \\
f'^{,n+2} &= -\Delta t_{n+1}^{n+2}\left[\frac{\partial f^{n+1} u_i'^{,n+1}}{\partial x_i}\right],
\end{aligned}$$

and

$$u_i'^{,n+1} = -\Delta t_n^{n+1}\frac{1}{\rho_i^{n+1}}\frac{\partial p^{n+1}}{\partial x_i}. \tag{B.10}$$

B.2 Multistage Pressure Correction Algorithm

In order to increase temporal accuracy, the pressure-correction algorithm is put into a multistage loop. We take the explicit form of a 4 stage Runge-Kutta scheme of a system

$\frac{\partial \psi}{\partial t} = G(\psi)$:

$$\begin{aligned}
\psi^{(0)} &= \psi^n \\
\psi^{(1)*} &= \psi^{(0)} + \alpha_1 \Delta t G\left(\psi^{(0)}\right) \\
\psi^{(2)*} &= \psi^{(0)} + \alpha_2 \Delta t G\left(\psi^{(1)}\right) \\
\psi^{(3)*} &= \psi^{(0)} + \alpha_3 \Delta t G\left(\psi^{(2)}\right) \\
\psi^{(4)*} &= \psi^{(0)} + \alpha_4 \Delta t G\left(\psi^{(3)}\right) \\
\psi^{n+1} &= \psi^{(4)}.
\end{aligned} \tag{B.11}$$

Between every stage, the velocity field is corrected in order to obey the velocity constraint. For a general stage l, the velocity field $u^{(l)}$ follows from the requirement that the density, following from continuity or mixture fraction equation, should be the same:

$$\begin{aligned}
\rho^{(l+1)} &= \rho^{n,(0)} - \alpha_{l+1} \Delta t_n^{n+1} \left[\frac{\partial (\rho u_i)}{\partial x_i}\right]^{(l)} \\
f^{(l+1)} &= f^{n,(0)} - \alpha_{l+1} \Delta t_n^{n+1} \left[\frac{\partial (f u_i)}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\rho D \frac{\partial \xi}{\partial x_i}\right)\right]^{(l)},
\end{aligned} \tag{B.12}$$

such that the same Poisson equation follows as for the forward Euler scheme, now at every stage. The velocity field is a combination of the predicted and the corrected one: $u^{(l)} = u^{*,(l)} + u'^{,(l)}$, with

$$u_i'^{,(l)} = -\alpha_l \Delta t_n^{n+1} \frac{1}{\rho_i^{(l)}} \frac{\partial p^{(l)}}{\partial x_i} \tag{B.13}$$

The constraint is now formulated as:

$$\rho^{*,(l+1)} - \delta\rho^{(l)} + \rho'^{,(l+1)} = \mathcal{H}_C\left(f^{*,(l+1)} - \delta f^{(l)} + f'^{,(l+1)}\right), \tag{B.14}$$

with

$$\begin{aligned}
\delta\rho^{(l)} &= \rho^{(l)} - \rho^{n,(0)} \\
\delta f^{(l)} &= f^{(l)} - f^{n,(0)} \\
\rho^{*,(l+1)} &= \rho^{(l)} - \alpha_{l+1} \Delta t_n^{n+1} \left[\frac{\partial \rho^{(l)} u_i^{*,(l)}}{\partial x_i}\right]
\end{aligned}$$

$$
\begin{aligned}
f^{*,(l+1)} &= f^{(l)} - \alpha_{l+1}\Delta t_n^{n+1}\left[\frac{\partial f^{(l)}u_i^{*,(l)}}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\rho D\frac{\partial \xi^{(l)}}{\partial x_i}\right)\right] \\
\rho'^{,(l+1)} &= -\alpha_{l+1}\Delta t_{n+1}^{n+2}\left[\frac{\partial \rho^{(l)}u_i'^{,(l)}}{\partial x_i}\right] \\
f'^{,(l+1)} &= -\alpha_{l+1}\Delta t_{n+1}^{n+2}\left[\frac{\partial f^{(l)}u_i'^{,(l)}}{\partial x_i}\right].
\end{aligned}
$$

Special Treatment in the Final Multistage Step ($l = 4$)

If $l = 4$, the constraint follows from a combination of these 2 equations:

$$
\begin{aligned}
\rho^{n+1,(1)} &= \rho^{n,(4)} - \alpha_1\Delta t_{n+1}^{n+2}\left[\frac{\partial(\rho u_i)}{\partial x_i}\right]^{n,(4)} \\
f^{n+1,(1)} &= f^{n,(4)} - \alpha_1\Delta t_{n+1}^{n+2}\left[\frac{\partial(f u_i)}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\rho D\frac{\partial \xi}{\partial x_i}\right)\right]^{n,(4)}.
\end{aligned}
\tag{B.15}
$$

As a result, if $l = 4$, in the equations, $\alpha_5\Delta t_n^{n+1} = \alpha_1\Delta t_{n+1}^{n+2}$ and $n,(4) = n+1,(0)$ and $\delta\rho^{(4)} = \delta f^{(4)} = 0$. If the time steps do not change, then the first equality simplifies to $\alpha_5 = \alpha_1$. So, we obtain:

$$
\rho^{*,(5)} - \delta\rho^{(4)} + \rho'^{,(5)} = \mathcal{H}_C\left(f^{*,(5)} - \delta f^{(4)} + f'^{,(5)}\right), \tag{B.16}
$$

with

$$
\begin{aligned}
\delta\rho^{(4)} &= 0 \\
\delta f^{(4)} &= 0 \\
\rho^{*,(5)} &= \rho^{(4)} - \alpha_5\Delta t_n^{n+1}\left[\frac{\partial \rho^{(4)}u_i^{*,(l)}}{\partial x_i}\right] \\
f^{*,(5)} &= f^{(4)} - \alpha_5\Delta t_n^{n+1}\left[\frac{\partial f^{(4)}u_i^{*,(l)}}{\partial x_i} - \frac{\partial}{\partial x_i}\left(\rho D\frac{\partial \xi^{(4)}}{\partial x_i}\right)\right] \\
\rho'^{,(5)} &= -\alpha_5\Delta t_{n+1}^{n+2}\left[\frac{\partial \rho^{(4)}u_i'^{,(l)}}{\partial x_i}\right] \\
f'^{,(5)} &= -\alpha_5\Delta t_{n+1}^{n+2}\left[\frac{\partial f^{(4)}u_i'^{,(l)}}{\partial x_i}\right].
\end{aligned}
$$

Appendix C

Analytical Compatibility Constraint Derivation

C.1 Single Fluid Flow: Ideal Gas

The equation of state for an ideal gas, reads:

$$\mathcal{G}\left(\rho, T, \check{p}_0\right) \;=\; \rho T - \check{p}_0 = 0. \tag{C.1}$$

Taking the material derivative, yields:

$$\frac{\partial \mathcal{G}}{\partial \rho}\frac{D\rho}{Dt} + \frac{\partial \mathcal{G}}{\partial T}\frac{DT}{Dt} + \frac{\partial \mathcal{G}}{\partial \check{p}_0}\frac{D\check{p}_0}{Dt} = 0, \tag{C.2}$$

with

$$\frac{\partial \mathcal{G}}{\partial \rho} = T, \tag{C.3}$$

$$\frac{\partial \mathcal{G}}{\partial T} = \rho \tag{C.4}$$

and

$$\frac{\partial \mathcal{G}}{\partial \check{p}_0} = -1. \tag{C.5}$$

The material derivatives are gained from the continuity equation:

$$\frac{D\rho}{Dt} = -\rho\frac{\partial u_i}{\partial x_i}, \tag{C.6}$$

and from the temperature equation:

$$\frac{DT}{Dt} = \frac{1}{\rho}\left[\frac{\gamma-1}{\gamma}\frac{\mathrm{d}p_0}{\mathrm{d}t} + \frac{1}{\mathrm{RePr}}\frac{\partial q_i}{\partial x_i}\right], \tag{C.7}$$

whereas the material derivative of thermodynamic pressure reduces to:

$$\frac{D\check{p}_0}{Dt} = \frac{\mathrm{d}p_0}{\mathrm{d}t}. \tag{C.8}$$

Combination of all expressions, yields a constraint for the velocity:

$$\frac{\partial u_i}{\partial x_i} = -\frac{1}{\gamma}\frac{1}{p_0}\frac{\mathrm{d}p_0}{\mathrm{d}t} + \frac{1}{p_0}\frac{1}{\mathrm{RePr}}\frac{\partial q_i}{\partial x_i}. \tag{C.9}$$

C.2 Two-Fluid Flow: Inert Mixing

The equation of state for an ideal gas reads:

$$\mathcal{G}(\rho,\xi) \;=\; \rho - \rho_B - \left(1-\frac{\rho_B}{\rho_A}\right)\rho\xi = 0. \tag{C.10}$$

Taking the material derivative, yields:

$$\frac{\partial\mathcal{G}}{\partial\rho}\frac{D\rho}{Dt} + \frac{\partial\mathcal{G}}{\partial\xi}\frac{D\xi}{Dt} = 0, \tag{C.11}$$

with

$$\frac{\partial\mathcal{G}}{\partial\rho} = 1 - \left(1-\frac{\rho_B}{\rho_A}\right)\xi = \frac{\rho_B}{\rho} \tag{C.12}$$

and

$$\frac{\partial \mathcal{G}}{\partial \xi} = \left(1 - \frac{\rho_B}{\rho_A}\right) \rho. \tag{C.13}$$

The material derivatives are gained from the continuity equation:

$$\frac{D\rho}{Dt} = -\rho \frac{\partial u_i}{\partial x_i}, \tag{C.14}$$

and from the mixture fraction equation:

$$\frac{D\xi}{Dt} = \frac{1}{\rho}\left[\frac{1}{\mathrm{RePrLe}} \frac{\partial J_i}{\partial x_i}\right]. \tag{C.15}$$

Combination of all expressions, yields a constraint for the velocity:

$$\frac{\partial u_i}{\partial x_i} = -\frac{1}{\mathrm{RePrLe}} \left(\frac{1}{\rho_A} - \frac{1}{\rho_B}\right) \frac{\partial J_i}{\partial x_i}. \tag{C.16}$$

C.3 Two-Fluid Flow: Non-Premixed Combustion

The equation of state for an ideal gas, reads:

$$\mathcal{G}(\rho, \xi) \;=\; \rho - \mathcal{H}_C(\rho\xi) = 0. \tag{C.17}$$

Taking the material derivative, yields:

$$\frac{\partial \mathcal{G}}{\partial \rho}\frac{D\rho}{Dt} + \frac{\partial \mathcal{G}}{\partial \xi}\frac{D\xi}{Dt} = 0, \tag{C.18}$$

with

$$\frac{\partial \mathcal{G}}{\partial \rho} = 1 - \xi \frac{\mathrm{d}\mathcal{H}_C}{\mathrm{d}\rho\xi} \tag{C.19}$$

and

$$\frac{\partial \mathcal{G}}{\partial \xi} = \rho \frac{\mathrm{d}\mathcal{H}_C}{\mathrm{d}\rho\xi}. \tag{C.20}$$

The material derivatives are gained from the continuity equation:

$$\frac{D\rho}{Dt} = -\rho \frac{\partial u_i}{\partial x_i}, \tag{C.21}$$

and from the mixture fraction equation:

$$\frac{D\xi}{Dt} = \frac{1}{\rho}\left[\frac{1}{\mathrm{RePrLe}}\frac{\partial J_i}{\partial x_i}\right]. \tag{C.22}$$

Combination of all expressions, yields a constraint for the velocity:

$$\frac{\partial u_i}{\partial x_i} = -\frac{1}{\mathrm{RePrLe}}\frac{\frac{\mathrm{d}\mathcal{H}_C}{\mathrm{d}\rho\xi}}{\rho - \rho\xi\frac{\mathrm{d}\mathcal{H}_C}{\mathrm{d}\rho\xi}}\frac{\partial J_i}{\partial x_i}. \tag{C.23}$$

Because the equation of state is only imposed through a material derivative, the solution can drift away from the state equation if the state equation is non-linear. The drift can be controlled by means of a defect correction term, with damping factor $0 < \zeta < 1$. The velocity constraint then reads:

$$\frac{\partial u_i}{\partial x_i} = -\frac{1}{\mathrm{RePrLe}}\frac{\frac{\mathrm{d}\mathcal{H}_C}{\mathrm{d}\rho\xi}}{\rho - \rho\xi\frac{\mathrm{d}\mathcal{H}_C}{\mathrm{d}\rho\xi}}\frac{\partial J_i}{\partial x_i} + \frac{\zeta}{\Delta t}\frac{\rho - \mathcal{H}_C(\rho\xi)}{\rho}. \tag{C.24}$$

Appendix D

Solvable Discretisation of the Conductive Fluxes

We start from the semi-discretised equation (8.12), from which the velocity field in a 1D enclosure is derived:

$$-\frac{\rho_R u_{i+\frac{1}{2}} - \rho_L u_{i-\frac{1}{2}}}{\Delta x} + u_i \frac{\rho_R - \rho_L}{\Delta x} = -\frac{\rho}{\mathrm{RePr}} \frac{\partial q}{\partial x}, \tag{D.1}$$

where the cell-face velocities are defined as $u_{i+\frac{1}{2}} = (u_i + u_{i+1})/2$. We consider adiabatic walls at both ends.

A fully discretised version of (D.1) can formally be written in system notation as

$$DU \;\; = \;\; D'Q \Leftrightarrow LP = -DU^* - D'Q, \tag{D.2}$$

with

$$U = [u_1 \;\; u_2 \;\; ... \;\; u_N]^T$$

the velocity vector,

$$Q = [q_1 \;\; q_2 \;\; ... \;\; q_N]^T$$

the conductive flux vector and D and D' discrete divergence operators, $L = DG$ the discrete Laplacian and G the discrete gradient operator. In all operators, extrapolated values for the density can be found.

The system is singular and contains a nullspace of dimension 2, for which $LP = 0$, based on the two pressure vectors

$$P_H = [1 \;\; 1 \;\; ... \;\; 1]^T$$

and

$$P_\pi = \left[1 \;\; -1 \;\; ... \;\; (-1)^{N+1}\right]^T.$$

The same holds for the transpose of the operator L, resulting again in a nullspace of dimension 2, based on 2 vectors R_H and R_π, for which $R^T L = 0$.

(D.2) should be solvable for any choice of U^* and Q. For now, we are not interested in divergence operator D, so we choose $U^* = 0$. Requiring (D.2) to be solvable, yields following solvability conditions:

$$\begin{array}{lll} R_\pi LP = 0 = & -R_\pi D'Q & \text{and} \\ R_H LP = 0 = & -R_H D'Q & \forall Q \in \mathbb{R}^N. \end{array} \tag{D.3}$$

The discrete divergence operator D' must thus have a nullspace of dimension 2, based on the two basis vectors R_π and R_H. Since these two basisvectors cannot easily be determined, and strongly depend on the extrapolated values of the density, the most obvious (and least computationally intensive) choice for the system to be solvable is $D' = D$. The fully discretised equation of (D.1) then becomes

$$\begin{aligned} -\frac{\rho_R u_{i+\frac{1}{2}} - \rho_L u_{i-\frac{1}{2}}}{\Delta x} + u_i \frac{\rho_R - \rho_L}{\Delta x} \;\; = \;\; & -\frac{1}{\mathrm{RePr}} \frac{\rho_R q_{i+\frac{1}{2}} - \rho_L q_{i-\frac{1}{2}}}{\Delta x} \\ & + \frac{1}{\mathrm{RePr}} q_i \frac{\rho_R - \rho_L}{\Delta x}, \end{aligned} \tag{D.4}$$

with $q_{i+\frac{1}{2}} = (q_i + q_{i+1})/2$.

If the equation is discretised as (D.4), it is trivial to see that the system is solvable. Indeed, a solution exists:

$$u_i = q_i + \alpha_1 (P_H)_i + \alpha_2 (P_\pi)_i,$$

with $\alpha_l \in \mathbb{R}, l = 1, 2$.

Bibliography

[1] *http://www.ca.sandia.gov/TNF/abstract.html*.

[2] A.S. Almgren, J.B. Bell, P. Colella, L.H. Howel, and M.L. Welcome. 'A conservative adaptive projection method for variable density incompressible Navier-Stokes equations'. *J. Comput. Phys.*, 142:1–46, 1998.

[3] J.B. Bell, P. Colella, and H.M. Glaz. 'A second order projection method for the incompressible Navier-Stokes equations'. *J. Comput. Phys.*, 85:257–283, 1989.

[4] J.B. Bell, M.S. Day, C.A. Rendleman, S.E. Woosly, and M.A. Zingale. 'Adaptive low Mach number simulations of nuclear flame microphysics'. *J. Comput. Phys.*, 195:677–694, 2004.

[5] J.B. Bell and D.L. Marcus. 'A second order projection method for variable density flows'. *J. Comput. Phys.*, 101:334–348, 1992.

[6] H. Bijl and P. Wesseling. 'A unified method for computing incompressible and compressible flows in boundary-fitted coordinates'. *J. Comput. Phys.*, 141:153–173, 1998.

[7] R.W. Bilger. 'Conditional moment closure for turbulent reacting flow'. *Phys. Fluids*, 5(2):327–334, 1993.

[8] R.W. Bilger. 'Future progress in turbulent combustion research'. *Prog. Energy Combust. Sci.*, 26:367–380, 2000.

[9] R.W. Bilger, S.B. Pope, K.N.C. Bray, and J.F. Driscoll. 'Paradigms in turbulent combustion research'. *Proc. Combust. Inst.*, 30:21–42, 2005.

[10] S.P. Burke and T.E.W. Schumann. 'Diffusion flames'. *Ind. Eng. Chem.*, 20:998–1005, 1928.

[11] V. Bykov and U. Maas. 'The extension of the ILDM concept to reaction-diffusion manifolds'. *Combust. Theory Model.*, 11(6):839–862, 2007.

[12] D.R. Chenoweth and S. Paolucci. 'Natural convection in an enclosed vertical air layer with large horizontal temperature differences'. *J. Fluid Mech.*, 169:173–210, 1986.

[13] A.J. Chorin. 'Numerical solution of the Navier-Stokes equations'. *Math. Comp.*, 22:745–762, 1968.

[14] A.J. Chorin. 'On the convergence of discrete approximations to the Navier-Stokes equations'. *Math. Comp.*, 23:342–353, 1969.

[15] P. Colella and K. Pao. 'A projection method for low speed flows'. *J. Comput. Phys.*, 149:245–269, 1999.

[16] A.W. Cook and J.J. Riley. 'Direct numerical simulation of a turbulent reactive plume on a parallel computer'. *J. Comput. Phys.*, 129:263–283, 1996.

[17] B. Cuenot and T. Poinsot. 'Asymptotic and numerical study of diffusion flames with variable Lewis number and finite rate chemistry'. *Combust. Flame*, 104:111–137, 1996.

[18] A.W. Date. 'Solution of the Navier-Stokes equations on non-staggered grids'. *Int. J. Heat Mass Transf.*, 36(7):1913–1922, 1993.

[19] A.W. Date. 'Fluid-dynamical view of pressure checkerboarding problem and smoothing pressure correction on meshes with colocated variables'. *Int. J. Heat Mass Transf.*, 46(25):4885–4898, 2003.

[20] M.S. Day and J.B. Bell. 'Numerical simulation of laminar reacting flows with complex chemistry'. *Combust. Theory Model.*, 4:535–556, 2000.

[21] G. De Vahl Davis and I.P. Jones. 'Natural convection in a square cavity, a comparison exercise'. *Int. J. Numer. Methods Fluids*, 3:249–264, 1983.

[22] E. Dick. 'A flux-vector splitting method for steady Navier-Stokes equations'. *Int. J. Numer. Methods Fluids*, 9:113–120, 1988.

[23] J.R. Edwards and M.S. Liou. 'Low diffusion flux-splitting methods for flows at all speeds'. *AIAA J.*, 36:1610–1617, 1998.

[24] H. Forkel. 'Über die Grobstruktursimulation turbulenter Wasserstoff-Diffusionsflammen'. PhD Thesis VDI-6-428, TU Darmstadt, Germany, 1999.

[25] O. Gicquel, N. Darabiha, and D. Theévenin. 'Laminar premixed hydrogen/air counter-flow flame simulations using flame prolongation of ILDM with differential diffusion'. *Proc. Combust. Inst.*, 28(2):1901–1908, 2000.

[26] F.H. Harlow and J.E. Welsch. 'Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface'. *Phys. Fluids*, 8:2182–2189, 1965.

[27] R. Hilbert, F. Tap, H. El-Rabii, and D. Thévenin. 'Turbulent Combustion Modelling'. *Prog. Energy Combust. Sci.*, 30:61–117, 2004.

[28] A. Jameson. 'Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings'. Paper 91-1596, AIAA, 1991.

[29] S.Y. Kadioglu, R. Klein, and M.L. Minion. 'A fourth order auxiliary variable projection method for zero-Mach number gas dynamics'. *J. Comput. Phys.*, 227:2012–2043, 2008.

[30] K.C. Karki and S.V. Patankar. 'Pressure based calculation procedure for viscous flows at all speeds in arbitrary configurations'. *AIAA J.*, 27(9):1167–1174, 1989.

[31] J.C. Keck and D. Gillespie. 'Rate-controlled partial-equilibrium method for treating reacting gas mixtures'. *Combust. Flame*, 17(2):237–241, 1971.

[32] R.J. Kee, F.M. Rupley, and J.A. Miller. 'CHEMKIN-II: A FORTRAN chemical kinetics package for the analysis of gas-phase chemical kinetics'. Report SAND80-8003, Sandia National Laboratories, 1989.

[33] A. Kempf. 'Large-eddy simulation of non-premixed turbulent flames'. PhD Thesis VDI-6-513, TU Darmstadt, Germany, 2004.

[34] A.Y. Klimenko. 'Multicomponent diffusion of various admixtures in turbulent flow'. *Fluid Dynam.*, 25:327, 1990.

[35] K.K. Kuo. *Principles of combustion*. John Wiley New York, 1986.

[36] S.H. Lam and D.A. Goussis. 'Understanding complex chemical kinetics with computational single perturbation'. *Proc. Combust. Inst.*, 22:931–941, 1989.

[37] C.K. Law. 'Combustion at crossroads: status and prospects'. *Proc. Combust. Inst.*, 31:1–29, 2007.

[38] P. Le Quéré. 'Accurate solutions to the square thermally driven cavity at high Rayleigh number'. *Comput. Fluids*, 20:29–41, 1991.

[39] B. Lessani and M.V. Papalexandris. 'Time-accurate calculation of variable density flows with strong temperature gradients and combustion'. *J. Comput. Phys.*, 212:218–246, 2006.

[40] F.S. Lien, W.L. Chen, and M.A. Leschziner. 'A multiblock implementation of a non-orthogonal collocated finite volume algorithm for complex turbulent flows'. *Int. J. Numer. Methods Fluids*, 23:567–588, 1996.

[41] U. Maas and S.B. Pope. 'Simplifying chemical kinetics: intrinsic low-dimensional manifods in composition space'. *Combust. Flame*, 88:239–264, 1992.

[42] F. Moukalled and M. Darwish. 'A high-resolution pressure-based algorithm for fluid flow at all speeds'. *J. Comput. Phys.*, 168:101–133, 2001.

[43] V. Moureau, P. Minot, H. Pitsch, and C. Bérat. 'A ghost-fluid method for large-eddy simulations of premixed combustion in complex geometries'. *J. Comput. Phys.*, 221:600–614, 2007.

[44] B. Müller. 'Low-Mach number asymptotics of the Navier-Stokes equations '. *J. Eng. Math.*, 34:97–109, 1998.

[45] C.-D. Munz, S. Roller, R. Klein, and K.J. Geratz. 'The extension of incompressible flow solvers to the weakly compressible regime'. *Comput. Fluids*, 32:173–196, 2003.

[46] H.N. Najm, P.S. Wyckoff, and O.M. Knio. 'A semi-implicit numerical scheme for reacting flow, I. Stiff chemistry'. *J. Comput. Phys.*, 143:381–402, 1998.

[47] K. Nerinckx, J. Vierendeels, and E. Dick. 'Mach uniformity through the coupled pressure and temperature correction algorithm'. *J. Comput. Phys.*, 206:597–623, 2005.

[48] F. Nicoud. 'Conservative high order finite difference schemes for low-Mach number flows'. *J. Comput. Phys.*, 158:71–97, 2000.

[49] C.W. Oosterlee and P. Wesseling. 'A robust multigrid method for a discretization of the incompressible Navier-Stokes equations in general coordinates'. *Impact Comp. Sci. Eng.*, 5:128–151, 1993.

[50] S. Paolucci. 'On the filtering of sound from the Navier-Stokes equations'. Report SAND82-8257, Sandia National Laboratories, 1982.

[51] S.V. Patankar and D.B. Spalding. 'A calculation procedure for heat and mass transfer in three-dimensional parabolic flows'. *Int. J. Heat Mass Transf.*, 15:1787–1806, 1972.

[52] M. Perić, R. Kessler, and G. Schreuerer. 'Comparison of finite-volume numerical methods with staggered and colocated grids'. *Comput. Fluids*, 16(4):389–403, 1998.

[53] N. Peters. 'Laminar diffusion flamelet models in non-premixed turbulent combustion'. *Prog. Energy Combust. Sci.*, 10(1):319–339, 1984.

[54] T. Poinsot. 'Using direct numerical simulations to understand premixed turbulent combustion'. *Proc. Combust. Inst.*, 26:219–232, 1996.

[55] T. Poinsot, A. Trouvé, and S. Candel. 'Applications of direct numerical simulations of premixed turbulent combustion'. *Prog. Energy Combust. Sci.*, 21:531–576, 1996.

[56] T. Poinsot and D. Veynante. *Theoretical and numerical combustion*. Edwards, Philadelphia, PA, 2001.

[57] S.B. Pope. 'Ten questions concerning the large-eddy simulation of turbulent flows'. *New J. Phys.*, 6(35), 2004.

[58] P. Rauwoens, J. Vierendeels, and B. Merci. 'A solution for the odd-even decoupling problem in pressure-correction methods for variable density flows'. *J. Comput. Phys.*, 227:79–99, 2007.

[59] P. Rauwoens, J. Vierendeels, and B. Merci. 'A stable pressure-correction scheme for variable density flows involving non-premixed combustion'. *Int. J. Numer. Methods Fluids*, 56(8):1465–1471, 2008.

[60] P. Rauwoens, J. Vierendeels, and B. Merci. 'Numerical study of the flow in a three-dimensional thermally driven cavity'. *J. Comput. Appl. Math.*, 512(2):538–546, 2008.

[61] Z. Ren, S.B. Pope, A. Vladimirsky, and J.M. Guckenheimer. 'The invariant constrained equilibrium edge preimage curve method for the dimension reduction of chemical kinetics'. *J. Chem. Phys.*, 124(114111), 2006.

[62] C.M. Rhie and W.L. Chow. 'Numerical study of the turbulent flow past an isolated airfoil with trailing edge separation'. *AIAA J.*, 21(11):1525–1532, 1982.

[63] T. Schneider, N. Botta, K.J. Geratz, and R. Klein. 'Extension of finite volume compressible flow solvers to multi-dimensional, variable density zero Mach number flows'. *J. Comput. Phys.*, 155:248–286, 1999.

[64] A. Segal, P. Wesseling, J. Van Kan, C.W. Oosterlee, and K. Kassels. 'Invariant discretization of the incompressible Navier-Stokes equations in boundary fitted coordinates'. *Int. J. Numer. Methods Fluids*, 15:411–426, 1992.

[65] A.X. Sengissen, J.F. Van Kampen, R.A. Huls, G.G.M. Stoffels, J.B.W. Kok, and T.J. Poinsot. 'LES and experimental studies of cold and reacting flow in a swirled partially premixed burner with and without fuel modulation'. *Combust. Flame*, 150:40–53, 2007.

[66] I. Senocak and W. Shyy. 'A pressure-based method for turbulent cavitating flow computations'. *J. Comput. Phys.*, 176:363–383, 2002.

[67] J. Smagorinsky. 'General circulation experiments with primitive equations, 1, the basic experiment'. *Mon. Weather Rev.*, 91:99–164, 1963.

[68] D. Tafti. 'Alternate formulations for the pressure equation Laplacian on a collocated grid for solving the unsteady incompressible Navier-Stokes formulations'. *J. Comput. Phys.*, 116:143–453, 1995.

[69] A.G. Tomboulides, Y.C.Y Lee, and S.A. Orszag. 'Numerical simulation of low-Mach number reactive flows'. *J. Sci. Comput.*, 12(2):139–167, 1997.

[70] A.G. Tomboulides and S.A. Orszag. 'A quasi-two-dimensional benchmark problem for low Mach number compressible codes'. *J. Comput. Phys.*, 146:691–706, 1998.

[71] D.R. Van der Heul, C. Vuik, and P. Wesseling. 'A conservative projection method for flow at all speeds'. *Comput. Fluids*, 32:1113–1132, 2003.

[72] S. Van der Hoeven, B.-J. Boersma, H.J.J. Jonker, and D.J.E.M. Roekaerts. 'Development of a large eddy simulation code for turbulent non-premixed jet flames'. In *2nd ECCOMAS Thematic Conference on Computational Combustion*, Delft, The Netherlands, 2007. TU Delft.

[73] J.A. van Oijen, F.A. Lammers, and L.P.H. de Goey. 'Modelling of complex premixed burner systems using flamelet-generated manifolds'. *Combust. Flame*, 127:2124–2134, 2001.

[74] L. Vervisch and P. Domingo. 'Two recent developments in numerical simulation of premixed and partially premixed turbulent flames '. *C. R. Mec.*, 334:523–530, 2006.

[75] L. Vervisch and T. Poinsot. 'Direct numerical simulation of non-premixed turbulent flame.'. *Annu. Rev. Fluid Mech.*, 30:655–692, 1998.

[76] D. Veynante and L. Vervisch. 'Turbulent Combustion Modelling'. *Prog. Energy Combust. Sci.*, 28:193–266, 2002.

[77] J. Vierendeels, B. Merci, and E. Dick. 'Benchmark solutions for the natural convective heat transfer problem in a square cavity with large horizontal temperature differences'. *Int. J. Num. Methods Heat and Fluid Flow*, 13(8):1057–1078, 2003.

[78] J. Vierendeels, B. Merci, and E. Dick. 'A multigrid method for natural convective heat transfer with large temperature differences'. *J. Comput. Appl. Math.*, 168:509–517, 2004.

[79] J. Vierendeels, K. Riemslagh, and E. Dick. 'A multigrid semi-implicit line-method for viscous incompressible and low-Mach-number flows on high aspect ratio grids'. *J. Comput. Phys.*, 154:310–341, 1999.

[80] C. Wall, C.D. Pierce, and P. Moin. 'A semi-implicit method for resolution of acoustic waves in low Mach number flows'. *J. Comput. Phys.*, 181:545–563, 2002.

[81] Y. Wang and A. Trouvé. 'Artificial acoustic stiffness reduction in fully compressible, direct numerical simulation of combustion'. *Combust. Theory Model.*, 9(3):633–660, 2004.

[82] C.K. Westbrook, Y. Mizobuchi, T.J. Poinsot, P.J. Smith, and J. Warnatz. 'Computational combustion'. *Proc. Combust. Inst.*, 30:125–157, 2005.