NUMERICAL SIMULATION OF SHIP STERN FLOWS
WITH A SPACE-MARCHING NAVIER-STOKES METHOD

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. ir. K.F. Wakker,
in het openbaar te verdedigen ten overstaan van een commissie,
door het College voor Promoties aangewezen,

op donderdag 7 oktober 1999 te 10.30 uur

door

Martin HOEKSTRA
scheepsbouwkundig ingenieur
geboren te Heerenveen
Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. ir. P. Wesseling

Samenstelling promotiecommissie:

Rector Magnificus, voorzitter
Prof. dr. ir. P. Wesseling
Prof. V.C. Patel
Prof. dr. ir. J.W. Boerstoel
Prof. dr. ir. A.J. Hermans
Prof. dr. ir. G. Kuiper
Prof. dr. ir. F.T.M. Nieuwstadt
Prof. dr. A.E.P. Veldman

Technische Universiteit Delft, promotor
University of Iowa, USA
Technische Universiteit Delft
Technische Universiteit Delft
Technische Universiteit Delft
Technische Universiteit Delft
Rijksuniversiteit Groningen

ISBN 90-9013026-8

Copyright © 1999 by M. Hoekstra, MARIN, Wageningen. All rights reserved.
5.7 Momentum equation ........................................ 62
  5.7.1 Convection terms ...................................... 62
  5.7.2 Pressure terms ....................................... 64
  5.7.3 Viscous terms ....................................... 64
  5.7.4 External force terms ................................ 65
5.8 Resulting stencils ....................................... 66
5.9 Change of variables ...................................... 66
5.10 Eddy-viscosity transport equation ..................... 66
5.11 Implementation of boundary conditions ................. 66

6 Iterative solution process ............................... 71
  6.1 Global iteration process ............................... 75
    6.1.1 Grid sequencing .................................. 76
    6.1.2 Symmetric Gauss-Seidel ......................... 76
    6.1.3 Approximate multigrid ....................... 80
    6.1.4 Over-all strategy ................................ 81
  6.2 Local iteration process ................................ 81
    6.2.1 CILU or preconditioned GMRES .................. 82
    6.2.2 CILU7 or CILU(ε) ................................. 83

7 Numerical verification ................................... 89
  7.1 The laminar wake of a flat plate ..................... 89
  7.2 The flow at the tail of a modified spheroid ........ 96
  7.3 The flow near the stern of the Wigley hull ........ 100
  7.4 Summary of results ................................... 106

8 Application and validation ............................... 107
  8.1 Two-dimensional and axisymmetric flows ............. 107
    8.1.1 Flat plate, laminar flow ....................... 107
    8.1.2 Flat plate, turbulent flow ..................... 111
    8.1.3 Modified spheroid ................................ 113
    8.1.4 Spheroid ......................................... 116
  8.2 Three-dimensional flows ................................ 119
    8.2.1 HSVA tanker at Re = 5 \times 10^6 .................. 119
    8.2.2 HSVA tanker at Re = 2 \times 10^9 ................ 128
    8.2.3 HSVA tanker with propeller ..................... 130

9 Closure .................................................. 135

A Implementations of flux-limiting convection scheme .... 137

B Stability analysis of predictor-corrector scheme ....... 140

C Collected results for HSVA tanker ........................ 144
Bibliography 183
List of symbols 193
Summary 195
Samenvatting 197
Acknowledgement 200
Curriculum Vitae 201
Chapter 1

Introduction

When a ship is viewed from some distance while it steadily advances on a straight course through still water, the aesthetically appealing wave pattern, trailing with the ship, is usually the most apparent flow disturbance. Below the water surface, however, largely invisible for the human observer, additional turmoil is created. In spite of the small viscosity of water, a motion is imparted to a substantial amount of fluid because of the no-slip condition on the hull surface. Related to this imparted motion is the viscous drag of the hull, nearly always the dominant contribution to the hull resistance. The propulsor of the ship, mounted behind the stern and thus operating in the ship-disturbed flow field, adds considerably to the complexity of the water motion in overcoming both the viscous and the wave-making resistance.

The flow phenomena touched upon above are at the basis of some classical problems of hydrodynamic ship design. For example: the minimisation of the ship's resistance under certain hull-geometric constraints; the design and the analysis of the propulsor, usually a screw propeller; the improvement of the performance of the hull and the propeller as two interacting bodies, with regard not only to the propulsive efficiency but also to the suppression of cavitation, vibrations and noise. The idealised environmental condition of the water being at rest until the ship arrives to disturb it is herein considered to yield a representative average of what happens in a wind-disturbed sea.

In dealing with these problems, naval architects have, for more than a century, relied primarily on experiments with small scale models of the ship under investigation; the role of computational methods has been comparatively modest. This cannot be attributed to the lack of an accurate mathematical description of the flow. On the contrary, fluid mechanics is a branch of physics with a completely satisfactory theoretical model, consisting of the so-called Navier-Stokes equations. As a matter of fact, these equations were established even before the time that systematic ship model experiments were initiated. The minor role of computational techniques is rather due to the intrinsic complexity of the equations, prohibiting analytical solution while requiring for numerical solution much greater computer power than currently available. At present, only approximate forms of the Navier-Stokes
equations can be solved in practical applications.

A major difficulty is that the Navier-Stokes equations allow the occurrence of turbulence. There is no prospect yet of direct simulation of this intricate phenomenon over its full range of time and length scales in the prediction of the flow around a ship; only the time-mean effect of turbulence via the apparent Reynolds stresses can be considered. A satisfactory modelling of these stresses has been achieved for 'simple' flows such as for instance thin shear layers, if the local production of turbulence is in a fair balance with its dissipation; but in more complex flows turbulence modelling is a persistent trouble.

Even the mathematical model that remains after Reynolds averaging and the consequent introduction of closure relations was for a long time beyond the reach of numerical solution, at least for flows of practical importance. A further reduction of the mathematical model, either by dropping terms in the equations or in the form of approximate boundary conditions, was necessary to bring the cost of computation down to an acceptable level. All such approximations introduce phenomenological errors, in the sense that certain physical processes are not, or only approximately, represented in the computational simulation. This does not necessarily spoil the outcome, but obviously brings about a diminished range of applicability. It should not be surprising then that engineers, responsible for the design of a ship, have in many instances confided, and are still confiding, in model tests more than in computations, and have chosen their design strategy accordingly.

But computational capability is growing rapidly, if only because of the continuing increase in the capacity and improvement of the performance of computers. Indeed, Kutler (1985) lists the development of computer technology as one of the four items which effectively determine the progress in computational fluid dynamics (the others being related to turbulence modelling, grid generation and algorithm development). With the advent of supercomputers, the numerical solution of the Reynolds-averaged Navier-Stokes (RANS) equations has become feasible for many flow problems of practical relevance. Accordingly, construction of suitable algorithms has been taken up for a variety of problems in several branches of engineering. From the great diversity of solution methods proposed — with remarkable differences between the handling of compressible and incompressible flows or internal and external flows — it may be concluded that this is not a straightforward exercise. But such an algorithm, when sufficiently general and carefully validated, is envisaged to yield results comparable in quality and truthfulness with those of model tests, and to open up new ways of acquiring design information. Impressive applications to increasingly complex flows are being reported one after the other; they will not fail to have an impact on design procedures sooner or later.

This is not to say that numerical ship hydrodynamics is expected to make towing tank experiments redundant for investigations related with the steady rectilinear motion of ships in still water, let alone for answering questions connected with the more practical situation where the ship operates in an irregular seaway. But as a complement or a preliminary to model tests its good prospects are widely recognised.
Chapter 1: Introduction

After all, experimental investigations are not without shortcomings either. Model tests have the inherent difficulty that complete dynamic similarity with the full-scale flow cannot be achieved. Flow calculations have practically no restrictions in this respect, and may help to solve or elucidate scale effect problems. Perhaps even more significant is the inefficiency of model experiments in giving guidance for possible improvements in the design of the hull under consideration. Computations, on the other hand, produce detailed information about the complete flow field from which useful indications for modifications can often readily be inferred. An important aspect is further that the expenses involved in experimentation will undoubtedly increase continually, while the cost of computation tends to do the opposite.

What is the current situation in ship design with regard to the application of numerical flow simulation? If the purport of this question is restricted to the study of the performance of ships in still water, we observe that almost exclusively use is made of inviscid flow models. The basis of propeller design and analysis procedures is usually a lifting line or lifting surface model. In spite of the fact that such models assume the inflow to the propeller to be irrotational – in great contrast with reality –, their role in the design process is well-established; advanced propeller design relies to an appreciable extent on the tools of computational fluid dynamics. The calculation of the inviscid flow around a ship’s hull, mostly with source panel methods, is also carried out routinely, and is typically used to judge certain aspects of the hull form design and to minimize wave generation by optimization of the fore body shape or the transom stern. Incidentally free surface effects are neglected, but as a rule the ship-generated wave pattern is included in the numerical simulation, and the success of the methods employing the exact non-linear free surface condition has made those using a linearised free surface condition practically obsolete.

Viscous flow computations are applied to a much lesser extent. It is true that considerable efforts have been made in developing boundary-layer codes, but routine use has never been established, the primary reason being failure in predicting the flow in the vicinity of the stern. Boundary-layer theory is too restrictive to produce reliable figures for the viscous resistance, or to give useful information about the inflow to the propeller. For more realism in the predictions, recourse must be taken to the RANS equations (or a close approximation of them) and to a truly three-dimensional geometric description of the flow domain as a basis. The first attempts to develop such methods date from the late 1970’s. Stern flow prediction has been a prominent research topic in ship hydrodynamics ever since.

By now, considerable progress has been made and the reader might consult the proceedings of two workshops on ship-related CFD, one held in Gothenburg (Larsson et al., 1991) and the other in Tokyo (Kodama et al., 1994), to get an impression of the achievements. We observe that the great majority of the methods taking part in the contests of these workshops neglected the free surface disturbances; indeed, the tools which aimed at solving the RANS equations with a free surface need a substantial further effort to mature. Furthermore, none of the participants could claim to have produced results which were satisfactory in all respects. Notably
Numerical simulation of ship stern flows

<table>
<thead>
<tr>
<th>Code name</th>
<th>Developers</th>
<th>Country</th>
<th>Representative reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>NICE</td>
<td>Kodama, Hinatsu</td>
<td>Japan</td>
<td>Kodama (1992)</td>
</tr>
<tr>
<td>SHIPFLOW</td>
<td>Larsson, Broberg</td>
<td>Sweden</td>
<td>Larsson et al. (1990)</td>
</tr>
<tr>
<td>RANSTERN</td>
<td>Patel, Stern</td>
<td>USA</td>
<td>Chen, Patel &amp; Ju (1990)</td>
</tr>
<tr>
<td>HORUS</td>
<td>Piquet, Visonneau, Deng</td>
<td>France</td>
<td>Deng et al. (1993)</td>
</tr>
<tr>
<td>PARNASSOS</td>
<td>Hoekstra, Raven, Eça</td>
<td>Netherlands</td>
<td>Hoekstra &amp; Eça (1998)</td>
</tr>
</tbody>
</table>

Table 1.1: List of main ship-oriented RANS-code developments

the predictions of the velocity distribution in the vicinity of the propeller for high-block ships like tankers fell short, which is now generally attributed to inadequate turbulence modelling, though insufficient grid resolution may also play a part.

Even with such limitations in modelling accuracy or in applicability, the use of RANS methods for the analysis of practical ship designs is now cautiously being taken up. At some places commercial general-purpose codes like Fluent, CFX, Tascflow and StarCD are applied, at other places the special requirements of ship-hydrodynamic applications were considered to justify an own development. In the latter category the best-known and openly published contributions are listed in Table 1.1.

As appears from the table, the author – as an employee of the Maritime Research Institute Netherlands (MARIN) – has been involved in the development of PARNASSOS. This RANS-code will be the subject of the present thesis; its main features will be discussed and analysed, and results of application will be shown. Whereas the first four methods listed in Table 1.1 use either the pressure-correction technique or the artificial compressibility approach as the basic iteration strategy, PARNASSOS is based on space-marching iteration. A major purpose of this thesis is therefore to give a justification of this approach for ship stern flow simulation, or more generally, for simulation of external flows around more or less slender bodies at high Reynolds number, and to show its dependability in representative applications.

To lighten our task, we assume that the free surface disturbances created by the ship are of little significance, so that we are allowed to eliminate gravity wave formation by treating the undisturbed free surface as a symmetry plane. Else we would have to cope with some serious complications. For example, the shape of the free surface as well as the trim and sinkage of the ship, being not known \textit{a priori}, would have to be obtained as a part of the solution; in other words, the location of the boundaries of the flow domain would not be known in advance. Also the avoidance of numerical damping of the ship-generated waves requires a grid resolution that is not currently feasible in the ship design practice.

We do include the possibility of simulating the effects of a propeller on the flow, but only via a simple infinitely-bladed propeller model (actuator disk). Among

\footnote{Now superseded by CFDSHIP-IOWA (Stern et al., 1996)}
other problems, the essential unsteadiness of the flow caused by the finite number of propeller blades is thereby avoided. Since, furthermore, turbulence is considered only in so far as its approximate time-mean effect is concerned, a steady flow problem remains. Moreover, for our purposes compressibility effects may be safely left out of account, since the ship speed is far below the speed of sound.

External flows, like the flow around a ship in what is assumed to be an unbounded fluid domain, call for a zonal approach. While the ship moves at service speed, the Reynolds number based on this speed and the ship’s length is in the order of $10^6$ to $10^9$, when the full range of towing tank model to actual ship is considered. This implies that viscous effects are confined to fairly thin shear layers: the boundary layer and the wake. The flow outside these shear layers may be considered as effectively inviscid and irrotational. The RANS equations are unnecessarily complicated as a mathematical model for that part of the flow, and the same holds for the part where the boundary layer is sufficiently thin; they are required only in a region in the proximity of the stern. So, for reasons of computational economy, the flow field is usually divided into three zones, each with its own mathematical model of the flow (Fig. 1.1), viz.:

- the potential flow zone (I), the outer region where the flow is considered as inviscid and irrotational; the flow velocity may be written as the gradient of a scalar potential while the dynamics of the flow is governed by Bernoulli’s law;

- the boundary-layer zone (II), being a thin layer covering the forward part of the ship hull in which the flow is essentially viscous, but where simplifications
Numerical simulation of ship stern flows

in the modelling are allowed in view of the thinness of the layer in which the viscous effects manifest themselves;

- the stern-flow-and-wake zone (III), again a viscous flow region, enclosing the rapidly growing boundary layer on the aft part of the hull and the ensuing wake, where a free interaction between velocity and pressure fields must be allowed.

The location of the borders between these zones is not well-defined. In fact, there is a trend to combine zones II and III into one, implying that the RANS equations are applied in the bow region as well. If, in addition, the border between the viscous and the inviscid flow regions is chosen sufficiently far from the ship, the need for an explicit solution of the outer flow may disappear, so that a single-zone calculation would suffice. However, we favour the three-zones approach under currently prevailing circumstances as yielding the best accuracy for a given computational effort.

The potential flow in the outer zone is efficiently and — in consideration of the infinite extent of the domain — conveniently determined by a boundary-integral method. We use the Hess and Smith panel method (Hess & Smith, 1962) for that purpose. This method is well established and does not need to be further discussed here.

A solution of the boundary-layer equations is obtained by a procedure described in Hoekstra (1981). By leaving a detailed review of that part of the computational technique out of the present account, we do not in the least wish to suggest that there are no problems left in that area. Notably the prediction of the initial development of the boundary layer on the bow is still a matter of great concern. Apart from geometrical complexity, there is the notorious difficulty of correctly predicting the transition of the boundary-layer from a laminar to a turbulent state, aggravated by the possible occurrence of multiple stagnation points. The author dealt with such matters for a certain class of ship bows (Hoekstra, 1978), but without the pretension of giving a conclusive answer.

The computation method for the flow in the stern-flow-and-wake zone will be outlined in what follows. The need for high computational efficiency has played a significant role in its development. The time available for any numerical flow analysis in a typical ship design process is extremely short, in the order of 2 to 4 weeks. Moreover, budgets are low. In developing a solution strategy, we shall therefore exploit to some extent a specific feature of this flow. Ships are designed to have a relatively low drag when steaming ahead. The length is always much greater than the breadth and the draught, and massive flow separation will be absent. Thus the forward speed gives rise to a predominant flow direction in a ship-fixed reference frame. This feature, typical for many high Reynolds number flows, will be presupposed and utilized in an attempt to reduce thereby the computational effort. It would not be realistic, however, to exclude completely the possibility of flow separation; small regions of flow reversal can occur and must be allowed. This
requirement will turn out to be reconcilable with the exploitation of the fact that we roughly know in advance the direction which the flow will take.

The method to be described will here in many instances be referred to as PARNASSOS, the name of the code currently in use at the Maritime Research Institute Netherlands. But while this thesis will present the essential features of PARNASSOS, it does not describe all details. Moreover, several new elements will be introduced, so that the present work is rather a contribution to the development of PARNASSOS than a full documentation of the code.

Let us close this introduction with an outline of the remainder of this thesis. In Chapter 2 the mathematical model of the flow in the stern-flow-and-wake zone is introduced and its intrinsic features are analysed. The formulation is in terms of a Cartesian coordinate system. A reformulation in general curvilinear coordinates is given in Chapter 4, where a choice is made between various alternatives. The problem of grid generation will then already have been considered in Chapter 3. Also the properties of the grid structure will there be reviewed. We then turn to the subject of discretisation in Chapter 5, followed by an outline of the process for iterative solution of the ensuing set of algebraic equations in Chapter 6. In Chapter 7 the numerical behaviour of the method is evaluated, while Chapter 8 is devoted to the application of the method to the type of problems for which it has been developed. The main achievements are summarised in the final Chapter 9. For convenience of the reader a list of the main symbols is included, directly following the bibliography.
Chapter 2

Mathematical model

The purpose of this chapter is to introduce the mathematical equations which form the basis of our calculation scheme. Striving for compact and transparent formulations, we have chosen to use various forms of notation. In particular, the reader is supposed to be familiar with vector, suffix and tensor notation, including the summation convention implied by the latter (Aris, 1962).

Instead of considering a ship proceeding in a fluid at rest, we prefer to move as an observer with the ship, i.e. the ship is held fixed in a uniform onset flow of velocity $\vec{U}_\infty$. The undisturbed free surface is treated as a symmetry plane; adding the mirror image above this plane, we actually consider a so-called double model of the submerged part of the ship in an unbounded fluid domain. The fluid is incompressible, has uniform mass density $\rho$, and its dynamic viscosity is $\mu$; the kinematic viscosity $\nu = \mu/\rho$ is, as usual, represented by the symbol $\nu$. When the length of the ship is denoted by $L$, a characteristic Reynolds number can be defined as $Re = |\vec{U}_\infty|L/\nu$.

Let a right-handed rectangular Cartesian coordinate system $(x, y, z)$ be attached to the ship, with the $x$-axis pointing from bow to stern, i.e. in the direction of $\vec{U}_\infty$. In tensor notation, $x_1, x_2, x_3$ is equivalent to $x, y, z$. The components of the velocity vector $\vec{U}$ in the $x, y, z$ directions are denoted either as $u, v, w$ or as $u_1, u_2, u_3$, according to which is most convenient; the symbol used for the pressure is $p$. To avoid confusion in the mixed usage of suffix and tensor notation, it is understood that a partial derivative is implied by a suffix only if $x, y$ or $z$ appear as such.

In this chapter all equations will be formulated in terms of the Cartesian coordinates $x, y, z$ and the dependent variables $u, v, w, p$. As far as needed in the discussion, we will assume for simplicity that the $x$-direction can everywhere be considered as the dominant flow direction, and when reference needs to be made to the hull-normal direction, the $y$-coordinate will serve as such. To make these assumptions more realistic, the reader may imagine the ship’s hull to be reduced to a flat plate. Later, in Chapter 4, a general formulation in curvilinear coordinates, fitted around the ship in its true proportions, is presented. Dominant flow direction and dependent variables will then be redefined.
2.1 Equations of mass and momentum conservation

An incompressible fluid flow is governed by two conservation laws, those of mass and of momentum. Under the assumptions of a linear dependence of the stress on the rate of strain (Newtonian fluid) and a uniform viscosity (no appreciable temperature variations), Newton's second law leads to the Navier-Stokes equation, a vector equation with the velocity components and the pressure as the dependent variables. Mass conservation yields an additional scalar equation, implying a kinematic constraint on the velocity field. For steady flow and after Reynolds averaging, these equations read:

\[
\begin{align*}
\rho u_j u_{i,j} + p_i - \mu u_{i,jj} + \rho (u'^i u'_j)_{,j} &= f_i \quad (2.1) \\

u_{i,i} &= 0 \quad (2.2)
\end{align*}
\]

where \( \rho u'_i u'_j \) is the Reynolds stress tensor, to be further dealt with in the following section, and \( f_i \) is an external force per unit of volume. The force term \( f_i \) will be used for the representation of the effects of the propeller on the flow; the influence of gravitation, expressible as the gradient of a scalar, is absorbed in the pressure terms.

As mentioned in the introduction, this equation system has been derived long ago, but is, even after Reynolds averaging, not easily solved. Consequently, a fluid dynamicist normally starts to consider whether his problem allows some reduction or simplification of the equations. The boundary-layer equations are such a reduced system. An order-of-magnitude analysis, set up under the thin-shear-layer assumption, gives the attractive result that the pressure is determined to first approximation by the boundary conditions alone, leaving only the velocity field to be evaluated. Moreover, several stress terms can be neglected.

Unfortunately, the flow around the stern of a ship does not permit a reduction of the equations with similar implications. In any trailing-edge flow, let the body shape be ever so simple, pressure/velocity interaction is important, as substantiated by triple-deck theory (Stewartson, 1974). Interacting boundary-layer theory, treating the mainstream pressure gradient in the viscous layer as dependent on the derivative of the displacement thickness via a Cauchy integral while the lateral pressure gradients are assumed to vanish, may still give satisfactory results for the flow near the trailing edge of an airfoil (Veldman, 1984). But it is a matter of no debate that, near the aft end of a ship, streamline curvature is strong enough to induce significant pressure variations across the viscous layer. Therefore, there is no ground to specify in advance the pressure field near the stern.

There are however good reasons to estimate some viscous terms as being negligible. As pointed out in the introduction, the flow around a ship is, like many high-\( Re \) flows, characterised by a predominant flow direction. Convection dominates diffusion, and precludes large second derivatives in the flow direction. The only exception to this rule might be in the immediate vicinity of a trailing edge.
Within the framework of triple-deck theory it has been shown that the full Navier-Stokes equations are required there, but the size of the region is only $O(Re^{-3/4}L)$, which is $O(Re^{-1/4}\delta)$ if $\delta$ is the boundary-layer thickness (Veldman, 1976). For the large Reynolds numbers of interest to us this is so small that it is not properly resolved in the grids, used in our practice. Or, put in another way, if the triple-deck formulation is accepted as a sufficient description of the trailing edge flow, we are allowed to neglect mainstream diffusion, because it is not part of the mathematical model of the lower deck. As a consequence, there is for us no reason to include diffusion effects in the mainstream direction — here taken to be the $x$-direction —; the associated terms in equation (2.1) may as well be dropped and we shall in fact do so by limiting the summation of $j$ in those terms to the range $[2,3]$.

The resulting, slightly reduced set of equations has frequently been called the 'parabolised' or 'partially parabolic' or 'semi-elliptic' or 'reduced' Navier-Stokes equations. Although having adopted the first of these adjectives too for some time, we are reluctant now in using any of them. If not inappropriate at all, they fix the attention on a point of little relevance, at least if incompressible flows are to be modelled. As such they have given rise to a great deal of confusion.

To avoid any such confusion and to bring out what we have gained by dropping some viscous terms, it is worthwhile to make a short detour at this stage and to trace back the origin of some of the above cited adjectives. The whole issue hinges on the classification of the set of partial differential equations as elliptic or parabolic. We start therefore to describe this classification in mathematical terms. In doing so, we follow closely the analysis given by Brandt & Dinar (1979) who have considered various elliptic systems of equations, among which the Navier-Stokes equations.

### 2.1.1 Classification as an elliptic system

Systems of nonlinear equations are considered to have elliptic character when their linearised form is elliptic. After linearisation the Navier-Stokes equations can be written as

\[
\begin{pmatrix}
\mathcal{L} & 0 & 0 & \partial/\partial x \\
0 & \mathcal{L} & 0 & \partial/\partial y \\
0 & 0 & \mathcal{L} & \partial/\partial z \\
\partial/\partial x & \partial/\partial y & \partial/\partial z & 0
\end{pmatrix}
\begin{pmatrix}
u \\
v \\
w \\
p
\end{pmatrix}
= 
\begin{pmatrix}
f_1 \\
f_2 \\
f_3 \\
0
\end{pmatrix}
\]  

(2.3)

in which $\mathcal{L}$ is the linear differential operator

\[
\mathcal{L} = \rho u_j \partial/\partial x_j - \mu \partial^2/\partial x_j^2.
\]

This system is called elliptic if it is solvable for all components of the Fourier expansion $\phi_i = A_\phi \exp(ik_i x_j)$ in which $\phi$ stands for any one of the dependent variables
or \( f, \ k_j \) is the wave number vector and \( \iota = \sqrt{-1} \). The Fourier transform of (2.3) is

\[
\begin{pmatrix}
\mathcal{L}_k & 0 & ik_1 \\
0 & \mathcal{L}_k & 0 \\
-ik_1 & 0 & \mathcal{L}_k
\end{pmatrix}
\begin{pmatrix}
A_u \\
A_v \\
A_p
\end{pmatrix}
= \begin{pmatrix}
A_{f_1} \\
A_{f_2} \\
A_{f_3}
\end{pmatrix}
\] (2.4)

with

\[ \mathcal{L}_k = \iota pu_j k_j + \mu k_j k_j. \]

The determinant of the matrix on the left-hand side of (2.4), \( \text{viz.} \)

\[ \text{det} = (\mathcal{L}_k)^2 k_j k_j, \]

is called the characteristic symbol of the equation system. It should not vanish for all real \( k_j \) with \( |k| > 0 \) for the system to be called elliptic. Thus ellipticity requires

\[ (\mathcal{L}_k)^2 k_j k_j \neq 0 \quad \text{for} \quad |k| = \sqrt{k_1^2 + k_2^2 + k_3^2} > 0. \]

As a matter of fact it suffices that the condition holds in a small domain, hence for sufficiently high wave numbers, so that only the part of the determinant of highest degree in \( k \) needs to be considered. That part is called the principal symbol \( \mathcal{P} \) of the equation system; it is

\[ \mathcal{P} = \mu(k_1^2 + k_2^2 + k_3^2)^2(k_1^2 + k_2^2 + k_3^2) = \mu^2 |k|^6, \] (2.5)

whence the Navier-Stokes equations are called an elliptic system of sixth order.

The character of the equations is not changed by the process of Reynolds averaging, if the Reynolds stresses are assumed to be related to the mean flow through an eddy viscosity. It is easily verified that the principal symbol changes then only as regards the value of the viscosity.

If we turn next to the equations obtained after dropping the mainstream diffusion terms, the principal symbol takes the modified form:

\[ \mathcal{P}^* = \mu(k_2^2 + k_3^2)^2(k_1^2 + k_2^2 + k_3^2). \] (2.6)

This form does not strictly satisfy the above criterion, and it is tempting to call the system non-elliptic. But upon further consideration, neither can it be called parabolic or hyperbolic (Courant & Hilbert, 1968). Hyperbolicity requires all roots of each wave number component for fixed values of the other two to be real, which is evidently not the case. Conversely, a parabolic character could have been claimed only if the principal symbol (2.6) had become degenerate by a complete disappearance of \( k_1 \). Thus, we have a case not covered by the classical definitions.

In order to allow classification at all, a less strict definition must be used. Briley & McDonald (1984) have studied the character of various reduced forms of the
Navier-Stokes equations. They consider equations to be elliptic, or at least ill-posed for solution as an initial-value problem, if the principal symbol has one or more imaginary roots. In the first quadratic factor of (2.6), $k_1$ is missing: this is a degenerate parabolic factor with $x$ as a time-like coordinate. However, the last factor in the principal symbol, related to the incompressibility property, has two imaginary roots which makes the system elliptic according to the less stringent definition used by Briley & McDonald. Only if the pressure gradient in the $x$-momentum equation or the $u_2$ term in the continuity equation would somehow be known, $k_1$ would drop from this factor also; the system might then properly be called parabolic. This observation is important and will be referred to at several later stages.

The positive definite character of the last factor of (2.5) and (2.6) is of great significance for the discretisation of the equations also. We shall come back to this matter in Chapter 5, where we try to establish a discrete analogue of the elliptic behaviour by appropriately modelling the pressure gradient terms and the mass conservation equation.

It is not difficult to give a physical interpretation of the above analysis. In a flow governed by the Navier-Stokes equations, the influence of disturbances is carried instantly in all directions through diffusion; with the speed of sound in any direction through compressibility; and with finite local velocity in the flow direction through convection. The combination of these effects makes that a disturbance is felt immediately everywhere, which is typical for an elliptic set of equations. In a flow with a predominant velocity component in the positive $x$-direction, the propagation of influences via convection is unidirectional in $x$. Moreover, by deleting the terms responsible for diffusion in $x$-direction, one eliminates the source of instantaneous transfer of disturbances along the $x$-axis. But sound waves are still there to carry information everywhere. If the flow is locally supersonic, convection prevents the upstream travelling of sound waves, and there is no mechanism left for transmitting information in the negative $x$-direction. The set of equations is then parabolic: the flow can be solved as an initial value problem with $x$ as a time-like coordinate. Evidently, this fortunate circumstance ceases to exist in subsonic flow, and, a fortiori, in an incompressible flow, where the sound speed is infinite. Although convection carries information only in positive $x$-direction, the incompressibility allows information to be transmitted against the flow direction. The ellipticity is not destroyed by deleting some diffusion terms.

We emphasize that the characterization of a set of equations may depend on the local flow conditions; it may change from place to place or from flow to flow. This explains why the omission of the main-stream diffusion terms can have different consequences in subsonic and supersonic (or transonic) flows.

From a computational point of view, parabolic equations are more attractive than elliptic ones, because the computation may be carried out step-by-step in the parabolic (time-like) direction. It is not surprising therefore that attempts have been made to identify flows that admit a description by a parabolic equation set. We will now, in retrospect, recall some of these attempts, both as a piece of history
connected with our mathematical model and as a clarification of terminology.

The term 'parabolic flows' has its origin at Imperial College, London, and appears for the first time in the open literature in Patankar and Spalding (1972)\(^1\). These authors characterise flows as parabolic when

- there exists a predominant flow direction;
- the diffusion of momentum is negligible in that direction;
- the downstream pressure field has negligible influence on upstream flow conditions.

They point out that such flows can be computed in a downstream marching process (as in the solution of the boundary-layer equations) if a dual pressure definition is adopted: there is a specified pressure for the pressure gradient in the dominant flow direction while another pressure, obtained as a part of the solution, is used for the evaluation of the lateral pressure gradients. This is concordant with our analysis: when \( p_z \) is 'known', \( k_1 \) disappears from the principal symbol in (2.6). Notice, by the way, the difference of this approach with the interacting boundary-layer theory mentioned earlier, where the pressure gradient transverse to the primary flow direction is fixed while the mainstream pressure gradient is part of the solution.

In later publications, Patankar and Spalding abandon the dual pressure definition, and update the pressure in a multiple sweep marching process. Thereby pressure influences may be propagated upstream. In other words, the third condition above is removed at the cost of an extra iteration process; for those cases, the term 'partially parabolic flows' is used, which is misleading in our opinion.

A seemingly independent development took place at the same time in the United States. The origin of the term 'parabolised Navier-Stokes' is probably to be found there, although it is not quite clear who actually introduced it. S.G. Rubin, a prominent investigator in the field, uses the name for the first time in 1980 (Rubin & Lin, 1980; Davis & Rubin, 1980). He claims, though, that the first application of the 'parabolised' equations goes back to 1968 (Rudman & Rubin, 1968). Again, these first applications were based on truly parabolic equation sets, and the name was quite appropriate. Also here the generalisation to full pressure interaction soon followed, but without a change of the name of the mathematical model. Having recognised the inadequate denomination, Rubin (1984) later proposed to use the name 'reduced' Navier-Stokes when the full pressure/velocity interaction is retained. Some might consider the term 'thin-layer' Navier-Stokes more appropriate as conveying better something of the restriction in the applicability, but it is usually attached to the equations omitting diffusion in two directions. Anyway, a clear distinction between truly 'parabolised' and 'reduced' is made in a recent review paper (Rubin & Tannehill, 1992), which is an important reference in the present context.

\(^1\)It may be noted however that in Patankar and Spalding (1972) reference is made to a preceding internal report by Carretto et al., appearing later under the same title as Carretto, Curr and Spalding (1972), in which the term is also used.
Chapter 2: Mathematical model

The above account will have made clear that serious attempts have been made to exploit the parabolisation of the Navier-Stokes equations, but that the number of useful applications is limited to just a few: fixing the main-stream pressure gradient in advance is too restrictive in most cases. By just dropping the mainstream diffusion terms, the Navier-Stokes equations do not become parabolic, unless the flow is supersonic. The significance of the deletion of those terms is almost completely lost in a subsonic flow; the change of a boundary-value to an initial-value problem is not brought about. On the other hand, there is little use in maintaining the terms in flows with a predominant flow direction, because it will not improve the results to any significant extent. Except for the boundary condition requirements (see below), the choice between inclusion or deletion is largely immaterial. We have chosen to delete them in our mathematical model.

2.2 Turbulence model

To close the set of equations presented in the preceding section, a turbulence model is needed to relate the Reynolds stresses to the mean flow field and possibly to some known geometrical entities. A variety of models has been reported in the literature, and their performance in external flows has been reviewed or compared by Marvin (1983), Lakshminarayana (1986), Menter (1992) and Bettelini & Fannelløp (1993), to name a few. A model that produces satisfactory results in all flow situations has not emerged. The current feeling among fluid dynamicists is that such a universal model is not likely to become available. Instead, attempts are made to establish a range of comparatively simple turbulence models, each tuned as well as possible to a certain class of flows. In most solution methods for the (slightly reduced) Navier-Stokes equations, only comparatively simple models are currently used, usually not exceeding the level of so-called two-equation models. We support this pragmatic attitude, and have implemented a set of turbulence models; the one that seems best suited for a particular application can then be selected. In this study we will operate with only two of them, an algebraic and a one-equation model.

Like most others, our turbulence models are based on the fundamental assumption that the Reynolds stresses (divided by $\rho$) can be described by a Newtonian constitutive equation of the form

$$\overline{u_i' u_j'} = \frac{2}{3} k \delta_{ij} - \nu_t (u_{i,j} + u_{j,i}).$$

(2.7)

This is a generalisation of the Boussinesq hypothesis, in which $k$ is the turbulence kinetic energy, $\delta_{ij}$ is the Kronecker delta, and $\nu_t$ is the (isotropic) eddy viscosity, a scalar quantity that - in contrast to the molecular viscosity - may vary spatially. The eddy viscosity is derived from either of the turbulence models, which will be described below. The first term on the right-hand side of (2.7) is not evaluated explicitly. Instead, the definition of the pressure is modified to include $\frac{2}{3} k$. 
2.2.1 Algebraic model

The algebraic (or zero-equation) model is a generalisation of a turbulence model due to Cebeci & Smith (1974), used successfully in thin-boundary-layer calculations. It relates the eddy viscosity to the mean flow field and to certain geometrical quantities.

The Cebeci-Smith model is a two-layer model which can be written as

\[ \nu_t = \min\{\nu_{ti}, \nu_{to}\} \]
\[ \nu_{ti} = \ell^2 \sqrt{u_y^2 + w_y^2} \]
\[ \nu_{to} = \alpha \gamma \int_0^\delta (q_e - q) dy \]

in which

\[ \ell = 0.40y[1 - \exp(-y_+/A_+)] \]
\[ y_+ = y u_*/\nu \quad u_* = \sqrt{\tau_w/\rho} \quad A_+ = 26 \]
\[ \alpha = 0.0168 \]
\[ \gamma = [1 + 5.5(y/\delta)^6]^{-1} \]
\[ q = \sqrt{u^2 + w^2} \]

\( q_e \) being the value of \( q \) at the edge of the boundary layer with thickness \( \delta \), and \( \tau_w \) being the wall shear stress. Sometimes a pressure gradient correction is applied to \( A_+ \).

This model cannot be applied directly in our case. Postponing a reformulation in terms of curvilinear coordinates until Chapter 4, we still have to remove at least three difficulties that are encountered on employing it in a Navier-Stokes code: (i) the determination of the outer layer eddy viscosity may be difficult, because the location of the edge of the viscous layer is not well-defined, (ii) it becomes unrealistic near a point of vanishing skin friction, because the van Driest damping function makes the mixing length \( \ell = 0 \) throughout the boundary layer, and (iii) it is not suitable for application in a wake.

Baldwin & Lomax (1978) have endeavoured to resolve the first difficulty by adopting the maximum of the function

\[ F = y|\vec{\omega}| \]

(where \( \vec{\omega} \) is the vorticity vector) as the velocity scale of \( \nu_t \) and the distance between the wall and the point where this maximum occurs as the length scale. For this model to give a consistent value of \( \nu_{to} \), however, the functional dependence of \( y|\vec{\omega}| \) on \( y \) ought to display a distinct single peak. Often as this happens to be the case, there are circumstances in which two peaks develop, causing an ambiguity in the value of the outer layer eddy viscosity (e.g. Visbal & Knight, 1984).
In the flows we are considering, there is another way to find the edge of the viscous layer as required by the Čebeci & Smith model. Admittedly, the velocity profile gives no clear indication where the edge of the boundary layer is, but the total head may take over that role. This scalar variable, defined as

\[ H = p + \frac{1}{2} \rho (u^2 + v^2 + w^2), \]

is uniformly equal to \( H_\infty \) in the irrotational outer flow, where

\[ H_\infty = p_\infty + \frac{1}{2} \rho u_\infty^2. \]

A total head loss occurs only where the flow is significantly affected by viscosity. So the edge of the viscous layer, which in boundary-layer calculations may be chosen at the location where the velocity is, say, 99 percent of its external value, can now be chosen where \( H = 0.99 H_\infty \). The integral in the expression for \( \nu_{to} \) should nevertheless be rather insensitive to a slight misplacement of the boundary-layer edge. This is achieved by replacing \( q_\tau \) by the \( y \)-dependent function

\[ \sqrt{\frac{2}{\rho} (H_\infty - p) - v^2} \]

which is nominally equal to \( q \) outside the viscous layer. Thus the outer layer eddy viscosity is determined from

\[ \nu_{to} = \alpha \gamma \int_0^\delta \left( \sqrt{\frac{2}{\rho} (H_\infty - p) - v^2 - q} \right) dy. \]

Ideally, the integrand vanishes outside the viscous layer, and the integration might be extended to the boundary of the computation domain, making the determination of \( \delta \) superfluous. But to avoid a contamination of the result by small numerical errors, \( \delta \) is retained as the integration limit.

The difficulty (ii), mentioned above, is related to the use of \( u_\tau \) in the van Driest damping function. When \( u_\tau \) vanishes, the eddy viscosity is identically zero from wall to boundary-layer edge. Apart from the fact that this is unrealistic, experience has shown that it gives trouble to the extent that a steady state solution cannot be obtained near a separation or reattachment point due to the extreme sensitivity of the eddy viscosity to small changes in the velocity near the wall. The usual remedy is to replace \( u_\tau \) in the damping function by either the local shear velocity or its maximum value across the layer. This course is followed by us too. Its implementation will be described more precisely below.

The unsuitability of the Čebeci-Smith model for wakes is not removed by just omitting the van Driest damping term in the expression for the mixing length. The eddy viscosity still vanishes then on the wake symmetry plane, as it does on a solid
Numerical simulation of ship stern flows

wall. This is unrealistic, and allowance must therefore be made for the gradual growth of the eddy viscosity value along the wake centreline. This may be done via a three-layer model: a central wake layer with a \( y \)-independent eddy viscosity, embedded in the two-layer model for wall-bounded shear flows.

Bogucz & Walker (1988) have shown that behind a flow-aligned flat plate the eddy viscosity on the wake centreline increases (to leading order) linearly with the distance to the trailing edge and may be written as

\[
\nu_{te} = c(C_f)_{te}(x - x_{te})u_\infty
\]

where \( c \) is a constant, \( C_f \) is the friction coefficient while the subscript \( te \) refers to the trailing edge of the plate. For a flat plate flow the skin friction coefficient at the trailing edge may be approximated as

\[
(C_f)_{te} \approx \frac{0.06}{(10\log Re - 2)^2}.
\]

We use this relation to get rid of the impractical dependence of the eddy viscosity on the wake centreline upon the local flow behaviour at the trailing edge.

A remark can further be made about the value of the Clauser constant \( \alpha \) in the wake. Measurements indicate that it should assume a much greater value, near 0.064, in the far wake. However, since we do not reach the stage of asymptotic behaviour in the restricted length of the computation domain to be chosen in our applications, we have abstained from a gradual adjustment as suggested by Cebeci et al. (1986).

The algebraic turbulence model to be adopted in our method can then be summarised as

\[
\nu_t = \min\{\max\{\nu_{ti}, \nu_{tc}\}, \nu_{to}\}
\]

\[
\nu_{ti} = \ell^2 \sqrt{u_y'^2 + \frac{w_y'^2}{\delta}}
\]

\[
\nu_{to} = \alpha \gamma \int_0^\delta \left( \frac{2}{\rho} \left( H_\infty - p \right) - \nu^2 - q \right) dy
\]

\[
\nu_{te} = \begin{cases} 
0 & (x \leq x_{te}) \\
\frac{0.004}{(10\log Re - 2)^2} (x - x_{te})u_\infty & (x > x_{te})
\end{cases}
\]

in which

\[
\ell = 0.40y[1 - \exp(-y_+/A_+)]
\]

\[
y_+ = y u_*/\nu ; \quad u_* = \sqrt{\tau_*/\rho} ; \quad A_+ = 26
\]

\[
\alpha = 0.0168 ; \quad \gamma = [1 + 5.5(y/\delta)^6]^{-1} ; \quad q = \sqrt{u^2 + w^2}
\]

and \( \tau_* \) is the maximum shear stress between the wall and the point of evaluation (i.e. \( \tau_* \) cannot decrease with increasing \( y \)).
2.2.2 One-equation model

Traditional one-equation models solve an equation for the turbulence kinetic energy $k$, which has to be supplemented with an algebraic length scale to allow $\nu_t$ to be determined. Modern one-equation models, like those of Baldwin & Barth (1990), Spalart & Allmaras (1994) and Menter (1997) are based on a transport equation for $\nu_t$ itself or a related quantity.

We use here Menter's model which has been derived from the $k - \varepsilon$ model and the additional relation
\[
\left| -\overline{u'v'} \right| = \bar{\nu}_t \left| \frac{\partial u}{\partial y} \right| = a_1 k,
\] (2.8)
with the constant $a_1$ taken as $a_1 = 0.3$, while the tilde is used to indicate that low Reynolds effects (wall damping) are not included. In the $k - \varepsilon$ model the eddy viscosity is determined from
\[
\tilde{\nu}_t = c_\mu k^2 / \varepsilon,
\] (2.9)
where the constant $c_\mu$ is usually chosen as $c_\mu = 0.09$. Taking the total derivative of this equation, we get
\[
\frac{D\tilde{\nu}_t}{Dt} = c_\mu \left( \frac{2}{\varepsilon} \frac{Dk}{Dt} - \frac{k^2}{\varepsilon^2} \frac{D\varepsilon}{Dt} \right).
\]
Substituting the standard equations for $k$ and $\varepsilon$ and eliminating $\varepsilon$ with (2.9) and $k$ with (2.8), the high Reynolds form of the one-equation model results as
\[
\frac{D\tilde{\nu}_t}{Dt} = c_1 \tilde{\nu}_t \left| \frac{\partial u}{\partial y} \right| - c_2 \left( \frac{\tilde{\nu}_t}{L_{vK}} \right)^2 + \frac{\partial}{\partial y} \left( \tilde{\nu}_t \frac{\partial \tilde{\nu}_t}{\partial y} \right),
\]
with the constants
\[
c_1 = 0.144, \quad c_2 = 1.86, \quad \sigma = 1,
\]
and the von Karman length scale $L_{vK}$ defined by
\[
L_{vK} = \left| \frac{\partial u}{\partial y} \right| \left| \frac{\partial u}{\partial y} \right|.
\]
To avoid that $L_{vK}$ goes to zero, causing the destruction term to become unbounded, it is limited. Instead of
\[
D_{k-\varepsilon} = c_2 \tilde{\nu}_t^2 L_{vK}^{-2},
\]
the destruction term is written as
\[
D_m = D_{BB} \tanh \left( \frac{D_{k-\varepsilon}}{D_{BB}} \right),
\]
with
\[
D_{BB} = \frac{7}{8} \frac{\partial \tilde{\nu}_t}{\partial y} \frac{\partial \tilde{\nu}_t}{\partial y}.
\]
Moreover, for application of the model to wake flows, $L_{K_1}$ is forced to be greater than a value increasing linearly with the distance behind the trailing edge.

To make the model applicable to near-wall flows, two damping terms are introduced

$$D_1 = \frac{\nu t + \nu}{\nu t + \nu} \quad \text{and} \quad D_2 = 1 - e^{-(\nu t / A_x \kappa \nu)^2},$$

to let the final form of the one-equation model appear as

$$\left\{ \begin{array}{l}
\nu t = D_2 \tilde{\nu}_t \\
\frac{D \tilde{\nu}_t}{D t} = c_1 D_1 \tilde{\nu}_t \left[ \frac{\partial u}{\partial y} \right] - D_m + \frac{\partial}{\partial y} \left( \nu + \frac{\tilde{\nu}_t}{\sigma} \frac{\partial \tilde{\nu}_t}{\partial y} \right)
\end{array} \right. \quad (2.10)$$

### 2.2.3 Correction for longitudinal vortices

An important goal of numerical ship stern flow simulation is to provide propeller designers with accurate inflow data. This means that the prediction of the flow in the propeller disk area is of special importance. The results of two earlier-mentioned workshops (Larsson et al., 1991 and Kodama et al., 1994) have clearly revealed, however, that numerical predictions have considerable difficulty to match the experimental data in the near wake of ship hulls, generating longitudinal vortices embedded in the boundary layer. While experimental information shows a reduction of the longitudinal velocity component near the vortex core, numerical simulations using standard turbulence models fail to reproduce this feature correctly. Also, the predicted vortices are usually weaker than measurements indicate.

There is an inviscid mechanism to produce the observed reduction of the longitudinal velocity. The phenomenon, occurring in the development of a swirling flow in an adverse pressure gradient, has been described and analysed by Batchelor (1967, chapter 7.5). When the fluid surrounding the vortex decelerates, the vortex radius increases in downstream direction so that centrifugal action diminishes at the same time, which gives rise to an additional positive pressure gradient in the core region of the vortex. Thus the flow deceleration inside the vortex is stronger than in the surrounding fluid.

But since this mechanism is evidently included in the mathematical models on which the computation methods used in the workshops are based, the effects apparently are either too weak or annihilated by other influences, such as an excess of diffusion. Since the appearance of a study by Deng et al. (1993), who investigated the influence of discretisation schemes, grid density and ad hoc local modification of the eddy viscosity level and found that especially the latter gave significant improvement of the results, the inadequacy of the turbulence model is now commonly held responsible for the shortcoming in the numerical predictions.

A well-known feature of turbulent flows with pronounced longitudinal vortices is the damping of turbulence near the core of the vortex. The Raleigh criterion says that axisymmetric rotating flows are unstable when angular momentum decreases
radially away from the axis. Conversely, with angular momentum increasing outwardly a stabilizing effect occurs. This stabilizing effect suppresses the radial mixing and thus the Reynolds stress. In turbulence models based on the eddy viscosity concept the effect of the Reynolds stress reduction must be obtained via a reduction of the eddy viscosity. With most common turbulence models the required reduction is not predicted. The computed flow field is therefore too diffusive, and the vortices appear as too weak.

Since most modern single-screw ships are designed to produce a vortex pair passing through the propeller disk (in order to reduce circumferential velocity variations and thus the angle-of-attack variation of the propeller blade sections), a better prediction capability is strongly desired. Not surprisingly, therefore, several attempts to improve the turbulence modelling for ship stern flow prediction have been reported since the workshops. These attempts are focussed either on a direct modelling of the Reynolds stress transport equations or on adding an extra correction to simpler models. Sotiropoulos & Patel (1995) show that second-moment Reynolds-stress modelling gives improvement. This is confirmed by Deng & Visonneau (1997), although they report an over-estimation of the effects and prefer the two-equation $k - \omega$ model as giving results in better agreement with experimental information. They also have discussed the extra computational effort and the reduced robustness inherent in the use of second-moment closure, which has been the main incentive for others to devise ad hoc corrections to numerically safer turbulence models (e.g. Tahara & Himeno, 1996; Kodama, 1998). We shall also follow the second course and make an empirical correction in the determination of the eddy viscosity, which is effective only in the presence of longitudinal vortices in ship wakes. The details of the correction will be given in Chapter 4.

### 2.3 Boundary conditions

The set of partial differential equations used to describe the fluid motion must be supplemented with boundary conditions so as to guarantee a unique solution. We recall that in our case the computation domain is just a small part of the fluid domain. Anticipating on the following chapter, we assume that it can be cast in a hexahedral shape. In dealing with boundary conditions, it is then convenient to subdivide the boundary of the computation domain in accordance with its box-like shape (Fig. 2.1) into six parts: an inlet plane at $x = x_{in}$ and an outlet plane at $x = x_{out}$; the internal boundary $y = y_b$ with a solid wall part and a symmetry plane part and the plane $y = y_e$, called the external boundary, bordering on the inviscid flow region; the lateral boundaries $z = z_l$ and $z = z_r$, both being symmetry planes. The terminology for the $x = \text{constant}$ planes is intentional: the flow is assumed neither to leave the domain through $x = x_{in}$ nor to enter it through $x = x_{out}$.

The Navier-Stokes equations for incompressible flow are well known to require three boundary conditions on all boundaries because, as we have seen, they consti-
Figure 2.1: Schematic computation domain

tute a $6^{th}$ order elliptic system (cf. e.g. Brandt & Dinar, 1979; or Strikwerda, 1984). When Dirichlet conditions for the three velocity components are given, they must satisfy an integral constraint to be consistent with the mass conservation property; moreover, the pressure is determined only up to a constant. For external flows it is more natural to replace the Dirichlet condition for the velocity component normal to the domain boundary by a condition for the pressure on the external boundary. The integral constraint as well as the freedom in the pressure level then disappear.

Our mathematical model has Navier-Stokes character in $y$ and $z$ direction. Therefore three boundary conditions must be given on the internal, external and lateral boundaries of the domain. In the $x$-direction diffusion terms have been deleted and an Euler equations character results. The Euler equations for incompressible fluid flows require three boundary conditions at inflow and only one boundary condition at outflow (Oliger & Sundström, 1978).

Once the required number of boundary conditions is settled, we have to decide on the type of conditions. This decision is rather obvious for the natural boundaries of the domain: the hull surface and the symmetry planes. On the solid-wall part of the
Chapter 2: Mathematical model

internal boundary we specify Dirichlet conditions for the three velocity components, as dictated by the no-slip and impermeability conditions. On its remaining part and on the lateral boundaries, symmetry properties are used to yield a Dirichlet condition for the velocity component normal to the boundary and a Neumann condition for both tangential velocity components.

On the artificial boundaries the choice is less evident. For the one condition on the outlet boundary we take a condition for the pressure. An exact condition is not available for a computation domain of finite extent. But the ship-created pressure disturbances will diminish with increasing distance behind the ship. This means that the location of the outlet boundary determines for a great deal the quality of the boundary condition. We take as the boundary condition that the gradient of the pressure in longitudinal direction vanishes. This Neumann type of condition is preferable because it is less restrictive than the Dirichlet condition \( p = 0 \) by allowing some pressure variation over the outlet plane, and it is not conflicting with the condition imposed on the external boundary.

Also the external boundary is an artificial boundary. We impose there Dirichlet conditions for the pressure and for the velocity components tangential to the boundary. Again, exact values for these quantities are not available and what is actually imposed is directly related to the question how the solution in the stern-flow-and-wake zone is linked to those in the other zones (cf. Fig. 1.1). For there are various levels of sophistication in this coupling process. The simplest approach is that in which the solution is obtained in a one-cycle sequence of computations in the three zones; the potential flow zone overlaps the other two in that case. It may be described in three stages as follows:

1) The potential flow around the given hull shape is calculated as if no boundary layer were present; the resulting pressure (or tangential velocity distribution) on the hull is used as a boundary condition for subsequent boundary-layer calculations. This is the classical first-order matching procedure. It follows that any interaction between the two zones is neglected.

2) The boundary layer over the forward part of the ship is computed. In accordance with the parabolic nature of the boundary-layer equations, which implies no upstream influence of downstream occurrences, this computation can be carried out without regard of the flow behaviour in the stern-flow-and-wake zone. The coupling between the boundary-layer and stern-flow-and-wake zones is non-interactive.

3) As a third stage, the flow in the stern region is calculated. Inflow boundary conditions are obtained from the results of step 2). Information about the velocity and pressure on the external boundary of the stern-flow-and-wake zone is derived from step 1). This completes the numerical flow simulation.

The boundary conditions used in step 3) of the above one-cycle procedure on the external boundary are evidently not exact: being derived from step 1), they do not include the displacement effect of the boundary layer and the wake. The greater the lateral extent of the stern-flow-and-wake zone, the more satisfactory the boundary conditions, but also the less efficient the stern flow calculations will be.
At a more sophisticated level, interaction between the flow zones is taken into account. In that case, a new sequence of calculations is started once the above three-step procedure is completed. First, the external flow field is updated by a new potential flow calculation. The source panels are not distributed over the bare hull this time, but over the boundary of the union of hull and stern-flow-and-wake zone. On the hull surface a transpiration velocity may be prescribed to represent the displacement effect of the boundary layer, while the normal velocity on the external boundary of the stern-flow-and-wake zone obtained from the viscous flow calculations of the first sequence serves as a boundary condition there; on the inlet and outlet boundaries of the stern-flow-and-wake zone the through-flow is left free which implies a zero singularity strength. With the boundary conditions derived from the improved representation of the external flow, the boundary layer on the bow and the flow in the stern-flow-and-wake zone are recomputed. The process may be repeated until the solutions in the three zones match well enough.

Instead of this iterative approach, there might yet be another possibility. It is true that we cannot give a priori exact conditions for the tangential velocity components and the pressure on the external boundary. But the external flow is known to have the properties of irrotationality and uniform total head. Thus, by specifying the tangential vorticity components to vanish and the pressure to satisfy the Bernoulli equation, we have three exact conditions. Of course these conditions require that the outer flow is fully determined by the hull form and the displacement effects of the boundary layer and the wake; other bodies or near-by walls must be absent. Moreover, the conditions do not convey information about the flow velocity at infinity. That information must come from the velocity data specified on the inflow boundary, so as to make sure that we obtain the solution at the correct Reynolds number. Whether it is sufficient for a unique solution is still to be proven.

In the applications presented in this thesis the first option (one cycle process) is used. An application of the third option is included in Hoekstra (1989).

If the one-equation turbulence model is used, suitable values for the eddy viscosity (derived with the aid of the algebraic model) are imposed on the inlet boundary, while \( \nu_t \) is set to zero on a no-slip boundary as well as on the external boundary. On symmetry boundaries a symmetry condition is prescribed. Since the diffusion terms in the eddy-viscosity transport equation are treated in the same way as in the momentum equation, a boundary condition for \( \nu_t \) at the outlet is not needed.
Chapter 3

Computational grid

Reference is made to Figure 1.1 for a 2D sketch of the top view of the calculation domain (zone III). For the purpose of a numerical discretisation of the equations, a set of points has to be identified, distributed orderly over this domain and connected by straight or curved line elements, to form the computational grid. One might consider to set up a rectilinear grid in physical space directly. But proper resolution of steep velocity gradients near the hull surface, easy implementation of boundary conditions and convenience in programming are needs which make the use of a boundary-fitted grid preferable. An extra argument in favour of such a grid in the particular application that we have in mind is our wish to have one family of coordinate lines more or less aligned with the predominant flow direction.

Boundary-fitted grids can be either structured or unstructured. The unstructured option offers more flexibility, but on the other hand causes the data management to be quite cumbersome and the most efficient numerical solution techniques are not suited for or have a strongly degraded performance on unstructured grids. Although considerable progress has been made in the use of unstructured grids, in our kind of application the structured grid is still the common choice. Also in this work a structured grid will be used.

To that end we introduce, in addition to the Cartesian \((x, y, z)\) coordinate system, a curvilinear \((\xi, \eta, \zeta)\) system, conforming to the boundaries of the domain. Once the transformation relations between \(x, y, z\) and \(\xi, \eta, \zeta\) have been established, a suitable rectangular mesh in \((\xi, \eta, \zeta)\) space defines the computational grid in physical space. The choice of the transformation relations should first of all be governed by the requirement of proper resolution of the flow phenomena to be modelled. Besides, desirable features of the grid in physical space are: conceptual simplicity, orthogonality, regularity (e.g. smoothly varying grid cell volumes) and controllability (easy adjustment of grid node distribution). Grid generation amounts to finding a good compromise between these often conflicting requirements.

Because we are going to operate on the curvilinear \(\xi, \eta, \zeta\) system, it is convenient to switch at this point to general tensor notation which, in contrast to Cartesian tensor formalisms, allows two types of behaviour under transformation. A set of unit
base vectors $\hat{e}_i$ ($i = 1, 2, 3$) is defined in relation to the $x, y, z$ coordinate system. Similarly a set of grid-line oriented base vectors $\vec{a}_i$ for the $\xi, \eta, \zeta$ system is introduced with
\[ \vec{a}_i = \frac{\partial x^j}{\partial \xi^i} \hat{e}_j, \]
while the reciprocal base vectors are given by
\[ \vec{a}^i = \frac{\vec{a}_j \times \vec{a}_k}{\vec{a}_i \cdot (\vec{a}_j \times \vec{a}_k)}, \]
the values of $j$ and $k$ being such that $i, j, k$ is an even permutation of $1, 2, 3$. Further, we use interchangeably $\xi^1, \xi^2, \xi^3$ for $\xi, \eta, \zeta$ and $x^1, x^2, x^3$ for $x, y, z$. Suffix notation will be employed as well: when $\xi, \eta$ or $\zeta$ appear as a suffix, a partial derivative is implied.

The transformation between the $x, y, z$ and $\xi, \eta, \zeta$ systems is determined by the transformation matrix
\[ \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} \text{ or } \frac{\partial x^i}{\partial \xi^j}. \]
The determinant of this matrix is called the Jacobian, $J$. It can be interpreted as the ratio of grid-cell volumes in physical and computational domains. It should not vanish for non-singular transformations, and is positive for right-handed coordinate systems, as adopted here.

From the components of the transformation matrix the metric tensor $g_{ij}$, defined as
\[ g_{ij} = \vec{a}_i \cdot \vec{a}_j = \sum_{k=1}^{3} \frac{\partial x^k}{\partial \xi^i} \frac{\partial x^k}{\partial \xi^j}, \]
can be determined. The conjugate metric tensor
\[ g^{ij} = \vec{a}^i \cdot \vec{a}^j \]
is related to $g_{ij}$ by $g^{ij} = G^{ij}/g$ where $G^{ij}$ is the cofactor of the element $g_{ij}$ in the determinant $g$. Further we have
\[ g_{ij} = \vec{a}^i \cdot \vec{a}_j = \delta^i_j, \]
where $\delta^i_j$ is the Kronecker delta, which equals 1 if $i = j$ and 0 otherwise. It may be shown that $g = J^2$, so that we will write $\sqrt{g}$ instead of $J$, as is customary in tensor analysis.

The spatial variation of the base vectors can be expressed with the aid of the well-known Christoffel symbols:
\[ \frac{\partial \vec{a}_i}{\partial \xi^j} = \Gamma_{ijk} \vec{a}^k \]
and
\[ \frac{\partial \vec{a}^i}{\partial \xi^j} = -\Gamma_{ijk} \vec{a}^k; \quad \frac{\partial \vec{a}_i}{\partial \xi^j} = \Gamma_{ij} \vec{a}_k. \]
3.1 Grid lay-out

Ultimately, when the appendages of a ship – varying from rudders and stabilizers to hull-protruding shaft bossings, struts or ducts – are to be included, either a decomposition of the calculation domain into several subregions or the use of an unstructured mesh will be unavoidable. In this thesis, however, we shall content ourselves with a single-block regular grid.

![Diagram](image)

Figure 3.1: Natural borders of computation domain

Taking advantage of the symmetry of the flow with respect to both the longitudinal centreplane of the hull and the undisturbed free surface, and selecting a suitable inlet plane, we arrive at a situation as sketched in Figure 3.1. The natural borders of the computation domain, *i.e.* the hull surface and the symmetry planes \( y = 0 \) and \( z = 0 \), are shown together with the chosen inlet plane \( x = 0 \); a decision on the closure of the domain has yet to be taken. This decision is largely based on the type of grid that is chosen. In the \( x - y \) plane at \( z = 0 \), either an H or an O-type grid might be chosen. However, where we prefer to have one family of coordinate lines roughly aligned with the flow, the H-type grid is selected. In the \( y - z \) plane at \( x = 0 \), an O-type grid is the best choice. Since ships are usually flat-bottomed, an H-type grid would be objectionable.

Thus we arrive at an H-O grid as the best option, and can close the computation domain accordingly, as shown in Figure 3.2.
For the complete settlement of the global lay-out of the grid in real space, the matter of boundary correspondence requires further decisions. From Figure 3.2 can be inferred that the shape of a cross-section of the computation domain in physical space changes topologically from quadrangular to triangular with increasing \( x \), whereas the transformed domain exhibits a rectangular shape throughout. So the natural boundary correspondence of the cross-sections in transformed and physical space in the upstream part of the calculation domain gets lost behind the hull. The question is then how the four corner points of a cross-section of the transformed space are to be mapped onto the physical space.

If the hull would resemble an axisymmetric body, the coordinate surface \( \eta = 0 \) — partly representing the hull surface — would naturally shrink to a single line behind the hull. Two of the four corner points in transformed space would then be mapped on a single corner point in physical space. However, most ship sterns are far from axisymmetric because a propeller has to be fitted at sufficient submergence. Hence, the shrinking of the \( \eta = 0 \) surface to a line is not suitable.

The decision on an alternative continuation of the \( \eta = 0 \) surface depends on the shape of the aft body and the stern contour. For single-screw ships, it can usually be extended into the vertical symmetry plane. For its lower edge, i.e. the line \( \eta = \zeta = 0 \), a straight extension of the keel line is sometimes adopted (as in Figure 3.2), but in the case of a bulbous-type stern, for instance, a line extending from the propeller axis may be more appropriate. In both cases the coordinate surface \( \eta = 0 \) consists partly of a solid wall (hull surface) and partly of a free-slip
symmetry plane (upper slice of vertical symmetry plane boundary). For twin-screw ships, on the other hand, the \( \eta = 0 \) surface ends usually in the horizontal symmetry plane.

As an awkward consequence of the H-O grid, the stern contour does not normally coincide with a coordinate line. Such a coincidence is desirable to facilitate the resolution of the sudden change of no-slip solid wall to free-slip symmetry plane conditions. But all grids conforming to the stern contour that we have been able to conceive had to be rejected because of unacceptable non-orthogonality, \( \xi \)-lines not following the dominant flow direction, or other obvious shortcomings.

Because the cross-section of the physical domain is topologically not everywhere rectangular, we must be able to cope with grid singularities on the domain boundary. When the inner boundary of the cross-section either exhibits a slope discontinuity or does not meet the symmetry planes at right angles, a singularity occurs. Behind the hull we encounter the most extreme case, viz. the mapping of a 90 degrees corner on one of 180 degrees. Hence, either on the line \( \eta = \zeta = 0 \) or the line \( \eta = 0; \zeta = \zeta_{\text{max}} \) or both the mapping is singular behind the ship.

Grid singularities may adversely affect the accuracy and the robustness of the iterative solution, particularly if the flow speed has a finite value at the singularity. Imagine, for example, that one operates with physical, gridline-oriented velocity components \( u_{\text{phys}}, v_{\text{phys}}, w_{\text{phys}} \) as the dependent variables. Suppose furthermore that there is a finite vertical velocity \( v_c \) along the vertical symmetry plane in the wake. Then \( v_{\text{phys}} = 0, w_{\text{phys}} = v_c \) above the wake singularity, while \( v_{\text{phys}} = -v_c, w_{\text{phys}} = 0 \) below the singularity. Hence, there is a conflict at the singularity because \( v_{\text{phys}} \) and \( w_{\text{phys}} \) cannot both be zero.

We may observe the following, however. Near the singularity in the wake, the cross-sectional grid (which is for convenience assumed to be in a \( x = \text{constant} \) plane) can locally be described in complex variable notation as

\[
y + i(z - z_s) = -i\alpha(R\eta + i\zeta)^2
\]

where \( y = 0, z = z_s \) is the location of the singularity and \( \alpha \) is a proportionality constant. The singularity is therefore said to be of the parabolic type. It follows that

\[
y = 2\alpha R\zeta; \quad z = 2\alpha R\eta; \quad z_s = -2\alpha R^2\eta; \quad z_s = 2\alpha \zeta.
\]

A covariant variable behaves regularly near such a singularity. Indeed, the transverse components of

\[
A_i = \frac{\partial f}{\partial \xi^i} = \frac{\partial f}{\partial x^j} \frac{\partial x^j}{\partial \xi^i}
\]

are

\[
A_2 = f_\eta = f_y y_\eta + f_z z_\eta \\
A_3 = f_\zeta = f_y y_\zeta + f_z z_\zeta
\]

which, with fixed values of \( f_y \) and \( f_z \), are seen to vanish at the singularity and to increase linearly away from it. Accurate discretisation of spatial derivatives is
therefore possible with covariant velocity components, and there is no conflict in the application of boundary conditions at the singularity.

Conversely, the transverse components of a contravariant variable

$$A^i = g^{ij} A_j$$

can be verified to tend to infinity near the singularity. So the use of contravariant velocity components near a parabolic singularity is likely to result in gross numerical inaccuracies. This singular behaviour may be removed, however, by multiplication of a contravariant variable with $\sqrt{g}$.

 Needless to say that with Cartesian velocity components as dependent variables, there are no ambiguities either.

### 3.2 Grid generation procedure

Having established the borders of the computation domain and the global layout of the grid, we are now in a position to discuss the procedure for generating a suitable grid.

We have basically two different methods for generating the grid. In the first and simplest method the 3D mesh is constructed by stacking a set of 2D grids, i.e. a number of transverse sections of the domain is chosen and in each section a 2D grid is established; by connecting points with equal $\eta, \zeta$ values, the $\xi$-lines are obtained. The second method is based on the solution of an elliptic set of equations and has more flexibility. Both methods lead to non-orthogonal grids.

The first method will be described with some detail, because it has been developed by the author. The second grid generation technique has been described in the open literature and a brief account suffices here.

#### 3.2.1 Stack of two-dimensional grids

The cross-section of the physical domain is of varying shape (Fig. 3.3). Alongside the hull it has a curved inner boundary (A-D, a frame line of the hull), a curved outer boundary (B-C, a line outside the shear-layer region, but conveniently chosen otherwise) and two straight boundaries located in symmetry planes (Fig. 3.3a,b). Behind the hull the inner boundary is straight (Fig. 3.3e). There is a transition region in between where the inner boundary is non-smooth (Fig. 3.3c,d).

The two-dimensional grid in such a cross-section is obtained by mapping a rectangular region onto the given shape of the cross-section. The mapping is obtained in a sequence of transformations. The basic transformation is a conformal mapping of a rectangle on the cross-section by a generalised Schwarz-Christoffel method (Hoeckstra, 1986). The original Schwarz-Christoffel method was conceived for the conformal mapping of regions with polygonal boundaries, but it can easily be generalised for application to smoothly curved boundaries. The resulting transformation
technique is particularly suited for domains with boundaries that have slope discontinuities (as in Fig. 3.3c,d,e).

If the S-C method is applied to annular domains, the transformation between the complex variables \( h = y + iz \) in the physical plane and \( t = \ln r + i\theta \) in the transformed plane is expressed as:

\[
\frac{dh}{dt} = A \exp \left\{ - \frac{1}{\pi} \int_{-1}^{+1} \frac{d\sigma_1}{ds'} \ln(s_1 - s') - \frac{d\sigma_2}{ds'} \ln(s_2 - s') ds' \right\}
\]  

(3.1)

under the constraints

\[
\int_{-1}^{+1} \frac{d\sigma_1}{ds'} ds' = 2\pi \quad \text{and} \quad \int_{-1}^{+1} \frac{d\sigma_2}{ds'} ds' = 0
\]

while

\[
\begin{align*}
s_1 &= \cosh k(\ln \frac{r}{r_1} + i\theta) \\
s_2 &= \cosh k(\ln \frac{r}{r_2} + i\theta) \\
s' &= \cos k\theta'
\end{align*}
\]
in which

\[ k = \begin{cases} 
1 & \text{for symmetrical domains}^1 \\
2 & \text{for doubly symmetrical domains}
\end{cases} \]

and the subscripts 1 and 2 refer to the inner and outer boundary respectively. The fact that two edges of the domain are located on a symmetry plane is implied in these formulae. See Hoekstra (1986) for the derivation of equation (3.1).

Being based on a boundary-integral formulation, the solution procedure does not require an initial guess of the mesh. It suffices to specify the position of a number of discrete points \( h_{n,m} (n = 1, 2, \ldots, N_m; m = 1, 2) \) on the inner \((m=1)\) and the outer \((m=2)\) contour of the physical domain, and to add for each point the local inclination angle of the contour. Possible step changes in the inclination angle at knuckle points have to be given as well. It may be worth noting that \( h_{n,m} \) are not the grid nodes.

The mapping procedure starts with a reasonable guess of the location of the points \( t_{n,m} \) in the \( t \)-plane corresponding with the points \( h_{n,m} \). On each interval \( t_{n-1,m} - t_{n,m}, \sigma_m \ (m=1,2) \) is written as a third order polynomial in \( s' \) so that the integral in equation (3.1) can be evaluated analytically and transformed to an algebraic sum with some unknown coefficients. Next, equation (3.1) is integrated along the inner and outer boundaries with guessed values of these coefficients. The resulting \( h \)-values are compared with the true ones and the right-hand side is improved accordingly. The process is repeated until the solution has converged to a desired accuracy.

The result is a known right-hand side of equation (3.1), which can be integrated along some rectangular mesh in the \( t \)-plane to form the grid.

The aspect ratio of the rectangle in the \( t \)-plane is a part of the solution and will generally vary from section to section. To obtain a box-like domain as shown on the right in Figure 3.2, a linear transformation on the real part of \( t \) is applied to yield a new coordinate \( \eta_0 \), related to \( R(t) = \ln r \) by:

\[ \eta_0 = (\ln r - \ln r_1)/(\ln r_2 - \ln r_1) \]

and having a range \([0,1]\) in all transverse sections. In the other direction a coordinate with the same range can be defined as

\[ \zeta_0 = k\theta/\pi. \]

The mesh in physical space corresponding with a uniform rectangular grid in \( \eta_0, \zeta_0 \) space will be called the basis grid. It is not yet suitable for our purposes because i) we need a much finer grid near the hull surface than further away and ii) some control over the distribution of grid nodes on the inner and outer boundaries is desirable.

---

\(^1\)This case is relevant for computation of the flow around submarines or ships with nonsymmetric sterns or when the rotation of the propeller slip stream is taken into account, implying a loss of port/starboard flow symmetry.
The first problem is resolved by stretching the grid in \( \eta_0 \)-direction. We use Vinokur's two-sided stretching function (Vinokur, 1983). It relates \( \eta_0 \) to a new variable \( \eta_1 \) by the following rules:

Specify

\[ s_0 = (d\eta_1 / d\eta_0)_{\eta_1 = \eta_0 = 0} \quad \text{and} \quad s_1 = (d\eta_1 / d\eta_0)_{\eta_1 = \eta_0 = 1} \]

and define

\[ A = \sqrt{s_0 / s_1}; \quad B = \sqrt{s_0 s_1}. \]

Then, supposing that \( B > 1 \), let

\[ \eta_0 = \frac{X}{A + (1 - A)X} \]

with

\[ X = \frac{1}{2} + \frac{\tanh | \Lambda(\eta_1 - \frac{1}{2}) |}{2 \tanh(\Lambda/2)} \]

\[ B = \frac{\sinh \Lambda}{\Lambda}. \]

Vinokur has given an approximate explicit expression for \( \Lambda \) as a function of \( B \) to replace the latter relation.

As to grid node distribution control in \( \zeta \)-direction, we point out first that the distribution of the basis-grid nodes on the boundaries is fixed once the right-hand side of equation (3.1) is established for the given shape of the inner and outer boundary. An increase of the number of grid points (i.e. a local refinement) is the only means by which a local refinement can be obtained. It is a price paid for the good regularity properties of the \( \eta_0, \zeta_0 \) mesh. But as has been pointed out, successive meshes have to be tied together to form \( \xi \)-lines. If the shape of inner and outer boundaries vary smoothly with \( \xi \), the \( \xi \)-lines will be smooth too. Near the stern, however, the change of the inner boundary shape may be rather drastic. Experience shows that undesirable ridges can occur in the \( \xi, \eta \)-coordinate surfaces, implying an unnecessarily severe non-orthogonality of the grid in \( \xi, \zeta \) surfaces. Therefore, we allow the additional transformation:

\[ \zeta_0 = \zeta_1 + c_1(x) \sin \zeta_1 \pi + c_2(x) \sin 2\zeta_1 \pi \]

where the functions \( c_1(x) \) and \( c_2(x) \) may be selected to improve the smoothness of the grid in \( \xi \)-direction.

Summarizing, we construct a basis grid by conformal mapping and subsequently stretch it in the two coordinate directions separately. Thereby the conformity is lost but the orthogonality is maintained. The strength of the proposed conformal mapping technique is its robustness and accuracy near mapping singularities on the boundary.
3.2.2 Elliptic PDE grid

Even after sacrificing conformity, the control of the grids produced by the method of the previous section is very limited. Typically, grid clustering is found in regions near body convexity, while the opposite holds for body concavity, which is not always what is desired. The procedure of stacking 2D grids is therefore often too restrictive to obtain a good-quality grid. In such cases recourse can be taken to a general 3D generator based on the solution of a set of elliptic partial differential equations. More precisely, the grid is determined by the Poisson equation:

$$\frac{\partial^2 \xi_i}{\partial x^2} + \frac{\partial^2 \xi_i}{\partial y^2} + \frac{\partial^2 \xi_i}{\partial z^2} = g^{ii} P_i \quad (i = 1, 2, 3),$$

in which $P_i$ are the so-called "control functions". Changing from $\xi_i$ to $x_i$ as dependent variables, the equation to be solved becomes:

$$g^{jk} \frac{\partial \xi_j}{\partial \xi_i} \frac{\partial x_i}{\partial \xi_k} + g^{ij} P_j \frac{\partial x_i}{\partial \xi_j} = 0 \quad (i = 1, 2, 3).$$

We refer to Thompson et al. (1985) for its derivation.

This Poisson equation can be solved with standard techniques, if either Dirichlet (fixed grid nodes) or Neumann (e.g. orthogonality requirement) boundary conditions are prescribed. In generating a 3D grid, first the 2D version of (3.3) or a 2D orthogonal grid generator is applied to obtain grids on the domain boundaries (block faces), upon which the 3D version of (3.3) is solved to generate the interior grid nodes. If the domain boundaries are curved surfaces, the parametric coordinates of the spline representation of those surfaces take the role of $x_i$ in (3.3).

The primary difficulty in generating grids with (3.3) is the specification of suitable values for the three forcing terms $P_j$, which control the grid line spacing and the grid orthogonality. To obtain proper grid line spacings and acceptable deviations from orthogonality, it is highly desirable that their determination is governed by the available information: the boundary shape and the boundary point distribution. On a boundary, equation (3.3) can be seen as an equation for $P_j$ rather than $x_i$. With fixed grid nodes on a domain boundary $\xi^k = constant$, the values of all three control functions $P_j$ on that boundary are completely determined once additional assumptions have been made about the derivatives of $x_i$ with respect to $\xi^k$. Since the control functions can essentially be chosen arbitrarily, the number of additional conditions that can be imposed equals the number of control functions, i.e. the number of space dimensions used in (3.3). We follow Sorensen (1986) and prescribe orthogonality and a specified distance between the boundary and the neighbouring grid line or surface. These fix the first derivatives, but leave the second derivatives with respect to $\xi^k$ free. Equation (3.3), applied at the boundary, is therefore conveniently split into a part fixed by the assumed orthogonality and the off-boundary spacing and another part dependent on the second derivatives. For the treatment of the first part we follow the suggestions of Thomas and Middlecoff (1980). Once
defined on the boundaries, its value in the interior is derived by linear interpolation. The second part is determined iteratively as proposed by Sorensen (1986), and is applied only in the near-boundary region. Details of the procedure have been reported by Eça (1994).

In some cases Dirichlet conditions are not imposed on all boundaries but instead on five boundary faces of the domain (three boundary edges in 2D) while a Neumann-type boundary condition is imposed on the remaining face (edge). Even if Neumann conditions are imposed, the Poisson equation is solved with Dirichlet conditions (guessed boundary points), but every iteration is followed by a movement of the grid points along the given boundary to satisfy the Neumann condition. The second part of the control functions is omitted if Neumann conditions are imposed.

If (3.3) is used for grid generation, the procedure is as follows. Given the hull geometry of the ship, we start with creating a spline representation of the $\eta = 0$ boundary, partly coinciding with the hull surface and partly with a symmetry plane. A grid is then generated on this curved surface by a 2D version of (3.3), using fixed grid node distributions on the edge curves and substituting the parametric coordinates of the surface for $x^i$ in (3.3). Subsequently, grids are generated on the boundaries $\xi = \xi_{\text{min}}, \xi = \xi_{\text{max}}, \zeta = \zeta_{\text{min}}, \zeta = \zeta_{\text{max}},$ again with Dirichlet conditions except on the $\eta = \eta_{\text{max}}$ edge curves, where a Neumann condition (orthogonality) is imposed. The grid on the $\eta = \eta_{\text{max}}$ boundary face can then be generated with fixed nodes on all boundaries. Finally the interior grid is generated, based on (3.3), using the iterative adjustment of the control functions.

As in the first grid generation technique, we do not attempt to generate the grid directly with the required clustering towards the $\eta = 0$ boundary. Instead, grid stretching towards the $\eta = 0$ surface is carried out algebraically as a postprocessing on the grid generated with (3.3), which is then allowed to be more regular and hence has smoother control functions.
Chapter 4

Equations in curvilinear coordinates

The transformation of the mathematical model of Chapter 2 to the curvilinear \( \xi, \eta, \zeta \) system, constructed in Chapter 3, is in principle straightforward. Yet, some hard decisions on the decomposition of the momentum and velocity vectors into three scalar quantities must be made, with far-reaching consequences for the coding of the solution method, while the impact on the final performance of the computation method is not obvious. Before going into the arguments which have guided us in these decisions, let us first deal with some matters of notation.

4.1 Notation

Let \( \vec{Q} \) be any vector quantity which in the Cartesian reference frame \( x, y, z \) has components \( q^i : \vec{Q} = q^i \vec{e}_i \). Using, as in the previous chapter, \( \vec{a}_i \) and \( \vec{a}^i \) as the covariant and contravariant base vectors of the \( \xi, \eta, \zeta \) system, the vector \( \vec{Q} \) can then also be written either as \( Q^i \vec{a}_i \) or as \( Q_i \vec{a}^i \), a distinction that is not relevant in rectangular Cartesian coordinates. Thus in relation to the \( \xi, \eta, \zeta \) system we distinguish the covariant and contravariant vector components \( Q_i \) and \( Q^i \) respectively. They are related by \( Q_i = g_{ij} Q^j \) or \( Q^i = g^{ij} Q_j \). The physical components of \( \vec{Q} \) along the coordinate lines, denoted by \( Q(i) \), are related to \( Q^i \) by

\[
Q(i) = Q^i |\vec{a}_i| = Q^i \sqrt{g_{ii}} \quad \text{(no sum on } i)\]

Notice that we use a lower case symbol for the Cartesian components of a vector and an upper case symbol for the covariant or contravariant components. So if we define a velocity vector \( \vec{U} \), its Cartesian components in the directions \( x^i \) are now denoted as \( u^i \), i.e. \( \vec{U} = u^i \vec{e}_i \), while the decomposition along the covariant or contravariant base vectors reads: \( \vec{U} = U^i \vec{a}_i = U_i \vec{a}^i \). Similarly, the following decompositions of the external force field \( \vec{F} \) apply: \( \vec{F} = f^i \vec{e}_i = F^i \vec{a}_i = F_i \vec{a}^i \).
The relation between the contravariant and the Cartesian vector components are easily established as:

\[ Q^i = \bar{a}^i \cdot \bar{Q} = \bar{a}^i \cdot (q^j \bar{e}_j) = q^j (\bar{a}^i \cdot \bar{e}_j) = q^j a_{ij} \]

and conversely

\[ q^i = \bar{e}^i \cdot \bar{Q} = \bar{e}^i \cdot (Q^j \bar{a}_j) = Q^j (\bar{e}^i \cdot \bar{a}_j) = Q^j a_{ij}^{(j)} \].

Individual components of base vectors have here been written as \( a_{ij}^{(j)} \) and \( \alpha_{ij} \).

For later reference we introduce also a scaled velocity vector

\[ \bar{V} = \sqrt{g} \bar{U}. \]

If in addition scaled basis vectors are defined as:

\[ \bar{b}_i = \frac{\bar{a}_i}{\sqrt{g}} \text{ and } \bar{b}^i = \sqrt{g} \bar{a}^i, \]

we have:

\[ \bar{U} = \bar{V}^i \bar{b}_i = \frac{V_i \bar{b}_i}{g}. \]

### 4.2 Decomposition of momentum equation

The general tensor formulation of the equations (2.1) and (2.2) under an isotropic eddy viscosity assumption for the Reynolds stresses is

\[ (T^{ij} a_i)_j = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi_j} \left( \sqrt{g} T^{ij} a_i \right) = F^i a_i \tag{4.1} \]

\[ U^i_{\cdot i} = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^i} \left( \sqrt{g} U^i \right) = 0 \tag{4.2} \]

in which

\[ T^{ij} = \rho U^i U^j + g^{ij} p - \tau^{ij}, \]

with the viscous stress tensor

\[ \tau^{ij} = 2(\mu + \mu_t) S^{ij} \]

\[ = 2\mu g^{ik} g^{jm} S_{km} \]

\[ = \mu g^{ik} g^{jm} (U_{k,m} + U_{m,k}) \]

\[ = \mu (g^{im} U^j_{\cdot m} + g^{ik} U^j_{\cdot k}) \]

which can be expanded to

\[ \tau^{ij} = \mu \left( g^{ik} \frac{\partial U^i}{\partial \xi^k} + g^{ik} \frac{\partial U^j}{\partial \xi^k} - U^k \frac{\partial g^{ij}}{\partial \xi^k} \right) \]
or to
\[ \tau^{ij} = \mu_e \left( g^{ik} a_i^{(n)} \frac{\partial U^m a_m^{(n)}}{\partial \xi^k} + g^{ik} a_i^{(n)} \frac{\partial U^m a_m^{(n)}}{\partial \xi^k} \right) \]
if use is made of the identity:
\[ U_j^i = d_j^i \frac{\partial U^i}{\partial \xi_j} = d_j^i \frac{\partial U^k a_k}{\partial \xi_j} = a_j^{(n)} \frac{\partial U^k a_k^{(n)}}{\partial \xi_j}. \]

The equivalent formulation of the momentum equation in vector notation is:
\[ \vec{\nabla} \left[ \rho \vec{U} \vec{U} + p \vec{I} - \mu_e \left( \vec{\nabla} \vec{U} + (\vec{\nabla} \vec{U})^T \right) \right] = \vec{F}. \]
in which \( \vec{I} \) is the identity tensor and the superscript \( ^T \) denotes the transpose operation.

Numerical solution requires three scalar components of the vector-momentum equation. In the literature appear primarily two options. The natural choice is to take the components along the coordinate lines, giving:
\[ \left[ \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} (\sqrt{g} T^{ij}) + T^{jk} \Gamma_{jk}^i \right] \vec{a}_i = F^i \vec{a}_i. \quad (4.3) \]
The other option is to expand \( \vec{a}_i \) into its \( x, y, z \) components, in other words, to split the momentum equation into its three Cartesian components. This yields:
\[ \left[ \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} \left( \sqrt{g} \frac{\partial x^i}{\partial \xi^j} T^{jk} \right) \right] \vec{c}_i = f^i \vec{c}_i. \quad (4.4) \]

The most apparent difference between the two formulations is that the left-hand side of (4.4) is in so-called strong conservation form, while in (4.3) ‘volume force’ terms appear, involving the Christoffel symbols. This is due to the fact that \( \vec{c}_i \) is spatially invariant, while \( \vec{a}_i \) is not.

It may be noted that the compact notation of (4.4) is misleading in so far as it suggests a smaller amount of work involved in the numerical evaluation. Suppose that numerical differentiation requires the evaluation of the term to be differentiated at two grid nodes, while for the ‘volume force’ terms evaluation at one node suffices. Notice further that \( T^{jk} \) and \( \Gamma_{jk} \) are symmetric. Then the left-hand side of (4.3) requires \( 2 \times 3 + 6 = 12 \) evaluations per scalar equation while the same part of (4.4) requires \( 2 \times 3 \times 3 = 18 \) evaluations. Of course, the exact number of basic operations (additions, multiplications and divisions) depends on the details of the discretisation, but in a global sense (4.3) is likely to require less work than (4.4).

But the strong conservation form is attractive in many other respects. It is for instance a necessary (though not sufficient) condition for conservative differencing, i.e. a way of differencing by which a discrete analogue of the divergence theorem is satisfied. And, where (4.1) is in divergence form, conservative differencing preserves
an integral property of the continuum equations, which is something to be pursued. There are also indications that the strong conservation form leads to more stable algorithms than other forms near grid singularities (Eriksson, 1984). The handling of interface conditions in multi-block grid applications is also facilitated. Furthermore it allows consistent drag calculation via a global momentum balance. So even if the strong conservation form is not essential for accuracy, there are several practical advantages.

The preference for the strong conservation form has many others (e.g. Rodi et al., 1989; Maliska & Raittby, 1984) led to the adoption of (4.4) as a basis for their solution procedure. Apparently an important third option is left out of consideration: instead of the spatially invariant \( \bar{e} \), a locally grid-aligned, yet spatially invariant, set of base vectors can be used to achieve a local decomposition of the momentum equation into grid-oriented scalar components, which maintains the strong conservation form. This is realised by expanding \( \bar{a}_i \) with respect to a local set of spatially invariant base vectors (Vinokur, 1974 and Rosenfeld et al., 1988). If the equations have to be evaluated near a point \( P \), an expansion of \( \bar{a}_i \) in the neighbourhood of \( P \) as

\[
\bar{a}_i = (\bar{a}_i^p, \bar{a}_i) \bar{a}_i^p
\]

leads to the formulation:

\[
\left[ \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi_j} \left( \sqrt{g} T_{ij} \left( \bar{a}_k^p, \bar{a}_k \right) \right) \right] \bar{a}_i^p = (\bar{a}_i^p, \bar{F}) \bar{a}_i^p,
\]

which is as compact as (4.4). Of course \( \bar{a}_i^p, \bar{a}_k \neq \delta_i^k \), except at the point \( P \).

One advantage of this formulation, compared with (4.4), is the more convenient implementation of boundary conditions. Furthermore it may be mentioned that the Cartesian decomposition has a drawback in combination with the choice of a staggered grid, as explained in Rodi et al. (1989) for example. We shall explore the need for grid staggering later, but its use in incompressible flow simulations is widespread. Incidentally, this illustrates how strongly the choice of the scalar decomposition is interwoven with the discretisation.

The main reasons for us to favour the decomposition (4.5) over (4.4) are a) that a grid-aligned decomposition is the natural choice in the transition from rectilinear to curvilinear coordinate systems and b) that the existence of a dominant flow direction can better be exploited with grid-aligned momentum components. Accordingly, we shall proceed with the formulation (4.5).

### 4.3 Decomposition of velocity vector

Besides momentum, also the velocity is a vector quantity and we are free to choose any convenient set of components for defining three dependent variables in addition to the pressure. It is generally preferable to make a choice in accordance with the decomposition of the momentum equation, i.e. to combine (4.5) with some
gridline-oriented set of variables or (4.4) with a Cartesian set. This preference is related to the better matrix properties obtained after discretisation. Having selected (4.5), we favour gridline-oriented velocity components. Even then there are still some options available: contravariant, \( \sqrt{g} \)-scaled contravariant or physical components.

Equation (4.5) becomes upon expanding \( T^{jk} \):

Convection part:
\[
\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} \left( \sqrt{g} \rho U^j U^k (\tilde{a}_p^i, \tilde{a}_k) \right) \tilde{a}_i
\]

Pressure part:
\[
\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} \left( \sqrt{g} g^{jk} p (\tilde{a}_p^i, \tilde{a}_k) \right) \tilde{a}_i = \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} \left( \sqrt{g} (\tilde{a}_p^i, \tilde{a}_k) p \right) \tilde{a}_i
\]

Viscous part:
\[
\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} \left[ \sqrt{g} \mu_e \left( g^{im} (\tilde{a}_p^i, \frac{\partial U^m}{\partial \xi^k} \tilde{a}_n) + g^{km} (\tilde{a}_p^i, \frac{\partial U^m}{\partial \xi^k} \tilde{a}_n) \right) \right] \tilde{a}_i
= \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} \left[ \sqrt{g} \mu_e \left( (\tilde{a}_p^i, \tilde{a}_m) (\tilde{a}_p^i, \frac{\partial U^m}{\partial \xi^k} \tilde{a}_n) + (\tilde{a}_p^i, \tilde{a}_m) (\tilde{a}_p^i, \frac{\partial U^m}{\partial \xi^k} \tilde{a}_n) \right) \right] \tilde{a}_i
\]

External-force part:
\[
(\tilde{a}_p^i, \tilde{F}) \tilde{a}_i
\]

With the scaled contravariant velocity components \( \tilde{V}^i \) and the scaled base vectors \( \tilde{b}^i \), this can be recast in the form:

Convection part:
\[
\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} \left( \rho \tilde{V}^j V^k (\tilde{b}_p^i, \tilde{b}_k) \right) \tilde{b}_i
\]

Pressure part:
\[
\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} \left( (\tilde{b}_p^i, \tilde{b}_k) p \right) \tilde{b}_i
\]

Viscous part:
\[
\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} \left[ \frac{\mu_e}{\sqrt{g}} \left( (\tilde{b}_p^i, \tilde{b}_m) (\tilde{b}_p^i, \frac{\partial V^n b_n}{\partial \xi^m}) + (\tilde{b}_p^i, \tilde{b}_m) (\tilde{b}_p^i, \frac{\partial V^n b_n}{\partial \xi^m}) \right) \right] \tilde{b}_i
\]

External-force part:
\[
(\tilde{b}_p^i, \tilde{F}) \tilde{b}_i
\]
while the mass conservation equation reads:

\[
\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} V^j = 0.
\]

Substituting \(U(i)/|\vec{b}_i|\) for \(V^i\), one obtains the equation for the grid-aligned physical velocity components \(U(i)\).

The equations for \(V^i\) or \(U(i)\) seem quite suitable for numerical solution. As a matter of fact we have used \(V^i\) as dependent variables for quite a long time. Yet there is a difficulty with this form in the exact reproduction of a uniform flow under appropriate boundary conditions. This difficulty will be explained in the following chapter on discretisation. In anticipation thereof, it suffices here to say that it will induce us to discretise the equations with the Cartesian velocity components and to turn to the physical grid-aligned velocity components as dependent variables only after discretisation. So initially the equation

\[
\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} \left[ \rho(\vec{b}^i, \vec{U})(\vec{b}_p \cdot \vec{U}) + (\vec{b}^i \cdot \vec{b}^j) p \right. \\
- \frac{\mu_e}{\sqrt{g}} \left( (\vec{b}^i \cdot \vec{b}^m)(\vec{b}_p \cdot \frac{\partial \vec{U}^l}{\partial \xi^m}) + (\vec{b}_p \cdot \vec{b}^m)(\vec{b}^i \cdot \frac{\partial \vec{U}^l}{\partial \xi^m}) \right) \left. \right] \vec{b}_p = (\vec{b}_p \cdot \vec{F}) \vec{b}_i, \tag{4.6}
\]

will be used.

Referring to the discussion in Chapter 2 about the negligibility of diffusion in the mainstream direction, there remains a statement to be made about what part of the diffusion terms is to be neglected. In the general formulation of this chapter we identify the \(\xi\)-direction as the dominant flow direction, which is of course a much better choice than the \(x\)-direction, particularly near the hull surface. Diffusion along the \(\xi\)-direction is therefore neglected by stipulating \(j \neq 1\) in the viscous part of equation (4.6).

### 4.4 Final form of equations

Let us summarise now the equations to be discretised, written in terms of the general non-orthogonal curvilinear coordinate system \(\xi^i\).

Momentum equation

\[
\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^i} \left[ \rho u^k b_{(n)}^i b_{(n)}^j + (b_{(n)}^i) p \right. \\
- \left. \frac{\mu_e}{\sqrt{g}} b_{(n)}^i b_{(n)}^j \left( b_{(n)}^m \frac{\partial u^k}{\partial \xi^m} + b_{(n)}^j \frac{\partial u^k}{\partial \xi^m} \right) \right] \vec{b}_p = (\vec{b}_p \cdot \vec{F}) \vec{b}_i, \tag{4.7}
\]
Continuity equation
\[
\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} \left( \gamma_{(k)} u^k \right) = 0
\] (4.8)

After discretisation, equations for grid-aligned velocity components are obtained by carrying out the substitution
\[
u^i = U(j)b_j^{(i)} / |\vec{b}_j|.
\]

4.5 Turbulence model

4.5.1 Algebraic model

The algebraic turbulence model, summarised at the end of section 2.2.1, is transformed and generalised to curvilinear coordinates as follows.

a) The 'hull-normal' coordinate \( y \), as it appears in the \( \nu_0 \)-formula, in the expression for the mixing length \( \ell \) and in the intermittency function \( \gamma \), is replaced by \( \eta_n \), a quasi normal distance to the hull surface, measured along the \( \eta \)-coordinate line:

\[
\eta_n = \int_0^\eta \frac{\tilde{a}_2 \tilde{a}_2^2}{|\tilde{a}_2| |\tilde{a}_2^2|} \sqrt{g_{22}} d\eta' = \int_0^\eta \frac{1}{|\tilde{a}_2^2|} d\eta' = \int_0^\eta \frac{\sqrt{|g|}}{|\tilde{b}|} d\eta'.
\]

The integrand can be interpreted as the infinitesimal physical distance along the \( \eta \)-line, multiplied by the cosine of the local angle between the \( \eta \)-line and the normal to the \( \xi-\zeta \) surface.

b) The quantity \( q \) is generalised to the magnitude of the velocity component locally tangential to a \( \xi-\zeta \) surface:

\[
q = \sqrt{U \cdot \tilde{U} - (\tilde{U} \cdot \tilde{a})^2 / |\tilde{a}|^2} = \sqrt{U_i U^i - (U^2 / |\tilde{a}|)^2}
\]

c) The rate of strain term \( \sqrt{u_y^2 + w_y^2} \) in the expression for \( \nu_0 \) might be generalised to the coordinate-invariant quantity

\[
S = \sqrt{2 S^{ij} S_{ij}}
\]

with

\[
S_{ij} = \frac{1}{2} \left( \tilde{a}_i \frac{\partial \tilde{U}}{\partial \xi^j} + \tilde{a}_j \frac{\partial \tilde{U}}{\partial \xi^i} \right)
\]

\[
S^{ij} = \frac{1}{2} \left[ g_{im} \left( \tilde{a}^i \frac{\partial \tilde{U}}{\partial \xi^m} \right) + g_{jm} \left( \tilde{a}^j \frac{\partial \tilde{U}}{\partial \xi^m} \right) \right]
\]

But instead of \( S \) we (like many others) use the magnitude of the local vorticity \( \tilde{\omega} \), the argument being that in the flows of interest to us turbulence occurs only
where mean-flow-field vorticity is present, while both quantities are only marginally different in thin shear layers. The magnitude of the vorticity, obviously coordinate-invariant too, is:

$$|\omega| = (\omega_i\omega^i)^{1/2} = (g_{ij}\omega^i\omega^j)^{1/2}$$

with $\omega^i$ given by

$$\omega^i = -\frac{1}{\sqrt{g}} e^{ijk} \left( \frac{\partial U_j}{\partial \xi^k} - \Gamma^m_{jk} U_m \right)$$

$$e^{ijk} = 1 \quad \text{for even permutation of } i, j, k$$

$$= -1 \quad \text{for odd permutation of } i, j, k$$

$$= 0 \quad \text{otherwise.}$$

It yields

$$\omega^1 = -\frac{1}{\sqrt{g}} \left( \frac{\partial U_2}{\partial \xi^3} - \frac{\partial U_3}{\partial \xi^2} \right)$$

$$\omega^2 = -\frac{1}{\sqrt{g}} \left( \frac{\partial U_3}{\partial \xi^1} - \frac{\partial U_1}{\partial \xi^3} \right)$$

$$\omega^3 = -\frac{1}{\sqrt{g}} \left( \frac{\partial U_1}{\partial \xi^2} - \frac{\partial U_2}{\partial \xi^1} \right)$$

d) $x_{tc}$ in the expression for $\nu_{tc}$ is treated as a function of $\zeta$.

### 4.5.2 One-equation model

The general form of Menter's one-equation model becomes

$$\nu_t = \hat{\nu}_t \left[ 1 - e^{-\left( \hat{\nu}_t / A_+ \kappa \nu \right)^2} \right]$$

$$-\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} \left( V^j \hat{\nu}_t - \sqrt{g} g^{jk} (\nu + \hat{\nu}_t) \frac{\partial \hat{\nu}_t}{\partial \xi^k} \right) = c_1 \left( \frac{\nu_t + \nu}{\hat{\nu}_t + \nu} \right) \hat{\nu}_t S - c_2 \left( \frac{\hat{\nu}_t}{L_{vK}} \right)^2$$

with

$$c_1 = 0.144, \quad c_2 = 1.86, \quad A_+ = 13, \quad \kappa = 0.41,$$

and

$$S = \sqrt{2 S^{ij} S_{ij}}$$

$$S_{ij} = \frac{1}{2} \left( \tilde{a}_i \frac{\partial \tilde{U}}{\partial \xi^j} + \tilde{a}_j \frac{\partial \tilde{U}}{\partial \xi^i} \right)$$

$$S^{ij} = \frac{1}{2} \left[ g^{im} \left( \tilde{a}^j \frac{\partial \tilde{U}}{\partial \xi^m} \right) + g^{jm} \left( \tilde{a}^i \frac{\partial \tilde{U}}{\partial \xi^m} \right) \right]$$
\[ L_{vK} = \left( \frac{S^2}{g^{ij} \frac{\partial S}{\partial x^i} \frac{\partial S}{\partial x^j}} \right)^{1/2} \]

To avoid anomalies with \( L_{vK} \) when \( S \) goes to zero, the destruction term is limited.

### 4.5.3 Correction for longitudinal vortices

As explained in Chapter 2, ship stern flows with streamwise vortices put a high demand on turbulence modelling. Most models do not properly account for the damping effect on turbulence near the core of a vortex. In order to obtain acceptable flow field predictions complex models must be used or simple models must be adjusted. We prefer to tune a simple model to the typical stern flow features rather than to apply second-moment closure. But the difficulty is then to find a measure which produces the desired damping where needed but is passive elsewhere.

In a paper by Dacles-Mariani et al. (1995) on an investigation of a near-field wingtip vortex, a suggestion by Spalart is followed by using the quantity

\[ \min(S - |\omega|, 0) \]

as a correction term, with the strain rate \( S \) and the vorticity \( \omega \) defined as above. This quantity is effective in regions where the vorticity exceeds the strain rate, which happens in the core region of a streamwise vortex, while in a thin shear layer \( S \) and \( \omega \) are practically of equal magnitude. Thus we introduce a "vortex correction" in the turbulence model, which consists of multiplying the production term in the transport equation for the eddy viscosity by the spatially varying factor:

\[ c_1 = \frac{S + c_{vor} \min(S - |\omega|, 0)}{S} \]

where we adopt the constant \( c_{vor} = 4 \).
Chapter 5

Discretisation

Discretisation, in the present context, is in essence the replacement of the partial differential equations describing the continuum flow by a set of algebraic equations, for example by replacing the derivatives of the differential equations by finite difference quotients. Thus described, discretisation seems an easier job than it actually is. Not apparent in the description is that there are certain rules to be obeyed, while also accuracy, stability and compactness requirements call for attention. To begin with, the discrete approximation of each term must be consistent, i.e. approach the analytical term arbitrarily closely when the grid spacing is suitably reduced. This is usually not so difficult to achieve. But in practice, the grid spacing is finite, meaning that there is always a loss of quantitative accuracy, measured by the truncation error of the difference approximation. More serious is that the discretisation, in spite of being consistent, can change the qualitative behaviour of the equations, sometimes in an unacceptable manner. Numerical instabilities may occur; a numerical diffusion effect, exceeding the physical diffusion, may be introduced; the conservative property of the continuum equations may be lost in the discrete case, etc. This makes that discretisation of the Navier-Stokes equations requires considerable care, and that the selection of a suitable discretisation scheme is often determined by the kind of application that is foreseen. Fortunately, we can rely to a large extent on previous work by numerous others to select proper discretisation schemes. We shall therefore refer to relevant material where appropriate.

In devising numerical approximations for the individual terms in the equations, one can operate from various backgrounds: Taylor series expansion, polynomial fitting or the control-volume method (Roache, 1972). In simple cases they lead to identical formulations, but noticeable differences appear for complex equations and for curvilinear grids. We shall follow here the control-volume (or finite-volume) approach, because it aims at satisfying the physical laws macroscopically rather than in the hypothetic limit of vanishing grid spacing.

To the finite-volume class belong all methods which discretise after integration of the differential equations over a small control volume – typically a primary cell of the grid – and the subsequent application of Gauss' theorem. An important distinction
between various finite-volume discretisations is the choice of the locations where discrete values of the dependent variables are defined. Thus there are cell center schemes, which choose the variables to be associated with the center of a cell; cell vertex schemes, defining the variables in the cell vertices, i.e. the grid nodes; and cell face schemes using the middle of the cell faces as the variable locations. Of these three options we have selected the first one.

Having thus chosen our starting point, we will elaborate relevant details of our discretisation in this chapter. As a prelude and as a motivation and clarification of our choices, we begin with a brief discussion of three important aspects. The first is the link between the discretisation of the continuity equation on the one hand and of the pressure gradient in the momentum equation on the other hand. It is related to the discrete modelling of the ellipticity of the equations. This is an important issue in incompressible flow calculation, in particular when all dependent variables are defined in the same grid nodes and not in a staggered arrangement. The second aspect is the discretisation of the convection terms, governed by considerations of stability, accuracy, transport properties and robustness. The discussion will finally lead us to solution-adaptive discretisation. Thirdly, some remarks will be made about the problem of uniform-flow-preservation in a discrete formulation of the equations.

5.1 Discrete ellipticity

In Chapter 2 we have considered the elliptic nature of the set of equations to be solved. The related concept of ellipticity for systems of difference equations is quite involved. An elaborate account of several variants has been given by Brandt & Dinar (1979). Here we shall concentrate on the significance of the positive definite character of the last factor of equations (2.5) and (2.6) for the discretisation process. To that end, let us repeat the analysis of Chapter 2 for the discrete case.

Consider for simplicity a rectangular mesh with uniform grid spacing $h$. In order to establish the discrete analogue of eq. (2.3), we introduce a translation operator $T$, defined as:

$$T_h^t[\phi(\vec{x})] = \phi(\vec{x} + \vec{\mu}h)$$

in which $\phi$ is the discrete approximation of a scalar variable (velocity component or pressure), $\vec{x} = (x, y, z)$ is the position vector and $\vec{\mu}$ is a vector which will typically have integer or half integer components. Since most differential operators are one-dimensional, $\vec{x}$ and $\vec{\mu}$ are often replaced by scalars; an index is then added to $h$ to indicate the directionality of the operator.

The three-point central difference approximation of the first $x$-derivative of $\phi$ is

$$\left( \frac{\Delta \phi}{\Delta x} \right)^C = \delta_x^C(\phi) = (\phi_{x+h} - \phi_{x-h})/2h.$$
The corresponding approximation of the differential operator $\frac{\partial}{\partial x}$ can therefore be written in terms of $T$ as

$$\delta_x^C = (T_{h+}^1 - T_{h-}^{-1})/2h.$$  

Similar expressions can be derived for other derivatives. The second $y$-derivative, for example, becomes

$$\delta_{yy} = (T_{h+}^1 - 2 + T_{h-}^{-1})/h^2.$$  

Following this convention, we can formulate the discrete equivalent of the operator matrix in eq. (2.3) in terms of $T$, operating on the discrete approximation of $u, v, w, p$, once the finite-difference approximations have been chosen.

We proceed to write the discrete solution in a Fourier series. It is then easily established that

$$T_h^d[\phi(\vec{x})] = \phi(\vec{x} + \vec{\mu}h) = A_{\phi} \exp(i\vec{k}.\vec{x}) \exp(i\vec{\mu}h).$$

The Fourier transform of $T_h^d$ is therefore $T_h^d = \exp(i\vec{k}.\vec{\mu}h)$, whence the Fourier transform of $\delta_x^C$ becomes:

$$\tilde{\delta}_x^C = (e^{i\bar{h}k_1} - e^{-i\bar{h}k_1})/2h = \frac{i}{h} \sin(hk_1),$$

which is to be compared with $i\bar{k}_1$ in eq. (2.4). We must keep in mind here that in the numerical approximation the components of the wave number vector have an upper limit: the shortest waves which can be resolved on the grid have wave number $k = \pi/h$. If we introduce the compact notation

$$\bar{\theta} = h\bar{k},$$

it follows that the components of $\bar{\theta}$ cannot be greater than $\pi$:

$$|\theta_j| \leq \pi.$$

Now let us, as an example, suppose that the differential operators in eq. (2.3) are approximated by the standard central difference formulas. Then the discrete analogue of $L_k$ becomes

$$L_\theta = \sum_{j=1}^3 \mu \rho n_j (e^{i\theta_j} - e^{-i\theta_j})/2h + \mu (e^{i\theta_j} - 2 + e^{-i\theta_j})/h^2,$$

and the principal symbol is given by

$$P_\theta = -\left[ \sum_{j=1}^3 \mu (e^{i\theta_j} - 2 + e^{-i\theta_j}) \right]/h^2 \sum_{j=1}^3 \left[ (e^{i\theta_j} - e^{-i\theta_j}) \right]^2 =$$

$$= -\left[ \sum_{j=1}^3 \mu (2 \cos \theta_j - 1) \right]/h^2 \sum_{j=1}^3 \left( \frac{\iota \sin \theta_j}{h} \right)^2.$$
\[
= \left[ \sum_{j=1}^{3} \mu \left( k_j \frac{\sin(\theta_j/2)}{(\theta_j/2)} \right)^2 \right]^2 \sum_{j=1}^{3} \left( k_j \frac{\sin \theta_j}{\theta_j} \right)^2.
\]

In analogy with (2.5) it should not vanish for |\bar{\theta}| > 0.

Unfortunately, this discrete principal symbol does vanish if one or more components of \( \bar{\theta} \) equal \( \pi \) and the remaining components are 0. The numerical solution is therefore unstable for some Fourier components. A small perturbation in these components can cause large changes in the solution without a corresponding behaviour in the solution of the differential equations. The problem is seen to be caused by the \( \sin \theta_j \) term in the second sum, originating from the discretisation of the pressure gradients and the continuity equation; the contribution of the diffusion terms to the principal symbol is positive definite over the whole range 0 < \( \theta_j \) ≤ \( \pi \).

This explains the well-known difficulty in the numerical simulation of incompressible flows which is often referred to as 'checkerboarding' or 'odd-even decoupling'. The solution is insensitive to point-to-point oscillations, notably in the pressure field. Experience tells that the boundary conditions cannot prevent this anomaly.

The most elegant way to avoid the decoupling problem is by using a staggered grid arrangement. The pressure nodes are then located at the centre of the grid cell and the velocity components at appropriate cell faces. By discretising the continuity equation at the pressure node and each of the momentum equations at the appropriate velocity node, the symbol for the pressure gradients as well as for the derivatives in the continuity equation becomes

\[
(e^{i\theta/2} - e^{-i\theta/2})/h = \frac{2i}{h} \sin(\theta/2)
\]

(N.B. the phase shift between momentum and continuity equations drops out). Their product in the principal symbol is therefore negative definite and the anomaly encountered with central differencing does not show up.

If for one reason or another grid staggering is not applied, however, it is easy to construct schemes, other than central differencing, yet based on a definition of all variables in the same grid nodes (collocated variables), which do preserve the ellipticity. In the construction of such schemes it is important to realise that \( \mathcal{P}_\theta \) may in general be a complex number and that satisfaction of the condition

\[
\mathcal{P}_\theta \neq 0
\]

is insufficient for stability. As \( \mathcal{P} \) is real and positive for real \( \tilde{k} \), we must require

\[
\Re(\mathcal{P}_\theta) > 0 \quad \text{for} \quad 0 < \theta_j \leq \pi,
\]

where \( \Re \) denotes the real part of a complex number. (This corresponds to Brandt & Dinar’s concept of R-ellipticity.) For suppose that we would allow \( \Re(\mathcal{P}_\theta) \) to be negative by some discretisation of pressure gradients and continuity equation. The
corresponding case in the continuum equations would then imply imaginary wave numbers and thus solutions with exponential-growth behaviour.

Suitable schemes are therefore those which are weighted in opposite directions for any component of the pressure gradient and the associated derivative in the continuity equation. The simplest case is with first-order schemes. The forward scheme for, say, the pressure gradient has then the symbol

\[ (e^{i\theta_j} - 1)/h, \]

and the backward scheme for the velocity gradient

\[ (1 - e^{-i\theta_j})/h. \]

The principal symbol now becomes

\[ P_\theta = \left[ \sum_{j=1}^{3} \mu \left( k_j \sin(\theta_j/2) \right)^2 \right]^2 \sum_{j=1}^{3} \left( k_j \sin(\theta_j/2) \right)^2, \]

and the condition (5.1) is seen to be satisfied. Examples of application can be found in Fuchs & Zhao (1984) and Elliston et al. (1987).

To obtain a better accuracy, second or third-order schemes can be used, as in Strikwerda (1984) and Strikwerda & Nagel (1988). Using the general \( \kappa \)-formulation for such schemes, suggested by van Leer (1977), the forward and backward difference formula are

\[ \delta_x^\kappa+(\phi) = [(\kappa - 1)\phi_{x+2h} + (5 - 3\kappa)\phi_{x+h} + (3\kappa - 3)\phi_x - (1 + \kappa)\phi_{x-h}] / 4h \]

\[ \delta_x^\kappa-(\phi) = [(1 - \kappa)\phi_{x-2h} + (3\kappa - 5)\phi_{x-h} + (3 - 3\kappa)\phi_x + (1 + \kappa)\phi_{x+h}] / 4h \]

where the value of \( \kappa \) is normally restricted to the range \([-1, 1]\]. With a little algebra, the contribution of the pressure gradients and continuity equation to the principal symbol can be derived as:

\[ -\sum_{j=1}^{3} \left( k_j \sin(\theta_j/2) \right)^2 \left[ 1 - (2\kappa - 1) \sin^2(\theta_j/2) + (\kappa^2 - 1) \sin^4(\theta_j/2) \right]. \quad (5.2) \]

For \( \kappa = 1 \), which corresponds to the central difference scheme, we recover the earlier problematic result. But for all \( \kappa < 1 \) the contribution is negative definite, which in combination with central differencing for the diffusion terms leads to the desired property (5.1) for the principal symbol.

With equivalent (symmetric) expressions for the forward and the backward difference schemes, as used in all examples above, the discrete principal symbol \( P_\theta \) is real valued. If the equivalence is dropped by introducing schemes of different order or different schemes of the same order (two \( \kappa \) values), the principal symbol will have an imaginary part also. If the opposite bias is maintained, however, its real part remains positive definite.
Instead of applying grid staggering, we will make use of such opposite-weighted schemes for the divergence operator in the continuity equation and the pressure gradient in the momentum equation, respectively, to guarantee the preservation of discrete ellipticity. Grid staggering, though attractive for the purpose of avoiding 'checkerboarding', has several inconveniences in other respects, particularly in 3D flow simulations on non-orthogonal grids (e.g. Peric et al., 1988).

5.2 Modelling of convection terms

In contrast to the modelling of the diffusion terms, the discretisation of the convection terms has been the subject of many discussions and of a vast amount of publications; and the debate is not yet finished. One might call it a controversial issue, but then the controversy is on permissible shortcomings of current schemes rather than on the final aim: a reconciliation of the conflicting requirements on accuracy on the one hand and stability, robustness and computational efficiency on the other.

In order to have a proper background for some further explanations, let us consider the variation of a convected scalar, \( \phi \) say, in the direction of a grid line. In the context of a finite-volume discretisation we need the cell face value \( \phi_f \) of \( \phi \), expressed in terms of the data at neighbouring cell centers. Following B.P. Leonard (1988), we denote the value of the convected variable in the cell center immediately downstream of the cell face by \( \phi_d \), the adjacent upstream node value by \( \phi_u \), while the value of \( \phi \) at the next upstream cell center is \( \phi_u \). 'Downstream' and 'upstream' refer to the direction of the convecting speed at the cell face, which will be represented here by \( u_n \).

The robust first-order upwind scheme postulates that the face value of the convected scalar \( \phi \) be equal to \( \phi_u \). Although this simple scheme has several attractive features, it is now widely accepted that its inherent artificial diffusion causes a loss of accuracy which cannot be tolerated, even in engineering applications. Accuracy requires therefore higher-order schemes.

Second and third-order schemes are conveniently written in the \( \kappa \)-formulation, proposed by van Leer (1977). Using the notation introduced above, we obtain for the cell face value of \( \phi \):

\[
\phi_f = \phi_c + \frac{1}{4}[(1 + \kappa)(\phi_d - \phi_c) + (1 - \kappa)(\phi_c - \phi_u)].
\] (5.3)

Herein is \( \kappa \) a parameter in the range \([-1,1]\). For \( \kappa = -1 \), for instance, the second-order linear upwind difference scheme (LUDS) is obtained, while \( \kappa = 1 \) yields the second-order central difference scheme (CDS). We have been enthusiastic users of Leonard’s QUICK scheme (Leonard, 1979) which is represented by the \( \kappa \)-formulation with \( \kappa = \frac{1}{2} \).

The QUICK scheme has several attractive properties: it is free of numerical diffusion, has low dispersion and good convective stability. But under extreme
conditions (sudden change of the convected variable under high convection) it can produce overshoots and slight oscillations. In most incompressible fluid flows such extreme conditions do not occur (no shock waves), but with a strong source term in the equations (in our case the external forces representing the propeller action) or a rapid change of grid line orientation, abrupt changes in the gradients of the dependent variables are to be expected. It is relevant therefore to look for schemes that give a solution free of non-physical oscillations also under such circumstances.

The so-called Bounded Higher Order (BHO) or flux-limiting schemes, the first of which appeared in papers by Gaskell & Lau (1988) and Leonard (1988), eliminate effectively the non-physical oscillatory behaviour that the conventional higher-order schemes can reveal. The construction of these schemes is conveniently explained with Leonard’s Normalised Variable Diagram (NVD) as follows.

If the original variable is \( \phi \), the locally normalised variable is defined by:

\[
\tilde{\phi} = \frac{\phi - \phi_v}{\phi_D - \phi_v},
\]

which implies

\[
\tilde{\phi}_D = 1 \quad \text{and} \quad \tilde{\phi}_v = 0.
\]

The change from real variables to normalised variables is illustrated in Figure 5.1 for the case \( u_n < 0 \).

![Figure 5.1: Original and normalised variables](image)

It will be clear that, where \( \phi_f \) is given by some relation between \( \phi_C, \phi_D \), and \( \phi_v \), the value of \( \phi_f \) is dependent on only one variable, viz. \( \tilde{\phi}_C \). The NVD is now constructed by plotting \( \tilde{\phi}_f \) versus \( \tilde{\phi}_C \) (Figure 5.2).
The general $\kappa$ formulation of second and third-order schemes (5.3) becomes in normalised variable notation:

\[ \tilde{\phi}_f = \tilde{\phi}_c + \frac{1}{\kappa} \left[ (1 + \kappa) - 2\kappa \tilde{\phi}_c \right]. \tag{5.4} \]

This means that the traditional second-order schemes are represented by straight lines in the NVD, some examples being shown in Figure 5.2. These lines have different slope for different values of $\kappa$, but all pass through $(0.5, 0.75)$. In fact, the latter is a necessary condition for second-order accuracy (Leonard, 1988; Leonard & Mokhtari, 1990).

It is now easily verified that with the QUICK scheme, which derives the value of $\phi_f$ from a quadratic interpolation of $\phi_u, \phi_c$ and $\phi_D$, a monotone variation of $\phi$ between $\phi_u$ and $\phi_D$ is ensured only if $0.25 \leq \tilde{\phi}_c \leq 0.75$. For values of $\tilde{\phi}_c$ outside this range the interpolant is non-monotone, although the discrete value $\tilde{\phi}_f$ does not necessarily reveal it. In order to prevent the occurrence of a physically unrealistic local maximum or minimum in $\phi$, the linear relationship between $\tilde{\phi}_f$ and $\tilde{\phi}_c$ must be abandoned. Where quadratic interpolation results in overshoots, limits must be set to $\tilde{\phi}_f$, leading to the so-called flux-limiting schemes. These are obtained by replacing (5.4) by a more general relation

\[ \tilde{\phi}_f = f(\tilde{\phi}_c), \]
where \( f \) is a piecewise continuous function that must satisfy \( f(0.5) = 0.75 \) for second-order accuracy and some further criteria to guarantee monotonicity.

Gaskell \& Lau (1988) have introduced the following boundedness criterion:

\[
\hat{\phi}_c \leq \hat{\phi}_f \leq 1 \quad \text{for} \quad \hat{\phi}_c \in [0, 1]
\]

\[
\hat{\phi}_f = \hat{\phi}_c \quad \text{for} \quad \hat{\phi}_c \leq 0, \hat{\phi}_c \geq 1
\]

This implies an admissibility region for the graph of \( \hat{\phi}_f(\hat{\phi}_c) \), which is displayed in Figure 5.3. It requires that if \( \phi_v \leq \phi_c \leq \phi_D \) the condition \( \phi_v \leq \phi_f \leq \phi_D \) must be satisfied, while \( \phi_f = \phi_c \) if \( \phi_c \) is not in the range \([\phi_v, \phi_D]\).

Several flux-limiting schemes have been proposed satisfying this criterion. Gaskell \& Lau’s SMART scheme is:

\[
\hat{\phi}_f = \left( \frac{6}{8} + 2\alpha \right) \hat{\phi}_c + \left( \frac{3}{8} - \alpha \right), \quad \alpha = \alpha(\hat{\phi}_c).
\]

Or in terms of non-normalised variables:

\[
\phi_f = \left( \frac{3}{8} - \alpha \right) \phi_D + \left( \frac{6}{8} + 2\alpha \right) \phi_c - \left( \frac{1}{8} + \alpha \right) \phi_v
\]

where \( \alpha \) is a solution-dependent flux-limiting parameter. Its value follows indirectly
from the relations

\[
\tilde{\phi}_f = \begin{cases} 
\tilde{\phi}_c & \tilde{\phi}_c \leq 0 \\
3\tilde{\phi}_c & 0 < \tilde{\phi}_c < \frac{1}{6} \\
\frac{3}{8}(2\tilde{\phi}_c - 1) & \frac{1}{6} \leq \tilde{\phi}_c \leq \frac{5}{6} \\
1 & \frac{5}{6} < \tilde{\phi}_c < 1 \\
\tilde{\phi}_c & \tilde{\phi}_c \geq 1
\end{cases}
\]

which describe a piece-wise linear function in the NVD.

Leonard & Mokhtari (1990) have suggested to use either

\[
\tilde{\phi}_f = \tilde{\phi}_c + \text{DWF}(1 - \tilde{\phi}_c), \quad \text{DWF} = \text{DWF}(\tilde{\phi}_c),
\]

implying

\[
\phi_f = \phi_c + \text{DWF}(\phi_d - \phi_c),
\]

or

\[
\tilde{\phi}_f = \frac{1}{2}(1 + \tilde{\phi}_c) - \text{VCF}(1 - 2\tilde{\phi}_c), \quad \text{VCF} = \text{VCF}(\tilde{\phi}_c),
\]

which becomes in non-normalised variables:

\[
\phi_f = \frac{1}{2}(\phi_d + \phi_c) - \text{VCF}(\phi_d - 2\phi_c + \phi_u).
\]

Appropriate flux-limiting expressions for DWF or VCF can be found in the cited reference.

Notice that where Leonard’s DWF and VCF formulations use respectively the first-order upwind and the central difference approximation of \(\phi_f\) as a basis to which a variable correction is applied, the SMART scheme is a curvature-corrected QUICK scheme.

Leonard & Mokhtari (1990) have also recognised the relation with the Total-Variation-Diminishing (TVD) schemes, first suggested by Harten (1984). These schemes, which were developed for the solution of hyperbolic sets of conservation laws, are usually written as

\[
\phi_f = \phi_c + \frac{1}{2}\Psi(\phi_c - \phi_u), \quad (5.5)
\]

where the flux-limiter \(\Psi\) is a function of the parameter

\[
r = \frac{\phi_d - \phi_c}{\phi_c - \phi_u}.
\]

Since

\[
r = \frac{1 - \tilde{\phi}_c}{\tilde{\phi}_c},
\]
a one-to-one correspondence with the NVD terminology can readily be established. Equation (5.5) implies
\[ \tilde{\phi}_f = f(\tilde{\phi}_c) = \tilde{\phi}_c (1 + \frac{1}{2} \Psi(r)) = \tilde{\phi}_c \left( 1 + \frac{1}{2} \Psi \left( \frac{1 - \tilde{\phi}_c}{\phi_c} \right) \right). \]

Conversely
\[ \Psi(r) = 2 \left( \frac{f(\tilde{\phi}_c)}{\phi_c} - 1 \right) = 2 \left( (1 + r)f \left( \frac{1}{1+r} \right) - 1 \right). \]

The essential properties of flux-limiting schemes are their solution-adaptivity and their non-linearity. The latter requires a linearisation which effectively determines the explicit and implicit parts of the scheme in an iterative solution process. The schemes presented above have been formulated so that the limiters (\(\alpha\), DWF, VCF and \(\Psi\), respectively) are evaluated explicitly. This means that the SMART and VCF scheme allow an implicit treatment of all three node values \(\phi_v\), \(\phi_c\) and \(\phi_o\). On the other hand, in Leonard’s DWF scheme the \(\phi_v\) contribution is necessarily explicit and the same holds for the \(\phi_o\) contribution in the \(\Psi\)-formulation.

The value of a limiter must be evaluated for each convected scalar, in all grid cells and for each iteration. Therefore the computational effort required for the evaluation of the limiter should be minimised. With this in mind, an easily determinable limiter can be devised by replacing the factor \(\frac{1}{4}\) in (5.4) by the flux-limiter \(\lambda\), yielding the flux-limiting \(\kappa\)-scheme:
\[ \bar{\phi}_f = \bar{\phi}_c + \lambda \left( (1 + \kappa) - 2\kappa \bar{\phi}_c \right), \quad \lambda = \lambda(\bar{\phi}_c), \quad (5.6) \]
or in terms of non-normalised variables:
\[ \phi_f = \phi_c + \lambda \left[ (1 + \kappa)(\phi_o - \phi_v) - 2\kappa(\phi_c - \phi_v) \right], \quad (5.7) \]
in which we choose:
\[ \lambda = \max \left( 0, \bar{\phi}_c(1 - \bar{\phi}_c) \right) \quad (5.8) \]

The corresponding non-linear relation between \(\bar{\phi}_f\) and \(\bar{\phi}_c\) is shown for \(\kappa = \frac{1}{2}\) in Figure 5.4. We note that for \(\bar{\phi}_c = 0.5\) the flux-limiting \(\kappa\)-scheme yields the same value and the same slope of \(\bar{\phi}_f\) as the non-limiting \(\kappa\)-scheme. Furthermore, with the choice (5.8), the scheme (5.6) implies
\[ \bar{\phi}_f = 2\kappa \bar{\phi}_c^2 - (1 + 3\kappa)\bar{\phi}_c^2 + (2 + \kappa)\bar{\phi}_c, \quad \bar{\phi}_c \in [0, 1] \]
which equals the ISNAS flux-limiting scheme for \(0 \leq \kappa \leq 1\) proposed by Zijlma (Zijlma, 1996 & 1996a; Zijlma & Wesseling, 1998). However Zijlma applies it in the \(\Psi\)-formulation. The \(\lambda\)-formulation is generally to be preferred for two reasons:

- the flux-limiter \(\lambda\) is more easily evaluated than the \(\Psi\)-limiter;
Figure 5.4: NV characteristic of flux-limiting QUICK scheme

- the λ-limiter leaves more freedom as to the treatment of the scheme in an iterative solution process: fully implicit or partly implicit/explicit.

The validity of the first advantage of the λ limiter is not restricted to the comparison with the Ψ limiter. Actually, we are not aware of any limiter that could claim to require less computational effort for its determination than the λ limiter.

On the other hand it may be noted that for $-1 \leq \kappa \leq 0$, the λ-limiter (5.8) does not satisfy the boundedness criterion of Gaskell & Lau. There is a slight overshoot of $\tilde{\phi}_f$ when $\tilde{\phi}_c$ approaches 1. The λ-limiter is therefore not recommended for application with $\kappa < 0$.

We shall use the scheme described by (5.6) and (5.8) with $\kappa = \frac{1}{2}$ in the discretisations to be presented later in this chapter. Some information on the practical implementation is given in Appendix A.

### 5.3 Uniform flow reproduction

It is a natural requirement on a discretisation scheme to produce exactly a uniform flow on an arbitrary, curved grid if boundary conditions appropriate for that flow condition are imposed. But the chain rule in differentiation, valid for the analytical equations, cannot be maintained usually in the discretisation process,
which makes the requirement harder to satisfy than may seem, in particular if the equations are written in strong conservation form. We do emphasize that a scheme, reproducing a uniform flow, is not necessarily a good one, nor is it necessarily more accurate than a scheme that does not. But from a practical point of view, it is important to avoid the usually small but very obvious numerical errors, which can appear when a uniform flow is modelled on a curved mesh, as we experienced in our earlier work (Eça & Hoekstra, 1992).

Let us take the strong-conservation form of the equations as a starting point for further discussion of this issue. Then the mass conservation equation (4.8) implies for an arbitrary uniform flow field

\[
\frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^j} \tilde{\rho} = 0,
\]

a geometrical constraint, which in the context of finite-volume discretisation is usually referred to as the 'closed-cell' condition. The satisfaction of this constraint by the discretisation scheme is a necessary, though not sufficient, ingredient if a uniform flow is to be modelled correctly. Also the convection terms and the pressure gradients in the momentum equation require it.

The same requirement holds if the scaled contravariant velocity components \( V^j \) are used as dependent variables. Then the constraint does not appear in the discretisation directly, but is needed upon reconstruction of the Cartesian velocity components. However, this works properly only if \( V^j \) are defined on the cell faces. For example, a discretisation scheme for 2D flows for the contravariant equations with scaled contravariant velocity variables, which is uniform-flow reproducing on an arbitrary grid, has been reported by Segal et al. (1992).

But as soon as face values must be derived by interpolation, difficulties arise. A way-out might be to evaluate the uniform flow error locally and to insert a compensating term explicitly in the right-hand side of the equations. But we prefer to follow yet another route which is based on the following steps:

- discretisation of the equations with the Cartesian velocity components to yield a system of equations for \( u^i \) and \( p \);
- subsequent replacement of the Cartesian velocity components by

\[
 u^i = U(j) \frac{\tilde{b}^i_j}{|\tilde{b}_j|};
\]

- formation of a new system of equations for \( U(i) \) and \( p \).

This guarantees uniform-flow preservation (up to machine accuracy) if the discretisation is consistent, the closed-cell condition is satisfied numerically and the boundary conditions are implemented correctly.
5.4 Geometric quantities

Having presented in the preceding sections of this chapter some background information on three important aspects, we come now to the details of the discretisation.

In Chapter 3, the generation of a boundary-fitted grid in the computation domain has been discussed. As a result we have a one-to-one mapping from a hexahedron \( G \) to the physical domain \( \Omega \). The hexahedral space \( G \) can be regularly divided into hexahedral cells, which without loss of generality can be assumed to have sides \( \Delta \xi^1 = \Delta \xi^2 = \Delta \xi^3 = 1 \).

Each grid cell has normally six faces. We distinguish the front (F) and the back (B) face across the \( \xi^1 \)-direction, so that \( \vec{a}_1 \) is pointing from back to front face. Further we identify a north (N) and a south (S) face across the \( \xi^2 \)-direction with \( \vec{a}_2 \) pointing to north; and finally an east (E) and a west (W) face closing the cell, where \( \vec{a}_3 \) points from east to west.

A particular cell volume is referred to as \( G_{i,j,k} \), where the indices \( i, j, k \) are related to the \( \xi^1, \xi^2, \xi^3 \) directions, respectively. Its counterpart in physical space is the cell volume \( \Omega_{i,j,k} \). A quantity or variable defined in the centre of that cell is also given the indices \( i, j, k \). The grid nodes constituting the vertices of the grid cell have indices \( i \pm \frac{1}{2}, j \pm \frac{1}{2}, k \pm \frac{1}{2} \). Where in this chapter incidentally fewer than three grid node indices appear, the missing ones default to \( i, j \) or \( k \), as appropriate. Average values over cell faces will be indicated by the indices \( F, B, N, S, E \) and \( W \) respectively.

The covariant base vector

\[
\vec{a}_n = \left( \frac{\partial x}{\partial \xi^n}, \frac{\partial y}{\partial \xi^n}, \frac{\partial z}{\partial \xi^n} \right)
\]

is in discrete form approximated by central differencing; it is determined at the cell center and at the center of each cell face. For example, the \( x \)-component of \( \vec{a}_1 \) at the cell centre becomes

\[
(a_1^{(x)})_{i,j,k} = \frac{1}{4} \left( x_{i+\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} + x_{i+\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}} + x_{i+\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} + x_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - x_{i-\frac{1}{2},j-\frac{1}{2},k-\frac{1}{2}} - x_{i-\frac{1}{2},j-\frac{1}{2},k+\frac{1}{2}} - x_{i-\frac{1}{2},j+\frac{1}{2},k-\frac{1}{2}} - x_{i-\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \right)
\]

Once the discrete covariant base vector is available in the cell center and on all six cell faces, the scaled contravariant base vector at the same locations is determined from

\[
(\vec{b}_\ell)_{i,j,k} = (\vec{a}_m)_{i,j,k} \times (\vec{a}_n)_{i,j,k}
\]

with \( \ell, m, n \) in cyclic order. Its covariant counterpart follows from

\[
(\vec{b}_\ell)_{i,j,k} = \frac{(\vec{a}_\ell)_{i,j,k}}{(\sqrt{g})_{i,j,k}}
\]

\(^1\)To avoid confusion, \( i,j,k \) are used in this chapter as grid node indices only, not in tensor formulations.
with
\[(\sqrt{g})_{i,j,k} = (\tilde{a}_\ell, (\tilde{a}_m \times \tilde{a}_n))_{i,j,k}\]
where again \(\ell, m, n\) are to be taken in cyclic order. This ensures that also in the discrete approximation we satisfy
\[\tilde{b}_m \cdot \tilde{b}_n = \delta^m_n,\]
while moreover the closed-cell condition
\[
\int \int \int_{G_{i,j,k}} \frac{\partial \tilde{b}_n}{\partial \xi^n} \, d\xi^1 \, d\xi^2 \, d\xi^3 = \tilde{b}_{F} - \tilde{b}_{B} + \tilde{b}_{N} - \tilde{b}_{S} + \tilde{b}_{W} - \tilde{b}_{E} \\
\approx \tilde{b}_{i-\frac{1}{2}}^{j+\frac{1}{2}} + \tilde{b}_{i+\frac{1}{2}}^{j+\frac{1}{2}} + \tilde{b}_{i}^{k+\frac{1}{2}} = 0
\]
is satisfied.

### 5.5 Basic finite-volume discretisation

Having established the determination of the base vectors, we can next consider the discretisation of the set of equations to be solved. The basis of the finite-volume discretisation is the integration of the equations over the volume of a grid cell. The volume integral of a quantity \(Q\) is
\[
\int \int \int_{n_{i,j,k}} Q \, d\Omega = \int \int \int_{G_{i,j,k}} Q \sqrt{g} \, d\xi^1 \, d\xi^2 \, d\xi^3.
\]
If \(Q\) is the divergence of a vector:
\[
Q = \frac{1}{\sqrt{g}} \frac{\partial Q^n}{\partial \xi^n},
\]
we obtain, when integrating over a grid cell:
\[
\int \int \int_{G_{i,j,k}} \frac{\partial Q^n}{\partial \xi^n} \, d\xi^1 \, d\xi^2 \, d\xi^3 \approx Q^n_{F} - Q^n_{B} + Q^n_{N} - Q^n_{S} + Q^n_{W} - Q^n_{E},
\]
which is approximated as:
\[
\int \int \int_{G_{i,j,k}} \frac{\partial Q^n}{\partial \xi^n} \, d\xi^1 \, d\xi^2 \, d\xi^3 \approx Q^n_{i-\frac{1}{2}}^{j+\frac{1}{2}} + Q^n_{i+\frac{1}{2}}^{j+\frac{1}{2}} + Q^n_{i}^{k+\frac{1}{2}}.
\]
All four equations will be discretised on primary cells, which is in contrast with methods based on a staggered variable arrangement. The dependent variables are defined in the cell center. The exact location of this center is not required in the calculation process; it is relevant for output purposes only, in which case it is chosen as the average of the coordinates of the cell vertices. The details of the discretisation will be described in the following sections.
5.6 Continuity equation

The finite volume discretisation of the mass conservation equation (4.8) becomes after integration over the cell volume:

$$\int \int \int \frac{\partial b_{(n)}^m}{\partial \xi^m} u^n d\xi^1 d\xi^2 d\xi^3 =$$

$$\sum_{n=1}^{3} \left[ (b_{(n)}^1 u^n)_F - (b_{(n)}^1 u^n)_B + (b_{(n)}^2 u^n)_N - (b_{(n)}^2 u^n)_S + (b_{(n)}^3 u^n)_W - (b_{(n)}^3 u^n)_E \right] = 0.$$

Face-average values of products are approximated by the product of face-center values. Since we have chosen a cell-centered variable allocation, the velocity components on the cell faces have to be derived. This is accomplished with:

$$u^n_F = \frac{3}{2} u^n_{i,j,k} - \frac{1}{2} u^n_{i-1,j,k}$$
$$u^n_B = \frac{3}{2} u^n_{i-1,j,k} - \frac{1}{2} u^n_{i-2,j,k}$$
$$u^n_N = \frac{3}{8} u^n_{i,j+1,k} + \frac{3}{8} u^n_{i,j,k} - \frac{1}{8} u^n_{i,j-1,k}$$
$$u^n_S = \frac{3}{8} u^n_{i,j,k+1} + \frac{3}{8} u^n_{i,j,k} - \frac{1}{8} u^n_{i,j,k-1}$$
$$u^n_W = \frac{3}{8} u^n_{i-1,j,k+1} + \frac{3}{8} u^n_{i-1,j,k} - \frac{1}{8} u^n_{i-1,j,k-1}$$
$$u^n_E = \frac{3}{8} u^n_{i+1,j,k} + \frac{3}{8} u^n_{i+1,j,k+1} - \frac{1}{8} u^n_{i+1,j,k-1}$$

The estimates for $u^n_F$ and $u^n_B$ can be recognised as linear extrapolations with a bias towards negative $\xi^1$. The estimates for $u^n_N$, $u^n_S$, $u^n_W$, and $u^n_E$ follow the QUICK approximation, which is a quadratic interpolation with a bias towards the negative $\xi^2$ and $\xi^3$ directions. The bias is needed as explained in section 5.1 above. The direction of the bias is arbitrary but must be opposite to the bias of the discretised pressure gradients (section 5.7.2).

5.7 Momentum equation

The momentum equation, as formulated in Chapter 4, includes the (locally) spatially-invariant base vectors $\vec{b}_F$ and $\vec{b}_B$. The point P, where they are evaluated, is chosen (per grid cell) as the cell center.

5.7.1 Convection terms

The convection terms of the $\xi^n$-momentum equation ($n = 1, 2, 3$) are discretised as:

$$\int \int \int \frac{\partial}{\partial \xi^n} \left( \rho u^n b_{(n)}^m \right) d\xi^1 d\xi^2 d\xi^3 \approx$$
\[
\begin{aligned}
\rho \sum_{m=1}^{3} u_{F+}^m b_{(m)F}^1 \sum_{\ell=1}^{3} u_{F+}^\ell b_{(\ell)P}^n - \rho \sum_{m=1}^{3} u_{F+}^m b_{(m)B}^1 \sum_{\ell=1}^{3} u_{F+}^\ell b_{(\ell)P}^n \\
+ \rho \sum_{m=1}^{3} u_{N+}^m b_{(m)N}^1 \sum_{\ell=1}^{3} u_{N+}^\ell b_{(\ell)P}^n - \rho \sum_{m=1}^{3} u_{S+}^m b_{(m)S}^1 \sum_{\ell=1}^{3} u_{S+}^\ell b_{(\ell)P}^n \\
+ \rho \sum_{m=1}^{3} u_{W+}^m b_{(m)W}^1 \sum_{\ell=1}^{3} u_{W+}^\ell b_{(\ell)P}^n - \rho \sum_{m=1}^{3} u_{E+}^m b_{(m)E}^1 \sum_{\ell=1}^{3} u_{E+}^\ell b_{(\ell)P}^n
\end{aligned}
\]

with

\[
\begin{aligned}
u_{F+}^m &= \frac{3}{8} u_{i,j,k}^m - \frac{1}{8} u_{i-1,j,k}^m \\
u_{N+}^m &= \frac{3}{8} u_{i-1,j,k}^m - \frac{1}{8} u_{i-2,j,k}^m \\
u_{F-}^m &= 0 \\
u_{B-}^m &= 0
\end{aligned}
\]

\[
\begin{aligned}
u_{V+}^\ell &= \begin{cases} u_{V+}^\ell & \text{for } V_+^{1} = \sum_{m=1}^{3} u_{V+}^m b_{(m)F}^1 \geq 0 \\
u_{V+}^\ell & \text{for } V_+^{1} = \sum_{m=1}^{3} u_{V+}^m b_{(m)F}^1 < 0 \end{cases} \\
u_{V-}^\ell &= \begin{cases} u_{V-}^\ell & \text{for } V_-^{1} = \sum_{m=1}^{3} u_{V-}^m b_{(m)F}^1 \geq 0 \\
u_{V-}^\ell & \text{for } V_-^{1} = \sum_{m=1}^{3} u_{V-}^m b_{(m)F}^1 < 0 \end{cases}
\end{aligned}
\]

\[
\begin{aligned}
u_{N+}^m &= \frac{3}{8} u_{j+1}^m + \frac{6}{8} u_j^m - \frac{1}{8} u_{j-1}^m \\
u_{N-}^m &= \frac{3}{8} u_j^m + \frac{6}{8} u_{j+1}^m - \frac{1}{8} u_{j+2}^m \\
u_{S+}^m &= \frac{3}{8} u_j^m + \frac{6}{8} u_{j-1}^m - \frac{1}{8} u_{j-2}^m \\
u_{S-}^m &= \frac{3}{8} u_{j-1}^m + \frac{6}{8} u_j^m - \frac{1}{8} u_{j+1}^m
\end{aligned}
\]

\[
\begin{aligned}
u_{N+}^\ell &= \begin{cases} u_{N+}^\ell & \text{for } V_+^{2} = \sum_{m=1}^{3} u_{N+}^m b_{(m)N}^2 \geq 0 \\
u_{N+}^\ell & \text{for } V_+^{2} = \sum_{m=1}^{3} u_{N+}^m b_{(m)N}^2 < 0 \end{cases} \\
u_{N-}^\ell &= \begin{cases} u_{N-}^\ell & \text{for } V_-^{2} = \sum_{m=1}^{3} u_{N-}^m b_{(m)N}^2 \geq 0 \\
u_{N-}^\ell & \text{for } V_-^{2} = \sum_{m=1}^{3} u_{N-}^m b_{(m)N}^2 < 0 \end{cases}
\end{aligned}
\]

\[
\begin{aligned}
u_{S+}^\ell &= \begin{cases} u_{S+}^\ell & \text{for } V_+^{2} = \sum_{m=1}^{3} u_{S+}^m b_{(m)S}^2 \geq 0 \\
u_{S+}^\ell & \text{for } V_-^{2} = \sum_{m=1}^{3} u_{S+}^m b_{(m)S}^2 < 0 \end{cases} \\
u_{S-}^\ell &= \begin{cases} u_{S-}^\ell & \text{for } V_-^{2} = \sum_{m=1}^{3} u_{S-}^m b_{(m)S}^2 \geq 0 \\
u_{S-}^\ell & \text{for } V_-^{2} = \sum_{m=1}^{3} u_{S-}^m b_{(m)S}^2 < 0 \end{cases}
\end{aligned}
\]

The discretisation of the ξ^3-derivatives is completely analogous with the discretisation of the ξ^2-derivatives as described above.
5.7.2 Pressure terms

The pressure part of the $\xi^n$-momentum equation ($n = 1, 2, 3$) is approximated as:

$$\int \int \int_{G_{i,j,k}} \frac{\partial (\vec{b}_p \cdot \vec{b}_m)}{\partial \xi^m} \, d\xi^1 \, d\xi^2 \, d\xi^3 \approx$$

$$p_F (\vec{b}_p \cdot \vec{b}_F) - p_B (\vec{b}_p \cdot \vec{b}_B) + p_N (\vec{b}_p \cdot \vec{b}_N) - p_S (\vec{b}_p \cdot \vec{b}_S) + p_W (\vec{b}_p \cdot \vec{b}_W) - p_E (\vec{b}_p \cdot \vec{b}_E)$$

with

$$p_F = \frac{3}{8} p_t + \frac{6}{8} p_{t+1} - \frac{1}{8} p_{t+2}$$
$$p_B = \frac{3}{8} p_{t-1} + \frac{6}{8} p_t - \frac{1}{8} p_{t+1}$$
$$p_N = \frac{3}{8} p_{j+1} + \frac{6}{8} p_j - \frac{1}{8} p_{j+2}$$
$$p_S = \frac{3}{8} p_{j-1} + \frac{6}{8} p_j - \frac{1}{8} p_{j+1}$$
$$p_W = \frac{3}{8} p_{k+1} + \frac{6}{8} p_k - \frac{1}{8} p_{k+2}$$
$$p_E = \frac{3}{8} p_{k-1} + \frac{6}{8} p_k - \frac{1}{8} p_{k+1}$$

Notice that the weighting of the QUICK discretisation of the pressure on the lateral cell faces is opposite to that of the velocity on the same faces in the continuity equation. The reason for this has been given in section 5.1.

5.7.3 Viscous terms

If the viscous flux difference in $\xi^1$ direction is neglected, the discretisation of the viscous terms in $\xi^n$-momentum equation ($n = 1, 2, 3$) is:

$$\int \int \int_{G_{i,j,k}} \frac{\partial}{\partial \xi^1} \left[ \frac{\mu}{\sqrt{g}} \begin{pmatrix} b_{(t)}^r \\ b_{(r)}^g \end{pmatrix} \left( b_{(t)}^m \frac{\partial u^r}{\partial \xi^m} + b_{(r)}^m \frac{\partial u^t}{\partial \xi^m} \right) \right] \, d\xi^1 \, d\xi^2 \, d\xi^3 \approx$$

$$\frac{\mu_N}{\sqrt{g}_N} \sum_{m=1}^{3} \sum_{r=1}^{3} \left[ (\vec{b}_p \cdot \vec{b}_N^m) b_{(r)}^N + (\vec{b}_N \cdot \vec{b}_N^m) b_{(r)}^m \right] \Delta m u^r_N$$
$$- \frac{\mu_S}{\sqrt{g}_S} \sum_{m=1}^{3} \sum_{r=1}^{3} \left[ (\vec{b}_p \cdot \vec{b}_S^m) b_{(r)}^S + (\vec{b}_S \cdot \vec{b}_S^m) b_{(r)}^m \right] \Delta m u^r_S$$
$$+ \frac{\mu_W}{\sqrt{g}_W} \sum_{m=1}^{3} \sum_{r=1}^{3} \left[ (\vec{b}_p \cdot \vec{b}_W^m) b_{(r)}^W + (\vec{b}_W \cdot \vec{b}_W^m) b_{(r)}^m \right] \Delta m u^r_W$$
$$- \frac{\mu_E}{\sqrt{g}_E} \sum_{m=1}^{3} \sum_{r=1}^{3} \left[ (\vec{b}_p \cdot \vec{b}_E^m) b_{(r)}^E + (\vec{b}_E \cdot \vec{b}_E^m) b_{(r)}^m \right] \Delta m u^r_E$$
where

\[
\begin{align*}
\Delta_1 u^r_N &= \frac{1}{2} (u^r_{i,j} + u^r_{i,j+1} - u^r_{i-1,j} - u^r_{i-1,j+1}) \\
\Delta_1 u^r_S &= \frac{1}{2} (u^r_{i,j} + u^r_{i-1,j} - u^r_{i,j-1} - u^r_{i-1,j-1}) \\
\Delta_1 u^r_W &= \frac{1}{2} (u^r_{i,k} + u^r_{i,k+1} - u^r_{i-1,k} - u^r_{i-1,k+1}) \\
\Delta_1 u^r_E &= \frac{1}{2} (u^r_{i,k} + u^r_{i-1,k} - u^r_{i,k-1} - u^r_{i-1,k-1}) \\
\Delta_2 u^r_N &= u^r_{i,j+1} - u^r_{i,j} \\
\Delta_2 u^r_S &= u^r_{i,j} - u^r_{i,j-1} \\
\Delta_2 u^r_W &= \frac{1}{2} \left\{ \begin{array}{l}
(u^r_{j,k+1} - u^r_{j-1,k+1} + u^r_{j+1,k} - u^r_{j,k}) \\
(u^r_{j+1,k+1} - u^r_{j,k+1} + u^r_{j,k} - u^r_{j-1,k})
\end{array} \right\} \\
\Delta_2 u^r_E &= \frac{1}{2} \left\{ \begin{array}{l}
(u^r_{j,k} - u^r_{j-1,k} + u^r_{j+1,k-1} - u^r_{j,k-1}) \\
(u^r_{j+1,k} - u^r_{j,k} + u^r_{j,k-1} - u^r_{j-1,k-1})
\end{array} \right\} \\
\Delta_3 u^r_N &= \frac{1}{2} \left\{ \begin{array}{l}
(u^r_{j+1,k} - u^r_{j+1,k+1} + u^r_{j,k+1} - u^r_{j,k}) \\
(u^r_{j+1,k+1} - u^r_{j,k+1} + u^r_{j,k} - u^r_{j-1,k})
\end{array} \right\} \\
\Delta_3 u^r_S &= \frac{1}{2} \left\{ \begin{array}{l}
(u^r_{j,k} - u^r_{j,k+1} + u^r_{j,k-1} - u^r_{j-1,k}) \\
(u^r_{j,k+1} - u^r_{j,k} + u^r_{j-1,k} - u^r_{j-1,k+1})
\end{array} \right\} \\
\Delta_3 u^r_W &= u^r_{i,k+1} - u^r_{i,k} \\
\Delta_3 u^r_E &= u^r_{i,k} - u^r_{i,k-1}
\end{align*}
\]

Where two options are indicated, the upper line is chosen if the multiplier of \(\Delta_m u^r\) is positive, the lower line in the opposite case.

5.7.4 External force terms

The right-hand side of the momentum equation, containing optionally applied external forces, is approximated as:

\[
\int \int \int_{G_{i,j,k}} (\vec{h}_F, \vec{F}) \sqrt{g} d\xi^1 d\xi^2 d\xi^3 \approx \sum_{m=1}^{3} (\sqrt{g})_{i,j,k} b^m_{(m)} (F^m)_{i,j,k},
\]

supposing that the external force vector \(\vec{F}\) is given in the cell center.
5.8 Resulting stencils

The discretisation stencils for the various terms are summarised in Fig. 5.5. It shows which neighbouring cells are involved in the discretisation of the various terms at the central cell.

5.9 Change of variables

As a result of the above discretisations we obtain per grid cell four discrete equations with the Cartesian velocity components and the pressure in the cell center as well as those in the center of neighbouring cells as unknowns. The final step in the discretisation process is a change of variables, i.e. a replacement of the Cartesian velocity components \( u^m \) by the grid-aligned physical velocity components \( U(m) \):

\[
 u^m = \frac{b_1^{(m)}}{|b_1|} U(1) + \frac{b_2^{(m)}}{|b_2|} U(2) + \frac{b_3^{(m)}}{|b_3|} U(3).
\]

After the substitution, the coefficients for \( U(1), U(2) \) and \( U(3) \) respectively can be collected to yield a modified linear system of equations with better properties than the system for \( u^m \). This is a kind of right-oriented preconditioning.

5.10 Eddy-viscosity transport equation

Having completed the description of the discretisation of the momentum equation, we can now be brief about the discretisation of the one-equation turbulence model. The convection and the diffusion terms in this transport equation for the eddy viscosity are treated in a similar way as the corresponding terms in the momentum equation, \( \tilde{\nu}_t \) taking the place of the transported velocity \( u^d \) in the convection part. The remaining production and dissipation terms are source terms, of which the first is written as an explicit term, while the second, after linearisation, makes an implicit contribution. This may be considered as a standard approach.

5.11 Implementation of boundary conditions

For the numerical implementation of the boundary conditions on the lateral boundaries (North, South, East and West boundaries) it is convenient to introduce two grid cell layers external to the boundary. The discretisations of \( \eta \) and \( \zeta \) derivatives presented above can then be applied without exceptions. What remains to be done is to express the variable values in the virtual grid cells in terms of interior node values and the boundary conditions.

Let a variable be denoted by \( \phi \) and consider a typical situation near a boundary of the uniformly spaced computational domain. The values of \( \phi \) in the center of the
Chapter 5: Discretisation

Figure 5.5: Discretisation stencils
cells adjacent to the boundary and inside the computation domain will be referred to as \( \phi_2 \) and \( \phi_3 \) respectively (cf. fig. 5.6). To the virtual grid cells on the other side of the boundary we attach the values \( \phi_1 \) and \( \phi_0 \).

Figure 5.6: Virtual grid points for boundary condition implementation

If the boundary condition for \( \phi \) is of Dirichlet type, the value of \( \phi \) on the boundary, \( \phi_{BC} \) say, is given. The chosen discretisation, based on a quadratic interpolant of \( \phi_2, \phi_3 \) and the virtual \( \phi_1 \), would evaluate this boundary value as:

\[
\phi_{BC} = \frac{3}{8} \phi_1 + \frac{6}{8} \phi_2 - \frac{1}{8} \phi_3
\]

or as

\[
\phi_{BC} = \frac{3}{8} \phi_2 + \frac{6}{8} \phi_1 - \frac{1}{8} \phi_0.
\]

The first condition implies:

\[
\phi_1 = \frac{8}{3} \phi_{BC} - 2 \phi_2 + \frac{1}{3} \phi_3,
\]

while the second yields:

\[
\phi_0 = 8 \phi_{BC} - 9 \phi_2 + 2 \phi_3,
\]

or alternatively

\[
\phi_0 = 3 \phi_1 - 3 \phi_2 + \phi_3.
\]

If a Neumann boundary condition for \( \phi \) is imposed, the gradient \( \phi'_{BC} \) is prescribed. We apply then:

\[
\phi_1 = \phi_2 - \phi'_{BC},
\]

\[
\phi_0 = 3 \phi_1 - 3 \phi_2 + \phi_3 = \phi_3 - 3 \phi'_{BC}.
\]
Chapter 5: Discretisation 69

For the special case of symmetry conditions ($\phi'_{BC} = 0$) this becomes:

$$\phi_1 = \phi_2$$

$$\phi_0 = \phi_3$$

If there is no boundary condition for $\phi$, the value of $\phi_{BC}$ is obtained by extrapolation from the interior of the computation domain. For the virtual $\phi_0$ and $\phi_i$ this implies:

$$\phi_i = 3\phi_2 - 3\phi_3 + \phi_4$$

$$\phi_0 = 3\phi_1 - 3\phi_2 + \phi_3$$

$$= 6\phi_2 - 8\phi_3 + 3\phi_4$$

Notice that for all three types of boundary condition $\phi_0$ can be written as

$$\phi_0 = 3\phi_1 - 3\phi_2 + \phi_3.$$  

Remark: A boundary condition is usually applied on a cell face, where the variable values have to be derived by some discretisation. Since we discretise in the Cartesian velocity components, the boundary conditions must be given in terms of these components. Sometimes this is easily accomplished. The conditions on a solid wall, for example, are as easily specified in Cartesian as in grid-aligned velocity components. The situation is different however on the external boundary, where we specify the tangential velocity components and the pressure; the velocity vector is not completely fixed. While specification of the boundary condition in grid-aligned velocity components would be straightforward, this is not true for the Cartesian components. We get around this difficulty by specifying the boundary conditions for the external boundary, which is artificial and can be chosen conveniently, in cell centers. Then the boundary conditions can be applied to the grid-aligned velocity components after the transformation described in Section 5.9.
Chapter 6

Iterative solution process

Once the discretisation of the equation system has been applied for all grid cells, a system of coupled algebraic relations is obtained, which can be represented symbolically by

$$\mathcal{F}(\phi) = 0,$$

if $\phi$ is used as a general notation for the unknowns. Because of the non-linearity of the convection terms, $\mathcal{F}$ is quadratic in $\phi$. We apply Newton linearisation to this system, implying that $\phi$ is solved iteratively with

$$\phi^{r+1} = \phi^r - \left( \frac{\partial \mathcal{F}}{\partial \phi} \right)_r^{-1} \mathcal{F}(\phi^r)$$

where $r$ is the iteration counter. Reformulation leads to the matrix-vector equation:

$$\left( \frac{\partial \mathcal{F}}{\partial \phi} \right)_r \phi^{r+1} = \left( \frac{\partial \mathcal{F}}{\partial \phi} \right)_r \phi^r - \mathcal{F}(\phi^r)$$

So in each Newton step, a linear system is to be solved which is in shorter notation and after dropping the index $r$ written as:

$$A\phi = b,$$  \hspace{1cm} (6.1)

where $A$ is the Jacobi matrix which is square and sparse, $\phi$ the vector of unknowns (pressure, velocity components and possibly $\tilde{v}_t$) and $b$ a known vector. If the number of grid cells in $\xi, \eta$ and $\zeta$ direction is denoted by $I, J$ and $K$ respectively, $A$ has $m \times I \times J \times K$ rows and columns, where $m = 4$ if the algebraic turbulence model is used and $m = 5$ if the transport equation for the eddy viscosity is involved. In practice, we always uncouple any turbulence model equations (to facilitate the use of various models), so that $m = 4$ in the discussions to follow.

To solve the linear system (6.1) an iterative process will be applied. An iterative approach fits well in the iteration sequence necessary to account for the effect of the non-linearity of the convection terms (Newton process), but even if $A$ would
have merely constant coefficients, its usual size and its sparsity favour an iterative solution.

While an iterative solution process is the common approach for finite-difference or finite-volume discretisations of the Navier-Stokes equations, there is a variety of iteration strategies, all with their merits and weaknesses. What most of them have in common, though, is the tendency towards size reduction, i.e. the partitioning of the original system, each part being treated successively or in parallel. That this will reduce the convergence rate is accepted as a price to be paid.

Reduction of the size of the equation system can be obtained by:

- uncoupling; the momentum and continuity equations are uncoupled and each equation is solved separately (segregated solution).

- subdivision; the computation domain is divided into suitable sub-domains and the equations are solved per sub-domain;

In both cases the iterative process must take care of restoring the original coupling of the system.

The first option, the uncoupling of the equations, immediately introduces a problem: if the velocity components are associated with the three momentum equations, the pressure is left to be associated with the continuity equation, in which it does not appear, however. This difficulty is usually circumvented by making use – in one way or another – of a Poisson equation for the pressure.

Gresho & Sani (1987) have discussed several pitfalls related to the use of the pressure Poisson equation and have listed a number of objections against it. Notably, they emphasize the need to ensure that the pressure Poisson equation plus the momentum equations imply the mass conservation equation, which is not self-evident. Furthermore, they discuss the treatment of boundary conditions at length. But some objections remain, which we can summarize as follows:

- Being an equation derived by differentiation, the pressure Poisson equation requires a greater smoothness of the solution (in both the velocity and the pressure) than the original system. In other words, the original system admits a larger class of solutions.

- There is some ambiguity in the boundary conditions and evaluation of the correct ones is laborious; the boundary conditions are of gradient type, at least on no-slip walls, which is unfavourable for the convergence rate of an iterative method.

- There is often no discrete divergence-free condition satisfied by the computed velocity field; if there is a solvability constraint (dependent on the type of boundary conditions; see also Chapter 2.3), it will be difficult to satisfy.
Nevertheless, there are several successful methods referring to a Poisson equation for the pressure. As a matter of fact, the majority of solution methods for the Navier-Stokes equations for incompressible fluids belongs to this class.

For example, in the Marker-and-Cell approach (Harlow & Welch, 1965), the momentum equations are solved in combination with the pressure Poisson equation, derived by taking the divergence of the momentum equation (2.1) and by simplifying the result, using (2.2). The property of incompressibility is enforced on the new system of equations by retaining an important time-derivative of the mass flow balance. In the MAC method\textsuperscript{1} the continuity equation is not satisfied until a converged solution is obtained.

This objection does not apply to the so-called Pressure Correction methods – with SIMPLE (Patankar & Spalding, 1972) as a typical example –, which is presumably the reason why they are more often used in steady flow problems. The strategy is here to solve first the momentum equations with a guessed pressure field $p^*$, yielding an intermediate solution $\tilde{u}^*$ for the velocity. Next, equations for velocity and pressure corrections $\delta \tilde{u}$ and $\delta p$ are derived from the momentum and continuity equations, so that $\tilde{u}^* + \delta \tilde{u}$ and $p^* + \delta p$ satisfy the original equations. By substituting the (simplified) expressions for $\delta \tilde{u}$ in the continuity equation, a Poisson equation for $\delta p$ results, which is solved to yield $\delta p$. Once $\delta p$ is known, $\delta \tilde{u}$ can be evaluated. This sequence is repeated until convergence.

In another variant, the Projection Method (Chorin, 1968), the momentum equations are initially solved for a uniform pressure field.

As an alternative for using a Poisson equation for the pressure, the Artificial Compressibility method has been suggested. The starting point is here the time-dependent form of the equations (i.e. with a time-derivative of the velocity added to the momentum equation). An essential difference between these equations and those for compressible fluid flows is the absence of a time-derivative in the continuity equation. In the Artificial Compressibility method a (quasi) time derivative of the pressure is added to the continuity equation, so that all four equations are in evolution form. Once in the time-iterative solution a steady state is obtained, the added term vanishes. This method has been introduced by Chorin (1967), while also earlier but not easily accessible work by Vladimirova \textit{et al.} (Yanenko, 1971) is usually referenced. Although the Artificial Compressibility method can be interpreted as a special iteration technique for solving a pressure Poisson equation (Peyret & Taylor, 1983), it does not directly refer to this equation and must therefore be considered as distinct. Like the MAC method, the Artificial Compressibility method has the property that the continuity equation is not satisfied until full convergence has been obtained.

With regard to segregation of the equations, we readily agree that the uncoupled equations can be solved with simpler algorithms which require less effort per cycle in an iterative process. But the coupling between the momentum and continuity

\textsuperscript{1}It is fair to say that the MAC method has been designed for computing unsteady rather than steady flows.
Numerical simulation of ship stern flows

equations is strong and therefore segregation is likely to deteriorate the convergence rate of the iterative solution and may even be the cause of divergence. There are also clear indications that iterative strategies based on uncoupling of the equations deteriorate with increasing Reynolds number (e.g. Braaten & Shyy, 1986). As a result, the over-all efficiency of the calculation procedure applied to high-\(Re\) flows (which is our purpose) is worse than what can be achieved with a solution of the coupled equations.

In the second way of obtaining size reduction, domain subdivision, any convenient splitting can be considered. In the most extreme case the subdomains are the individual grid cells (Vanka, 1986). But in most cases larger domains are chosen (e.g. Galpin et al., 1985; Vanka, 1985). Subdivision causes the convergence rate to be influenced unfavourably by the delayed communication of information between the subdomains. The main concern in this case is therefore to find a way to restore that information exchange completely and quickly.

The solution procedure that we have chosen achieves size reduction by subdivision, while the coupling of the equations is retained. The sub-domains are chosen by slicing the computation domain along \(\xi = \text{constant}\) planes, so that there are \(I\) sub-domains. This choice is of course motivated by the notion of a mainstream direction, which implies that information transport by convection is (primarily) one-way. Therefore a favourable sequence for visiting the subdomains is in downstream order. Thus emerges a multiple-sweep marching process with two iteration cycles: one cycle for the simultaneous solution of all variables in a subdomain, a layer of grid cells between two successive \(\xi\) stations, and another cycle which must take care of the propagation of information against the mainstream direction. The former cycle will be denoted as the inner or local iteration process, the latter as the outer or global iteration process. One global cycle (sweep) contains \(I\) local iteration processes. The details of both processes will be described in separate sections.

An iterative solution requires an initial guess of the velocity and pressure field. For the pressure we choose a thin-boundary-layer type of guess, \textit{viz.}:

\[ p_{i,j,k} = p_{i,j,k}; \]

where the right-hand side is provided by the boundary conditions. The initial guess for the velocity field is determined by

\[
U(1)_{i,j,k} = U(1)_{i-1,j,k} \frac{U(1)_{i,j,k}}{U(1)_{i-1,j,k}}
\]

\[
U(2)_{i,j,k} = U(2)_{i-1,j,k}
\]

\[
U(3)_{i,j,k} = U(3)_{i-1,j,k}
\]

which implies that the grid-aligned physical velocity components are transferred to the next \(\xi\)-station without change, except for a scaling of \(U(1)\). From the guessed grid-aligned velocity components the Cartesian components are derived by

\[
\vec{u} = U(n)\hat{b}_n/|\hat{b}_n|.
\]
At the first station boundary conditions are available. As will become clear below, the evaluation of the guess at station \( i \) can be postponed until a first-iteration solution at \( i - 1 \) has been determined.

While many time-iterative solution procedures require that the initial velocity field satisfies the continuity equation, no such requirement holds for our iteration scheme.

### 6.1 Global iteration process

Let the entries of \( A \) in (6.1) be grouped in blocks so that all elements multiplying the unknown variables in a grid plane \( \xi = \text{constant} \) form a block. If such a block, which itself is a matrix of \( 4 \times J \times K \) squared, is represented by one element and a corresponding grouping is carried through for the vectors \( \phi \) and \( b \), we obtain the quasi one-dimensional system

\[
A_{(\xi)} \phi_{(\xi)} = b_{(\xi)},
\]

where \( A_{(\xi)} \) is a \( I \times I \) square matrix; \( \phi_{(\xi)} \) and \( b_{(\xi)} \) are vectors of size \( I \), each element of \( \phi_{(\xi)} \) containing all unknowns at a particular \( \xi \)-station.

If the ordering is such that the first element of \( \phi_{(\xi)} \) is related to the inlet station, the sparsity pattern of \( A_{(\xi)} \) is for our discretisation shown in Fig. 6.1.

![Figure 6.1: Sparsity pattern of block-matrix](image)

The discretisation schemes selected in Chapter 5 for the \( \xi \) direction are in most cases different from those for the \( \eta \) and \( \zeta \) direction. This was done deliberately: when the discrete equations for a particular cell are considered, a minimum of references occurs to the variables in the downstream cells, i.e. the next cells in positive
\( \xi \) direction. As a matter of fact, there are references to the pressure in downstream cells only. Therefore the two diagonals with non-zero entries in the upper triangle contains multipliers for the pressures only. This suggests a Gauss-Seidel type iteration process:

\[
(L + D)\phi^*_n = b^*_n - U\phi^{n-1}_n
\]  
(6.2)

where \( L \) is the lower triangle part of \( A_n^\xi \), \( D \) its main diagonal, and \( U \) the upper triangle.

However, an obvious objection against the Gauss-Seidel process is that a pressure disturbance is propagated in upstream direction with a speed of only one cell length per iteration cycle. Therefore we have made several attempts to improve on it and have found a combination of three acceleration techniques to be quite effective. These three techniques will first be discussed separately before the combined global iteration strategy is presented. Another acceleration method, viz. increasing the size of the sub-domains from one layer of grid cells for \( \xi = \text{constant} \) to two or more, has not been tried yet.

### 6.1.1 Grid sequencing

An easy way to obtain faster propagation of pressure influences in upstream direction is to start the computation on a mesh coarsened in the mainstream direction and to refine gradually. This is called "grid sequencing", and we apply it optionally. Particularly on grids dense in \( \xi \)-direction, it is a time-saving and usually convergence-accelerating technique. It is to be distinguished from multi-grid: in grid sequencing there is no restriction process from fine to coarse grids.

Grid sequencing is invoked by the specification of a parameter \( mstep \), which specifies that the computation starts with an \( mstep \)-fold grid size. By choosing it equal to a power of 2, it is easy to set up an automatic grid refinement process. To facilitate this process, the number of grid cells in \( \xi \)-direction is assumed to be a multiple of \( mstep \).

If grid sequencing is applied, the discretisation of \( \xi^1 \) derivatives as described in Chapter 5 is adjusted and reduced to first order on all coarse grids. Only on the finest grid the original discretisation schemes are applied.

### 6.1.2 Symmetric Gauss-Seidel

As explained in some detail in Chapter 2, the mainstream pressure gradient is solely responsible for upstream influences once the mainstream diffusion terms are neglected. This pressure gradient can readily be isolated from the general form of the momentum equation (see Chapter 4, e.g. eq. (4.6)). If \( \tilde{M} \) is chosen to represent all terms except those in which the pressure occurs, we have

\[
\tilde{M} + \frac{1}{\sqrt{g}} \frac{\partial}{\partial \xi^i} (\tilde{v}^i p) = \bar{M} + \frac{\tilde{v}^i}{\sqrt{g}} \frac{\partial p}{\partial \xi^i} = 0.
\]
Taking the dot product with $\tilde{b}_i$, and subsequently choosing $i = 1$, we arrive at:
\[
\frac{\partial p}{\partial \xi^1} = -\sqrt{g}b_1.\tilde{M} = D(u, v, w, p),
\]
which is nothing else than a (weak-conservation) representation of the covariant $\xi^1$-momentum equation.

Imagine now a symmetric Gauss-Seidel iteration sequence for this scalar equation for the streamwise pressure gradient, modelled with a two-point downstream discretisation:

forward : \( (p_i^{n-1} - p_i^{n*})/\Delta \xi_i = D_\Delta(u^n, v^n, w^n, p^{n*})_i \)

backward : \( (p_i^{n**} - p_i^{n***})/\Delta \xi_i = D_\Delta(u^n, v^n, w^n, p^{n**})_i \)

where \( D_\Delta \) is an algebraic expression containing the velocity components at iteration level \( n \) and first-guess pressures \( p^{n*} \). With relaxation factors \( \omega_f \) and \( \omega_b \) we have:

forward : \( \left( \omega_f p_i^{n-1} - p_i^{n*} + (1 - \omega_f)p_i^{n-1} \right)/\Delta \xi_i = \omega_f D_\Delta(u^n, v^n, w^n, p^{n*})_i \) \hspace{1cm} (6.3)

backward : \( \left( \omega_b p_i^{n**} - p_i^{n***} + (1 - \omega_b)p_i^{n*} \right)/\Delta \xi_i = \omega_b D_\Delta(u^n, v^n, w^n, p^{n**})_i \) \hspace{1cm} (6.4)

Elimination of \( D_\Delta \) gives:

\[
p_i^{n**} = p_i^{n-1} + \left( 1 - \omega_b + \frac{\omega_b}{\omega_f} \right) (p_i^{n*} - p_i^{n-1}) + \omega_b \left( p_i^{n***} - p_i^{n-1} \right) \hspace{1cm} (6.5)
\]

This is the background for the construction of the following predictor-corrector scheme. We start to make one cycle in the iteration process (6.2), i.e. the solution of the coupled system of momentum and continuity equations. Let this process – as far as the pressure is concerned – constitute the predictor to give \( p^{n**} \) as a first estimate of the new pressure field. So we proceed from inlet to outlet station, evaluating \( p_i^{n*} \) (and the velocity field \( \tilde{u}_i^n \)) while using the downstream pressure \( p_{i+1}^n \) from the previous iteration. We consider (6.3) with \( \omega_f = 1 \) to be implied by this predictor step. Next we define an improved guess for the pressure as

\[
p^n = \omega_c p^{n**} + (1 - \omega_c)p^{n-1},
\]

referring to (6.4) for \( p^{n***} \). Substitution of this definition in (6.5) gives the corrector step:

\[
p_i^n = p_i^{n-1} + \omega_c \left( 1 - \omega_b + \frac{\omega_b}{\omega_f} \right) (p_i^{n*} - p_i^{n-1}) + \omega_b \left( p_i^{n***} - p_i^{n-1} \right). \hspace{1cm} (6.6)
\]

This is an extremely simple algebraic update of the pressure field. It must be evaluated by marching from the outlet to the inlet station, because of the appearance of \( p_i^{n+1} \). As a start for this upstream sweep through the domain, we have the condition

\[
p_i^{n} = p_i^{n-1} + \frac{\omega_c}{1 - \omega_b} \left( 1 - \omega_b + \frac{\omega_b}{\omega_f} \right) (p_i^{n*} - p_i^{n-1}),
\]
as immediately follows from (6.6) if the spatial pressure gradient is prescribed by a boundary condition at the outlet plane.

Notice that this predictor-corrector scheme does not require the solution of (6.3) and (6.4). The solution procedure is determined by (6.2) and (6.6).

While most convergence acceleration techniques increase the computational effort per iteration cycle substantially, the corrector step is very cheap in terms of computation cost, yet improves the convergence rate appreciably, as will be demonstrated with some examples in Chapter 7, and is supported by the linear stability analysis in Appendix B.

It is instructive to reformulate the corrector in terms of pressure differences with respect to iteration level \( n - 1 \). So let

\[
\Delta p_i^n = p_i^\ell - p_i^{n-1}, \quad \text{where } \ell = n, n^* \text{ or } n^{**}.
\]

Then the corrector becomes:

\[
\Delta p_i^n = \omega_c \left( 1 - \omega_b + \frac{\omega_b}{\omega_f} \right) \Delta p_i^{n*} + \omega_b \Delta p_i^{n+1}.
\]

Repeated substitution leads to:

\[
\Delta p_i^n = \omega_c \left( 1 - \omega_b + \frac{\omega_b}{\omega_f} \right) \left[ \sum_{k=i}^{l-1} \omega_b^{k-i} \Delta p_k^{n*} + \frac{\omega_b^{l-i}}{1 - \omega_b} \Delta p_l^{n*} \right]
\]

This shows that the pressure change at station \( i \) is determined by a fraction of downstream pressure changes resulting from the predictor, where the fraction decreases with distance, or more precisely with the number of stations in between. Thus we have incorporated the desirable feature of an infinite propagation speed of pressure influences in upstream direction (a property of the continuum equations) in the numerical scheme.

By expanding \( \Delta p^n \) and \( \Delta p^{n*} \) in Fourier series and assuming the computation domain to extend to infinity in downstream direction, we can evaluate the effect of the corrector on individual Fourier modes. The result for \( \omega_f = 1 \) is

\[
\frac{A^n_{\Delta p}}{A^{n*}_{\Delta p}} = \frac{\omega_c}{1 - \omega_b \exp(ikh)},
\]

which clearly indicates that \( \omega_b \) must be chosen smaller than 1, while \( \omega_c \leq 1 - \omega_b \), to let the corrector be a smoother.

We show a typical result in a graph (Fig. 6.2) which displays the amplification factor of the amplitude of a Fourier mode as a function of its wave length \( \lambda \) and of the relaxation factor \( \omega_b \), after having set \( \omega_f = 1 \) and \( \omega_c = 1 - \omega_b \). For \( 0 < \omega_b < 1 \) the corrector is clearly a smoother: it damps all Fourier modes. But the amplitude reduction is more pronounced for the short wave length components, while with increasing \( \omega_b \) the damping becomes stronger and affects a greater range of wave
lengths. Suitable values for the relaxation parameters are $\omega_f = 1$, $\omega_b = 0.5$ and $\omega_c = 0.5$.

In the previous chapter we have chosen, for accuracy reasons, not to use a first-order two-point discretisation for the pressure gradient. So the question is now: what to do if a higher order discretisation for the streamwise pressure gradient is employed? In that case the approach can be modified as follows. Suppose the QUICK discretisation is used as the alternative discretisation. Instead of (6.3) and (6.4) we obtain then

$$-\frac{3}{8}\omega_f p_i^{n-1} - \frac{3}{8}(p_i^{*} - (1 - \omega_f)p_i^{n-1}) + \frac{3}{8}\omega_f p_{i+1}^{n-1} - \frac{1}{8}\omega_f p_{i+2}^{n-1} = \omega_f D_{\Delta}(u^n, v^n, w^n, p^{*})_i$$

and

$$-\frac{3}{8}\omega_b p_i^{n*} - \frac{3}{8}(p_i^{**} - (1 - \omega_b)p_i^{n*}) + \frac{3}{8}\omega_b p_{i+1}^{n*} - \frac{1}{8}\omega_b p_{i+2}^{n*} = \omega_b D_{\Delta}(u^n, v^n, w^n, p^{*})_i$$

Elimination of $D_{\Delta}$ and $p^{***}$ yields

$$p_i^n = p_i^{n-1} + \omega_c(1 - \omega_b + \frac{\omega_b}{\omega_f})(p_i^{*} - p_i^{n-1})$$

$$+ \omega_b \left[\frac{1}{3}(p_{i+1}^n - p_{i+1}^{n-1}) - \frac{1}{3}(p_{i+2}^n - p_{i+2}^{n-1})\right]$$

We choose for $\omega_f$ the value $\frac{3}{8}$, or, more general, the value of the coefficient of $p_i$ in the discretisation scheme for the pressure gradient, multiplied by $-1$. Suitable
values for the other relaxation parameters are

\[ \omega_h = \frac{\omega_f}{1 + \omega_f} \quad \text{and} \quad \omega_c = 0.5. \]

In summary, with the application of the correction of the pressure field according to this second convergence acceleration method, we can call the global iteration process an Incomplete Symmetric Coupled Plane Gauss-Seidel process. Coupled Plane Gauss-Seidel refers to (6.2) where a coupled set of equations is solved in Gauss-Seidel manner with a simultaneous update of the variables in a \( \xi \)-plane (governed by the local iteration process, to be described below), while Incomplete Symmetric refers to the fact that in the process a backward loop is involved in which only the pressure is updated.

### 6.1.3 Approximate multigrid

The obvious acceleration technique for the global iteration process is the multigrid method (Hackbusch, 1985; Wesseling, 1992). A semi-coarsening variant, with grid coarsening and refinement in the \( \xi \)-direction only, seems a suitable choice. We have chosen to use an approximate form of it, based on an observation of the nature of the residual of the global iteration. This residual can be written as

\[ r^n_{\text{glob}} = b(\xi) - A(\xi)\phi^n_{\xi} = b(\xi) - (L + D + U)\phi^n_{\xi}. \]

The residual of the local iteration process, on the other hand, can be inferred from (6.2) as

\[ r^n_{\text{loc}} = b(\xi) - (L + D)\phi^n_{\xi} - U\phi^{n-1}_{\xi}, \]

so that

\[ r^n_{\text{glob}} = r^n_{\text{loc}} - U(\phi^n_{\xi} - \phi^{n-1}_{\xi}). \]

If it is supposed that \( r^n_{\text{loc}} \) is driven to a small value in the local iteration process, it follows that the global iteration residual is dominated by \( U(\phi^n_{\xi} - \phi^{n-1}_{\xi}) \). We have noted already that \( U \), the upper triangular part of \( A(\xi) \), contains multipliers only for downstream pressures, coming from the discretisation of the mainstream pressure gradient. This gradient can be isolated in the covariant \( \xi \) momentum equation. So the residual vector of the momentum equation is approximately aligned with the contravariant base vector \( \bar{a}^1 \), i.e. locally perpendicular to \( \xi = \text{constant} \) grid planes. And if, for simplicity, a two-point discretisation of the gradient \( \partial p/\partial \xi \) is assumed, its approximate magnitude is

\[ r^n_i \approx -(p^n_{i+1} - p^{n-1}_{i+1}). \]

This residual can now be restricted to a coarser grid:

\[ r^{n,2h} = R(r^{n,h}), \]
with $R$ representing the restriction operator and the upper indices $h$ and $2h$ the typical grid cell size. One would normally continue to compute corrections for the pressure and the velocity field on the coarse grid and after prolongation add them to the original solution. But we grossly simplify this process by setting all coarse-grid velocity corrections to zero and computing a correction for the pressure only. Since the residual vector has its main component in the covariant $\xi^1$ equation, we just put the coarse grid pressure gradient $(\partial p/\partial \xi^1)^{2h}$ equal to minus the residual. So we solve on the coarse grid in a forward sweep

$$p^{n*+2h}_{2i} - p^{n-1,2h}_{2i+2} = -r^{n,2h}_{2i+2},$$

with a first guess $p^{n-1,2h} = 0$, followed by a corrector step according to (6.6):

$$p^{n,2h}_{2i} = p^{n-1,2h}_{2i} + \omega_c \left( 1 - \omega_b + \frac{\omega_b}{\omega_f} \right) \left( p^{n*+2h}_{2i} - p^{n-1,2h}_{2i} \right) + \omega_b \left( p^{n,2h}_{2i+2} - p^{n-1,2h}_{2i+2} \right).$$

The resulting correction on the pressure field can be prolonged to the fine grid and subsequently added to the available fine-grid pressure.

The above description refers to a two-grid method, but extension to more grids is straightforward. We have found a saw-tooth multigrid cycle to be satisfactory. Since the proposed multigrid-like method is only approximate, attempts to use more complicated cycles like $W$ or $F$ cycles are not worthwhile.

### 6.1.4 Over-all strategy

The over-all strategy for the global iteration process is based on some combination of the three acceleration techniques described above. Grid sequencing and the predictor-corrector method are easily combined. If the approximate multigrid method is invoked, it is not applied in every global cycle but, say, every third cycle. Our experience is that with the increase of the grid level (the number of coarser grids involved in the process), the calling frequency of the approximate multigrid can be reduced.

### 6.2 Local iteration process

During a cycle of the global iteration process, we solve (6.2) successively at all $\xi = constant$ planes. One step in this Gauss-Seidel process can be formulated as:

$$A_{i,i-2}\phi(\xi)_{i-2} + A_{i,i-1}\phi(\xi)_{i-1} + A_{i,i}\phi(\xi)_{i} = b_{\xi} - A_{i,i+1}\phi(\xi)_{i+1} + A_{i,i+2}\phi(\xi)_{i+2}. \quad (6.7)$$

The matrices $A_{i,i}$ are elements of $A(\xi)$. As soon as station $i$ is reached, solutions for $\phi(\xi)$ at the upstream stations $i - 1$ and $i - 2$ are available, while $\phi(\xi)_{i+1}$ and $\phi(\xi)_{i+2}$ are taken from a previous iteration, as described earlier. So (6.7) reduces to

$$A_{i,i}\phi(\xi)_{i} = b'_{\xi}.$$
where the prime denotes that the contributions from stations \(i-2, i-1, i+1\) and \(i+2\) are incorporated. The local iteration process is concerned with the solution of this equation system.

### 6.2.1 CILU or preconditioned GMRES

For the solution algorithm in the local iteration procedure in PARNASSOS we can choose from a few alternatives, either belonging to the family of Coupled Incomplete LU-Factorisation (CILU) methods or being the Generalized Minimum Residual Method (GMRES) (Saad & Schultz, 1986), applied to the CILU-preconditioned linear system.

Application of a CILU means that \(A_{i,i}\) will be split as

\[
A_{i,i} = M - R,
\]

where \(M = LD^{-1}U\) is an easily factorisable part of \(A_{i,i}\) and \(R\) is a rest matrix, while we assume \(\text{diag}(L) = \text{diag}(U) = D\). The solution is found from:

\[
M\phi_{(\xi)}^{n+1} = b'_{(\xi)} + R\phi_{(\xi)}^{n},
\]

which can be split into the steps

\[
L\psi = b'_{(\xi)} + R\phi_{(\xi)}^{n} \quad \text{(forward substitution)}
\]

\[
D^{-1}U\phi_{(\xi)}^{n+1} = \psi \quad \text{(backward substitution)}
\]

If needed, these iteration sequences are damped by an underrelaxation factor \(\omega\):

\[
M\phi_{(\xi)}^{*} = b'_{(\xi)} + R\phi_{(\xi)}^{n},
\]

\[
\phi_{(\xi)}^{n+1} = \omega\phi_{(\xi)}^{*} + (1 - \omega)\phi_{(\xi)}^{n},
\]

so that

\[
M(\phi_{(\xi)}^{n+1} - \phi_{(\xi)}^{n}) = \omega \left[ b'_{(\xi)} + (R - M)\phi_{(\xi)}^{n} \right] = \omega r^{n},
\]

where \(r^n\) is the residual at iteration level \(n\).

Nothing has been said yet about the ordering of the coefficients in \(A_{i,i}\). We have to solve a set of four coupled equations in all grid cells and we can order the matrix entries per cell, per equation \(i.e.\) the four equations appearing on successive rows) or vice versa (all \(\xi\) momentum equations appearing first, then the \(\eta\) momentum equations, etc.). In addition one can use a regular ordering, but also red-black, diagonal or zebra ordering (to facilitate vectorisation) and these can be done in forward or backward mode as appropriate. They all lead to different matrix structures and therefore to potentially different behaviour of the approximate factorisation.

We order the entries per cell, per equation in a regular manner. In our case – a flow at high Re, with a solid wall as one of the boundaries of the grid and,
consequently, high stretching of the grid towards that boundary — the high aspect ratio of the cells causes an anisotropy of the coefficients in $A_{i,i}$. An ordering with $k$ running faster than $j$ is then to be preferred. This leaves still four options for ordering:

\[
\begin{align*}
m &= k + (j - 1) * K \\
m &= K - (k - 1) + (j - 1) * K \\
m &= J * (K - 1) + k - (j - 1) * K \\
m &= J * K - (k - 1) - (j - 1) * K
\end{align*}
\]

where $j, k$ are the grid cell indices as used in Chapter 5. It has been shown by others (e.g. Wesseling, 1992) that an alternation between different orderings can enhance the robustness of the procedure and improve the convergence rate. We have experimented with various combinations and found indeed that cycling between two or even all four orderings is often beneficial.

Further details of the factorisation process will be given in the following section. Instead of the above-described iterative procedure based on incomplete factorisation, the widely-used GMRES method can be chosen to solve the linear system of equations. Incomplete factorisation is then used as a preconditioner, and GMRES is applied to the preconditioned system:

\[
\left( L D^{-1} U \right)^{-1} A_{i,i} \phi_{(t)}, = \left( L D^{-1} U \right)^{-1} b_{(t)}^\prime,
\]

where $L$ and $U$ are obtained from one of the available incomplete factorisations.

As usual, to avoid the practical difficulty of GMRES that storage requirements and the computational effort increase with the number of steps of the solution process, GMRES is restarted once in a while. As a restart criterion we use the decrease of the $L_2$ norm of the residual vector by a factor 5, while the maximum number of search vectors is never allowed to exceed 20.

It is not necessary to carry out the approximate factorisation process in each Newton iteration. The preconditioner is held fixed once the maximum change of the variables between successive iterations falls below a preset tolerance.

### 6.2.2 CILU7 or CILU($\varepsilon$)

Of the several variants of incomplete factorisations, we have implemented a classical 7-point CILU with modifications, which allows entries in the factors $L$ and $U$ only within the sparsity pattern of the original matrix $A_{i,i}$ (factorisation by position) and a more general factorisation which accepts entries on any position as long as their value in some norm is large enough (factorisation by value). These factorisations will be referred to as CILU7 and CILU($\varepsilon$), $\varepsilon$ being the level of fill-in or drop tolerance.
In describing the construction of the incomplete LU decomposition of $M$ according
to the CILU7 factorisation, we limit ourselves to the one ordering option for $A_{i,i}$. The construction for the other alternatives proceeds analogously.

With a backward ordering of the grid cells:

$$m = J * K - (k - 1) - (j - 1) * K,$$

the sparsity pattern or graph of $A_{i,i}$ is

$$\mathcal{G} = \{(m, m \pm K \pm K), (m, m \pm K - 1), (m, m \pm K), (m, m \pm K + 1),
\{(m, m \pm 2), (m, m \pm 1), (m, m)\}$$

We recall that each element of $A_{i,i}$ is a $4 \times 4$ block. We shall now rename these elements per diagonal and use the grid cell indices $j, k$ directly as follows:

$$\begin{align*}
AN_{j,k} &= A_{m,m-k-k} \\
AW_{j,k} &= A_{m,m-k-1} \\
AN_{j,k} &= A_{m,m-k} \\
AE_{j,k} &= A_{m,m-k+1} \\
AW_{j,k} &= A_{m,m-2} \\
AW_{j,k} &= A_{m,m-1} \\
AC_{j,k} &= A_{m,m} \\
AE_{j,k} &= A_{m,m+1} \\
AE_{j,k} &= A_{m,m+2} \\
AS_{j,k} &= A_{m,m+k-1} \\
AS_{j,k} &= A_{m,m+k} \\
ASE_{j,k} &= A_{m,m+k+1} \\
ASS_{j,k} &= A_{m,m+k+k}
\end{align*}$$

A 7-point CILU implies a 7-diagonal sparsity pattern for $M$, which will be reduced here to a 5-diagonal pattern in order to save memory. These five diagonals will be denoted as $MC, MN, MW, MS, ME$, in analogy with the naming convention for $A$. Thus the sparsity pattern of $M$ has eight diagonals less than the pattern of $A_{i,i}$. Instead of simply moving the elements on these eight diagonals to the rest matrix $R$, we add them also to both $M$ and $R$. Gustafsson (1978) has suggested to put these elements on the main diagonal. We do so only for $ANW, ANE, ASW$ and $ASE$; the remaining elements are moved to the diagonal related to the nearest neighbour in the discretisation stencil. So, for example, $ASS$ goes to $MS$ and $R_S$ in addition to $RSS$.  


So, where the stencils of $A_{i,d}$ and $M$ are

$$[A_{i,d}] = \begin{bmatrix} 0 & 0 & AW & 0 & 0 \\ 0 & AS & AW & AN & 0 \\ AS & AS & AC & AN & AN \\ 0 & ASE & AE & ANE & 0 \\ 0 & 0 & AEE & 0 & 0 \end{bmatrix},$$

$$[M] = \begin{bmatrix} 0 & MW & 0 \\ MS & MC & MN \\ 0 & ME & 0 \end{bmatrix},$$

we have for $R$:

$$[R] = \begin{bmatrix} 0 & 0 & RW & 0 & 0 \\ 0 & RSW & RW & RN & 0 \\ RSS & RS & RC & RN & RNN \\ 0 & RSE & RE & RNE & 0 \\ 0 & 0 & REE & 0 & 0 \end{bmatrix}.$$  

With the fourth (backward) ordering the stencils for $L, D$ and $U$ are:

$$[L] = \begin{bmatrix} 0 & LW & 0 \\ 0 & LC & LN \\ 0 & 0 & LNE \end{bmatrix},$$

$$[D] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & DC & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$[U] = \begin{bmatrix} US & 0 & 0 \\ US & UC & 0 \\ 0 & UE & 0 \end{bmatrix}.$$  

This gives two extra entries $RWSW$ and $RENE$ for the rest matrix $R$:

$$[R] = \begin{bmatrix} 0 & RWSW & RW & 0 & 0 \\ 0 & RSW & RW & RN & 0 \\ RSS & RS & RC & RN & RNN \\ 0 & RSE & RE & RNE & 0 \\ 0 & 0 & REE & RNE & 0 \end{bmatrix}.$$  

The recursive construction of $L, D, U$ and $R$ proceeds as follows:

$$LN_{j,k} = AN_{j,k}$$
$$LNW_{j,k} = -LN_{j,k}DC_{j+1,k}^{-1}UW_{j+1,k}$$
$$LE_{j,k} = AE_{j,k} - LN_{j,k}DC_{j+1,k}^{-1}USE_{j+1,k}$$
$$LC_{j,k} = DC_{j,k}$$
$$DC_{j,k} = AC_{j,k} + AN_{j,k} + ASW_{j,k} + ASE_{j,k} + ANE_{j,k} + ANN_{j,k} + AWW_{j,k} + ASS_{j,k} + AEE_{j,k}$$
$$-LN_{j,k}DC_{j+1,k}^{-1}US_{j+1,k} - LE_{j,k}DC_{j,k-1}^{-1}UW_{j,k-1}$$
$$-LN_{j,k}DC_{j+1,k+1}^{-1}USE_{j+1,k+1}$$
\[ UC_{j,k} = DC_{j,k} \]
\[ UW_{j,k} = AW_{j,k} - LN_{j,k}^{-1} DC_{j+1,k+1} \]
\[ USE_{j,k} = -LE_{j,k} DC_{j,k-1}^{-1} U N_{j,k-1} \]
\[ US_{j,k} = AS_{j,k} \]

while
\[ RE_{j,k} = -AE_{j,k} \]
\[ RE_{j,k} = AEE_{j,k} \]
\[ RNN_{j,k} = -ANN_{j,k} \]
\[ RN_{j,k} = ANN_{j,k} \]
\[ RENE_{j,k} = LN_{j,k}^{-1} DC_{j+1,k+1} \]
\[ RNE_{j,k} = -ANE_{j,k} \]
\[ RSE_{j,k} = -ASE_{j,k} \]
\[ RC_{j,k} = ANW_{j,k} + ASW_{j,k} + ASE_{j,k} + ANE_{j,k} \]
\[ RSW_{j,k} = -ASW_{j,k} \]
\[ RNW_{j,k} = -ANW_{j,k} \]
\[ RW_{j,k} = LE_{j,k} DC_{j,k-1}^{-1} USE_{j,k-1} \]
\[ RS_{j,k} = ASS_{j,k} \]
\[ RSS_{j,k} = -ASS_{j,k} \]
\[ RW_{j,k} = AWW_{j,k} \]
\[ RW_{j,k} = -AWW_{j,k} \]

This process requires the inversion of the 4 × 4 block DC. Since the pressure does not appear in the continuity equation, this block has a zero on the main diagonal. The inversion of DC is therefore carried out with a direct method, which for such a small system is a trivial task. Thus the zero’s on the main diagonal of the matrix, which in the literature on solution of the Navier-Stokes equations for incompressible flows is so often introduced as a difficulty, are dealt with in a simple and straightforward manner.

The CILU(ε) allows non-zero entries of the factors L and U outside the sparsity pattern of \( A_{i,j} \). The criterion is here that all non-zero elements of the rest matrix \( R \) must in some norm be smaller than \( \varepsilon \). Denoting the elements of \( A_{i,j} \) and its parts \( L, D^{-1}, U \) and \( R \) by lower case symbols with row and column indices, the complete LU decomposition of \( A_{i,j} \) can be written as

\[ a_{mn} = \sum_{s=1}^{\min(m,n)} l_{ms} d_{ss} u_{sn}. \]

The CILU(ε) method constructs the factors \( L \) and \( U \) block row by block row from

\[ u_{mn} = a_{mn} - \sum_{s=1}^{m-1} l_{ms} d_{ss} u_{sn} \quad \text{for } m \leq n \]
and

\[ l_{mn} = \left( a_{mn} - \sum_{s=1}^{n-1} l_{ms} d_{ss} u_{sn} \right) / d_{nn} u_{nn} \quad \text{for } m > n. \]

If the $L_2$-norm of the right-hand side is greater than a chosen threshold value $\varepsilon$, the fill-in block $u_{mn}$ or $l_{mn}$ is retained and $r_{mn} = 0$. Otherwise, the fill-in block is discarded and the right-hand side is moved to the rest matrix.

The Compact Row Storage format is used to store the entries of $LD^{-1}U$ and $R$. This format has been chosen because it results in easy coding of matrix-vector multiplication. It involves the use of three arrays: coA[1, $\cdots$, Nonzero], containing the data of each non-zero block, jcoA[1, $\cdots$, Nonzero], containing the column number of each such block and begA[1, $\cdots$, $N + 1$], providing the address in coA of the first non-zero entry found on row $m$, supplemented with the address of the last element of coA. Herein is Nonzero the total number of non-zero entries and $N$ is the number of rows of the matrix.

Also here the inversion of the $4 \times 4$ block $d_{nn} u_{nn}$ (pivot block) is needed, which is again done with a direct method.

The CILU($\varepsilon$) method is more general and more robust than the CILU7, but requires a greater computational effort and greater storage.
Chapter 7

Numerical verification

In the foregoing chapters we have described various aspects of a computation method for ship stern flows: the mathematical equations to be solved, their discretisation and the iterative solution procedure for the resulting system of discretised equations. Scrutinizing the method in various applications is the appropriate next step, which will be reported in two stages: verification and validation. Before embarking on a demonstration of the physical realism of the predictions made with the method (validation), we shall devote this chapter to some numerical studies for verification of the global order of accuracy and the consistency of the method.

Among the various methods for the quantification of discretisation accuracy (Roache, 1997), systematic grid refinement is the one that is most commonly used. We shall apply it to a couple of cases, in two and three-dimensional flows. In some instances this will be done with discretisations which are partly of different order. We shall also verify the convergence behaviour of the iterative solution process, and in particular consider the effects of the convergence acceleration techniques presented in Chapter 6, which can be tested in two-dimensional flows. Moreover, the preservation of a good iterative convergence rate of the method in applications at very high Reynolds number needs verification. Finally, the effect of flux limiting in a representative flow simulation will be demonstrated.

7.1 The laminar wake of a flat plate

The laminar flow past a flow-aligned flat plate of finite length is a suitable first case for numerical verification. The geometry is simple and allows the use of rectangular Cartesian grids; boundary layer and triple-deck theories provide a good framework for comparison; and possibly disturbing influences of turbulence modelling are avoided. At the same time it represents the most fundamental form of the type of flows that we wish to simulate.

For the numerical tests we have chosen a Reynolds number based on the length of the plate of $Re = 10^5$. A rectangular coordinate system $x, y$ was adopted with the origin at the leading edge of the plate and the $x$-axis along the plate. The
Numerical simulation of ship stern flows

![Graph showing laminar wake of flat plate]  

Figure 7.1: Convergence behaviour on five grids

computation domain extended in lengthwise direction from $x/L = 0.5$ to $x/L = 1.5$, while the lateral extent was $0 \leq y/L \leq 0.125$, which is approximately 8 times the boundary-layer thickness at the trailing edge. The boundary conditions $u = u_\infty$ and $p = 0$ were applied on the external boundary $y = y_{max}$, while on the inlet boundary a close approximation of the Blasius solution (Schlichting, 1968) was imposed.

Five rectangular grids were generated with the grid points uniformly distributed in $x$-direction, but stretched towards the plate and the wake centreline in normal direction. The grids had respectively $33 \times 33$, $45 \times 45$, $65 \times 65$, $89 \times 89$ and $129 \times 129$ nodes. If the typical grid size of the finest grid is $h$, the covered sequence of grid sizes is: $h, 1.45h, 2h, 2.91h, 4h$.

The convergence histories of the solutions on the various grids are compared in Figure 7.1, where the maximum change of $Cp$ between successive global iterations (i.e. the $L_\infty$ norm of $\Delta Cp$) has been chosen as the measure of convergence. The predictor-corrector method and the approximate multigrid technique have been applied without grid sequencing to generate these results. The convergence is seen to be uniform and fast, but the rate of convergence decreases somewhat with the increase of the grid density.

In Figure 7.2 we have plotted $N_{iter}$, the number of iterations needed to obtain
\[ \Delta C_p \leq 2 \times 10^{-6}, \] against \( N \), the number of grid points in \( x \)-direction. The results denoted by the plus symbols correspond with those of Figure 7.1, which indicates \( N_{\text{iter}} \approx O(N^{0.7}) \). Ideally, multigrid can result in \( N_{\text{iter}} \) being independent of \( N \). This is clearly beyond the reach of the simple approximate multigrid strategy used here; but in view of its easy implementation, the technique yields a satisfactory convergence speed.

Also shown in Figure 7.2 are results which were obtained with application of the predictor-corrector accelerator only (without the approximate multigrid). These appear to give \( N_{\text{iter}} = O(N) \), with a proportionality factor of approximately 0.7. This is a clear indication that the predictor-corrector scheme is superior to one-sided Gauss-Seidel, which in the solution of a Poisson equation on a uniform rectangular grid has a convergence rate proportional to \( N^2 \) (Hageman & Young, 1981). We recall that the solution process without the corrector sweep is a one-sided Gauss-Seidel iteration.

Corroboration of this superiority is given in Figure 7.3, where a comparison is shown of the convergence of the solution on the \( 45 \times 45 \) grid for three strategies: one-sided Gauss-Seidel, symmetric Gauss-Seidel (predictor-corrector) and symmetric Gauss-Seidel plus approximate multigrid. Figure 7.4 serves as an elucidation of the drastic improvement of the convergence behaviour by the corrector. It shows the maximum pressure change for each \( x \)-station during the sequence of global iterations. If the corrector is not applied, pressure waves appear which are only slowly damped. The elimination of these upstream travelling waves is much more effective.
with the corrector activated. The approximate multigrid strategy further enhances
this damping.

By application of grid sequencing we could not further reduce the number of
global iteration cycles needed to reach a certain tolerance level, not even on the
finest grid. But it resulted in slightly smaller computation times because the first
few sweeps require less work.

In Figures 7.1 and 7.3 the change in the pressure variable was chosen to measure
the convergence. This is not unreasonable, since we have indicated in the discussion
of the approximate multigrid technique that the residual of the momentum equation
is dominated by the pressure change. But convergence is properly obtained if it
can be shown that besides variable changes between successive iterations, also the
residuals of the equations being solved tend to zero. Therefore, as an example,
the convergence of the global iteration process on the $65 \times 65$ grid is alternatively
presented in Figure 7.5, showing the maximum residuals of the three equations
per iteration, and the maximum change per iteration of all three variables. The
oscillatory behaviour in some of the curves shown is due to the approximate multigrid
technique, which has been applied every other iteration cycle; the corrector is active
in every iteration.

The residual of the continuity equation does not appear because it is in this
case at machine accuracy level ($10^{-17}$) from the beginning, and thus outside the
range of the figure. By the construction of our iteration scheme, this residual is
only affected by the convergence tolerance of the local or inner iteration process.
In the global iteration process it is therefore orders of magnitude smaller than the
residual of the momentum equation. This is a characteristic feature of PARMASSOS.
While the velocity and pressure fields gradually adjust to satisfy the momentum
equation, all intermediate velocity fields satisfy the continuity equation. As has
been observed before, this is a major distinction from the Marker-and-Cell or the
Artificial Compressibility method.

By numerical experimentation we found that for a fixed size of the computation
domain the convergence rate of the global iteration process is not affected by the
amount of grid stretching in the direction normal to the plate, nor by a variation of
the number of grid points in that direction. On the other hand, the convergence rate
deteriorates somewhat with an increase of the lateral extent of the domain. This is
in accordance with the stability analysis of the predictor-corrector method given in
Appendix B. This analysis makes plausible that the ratio $y_{\text{max}}/h_x$, where $y_{\text{max}}$
is the width of the domain and $h_x$ the grid size in $x$-direction, is a determining factor
for the convergence rate.

From the (sufficiently converged) solutions on a set of three similar grids, varying
in density, the apparent order of accuracy of a numerical method for a particular
test case can be established from

\[
\left( \frac{h_2}{h_1} \right)^n = \frac{(f_2 - f_3) \left( \frac{h_2}{h_1} \right)^n - 1}{(f_1 - f_2) \left( \frac{h_2}{h_1} \right)^n - 1}, \tag{7.1}
\]
Figure 7.3: Convergence behaviour for three strategies on 45x45 grid

Laminar wake of flat plate (45 x 45 grid)

Without corrector

With corrector

Figure 7.4: Maximum pressure change per x-station during iteration process
where $n$ is the order of accuracy, $f_1$, $f_2$ and $f_3$ are representations of the same scalar quantity, obtained as a part of the solutions on the finest, intermediate and coarsest grid respectively, and $h_1$, $h_2$ and $h_3$ are the corresponding representative grid sizes. If the ratios $h_2/h_1$ and $h_3/h_2$ are equal, $n$ follows directly; else $n$ must be determined iteratively.

We like to point out that equation (7.1) assumes that the solution $f_1$, $f_2$ and $f_3$ are in the asymptotic range, which means that the error behaviour is dominated by the leading order truncation term. The validity of this assumption may be hard to assess in practice.

For $f$, any local or integral quantity can be chosen. We have selected four quantities for the order verification of our method, viz. $C_D$, the total drag of the plate, determined from

$$C_D = \frac{2}{u_\infty^2 L} \int_{0}^{y_{\text{max}}} (p + u(u_\infty - u))_{\text{outlet}} dy,$$

(7.2)

$C_{FA}$, the drag of the aft half of the plate (obtained by skin friction integration), $C_{Pte}$, the pressure at the trailing edge and $u_{cl}$, the centreline velocity at the outlet plane. The computed values of these quantities are listed in Table 7.1, while the order of accuracy derived from various grid triplets is presented in Table 7.2.
Table 7.1: Key quantities for order derivation

<table>
<thead>
<tr>
<th>cases</th>
<th>$n(C_D)$</th>
<th>$n(C_{FA})$</th>
<th>$n(C_{Pe})$</th>
<th>$n(u_{cl})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1+3+5</td>
<td>2.52</td>
<td>1.68</td>
<td>1.09</td>
<td>1.75</td>
</tr>
<tr>
<td>1+2+3</td>
<td>2.84</td>
<td>1.61</td>
<td>1.04</td>
<td>1.65</td>
</tr>
<tr>
<td>2+3+4</td>
<td>2.92</td>
<td>1.15</td>
<td>1.12</td>
<td>2.48</td>
</tr>
<tr>
<td>3+4+5</td>
<td>1.70</td>
<td>2.59</td>
<td>1.09</td>
<td>0.46</td>
</tr>
<tr>
<td>1+2+4</td>
<td>2.88</td>
<td>1.43</td>
<td>1.07</td>
<td>2.00</td>
</tr>
<tr>
<td>2+3+5</td>
<td>2.46</td>
<td>1.70</td>
<td>1.11</td>
<td>1.78</td>
</tr>
<tr>
<td>1+3+4</td>
<td>2.90</td>
<td>1.29</td>
<td>1.10</td>
<td>2.26</td>
</tr>
<tr>
<td>2+4+5</td>
<td>1.97</td>
<td>2.23</td>
<td>1.10</td>
<td>0.96</td>
</tr>
</tbody>
</table>

Table 7.2: Apparent order of grid convergence

The results are rather diverse. For the total drag $C_D$ the order seems to be better than 2, but for the frictional drag of the aft half of the plate it falls in the mean below 2. The conjecture that the latter result might be due to using simple midpoint integration for $C_{FA}$ did not come true; with Simpson integration, essentially the same figures for the order of convergence were obtained. Turning to the local quantities $C_{Pe}$ and $u_{cl}$, the result is uniformly 1.1 for the pressure at the trailing edge, but approximately 2 for the centreline velocity at the outlet, provided the results involving the coarsest grid (case 5) are discarded. The low convergence order for the pressure at the trailing edge is not unusual, considering the fact that this quantity is evaluated at the location where it reveals a sharp peak. We verified that with a first-order two-point discretisation of the mainstream pressure gradient the order drops to even lower values (approximately 0.65).

The drag converges to $C_D = 0.0043153$ for $h \to 0$ which is in good agreement with triple deck theory that gives (see Veldman, 1976)

$$C_D = \frac{D}{\frac{1}{2} \rho u_\infty^2 L} = 1.32824Re^{-1/2} + 2.651Re^{-7/8} + O(Re^{-1}) = 0.004312 + O(10^{-5}).$$

Taking $C_{D_{h=0}}$ as a reference, the relative error in the drag obtained on the five grids is 0.014, 0.032, 0.074, 0.211 and 0.419 per cent, respectively.

Additional calculations were made with skewed grids of $45 \times 45$ grid nodes, in
order to detect a possible sensitivity of the convergence rate to non-orthogonality of the mesh. Both forward and backward skew was applied. The amount of skew was chosen to increase from zero at the inlet to 30 degrees at the trailing edge and then to diminish to zero at the outlet plane. The resulting grids are compared with the rectangular grid in Figure 7.6. It turned out that backward skew is favourable for the global convergence, while the opposite holds for forward skew. While on the rectangular grid $\Delta C_{p_{\text{max}}} \leq 2 \times 10^{-6}$ was reached in 23 iterations, on the backward-skewed grid 20 iterations sufficed, while on the forward-skewed grid 37 iterations were needed to attain the same convergence level.

7.2 The flow at the tail of a modified spheroid

For a further verification of our numerical method we move on to the turbulent flow around the aft end of an axisymmetric body. As the body we have chosen a 6:1 prolate spheroid, modified at the tail to avoid flow separation. The modification consists of replacing the blunt aft end by a pointed tail, thus making the body about 3.57 per cent longer.
Figure 7.7: Coarsest grid (49 x 65) for modified 6:1 spheroid

More precisely, the body geometry is given by:

\[
\begin{align*}
  y &= \frac{1}{6} \sqrt{x (0.9655L - x)} & 0 \leq x \leq 0.9333L \\
  y &= 0.4333(L - x) & 0.9333L \leq x \leq L
\end{align*}
\]

The flow around this body at a Reynolds number of \(Re = 1.264 \times 10^6\) was simulated in a domain extending in lengthwise direction from \(x/L = 0.4828\) (the location of maximum thickness of the body) to \(x/L = 1.4483\), and in the radial direction from the axis to \(y_{max}/L = 0.1448\). The boundary conditions on the external boundary \(y = y_{max}\) were derived from a potential flow solution, while on the inlet boundary suitable velocity profiles were prescribed. The algebraic turbulence model was applied.

Five grids were generated in the chosen domain: 49 x 65, 61 x 81, 73 x 97, 85 x 113 and 97 x 129. The coarsest grid is shown in Figure 7.7 (notice that the horizontal and vertical axes are scaled differently); the other grids were similar, but with greater density. The grid stretching towards the body surface was chosen so as to give a maximum \(y_+\)-value based on the thickness of wall-contiguous grid cells of about 1.0 on the coarsest grid.

The apparent order of accuracy was determined for the frictional and the pressure drag of the aft part of the body (i.e. the part inside the computation domain). Results were obtained for the normal QUICK discretisation of the streamwise pressure gradient, but also for the simple two-point downstream discretisation. The results are summarized in Table 7.3, while the derived order for several grid combinations is presented in Table 7.4.

With the standard method (with QUICK discretisation for the mainstream pressure gradient) the order of accuracy of \(C_{fI}\) is about 2, of \(C_{ps}\) about as much as 5. A lower-order discretisation for the pressure gradient results evidently in a lower
Table 7.3: Modified spheroid - Key quantities for order derivation

<table>
<thead>
<tr>
<th>cases</th>
<th>Quick $\partial p/\partial \xi$</th>
<th>2-pnts $\partial p/\partial \xi$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$n(C_{f_x})$</td>
<td>$n(C_{pz})$</td>
</tr>
<tr>
<td>1+2+3</td>
<td>1.55</td>
<td>6.57</td>
</tr>
<tr>
<td>2+3+4</td>
<td>2.87</td>
<td>5.46</td>
</tr>
<tr>
<td>3+4+5</td>
<td>2.00</td>
<td>4.53</td>
</tr>
<tr>
<td>1+3+5</td>
<td>2.34</td>
<td>5.18</td>
</tr>
</tbody>
</table>

Table 7.4: Modified spheroid - Apparent order of grid convergence

Figure 7.8: $C_{f_x}$ and $C_{pz}$ as functions of grid density
order of accuracy. But the solutions for both discretisations plausibly converge to the same result for $h \to 0$, as illustrated in Figure 7.8. This figure shows the results plotted on a basis of the characteristic cell size $h$, taking $h = 1$ for the finest grid; moreover, for each data set a close fit is given by a simple power law formula, which reflects the mean of the accuracy orders of Table 7.4. The same figure might also give a clue to why the order of accuracy of $C_{px}$ is unexpectedly high: the pressure drag value happens to be almost independent of the grid resolution. The evaluation of the order of accuracy becomes very sensitive then to minor deviations in the data; at the same time, its significance for the uncertainty in the result becomes small. Objectionable for such an explanation is, however, that the order for $C_{px}$ is at the same high level for all grid combinations. Maybe, we rather have to explain the result with a fortunate cancellation of truncation errors, $C_{px}$ being determined by negative contributions near the tail and positive contributions further upstream.

Taking the extrapolated values for $C_{fz}$ and $C_{px}$ as a reference, the numerical errors of the various solutions can be directly inferred from Fig. 7.8.

![Modified spheroid](image)

Figure 7.9: Convergence behaviour for various Reynolds numbers

The accuracy study was made for the flow at a Reynolds number typical for model testing in a towing tank or a wind tunnel. But our purpose is also to be able to analyse the prototype flow, characterised by a Reynolds number which may be two orders of magnitude larger. It is relevant therefore to examine the behaviour of the solution process with increasing Reynolds number. Accordingly, we have calculated the flow around the modified spheroid for $Re = 1.264 \times 10^7$ and $Re = 1.264 \times 10^8$ in addition to $Re = 1.264 \times 10^9$. The grids used were similar but the number of cells in radial direction was slightly increased with higher $Re$ ($73 \times 97, 73 \times 115, 73 \times 129$...
nodes respectively); the stretching of the grid towards the body surface was chosen so that the $y_4$-value, based on the thickness of the grid cells adjacent to the body, was about the same in all three calculations and not exceeding 1.0. This implies for the highest Reynolds number a smallest cell thickness of about $2 \times 10^{-7} L$ and a maximum grid cell aspect ratio of $\sim 10^5$.

The calculations were made with all three acceleration techniques applied. Grid sequencing was started with a fourfold stepsize; the approximate multigrid was applied every third iteration cycle.

We compare the convergence histories of the three cases in Figure 7.9, where again the $L_{\infty}$-norm of the pressure change between successive global iterations is used as the convergence measure. It appears that the convergence rate is hardly affected by the Reynolds number, which is a promising result. As far as there is a change, the convergence rate even improves with increasing $Re$. It may be worthwhile to point out that the jumps in the convergence rate curves in the initial phase of the iteration process correspond with grid refinement steps in the grid sequencing technique.

The computation times came out as directly proportional to the number of grid cells, so that also the time spent in the local iteration process (almost a direct solver) does not significantly change with $Re$.

### 7.3 The flow near the stern of the Wigley hull

For numerical verification in a 3D application, we analyse the flow at the stern of the Wigley hull, a well-known test case in numerical ship hydrodynamics. It is a ship-like form defined by the simple mathematical offset function:

$$y = \frac{B}{2} \left[ \frac{4x}{L} \left( 1 - \frac{x}{L} \right) \right] \left[ 1 - \left( \frac{z}{H} \right)^2 \right],$$

with $x$ and $z$ in the range

$$0 \leq x \leq L \quad -H \leq z \leq 0,$$

while

$$B = 0.1L \quad H = 0.0625L.$$

An impression of the hull shape can be obtained from Fig. 7.10, which shows half of the hull (fore and aft body are symmetric).

The flow around this slender and sharp-keeled body was simulated at a Reynolds number of $Re = 7.4 \times 10^6$. The computation domain, extending in lengthwise direction from $x/L = 0.52$ to $x/L = 1.48$, was bounded externally by the relevant part of the elliptic cylinder

$$(y/0.07)^2 + (z/0.09)^2 = L^2; \quad x \text{ arbitrary.}$$
The boundary conditions imposed on the external boundary were derived from a potential flow solution, while suitable velocity profiles were prescribed on the inlet boundary.

Computations were made on four grids, generated by stacking 2D grids obtained with the Schwarz-Christoffel technique described in Chapter 3, with $I \times J \times K = 49 \times 49 \times 25$, $65 \times 65 \times 33$, $81 \times 81 \times 41$, $97 \times 97 \times 49$ nodes, respectively. The grid node distribution on the domain boundaries is displayed for the coarsest of these grids in Figures 7.11 and 7.12. The grid line $j = 1, k = 1$ is a singular line; $\sqrt{g}$, the Jacobian of the transformation, vanishes there.

It may be worthwhile to mention that the grids were somewhat coarse near the body surface. The maximum $y_+$ value associated with the thickness of the grid cells adjacent to the body was about 3 for the finest grid, so about 6 for the coarsest grid.

The convergence history, measured again by the $L_\infty$ norm of the pressure change between successive iterations, is shown in Figure 7.13. As revealed by the plot, we have applied grid sequencing in three stages, the finest grid being reached in the 9th iteration on all grids. Also the predictor-corrector and the approximate multigrid methods were applied. The results given are for the algebraic turbulence model; with Menter's one-equation model the convergence rate is slightly worse.

The apparent order of accuracy was determined for the frictional and the pressure drag of the aft part of the body (i.e. the part inside the computation domain). Results are presented for both turbulence models (described in Chapters 2 and 4) in Table 7.5. The derived orders for several grid combinations are presented in Table 7.6. Although the results for both models are monotonous on a basis of the characteristic cell size, the convergence order is quite different for the frictional
Figure 7.11: Grid planes $i=1$, $j=1$, $k=1$ for Wigley hull

Figure 7.12: Grid planes $i=I$, $j=J$, $k=K$ for Wigley hull
resistance. Suspecting that conditional statements used in the determination of the outer length scale might be responsible for the low order of the algebraic-model results, we examined the value of the maximum eddy viscosity per \( \xi \)-station on the four grids, but they turned out to differ only near the stern and in the near wake; the differences being very systematic, no clue for the deterioration of the convergence order was obtained. Maybe the cause is to be found in the rather coarse grid near the hull surface used in these calculations.

For the Wigley hull we shall also demonstrate the effect of flux limiting, implemented as described in Chapter 5 and Appendix A. In the same computation domain as used above we generated a \( 81 \times 65 \times 33 \) grid. To obtain circumstances in which flux limiters may become effective, we applied an external force field to represent a propeller. The propeller disk was positioned at \( x/L = 1.018 \) and the propeller radius chosen as \( R_p/L = 0.02 \), with the propeller axis located at \( y_\alpha/L = 0 \); \( z_\alpha/L = -0.04 \). We applied a total thrust of

\[
C_T = \frac{F}{\frac{1}{2} \rho U_\infty^2 \pi R_p^2} = 0.45,
\]
Table 7.5: Wigley hull - Key quantities for order derivation

<table>
<thead>
<tr>
<th>cases</th>
<th>Algebraic model</th>
<th>One-equation model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( n(C_{fx}) )</td>
<td>( C_{fx} \times 10^{4} )</td>
</tr>
<tr>
<td>1+2+3</td>
<td>0.53</td>
<td>2.17</td>
</tr>
<tr>
<td>1+2+4</td>
<td>0.11</td>
<td>2.01</td>
</tr>
<tr>
<td>1+3+4</td>
<td>-0.23</td>
<td>1.90</td>
</tr>
<tr>
<td>2+3+4</td>
<td>-0.50</td>
<td>1.83</td>
</tr>
</tbody>
</table>

Table 7.6: Wigley hull - Apparent order of grid convergence

which represents a condition close to self-propulsion (zero momentum deficit in the far wake). The thrust distribution on the propeller disk was assumed to be circumferentially uniform, but radially varying as

\[
F(r) = F_0 \sqrt{\frac{r}{R_p}} \left(1 - \frac{r}{R_p}\right),
\]

with

\[
r = \sqrt{(y - y_a)^2 + (z - z_a)^2}.
\]

The convergence history of the solution process was not seriously affected by the inclusion of external forces: the \( L_\infty \) norm of the pressure change between successive iterations followed the line indicated for the 81 \( \times \) 81 \( \times \) 41 grid in Figure 7.13.

To illustrate the effect of flux limiting, we have selected some results in a transverse plane behind the propeller. In Figure 7.14 the axial velocity along a set of gridlines \( \zeta = \text{constant} \) is plotted on the basis of the transverse coordinate \( y \). The selected gridlines are indicated in the subfigure inserted at the right, which also shows the size of the propeller disk. The velocity profiles reveal the jet generated by the propeller, with a sharp change of the velocity gradient at the edge of the jet. The results without flux limiting show an undershoot near the jet edge, while the results with activated flux limiting are free of oscillatory behaviour. Otherwise, the solutions are virtually the same.
Wigley hull with propeller

Figure 7.14: Example of the effect of flux limiting; axial velocity profiles along six successive $\zeta = constant$ lines as indicated
7.4 Summary of results

We briefly summarize the results of the verification studies here as follows:

1. In all tests, reduction of variable changes or residuals by several orders of magnitude was achieved without any indication of stagnation of the convergence process.

2. Effectiveness of the three convergence acceleration techniques introduced in Chapter 6 was demonstrated.

3. The convergence speed of the global iteration process depends on the number of grid cells in the mainflow direction and on the relative size of the domain width (which is confirmed by a linear stability analysis); it is also affected by grid skewness. On the other hand it is insensitive to the number of grid nodes in the transverse direction, and to the Reynolds number.

4. The formal second order accuracy of the method has reasonably been confirmed by the observed order in the three tests, at least away from extrema. The observed order is not the same for all aspects of the solution, though. Also indications have been found that in 3D turbulent flow simulations the order of grid-convergence can easily deteriorate.

5. The implementation of flux-limiters has successfully been verified.
Chapter 8

Application and validation

Where in the previous chapter the emphasis has been put on the numerical behaviour of the computation method, we will now rather concentrate on the quality of the solution as a correct representation of the physics, through the comparison of computed results with experimental data.

Our main purpose is to demonstrate the performance of the method in the simulation of ship stern flows. But it is worthwhile and informative to consider also simpler flows of essentially the same kind, if only because detailed experimental flow-field data for ships are scarce. We start therefore with the prediction of the flow around two-dimensional and axisymmetric bodies. The cases of a flow-aligned flat plate, a 6:1 spheroid with flow separation at the tail, and a slightly modified form of the same body, being free of separation, will be considered. For all cases comparison with experimental data is possible. After that, we shall present results of three applications to a single tanker hull form. The results can partly be validated against measurements. For another part they are meant as an illustration of the scope of problems that might be analysed with the current method.

8.1 Two-dimensional and axisymmetric flows

8.1.1 Flat plate, laminar flow

The laminar flow at the rear end of a flow-aligned flat plate has been considered already in Chapter 7. There the lateral extent of the computation domain was set at $y_{max} = 0.125L$, corresponding to about eight boundary layer thicknesses. That sufficed for numerical experimentation, but if physical realism comes on the foreground a greater extent may be required. Triple-deck theory indicates that the viscous-inviscid interaction at the tail is felt in a region with an extent of $O(Re^{-3/8})$, which goes well beyond the boundary layer thickness, being of $O(Re^{-1/2})$.

The influence of the lateral extent of the computation domain on the results is illustrated with the most sensitive variable, the pressure, in Figure 8.1. It shows the computed pressure on the plate surface and along the wake centreline for various
domain widths. For the smallest domain width, the pressure reaches a practically undisturbed level both at the upstream and the downstream end of the computation domain. For larger domain widths, the pressure interaction seems to extend over a greater length.

We have chosen $y_{\text{max}}/L = 0.25$, $Re = 10^5$ and a $65 \times 65$ grid for the final computations, the results of which will now be compared with data from other sources. These sources are the classical Blasius solution (Schlichting, 1968) and the Goldstein wake (Goldstein, 1930), Veldman’s numerical solution of the triple-deck equations (Veldman, 1976) and the results of an Interactive Boundary Layer (IBL) solution, also by Veldman (1979).

The development of the skin friction on the plate is shown in Figure 8.2. Initially, our numerical solution follows the Blasius line quite well. But on approaching the trailing edge, the flow near the plate surface accelerates and the friction increases accordingly. Considering the fact that in the tripledeck theory the solution near the trailing edge is matched with the Blasius solution at $x = -\infty$ and in our results at $x/L = 0.5$, the difference between the two results is plausible.

For the velocity on the wake centreline (Fig. 8.3), the correspondence between the tripledeck solution and our results is quite good. The Goldstein solution, omitting
Chapter 8: Application and validation

Figure 8.2: Development of wall friction coefficient

Figure 8.3: Velocity on wake centreline
Figure 8.4: Development of displacement thickness

Figure 8.5: Pressure on the plate and on the wake centreline
the pressure interaction at the trailing edge, clearly deviates from the other solutions. As a further comparison, the displacement thickness of the shear layer along the plate and in the wake is displayed in Figure 8.4. The sharp peak at the trailing edge in the Blasius/Goldstein solution is smoothed in the other three solutions. As for the wall friction, our solution for the displacement thickness follows initially the Blasius solution and there is some deviation from the triple-deck and IBL solutions. Further downstream the three solutions merge quite well.

The pressure distribution along the plate and the wake centreline, displayed in Figure 8.5, has the right trend in that it reveals a depression at the trailing edge, followed by an overshoot above the undisturbed level in the near wake. The overshoot is somewhat greater than in the tripledeck and IBL solutions, but as we have seen in Figure 8.1 the domain width has a significant influence on this result. We have not pursued this further, but a better quality boundary condition on the external boundary as in the IBL method seems a proper way to make the pressure less dependent on the domain width. In passing it may be noted that introduction of a finite thickness of the plate is sufficient to cause the underpressure at the trailing edge to be altered to an overpressure.

### 8.1.2 Flat plate, turbulent flow

If the Reynolds number for the flow along a flat plate is sufficiently high, the boundary layer on the plate and the evolving wake are fully turbulent. The viscous-inviscid interaction at the rear end of the plate is then in contrast to the laminar flow case confined to a region completely embedded in the boundary layer. In attempting to simulate this flow, it is justified therefore to limit the lateral extent of the computation domain to two or three boundary layer thicknesses.

Several independent experiments have been carried out on the symmetric turbulent wake of a flat plate. Best known are those by Chevray & Kovasznay (1969), Andreopoulos (1978), Ramaprian et al. (1982) and Nakayama & Liu (1990). We have chosen the data of Ramaprian et al. for comparison. They used a 1829 \( \text{mm} \) long and 635 \( \text{mm} \) wide plate of 19 \( \text{mm} \) thickness in a closed-circuit wind tunnel. The plate was tapered at the tail with a half included angle of 0.9 degrees. With a wind speed of 22\( m/s \) they achieved a Reynolds number based on the momentum thickness in the wake of \( Re_\theta = 5220 \). A total-head tube was used for the measurement of the mean flow and an x-wire anemometer for mean flow and turbulence.

For the numerical simulation of this flow we have assumed the plate to be infinitely thin and have taken a rectangular computation domain with an extent 0.6 \( \leq x/L \leq 1.4, 0 \leq y/L \leq 0.05 \). The domain was covered with a \( 81 \times 81 \) grid, stretched towards the plate and the wake centreline, as well as towards the trailing edge. The Reynolds number was set at \( Re = 3 \times 10^6 \) and appropriate inflow conditions were used to achieve the \( Re_\theta \) of the experiment in the wake. The algebraic turbulence model was applied.

As a result, the development of the wall friction coefficient along the plate is
shown in Figure 8.6. Well ahead of the trailing edge the behaviour of a twodimensional zero-pressure-gradient boundary layer must appear. This is confirmed by the good correspondence between our skin friction data and the Schultz-Grunow correlation line for the local skin friction (see Schlichting (1968), p. 604):

$$C_f = 0.370(10 \log Re_x)^{-2.584}.$$ 

Moreover, in a 2-D equilibrium turbulent boundary layer the ratio of the friction coefficient $C_f$ and the parameter $(H - 1)^2/H^2$ – where $H$ is the shape factor of the velocity profile: $H = \delta_z/\theta$ – is constant and approximately equal to $0.047 \pm 0.001$ for zero-pressure-gradient flow (East & Sawyer, 1979). Therefore, with a correct representation of the velocity profiles upstream of the plate’s trailing edge, also the quantity

$$G = 0.047 \left( \frac{H - 1}{H} \right)^2$$

should be close to the friction line.

The data for $G$ derived from our results are displayed in Figure 8.6. They are in excellent agreement with the wall friction derived from the velocity gradient at the wall up to about $x/L = 0.75$. Closer to the trailing edge, the equilibrium state gets lost. While the skin friction rises above the plate line, the value of $G$ drops quickly as a result of the adjustment of the displacement thickness to the transition
of no-slip to free-slip at the trailing edge, similar to what we have seen in laminar flow.

Figure 8.7 shows the development of the velocity on the centreline of the wake. Comparison is made with both the pitot tube and the hotwire measurements, which unfortunately differ substantially close to the trailing edge.

The same data are shown in a different form in Figure 8.8. The velocity, nondimensionalised with the shear velocity \( u_r \), is plotted against \( x_+ = (x - x_0)u_r/\nu \). Theoretical analyses indicate that \( u_+ \) is a linear function of \( x_+ \) in the near wake, which comes out quite well in our results.

### 8.1.3 Modified spheroid

Wind tunnel tests have been carried out by Patel, Nakayama and Damian (Patel et al., 1974) on an axisymmetric body with the shape of a slightly modified 6:1 prolate spheroid. The modification is restricted to the tail part. The spheroid was truncated at a section 2 inches from the tail and the cut-off was replaced by a conical tail piece, such that the surface slope (but not the curvature) is continuous at the junction. The modification was introduced because the true spheroid has flow separation at the aft end. We will deal with the true spheroid in the following section, but study first the behaviour of our computational method on the simpler flow around the modified spheroid.

The experiments were carried out in a closed-circuit low-speed wind tunnel having a test section of 7.32 m long with a 1.52 m octagonal cross section. The model of the modified spheroid had a length of \( L = 1.578m \) with a maximum diameter of \( D = 0.254m \). It was mounted at the center of the tunnel by wires attached to the nose and the tail of the body. The maximum blockage was less than 3 per cent. The boundary layer was tripped at 0.3\( D \) from the nose. The tests were conducted at a Reynolds number \( U_{\infty}L/\nu = 2.75 \times 10^6 \).

The acquired data comprise pressure distributions, mean velocity profiles and Reynolds stresses at 7 stations in the region \( 0.662 < x/L < 0.99 \). Unfortunately, no measurements were made in the near wake. The quality of the experimental data has been reviewed by Patel (1980). There is an apparent mistake in the reference pressure, bringing the non-dimensional total head outside the boundary layer to a level of 1.06 instead of 1.0. The experimental data presented here have been corrected accordingly.

The flow around the modified spheroid was simulated at the Reynolds number of the experiment on a domain extending in lengthwise direction from \( x/L = 0.4828 \) (the location of maximum thickness of the body) to \( x/L = 1.4483 \), while the lateral extent was \( 0 \leq y/L \leq 0.1448 \). The boundary conditions on the external boundary \( y = y_{\text{max}} \) were derived from a potential flow solution, while on the inlet boundary suitable velocity profiles were prescribed. The grid had 73 \( \times \) 97 nodes and is one of the four grids used in section 7.2. At the inlet as well as on the external boundary, the velocities resulting from a potential flow solution were adopted as the boundary
Figure 8.7: Velocity on wake centreline

Figure 8.8: Scaled representation of the centreline velocity
Figure 8.9: Wall friction coefficient

Figure 8.10: Wall pressure coefficient
conditions. The one-equation turbulence model was used.

The wall friction coefficient along the stern part is presented in Figure 8.9. Both the direct Preston tube measurements and the friction data derived from Clauser plots of the velocity profiles are shown as experimental data. The correspondence between measurements and calculations is fair and shows the right trend close to the tail.

Figure 8.10 shows the pressure distribution on the tail part of the body. Computed and measured results fit quite well. For reference, the data are also compared with a potential flow solution, to illustrate the viscous effect on the pressure distribution. Viscosity has evidently a smoothing effect on the pressure distribution. Since the pressure drag of the body in potential flow vanishes (paradox of d’Alembert), the viscous-flow pressure drag can be derived from an integration of the difference of viscous and potential flow pressures. If done so, there is a negative and a positive contribution, because the viscous flow pressure is first above the potential flow pressure before it falls far below it close to the aft end of the body. The positive contribution is dominating.

8.1.4 Spheroid

The flow along the unmodified 6:1 spheroid has been measured by Chevray (1968). The results form the most elaborate data set available for axisymmetric bodies. The measurements were carried out in the same wind tunnel as used for the modified spheroid of the previous section. The model was 1.524 m long with a maximum diameter of $D = 0.254m$. The Reynolds number was $Re = 2.75 \times 10^6$, while boundary layer tripping was applied as on the modified spheroid at $0.3D$. All measurements were made with hotwire anemometers and the results comprise the mean flow velocity and the four non-zero Reynolds stresses at 12 axial stations in the range $-0.042 < x/L < 3.0$.

An interesting feature of this flow is that a small zone of recirculating flow occurs at the tail. Since the detection of flow separation is an important objective of the application of our computational method, this test case allows a verification of our method in that respect.

![Figure 8.11: Grid for 6:1 spheroid](image)
In the numerical simulations presented so far, we have considered trailing edge or tail flows only. For the spheroid, the flow around the front part is included as well by locating the inflow station ahead of the bow. An impression of the H-type grid that we used can be obtained from Figure 8.11, where only every other grid line has been plotted with different scales for $x$ and $y$. So the grid actually used was twice as dense in both directions, while the $y$-scale has to be halved in Figure 8.11 to get the grid in true proportions. The computation domain extends from $x/L = -0.15$ to $x/L = 1.5$, while its radius is $y/L = 0.15$. The grid has $169 \times 67$ nodes. At the inlet as well as on the external boundary, the results of a potential flow solution have been employed to derive the boundary conditions.

The one-equation turbulence model was used. The laminar-turbulent transition of the boundary layer was initiated at the given position of the tripping device, at $0.05L$.

![Figure 8.12: Viscous effect on pressure distribution](image)

First of all we compare the pressure distribution on the body and along the axis, upstream and downstream of the body, with a potential flow solution (Figure 8.12). As is to be expected on physical grounds, the departure of the viscous-flow pressure from the potential flow results is confined to the tail region. The pressure peak at the aft end in the potential flow solution is smoothed in the viscous flow solution. The pressure at the front stagnation point is somewhat overpredicted; the H-type grid requires further refinement to bring the pressure coefficient to 1.0.

In the experiment the recirculation zone or separation pocket was found to be in a fairly stable position and is reported to start at 0.12 body diameter ahead of the
Figure 8.13: Velocity vectors and streamlines in computed separation zone

Figure 8.14: Comparison of computed and measured velocity profiles
tail \((x/L = 0.98)\) and to close at 0.10 body diameter downstream of the tail \((x/L = 1.017)\). Figure 8.13 shows the computed recirculation zone\(^1\). We found separation starting at \(x/L = 0.984\), while the recirculation zone closes at about \(x/L = 1.011\), to make the lengthwise extent marginally smaller than in the experiment.

The velocity profiles at several stations are compared in Figure 8.14. Upstream as well as downstream of the flow separation zone, there is a good agreement between measurement and computation. Results obtained with the algebraic turbulence model, not shown here, revealed considerably greater deviations from the experimental data at and just downstream of the separation pocket.

## 8.2 Three-dimensional flows

Now we come to our purpose of applying the calculation method to true ship forms and of demonstrating satisfactory correspondence between numerical predictions and experimental data. There are of course experimental data for many ships, certainly for model scale hulls. Of special interest for validation of a flow prediction tool are the velocity measurements, carried out practically as a standard part of model testing for any ship, in the region where the propeller is to be fitted. However, these measurements are confined to a relatively small circular area in a plane transverse to the propeller axis and include only some 200-250 data points. Although useful, it would not be enough for an over-all examination of the quality of the flow predictions.

Unfortunately, flow measurements covering a greater part of the flow field in sufficient detail are scarce. One of most extensive and the best documented data sets is the one obtained at the Institut für Schiffbau in Hamburg on a model of a tanker, which is commonly referred to as the HSVA tanker. It has been used as a test case in two workshops (Larsson et al., 1991 and Kodama et al., 1994) and will be adopted here as well.

### 8.2.1 HSVA tanker at \(Re = 5 \times 10^6\)

The bodyplan (transverse sections of fore and aft body) of the HSVA tanker is given in Fig. 8.15. It shows a typical, conventionally shaped tanker hull form with a block coefficient \(^2\) of 0.85. It was designed to have as its main dimensions: \(L = 253m, B = 19.165m, T_a = T_f = 14.2m\).

The measurements were carried out in a wind tunnel on a 1 : 95 double model of the underwater part of the ship. The wind tunnel is of the slotted-wall type and

---

\(^1\)The streamline tracing algorithm is slightly inaccurate in that it produces a spiraling streamline in the separation zone, of course, in 2D and axisymmetric flows streamlines in a separation zone should form closed loops.

\(^2\)The block coefficient is the ratio of displacement volume and the product of length, breadth and draught of a ship.
Figure 8.15: Bodyplan of HSVA tanker

Figure 8.16: Boundary grid for HSVA tanker
has a test section with a length of 5.5m and a diameter of 1.2m. The free stream turbulence level in the tunnel is reported to be less than 1 per cent. With a nominal wind speed of about 27m/s a Reynolds number of $5 \times 10^6$ was achieved. The double model of the hull was supported by a cylindrical rod, protruding from the rear end of the double model, and by 1.2 mm steel wires attached to the bottom of the hull.

Besides pressure and skin friction measurements on the hull surface, three-component velocity measurements were carried out with a five-hole Pitot tube in about 11,000 points, most of them in the vicinity of the stern. Later, these data were supplemented with LDV velocity recordings in a small region close to the propeller location. Also an oil flow test has been carried out to visualize the shear stress direction on the hull surface (limiting streamlines). The measurements have been reported by Hoffmann (1976) and Wieghardt and Kux (1980). It may be mentioned that the velocity field measurements are locally affected by the wake of the support wires. This is well known by insiders, but worthwhile to remember when the comparison between measurements and computational results comes up for discussion.

For the numerical simulation of this flow we have taken a computation domain with a lengthwise extent $0.5 \leq x/L \leq 1.25$. The PDE technique described in Chapter 3 was applied to generate the grid, which has $137 \times 81 \times 37$ nodes. Figure 8.16 shows a part of the resulting grid on the boundaries $j = 1, k = 1$ and $k = K$. Menter's one-equation turbulence model was applied, including the $S - |\omega|$ correction (Chapter 4).

Before starting with a detailed comparison of measured and calculated results, a brief global description of the flow behaviour may be helpful to set the stage. Since the flow around a ship hull can be classified as pressure-driven (in contrast to shear-driven), the pressure distribution on the hull surface is shown in Fig. 8.17 for reference. The HSVA tanker has a long parallel midbody, i.e. a midpart with invariant cross sections, which makes that the three-dimensionality of the flow field introduced at the bow diminishes further downstream, and quasi two-dimensional flow conditions are more or less restored at midship. As a result, the variation of the boundary-layer thickness along the midship frame line is small. Further downstream, as soon as the hull cross-sectional area starts to decrease the boundary layer thickness grows rapidly due to the adverse pressure gradient and an over-all lateral streamline convergence. Lateral pressure gradients cause crossflow to reappear because low-momentum fluid in the inner part of the boundary layer is deflected more strongly than the high-momentum fluid further out. These pressure gradients cause also the growth rate of the boundary layer to be unevenly distributed. Along the keel, the boundary layer remains comparatively thin, an order of magnitude thinner than further up. Close to the stern, where a low pressure zone is found in the bilge region, the inner-fluid deflection is so strong as to create a discrete vortex within the boundary layer. This so-called "bilge vortex" dominates the structure of the velocity field in the propeller disk area.

The reader can find confirmation of this global exposition of the flow behaviour in the elaborate graphical presentation of experimental and computational results.
given in Appendix C, which will now be discussed. We start this discussion with examining the comparison of measured and computed limiting streamline patterns on the hull (Fig. 8.18).

The photograph of the experimental results reveals a rather complicated pattern near the stern. There is a confluence of limiting streamlines in the bilge region, a feature that is quite common for single-screw ships, giving rise to the formation of a longitudinal vortex. There is another S-shaped line where limiting streamlines from different directions meet, demarcating a reversed-flow zone, and apparently merging with the first concentration line. The computed result reproduces these experimental observations quite well, although the extent of the lower part of the reversed-flow area is exaggerated and the first (horizontal) confluence line seems slightly misplaced. The latter is common to all computed limiting streamline patterns published so far for this hull. Indeed, if we make an over-all comparison of our results with those presented by others at the Tokyo workshop and in the open literature, we conclude that the predictions with the (modified) Menter model can easily compete in quality with those of two-equation models or second-moment closure models.

To give an idea of the role of the turbulence model in this result, we compare the limiting streamlines computed with the one-equation model and with the algebraic model in Fig. 8.19. Also the algebraic model predicts a reversed flow zone, but the separation line misses the S-shape. Moreover, the saddle point in the upper
Figure 8.18: Comparison of experimental (top) and computed (bottom) limiting streamline patterns for HSVA tanker ($Re = 5 \times 10^6$)
region has moved in upstream direction, worsening the correspondence with the experiment.

Let us next turn to the axial velocity field. A full series of pictures, comparing experimental and computed results, is given in Appendix C (Figs. C.1 through C.10). Here only the results at $x/L = 0.940$ and $x/L = 0.976$ are reproduced. Striking in Fig. 8.20 is the strong lateral variation of the shear layer thickness. This feature is well reproduced in the computations, the effect of the support wires on the measurements being taken into account. Further downstream, where a bilge vortex has developed, and the flow has passed the stern post, the correlation between calculation and experiment is apparently not perfect but still acceptable (see Fig. 8.21). In qualifying the comparison as acceptable, the uncertainty in the experimental data has been taken into consideration also. Although no precise estimate of the accuracy is known, the level of uncertainty of the Pitot tube data is likely to increase with the reduction of the flow speed. Even more relevant is the question by how much the bulging of the iso-velocity lines in the experiment may have been exaggerated by the support wire wake. The disturbance of the isovals for $u/U_\infty = 0.8$ and 0.9 seems to indicate that the velocity might locally be wrong by as much as $0.1U_\infty$. 
Figure 8.20: Axial velocity field at x/L=0.940. Comparison of experimental (top) and computed (bottom) results.
Figure 8.21: Axial velocity field at $x/L=0.976$. Comparison of experimental (top) and computed (bottom) results.
The Menter one-equation model is thus seen to give fairly realistic results for the HSVA tanker. In fact, more realistic than the $k - \varepsilon$ model, from which it has been derived. Results obtained with the $k - \varepsilon$ model have been reported for example by Sotiropoulos and Patel (1995) and by Hoekstra and Eça (1998). In spite of the different grids and the different solution methods used in these references, the predicted velocity fields near the stern are very similar, yet correspond not as well with the measurements as the present results. The $k - \varepsilon$ model is well-known to underestimate the effects of an adverse pressure gradient. As pointed out by Menter (1997), the direct proportionality between the turbulent shear stress and the turbulence kinetic energy, as assumed in his one-equation model, is better confirmed by measurements of flows in adverse pressure gradient than the relation implied by the $k - \varepsilon$ model.

In Chapter 2 we have discussed the general shortcoming of standard turbulence models in predicting the near wake of a ship if longitudinal vortices are dominantly present. Also we have chosen there—in order to improve the predictions—to use an empirical correction of the production term in the transport equation for the eddy viscosity, instead of attempting second-moment closure. The "vortex correction", as we called it, was presented in Chapter 4 and has an empirical coefficient $c_{vor}$. The results discussed above were obtained with $c_{vor} = 4$. To show the effect of the correction, we have plotted in Fig. 8.22 the axial velocity field at $x/L = 0.976$, comparing results for $c_{vor} = 0$, i.e. no correction at all, and $c_{vor} = 4$. We observe that the correction has a local effect; outside the region in the immediate vicinity of the longitudinal vortex there are no significant changes in the solution. Moreover, the local effect gives the desired trend and, with $c_{vor} = 4$, brings the results in better agreement with the experiment. It is encouraging that this value corresponds well with the 3.5 to 4 recommended by Dacles-Mariani et al. (1995).

A detail of Fig. 8.22, showing also the result for $c_{vor} = 2$, is included in Appendix C as Fig. C.33. It shows clearly the tendency of the correction to narrow the wake in the top part and to increase the bulging of the iso-velocity contours near the vortex core.

The transverse velocity field is shown as vector plots at several stations in Figures C.11 through C.20 in Appendix C. The series of pictures reveals the development of a discrete longitudinal vortex, which has well established itself at $x/L = 0.976$, i.e. the approximate propeller position. The center of the vortex is at a height of about $z/L = -0.04$ in the stations shown, but further downstream the core position drifts slowly away from the symmetry plane and, by induction of the symmetrical companion vortex on port side, also downward. While the calculations reveal a single discrete vortex (considering the starboard side only), the experimental data seem to show a double (co-rotating) vortex structure, the two vortices merging into a single vortex only at $x/L = 1.042$.

In the contour plots of the computed longitudinal vorticity (Figure C.32) there is not more than a weak indication of a double structure by the appearance of two peaks at station $x/L = 0.967$. In Wieghardt & Kux (1980), plots are given of
the longitudinal component of the vorticity vector, derived from the measurements. After non-dimensionalisation of their data, our computations appear to yield about the same level of longitudinal vorticity.

Contour plots of the computed eddy viscosity distribution are displayed in Figure C.31. In spite of suppressing the eddy viscosity locally by the vortex correction, we find its maximum value near the vortex core from station $x/L = 0.97$ on.

8.2.2 HSVA tanker at $Re = 2 \times 10^9$

Computational analysis of the flow around a ship's hull at model scale, as dealt with in the previous section, is relevant, both for validation purposes and for better understanding and interpretation of what happens during model testing. But in the end, the flow behaviour around a ship at full size is what matters, and that will generally differ from what is found on model scale, primarily due to a difference in Reynolds number. Model testing, currently the best option for predicting the performance of the actual ship, depends on the quality of the scale effect corrections applied. It is one of the great potential benefits of numerical predictions that they have no fundamental restrictions in setting the Reynolds number at any required
level. Deliberately, we speak of "fundamental restrictions", because there may well be practical restrictions. Many numerical solution procedures suffer from performance deterioration with increasing Reynolds number. As a matter of fact, only a few applications of RANS codes to ships at full-scale Reynolds number have been published. Ju and Patel (1991) have reported results for an axisymmetric body and for the HSVA tanker, but they alleviated the problem considerably by using wall functions, thus avoiding the resolution of the inner part of the boundary layer. Only Eca and Hoekstra (1996; with another version of PARNASSOS) and later Watson and Bull (1998) have so far succeeded in calculating the flow down to the wall for various Reynolds numbers up to $Re = 10^9$. But Watson and Bull, presenting results for four different Reynolds numbers, reported that they were unable to use a grid with the stretching required for the highest $Re$, due to numerical problems. It is important therefore that a numerical method intended for use in a ship design process can prove at least to have no difficulties in simulating flows at the very high Reynolds numbers implied by the actual operating conditions of the ship. Moreover, although the results of such computations cannot really be validated, in the absence of experimental data, we can show that they give at least qualitatively the trends expected on physical grounds.

To that end, we shall recalculate the flow around the HSVA tanker for a Reynolds number, typical for full-scale conditions. For a ship's length of 253$m$ and a representative advance speed of about 15 knots, we obtain $Re = 2 \times 10^9$. This is 400 times larger than the Reynolds number of the wind tunnel tests, and the grid density near the hull surface must accordingly be increased to get sufficient resolution. Taking the grid used in the previous section as a starting point, we increased the stretching of the grid towards the hull surface and augmented at the same time the number of grid points in the direction normal to the hull from 81 to 121, so as to maintain about the same grid density near the external boundary of the domain. So the grid has $137 \times 121 \times 37$ nodes.

Once more, we start the presentation of results with the limiting streamline pattern, the fingerprint of the flow. The computed result, shown in Figure 8.23, reveals remarkable differences with the lower $Re$ results of the previous section. In general the lines are smoother at high $Re$, indicating that flow deflection by lateral pressure gradients is less pronounced. Although the limiting streamlines converge at some places, a clear confluence does not appear until very close to the stern. There is still a region of reversed flow, but it is smaller than at $Re = 5 \times 10^6$.

A comparison of the velocity fields at $x/L = 0.976$ for $Re = 2 \times 10^9$ and $Re = 5 \times 10^6$, respectively, is made in Figure 8.24. Additional data for the high Reynolds number are shown in Appendix C. As expected, the boundary layer is thinner and the wake narrower at the full-scale Reynolds number. But the results show also a change of the structure of the wake field, corresponding with the change in the limiting streamline pattern.

In scaling model test results to the full-scale condition, it is common practice to divide the measured resistance into a viscous part and a residual (mainly wave-
making) part. The viscous resistance is estimated as

\[ C_V = (1 + k)C_F, \]

where \( k \) is the so-called form factor and \( C_F \) the frictional resistance determined from

\[ C_F = \frac{0.075}{(10 \log Re - 2)^2}, \]

a correlation established by the International Towing Tank Conference (ITTC). The ratio of the viscous resistances at model scale and at full scale becomes therefore for the HSVA tanker

\[ \frac{C_{V_{fs}}}{C_{V_{ms}}} = \left( \frac{10 \log Re_{ms} - 2}{10 \log Re_{fs} - 2} \right)^2 = 0.414. \]

From our calculations we obtained for this ratio the value 0.43, where we used an integration over the outlet plane to determine the viscous resistance (as in eq. (7.2)).

### 8.2.3 HSVA tanker with propeller

It has been mentioned in the introductory Chapter 1 that the prediction of the flow near the location where the propeller is going to be fitted to a ship is an important goal of the application of a numerical method as developed in this thesis. A propeller design depends very much on the specification of the inflow field. A simulation of the flow around the hull without propeller can already give relevant information in this respect. Also in experimental work in towing tanks, the
measurement of this so-called "nominal wake" serves as a good starting point for design and analysis of a propeller. But this practice assumes that the interaction between hull and propeller is weak, i.e. the flow with operating propeller is close to a sum of the flow around the hull alone and the flow around the propeller alone. Especially for single-screw ships with heavily loaded propellers, such as tankers and bulkcarriers, this assumption is violated. The study of the effect of propeller-hull interaction on the velocity field is then useful and instructive. The simplification of the propeller to an actuator disk is permitted in such studies, if the interest is in the mean flow field and not in, say, vibratory forces.

We have repeated the calculation of the flow around the HSVA tanker at \( Re = 5 \times 10^6 \), now with a propeller acting at the stern. The propeller plane was chosen to be located at \( x/L = 0.976 \). As in Chapter 7 for the Wigley hull, we have represented the propeller action by an external force field, applied in the grid cells covered by the propeller disk. Again, only thrust forces were applied, radially distributed as
described in Chapter 7. The propeller diameter was chosen as 7.60m with the propeller axis at \( z/L = -0.0394 \), while the total thrust was set at \( C_T = 1.35 \), which is near the self-propulsion condition.

In global terms, the propeller accelerates the flow and generates a jet, embedded in the wake of the hull. The pressure is lowered ahead of the propeller, then shows a jump across the propeller disk and returns to undisturbed level in the far wake. The bilge vortex, generated on the hull, will be stretched which results in the vorticity being concentrated.

![Diagram](image)

**Figure 8.25: Effect of propeller action on limiting streamlines**

Let us first consider the limiting streamline pattern on the aftbody and see how it is changed by the propeller action. The comparison of the cases with and without propeller is shown in Fig. 8.25. It turns out that the propeller significantly reduces the reversed-flow zone. What appeared on the hull without propeller as an area bordered by an S-shaped confluence line, has changed with active propeller into a much smaller region.

The propeller effect on the mean flow is locally quite drastic. As an illustration, Fig. 8.26 shows both the axial and the transverse velocity field just behind the propeller, with and without the propeller operating. Outside the propeller jet the flow is slightly retarded, in particular in the region above the propeller jet, but
within the jet a speed above $1.3u_\infty$ is found. The longitudinal vortex is clearly seen to be intensified and its core position has shifted.

From the results with and without propeller we can evaluate the thrust deduction fraction $t$, which is the increase of the hull resistance due to the propeller as a fraction of the resistance without propeller. Quite realistically, the increase of the frictional resistance is found to be only marginal; the change in the pressure resistance makes
up 99 per cent of the thrust deduction fraction, which amounts to $t = 0.18$. This is a reasonable value for the present ship.
Chapter 9

Closure

The design of ships is a process that in many respects differs from, say, the design of an aircraft or a motor-car. A salient particularity is that there is comparatively little series production in shipbuilding. Yet, every ship must come up to the requirements, negotiated between ship-owner and shipyard. To reduce the risks involved, the shipyard does early checks on the design, before the actual construction of the ship starts. With regard to the hydrodynamic aspects of the design, some kind of investigation of the performance of the ship under representative operating conditions is needed. As a routine, such an investigation is based on a series of model tests, carried out in a water basin. But with the advent of computational analysis capability, new tools become available which allow more detailed analyses in an earlier design stage or allow information to be gathered to supplement, or in some cases even replace, model test results.

We have described such a tool, viz. a numerical method for the simulation of the flow around the aft end of a ship. In more general terms, the method is suitable for external flows around fairly slender bodies at high Reynolds number (Re up to as much as $10^8$). The method belongs to the category of cell-centered finite-volume methods, solving the conservative form of the Reynolds-averaged Navier-Stokes equations in general coordinates. The main distinguishing feature is the space-iterative solution strategy, which was called an incomplete symmetric Gauss-Seidel process with approximate multigrid acceleration. The choice for this strategy has been inspired by the physical features of the kind of flows we try to simulate. And the strategy has been elaborated into a working code in the belief that a better result would be obtained than by adopting a general-purpose flow solver. We have explained the background, the foundations and the numerical details of the resulting method and shown its viability by a demonstration of the performance of the method for representative cases. We believe that in the limited range of applications for which it has been designed it can compete successfully with other methods.

Needless to say that this will not be the end of the development of the method. There are several areas where further enhancements will be realised. To name a few: the introduction of general multi-block grids to get more flexibility and accuracy in
the gridding of the computation domain; the inclusion of the ship-generated free surface disturbance in the simulations; the use of more advanced iterative solution algorithms; the application of more complicated turbulence models and, later maybe, large-eddy simulation; the extension of the range of applicability (manoeuvring ship, ship in wavy sea). Some work in these areas has already been done, elsewhere and at MARIN. But for us the leading principle in these developments will remain that they must produce computational analysis tools that fit in the typical ship design process, must produce results that can fruitfully be used and relied upon and must be able to compete with model testing both in outcome and price. We think that the method we have described here satisfies these criteria.
Appendix A

Implementations of flux-limiting convection scheme

The flux-limiting $\kappa$-scheme ($0 \leq \kappa \leq 1$), introduced in Chapter 5, is used in two different implementations: a general implementation, used in the $\xi^2$ and $\xi^3$ directions, and a special implementation for the $\xi^1$ (mainstream) direction. They are described below in the one-dimensional grid notation of Chapter 5. Herein is $\phi$ the convected scalar and $u_n$ the convecting speed on the cell face at $i + \frac{1}{2}$.

The convected quantity $\phi$ on a cell face is evaluated according to equation (5.6):

$$
\dot{\phi}_f = \phi_c + \lambda \left[ (1 + \kappa) - 2\kappa \phi_c \right],
$$

with

$$
\lambda = \lambda(\phi_c) = \max \left[ 0, \phi_c(1 - \phi_c) \right].
$$

Conversion to non-normalised variables and suppression of the argument of \( \lambda \) yields

$$
\phi_f = \phi_c - 2\kappa \lambda (\phi_c - \phi_v) + \lambda(1 + \kappa)(\phi_d - \phi_v).
$$

This relation is reformulated in two ways:

$$
\phi_j = \phi_c (1 - 2\kappa \lambda) + \phi_d \lambda(\kappa + 1) + \phi_v \lambda(\kappa - 1)
$$

(A.1)

and, with hidden $\phi_d$,

$$
\phi_f = \phi_c + (\phi_c - \phi_v) \left[ -2\kappa \lambda + (1 + \kappa) \frac{\lambda}{\phi_c} \right].
$$

(A.2)

If $\phi_f = \phi_{i+\frac{1}{2}}$ we have:

$$
\begin{cases}
    \text{if } u_n \geq 0 : & \phi_c = \phi_i \quad \phi_v = \phi_{i-1} \quad \phi_d = \phi_{i+1} \\
    \text{if } u_n < 0 : & \phi_c = \phi_{i+1} \quad \phi_v = \phi_{i+2} \quad \phi_d = \phi_i
\end{cases}
$$

(A.3)
(i) General implementation

Referring to equation (A.1) and the relations (A.3), the general procedure is to compute subsequently:

\[ \theta = \frac{1}{2} \left[ 1 + \text{sign}(u_n) \right] \]

\[ d = \phi_D - \phi_U = \theta(\phi_{i+1} - \phi_{i-1}) + (1 - \theta)(\phi_i - \phi_{i+2}) \]

\[ \tilde{\phi}_C = \frac{\phi_C - \phi_U}{\phi_D - \phi_U} = \frac{\theta(\phi_i - \phi_{i-1}) + (1 - \theta)(\phi_{i+1} - \phi_{i+2})}{\text{sign}(d) \max \|d\|_1 10^{-5}} \]

\[ \lambda = \max \left[ 0, \tilde{\phi}_C (1 - \tilde{\phi}_C) \right] \]

\[ \phi_{i+\frac{1}{2}} = \phi_{i-1} \left[ \lambda \theta (\kappa - 1) \right] + \phi_i \left[ \theta + \lambda \left\{ (1 - \theta)(\kappa + 1) - \theta(2\kappa) \right\} \right] + \phi_{i+1} \left[ (1 - \theta) + \lambda \left\{ \theta(\kappa + 1) - (1 - \theta)(2\kappa) \right\} \right] + \phi_{i+2} \left[ \lambda(1 - \theta)(\kappa - 1) \right] \]

Notes:

1. For \( \kappa = \frac{1}{2} \) we obtain the flux-limiting QUICK scheme:

\[ \phi_{i+\frac{1}{2}} = \phi_{i-1} \left[ -\frac{1}{2} \lambda \theta \right] + \phi_i \left[ \theta + \frac{1}{2} \lambda \left\{ 3(1 - \theta) - 2\theta \right\} \right] + \phi_{i+1} \left[ (1 - \theta) + \frac{1}{2} \lambda \left\{ 3\theta - 2(1 - \theta) \right\} \right] + \phi_{i+2} \left[ -\frac{1}{2} \lambda(1 - \theta) \right] \]

2. With \( \kappa = \frac{1}{2} \) and \( \lambda = \frac{1}{4} \) the standard QUICK scheme is recovered:

\[ \phi_{i+\frac{1}{2}} = \phi_{i-1} \left[ -\frac{1}{8} \lambda \theta \right] + \phi_i \left[ \frac{3}{8}(\theta + 1) \right] + \phi_{i+1} \left[ \frac{3}{8}(2 - \theta) \right] + \phi_{i+2} \left[ -\frac{1}{8}(1 - \theta) \right] \]

3. Because \( \lambda = \frac{1}{4} \) in the non-limiting \( \kappa \)-scheme and \( \tilde{\phi}_C (1 - \tilde{\phi}_C) \leq \frac{1}{4} \), we can write

\[ \lambda = \max \left[ \frac{1 - \sigma}{4}, \tilde{\phi}_C (1 - \tilde{\phi}_C) \right] \]

and use \( \sigma \) as a switch: for \( \sigma = 1 \) the limiter is activated, for \( \sigma = 0 \) it is switched off.
(ii) Special implementation

In the mainstream ($\xi^1$) direction flux-limiting is applied, if at all, for $u_n \geq 0$ only. The limiter $\lambda$ is rewritten as

$$\lambda = \beta \tilde{\phi}_c (1 - \tilde{\phi}_c),$$

with

$$\beta = \frac{1}{2} \left[ 1 + \text{sign}(\tilde{\phi}_c (1 - \tilde{\phi}_c)) \right],$$

and substituted in (A.2) to yield:

$$\phi_j = \phi_c + (\phi_c - \phi_v) \beta (1 - \tilde{\phi}_c)(1 + \kappa - 2\kappa \tilde{\phi}_c).$$

The practical implementation proceeds then in the following steps:

$$d = \phi_d - \phi_v = \phi_{i+1} - \phi_{i-1}$$

$$\tilde{\phi}_c = \frac{\phi_c - \phi_d}{\phi_c - \phi_v} = \frac{\phi_i - \phi_{i-1}}{\text{sign}(d) \max(|d|, 10^{-5})}$$

$$\beta = \frac{1}{2} \left[ 1 + \text{sign}(\tilde{\phi}_c (1 - \tilde{\phi}_c)) \right]$$

$$\Lambda = \beta (1 - \tilde{\phi}_c)(1 + \kappa - 2\kappa \tilde{\phi}_c)$$

$$\phi_{i+\frac{1}{2}} = (1 + \Lambda) \phi_i - \Lambda \phi_{i-1}$$
Appendix B

Stability analysis of predictor-corrector scheme

Most properties of the predictor-corrector scheme as found in the numerical verifications of Chapter 7 are essentially revealed by a von Neumann stability analysis of the 2D linearised equations on a Cartesian grid, as given below.

As in (2.3), the equations can be written without external forces as:

$$
\begin{pmatrix}
\mathcal{L} & 0 & \partial / \partial x \\
0 & \mathcal{L} & \partial / \partial y \\
\partial / \partial x & \partial / \partial y & 0
\end{pmatrix}
\begin{pmatrix}
u \\
0
\end{pmatrix}
= \begin{pmatrix}
0 \\
0 \\
0
\end{pmatrix}
$$

in which $\mathcal{L}$ is the linear differential operator

$$\mathcal{L} = \rho u_j \partial / \partial x_j - \mu \partial^2 / \partial x_j^2.$$

After substitution of

$$\begin{pmatrix}
u \\
0
\end{pmatrix}^n = \begin{pmatrix}
A_u \\
A_v \\
A_p
\end{pmatrix}^n \exp(ik_1h_x) \exp(ik_2h_y),$$

the predictor step can be written as

$$\begin{pmatrix}
G_{11} & 0 & G_{13} \\
0 & G_{22} & G_{23} \\
G_{31} & G_{32} & 0
\end{pmatrix}
\begin{pmatrix}
A_u \\
A_v \\
A_p
\end{pmatrix}^n = \begin{pmatrix}
0 & 0 & H_{13} \\
0 & 0 & 0 \\
0 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
A_u \\
A_v \\
A_p
\end{pmatrix}^{n-1}
$$

or

$$GA^{n*} = HA^{n-1}$$

so that

$$A^{n*} = G^{-1}HA^{n-1}.$$
Since
\[
G^{-1} = \frac{1}{G_{11}G_{32}G_{22} + G_{13}G_{31}G_{22}} \begin{pmatrix}
G_{23}G_{32} & -G_{32}G_{13} & G_{22}G_{13} \\
-G_{31}G_{23} & G_{13}G_{31} & G_{11}G_{23} \\
G_{31}G_{22} & G_{11}G_{32} & -G_{11}G_{22}
\end{pmatrix}
\]
we find that
\[
\frac{A^*_p}{A_p^{-1}} = \frac{H_{13}G_{31}G_{22}}{G_{11}G_{23}G_{32} + G_{13}G_{31}G_{22}}.
\]
Upon substituting the equality \( G_{11} = G_{22} \), this expression for the amplification ratio of the pressure can be simplified to
\[
\frac{A^*_p}{A_p^{-1}} = \frac{H_{13}G_{31}}{G_{23}G_{32} + G_{13}G_{31}}.
\] (B.1)

The denominator of (B.1) comes from pressure gradients and continuity equation, a term which has been considered in detail in Chapter 5.

If we introduce the notation \( \theta_1 = k_1h_x \) and \( \theta_2 = k_2h_y \), we have for a typical discretisation that we use:
\[
\begin{align*}
H_{13} &= -e^{i\theta_1}/h_x \\
G_{13} &= -1/h_x \\
G_{31} &= \frac{1}{2}(3 - 4e^{-i\theta_1} + e^{-2i\theta_1})/h_x \\
G_{23} &= \frac{1}{8}(-3e^{i\theta_2} - 3 + 7e^{i\theta_2} - e^{2i\theta_2})/h_y \\
G_{32} &= \frac{1}{8}(3e^{i\theta_2} + 3 - 7e^{-i\theta_2} + e^{-2i\theta_2})/h_y
\end{align*}
\]
which yields
\[
\frac{A^*_p}{A_p^{-1}} = \frac{e^{i\theta_1}(3 - 4e^{-i\theta_1} + e^{-2i\theta_1})}{(3 - 4e^{-i\theta_1} + e^{-2i\theta_1}) - 2d^2(\theta_2)},
\]
with
\[
d^2(\theta_2) = h_x^2G_{23}G_{32} = \left( \frac{h_x}{h_y} \right)^2 \left( \cos \theta_2 - 1 \right) \left[ 2 - \frac{3}{8}(\cos \theta_2 - 1)^2 \right].
\]

Upon evaluation we obtain
\[
\frac{A^*_p}{A_p^{-1}} = \frac{2(\cos \theta_1 - 1) + \iota \sin \theta_1}{(\cos \theta_1 - 1)^2 - d^2(\theta_2) + \iota \sin \theta_1(2 - \cos \theta_1)}.
\] (B.2)

The corrector step, equation (6.6), operating on the pressure only, yields
\[
A^n_p = A_p^{n-1} + \omega_c \left( 1 - \omega_b + \frac{\omega_b}{\omega_f} \right) (A^*_p - A_p^{n-1}) + \omega_be^{i\theta_1} \left( A^n_p - A_p^{n-1} \right).
\]
or
\[
\frac{A_p^n}{A_p^{n-1}} = 1 + \frac{\omega_c \left( 1 - \omega_b + \frac{\omega_b}{\omega_f} \right) \left( \frac{A_p^n}{A_p^{n-1}} - 1 \right)}{1 - \omega_b e^{\theta_1}}.
\]
Substitution of (B.1) gives
\[
\frac{A_p^n}{A_p^{n-1}} = 1 + \frac{\omega_c \left( 1 - \omega_b + \frac{\omega_b}{\omega_f} \right) \left( \frac{H_{13} G_{31}}{G_{23} G_{32} + G_{13} G_{31}} - 1 \right)}{1 - \omega_b e^{\theta_1}}. \tag{B.3}
\]
If (B.2) is used and the relaxation parameters are set as \(\omega_f = 1, \omega_b = 0.5\) and \(\omega_c = 0.5\), the result for the modulus of the amplification factor in B.3 is shown as a function of \(\theta_1\) and \(\theta_2\) in Figure B.1 and compared with the result of the predictor step only, i.e. (B.2).

From the figure a couple of conclusions can be drawn. First of all the improvement obtained by the corrector step is clearly revealed: the amplification factor is substantially reduced. Secondly, we observe that \(|A_p^n/A_p^{n-1}|\) is closest to unity for small values of \(\theta_1\) and \(\theta_2\). For a small but fixed value of \(\theta_2\) the amplification factor for the pressure in the predictor-corrector scheme shows peaks for rather small values of \(|\theta_1|\). This may explain why after a couple of iterations a pressure wave appears with a distinct length in \(x\)-direction, determined by the peak value of \(|\theta_1|\), as displayed for example in Fig. 7.4. Furthermore, we observe that these peaks shift to lower \(|\theta_1|\), leading to pressure waves with greater length in \(x\)-direction, with decreasing \(\theta_2\). Thirdly: for small \(\theta_2\), \(d^2(\theta_2)\) can be approximated as
\[
d^2(\theta_2) \approx -\left( \frac{h_x}{h_y} \right)^2 \theta_2^2 = -(h_x k_2)^2.
\]
Because of the Dirichlet boundary condition for the pressure on the external boundary of the calculation domain, the lowest possible wave number for the \(y\)-direction is
\[
k_{2,\text{min}} = \frac{\pi}{2y_{\text{max}}},
\]
if \(y_{\text{max}}\) is the width of the domain. This brings out the importance of the ratio
\[
\frac{h_x}{y_{\text{max}}}
\]
for the minimum value of \(|d^2(\theta_2)|\) and thus for the maximum value of the amplification factor. It follows that with greater \(h_x\) or smaller \(y_{\text{max}}\), the convergence rate will improve; \(h_y\) has no substantial influence. As B.3 shows, the amplification factor is influenced by the relaxation parameters, in particular \(\omega_b\). It can be verified that with decreasing \(h_x/y_{\text{max}}\), thus a tendency to slower convergence, the value of \(\omega_b\) should also be decreased to get the best convergence rate.
Figure B.1: Amplification ratio $|A_p^n/A_p^{n-1}|$ (above) and $|A_p^{n*}/A_p^{n-1}|$ (below)
Appendix C

Collected results for HSVA tanker

This appendix gives a graphical overview of the results obtained for the HSVA tanker. Corresponding experimental data are also shown, if available. The data given are, subsequently:

HSVA tanker at $Re = 5 \times 10^6$:

- Contour plots of the distribution of the longitudinal velocity component $u$;
- Vector plots of the transverse velocity components $v$ and $w$;
- Contour plots of the pressure distributions;
- Contour plots of the eddy viscosity;
- Contour plots of the longitudinal vorticity component;
- Influence of vortex correction on wake field at $x/L = 0.976$.

HSVA tanker at $Re = 2 \times 10^9$:

- Contour plots of the distribution of the longitudinal velocity component $u$;
- Vector plots of the transverse velocity components $v$ and $w$;
- Contour plots of the eddy viscosity;
- Contour plots of the longitudinal vorticity component.
Figure C.1: Axial velocity field at $x/L=0.940$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.2: Axial velocity field at $x/L=0.941$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.3: Axial velocity field at $x/L=0.947$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.4: Axial velocity field at $x/L=0.953$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 

Numerical simulation of ship stern flows
Figure C.5: Axial velocity field at $x/L = 0.960$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.6: Axial velocity field at $x/L=0.967$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.7: Axial velocity field at $x/L = 0.976$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.8: Axial velocity field at $x/L = 0.986$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.9: Axial velocity field at $x/L=1.005$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.10: Axial velocity field at $x/L=1.042$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.11: Transverse velocity field at $x/L=0.940$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.12: Transverse velocity field at $x/L=0.941$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.13: Transverse velocity field at $x/L = 0.947$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.14: Transverse velocity field at $x/L=0.953$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^5$. 
Figure C.15: Transverse velocity field at $x/L=0.960$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.16: Transverse velocity field at $x/L=0.967$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.17: Transverse velocity field at $x/L = 0.976$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.18: Transverse velocity field at $x/L=0.986$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^5$. 
Figure C.19: Transverse velocity field at $x/L = 1.005$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.20: Transverse velocity field at $x/L=1.042$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.21: Pressure field at x/L=0.940. Comparison of experimental (top) and computed (bottom) results at Re = 5 \times 10^6.
Figure C.22: Pressure field at $x/L=0.941$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.23: Pressure field at $x/L=0.947$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.24: Pressure field at $x/L = 0.953$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.25: Pressure field at $x/L=0.960$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.26: Pressure field at $x/L=0.967$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.27: Pressure field at $x/L = 0.976$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.28: Pressure field at x/L=0.986. Comparison of experimental (top) and computed (bottom) results at Re = 5 \times 10^6.
Figure C.29: Pressure field at $x/L=1.005$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.30: Pressure field at $x/L=1.042$. Comparison of experimental (top) and computed (bottom) results at $Re = 5 \times 10^6$. 
Figure C.31: Computed eddy viscosity (as a multiple of the molecular viscosity) at several stations for $Re = 5 \times 10^6$. Contour level step size is $\Delta \nu_t/\nu = 25$. 
Figure C.32: Computed axial vorticity at several stations for $Re = 5 \times 10^5$. Contour level step size is $\Delta \omega_y L / u_\infty = 20$. 

Numerical simulation of ship stern flows
HSVA tanker; $x/L = 0.976$

Figure C.33: Detail of effect of "vortex correction" on velocity field in propeller plane at $Re = 5 \times 10^6$. 
Figure C.34: Axial velocity field at $Re = 2 \times 10^3$ at several stations. Contour level step size is $\Delta u/U = 0.1$.
Figure C.35: Transverse velocity field at $Re = 2 \times 10^9$ at several stations.
Figure C.36: Eddy viscosity (as a multiple of the molecular viscosity) at several stations for $Re = 2 \times 10^9$. Contour level step size is $\Delta \nu_t/\nu = 5000$. 
Figure C.37: Axial vorticity at several stations for $Re = 2 \times 10^6$. Contour level step size is $\Delta \omega_z L/u_\infty = 20$. 
Bibliography


Numerical simulation of ship stern flows


List of symbols

\( \bar{a}_i \) - covariant base vectors of \( \xi, \eta, \zeta \) system
\( \bar{a}^i \) - contravariant base vectors of \( \xi, \eta, \zeta \) system
\( \bar{b}_i \) - scaled covariant base vector, \( \bar{b}_i = \bar{a}_i/\sqrt{\bar{g}} \)
\( \bar{b}^i \) - scaled contravariant base vector, \( \bar{b}^i = \sqrt{\bar{g}} \bar{a}^i \)
\( B \) - ship’s breadth
\( C_P \) - pressure coefficient, \( C_P = p/\frac{1}{2} \rho U_{\infty}^2 \)
\( C_f \) - friction coefficient, \( C_f = \tau_w/\frac{1}{2} \rho U_{\infty}^2 \)
\( C_T \) - propeller thrust coefficient, \( C_T = f_1/\frac{1}{2} \rho U_{\infty}^2 \pi R_p^2 \)
\( e_i \) - unit base vectors of \( x, y, z \) system
\( f^1, f^2, f^3 \) - Cartesian components of external force
\( F_1, F_2, F_3 \) - covariant components of external force
\( F^1, F^2, F^3 \) - contravariant components of external force
\( \sqrt{\bar{g}} \) - Jacobian of coordinate transformation
\( g_{ij} \) - covariant metric tensor
\( g^{ij} \) - contravariant metric tensor
\( H \) - total head
\( k \) - turbulence kinetic energy
\( L \) - ship’s length
\( p \) - pressure
\( Re \) - Reynolds number, \( Re = |\bar{U}_{\infty}|L/\nu \)
\( R_p \) - propeller radius
\( S \) - coordinate-invariant measure of strain rate
\( S_{ij} \) - covariant strain rate tensor
\( S^{ij} \) - contravariant strain rate tensor
\( T^{ij} \) - general stress tensor
\( \bar{U} \) - velocity vector
\( u, v, w \) - Cartesian velocity components
\( u^1, u^2, u^3 \) - Cartesian velocity components
\( U_1, U_2, U_3 \) - covariant velocity components, \( U_i = \bar{a}_i \bar{U} \)
\( U^1, U^2, U^3 \) - contravariant velocity components, \( U^i = \bar{a}^i \bar{U} \)
\( U(1), U(2), U(3) \) - grid-aligned physical velocity components
\( V_1, V_2, V_3 \) - scaled covariant velocity components, \( V_i = \sqrt{\bar{g}} U_i \)
\( V^1, V^2, V^3 \) - scaled contravariant velocity components, \( V^i = \sqrt{\bar{g}} U^i \)
Numerical simulation of ship stern flows

- Cartesian coordinates

- Cartesian coordinates

- distance to solid wall, nondimensionalised with \( \nu / \sqrt{\tau_w / \rho} \)

- curvilinear coordinates

- curvilinear coordinates

- boundary layer thickness

- boundary layer displacement thickness

- flux limiter

- dynamic viscosity of fluid, \( \mu = \rho \nu \)

- kinematic viscosity of fluid

- effective viscosity of fluid, \( \nu_e = \nu + \nu_t \)

- eddy viscosity

- eddy viscosity without wall damping

- mass density of fluid

- viscous stress tensor

- wall shear stress

- general notation for a dependent variable

- vorticity vector
Summary

This thesis is about a numerical method for the simulation of the flow around a ship, moving steadily on a straight course in still water, the focus being on the prediction of the water motion near the ship's aft end. The purpose is a simulation model that can fruitfully be used in the design process (as far as concerned with the hydrodynamic optimization of the hull shape and its propeller), and in the interpretation of model test results.

Even if a ship operates in still water, and not in a wind-disturbed sea, the complexity of the physical problem to be modelled is considerable. This is in the first place caused by the occurrence of turbulence in a part of the flow field. Although there is an adequate mathematical model describing the fluid motion including the turbulence, it is necessary for practical reasons to work with a time-averaged form of this model. The pertinent equations of motion are referred to as the Reynolds-averaged Navier-Stokes (RANS) equations. They require to be supplemented with a turbulence model, because the time-averaging has introduced new unknowns, the Reynolds stresses.

A second complication is that the ship moves at the interface of two media, water and air, with widely different density, so that a wave pattern is created. In the method described in this thesis, the wave formation is neglected by treating the undisturbed free surface as a symmetry plane. So we consider effectively the flow around a body deeply submerged in water, where the shape of the body is determined by the composition of the underwater part of the ship and its mirror image (double-body flow).

The flow field to be simulated can thus be classified as an external flow around a more or less streamlined body in a fluid of small viscosity. For normal operating conditions of a ship, the characteristic Reynolds number, based on the length and the speed of the ship, is in the range of, say, $10^6 < \text{Re} < 10^9$, if the ship is to be considered both at model scale and in its true size. The flow is steady in a ship-fixed reference frame, while the fluid is assumed to be incompressible.

Because we are concerned with an external flow, a division of the fluid domain into zones is for reasons of computational efficiency a natural thing to do. Viscous effects are noticeable only in a thin layer around the hull and in the wake. Further away from the hull, the fluid behaves as being essentially inviscid, which allows a considerable reduction of the complexity of the mathematical model. On the forward part of the ship the viscous layer is so thin that the boundary-layer equations are an adequate description. The solution of the RANS equations can therefore be restricted to a relatively small part of the fluid domain, enclosing the aft half of the ship and a part of its wake. For the required turbulence model we use the concept of the isotropic eddy viscosity. Two versions are used in this thesis, one based on an algebraic formulation, and the other on a transport equation for the eddy viscosity. The action of the propeller, if included, is modelled by a specified body force distribution (actuator disk).
Analytical solution being beyond reach, a discrete analogue of the RANS equations is established. The discretisation is based on the finite-volume technique. This requires first of all that the selected domain is overlayed with a 3D grid. A boundary-fitted, structured, mono-block, H-O grid is used. Two methods are described by which such a grid can be constructed. The first connects a set of 2D grids, essentially generated by a conformal mapping technique based on the Schwarz-Christoffel transformation. In the second, the grid is found via the solution of a Poisson equation for the grid coordinates.

All flow variables are defined in the center of the grid cells (collocated, cell-centered variable arrangement). The equations are integrated for each grid cell and Gauss theorem is applied to yield an integral over the cell faces. These cell face integrals are then approximated, using interpolations of the cell-centered data.

The properties of the continuum equations should as far as possible be carried over to the discrete analogue. Thus, the maintenance of the elliptic character of the equation system in the discrete approximation is discussed in some detail. It leads the way to a proper choice for the discretisation of the continuity equation and the pressure gradient in the momentum equation. Furthermore, an account is given of how flux-limiting properties are added to the discretisation of the convection terms. The resulting discretisation is conservative for mass and momentum, formally second-order accurate, and uniform-flow preserving.

The most distinguishing feature of the method is the solution strategy. While the majority of computation methods for the flow around a ship employs either the pressure-correction or the artificial-compressibility method, our approach is a space-marching scheme, in which the coupling between the momentum and the continuity equations is maintained in the solution process. Three convergence-accelerating techniques are introduced: grid sequencing, a predictor-corrector method for the pressure and an approximate multigrid strategy. The solution of the system of algebraic equations is accomplished with a coupled incomplete LU decomposition or with GMRES; in the latter case the LU-factorisation is employed as a preconditioner.

Verification of the computational method is carried through in three test cases: the laminar wake of a flat plate, the flow around the aft end of a modified prolate spheroid and the same for the Wigley hull. Convergence of the solution on grids of different density is examined and the effect of the Reynolds number on the convergence rate of the solution process is demonstrated to be practically nil.

Finally, results of application are shown and discussed, and validated against experimental data or other sources. Both the laminar and the turbulent flow along a flat plate are considered; in addition to the modified spheroid, the true spheroid with flow separation at its tail is dealt with. The 3D applications are completely focused on the HSVA tanker. Results are given for the model test condition \((Re = 5 \times 10^6)\) and the full scale case \((Re = 2 \times 10^9)\), while in a third case the effects of the propeller action are included.

The code is currently in use at MARIN as a tool for quality assessment of hull designs on request of ship yards, navies and other customers.
Samenvatting

Dit proefschrift gaat over een numerieke methode voor de simulatie van de omstrooming van een schip in stationaire, rechtlijnige vaart in stil water, waarbij de nadruk ligt op de voorspelling van de waterbeweging nabij de achtersteven. Het doel is een simulatiemodel dat zinvol kan worden gebruikt in een ontwerpproces (voor zover het gaat over de hydrodynamische optimalisatie van de rompver en de schroef), alsmede bij de interpretatie van modelproefresultaten.

Zelfs als een schip zich voortbeweegt in stil water en niet in een door de wind in beroering gebrachte zee, is de complexiteit van het te modelleren fysische probleem aanzienlijk. In de eerste plaats wordt dat veroorzaakt door het optreden van turbulentie in een deel van het stromingsveld. Hoewel er een goed mathematisch model voor de beschrijving van de vloeistofbeweging inclusief de turbulentie, is het om praktische redenen noodzakelijk de tijdsgemiddelde vorm ervan te gebruiken. De betreffende vergelijkingen worden aangeduid als de Reynolds-gemiddelde Navier-Stokes (RANS) vergelijkingen. Zij vereisen de aanvulling met een turbulentiemodel, omdat het middelproces de Reynoldsspanningen als nieuwe onbekenden heeft geïntroduceerd.

Een tweede complicatie is dat het schip beweegt aan het scheidingsvlak van twee media, water en lucht, met sterk verschillende dichtheden, zodat een golfpatroon wordt opgewekt. In de methode beschreven in dit proefschrift wordt de golfvorming verwaarloosd door het ongestoorde vrij oppervlak te behandelen als een symmetrievlak. Dit betekent dat we feitelijk de stroming om een diep in water ondergedompeld lichaam beschouwen, waarbij de vorm van het lichaam bepaald wordt door de samenvoeging van het onderwaterschip en het spiegelbeeld daarvan (dubbel-modelstroming).

Het te simuleren stromingsveld kan aldus worden geclassificeerd als een uitwendige stroming om een min of meer gestroomlijnd lichaam in een vloeistof met geringe viscositeit. Voor normale bedrijfsomstandigheden van een schip valt het kenmerkende Reynoldsgetal, gebaseerd op de lengte en de snelheid van het schip, in het bereik van ongeveer \(10^6 < Re < 10^9\), als we het schip zowel op modellschaal als op ware grootte willen beschouwen. De stroming is stationair in een scheepsvast assenstelsel, terwijl de vloeistof wordt verondersteld onszamendrukbaar te zijn.

Aangezien het een externe stroming betreft, ligt een splitsing van het vloeistofdomein in delen om redenen van reken-efficiëntie voor de hand. Viskeuze effecten zijn alleen van belang in een dunne laag om de scheepsromp en in het volgstromveld. Op wat grotere afstand van het schip gedraagt de vloeistof zich als niet-viskeus, hetgeen een drastische vereenvoudiging van de mathematische beschrijving toelaat. Over de boeg van het schip is de viskeuze laag dermate dun dat die met grenslaagtheorie adequaat beschreven wordt. De oplossing van de RANS vergelijkingen kan derhalve worden beperkt tot een relatief klein deel van het vloeistofdomein, dat de achtersteven en een deel van het volgstromveld omvat. Voor het vereiste turbulentiemodel hanteren we het concept van een isotrope effectieve viscositeit. Er worden
in het proefschrift twee versies gebruikt, waarvan de één werkt met een algebraïsch voorschrift, de ander met een transportvergelijking voor de effectieve viscositeit. De werking van de schroef, indien van toepassing, wordt gemodelleerd door middel van een extern krachteneveld (actuatorschijf).

Omdat een analytische oplossing buiten bereik ligt, wordt een discreet analogon van de RANS vergelijkingen opgesteld. De discretisatie is gegrondvest op de eindige-volume techniek. Dit vereist allereerst dat een 3D rooster over het gekozen domein wordt gelegd. Daarvoor wordt een gestructureerd, enkelbloks, H-O rooster gebruikt dat zich voegt naar de vorm van de domeinranden. Er worden twee methoden beschreven om zo'n rooster te maken. De eerste bestaat uit de stapeling van een reeks 2D roosters, elk in essentie tegenover elkaar met een vergelijkende afbeelding die gebruik maakt van de Schwarz-Christoffel transformatie. Bij de tweede wordt het rooster bepaald via de oplossing van een Poisson-vergelijking voor de roostercoördinaten.

Alle stromingsvariabelen worden gedefinieerd in celmiddens. De vergelijkingen worden geïntegreerd voor iedere roostercel en vervolgens wordt het theorema van Gauss toegepast om een integraal over de celwanden te verkrijgen. Deze oppervlak-integrallen worden benaderd door gebruik te maken van interpolatie op de celgecentreren gegevens.

De eigenschappen van de continuum vergelijkingen moeten zo volledig mogelijk worden overgedragen op het discrete analogon. Zo wordt het behoud van het elliptische karakter in de discrete benadering uitvoerig besproken. Het leidt tot een passende keus voor de discretisatie van de continuïteitsvergelijking en de drukgradient in de impulsvergelijking. Voorts wordt beschreven hoe flux-beperking eigen- en eigenschappen worden toegevoegd in de discretisatie van de convectieve termen. De uiteindelijke discretisatie is behoudend voor massa en impuls, is exact voor uniforme stroming en formee tweede orde nauwkeurig.

Waarin de methode zich het meest onderscheidt van andere is de oplos-strategie. Terwijl de meeste berekeningsmethoden voor omstoming van een schip de aanpak volgen van hetzij druk-correctie, hetzij kunstmatige compressibiliteit, maakt de hier voorgestelde methode gebruik van een ruimtelijk iteratieproces, waarin de koppeling tussen impuls- en continuïteitsvergelijking gehandhaafd blijft. Drie convergentie-versnellende methoden worden geïntroduceerd: stapsgewijze roosterverfijning, een predictor-corrector schema voor de druk en een ruwe benadering van de multigrid methode. De oplossing van het stelsel algebraïsche vergelijkingen geschiedt met een gekoppelde incomplete LU onttbinding of met GMRES, waarbij de LU-factorisering als preconditioner fungeert.

Numerische verificatie van de methode vindt plaats aan de hand van drie testgevallen: de laminaire stroming langs een vlakke plaat van eindige lengte, de stroming om het achtereind van een gemonificeerde sferoïde, en de omstoming van de "Wigley hull". Convergentie van de oplossing op roosters van verschillende dichtheid wordt bekeken. Bovendien wordt vastgesteld dat het effect van het Reynoldsgetal op de convergentiesnelheid van de iteratieve oplossing practisch nihil is.
Tenslotte worden resultaten van toepassingen getoond en becommentarieerd. Tegelijk worden de gegevens waar mogelijk gevalideerd met experimentele gegevens of andere informatie. Zowel de laminaire als de turbulente stroming langs een vlakke-plaat worden behandeld. De gemodificeerde spheœïde komt opnieuw aan de orde, maar daarnaast ook de echte spheœïde, die grenslaagloslating vertoont aan het staarteinde. De toepassing op drie-dimensionale stromingen wordt toegesplitst op de HSVA tanker. Resultaten worden gepresenteerd voor de modelproef omstandigheden \((Re = 5 \times 10^8)\) en voor de ware-grootte condities \((Re = 2 \times 10^9)\), terwijl in een derde geval het effect van de schroefwerking wordt meegenomen.

De methode is thans in gebruik bij het MARIN als een gereedschap voor de toetsing van de kwaliteit van scheepsontwerpen, afkomstig van werven, marines en andere opdrachtgevers.
Acknowledgement

It was a pleasure to work with Piet Wesseling as my supervisor. I have had great benefit of his detailed and relevant comments on earlier versions of this thesis.

The development of PARNASSOS, the subject of this thesis, has not been a one-man's job; several colleagues have made their contribution. I like to thank a few explicitly here.

With Hoyte Raven I worked together on the foundations of the method. Although he later changed subject, he has followed the developments with great interest and was a most inspiring discussion partner all the time.

Luis Eça, from Portugal, has had a significant role in the development of both PARNASSOS and the grid generation software. He stayed in Wageningen for several years as a visiting scientist, and has cooperated with us ever since. The number of publications that we have meanwhile written together is indicative of the closeness of our cooperation. He went through my notes several times, to my advantage.

Thanks are also due to Auke van der Ploeg for updating me on recent developments in the solution of systems of linear equations, and to Jaap Windt for assistance in grid generation.

I am further indebted to my employer, the Maritime Research Institute Netherlands. My appreciation is not only related to the funding made available for my work, but also to the continuous support and stimulation from various quarters to put my thoughts in writing.

I realize that I have made high demands on the patience and consideration of Duifje and Age during the last few years, and owe them my sincere thanks. In addition, I thank Duifje for her encouragement and her linguistic (and other) advice.
Curriculum Vitae


The author was born on February 17, 1949, in Heerenveen, the Netherlands, but grew up in the nearby village Gorredijk. He attended secondary school at the "Drachtster Lyceum" in Drachten, upon which he started studying Naval Architecture at Delft University of Technology in 1968. He graduated at the end of 1973 with a master thesis on the scale effect on the wake of ships. This work was largely carried out at the Netherlands Ship Model Basin, now known as the Maritime Research Institute Netherlands (MARIN), where he entered employment immediately after taking his degree. He started as a project manager at the newly built Depressurized Towing Tank. Soon, however, he moved to the research department. Those were the years of the rise of Computational Fluid Dynamics and he got actively involved. A source-panel method for potential flows was made available and introduced in MARIN’s consultancy service. Furthermore, he developed a boundary-layer method with which he participated in several workshops. Recognizing the strong dependence of the practical application of CFD on a fast and flexible geometry handling of the ship hull form, he got also involved in computer-aided design. From 1980 on, he has been engaged in the numerical solution of the Reynolds-averaged Navier-Stokes equations. Excluding the period 1990-1994, in which he had a managerial position, he has worked on this subject until to-day. This thesis is a consolidation of that work.
Stellingen
behorende bij het proefschrift van Martin Hoekstra

1. Bij de oplossing van de Navier-Stokes vergelijkingen voor onsamendrukbare media is het niet bezwaarlijk om de continuïteitsvergelijking te behandelen als ware het een vergelijking voor de druk, mits deze maar gekoppeld met de impulsvormingheid wordt opgelost.

2. De familie van $\kappa$-schema’s kan, voor $0 < \kappa < 1$, op eenvoudige wijze worden voorzien van flux-beperkende eigenschappen.

3. De validatie van berekeningen van turbulente stromingen is een Sisyphus-arbeid.

4. Aan het bezwaar dat vaak wordt ingebracht tegen B-spline oppervlakken als gereedschap voor het modelleren van scheepsgeometrie, n.i. dat ze zich slecht lenen tot locale verfijning, is effectief tegemoet te komen.

5. De term ”hull efficiency” als benaming van het onderdeel van het voortstuwingsrendement van een schip dat bepaald wordt door de verhouding van $1 - t$ en $1 - w$ (waar $t$ het zooggetal is en $w$ het volgstroomgetal) is slecht gekozen; ”interaction efficiency” zou een betere aanduiding zijn.


7. De toonsoorten die Bach heeft gekozen voor zijn cellosuites en de discordering die hij voorschreef voor de vijfde suite wijzen erop dat het gebruik van de losse snaren een belangrijke rol speelt bij de uitvoering van die suites.


9. De claxon, als vast attribuut van personenauto's, kan worden afgeschaft.


11. De kwaliteit van een werkomgeving voor toepassingsgericht onderzoek wordt in sterke mate bepaald door mensen die effectief leiding kunnen geven op grond van hun vakkennis en die tevens kans zien die kennis op peil te houden.

12. Als nieuwsmedium is de krant verre te verkiezen boven de tv.

13. De waarde die in sommige kringen wordt toegekend aan persoonlijke netwerken duidt er op dat het in die kringen belangrijker is wie je kent dan wat je kunt.

14. Wie nog iets weggooit is vaak iets kwijt.