STELLINGEN behorende bij
Numerical Simulation of 3-D Flow
through and around Hydraulic Structures
J. Versteegh

I

De methode van kunstmatige samendrukbaarheid geeft de mogelijkheid om de numerieke technieken, die ontwikkeld zijn voor de oplossing van de ondiepwatervergelijkingen, toe te passen op de 3-D Navier-Stokes vergelijkingen voor een onsamendrukbaar medium.

II

Het redelijk isotrope gedrag van het oorspronkelijke Leendertse schema kan verklaard worden uit het feit dat de expliciete tijdsintegratie van de convectie termen een anisotropie veroorzaakt die tegengesteld is aan de anisotropie ten gevolge van de ruimtelijke discretisatie.

Abbott, M. B.,Computational Hydraulics,
Pitman, 1979, pag. 207

III

Een stromingsprogramma gebaseerd op de volledige 3-D vergelijkingen kan theoretisch uit minder programma statements bestaan dan een quasi 3-D programma. Dit maakt toepassing van het eerstgenoemde programma aanvaardbaar ook in gevallen waarin de versnellingen in een van de coordinaatrichtingen verwaarloosd mogen worden.

IV

Door Brian is in 1961 een ADI methode gepubliceerd voor de oplossing van 3-D diffusie problemen. Ten onrechte noemt Noye deze methode instabiel en slechts 1^eorde nauwkeurig. Bewezen kan worden dat Brian's methode die berust op de combinatie van de ideeen van Douglas-Rachford (3-D stabiliteit) en Peaceman-Rachford (2^e orde nauwkeurig), volledig gelijkwaardig is aan de later gepubliceerde methode van Douglas-Gunn met een impliciethеidsfactor $\beta={}^1/_2$. Derhalve is Brian's methode onvoorwaardelijk stabiel en 2^e orde nauwkeurig.

B.J. Noye, Num. Sim. of Fluid Flow, North Holland Publ. Co.,1978

P.L.T. Brian, A Fin.Diff.Method of High Order Accuracy for the Sol. of 3-D Transient Heat Conduction Problems, A.I.Chem.Engg.J.,Vol.7,no 3,1961

V

Behalve door verticale integratie van de 3-D Navier-Stokes vergelijkingen voor een incompressibele vloeistof, kunnen de ondiepwater vergelijkingen ook direkt afgeleid worden van de 2-D vergelijkingen voor een compressibel medium. De fysisch meest logische substitutie is hierbij :

$$\rho = \rho_0(h+h_0) \qquad p = {}^1/_2\ \rho_0 g h^2$$

met ρ = 2-D dichtheid [kg/m^2]
ρ_0= 3-D dichtheid [kg/m^3]
h, h_0= waterstand,diepte [m]
p = 2-D druk [N/m]

C.B. Vreugdenhil,Computational Hydraulics, An Introduction, Springer Verlag, 1989

VI

De ondiepwatervergelijkingen (zonder horizontale diffusie termen) kunnen worden opgevat als de Euler vergelijkingen waaraan bodemwrijvingstermen zijn toegevoegd. Voor het backstep probleem is het theoretisch mogelijk het bijbehorende stromingspatroon te bepalen op grond van het feit dat de druk in het "schaduwgebied" constant is.

VII

De plannen om langs de vaarwegen een natuurlijke oeververdediging, b.v. in de vorm van rietbegroeing, toe te passen, dienen doorgezet te worden, ook wanneer dit beperkingen op de vaarsnelheid van de binnenvaart met zich mee zou brengen.

VIII

Gezien het gerechtvaardigde verlangen van de wereldbevolking tot gelijkberechtiging op dit punt, moet in Nederland de autodichtheid in relatie tot het inwonertal met minstens de helft worden teruggebracht.

IX

Een wiskunde methode voor HAVO/VWO die wat betreft het differentieren en integreren gebaseerd is op eindige differenties levert didaktische voordelen op, die ruimschoots opwegen tegen het nadeel dat de formuleringen "benaderend" zijn.

X

De ervaring leert dat de geestelijke prestatie veel minder dan evenredig afneemt met het aantal uren in een werkweek. In een aantal gevallen geldt dit ook voor de materiele prestatie. Vanwege de eruit voortvloeiende verbetering in de verdeling van de werkgelegenheid verdient daarom de part-time werkvorm meer dan alleen verbale stimulatie.

XI

Aangezien ongelukken gewoonlijk het gevolg zijn van onvoorziene omstandigheden, dienen veiligheidsanalyses gebaseerd op simulatie met modellen met enige argwaan bekeken te worden.

The Numerical Simulation of THREE-DIMENSIONAL FLOW through or around Hydraulic Structures

Proefschrift

ter verkrijging van de graad van doctor aan de Technische Universiteit Delft, op gezag van de Rector Magnificus, prof. drs. P.A. Schenck, in het openbaar te verdedigen ten overstaan van een Commissie aangewezen door het College van Dekanen op dinsdag 9 januari 1990 te 14.00 uur

door Jacob Versteegh
elektrotechnisch ingenieur,
geboren te Tiel, 13 april 1943.

Dit proefschrift is goedgekeurd door de promotor
Prof. dr. ir. C. B. Vreugdenhil

Voor J.N. Svasek
die mij heeft laten zien hoe ik
water bij m'n wiskunde kon doen

Contents

1. INTRODUCTION

1.1 Numerical Models in Hydraulic Engineering in the Netherlands

In the practice of civil engineering numerical models are accepted as a means of predicting waterflow. While for the final design of important projects physical scale models still are being built, the preliminary designs are more and more tested solely by numerical methods. The obvious reason for this is the fact that the numerical model is cheaper and easier to adapt to different design options.

Because of the high costs involved in the construction and maintenance of a large physical model, generally only the large technological institutes are able to use these models. At the other hand, the costs of the computing power needed to employ large mathematical models decreases. In the last few years is has become feasible, even for small or medium sized organizations like engineering consultants and design offices of construction firms, to utilize large numerical models. By means of such a model designs can be tested without the (often expensive) help of the large institutes.

This change in amount of attention given to physical models on the one hand and numerical models on the other has been going on for some time with respect to 2-D models. As an illustration we can consider the model research in connection to the Delta works, a large project involving the closure of the sea inlets in the SW of the Netherlands. Here, the emphasis has been on the prediction of the tidal movement in coastal waters. To this purpose many physical and mathematical models were constructed. For the

Oosterschelde project a large physical model was built representing an area of about 4000 km^2. For the same area one- and two-dimensional models (IMPLIC, WAQUA) were set up. The shift of interest between physical models and numerical models may appear from counting the number of references to these models in the projects Newsletter "Deltawerken". In the last four years physical models were referenced 2 times against 18 references to numerical models (Deltawerken, 1989).

The kind of modelling discussed so far concerns mainly the numerical simulation of water flow in estuaries and coastal seas. As indicated by the dimensions of the Oosterschelde model, the horizontal scale of such models is two or three orders of magnitude larger than the vertical dimension, i.e. the water depth. Consequently the vertical accelerations are hardly important and in most cases a two dimensional horizontal computation is sufficient to predict the flow pattern. To accomodate stratification and other strong variations in the vertical dimension the model can be organized in layers. A formulation of the mass and momentum exchange between the layers must be included in such a quasi 3-D model, but vertical accelerations are neglected. An example of such a model is described by Leendertse (1989).

Two-dimensional horizontal models of free surface flows are based on the so-called shallow water equations (SWE), which are derived by vertical integration of the Navier-Stokes equations for incompressible fluids. Many programs based on the SWE are in use in the Netherlands (e.g. WAQUA, FINEL, TWOFLOW, DUCHESS). Because of the free surface the SWE can also be regarded as a special form of the 2-D compressible Navier-Stokes equations.

If the three-dimensionality of the flow field is important, physical models are still often employed. But because of the dramatic decrease in computing costs in the last decade, especially for computations performed on mini computers and PC's, it has become possible to study 3-D flow fields by means of numerical models, even on small computer installations.

This thesis is concerned with the application of numerical models in those cases where the horizontal and vertical dimensions are of the same order of magnitude. There is then no justification for treating one space direction in another manner than the remaining two and the full three dimensional Navier-Stokes equations must be solved. Examples of such applications are the prediction of the local flow pattern around or through hydraulic structures like dam heads, bridge piers, lock gates, etc.

As the vertical dimension is governed by the water depth, the horizontal dimensions in this kind of applications will be in the range of say from ten to a hundred meters, i.e. a relatively small area compared to the tidal models mentioned above. This does not mean, however, that the extent of the numerical problem would be less. To describe the velocity variations properly the grid size should be chosen in the order of one meter, leading to a total number of grid points over the three dimensions that often exceeds the number of points in the above mentioned 2-D problems.

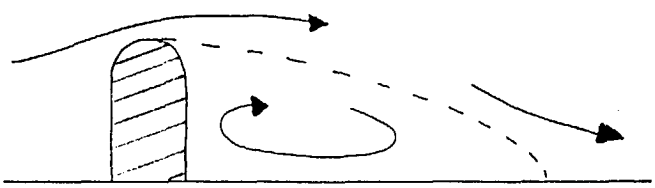
Nevertheless, because of the ever increasing computational power available, it has become feasible to use three-dimensional numerical models for local problems, either as a stand-alone model or as a part of a larger 2-D model.

Due to the relatively small horizontal dimensions it is often sufficient to calculate the stationary flow, as the

time scale in which the flow pattern develops is small as compared to tidal periods, for instance.

In the past many flow situations that are in reality three dimensional in nature have been simulated using 2-D numerical models. This may lead to erroneous results especially when the transport of sediment is considered.

As an example consider the back-flow behind a damhead protruding into a current carrying channel (fig.1.1).



2-D flow around dam head

fig. 1.1

In a stationary 2-D depth averaged calculated flow there will be no mass exchange between the main current and the back-flow region, because of the single dividing flow line between the two regions. For the same reason, in such a model no sediment or other dissolved matter will enter the back-flow region other than by diffusion. In practice, this diffusive transport is very small compared to the total transport, which is mainly convective.

The flow calculation itself is affected because there will be no convective exchange of momentum across the dividing line between the two regions. The backflow is driven

through viscous interaction only. In a 3-D calculation, however, mass flow and thus momentum may enter the region of secondary flow, say near the bottom and leave at the surface. Consequently the backflow will be driven by viscous and convective momentum exchange. In chapter 9 of this thesis we give an example of a flow situation where the results of a 3-D computation differ considerably from those of a 2-D computation.

1.2 Existing Programs

With modern computer facilities the 3-D computation of fluid flow has come within reach of practical engineering applications. Some general purpose-programs are commercially available (Phoenics,Fluent). They are based on numerical schemes that are very stable as robustness is a paramount attribute of this kind of programs. Unfortunately, in many cases this robustness comes at the expense of extraneous numerical diffusivity. Numerical diffusivity or viscosity is caused by the so-called upwind treatment of the convection terms in the Navier-Stokes equations. This upwind technique automatically stabilizes the computational process by adding viscosity as a function of the local convecting velocity. So, although the resulting flow may seem acceptable, there will be added viscosity at critical places in the flow pattern.

It is possible to use higher order upwind schemes. Those schemes are formally free from numerical viscosity, but the connectivity of the computational points increases. This means that for the calculation of a variable in a specific gridpoint more neighbouring points are used, as compared to more simple schemes. This circumstance may lead to problems

near the walls. Also these schemes are often sensitive to direction in respect to the computational grid.

The general-purpose character of commercial programs causes, in many cases, the use of an unnecessary amount of memory for specific applications. This is undesirable if one aims at the possibility of running the program on minicomputers or even PC's. For engineering purposes it is often still too expensive to run extensive programs on mainframe computers where cpu time has to be rented.

In the Netherlands a project has been started (ISNAS) by the three big technological institutes to design a general-purpose program for the Navier-Stokes solution of flow problems. This project proposes to use mainframe and vector machines to solve the equations in a sophisticated way.

In this situation there seems room for a simple and straightforward 3-D method especially designed for use on company owned minicomputers. Such a method is the subject of this thesis. To avoid the problem of numerical viscosity, while maintaining computational simplicity, the method is based on difference approximations that are central in space. The set up of the computation has a close relation to a well tested scheme that is extensively used in the Netherlands for the calculation of 2-D open water flow (the above mentioned SWE programs) and uses a minimum of computer memory.

In this respect it is noteworthy that our method needs only one set of flow variables to be stored during the computation. The method allows updating to be done by successive substitution.

1.3 Basic Principles

1.3.1 The Artificial Compressibility Method

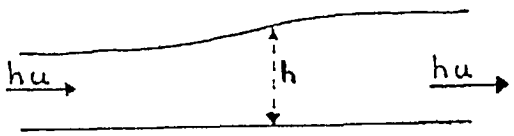
The method we present in this thesis is based on the principle of artificial compressibility. Much work in this area was done by Chorin, who first proposed the scheme (Chorin,1967), and more recently by a group of research workers in the USA (Chang and Kwak,1985, Hartwich,1987). The work of the latter group is based on an artificial compressibility scheme of Steger and Kutler,1977, using the Approximate-Factorization (=ADI) scheme of Beam and Warming (1976). In Europe the method is used by, among others, E. Dick (Dick and Desplanques, 1983).

The method of artificial compressibility (AC) circumvents the basic difficulty of the incompressible Navier-Stokes equations. This difficulty is the mathematical form of the continuity equation :

$$\nabla . \mathbf{V} = 0$$

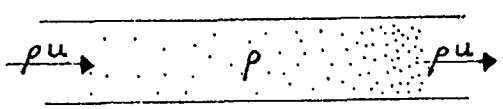
with $\mathbf{V}$ = velocity

which acts as a constraint on the equation of motion. The AC method adds an (artificial) pressure derivative $\partial p/\partial t$ to the continuity equation, which changes, during the transient part of the calculation, the system of equations into a more easily solvable variant of the compressible Navier-Stokes equations. If the solution converges to a stationary situation the extraneous pressure derivative will automatically become zero and the original incompressible continuity relation is satisfied again.



$\partial h/\partial t = - \partial(hu)/\partial x$

a: free surface flow (SWE)



$\partial \rho/\partial t = - \partial(\rho u)/\partial x$

b: real compressible flow

fig. 1.2

Our approach is from a slightly different angle. The numerical technique that is used for the solution of the Shallow Water Equations (SWE), mentioned section 1.1, is in fact a solution method for the compressible Navier-Stokes equations. This can be understood from fig. 1.2. Because of the free surface the continuity equation of the SWE :

$$\nabla.(h\mathbf{V}) = -\partial h/\partial t$$

with h = depth

takes the mathematical form of the compressible continuity equation :

$$\nabla(\rho \mathbf{V}) = \partial\rho/\partial t$$

with ρ = density

The SWE contain a gravity-wave part, convective and bottom-resistance terms. Later, lateral diffusivity terms were added (Kuipers and Vreugdenhil 1973). The solution technique uses a second order accurate Peaceman-Rachford ADI scheme for the wave part. Because the remaining terms cannot easily be fitted into the ADI scheme, they are treated more or less explicitly.

We have tried to use the same approach for the case of the 3-D incompressible flow using the AC principle. As stated above this principle transforms the incompressible Navier-Stokes equations into a variant of the equations for compressible flow, having a time derivative of the pressure in the continuity equation. For stationary flow the numerical solution will approach the incompressible flow solution.

According to this idea the (compressible) SWE technique is in principle applicable to incompressible stationary flow.

By using the AC principle we are able to use more or less the same numerical scheme for the incompressible Navier-Stokes equations as were used for the SWE modelling. Of course the 2-D equations must be extended to three dimensions.

1.3.2 Alternating Direction Implicit (ADI) Method

Because of its efficiency in multi-dimensional problems, the numerical technique most often used for the solution of the SWE is the alternating direction implicit or ADI time integration method (Peaceman and Rachford,1955, Douglas, 1955). In the well known SWE scheme of Leendertse (1967), which is the basis of many subsequent computational programs, this method is used in connection with a finite difference approximation on a staggered grid. (Other, less efficient, but potentionally more versatile procedures use a finite element approximation, which prevents the application of ADI (Praagman,1979)).

In order to use the numerical scheme for the full three dimensional computations the ADI method had to be extended to 3-D. This is not possible with the Peaceman-Rachford ADI scheme. Being marginally stable in 2-D, it is unstable in 3-D. The extension of the ADI technique to three dimensions is well known (Douglas-Gunn,1964) for diffusive problems, but proved to have some consequences on the stability of the numerical calculation for problems of a hyperbolical character like the convection dominated Navier-Stokes equations. An important part of the theoretical work in this thesis consists of the discussion of this aspect of the ADI method in three dimensions.

We propose to use the Douglas-Rachford version of ADI, which has better stability properties but is only first order accurate in time. But as we are already confined to stationary flows by the AC-method, this is not a great disadvantage.

As with the SWE the convective and diffusive terms cannot easily be incorporated in the ADI scheme. We decided to use

simple explicit formulas for the time advancing of these terms.

1.4 Turbulence

As most flows in engineering practice are turbulent, attention must be given to the way turbulence is treated in a numerical model. A common approach is the use of a turbulent viscosity coefficient. Following Boussinesq (1877), in this approach the momentum exchange due to the turbulent motion is modelled as a viscous effect, which is added to the molecular viscosity. The viscosity coefficient thus becomes dependent on the flow conditions. In the more advanced flow models it is calculated for every point of the computational grid (κ-ϵ model).

We use a more simple approach of a constant turbulent viscosity coefficient in combination with a stream-controlled slip condition at fixed walls, using a logarithmic law-of-the-wall approach in the grid cells adjacent to the wall.

1.5 Summary of the Remaining Chapters

In chapter 2 the Navier-Stokes equations are given. A number of commonly used simplifications of these equations is reviewed. As some confusion seems to exsist in literature about the part played by the different terms of these equations in the generation of secondary flows, the effect of the different simplifications on the solution of the equations is studied for a back-step configuration.

While we seek a solution method for incompressible flow, the proposed method is based on the compressible Navier-Stokes equations. The difference between the incompressible, the compressible and the artificial compressible equations is illustrated.

In chapter 3 solution methods for incompressible flow are reviewed. It is shown that there is no basic difference between the pressure correction (PC) method and the method of artificial compressibility (AC). The relative merits of the finite difference (FD) and the finite element (FE) methods are compared. The main advantage of the FD method is the possibility to use an efficient time integration method like ADI.

Chapter 4 deals with the actual solution method as proposed in this thesis. First the spatial approximation is treated in terms of Fourier components. For the time integration we use ADI. Because many variations of this method are in use, the relation of the different schemes to the general Douglas-Gunn formula is expounded. The convection terms are integrated explicitly, separately from the ADI scheme. The explicit steps can be incorporated in different ways in the total computational scheme. Finally the influence of the amount of artificial compressibility on the solution is discussed.

Chapter 5 describes the results of the theoretical study on the stability properties of ADI in three dimensions. To explain the effects of the extension from 2-D to 3-D a graphical method is designed. In this way the destabilizing influence of the extra terms, introduced by the factorization which is inherent in the ADI formulation, is shown in a

clear manner. The main conclusion of this study is that the Douglas-Gunn ADI formula is unconditionally unstable for the inviscid convection equation and only conditionally stable for the convection-diffusion equation.

In chapter 6 we apply the theory developed in chapter 5 to the 3-D Navier-Stokes equations. While is very difficult to derive exact stability limits for the numerical scheme that is adopted, direct calculation of the eigenvalues of the associated Fourier matrix suggests that the same principles are valid. By ordering the explicit and implicit operations in a certain way, the stability of the total scheme can be improved.

A short review of the theory on the relation between the Reynolds stresses and the turbulent or eddy viscosity is given in chapter 7. Some formulas for the eddy viscosity based on the mixing length theory of von Karman are given.

Chapter 8 deals with the boundary treatment. An important disadvantage of the finite difference grid is its inability to represent curved and oblique boundaries. By the use of the porosity method, which can be regarded as a variant of the finite volume technique, this disadvantage can be overcome to some extent. A partial slip is prescribed on solid wall boundaries, which is related to the logarithmic law of the wall. This results in a formulation that generates the correct wall shear stress, independent of the eddy viscosity coefficient used in the fluid body. Thus, the possible ill effect of the simple turbulence modelling near the wall is compensated.

In chapter 9 three examples of flow computations with the present mathematical model are presented. In a fictitious situation, representing a harbor in a river bend, the possible large difference between a depth-averaged 2-D and a full 3-D model is demonstrated. The remaining two applications of the numerical method are based on real projects.

Finally, conclusions are given in chapter 10.

2. THE GOVERNING EQUATIONS OF FLUID FLOW

2.1 Introduction

In this chapter we shall discuss the equations that describe the flow of fluids. The most general description is given by the Navier-Stokes equation, which is a combination of
Euler's equation of motion (1755) and Stokes' friction resistance terms (1845). Theoretical solutions of the full equations are difficult to obtain. In many cases a more useful approach is to neglect those terms in the general equation that are considered less important in the application to a specific class of problems. Since the advent of automatic computers the numerical solution of the full equations has become possible, but for the understanding of the effect of the various terms it is useful to review some of the theoretical solutions of the different simplified forms of the full equation that are in use.

In the next sections we present first the full Navier-Stokes equation and then the different approximations. As an illustration we give the solution for the 2-D flow over a backstep for each approximation.

The second part of the chapter deals with the Navier-Stokes equations for a compressible medium. While we are interested in the flow of water, which can be taken as incompressible in our application, the compressibility plays an important role in the solution technique we will use.

Because our discussion of the stability of the numerical process is based on Fourier analysis, a short description of the Fourier transform of the flow field is given at the end of this chapter.

2.2 The Navier-Stokes Equations

The Navier-Stokes equations are the mathematical form of the law of conservation of momentum. They state that the change in momentum in a infinitesimal volume must balance the pressure gradient and the net convective and diffusive transport of momentum through the surface of this volume-element. We will present the Navier-Stokes equations for the case of an incompressible fluid. They may be written as a single vector equation :

$$\partial \mathbf{V}/\partial t + {}^{1}/_{\rho}\ \text{grad}\ p + (\mathbf{V}.\text{grad})\ \mathbf{V} - \nu\ \text{div}(\text{grad}\ \mathbf{V}) = 0 \qquad (2.1)$$

This vector equation relates the local acceleration $\partial \mathbf{V}/\partial t$ to the pressure gradient grad p ,to the convective acceleration $(\mathbf{V}.\text{grad})\ \mathbf{V}$, and to the viscous momentum transfer ν div(grad V). For homogeneous incompressible fluids, the density ρ is a constant.

Together with the equation of continuity :

$$\text{div}\ \mathbf{V} = 0 \qquad (2.2)$$

equation (2.1) constitutes the basis for the flow-calculations in this thesis.

Remark :

Due to the homogeneity of the fluid gravitational effects are completely balanced by the buoyancy. Thus, the hydrostatic part of the vertical pressure distribution can be ignored.

Although we confine ourselves to stationary flow it is advantageous to keep the time derivative in (2.1). In the artificial compressibility (AC) method that we propose to use, we will even add a time derivative to (2.2). In many cases it is easier to treat stationary flow as the equilibrium solution for steady boundary conditions of the dynamic equations than to solve the stationary equations proper.

The last term on the left side of (2.1) has the form given here only if the fluid is incompressible and/or the coefficient of viscosity ν is a constant. This last restriction is not important as long as ν is regarded as the molecular viscosity coefficient because for water it is nearly constant under ordinary conditions. But whereas we will use the coefficient of viscosity to model turbulence according to the Boussinesq approximation the assumption that ν is constant is too restricted. This aspect is treated in chapter 7.

As given above the Navier-Stokes equations are expressed in the so-called "primitive variables" : u, v, w and p. Another possibility, for incompressible flow, is the use of the vorticity ω and the streamfunction **F**. These vector variables are related to the "primitive" velocity vector by :

$$\omega = \nabla \times \mathbf{V} \tag{2.3a}$$

$$\nabla \times \mathbf{F} = \mathbf{V} \tag{2.3b}$$

Because of its definition (2.3) the streamfunction **F** automatically satisfies the equation of continuity (2.2) while the use of the vorticity ω in the equation of motion eliminates the pressure term.

The formulation is especially useful in 2-D because then only the z-components of **F** and ω appear in the equations. Consequently, F_z and ω_z can be treated as scalars :

$$\omega_z = \partial u/\partial y - \partial v/\partial x$$

$$u = \partial F_z/\partial y \qquad v = -\partial F_z/\partial x$$

By cross differentiation the 2-D N.-S. equations then can be stated as :

$$\partial\omega/\partial t + u\,\partial\omega/\partial x + v\,\partial\omega/\partial y - \nu(\partial^2\omega/\partial x^2 + \partial^2\omega/\partial y^2) = 0$$
$$\partial^2 F/\partial x^2 + \partial^2 F/\partial y^2 = \omega \qquad (2.4)$$

The first equation of (2.4) is a transport equation for the vorticity, while the second is a Poisson equation for the stream function. The pressure does not explicitly appear.

2.3 Approximations of the N.-S. Equations

In order to give an impression of the importance of the different terms of (2.1) we will discuss the influence they have on the flow pattern in the well known configuration of the backstep (figs. 2.1-5).

2.3.1 Potential Flow

The simplest useful approximation of incompressible fluid flow is given by the potential flow representation. In potential flow, viscosity and wall friction are neglected.

It is further assumed that no vorticity enters the computational domain through the boundaries. Consequently the flow will be irrotational, which means that from the convective term :

$$(\mathbf{V}.\text{grad})\mathbf{V} = {}^1/_2 \text{ grad } \mathbf{V}^2 - \mathbf{V} \times \text{rot } \mathbf{V}$$

the last term on the righthand side equals zero. With these modifications the equation of motion for potential flow reads:

$$\partial \mathbf{V}/\partial t + \text{grad}(p/\rho) + {}^1/_2 \text{ grad } |\mathbf{V}|^2 = 0 \qquad (2.5)$$

or in the case of stationary flow:

$$\text{grad } (p/\rho + {}^1/_2 |\mathbf{V}|^2) = 0 \qquad (2.6)$$

The name potential flow is derived from the fact that because rot $\mathbf{V} = 0$, according to Helmholtz' rule, $\mathbf{V}$ can be written as :

$$\mathbf{V} = \text{grad } P$$

with P a scalar potential.
With div $\mathbf{V} = 0$ we get the equation of Laplace:

$$\text{div } (\text{grad } P) = 0$$

The flow pattern for the potential flow approximation for a backstep configuration is given in fig. 2.1b, while in fig. 2.1a the corresonding streamfunction is depicted. Because in the case of potential flow :

$$\nabla \times \mathbf{V} = \omega = 0$$

the streamfunction also satisfies the equation of Laplace, according to (2.4). Fig. 2.1a is the graphical result of the numerical solution of this equation, while the corresponding velocity field (fig. 2.1b) is obtained by taking the gradient of the streamfunction.

2.3.2 Euler Equations

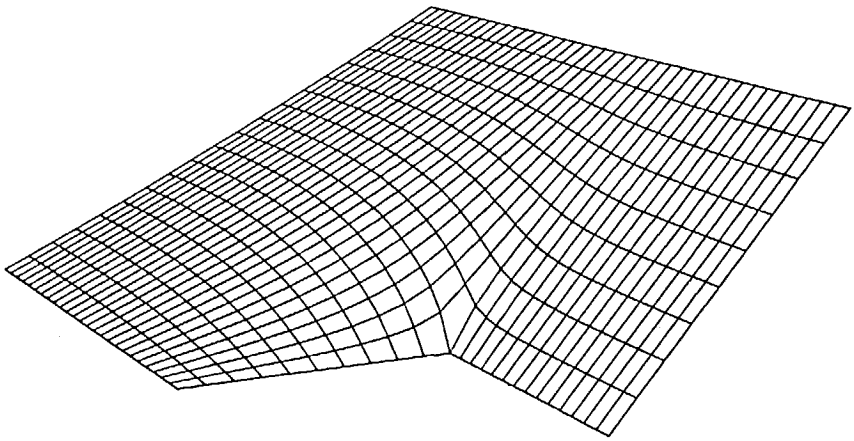
If the restriction, that the flow is irrotational, is removed we get the Euler equations for the flow of an ideal fluid viz. a flow with no viscosity and no wall friction

The equations then incorporate the full convective term:

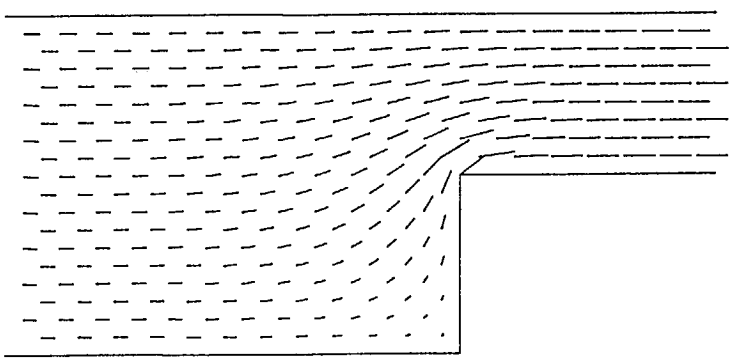
$$\partial \mathbf{V}/\partial t + \text{grad}(p/\rho) + (\mathbf{V}.\text{grad})\mathbf{V} = 0 \qquad (2.7)$$

Because of the now possible vorticity the streamfunction and the velocity field for the backstep configuration change drastically. In this special case the streamfunction and the corresponding velocity field can be constructed easily. The Euler streamfunction and flow field are given in fig. 2.2.

It is still possible to define a velocity potential but now this potential is a combination of a scalar and a vector potential (Helmholz' rule). Thus, the resulting flow field may be regarded as a potential flow field to which a rotational flow field is added. In our 2-D example the vectorpotential degenerates into a single component, so the rotational component of the flow can be obtained by a simple subtraction of the potential flow field from the Euler flow field. The result of this subtraction, both for the streamfunction and the velocity field, is given in fig. 2.3.

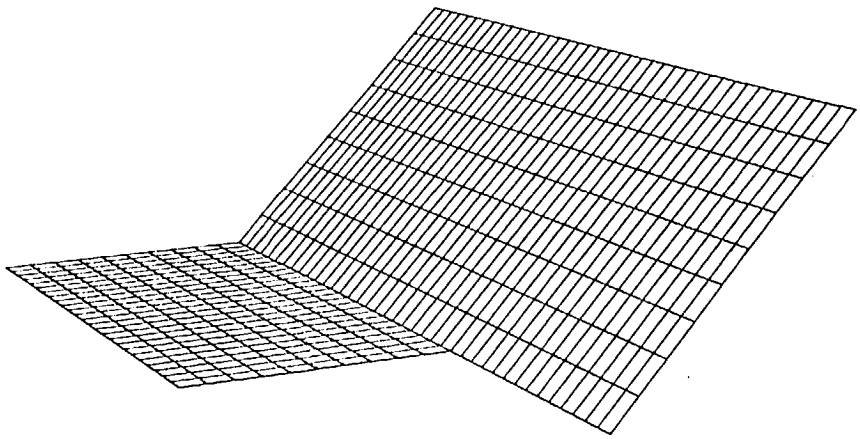


a: Potential Flow : Stream function

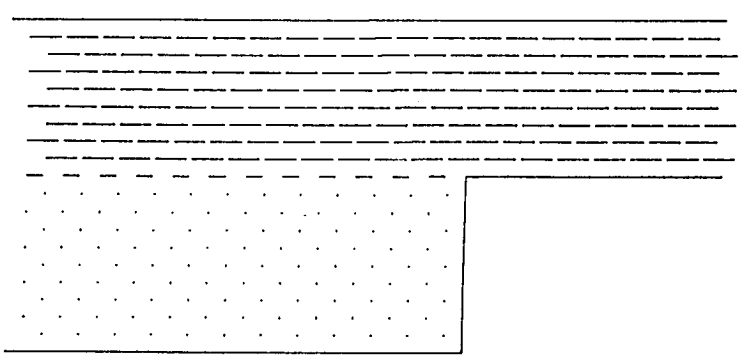


b: Potential Flow : Velocity field

fig. 2.1

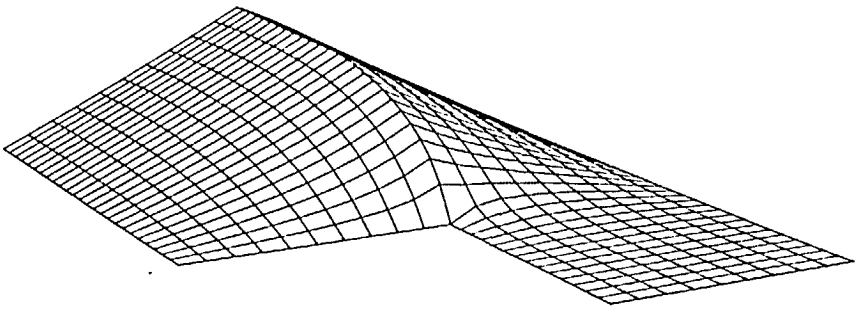


a: Euler equation : Stream function

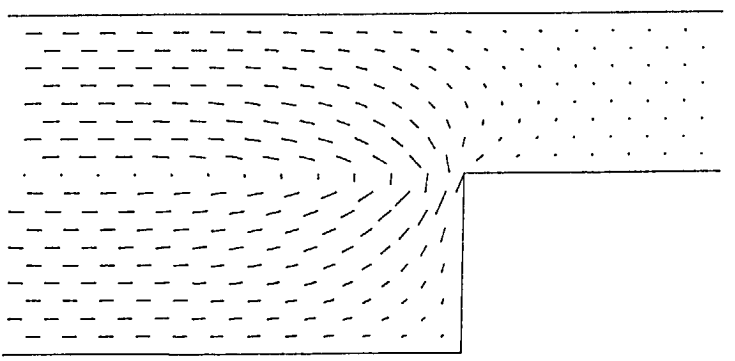


b: Euler equation : Velocity field

fig. 2.2

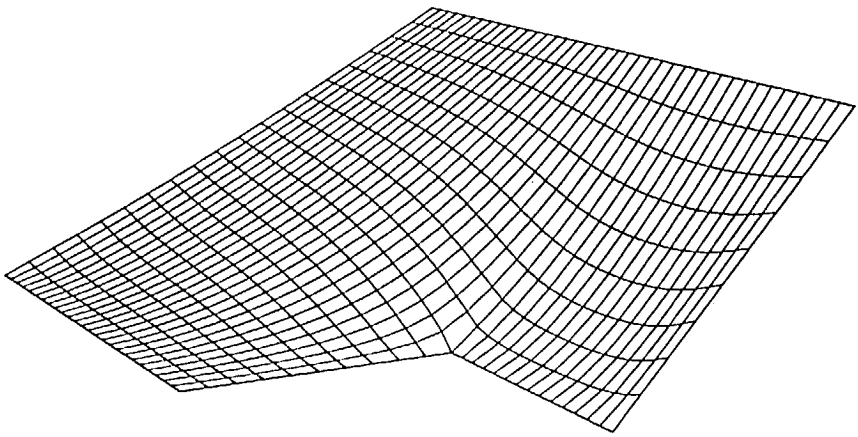


a: Euler minus Potential Flow : Stream function

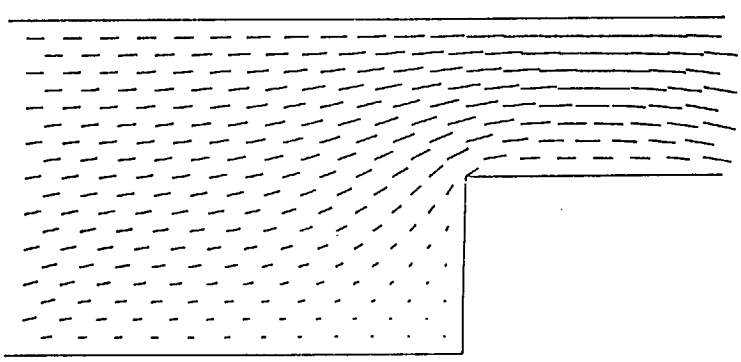


b: Euler minus Potential Flow : Velocity field

fig. 2.3



a: Stokes equation : Stream Function



b: Stokes equation : Velocity Field

fig. 2.3

2.3.3 Stokes Equation

While not of great interest for the simulation of water flow which generally can be regarded as low viscosity fluid, we mention, for sake of completeness, the Stokes equation. This is an approximation of the N.-S. equations where the convection terms are omitted, which is allowable when the viscous forces are predominant, for instance in the case of creeping flows. In 2-D the Stokes equation can be put in a very simple form if we use the vorticity-streamfunction representation :

$$\partial\omega/\partial t = \nu(\partial^2\omega/\partial x^2 + \partial\omega/\partial y^2) \qquad (2.8a)$$

$$\omega = \partial^2 F/\partial x^2 + \partial^2 F/\partial y^2) \qquad (2.8b)$$

For stationary flow (2.8a) and (2.8b) can be combined in the biharmonic equation:

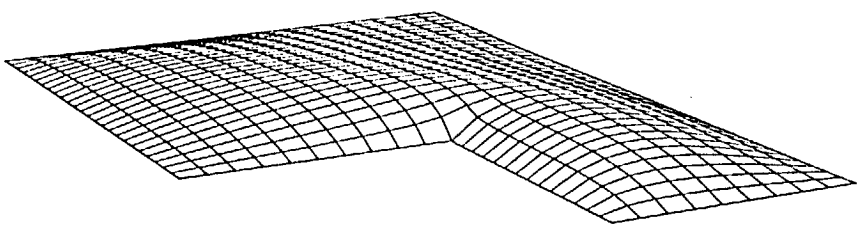
$$\partial^4 F/\partial x^4 + 2\ \partial^2 F/\partial x^2\ \partial^2 F/\partial y^2 + \partial^4 F/\partial y^4 = 0 \qquad (2.9)$$

Remark :

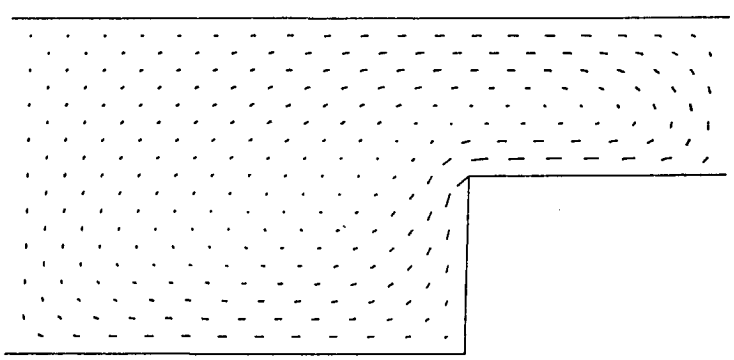
This equation also describes the (small) deflections of a thin plate hinged or clamped at the boundaries but otherwise without transverse loading (Thom and Apelt,1961). Such a plate thus is a direct analogue to the streamfunction for the case of creeping flow. A clamped boundary corresponds to a no-slip condition, a hinged boundary to a free slip condition.

In case of free slip boundary conditions we get, for the backstep problem, the same solution as for potential flow. With no-slip boundaries, however, the solution is as depicted in fig. 2.4 (Thom and Apelt,1961). This solution

again can be split according to Helmholz' rule in a potential component and a rotational component. The rotational component is given in fig. 2.5.

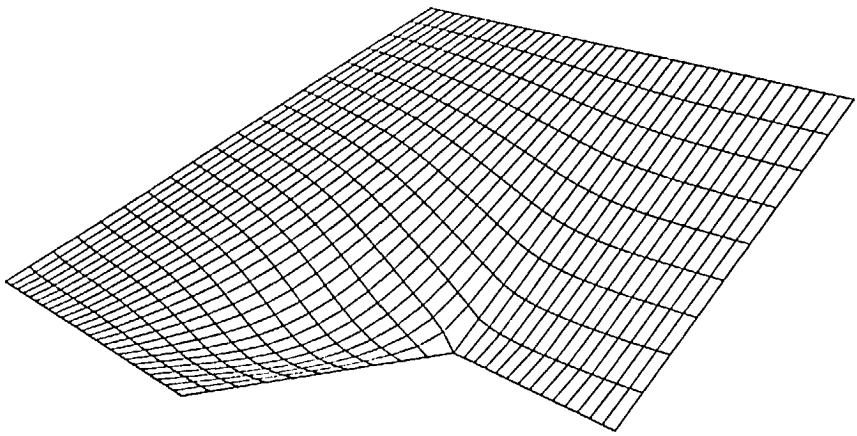


a: Stokes minus Potential Flow : Stream function

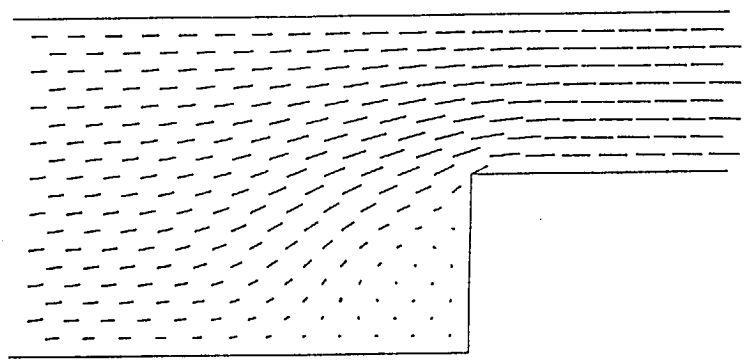


b: Stokes minus Potential Flow : Velocity Field

fig. 2.5



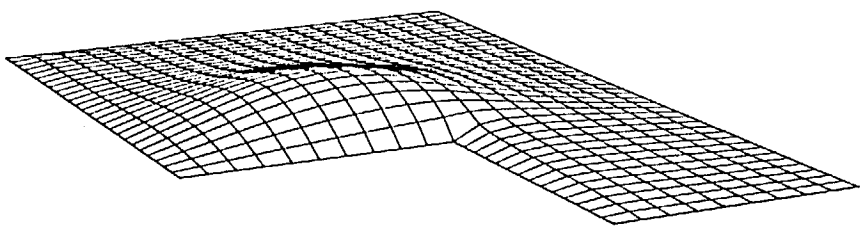
a: Navier-Stokes equation : Stream Function



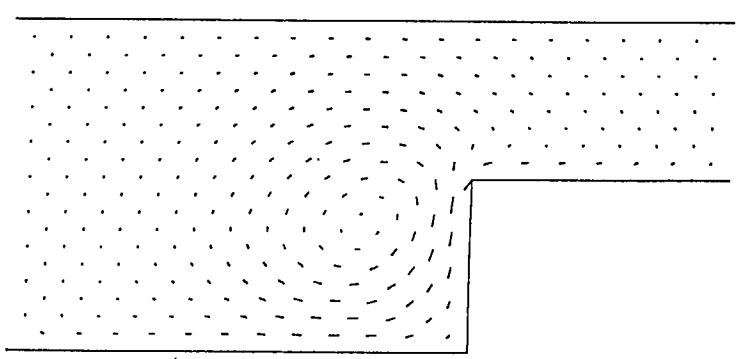
b: Navier-Stokes equation : Velocity Field

fig. 2.6

For comparison we give in fig. 2.6 the streamfunction (a) and the velocity (b) for the full Navier-Stokes equations. In this example a Courant number of 0.5 (based on the inflow) and a diffusion number of 0.025 was used (see section 5.3). In fig. 2.7 the difference with the potential flow field is given.



a: Navier-Stokes minus Potential Flow :
Stream Function



b: Navier-Stokes minus Potential Flow :
Velocity Field

fig. 2.7

2.3.4 Burgers Equation

Burgers equation is obtained by neglecting the pressure gradient in (2.1). The result is a non-linear transport equation. Because in this thesis we are considered with stationary flow the non-linearity of the equations poses no great problems. So we mention this approximation of the Navier-Stokes equation without further discussion.

2.4 Compressible Medium

2.4.1 Shallow Water Equations

Because we propose a solution method based on the concept of artificial compressibility, we now extend our discussion to compressible flows. For a compressible fluid the equation of motion (2.1) is the same as for the incompressible case (except for the viscosity term, which in this case is more complicated). The equation of continuity changes into:

$$\partial\rho/\partial t + \text{div}\ (\rho \mathbf{V}) = 0 \qquad (2.10)$$

For a complete description of the flow, a relation between the now variable density ρ and the pressure p is needed. In the general compressible case this relation is found in further thermodynamic equations.

A special case of the compressible equations are the so-called shallow water equations (SWE). These equations describe the 2-D movement of a fluid with a free surface. Due to this free surface the 2-dimensional divergence may differ from zero, giving the equation of continuity the same mathematical form as (2.10) :

$$\partial h/\partial t + \text{div}\ (h.\mathbf{V}) = 0 \qquad (2.11)$$

Instead of a variable density ρ, we now have a variable depth h and a constant density ρ_0. The relation between water depth h and pressure p follows from :

$$\text{grad}\ p = \rho_0 g\ \text{grad}\ h \qquad (2.12)$$

with g = acceleration of gravity

With (2.12) the pressure p can be eliminated from the equation of motion, so no further relations are necessary. In (2.11) and (2.12) a horizontal bottom is assumed. For a variable bottom topography, in (2.12) the surface elevation should be used instead of the depth h.

2.4.2 Artificial Compressibility

In the method of artificial compressibility that we will use in this thesis the time derivative of the pressure is used as an error term without real physical meaning. As long as this term differs from zero the equation of motion (2.1) will cause the flow pattern to change towards more divergence-free flow. For the artificial compressibility method the equation of continuity reads :

$$\alpha_p \partial p/\partial t + \rho_0 \text{div}(\mathbf{V}) = 0 \qquad (2.13)$$

with α_p a compressibility factor having a dimension of $[\text{sec}^2/\text{m}^2]$

The density ρ is again considered a constant ($=\rho_0$). Note that, in contrast to (2.11), (2.13) is a linear equation. Thus, the AC equation of continuity is less complicated than in the "true" compressible case.

Both the SWE and the AC forms of the continuity equation give the system the mathematical character of the true compressible equations. The main difference between incompressible and compressible equations is the possibility of compression waves in the latter case. In a compressible medium this compression wave is propagated with the speed of sound (for water about 1000 m/sec). In the case of the SWE this propagation speed follows from $c = \sqrt{(gh)}$, where h is the water depth. For the AC method the propagation speed of the compression wave amounts to:

$$c = \sqrt{(1/\alpha_p)} \qquad (2.14)$$

The propagation speed as given by (2.14) is valid only if the influence of the convection is neglected. The true propagation speed of the AC method is given in section 4.5. The value given by formula (2.14) can be regarded as the still-water propagation. In the next chapter a graphical illustration (fig. 3.1) is given of the different forms of the continuity equation.

2.5 Fourier Transformation

A general stationary flow field can be considered as being built up from spatial Fourier modes. Formally, this decomposition is possible if the flow satisfies the

linearized Navier-Stokes equations and periodic boundary conditions. We can discern pressure modes of the form:

$$p = \hat{p} \exp (i \mathbf{k}.\mathbf{r}) \qquad (2.15)$$

with $\hat{p}$ = complex amplitude of p

$i = \sqrt{-1}$

$\mathbf{k}$ = wave number vector

$\mathbf{r}$ = position vector

and flow modes:

$$\mathbf{V} = \hat{\mathbf{V}} \exp (i \mathbf{k}.\mathbf{r}) \qquad (2.16)$$

with $\hat{\mathbf{V}}$ = complex amplitude of $\mathbf{V}$

Note: We use the symbol i for the imaginary unit as well as for the grid numbering in the combination i,j,k for the three directions. The intended use will be clear from the context.

Flow modes can be divided in longitudinal modes where the velocity amplitude vector $\hat{\mathbf{V}}$ is parallel to the wave number vector $\mathbf{k}$, and transverse modes where $\hat{\mathbf{V}}$ is perpendicular to $\mathbf{k}$. Insertion of (2.16) into (2.2) shows that in an incompressible medium only transverse modes can exist. They move through the field with the speed of the convecting velocity $\mathbf{V}_c$ which consist of the summation over all modes of $\mathbf{V}$ itself.

In a compressible medium longitudinal flow modes together with the pressure modes travel with a finite propagation speed as compression waves.

This means that in the artificial compressibility method, during the transient part of the computation, we have to

deal with two different transfer velocities, viz. the speed of the artificial compression wave $c=\sqrt{1/\alpha_p}$ and the convecting velocity which is the velocity of the flow itself.

As will be explained in section 4.5 the true propagation speed of the pressure wave will always be higher than the convecting velocity. This means that the (artificial) Mach number of the flow stays below unity. The flow is subsonic for every value of the compressibility factor α_p.

3. SOLUTION METHODS FOR INCOMPRESSIBLE FLOW

3.1 Introduction

As mentioned before, the condition div V = 0 poses a difficulty for the numerical solution of the incompressible flow equations. The reason for this difficulty can be made clear in the following way.

Numerical solution of the system (2.1),(2.2) involves (after linearization) the solution of a system of linear equations which, for the 2-D case, can be written as:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ p \end{bmatrix} = 0 \qquad (3.1)$$

where the matrix elements are submatrices over the grid and u, v and p grid-vectors for the velocity components and the pressure.

The incompressible continuity equation (2.2) causes the submatrix a_{33} to be identically zero, because the pressure does not appear in the continuity equation. In any numerical solution technique it is awkward to have a zero on a diagonal of the linear system to be solved.

3.2 Vorticity-Streamfunction Representation

A well known technique to avoid the above mentioned problem is the use of the vorticity-streamfunction representation (see section 2.1). The streamfunction by definition satisfies the continuity equation, while the expression of vorticity allows a combination of the equations

of motion from which the pressure is eliminated. In 2-D this results in a single equation for the transport of vorticity and a Poisson equation for the streamfunction.

In 3-D, unfortunately, there is no reduction in the number of equations to be solved, because both the vorticity and the streamfunction have three components in this case, resulting in a system of six equations. Although it is possible to solve the equations in this way there is no advantage over the primitive equations as in the 2-D case. Boundary conditions are generally difficult to state in streamfunction-vorticity methods.

3.3 Pressure Correction Method

Harlow and Welch proposed in 1965 a 3-D method using the primitive variables p,u,v,w. Their solution is known as the Pressure Correction or PC method. In this method one advances in time by first calculating intermediate values of the velocity components from the equations of motion leaving out the pressure terms. In terms of (3.1) this means one temporarily leaves out the last column and the last row of the system (3.1). Using for instance a simple explicit scheme :

$$V' = V - \Delta t(\text{ Conv. } - \text{ Diff.}) \qquad (3.2)$$

where Δt is a time step and
Conv. and Diff. the convective
and diffusive terms resp.

The true new velocities can be expressed in the intermediate values **V'** plus a correction in terms of the still

unknown pressure gradient. The new velocities must satisfy the continuity equation div **V** = 0. Insertion then results in :

$$\nabla^2(p/\rho) = {}^1/\Delta t \quad \mathrm{div}(\mathbf{V}') \tag{3.3}$$

where ∇^2 stands for the Laplacian.

Solution of this Poisson equation yields the pressure distribution on the new time level, which then can be used to correct the **V'** values. A boundary condition for the pressure on the solid boundaries is necessary but can be derived from the equation of motion (Peyret and Taylor,1983).

3.4 Artificial Compressibility

In 1967 Chorin proposed a seemingly different method known as the Artificial Compressibility (AC) method. In this method an "artificial" time derivative of the pressure is added to the continuity equation, thus removing the difficulty mentioned at the beginning of this chapter. Instead of div **V** = 0 the equation of continuity now reads:

$$\alpha_p \partial p/\partial t + \rho_0 \mathrm{div}\ \mathbf{V} = 0 \tag{3.4}$$

This equation differs from the true compressible continuity equation in two ways. Firstly by the appearance of the compressibility factor α and secondly by the fact that the second term of (3.4) is independent of the density ρ. This last circumstance makes the solution easier because it causes (3.4) to be a linear equation. The compressibility factor determines the propagation speed of the artificial

pressure wave and can be used to affect the "time" in which the steady state is reached. Its influence is discussed in more detail in section 4.4.5.

Mathematically the system (2.1),(3.4) has the form of the compressible Navier-Stokes equations and in principle it can be solved by the same methods that are used for the SWE. This is an important advantage as there is a lot of computational experience in this kind of methods. The well known Leendertse method for the SWE is such a method based on the Peaceman-Rachford Alternating Direction Implicit (ADI) scheme. This scheme is centered in time, which means that the time advancing is based on a centered interpolation between the spatial values on the old and on the new time level.

As mentioned in 1.3 this particular scheme is not suitable for 3-D. The ADI scheme we use in this thesis is fully implicit, which means that the time advancing is based on the spatial derivatives of the new time level only. In chapter 5 we will show that, when the convection terms are included in the ADI procedure, such a scheme will never be unconditionally stable in 3-D. Consequently, in our procedure we will treat the convection-diffusion terms separately from the ADI process.

While there is a basic difference between the PC and AC method, because the AC method leads to a hyperbolic system and the PC method to a diffusive or elliptic system, the use of a fully implicit time integrating scheme tends to remove this difference. A heavily damped hyperbolic system behaves like an elliptic system. Indeed, Chorin introduced in 1968 a kind of PC method which he considered as a dynamic form of his earlier AC method.

To explain this we will write down both systems. For simplicity we give the 2-D case, but the same argument is

valid in 3-D. The convection-diffusion terms are denoted by the characters CD.

The PC method :

$$u' = u^n - \Delta t \; CD \tag{3.5a}$$

$$v' = v^n - \Delta t \; CD \tag{3.5b}$$

$$^1/\rho(\partial^2 p/\partial x^2 + \partial^2 p/\partial y^2) = \partial u'/\partial x + \partial v'/\partial y \tag{3.5c}$$

and the AC method :

$$u^{n+1} = u^n - \Delta t \; CD - {}^1/\rho \; \partial p/\partial x \tag{3.6a}$$

$$v^{n+1} = v^n - \Delta t \; CD - {}^1/\rho \; \partial p/\partial y \tag{3.6b}$$

$$p^{n+1} = p^n + \Delta t\{{}^1/\rho(\partial^2 p/\partial x^2 + \partial^2 p/\partial y^2) - \partial u'/\partial x - \partial v'/\partial y \tag{3.6c}$$

Comparing both systems, we see that 3.6c can be regarded as a single iteration step in the solution of (3.5c). Thus, the main difference between both methods is that in the PC method, after execution of steps (3.5a and b), equation (3.5c) is solved for the pressure at every time step, while in the AC method the stationary solution is attained by time stepping, or iterating, over the whole system (3.6).

In the following chapters we will use as pressure term in the Navier-Stokes equations the expression g grad h (g =

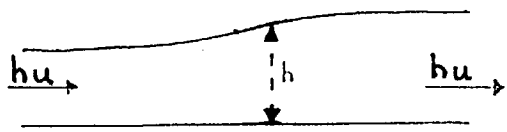
acceleration of gravity) instead of grad(p/ρ). The water level h in this expression should not be confused with a free surface level. In this thesis the letter h is used in the sense of a piezometric head i.e. a measure of the pressure within the fluid. Instead of the compressibility factor α_p we will use $\alpha = g\,\alpha_p$. This formulation permits a physical interpretation of the compressibility factor α. Having the dimension of 1/m it can be understood as the ratio between the free surface (m^2) of the fluid within an imaginary pitot tube and the unit fluid volume (m^3). A picture of this idea is found in fig. 3.1 where the different forms of the continuity equation are depicted schematically.

The still-water propagation speed of the artificial compression wave (2.13) becomes in the new notation:

$$c = \sqrt{(g/\alpha)} \qquad (3.7)$$

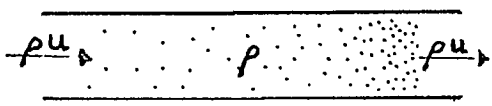
3.5 The Computational Grid

Numerical solution methods require some grid system on which the computation is done. There are methods that discretize only the boundary (boundary-element method) but generally a network of grid lines is constructed over the whole computational domain. Two basic methods of spatial discretisation can be discerned: the finite difference (FD) and the finite element (FE) method. A third, the finite volume (FV) method, is a FD-based method that tries to capture the conservation properties of the FE-method. (see also section 8.2.2.



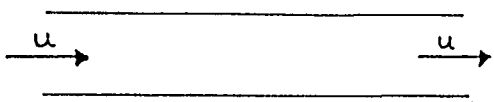
$\partial h/\partial t = - \partial(hu)/\partial x$

a: free surface flow (SWE)



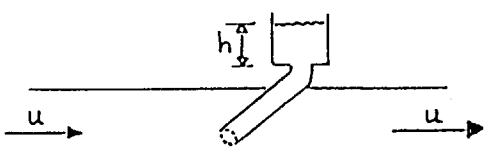
$\partial \rho/\partial t = - \partial(\rho u)/\partial x$

b: compressible flow



$\partial u/\partial x = 0$

c: incompressible flow



$\alpha\ \partial h/\partial t = - \partial u/\partial x$

d: artificial compressible flow

fig. 3.1

Our choice of grid system was guided by the wish to make use of the existing experience in (compressible) SWE solution techniques. Most programs in this area are based on finite differences, because this method is economical in its use of computer memory. With the development of cheaper computing power the more versatile FE methods have become economically feasible but, for 3-D use, these programs require the storage and processing of a great amount of geometrical data which still makes them too extensive for small computers. Moreover, the generation of a FE-grid is a major task.

On the other hand, a regular FD-grid is very easy to set up and requires hardly any geometrical data. In its most simple form, only the number of grid points and grid size in each direction, so six numbers in all, have to be recorded. Of course there are disadvantages, the most important being the inability of the FD-grid to describe irregular boundaries of the computational domain. To counteract this circumstance boundary fitted coordinates are sometimes employed. The grid then follows the boundaries, but is by some mapping method transformed into a regular grid in which the finite differences can be defined.

Another problem is the accuracy of the solution at the solid boundaries. With turbulent flow boundary layers will appear. To describe these boundary layers accurately a very small grid size is required locally. With a regular grid having a constant grid size over the whole field, this leads to a prohibitive number of grid points and, because of the Courant condition, to very small time steps.

Notwithstanding these disadvantages we propose to use the simple regular FD-grid. To enhance the accuracy at the boundaries we will use the so-called "porosity method". In this method grid cells that are cut by the boundary are

mathematically given a surface and a volume that corresponds to the part of the cell that is open to the flow. The geometry itself is left unchanged. The method will be explained in section 8.2. Simulation of the effect of boundary layers is attained by assuming a logarithmic velocity profile in the last grid cell towards a solid boundary according to the "law of the wall". This approach results in a relation between the velocity and its derivative in the last grid point, which can be used as a boundary condition. See section 8.3.

In the next chapter (section 4.2) a more detailed description of the grid is given.

3.6 Conclusions

In this chapter arguments were given for the choice of the solution method. The chosen method is based on the principle of artificial compressibility, making it possible to use solution techniques for compressible flow for the incompressible Navier-Stokes equations. For economical reasons we decided to use a simple finite difference grid.

Thus, we can use existing techniques like ADI for the time integration of the difference equations. These techniques will be discussed in the next chapter.

4. THE NUMERICAL SOLUTION OF THE AC EQUATIONS

4.1 Introduction

According to the method of artificial compressibility the following system of equations has to be solved :

$$\partial \mathbf{V}/\partial t + g \operatorname{grad} h + (\mathbf{V}.\operatorname{grad})\mathbf{V} - \nu \operatorname{div}(\operatorname{grad} \mathbf{V}) = 0$$

$$\partial h/\partial t + 1/\alpha \operatorname{div} \mathbf{V} = 0 \qquad (4.1)$$

with $\mathbf{V} = (u,v,w)^T$
and α = compressibility factor

Remark :

As explained at the end of section 3.4 the appearance of the acceleration of gravity g in (4.1) is connected to the use of the piezometric head h instead of the pressure proper.

The first terms of the two equations (4.1) are time derivatives, the remaining terms contain only spatial derivatives. The spatial terms can be taken together in a symbolic operator D_a operating on a vector $\phi = (u,v,w,h)^T$. We get:

$$\partial \phi/\partial t + D_a \phi = 0 \qquad (4.2)$$

If we choose as our computational domain a set of discrete points instead of the continuum, the spatial derivatives can be approximated by finite differences between grid points. It is then possible to rewrite the

symbolic formula (4.2) as a system of ordinary differential equations in time :

$$d\psi/dt + D_n \psi = 0 \qquad (4.3)$$

While in (4.2) ϕ and D_a are a 4 component vector and a 4 x 4 matrix respectively, in (4.3) ψ consists of 4 components for each grid point. For N grid points D_n will be a 4N x 4N matrix. Numerical solution of the system (4.3) is obtained by some time-integrating process. Usually this involves discretizing the time axis. The time integration proceeds then from one time level to the next so one can speak of an old and a new time level (ψ^n and ψ^{n+1}, respectively).

This integration process can be represented by :

$$\psi^{n+1} = G_n \psi^n \qquad (4.4)$$

For a simple explicit time integration scheme (Euler) G_n can be expressed as:

$$G_n = (I - \Delta t \, D_n)$$

with I the identity matrix.

Often one chooses an implicit time integrator because of the better stability properties of implicit schemes. In this case G takes the form $A^{-1}B$ and (4.4) becomes :

$$A \psi^{n+1} = B \psi^n \qquad (4.5)$$

For the well known Crank-Nicolson scheme for instance :

$$A = 0.5 (I + \Delta t \, D_n) \quad \text{and} \quad B = 0.5 (I - \Delta t \, D_n)$$

System (4.5) can be solved either iteratively or directly.

As stated in chapter 1 we will use partly an explicit, partly an implicit (ADI) scheme for the time integration of the system (4.3). This method will be discussed in the following chapters.

Some remarks must be made at this point regarding the linearity of the expressions used above. Although (4.1) is a non-linear equation due to the convective term (**V**.grad) **V** we treat the matrices D_a, D_n and G_n as being independent of ϕ and ψ respectively. The reason for this is that the non-linearity complicates the analysis considerably. By assuming the convective term in (4.1) to be ($\mathbf{V}_c$.grad) **V** where $\mathbf{V}_c$ is a (locally) constant convecting velocity (4.1) becomes a linear equation in **V** and the subsequent linear analysis will be valid.

The following argument for this assumption can be given : In the Fourier representation of the vector of unknowns that we will use in the following chapters, we analyse the behaviour of a single Fourier component. For such a component the convecting velocity consists of the summation of all components. For problems that converge to a stationary situation, this summation will behave more or less as a constant in respect to a single (short wavelength) component.

In the actual solving of (4.5) the system must be linear in ψ^{n+1} so D_n should at most be dependent on components of ψ^n, which means that the convecting velocity in the convective terms is always taken on the old time level. In an explicit method the derivatives are always taken on this level, so linearity of the system is not obligatory in that case.

In section 4.2 of this chapter we give a more detailed description of the spatial discretisation. In 4.3 the time discretisation is treated. A review of ADI methods is given. In the last part of the chapter we discuss the use of ADI for the 3-D artificial compressibility equations.

4.2 The Spatial Discretisation

4.2.1 Grid-staggering

As mentioned in section 3.5 we use a simple regular finite difference grid. The computational domain consists of a rectangular block (fig. 4.1). Technical details are given in section 9.2.1. The actual flow domain is embedded in this block, leaving a number of "dry" grid points that are not

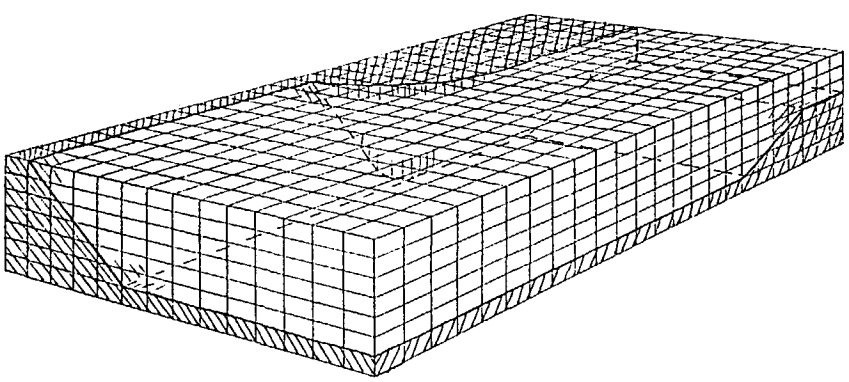


fig. 4.1 Computational domain

used in the computation. It would have been possible to store only those grid points that are actually used. This is common practice in several SWE programs. Such a system

however, requires a complicated addressing system, while the rectangular domain, although not as efficient in its use of memory, has the great advantage that every neighbouring point of the point (i,j,k) can be addressed directly as (i+1,j,k), (i,j-1,k), etc.

Another feature, which is well known from 2-D practice, is the so-called staggered grid. This name is derived from the fact that in this grid the points where the velocity components are defined are shifted away from the pressure points (Hansen,1956, Harlow e.a.,1965). In the Netherlands staggered grids were used in 1-D problems by Dronkers(1964) and in 2-D by Leendertse(1967) and in many subsequent programs.

Another way of looking at a staggered grid is by considering it as a portion of a dense grid (i.e. a grid where all flow variables are defined in each grid point, as in fig. 4.2a). To show this we consider the 1-D wave equation :

$$\partial u/\partial t + g\ \partial h/\partial x = 0$$

$$\alpha\ \partial h/\partial t + \partial u/\partial x = 0$$

If this system of equations is discretized with central difference approximations for the spatial derivatives we get :

$$\frac{du_i}{dt} + g/\Delta x\ (h_{i+1} - h_{i-1}) = 0$$

$$\alpha\ \frac{dh_i}{dt} + 1/\Delta x\ (u_{i+1} - u_{i-1}) = 0$$

This results in two independent systems of differential equations, one system involving "odd" u's and "even" h's,

the other "even" u's and "odd" h's (figs. 4.2b and c). Without loss of accuracy one system can be left out. The result is the staggered grid. In respect to the dense grid the staggered grid contains in 2-D a quarter and in 3-D a eighth of the number of variables of the dense grid.

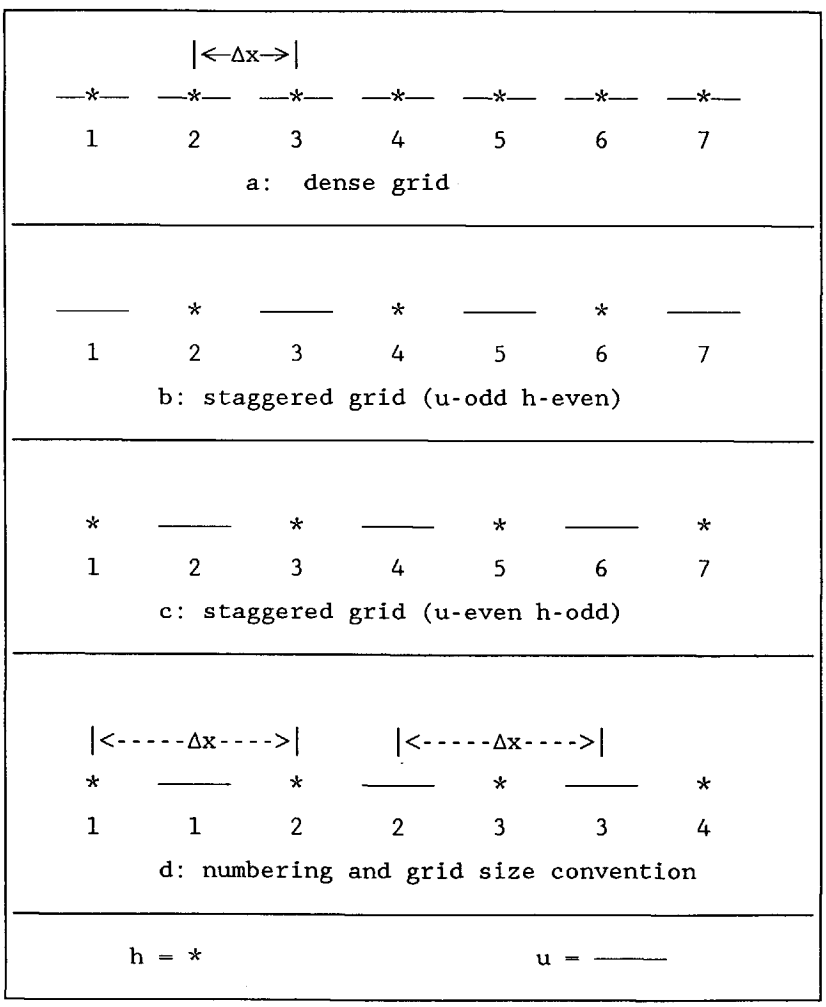


fig. 4.2

While the above derivation leads to a numbering system as in fig. 4.2a, it is customary, in accordance with the original shifting procedure, to consider the distance between two pressure points (or between two corresponding velocity components) as the grid size proper (fig.4.2d).

Of course, the argument given above for the wave equation is not valid for the full Navier-Stokes equations. On the staggered grid the convection and diffusion terms can only be approximated centrally with twice the grid size of the dense grid. For the convection terms we still have the problem mentioned earlier that there are actually two independent contributions to the total solution. A consequence of this fact is the possibility of spurious solutions (see also Stelling,1984).

4.2.2 Difference Approximations

In the system (4.2):

$$\partial\phi/\partial t + D_a\,\phi = 0$$

the symbolic operator D_a contains the following elements:

$$\begin{bmatrix} CD & 0 & 0 & g\ \partial/\partial x \\ 0 & CD & 0 & g\ \partial/\partial y \\ 0 & 0 & CD & g\ \partial/\partial z \\ 1/\alpha\ \partial/\partial x & 1/\alpha\ \partial/\partial y & 1/\alpha\ \partial/\partial z & 0 \end{bmatrix}$$

where CD stands for:

$$CD = u_c \partial/\partial x + v_c \partial/\partial y + w_c \partial/\partial z - \nu(\partial^2/\partial x^2 + \partial^2/\partial y^2 + \partial^2/\partial z^2)$$

The system (4.3):

$$d\psi/dt + D_n \psi = 0$$

is built up by evaluating (4.2) in every grid point and substituting central differences for every spatial derivative.

For most terms of (4.1) this is a straightforward matter. For the convection terms however, several possibilities exist. Taking the term $u\ \partial u/\partial x$ as an example, the most simple form is:

$$u\ \partial u/\partial x \approx u_{i,j,k}(u_{i+1,j,k} - u_{i-1,j,k})$$

Other possible formulations are given in section 8.2. If however, as done in the Fourier analysis given below, the convecting velocity is considered a constant, these formulations are equivalent to each other.

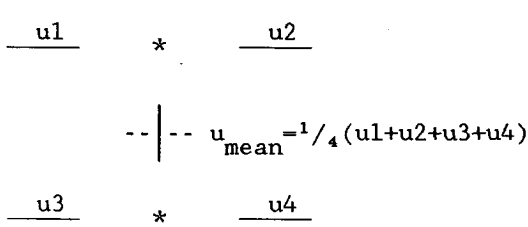


fig. 4.3 Averaging procedure

Because of the staggered grid it is not always possible to specify the velocity components on the location where they are needed by the difference formula. In this case the value is obtained by centrally averaging over 4 surrounding points. See fig. 4.3.

4.2.3 Fourier Analysis of the Spatial Solution

To analyse the spatial discretisation by Fourier methods we must assume a constant convecting velocity. Then the problem becomes linear and can be analysed using Fourier transforms. By means of such an analysis we can compare the analytical system (4.2) with its spatial difference approximation (4.3). Theoretically, this analysis is valid for periodic boundary conditions only, but it gives a clear insight into the effects of the discretisation on the amplitude and phase of the Fourier components of the solution.

For this purpose we substitute in both (4.2) and (4.3) a single Fourier component :

$$\psi(x,y,z,t) = \hat{\psi}(t)\ \exp(i\ \mathbf{k}.\mathbf{r}) \qquad (4.6)$$

where $\hat{\psi}$ = complex amplitude of ψ
$i = \sqrt{-1}$
$\mathbf{k}$ = vectorial wave number with components $\kappa_x, \kappa_y, \kappa_z$
$\mathbf{r}$ = position vector

Remark :

Because $\hat{\psi}$ is a complex entity it represents both the amplitude and the phase of the spatial solution.

Substitution of (4.6) into (4.2) results in :

$$d\hat{\psi}/dt + D_{fa}.\hat{\psi} = 0 \qquad (4.7)$$

where the elements of D_{fa} are :

$$\begin{bmatrix} i\ CD_a & 0 & 0 & i\ g\ \kappa_x \\ 0 & i\ CD_a & 0 & i\ g\ \kappa_y \\ 0 & 0 & i\ CD_a & i\ g\ \kappa_z \\ \frac{i\ \kappa_x}{\alpha} & \frac{i\ \kappa_y}{\alpha} & \frac{i\ \kappa_z}{\alpha} & 0 \end{bmatrix}$$

with $i\ CD_a = i\ (u_c\ \kappa_x + v_c\ \kappa_y + w_c\ \kappa_z) + \nu\ (\kappa_x^2 + \kappa_y^2 + \kappa_z^2)$

N.B. Contrary to its notational appearance $i\ CD_a$ contains a real part due to the diffusion term. The notation is used to indicate that $i\ CD_a$ has a large imaginary component.

In the discrete case (4.6) is defined only on the grid points (i,j,k). This means that the position vector **r** now has discrete components :

$$\mathbf{r}_n = (i\ \Delta x, j\ \Delta y, k\ \Delta z)^T$$

For a sinusoidal function the difference quotients can be expressed as trigonometric functions. Substitution of (4.6) in (4.3) results in a system of 4 equations :

$$d\hat{\psi}/dt + D_{fn}\ \hat{\psi} = 0 \qquad (4.8)$$

where the elements of D_{fn} are:

$$\begin{bmatrix} i\ CD_n & 0 & 0 & \dfrac{i\ \sin(^1/_2\kappa_x\Delta x)}{^1/_2\Delta x} \\ 0 & i\ CD_n & 0 & \dfrac{i\ \sin(^1/_2\kappa_y\Delta y)}{^1/_2\Delta y} \\ 0 & 0 & i\ CD_n & \dfrac{i\ \sin(^1/_2\kappa_z\Delta z)}{^1/_2\Delta z} \\ \dfrac{i\ \sin(^1/_2\kappa_x\Delta x)}{\alpha\ \ ^1/_2\Delta x} & \dfrac{i\ \sin(^1/_2\kappa_y\Delta y)}{\alpha\ \ ^1/_2\Delta y} & \dfrac{i\ \sin(^1/_2\kappa_z\Delta z)}{\alpha\ \ ^1/_2\Delta z} & 0 \end{bmatrix}$$

with $i\ CD_n =$

$$i\ (u_c\ \frac{\sin(\kappa_x\ \Delta x)}{\Delta x} + ..) + \nu\ (\frac{\sin^2(^1/_2\kappa_x\ \Delta x)}{^1/_4\Delta x^2} + ..\ )$$

N.B. Like $i\ CD_a$ above, $i\ CD_n$ contains a real part.

4.2.4 Accuracy of the Spatial Discretisation

Because $\sin(x) = (x + {}^1/_6x^3 + \ldots)$, D_{fn} in (4.8) is a second order accurate, consistent approximation of D_{fa} in (4.7).

In order to give a physical meaning to the error made by the spatial discretisation we write the system represented by (4.8) in a different form :

$$d\hat{\mathbf{V}}/dt + i\ g\ E_1\mathbf{k}\ \hat{h} + i(\mathbf{V}_c.E_2\mathbf{k})\hat{\mathbf{V}} + \nu(E_1\mathbf{k}.E_1\mathbf{k})\hat{\mathbf{V}} = 0$$

$$d\hat{h}/dt + 1/\alpha\ i(E_1\mathbf{k}).\hat{\mathbf{V}} = 0 \qquad (4.9)$$

where E_1 and E_2 are 3 x 3 diagonal matrices with elements :

$$E_1 = \begin{bmatrix} \dfrac{\sin(^1/_2\kappa_x\Delta x)}{^1/_2\kappa_x\Delta x} & 0 & 0 \\ 0 & \dfrac{\sin(^1/_2\kappa_y\Delta y)}{^1/_2\kappa_y\Delta y} & 0 \\ 0 & 0 & \dfrac{\sin(^1/_2\kappa_z\Delta z)}{^1/_2\kappa_z\Delta z} \end{bmatrix}$$

$$E_2 = \begin{bmatrix} \dfrac{\sin(\kappa_x\Delta x)}{\kappa_x\Delta x} & 0 & 0 \\ 0 & \dfrac{\sin(\kappa_y\Delta y)}{\kappa_y\Delta y} & 0 \\ 0 & 0 & \dfrac{\sin(\kappa_z\Delta z)}{\kappa_z\Delta z} \end{bmatrix}$$

Compared to the Fourier transform (4.7) of the original equation (4.1) the wave number $\mathbf{k}=(\kappa_x,\kappa_y,\kappa_z)^T$ is replaced in the spatial difference approximation by E $\mathbf{k}$ where E is a 3 x 3 matrix. By rearranging, these matrices can be combined with the physical coefficients of the Navier-Stokes equations resulting in :

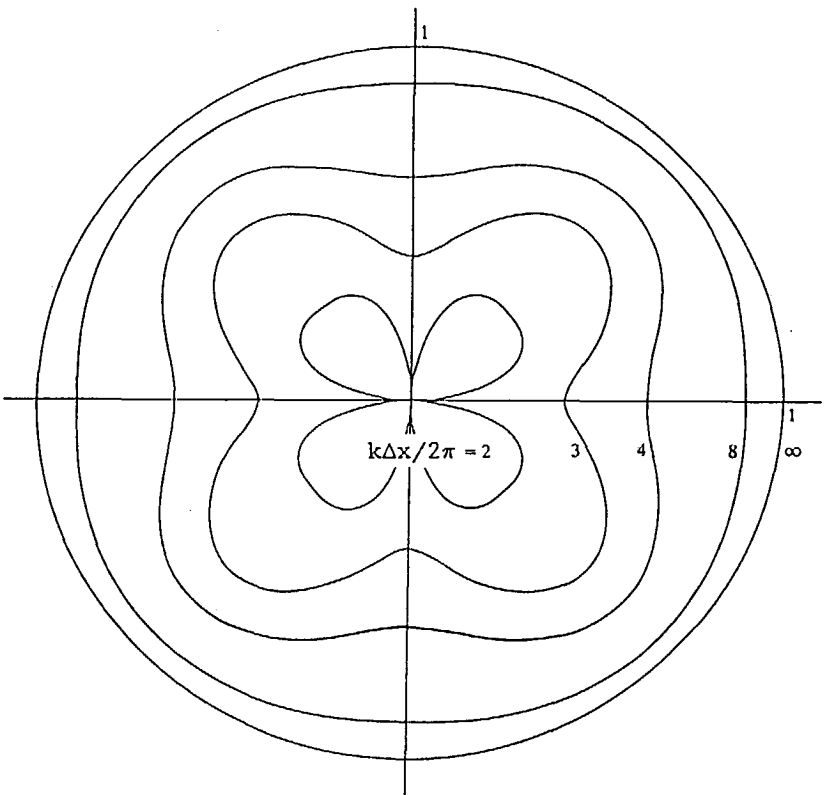
1. a numerical propagation speed: $c^* = E_1 \sqrt{g/\alpha} = E_1\ c$
   with c the physical still water propagation

2. a numerical convective velocity: $V_c^* = E_2\ V_c$

3. a numerical viscosity coefficient: $\nu^* = E_1 E_1\ \nu$

An important consequence of the spatial discretisation is that the medium becomes anisotropic and dispersive. This is caused by the dependency of the elements of E_1 and E_2 on the absolute value and the direction of $\mathbf{k}$. The error is least

for directions of k at equal angles to the three coordinate directions and largest for directions of k that coincide with the coordinate axes. Because the difference approximation of the convection term involves double grid spacings the error in the convection speed is more serious than that in propagation or viscosity. The error in the convection speed is depicted in fig. 4.4 (Vichnevetsky and Bowles,1982) for various values of the wave number.



Directional value of V_c^*/V_c as function of wavenumber k (2-D)
(from Vichnevetsky and Bowles,1982)

fig. 4.4

4.3 Time Discretisation

4.3.1 Explicit and Implicit Methods

Numerical time integration involves the step-wise calculation of the flow values on successive discrete time levels. The new values are calculated from one or more earlier, already calculated, time levels. The starting point is either a known situation or some arbitrary flow pattern. (The latter case, of course, applies to stationary flow problems). Unless the initial situation is known, one has to calculate enough time levels until the starting situation is "forgotten" due to the physical and/or numerical damping. At this point the solution is reached for stationary problems while in the dynamic case the solution has physical meaning from this point on.

Basically time integration methods can be divided into explicit and implicit methods. In an explicit method a new value (i.e. the value of a flow variable on the next time level) is expressed explicitly in flow variables on earlier time levels :

$$\psi^{n+1} = f(\psi^{n}, \psi^{n-1}, \ldots) \qquad (4.10)$$

Depending on the number of earlier time levels in (4.10) one speaks of single- or multi-step methods. Unless one employs time consuming replacing techniques, those earlier levels all have to be kept in memory. In 3-D calculations this leads to great amounts of memory space. In this respect it is advantageous to restrict the number of levels in memory. Therefore, we will limit ourselves to methods that use the last level only. In such a single-step method one may speak of the new and the old time level.

In implicit methods the function f as in (4.10) depends not only on the old time level but also on the new, still unknown, level :

$$\psi^{n+1} = f(\psi^{n+1}, \psi^{n}) \qquad (4.11)$$

Such a system can be solved by iteration (Gauss-Seidel, S.O.R.), but also directly by matrix manipulation (inversion, Gaussian elimination, etc.). A non-linear vector problem can only be solved by some kind of iteration process. However, such a process can consist of a repeated direct solution of the linearized system. In this case the system matrix is dependent on ψ and is adjusted after each direct solving. For stationary flow calculations this last problem is academic as the time evolution and the iteration procedure generally proceed as one and the same process.

The time discretisation transforms the system of ordinary differential equations (4.3) into a system of algebraic difference equations. We will consider the following methods :

a. The fully explicit method (Euler) :

$$\psi^{n+1} = \psi^{n} - \Delta t \, D_n \, \psi^{n} \qquad (4.12)$$

b. The fully implicit method :

$$\psi^{n+1} = \psi^{n} - \Delta t \, D_n \, \psi^{n+1} \qquad (4.13)$$

Here, the time difference $(\psi^{n+1} - \psi^{n})/\Delta t$ is exclusively based on values on the new time level.

c. The Crank-Nicolson formula :

$$\psi^{n+1} = \psi^{n} - {}^{1}/_{2}\Delta t \, (D_n \, \psi^{n+1} + D_n \, \psi^{n}) \qquad (4.14)$$

By this formula a central time differencing is obtained.

The three methods can be represented in one formula, which hereafter we will call the general implicit method :

$$\psi^{n+1} = \psi^{n} - \beta \, \Delta t \, D_n \, \psi^{n+1} - (1-\beta) \, \Delta t \, D_n \, \psi^{n} \qquad (4.15)$$

with β a real number between 0 and 1

Methods (4.12) and (4.13) are first order accurate in time, (4.14) (and thus (4.15) for β=0.5) is second order accurate. For hyperbolic problems method (4.12) is unstable, method (4.13) stable, while (4.14) is marginally stable. In (4.15) β must be between 0.5 and 1. for stability.

4.3.2 ADI Methods

In multi-dimensional problems, implicit methods lead generally to very large systems of equations. To reduce the amount of computational work the Alternating Direction Implicit or ADI methods were developed. In an ADI method the system of equations is split up in a number of smaller systems. Each of these systems typically solves the equations for one directional line of unknowns. All components of the solution vector ψ that are not on this line are taken explicitly that is to say on the old time level. In most applications of the method, only three unknowns per equation

are taken implicitly, which leads to 3-diagonal systems that are easily solved by a direct method (Thomas algorithm). Incorporating all terms of the Navier-Stokes equations in the ADI method would result in 5-diagonal systems that in principle could be solved in a similar manner.

ADI was first introduced by Peaceman and Rachford (simultaneously with Douglas, 1955) as a solution technique for the 2-D diffusion equation. The method consists of two successive time steps. In the first step the x-differences are taken on the new time level and the y-differences on the old. In the second step the roles of x and y are reversed : the y-differences are taken implicitly and the x-differences explicitly. Generally the two time steps are regarded as two half steps so that the whole scheme fits in one time step :

$$\psi^{n+1/2} = \psi^{n} - {}^{1}/_{2}\Delta t\, D_x \psi^{n+1/2} - {}^{1}/_{2}\Delta t\, D_y \psi^{n}$$
$$\psi^{n+1} = \psi^{n+1/2} - {}^{1}/_{2}\Delta t\, D_x \psi^{n+1/2} - {}^{1}/_{2}\Delta t\, D_y \psi^{n+1} \qquad (4.16)$$

where D_x and D_y are the parts of D_n containing the x-differences and y-differences, respectively.

By eliminating the intermediate value $\psi^{n+1/2}$ and a rearrangement of terms we can compare (4.16) with the Crank-Nicolson formula (4.14) :

Peaceman-Rachford :

$$(1 + {}^{1}/_{2}\Delta t\, D_x)(1 + {}^{1}/_{2}\Delta t\, D_y)\, \psi^{n+1} = (1 - {}^{1}/_{2}\Delta t\, D_x)(1 - {}^{1}/_{2}\Delta t\, D_y).\psi^{n} \qquad (4.17)$$

Crank-Nicolson :

$$(1 + {}^1/_2\Delta t\, D_x + {}^1/_2\Delta t\, D_y)\, \psi^{n+1} = (1 - {}^1/_2\Delta t\, D_x - {}^1/_2\Delta t\, D_y)\, \psi^n \qquad (4.18)$$

Comparison of (4.17) with (4.18) shows that the ADI scheme is obtained by factorisation of the expressions within the brackets in the Crank-Nicolson formula. Subtracting (4.17) from (4.18) gives the difference, being the error due to the ADI formulation :

$${}^1/_4\Delta t^2\, D_x\, D_y\, (\psi^{n+1} - \psi^n)$$

Because $(\psi^{n+1} - \psi^n)$ is of order Δt, this shows that the Peaceman-Rachford scheme is a second order accurate approximation of the Crank-Nicholson formula and thus also a second-order approximation of the differential equation (4.3).

In its original form the Peaceman-Rachford scheme is not extendible to 3-D, because it is based on two (half) time steps corresponding to two spatial directions.

In 1956 Douglas and Rachford published a method that can be used for the solution of the 3-D diffusion equation. This scheme is based on a rather different philosophy. Instead of being set up as two successive time steps the Douglas-Rachford procedure consists of a repeated application of the full time step. For each spatial direction such an iteration is executed, each to be regarded as a correction on the preceding one. In the original notation of the 1956 paper the correctional character of the method can be recognized :

$$\psi^{*} = \psi^{n} - \Delta t\ (D_x\psi^{*} + D_y\psi^{n} + D_z\psi^{n}) \qquad (4.19a)$$

$$\psi^{**} = \psi^{*} - \Delta t\ (D_y\psi^{**} - D_y\psi^{n}) \qquad (4.19b)$$

$$\psi^{n+1} = \psi^{**} - \Delta t\ (D_z\psi^{n+1} - D_z\psi^{n}) \qquad (4.19c)$$

Thus, (4.19b) corrects for the fact that in (4.19a) the y-differences were taken explicitly while (4.19c) does the same for the z-differences.

A better known notation for the same scheme is :

$$\begin{aligned} \psi^{*} &= \psi^{n} - \Delta t\ (D_x\psi^{*} + D_y\psi^{n} + D_z\psi^{n}) \\ \psi^{**} &= \psi^{n} - \Delta t\ (D_x\psi^{*} + D_y\psi^{**} + D_z\psi^{n}) \\ \psi^{n+1} &= \psi^{n} - \Delta t\ (D_x\psi^{*} + D_y\psi^{**} + D_z\psi^{n+1}) \end{aligned} \qquad (4.20)$$

Eliminating the intermediate values ψ^{*} and ψ^{**} and rearranging, we can write (4.20) as :

$$\begin{aligned} &(1 + \Delta t\ D_x)(1 + \Delta t\ D_y)(1 + \Delta t\ D_z)\ \psi^{n+1} = \\ &\quad \psi^{n} - \{\Delta t^2(D_xD_y + D_yD_z + D_xD_z) - \Delta t^3(D_xD_yD_z)\}(\psi^{n+1} - \psi^{n}) \end{aligned} \qquad (4.21)$$

From this it is evident that the Douglas-Rachford scheme is a second-order accurate approximation of the fully implicit scheme (4.13), but as the fully implicit method (4.13) is only first order accurate the same will apply to (4.21).

An interesting combination of the ideas of Peaceman-Rachford and Douglas-Rachford is due to Brian (1961). His scheme consists of the execution of a fully implicit step according to the Douglas-Rachford scheme, followed by an

explicit step. The complete scheme thus becomes again second order accurate in time like the Peaceman-Rachford scheme but now in 3-D.

All ADI schemes mentioned above are summarized by the general formulation of Douglas and Gunn (1964). They use the philosophy of the Douglas-Rachford scheme but instead of the fully implicit treatment of the "new direction" in each successive correction a weighted average of explicit and implicit terms is used as in (4.15) :

$$\psi^{*} = \psi^{n} - \Delta t\ (\beta D_x \psi^{*} + (1-\beta)\ D_x \psi^{n} + D_y \psi^{n} + D_z \psi^{n})$$

$$\psi^{**} = \psi^{n} - \Delta t\ (\beta\ D_x \psi^{*} + (1-\beta)\ D_x \psi^{n} + \beta\ D_y \psi^{**} + (1-\beta)\ D_y \psi^{n} + D_z \psi^{n}) \qquad (4.22)$$

$$\psi^{n+1} = \psi^{n} - \Delta t\ (\beta\ D_x \psi^{*} + (1-\beta)\ D_x \psi^{n} + \beta\ D_y \psi^{**} + (1-\beta)\ D_y \psi^{n} + \beta\ D_z \psi^{n+1} + (1-\beta)\ D_z \psi^{n})$$

with β between 0. and 1.

Remark :

The original Peaceman-Rachford formulation can be expressed by a 2-D version of the Douglas-Gunn formula by substituting the intermediate value $\psi^{n+1/2}$ in the PR formula (4.16) by (Vreugdenhil,1989) :

$$\psi^{n+1/2} = \beta\ \psi^{*} + (1-\beta)\ \psi^{n} \qquad \text{with } \beta = 0.5$$

Beam and Warming (1978, see also Briley and McDonald, 1980) give a more elegant notation of (4.22), the so-called delta-formulation :

$$
\begin{aligned}
&(1 + \beta \Delta t \, D_x)(\psi^* - \psi^n) = - \Delta t(D_x\psi^n + D_y\psi^n + D_z\psi^n) \\
&(1 + \beta \Delta t \, D_y)(\psi^{**} - \psi^n) = \psi^* - \psi^n \qquad (4.23) \\
&(1 + \beta \Delta t \, D_z)(\psi^{n+1} - \psi^n) = \psi^{**} - \psi^n
\end{aligned}
$$

This notation makes it clear that in an actual computation according to the Douglas-Gunn formula there is no need to store the intermediate time levels ψ^* and ψ^{**}. All relevant information can be kept in two arrays of variables, one for the old time level ψ^n and one for the successive differences $(\psi^* - \psi^n)$, $(\psi^{**} - \psi^n)$ and $(\psi^{n+1} - \psi^n)$. The new values are obtained by adding the last difference to the values of the old time level.

Because we are interested in the stationary solution the important property of a numerical time integration method is its stability rather than its accuracy. The time integration is used as an iterative procedure towards the stationary solution. If for instance we use a very large time step in an implicit method, rendering a crude time approximation, this may (and in fact often does) slow down the convergence, but the true solution of the stationary equations will in principle be reached independent of the inaccuracy of the time discretisation. This idea is expressed by the delta-formulation (4.23) which after elimination of the intermediate increments $(\psi^* - \psi^n)$ and $(\psi^{**} - \psi^n)$ reads :

$$
(1+\Delta t D_x)(1+\Delta t D_y)(1+\Delta t D_z)(\psi^{n+1} - \psi^n) = - \Delta t D \, \psi^n \qquad (4.24)
$$

If this procedure converges to a stationary solution ψ, this solution will satisfy :

$$D\ \psi = 0$$

independent of Δt.

4.4 ADI Schemes for the AC Navier-Stokes Equations

4.4.1 Introduction

In this section we give the reasons which led to the choice of the computational system. This choice has to do with the way the convective terms are incorporated in the total scheme. In dynamical computations an important problem concerning these terms is their non-linearity, but as we are considering stationary flow this is not a great issue in our application. However, there is another difficulty which complicates the extension of the scheme to three dimensions :

The compressible Navier-Stokes equations are a combination of the three basic problems in the theory of partial differential equations, i.e.:

I. the heat equation : $\partial\phi/\partial t = \nu\ \nabla^2\phi$

II. the transport equation : $\partial\phi/\partial t = -\ (\mathbf{V}.\nabla)\phi$

III. the wave equation : $\partial^2\phi/\partial t^2 = c^2\ \nabla^2\phi$

Problem I and II are often combined and denoted as the convection-diffusion equation. This equation arises when the pressure gradient in the Navier-Stokes equation (2.1) is

omitted. If on the other hand the convection-diffusion terms are omitted we get, in combination with (2.2), a form of the wave equation III.

As will be shown in chapter 5 there is a fundamental problem, connected with the transport problem II and thus with the convective terms in the N.-S. equations, in the extension of the ADI method to 3-D. It can be proven that, for the transport problem, the stability properties of the ADI scheme in 3-D differ from those in 2-D : while the scheme is unconditionally stable in 2-D (for $^1/_2 < \beta < 1$) this is not so in 3-D. If the convective terms are treated outside the ADI process, the latter regains its 2-D stability properties.

Another reason to treat both the convective terms and the diffusion terms separately from the ADI process is the then possible reduction in number of equations to be solved as well as in memory space needed for the field variables.

In the following sections three possible options to incorporate the convection-diffusion (or CD-) terms are given :

a: All terms included in the ADI process,

b: Explicit integration of the CD-terms, completely separated from the ADI process, (Alternative I)

c: Explicit integration of the CD-terms, integrated in the ADI process (Alternative II).

4.4.2 ADI Scheme with Convection-Diffusion Terms Included

Application of the Douglas-Gunn formula (4.22) to the AC Navier-Stokes equations results in the following computational scheme :

x-direction:

$$u^* = u^n - \beta(CD_x u^* + G_x h^*) - (1-\beta)(CD_x u^n + G_x h^n) - CD_y u^n - CD_z u^n$$

$$v^* = v^n - \beta(CD_x v^*) - (1-\beta)(CD_x v^n) - CD_y v^n - G_y h^n - CD_z v^n$$

$$w^* = w^n - \beta(CD_x w^*) - (1-\beta)(CD_x w^n) - CD_y w^n - CD_z w^n - G_z h^n \qquad (4.25a)$$

$$h^* = h^n - {}^1/\alpha\{\ \beta(D_x u^*) + (1-\beta)(D_x u^n) + D_y v^n + D_z w^n\}$$

y-direction:

$$u^{**} = u^n - \beta(CD_x u^* + G_x h^* + CD_y u^{**}) - \\ -(1-\beta)(CD_x + G_x h^n + CD_y u^n) - CD_z u^n$$

$$v^{**} = v^n - \beta(CD_x v^* + CD_y v^{**} + G_y h^{**}) - \\ -(1-\beta)(CD_x v^n + CD_y v^n + G_y h^n) - CD_z v^n$$

$$w^{**} = w^n - \beta(CD_x w^* + CD_y w^{**}) - \\ -(1-\beta)(CD_x w^n + CD_y w^n) - CD_z w^n - G_z h^n \qquad (4.25b)$$

$$h^{**} = h^n - {}^1/\alpha\{\ \beta(D_x u^* + D_y v^{**}) + (1-\beta)(D_x u^n + D_y v^n) + D_z w^n\}$$

z-direction:

$$u^{n+1} = u^n - \beta(\ CD_x u^* + G_x h^* + CD_y u^{**} + CD_z u^{n+1}) - \\ -(1-\beta)(\ CD_x u^n + G_x h^n + CD_y u^n + CD_z u^n)$$

$$v^{n+1} = v^n - \beta(\ CD_x v^* + CD_y v^{**} + G_y h^{**} + CD_z v^{n+1}) - \\ -(1-\beta)(\ CD_x v^n + CD_y v^n + G_y h^n + CD_z v^n)$$

$$w^{n+1} = w^n - \beta(CD_x w^* + CD_y w^{**} + CD_z w^{n+1} + G_z h^{n+1}) - \\ -(1-\beta)(\ CD_x w^n + CD_y w^n + CD_z w^n + G_z h^n) \qquad (4.25c)$$

$$h^{n+1} = h^n - {}^1/\alpha\{\ \beta(D_x u^* + D_y v^{**} + D_z w^{n+1}) + (1-\beta)(D_x u^n + D_y v^n + D_z w^n)\}$$

where CD_x = central difference approx. of

$$\Delta t(U\partial/\partial x - \nu\partial^2/\partial x^2)$$

G_x = central diff. approx. of g $\Delta t(\partial/\partial x)$

D_x = central diff. approx. of $\Delta t(\partial/\partial x)$

and likewise in the y- and z-directions

From (4.25) it is clear that all components of the vector $\phi=(u,v,w,h)^T$ are updated in each directional step of the ADI process. Because all three values of ϕ^*, ϕ^{**} and ϕ^{n+1} are needed in the w-step, three arrays of field-variables must be kept in storage during the computation. By using the delta formulation storage can be reduced to two arrays.
A technical possibility exists to employ a buffering technique to reduce storage to basically one array of field-variables, but this results in a rather complicated computational scheme.

For a 3-D computation even a single array of field-variables takes a great amount of memory space, so it is advantageous to use a scheme which needs only one of such arrays, especially if one wants to use the method on minicomputers. To reach this goal we propose some modifications to the scheme (4.25).

4.4.3 Explicit Convection-Diffusion Terms (Alternative I)

In this section we split the time-integration procedure into two separate steps, somewhat in the manner of a fractional step method. In the first step we advance the solution with the convection-diffusion terms only, using an explicit method. In this step all velocity values are updated by direct replacement.

Remark :

One "old" line and (in 3-D) one "old" plane are kept in memory to make a straight Euler method possible. If the velocity values are replaced without any saving of "old" values the resulting procedure resembles the angled derivative method of Robert and Weiss (1966, see also Stelling,1984). This method has better stability properties than the explicit method but only if the direction of the computational sequence is equal to that of the convecting velocity.

The second step consists of an ADI method for the pressure terms. Rather than the Douglas-Gunn method we use the Douglas-Rachford (β=1) variant of ADI. As a result the scheme (4.25) reduces to:

x-direction:

$$u^{*} = u' - G_x h^{*} \qquad (4.26a)$$

$$v^{*} = v' - G_y h^{n} \qquad (4.26b)$$

$$w^{*} = w' - G_z h^{n} \qquad (4.26c)$$

$$h^{*} = h^{n} - {}^{1}/\alpha(D_x u^{*} + D_y v' + D_z w') \qquad (4.26d)$$

y-direction:

$$u^{**} = u' - G_x h^{*} \qquad (4.26e)$$

$$v^{**} = v' - G_y h^{**} \qquad (4.26f)$$

$$w^{**} = w' - G_z h^{n} \qquad (4.26g)$$

$$h^{**} = h^{n} - {}^{1}/\alpha(D_x u^{*} + D_y v^{**} + D_z w') \qquad (4.26h)$$

z-direction:

$$u^{n+1} = u' - G_x h^{*} \tag{4.26i}$$

$$v^{n+1} = v' - G_y h^{**} \tag{4.26j}$$

$$w^{n+1} = w' - G_z h^{n+1} \tag{4.26k}$$

$$h^{n+1} = h^{n} - {}^{1}/\alpha\,(D_x u^{*} + D_y v^{**} + D_z w^{n+1}) \tag{4.26l}$$

where u',v' and w' are the already updated velocities from the explicit convection-diffusion procedure as in system (4.27).

Inspection of this scheme shows that (4.26e,i) and (4.26j) are just repetitions of (4.26a) and (4.26f) respectively, while the values on the lefthand side of (4.26b,c and g) are not used. Thus the same result can be achieved using only 6 instead of 12 equations. The intermediate values h^{*} and h^{**} appear only within the x- and y-step, respectively and do not have to be saved.

Thus, the above mentioned modifications mean an important reduction both in computing effort and memory capacity needed for a calculation. The fact that the accuracy of the scheme is only first order in time is not important as we are concerned with stationary problems. While the first step still is only conditionally stable, the second step is now unconditionally stable as will be shown in section 5.4

Using an abbreviated notation the resulting scheme can be given as:

$$\begin{aligned} u' &= u^{n} - CD(u^{n}) \\ v' &= v^{n} - CD(v^{n}) \\ w' &= w^{n} - CD(w^{n}) \end{aligned} \tag{4.27a}$$

$$
\begin{aligned}
u^{n+1} &= \mathrm{ADI}(u',h) \\
v^{n+1} &= \mathrm{ADI}(v',h) \\
w^{n+1} &= \mathrm{ADI}(w',h) \\
h^{n+1} &= \mathrm{ADI}(u^{n+1},v^{n+1},w^{n+1})
\end{aligned}
\qquad (4.27b)
$$

where CD again stands for the convection-diffusion operation and ADI for the ADI-procedure as given in (4.26)

4.4.4 Explicit CD-Steps Integrated in ADI Scheme (Alternative II)

Because the ADI procedure consists of three directional steps, another order of operations is also possible. Each directional ADI step can be combined with the explicit convection-diffusion operation of the corresponding velocity component. This results in:

$$
\begin{aligned}
u' &= u^n - \mathrm{CD}(u^n) \\
u^{n+1} &= \mathrm{ADI}(u',h) \\
\\
v' &= v^n - \mathrm{CD}(v^n) \\
v^{n+1} &= \mathrm{ADI}(v',h) \\
\\
w' &= w^n - \mathrm{CD}(w^n) \\
w^{n+1} &= \mathrm{ADI}(w',h) \\
h^{n+1} &= \mathrm{ADI}(u^{n+1},v^{n+1},w^{n+1})
\end{aligned}
\qquad (4.28)
$$

We will show in chapter 6 that scheme (4.28) has better stability properties than (4.27). Accordingly (4.28) is adopted as the final scheme we will use. We conclude this

chapter by writing out the last scheme in a more extensive way, using the same notation as in (4.25) :

x-direction:

$$u' = u^n - CD_x u^n - CD_y u^n - CD_z u^n$$

$$u^{n+1} = u' - G_x h^*$$

$$h^* = h^n - 1/\alpha(D_x u^{n+1} + D_y v' + D_z w')$$

y-direction:

$$v' = v^n - CD_x v^n - CD_y v^n - CD_z v^n$$

$$v^{n+1} = v' - G_y h^{**} \qquad (4.29)$$

$$h^{**} = h^n - 1/\alpha(D_x u^{n+1} + D_y v^{n+1} + D_z w')$$

z-direction:

$$w' = w^n - CD_x w^n - CD_y w^n - CD_z w^n$$

$$w^{n+1} = w' - G_z h^{n+1}$$

$$h^{n+1} = h^n - 1/\alpha(D_x u^{n+1} + D_y v^{n+1} + D_z w^{n+1})$$

Remark:

The first two equations of each directional step can of course be combined into one operation. Each of the resulting operations then is a consistent approximation of the corresponding component of the original AC equations (4.1). It has however advantages to keep the explicit and ADI steps separated. Apart from a more modular structure in programming, it is also possible to use a different time step for the explicit and the ADI parts of the computation.

4.5 Influence of the Compressibility Factor α

In section 3.4 the still water propagation of the artificial compression wave was given as :

$$c_s = \sqrt{(g/\alpha)}$$

Here we will compute the propagation if the convecting velocity differs from zero. We start with the Fourier transform of the AC equations (4.1). Equating the determinant of the coefficients to zero yields the characteristic equation :

$$(c - V_n)^2[(c - V_n) c - g/\alpha] = 0 \qquad (4.30)$$

where c = propagation speed
and V_n = component of convecting velocity parallel to $\mathbf{k}$ (or : $V_n = \mathbf{V_c}.\mathbf{k}$)

Equation (4.30) has four solutions :

A double solution $c = V_n$, representing the transport of the transversal modes of the flow in the two possible independent directions perpendicular to the wave number vector $\mathbf{k}$.

Two solutions representing the forward and backward propagated waves connected with the longitudinal flow modes :

Forward wave : $c = {}^1/_2 V_n + \sqrt{({}^1/_4 V_n{}^2 + g/\alpha)}$

Backward wave : $c = {}^1/_2 V_n - \sqrt{({}^1/_4 V_n{}^2 + g/\alpha)}$

Note that the propagation velocity always exceeds the convecting velocity. This means that in the artificial

compressibility method the flow stays "subsonic". However, for small values of g/α the backward or upstream propagation may become extremely slow.

According to Kwak and Chang (1984) the value of g/α should be chosen such that the time it takes for the pressure wave to reestablish the condition of incompressibility :

$$\nabla . \mathbf{V} = 0$$

is small compared to the time of the spreading of the viscous effects over the computational domain. The time scale of the spreading of turbulent diffusion can be estimated from the rate of growth of the boundary layer along a flat plate :

$$\delta = 2 \sqrt{(\nu_t t)}$$

where δ = thickness of bound. layer

and t = time

As an example we take a computational domain with a length of 100 m and a depth of 10 m. The turbulent viscosity is chosen as 0.1 m^2/sec. Taking for δ the depth of the channel we find for the time needed for the influence of the bottom friction to diffuse to the surface :

$$t_d = 250 \text{ sec}$$

The time needed for the upstream pressure wave to travel the length of the channel is :

$$t_p = \frac{100}{c} \text{ sec}$$

$$\text{with } c = {}^1/_2 V_c - \sqrt{(V_c^2/4 + g/\alpha)}$$

For a convecting velocity of 1 m/sec and values of the compressibility factor α less than 1 $m.^{-1}$, t_p will be about one order smaller than t_d.

On the other hand we cannot take value of α too small (or g/α too large). The reason for this is that with a large value of g/α the system of equations that has to be solved becomes "stiff", i.e. the time scale of the pressure wave is much smaller than the time scale of the convection and diffusion. This causes problems with the ADI solution method.

As an example, consider the scheme (4.28). We assume that the solution at $t=n\Delta t$ satisfies continuity :

$$\nabla.(u^n, v^n, w^n)^T = 0 \qquad (4.31)$$

In the explicit u-step the velocity component u^n is changed into u' by the CD-terms, so the intermediate solution :

$$(u', v^n, w^n)^T$$

in general will not satisfy the continuity equation.

If the value of g/α in the subsequent ADI-x step is large, the resulting velocity component u^{n+1} will have such a value that the flow is again more or less divergence free, because a small deviation of $\nabla.V$ from zero will cause large pressure differences, forcing the u-component towards $\nabla.V \approx 0$. Thus, after the ADI-x step we have :

$$\nabla.(u^{n+1}, v^n w^n)^T \approx 0 \qquad (4.32)$$

Comparing (4.32) with (4.31) we see that apparently :

$$u^{n+1} \approx u^{n}$$

(assuming steady boundary conditions)

The effect of the explicit CD step is more or less nullified by the subsequent ADI step and the influence of the CD terms per time step is very limited.

In the scheme according to (4.27) a similar behaviour is found. Here, the effects of the three directional explicit CD steps are, for large g/a, almost completely resolved in the subsequent ADI-x step. Thus, after the first ADI-x step, continuity is more or less satisfied :

$$\nabla.(u^{n+1},v',w') \approx 0$$

In the following ADI-y and ADI-z steps the v' and w' velocity components will hardly be modified. Again the pressure distribution, caused by the CD-terms, does not "survive" the first ADI step and thus cannot be used to spread the influence of these terms over the three spatial directions.

Theoretically, alternative II (=scheme (4.28)) will converge to the right solution, independent of the value of g/α, but it will be clear that the rate of convergence will be very low for large g/α.

The final solution in the case of alternative I (=scheme (4.27)) is not independent of the value of g/α. See also expression (6.10b) in section 6.3.

The above mentioned effects become serious if the propagation velocity $\sqrt{(g/a)}$ becomes an order of magnitude larger than the flow velocity itself. In case of a different time step for the convection-diffusion part of the equations and

for the ADI part, we must consider the respective Courant numbers rather than the velocities. In this case the Courant number based on the flow velocity should be of the same order as the Courant number associated with artificial pressure wave propagation.

5. THE STABILITY OF ADI TIME-INTEGRATION SCHEMES

5.1 Introduction

In this chapter we will analyse the numerical stability of ADI schemes. First we will discuss a number of different methods used to ascertain the stability of a time integration scheme. Next we will use the most commonly used method, the von Neumann stability analysis, to show that stability in 2D does not automatically ensure stability in 3D. In particular we will show that the Douglas-Gunn ADI procedure is not unconditionally stable for the 3D transport equation.

In section 5.6 the stability of the ADI method for the time integration of the discrete wave equation is analysed. It is shown that in this case there is no fundamental difference in stability properties between the 2-D and 3-D application of the scheme.

Finally it is concluded that, due to the convective terms, the application of ADI to the 3-D Navier-Stokes equations will result in a process that is only conditionally stable. Two Courant numbers can be defined : one connected to the wave propagation and one connected to the convective velocity. By splitting the total process in a "pressure wave" part and a "convection-diffusion" part, we have the possibility to apply the ADI integration to the wave part only, giving this part an unconditional stability. Thus, the stability of the total scheme is determined by the Courant number based on the convecting velocity. This velocity is generally much lower than the wave propagation speed, admitting a larger time step for the total scheme.

5.2 Methods to Analyse Stability

A general notion of the stability of a numerical time integration process is given by the demand that the eigenvalues of G in the representation of the system of difference equations:

$$\psi^{n+1} = G\,\psi^{n} \tag{5.1}$$

should not exceed unity in absolute value. The most direct method of investigation would be the calculation of these eigenvalues but in practice this is complicated by the fact that G is often a very large matrix and, unless the computational domain is very simple, irregular. Still, this method, known as the matrix method, can be used although generally the eigenvalues can only be estimated. An advantage of this method is that the boundaries are taken into consideration. See for example Praagman (1979).

Another possibility is to consider the Fourier transform of (5.1). The amplification matrix G then becomes manageable and the eigenvalues can be computed in many cases. The basis for methods of this class is the well known von Neumann stability analysis. Again we demand that the eigenvalues of G do not exceed unity in absolute value. Theoretically the method is restricted to simple boundary conditions and linear problems, so that the Fourier decomposition is valid. Several authors have developed more refined stability conditions based on the same principle (see Richtmyer and Morton,1967). For our purposes the original von Neumann method is considered adequate.

For the sake of completeness we will mention two methods that are based on other principles than the amplification

matrix, viz. the energy method (Friedrichs,1954) and Hirt's stability analysis (Hirt,1968).

In the energy method one chooses a suitable quantity which represents the value of the solution vector ψ. Usually this is some norm of ψ. In some problems this quantity represents an amount of energy, hence the name. Stability is ensured if this quantity does not grow in time which in some cases can be investigated by direct calculation.

Hirt's method is related to the modified equation approach (Warming and Hyett,1974) which is usually used to check the consistency of the difference equations. Each term of the difference equation is expanded in a Taylor series around the central point. If the numerical approximation is consistent the result of this operation will, at the limit, restate the original differential equation for Δt, $\Delta x \rightarrow 0$.

Used as a stability analysis this method groups the truncation error terms in odd- and even-order differential terms. The even order terms are related to the absolute value of the amplification, while the odd terms influence the propagation speed. For stability the even order terms should have such a sign that they act in the same direction as the viscosity (itself a second-order term).

5.3 The Stability of the 1-D Convection-Diffusion Equation

As an example of the von Neumann method we will derive the stability condition for the explicit in time, central in space, difference approximation (FTCS-method) of the 1D convective-diffusive transport equation:

$$\frac{\partial \psi}{\partial t} + U \frac{\partial \psi}{\partial x} - \nu \frac{\partial^2 \psi}{\partial x^2} = 0 \qquad (5.2)$$

The corresponding difference equation reads:

$$\psi_i^{n+1} - \psi_i^n =$$

$$- \frac{U\Delta t}{2\Delta x} (\psi_{i+1}^n - \psi_{i-1}^n) + \frac{\nu\Delta t}{\Delta x^2} (\psi_{i+1}^n + \psi_{i-1}^n - 2\psi_i^n) \qquad (5.3)$$

Insertion of

$$\psi_j^m = \hat{\psi}^m \exp(i\ k\ j\ \Delta x) \qquad m=n,n+1 \quad j=i-1,i,i+1$$
$$i=\sqrt{-1} \qquad k=\text{wavenumber}$$

and dividing by the common factor results in the following expression for the complex amplitude :

$$\hat{\psi}^{n+1} = [\ 1 - i \frac{U\Delta t}{\Delta x}\sin(k\ \Delta x) - \frac{2\nu\Delta t}{\Delta x^2}(1-\cos(k\ \Delta x))]\ \hat{\psi}^n \qquad (5.4)$$

The quantity between the square brackets forms the scalar amplification factor. Due to the second term it is a complex number which can be depicted in the complex plane as a function of the wave number k (fig 5.1). For $-\pi < k\ \Delta x < \pi$, G describes an ellipse in the complex plane with axes :

horizontal (real) axis :

$$(\ 1 - 2\nu\Delta t/\Delta x^2\ ) \pm 2\nu\Delta t/\Delta x^2 \qquad (5.5)$$

vertical (imaginary) axis :

$$(\ 1 - 2\nu\Delta t/\Delta x^2\ ) \pm i\ U\Delta t/\Delta x$$

For stability this ellipse should lie entirely within the unit circle.

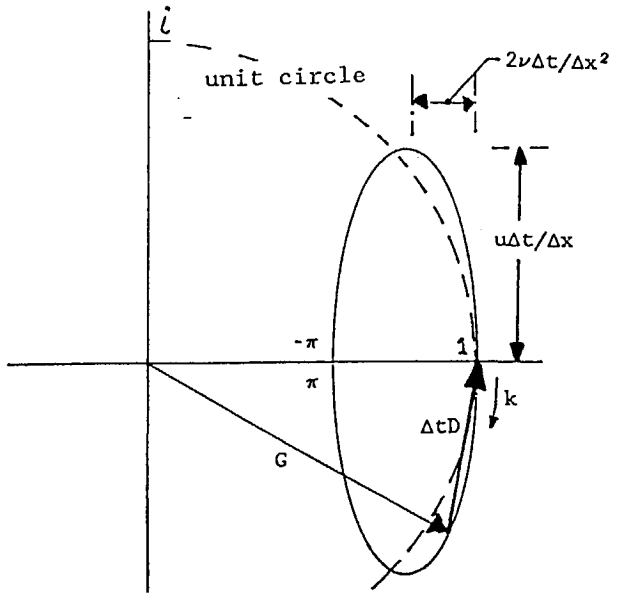


fig. 5.1 explicit scheme

From (5.5) two necessary, but not sufficient, conditions can be deduced:

1. If the Courant number $U\Delta t/\Delta x > 1$, the process is unstable for every value of the viscosity coefficient ν.
2. If the diffusion number $\nu\Delta t/\Delta x^2 > {}^1/_2$ the process will be unstable for every value of the convecting velocity U.

Graphically, these two conditions are represented by the vertical and horizontal axis of the above mentioned ellipse. The ratio between these axes forms the cell Reynolds number :

$$Re_c = U\Delta x/\nu$$

For $Re_c > 2$ the vertical axis will exceed the horizontal axis in length. We then consider the numerical problem convection dominated.

In practical 3-D applications we shall always try to use large grid-sizes in order to restrict the amount of necessary memory space. Moreover, because our interest is in low viscosity fluid, the viscosity coefficient will have a small value. Therefore, in our applications, the problem will generally be convection dominated.

Geometrically this means that, for a Courant number less than unity, the horizontal axis of the afore-mentioned ellipse will always be situated within the unit circle. Under this circumstance the ellipse will lie completely within the unit circle only if the curvature in point (1,0) is less than unity (fig. 5.1). The curvature in this point is given by c^2/a, where a and c are the horizontal and vertical semi-axes of the ellipse, respectively. Because $c = U\Delta t/\Delta x$ and $a = 2\nu\Delta t/\Delta x^2$, this leads to a stability condition :

$$U^2\Delta t/2\nu < 1 \quad \text{or} \quad \Delta t < 2\nu/U^2 \tag{5.6}$$

In most practical applications this restriction is much more severe than the Courant condition.

5.4 Extension to 3-D

The 3-D formulation of the convection-diffusion equation reads :

$$\partial\psi/\partial t + (\mathbf{V}.\nabla)\,\psi - \nu\nabla^2\psi = 0 \tag{5.7}$$

with $\mathbf{V} = (u,v,w)^T$ = the (constant) convecting velocity

Although the transported quantity may be a vector, the amplification can still be expressed as a complex scalar number because the three component equations of (5.7) are independent. In a similar manner as in section 5.3 an amplification factor for the FTCS-method can be derived :

$$G = 1 - i\,\Delta t\,\frac{u\,\sin(\kappa_x \Delta x)}{\Delta x} - 2\nu\Delta t\,\frac{1-\cos(\kappa_x \Delta x)}{\Delta x^2}$$
$$- i\,\Delta t\,\frac{v\,\sin(\kappa_y \Delta y)}{\Delta y} - 2\nu\Delta t\,\frac{1-\cos(\kappa_y \Delta y)}{\Delta y^2} \qquad (5.8)$$
$$- i\,\Delta t\,\frac{w\,\sin(\kappa_z \Delta z)}{\Delta z} - 2\nu\Delta t\,\frac{1-\cos(\kappa_z \Delta z)}{\Delta z^2}$$

The graphical representation of (5.8) in the complex plane consists in the 3-D case of three ellipses "in series". Just as in the 1-D case the stability for the convection dominated problem depends on the radius of curvature of this graph in the point (1,0). This radius is found by adding the three curvatures of the separate ellipses. Thus, the stability condition for the explicit time integration of the 3-D convection-diffusion equation is given by :

$$(u^2/2\nu + v^2/2\nu + w^2/2\nu)\,\Delta t < 1$$

or:

$$\Delta t < 2\nu/(u^2 + v^2 + w^2) \qquad (5.9)$$

In addition we have the demand that the total of the horizontal axes should not exceed unity or :

$$2\nu\ \Delta t\ (1/\Delta x^2+1/\Delta y^2+1/\Delta z^2) < 1$$

For small ν and reasonable values for Δx, Δy and Δz the problem will be convection dominated and (5.9) will be the determining stability condition. These results can also be found in Hindmarsh, Gresho and Griffiths (1984).

It should be noted that for a general direction of the convecting velocity the condition (5.9) is, in a certain sense, more restrictive than condition (5.6) derived for the 1-D case : For a convecting velocity with equal components :

$$u_c = v_c = w_c = V_c/\sqrt{3}$$

the maximum Δt according to (5.9) is equal to :

$$\Delta t_{crit} = 2\nu/V_c^2$$

In chapter 6 we shall show that the practical value of Δt for the combined explicit-ADI method that we use for the solution of the full 3-D Navier-Stokes equations is in the vicinity of the 1-D criterion (5.6) :

$$\Delta t_{crit} = 2\nu/u^2$$

which is three times as large.

5.5 The Stability of the Douglas-Gunn ADI Scheme for the Convection-Diffusion Equation

5.5.1 General Remarks

As mentioned in section 4.3 the Douglas-Rachford ADI method and its generalization by Douglas-Gunn were introduced by those authors as ADI scheme's that are extendible to 3-D, which the original Peaceman-Rachford scheme is not. Although this claim is correct for the (parabolic) diffusion equation it is not generally true for the (hyperbolic) transport equation. We will show that the convection-diffusion equation, and thus the Navier-Stokes equations as a whole, is not stable in the Douglas-Gunn approximation in 3-D. (See also South,1985)

5.5.2 The Scalar 3-D Convection-Diffusion Equation

We repeat the 3-D convection-diffusion equation as given by (5.7) :

$$\partial\phi/\partial t + (\mathbf{V}.\nabla)\phi - \nu \nabla^2\phi = 0 \qquad (5.7)$$

with $\mathbf{V} = (u,v,w)^T$ = the convecting velocity

For simplicity we will regard ϕ here as a scalar quantity.

The Fourier transform of the spatial central difference approximation results in :

$$d\hat{\psi}/dt + D\,\hat{\psi} = 0 \qquad (5.10)$$

with D =

$$i\,[U \sin(\kappa_x \Delta x)/\Delta x + V \sin(\kappa_y \Delta y)/\Delta y + W \sin(\kappa_z \Delta z)/\Delta z]$$

$$+ \nu \left[\frac{\sin^2(1/2 \kappa_x \Delta x)}{1/4 \Delta x^2} + \frac{\sin^2(1/2 \kappa_y \Delta y)}{1/4 \Delta y^2} + \frac{\sin^2(1/2 \kappa_z \Delta z)}{1/4 \Delta z^2} \right]$$

Before considering ADI-methods we will first analyse the stability of the general implicit method (4.13). In this analysis we will write alternatively:

$$\Delta t\, D = \Delta t(D_x + D_y + D_z)$$

or :

$$\Delta t\, D = a_1 + i\, b_1$$

to indicate either the dimensionality or the complex character of D. Thus, the diffusion term of (5.7) is represented by a_1, the convection term by b_1.

The general implicit method can be written as:

$$[1 + \beta\, \Delta t(D_x + D_y + D_z)]\, \psi^{n+1} = [1 - (1-\beta)\, \Delta t(D_x + D_y + D_z)]\, \psi^n \qquad (5.11)$$

Hence the amplification-factor of this method is represented by :

$$G = \frac{1 - (1-\beta)\, \Delta t(D_x + D_y + D_z)}{1 + \beta\, \Delta t(D_x + D_y + D_z)} \qquad (5.12)$$

Using $\Delta t(D_x + D_y + D_z) = a_1 + i b_1$ we can calculate the squared absolute value of G :

$$|G|^2 = GG^* = 1 - \frac{2a_1 + (2\beta-1)(a_1^2+b_1^2)}{1 + 2\beta a_1 + \beta^2(a_1^2+b_1^2)} \qquad (5.13)$$

From this it follows that for stability it is required that

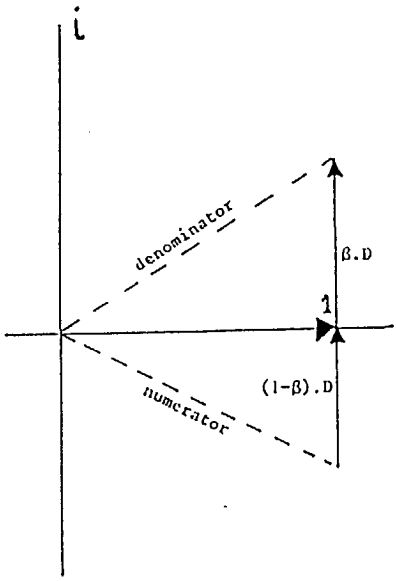
$$\beta > {}^1/_2[\ 1 - 2a_1/(a_1{}^2+b_1{}^2) \qquad (5.14)$$

For $(a_1{}^2+b_1{}^2) \to 0$, we have $a_1 \ll b_1$ and (5.14) is reduced to the well known stability condition of the general implicit method : $\beta > {}^1/_2$. In the viscid case ($a_1>0$) the lowest value of the wavenumber **k** is marginally stable for $\beta = {}^1/_2$, for higher values of **k** the viscosity has a stabilizing influence.

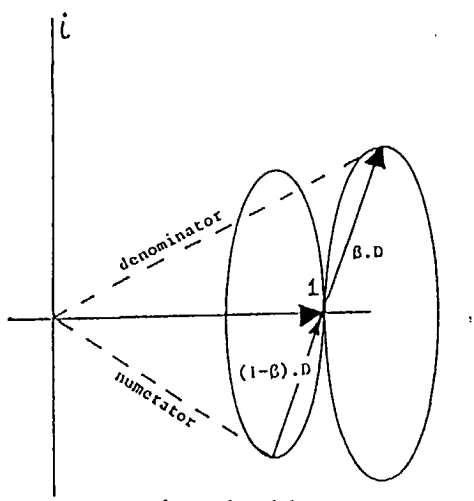
A graphical representation of formula (5.12) is given in fig. 5.2. In this figure the absolute value of the amplification factor G can be obtained as the ratio between the distances from the origin to the extremities of the line ΔtD. In the figure those distances are marked accordingly "numerator" and "denominator".

N.B. While in fig. 5.1 the end point of the vector ΔtD describes a true ellipse, the corresponding graphs in fig. 5.2 are true ellipses only in the case that $u/\Delta x=v/\Delta y=w/\Delta z$ and $\nu/\Delta x^2=\nu/\Delta y^2=\nu/\Delta z^2$. As mentioned in section 5.4 the ellipse-like graphs in this figure are constructed by adding up three separate ellipses, one for every space direction. It follows immediately from geometrical reasoning that for the inviscid case with $\beta={}^1/_2$ the numerator and denominator will have an equal length and thus $|G|=1$. (fig. 5.2a)

Note that the points for which the distance to the extremities of the complex ΔtD are equal will lie on a line normal to and bisecting the length ΔtD. If this normal passes the origin on the positive side (for positive ΔtD), the denominator will exceed the numerator in absolute value. From this observation it can be understood that, while for



a: inviscid



b: viscid

fig. 5.2 General Implicit Method

the higher wave-numbers $|G|$ is less than unity even for values of β smaller than $^1/_2$, for the lower limit of the wavenumber $|k| \to 0$, β must be at least $^1/_2$ for stability. Thus, the same results as found analytically can be deduced from the graphical representation fig. 5.2.

This graphical representation of the stability will be useful in the analysis of the ADI method which now follows. While in the foregoing example the dimensionality of the problem did not create a fundamental difference in the result, we shall see that in case of the ADI-method there is a marked difference between 2-D and 3-D.

The general ADI-formula of Douglass-Gunn (4.22) can be written, after elimination of the intermediate values ψ^* and ψ^{**}, as :

$$(1+\beta\Delta tD_x)(1+\beta\Delta tD_y)(1+\beta\Delta tD_z)\ (\psi^{n+1}-\psi^n) = -\Delta t(D_x+D_y+D_z)\ \psi^n \qquad (5.15)$$

The amplification factor for this scheme is :

$$G = 1 - \frac{\Delta t(D_x+D_y+D_z)}{(1+\beta\Delta tD_x)(1+\beta\Delta tD_y)(1+\beta\Delta tD_z)} \qquad (5.16)$$

We need some abbreviations:

$$\Delta t(D_x+D_y+D_z) = a_1 + ib_1 = a_x+a_y+a_z+i(b_x+b_y+b_z) \qquad (5.17a)$$

$$\Delta t^2(D_xD_y + D_yD_z + D_xD_z) = a_2 + ib_2 =$$

$$(a_x+ib_x)(a_y+ib_y)+(a_y+ib_y)(a_z+ib_z)+ +(a_x+ib_x)(a_z+ib_z)= \qquad (5.17b)$$

$$a_xa_y + a_ya_z + a_za_x - b_xb_y - b_yb_z - b_zb_x + i(a_xb_y + b_xa_y + a_yb_z + b_ya_z + a_zb_x + b_za_x)$$

$$\Delta t^3(D_xD_yD_z) = a_3 + ib_3 =$$

$$=(a_x+ib_x)(a_y+ib_y)(a_z+ib_z)=$$

$$a_xa_ya_z - a_xb_yb_z - b_xa_yb_z - b_xb_ya_z + i(a_xb_ya_z + b_xa_ya_z + a_xa_yb_z - b_xb_yb_z) \qquad (5.17c)$$

Note that, while a_1 and b_1 represent the diffusion term and convection term, respectively, this is not so for a_2, a_3, b_2 and b_3, which are combinations of these terms.

With these abbreviations we can express the amplification G of (5.15) as:

$$G = \frac{1 - (1-\beta)(a_1+ib_1) + \beta^2(a_2+ib_2) + \beta^3(a_3+ib_3)}{1 + \beta(a_1+ib_1) + \beta^2(a_2+ib_2) + \beta^3(a_3+ib_3)} \qquad (5.18)$$

After some calculation we find for the squared absolute value of G :

$$|G|^2 = 1 - \frac{2a_1+(2\beta-1)(a_1^2+b_1^2)+2a_1(\beta^2a_2+\beta^3a_3)+2b_1(\beta^2b_2+\beta^3b_3)}{(1+\beta a_1+\beta^2a_2+\beta^3a_3)^2 + (\beta b_1+\beta^2b_2+\beta^3b_3)^2} \qquad (5.19)$$

which is to be compared to (5.14).

It is convenient to use a further abbreviation :

$$R = \beta^2(a_2+ib_2) + \beta^3(a_3+ib_3) \qquad (5.20)$$

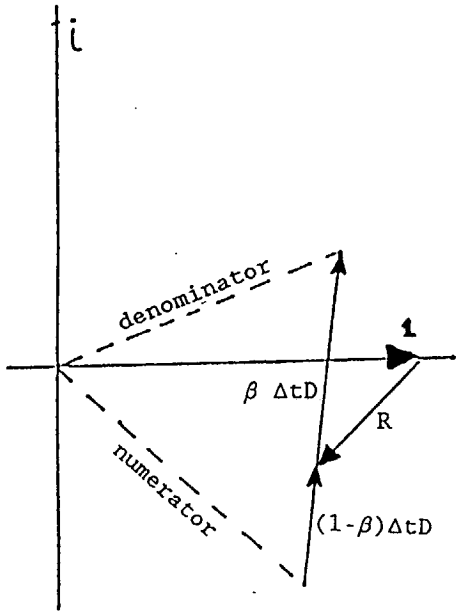
This R represents the terms that originate from the factorisation that is done in the ADI method (see e.g. (4.17) and (4.18)). Because the terms of R contain higher powers of Δt, the value of R goes faster to zero, for $\Delta t \to 0$, than the main terms of formula (5.16) so the ADI method is consistent with the general implicit method and, for $\beta = {}^1/_2$, second order accurate. However, when analysing stability, we consider high wave numbers and a finite value of Δt. In this case the value of R becomes important.

Using (5.20) we can rewrite (5.18) in such a way that it can be readily compared with the expression for the amplification-factor (5.8) of the general implicit method.

$$G = \frac{1 + R - (1-\beta)\,\Delta t(D_x + D_y + D_z)}{1 + R + \beta\,\Delta t(D_x + D_y + D_z)} \qquad (5.21)$$

As can be seen, apart from the extra terms R, the two expressions (5.12) and (5.21) are quite alike.

In fig. 5.3 a graphical representation of (5.21) is given in the same manner as was done for the general implicit method. Because R appears both in the numerator and in the denominator, the complex length ΔtD is shifted over the distance R. From fig. 5.3 the destabilizing influence of R can be understood. Destabilization occurs when the imaginary component of R has a sign opposite to that of the imaginary part of $\Delta t(D_x + D_y + D_z)$.



Destabilizing influence of ADI factorisation terms

fig.5.3

For a low viscosity convection-diffusion 3-D process, the main part of R consists of the term $-i\beta^3(b_x b_y b_z)$, which for high wave number and reasonable Courant number can attain large values, easily exceeding the value of $|\Delta tD|$.

Thus, it can be explained why the 3-D ADI process behaves differently in 3-D as compared to 2-D. In the two dimensional case R consists of only the first term in (5.20). For a small value of the diffusion number this term will contain only a small imaginary component, or none at all in the inviscid case. The real part of R does not endanger the stability.

In three dimensions the inviscid case leads to unconditional instability. For values of $\beta > {}^1/_2$ in the viscid case conditional stability is possible.

5.5.3 Examples

As an illustration of the theory developed in the foregoing section we will give a few examples.

A. 2-D inviscid:

$a_1 = 0$, $a_2 = -b_x b_y$, $b_2 = 0$

As for 2-D $a_3 = b_3 = 0$ we get from (5.20) $R = 0$. This means that the stability criterion of the ADI method is equal to that of the general implicit method (4.15).

B. 2-D viscid:

Now $R = \beta^2(a_x a_y - b_x b_y) + i(a_x b_y + a_y b_x)$, so R may possible have an imaginary component of a sign opposite to that of $(D_x + D_y)$. However, by putting $a_3 = b_3 = 0$ in (5.19), it can be shown that the numerator in the expression under the root is always positive, indicating unconditional stability.

C. 3-D inviscid:

$a_1 = 0$, $a_2 = -b_x b_y - b_y b_z - b_x b_z$, $b_2 = 0$, $a_3 = 0$, $b_3 = -b_x b_y b_z$.

With these values R becomes :

$$R = \beta^2(-b_x b_y - b_y b_z - b_x b_z) - i(\beta^3 b_x b_y b_z)$$

Depending on the sign of b_x, b_y and b_z, R may contain a large imaginary component of a sign opposite that of $b_1 = (D_x + D_y + D_z)$, resulting in unconditional instability for $\beta < {}^1/_2$ as in the 2-D case, but now also for $\beta = {}^1/_2$ (see fig. 5.3a).

For ${}^1/_2 < \beta < 1$ the scheme is conditionally stable. From (5.19) we have :

$$(2\beta - 1)b_1^2 + 2\beta^3 b_1 b_3 > 0$$

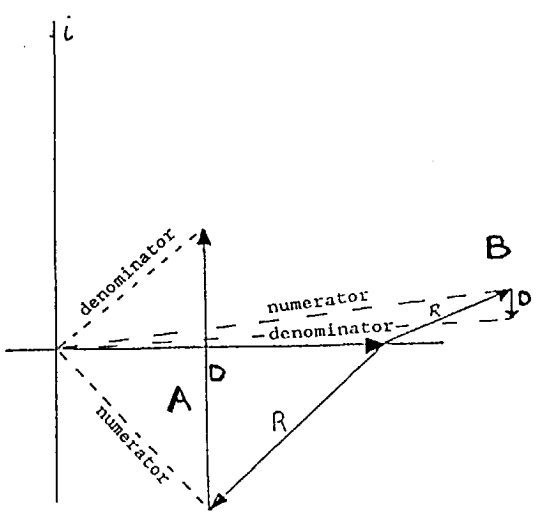
or:

for $b_1>0$: $(2\beta-1)b_1 > -2\beta^3 b_3$

for $b_1<0$: $(2\beta-1)b_1 < -2\beta^3 b_3$

Two different situations must be considered :
In the first situation we take the approximated wave number vector k' parallel to the convecting velocity V (The components of the approximated wave number k' differ from their exact value by the factors $\sin(\kappa_x \Delta x)/(\kappa_x \Delta x)$, etc.). We set $b_x=b_y=b_z=b$,i.e. $b_1= 3b$, $b_3= b^3$, and $u=v=w=|V|/\sqrt{3}$. The result is a stability condition for Δt :

$$\Delta t < \frac{3(2\beta-1)\ \Delta s}{\sqrt{2}\ |V|\ \beta^3} \qquad (5.22)$$



situation A: k'//V situation B: k'$\perp$ V

fig. 5.4 Douglas-Rachford ADI

The second situation arises when **k'** is almost perpendicular to **V**. In this case b_3 has a finite value for b_1 infinitely small, resulting in unconditional instability. Therefore, we must conclude that the ADI formulation of the 3-D inviscid convection equation is unconditional unstable.

The same results can be obtained graphically from fig. 5.4. For simplicity β=1 in this figure.

However, further analysis of the three dimensional function $(2\beta-1)b_1=-2\beta^3 b_3$ in the b_x, b_y, b_z-space reveals that the regions of instability connected to the second situation described above are very narrow near the origin. Consequently these regions will be strongly affected and in fact disappear if a viscosity term iadded to the equation.

D. 3-D viscid:

For the viscid 3-D case, which of course is the one we are primarily interested in, it is difficult to derive a stability condition, but it can be shown by direct calculation of G from (5.19) or (5.21) for the critical directions that the damping effect of the viscosity in the ADI method tends to be less than in the general implicit method. Thus a practical estimation of the stability condition will be the one that is derived in the inviscid case for the situation that **k'** is parallel to **V** :

$$\Delta t < \frac{3\ (2\beta-1)\ \Delta s}{\sqrt{2}\ |V|\ \beta^3} \qquad (5.22)$$

5.6 Stability of ADI for the Wave Equation

In this section we will apply the foregoing results to the 3-D wave equation:

$$\partial \mathbf{V}/\partial t + g\nabla h = 0$$

$$\alpha \partial p/\partial t + \nabla . \mathbf{V} = 0 \qquad (5.23)$$

Because we intend to use, in the final solution procedure, the fully implicit Douglas-Rachford version of ADI we choose $\beta = 1$. Because (5.23) is a vector equation the factors :

$$(1+\Delta t D_x) \ , \ (1+\Delta t D_y) \text{ and } (1+\Delta t D_z)$$

in the Douglas-Gunn formula (5.15) (with β=1) become in the Fourier transform of (5.23) the matrices :

$$\begin{bmatrix} 1 & 0 & 0 & ig\xi \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ i\xi/\alpha & 0 & 0 & 1 \end{bmatrix} , \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & ig\eta \\ 0 & 0 & 1 & 0 \\ 0 & i\eta/\alpha & 0 & 1 \end{bmatrix} , \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & ig\zeta \\ 0 & 0 & i\zeta/\alpha & 1 \end{bmatrix} \qquad (5.24)$$

with $\xi = \Delta t \ \sin(^1/_2 \kappa_x \Delta x)/(^1/_2 \Delta x)$

$\eta = \Delta t \ \sin(^1/_2 \kappa_y \Delta y)/(^1/_2 \Delta y)$

$\zeta = \Delta t \ \sin(^1/_2 \kappa_z \Delta z)/(^1/_2 \Delta z)$

We have seen in section 5.5 that the main reason of the instability of the Douglas-Gunn formula originates from the term $\Delta t^3 D_x D_y D_z$. Direct multiplication of the corresponding matrices (i.e. the matrices (5.24) with zeros on the diagonal) for the problem (5.23) shows this product to be equal to the zero matrix. Thus, we do not expect stability problems for ADI time integration of the wave equation (5.23). Analysis as was done for the scalar problem in sections 5.3 and 5.4 is not so easy, but it is possible to determine the eigenvalues of the amplification matrix. This is done in the following way :

First we write the Fourier transform of the inviscid wave equation (5.23) as:

$$A\ (\hat{\psi}^{n+1} - \hat{\psi}^{n}) = -\Delta tD\ \hat{\psi}^{n} \qquad (5.25)$$

$$\text{with } \hat{\psi} = (\hat{u},\hat{v},\hat{w},\hat{p})^{T}.$$

From (5.15) we see that A is the product of the matrices (5.24) :

$$A = \begin{bmatrix} 1 & -g/\alpha.\xi\eta & -g/\alpha.\xi\zeta & ig\xi \\ 0 & 1 & -g/\alpha.\eta\zeta & ig\eta \\ 0 & 0 & 1 & ig\zeta \\ i\xi/\alpha & i\eta/\alpha & i\zeta/\alpha & 1 \end{bmatrix}$$

while ΔtD is the summation of the same matrices (5.24) omitting the diagonals :

$$\Delta tD = \begin{bmatrix} 0 & 0 & 0 & ig\xi \\ 0 & 0 & 0 & ig\eta \\ 0 & 0 & 0 & ig\zeta \\ i\xi/\alpha & i\eta/\alpha & i\zeta/\alpha & 0 \end{bmatrix}$$

Now we calculate the matrix $B = A - \Delta tD$:

$$B = \begin{bmatrix} 1 & -g/\alpha.\xi\eta & -g/\alpha.\xi\zeta & 0 \\ 0 & 1 & -g/\alpha.\eta\zeta & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

With the matrices A and B we can write the Fourier transform of (5.23) as :

$$A\ \hat{\psi}^{n+1} = B\ \hat{\psi}^{n}$$

The eigenvalues $\lambda = \psi^{n+1}/\psi^{n}$, follow from the characteristic equation:

$$A.\lambda - B = 0$$

or:

$$(\lambda-1)^2(ddd.\lambda^2 - 2\lambda + 1) = 0 \qquad (5.26)$$

$$\text{with } ddd = (1+g/\alpha.\xi^2)(1+g/\alpha.\eta^2)(1+g/\alpha.\zeta^2).$$

Solution of (5.26) gives:

$$\lambda_{1,2} = \frac{1}{ddd} \pm \frac{i}{ddd}\sqrt{(ddd-1)} \qquad \lambda_{3,4} = 1$$

The eigenvectors associated with the two eigenvalues $\lambda_{1,2}$ lie in the (hyper)plane through the wave number vector **k** and the h-axis. They represent the positive and negative direction of the pressure waves along **k**. These pressure waves are accompanied by longitudinal velocity waves. The absolute value of these eigenvalues is always smaller than unity.

The two "trivial" eigenvalues $\lambda_{3,4}$ can be associated with the two independent space directions perpendicular to **k**. Their meaning is that any transversal Fourier component of the velocity field will be unaffected by the time integration of (5.23).

Thus, as expected, the ADI method applied to the wave equation yields an unconditionally numerical process.

5.7 Conclusions

In the preceeding section is was proven that the 3-D ADI time integrating method is, at best, conditionally stable for scalar convective problems. The reason for this behaviour can be summarized thus :

Basic to the ADI-method is the factorisation of the expression:

$$1+\Delta t(D_x+ D_y+ D_z) \quad \text{into} \quad (1+\Delta tD_x)(1+\Delta tD_y)(1+\Delta tD_z) \qquad (5.27)$$

which means that, compared to the original, the ADI process adds certain terms (R in our notation) to the amplification factor G:

$$G = \frac{1 - \Delta t(1-\beta)(D_x+ D_y+ D_z) + R}{1 + \Delta t\ \beta\ (D_x+ D_y+ D_z) + R} \qquad (5.28)$$

For R=0 (general implicit method) the absolute value of G is less than unity if:

1. $\beta < {}^1/_2$
2. the real part of $D_x+D_y+D_z$ positive.

For those conditions we have unconditional (i.e. for every V_c, k and Δt) stability.

In case of the ADI method R differs from zero. In 2-D applications ($D_z=0$):

$$R = \beta^2\Delta t\ D_xD_y$$

For complex D_x,D_y with a predominant imaginary component, R will contain a predominant real component and the absolute value of G remains less than unity.

In the 3-D case R becomes predominantly imaginary due to the term $D_x D_y D_z$ and G can take values over unity. Thus, stability is, at best, conditional.

For the 3-D wave equation the matrix $D_x D_y D_z$ equals zero. Consequently the ADI solution method is unconditionally stable. Due to the simple structure of the matrices this can be shown analytically.

For the compressible or AC Navier-Stokes equations the same principles as deduced for the convection-diffusion equation are valid, but a proof as given in section 5.5 is much more complicated for the general case, due to the fact that D_x, D_y, D_z and R in expression (5.28) are now 4 x 4 matrices. Relation (5.28), now properly written as:

$$G = (1+\Delta t\beta(D_x+D_y+D_z)+R)^{-1}(1-\Delta t(1-\beta)(D_x+D_y+D_z)+R) \qquad (5.29)$$

is still valid, but the geometrical interpretation as given in section 5.5 is not possible.

Although by using the fully implicit Douglas-Rachford ADI scheme for the solution of the system (4.1) the allowed time step can be chosen two or three times larger than with the simple explicit Euler (or FTCS) method, we decided, for reasons explained in section 4.4, to use separate time-integration methods for the compression wave part and the convection-diffusion part.

For the compression wave we use ADI. Because of the relatively high propagation speed of the artificial compression wave, the use of a simple explicit scheme for this part would result in a very small time step. The relationship

between the allowed time step for an explicit scheme is expressed by the Courant number $C\Delta t/\Delta x$ where C is the propagation speed. This Courant number is a measure of the "numerical propagation speed", i.e. the speed of the numerical information transfer through the computational grid. Heuristically, it can be stated that, in order to have a stable computation, this numerical propagation should always exceed the physical propagation speed.

For the pressure wave part of the equations this propagation speed is in our notation :

$$c = \sqrt{(g/\alpha)}$$

As explained in section 4.5 this propagation should be sufficiently high to maintain the condition of incompressibility. For values of $\alpha = 0.5$ to 1. the condition that the Courant number must be less than unity for a simple explicit process leads to a maximum time step $\Delta t \approx 0.25\ \Delta x$.

The convection-diffusion part of the solution moves with the speed of the convecting velocity, which for our applications will be 1 a 2 m/sec. The Courant condition then leads to a time step of the same order as the numerical value of the grid size.

Thus, it is important that the time step limitation can be based on the convection-diffusion part of the equations rather than on the wave part.

In the next chapter we will see that the combination of this explicit scheme with ADI for the pressure wave has some stabilizing effect on the FTCS scheme. As a result one has to comply with the 1-D stability condition rather than the full 3-D condition. This 1-D condition is less stringent and thus a larger time step can be used.

6. THE STABILITY OF THE ADI-SCHEME IN COMBINATION WITH EXPLICIT CONVECTION-DIFFUSION TERMS

6.1 Introduction

For reasons given in section 4.4 we have adopted a different time integration procedure for the terms of the Navier-Stokes equations that describe the (artificial) pressure wave and for the convection-diffusion terms. In chapter 5 it was shown that the ADI scheme is unconditionally stable if used for the pressure wave part. The convection-diffusion part is integrated in time by a simple Euler explicit scheme. One can expect the stability condition of the total computational system to be equal to the stability condition for the explicit process. If both steps are executed consecutively as in (4.27) this is indeed the case.

As mentioned in 4.4.2 another sequence of operations is possible. Here we will approach this possibility from a different starting point. In 4.3.2 the "delta"- formulation of the ADI process was brought to attention mainly as a means of reducing the amount of necessary memory space. Here we use this formulation to illustrate a more fundamental aspect.

We repeat this formulation (4.24) in a different form :

$$A\ (\ \psi^{n+1} - \psi^{n}) = -\Delta t D\ \psi^{n} \qquad (6.1)$$

N.B. Contrary to its use in (5.25) where the wave equation was considered, (6.1) now is intended to contain all the terms of the N.-S. AC equations (4.1).

If we consider (6.1) as an iteration process towards a stationary solution ψ, this solution will be independent of

the elements of A. This idea is expressed by MacCormack (1985) symbolically as :

$$\text{(Numerics)}\ \Delta\psi = \text{(Physics)}\ \psi$$

Thus, we can omit the convection-diffusion terms on the left-hand side of (6.1) without consequences for the final (stationary) solution which will be reached as ψ^{n+1} approximates ψ^n. This happens to result in a scheme that is identical to the scheme (4.29) where the explicit steps are executed intermittently with the directional steps of the ADI process. It will be shown in section 6.3 that this ordering gives better stability properties to the total scheme.

Using the same technique as in section 5.6, we will determine the characteristic equation associated with systems of the form :

$$A\ \hat{\psi}^{n+1} = B\ \hat{\psi}^n \qquad (6.2)$$

which we will use as a model for the time integration of the AC Navier-Stokes equations. In the next sections we will determine the characteristic equation of (6.2) for the two alternative ways to incorporate the explicit convection-diffusion terms in the total system as mentioned in section 4.4.

6.2 Alternative I : Separate Treatment of the Explicit Terms

In the first alternative (4.27) we start by advancing the numerical solution ψ with the explicit convection-diffusion terms for the three velocity components :

$$\psi' = \begin{bmatrix} V' \\ h' \end{bmatrix} = \begin{bmatrix} V^n - \Delta t \text{ (Conv. + Diff.)} \\ h \end{bmatrix}$$

or, using the abbreviation iCD for the convection-diffusion terms of the Fourier transform (as in 4.8)):

$$\psi' = \begin{bmatrix} (1\text{-iCD}) & 0 \\ 0 & 1 \end{bmatrix} \psi^n \qquad (6.3a)$$

This step is followed by the Douglas-Rachford ADI scheme which advances the pressure values and at the same time corrects the velocities :

$$A\ \psi \overset{n+1}{=} B\ \psi'$$

or:

$$A\ \psi^{n+1} = B\ (1\text{-iCD})\ \psi^n \qquad (6.3b)$$

where A and B are the matrices as given for the wave equation (5.25)

Performing the multiplication on the righthand side of (6.3b) we get:

$$A\ \hat{\psi}^{n+1} = B_1 \hat{\psi}^n \qquad (6.4)$$

with B_1 :

$$B_1 = \begin{bmatrix} 1\text{-iCD} & -g/\alpha\ xy(1\text{-iCD}) & -g/\alpha\ xz(1\text{-iCD}) & 0 \\ 0 & 1\text{-iCD} & -g/\alpha\ yz(1\text{-iCD}) & 0 \\ 0 & 0 & 1\text{-iCD} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The corresponding characteristic equation is :

$$A\,\lambda - B_1 = 0$$

or, after some calculation :

$$(\lambda - (1\text{-}iCD))^2[ddd\lambda^2 - (2\text{-}iCD)\lambda + (1\text{-}iCD)] = 0 \qquad (6.5)$$

with ddd as in (5.26)

giving the eigenvalues :

$$\lambda_{1,2} = \frac{1-{}^1/_2 iCD}{ddd} \pm \frac{i}{ddd}\sqrt{\{(1\text{-}iCD)(ddd\text{-}1) - {}^1/_4(iCD)^2\}} \qquad (6.6)$$

$$\lambda_{3,4} = 1\text{-}\ iCD$$

From this result it is clear that the longitudinal components (i.e. the compression wave) are damped by the fully implicit ADI method. The two transverse components of the solution vector are amplified with a factor identical to the separate explicit convection-diffusion equation (6.2) as was to be expected. Thus, the stability properties of the total scheme are equal to those of the explicit Euler scheme.

6.3 Alternative II : Explicit Terms Integrated in ADI Scheme

The second alternative is represented by the system :

$$A\,\hat{\psi}^{n+1} = B_2\,\hat{\psi}^n \qquad (6.7)$$

Again the matrix A is as in (5.25). To obtain the matrix B_2 we first write (6.7) in the delta-form:

$$A\ (\hat{\psi}^{n+1} - \hat{\psi}^{n}) = (B_2 - A)\ \hat{\psi}^{n} \qquad (6.8)$$

Within the brackets on the righthand side of this expression we must have the coefficients of the stationary Navier-Stokes equations:

$$B_2 - A = \begin{bmatrix} -iCD & 0 & 0 & -ig\xi \\ 0 & -iCD & 0 & -ig\eta \\ 0 & 0 & -iCD & -ig\zeta \\ -i\xi/\alpha & -i\eta/\alpha & -i\zeta/\alpha & 0 \end{bmatrix}$$

Adding A to this matrix we get for B_2 :

$$B_2 = \begin{bmatrix} 1-iCD & -g/\alpha.\xi\eta & -g/\alpha.\xi\zeta & 0 \\ 0 & 1-iCD & -g/\alpha.\eta\zeta & 0 \\ 0 & 0 & 1-iCD & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

With this result the characteristic equation for the scheme (4.28) reads :

$$A\ \lambda - B_2 = 0 \qquad (6.9)$$

or :

$$ddd.\lambda^4 - \{2ddd+2-iCD(d_x d_y+d_x d_z+d_y d_z)\}\lambda^3 +$$

$$+\{ddd+5-iCD(6+d_x d_y+d_x d_z+d_y d_z)+(iCD)^2(d_x+d_y+d_z)\}\lambda^2 -$$

$$-\{3(1-iCD)^2+(1-iCD)^3\}\lambda + (1-iCD)^3 = 0$$

with ξ, η, ζ as in (5.24)

$ddd = d_x d_y d_z$

$d_x = (1+g/\alpha.\xi^2)$

$d_y = (1+g/\alpha.\eta^2)$

$d_z = (1+g/\alpha.\zeta^2)$

The analytical solution of this equation is not as easy as in the case of the first alternative. Some confidence in its correctness can be obtained by noting that for the approximations :

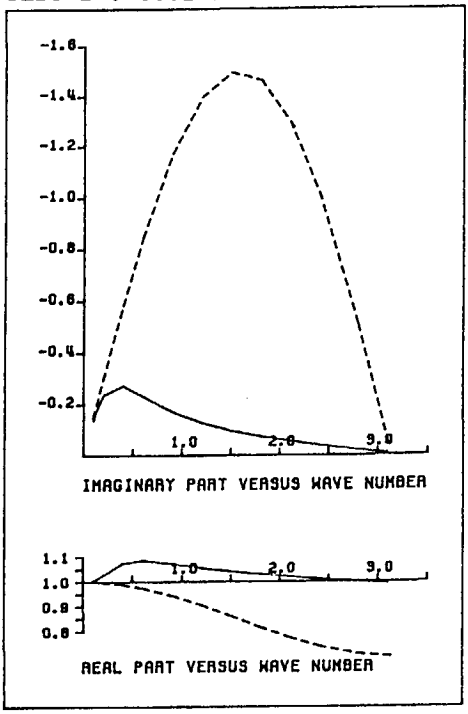
$$d_x d_y + d_y d_z + d_x d_z \approx 1 + 2\, d_x d_y d_z$$

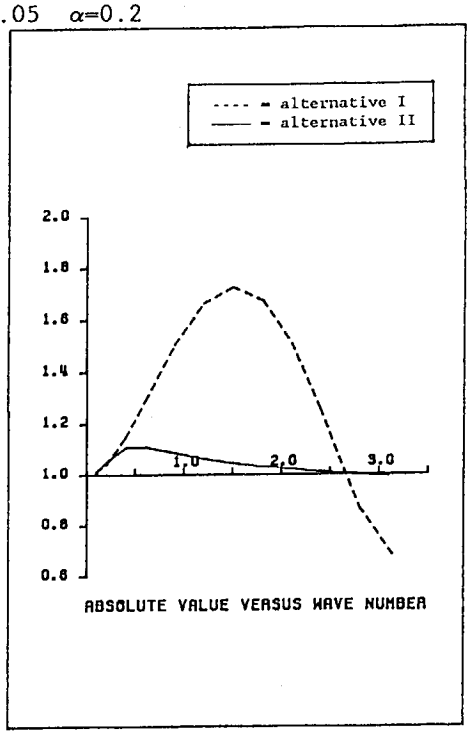
$$d_x + d_y + d_z \approx 2 + d_x d_y d_z ,$$

equation (6.9) reduces to (6.5). If the wave number vector $\mathbf{k}$ is chosen in the direction of one of the coordinate axes the approximations mentioned above are exact and thus the same eigenvalues are found as in alternative I.

Using numerical methods it is possible to calculate the eigen-values and -vectors of (6.9) for specific values of $\mathbf{k}$, V_c, α, β and ν. In fig. 6.1 the results of such a computation are shown. In fig. 6.1a the real and imaginary

case I : Courant=0.5 diff.number=0.05 α=0.2





case II : Courant=0.2 diff.number=0.05 α=0.2

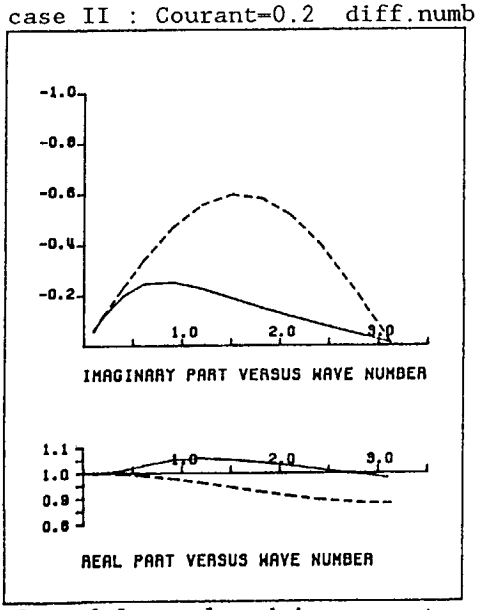


fig. 6.1a real and imag. parts

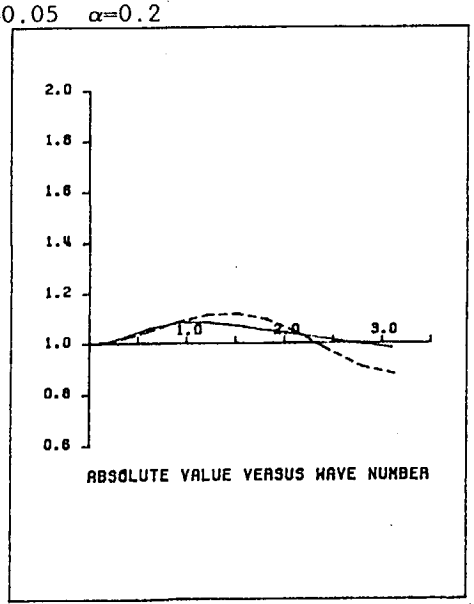


fig. 6.1b absolute value

components, and in fig. 6.1b the absolute value of the largest eigenvalue is plotted versus the absolute value of the wavenumber **k**. In the same figure the corresponding value for alternative I (or purely explicit) is indicated. In this case the direction of **k** is such that :

$$\kappa_x \Delta x = \kappa_y \Delta y = \kappa_z \Delta z \quad (\text{or } \xi = \eta = \zeta)$$

It can be observed that, while the theoretical stability limits are about the same, the value of the amplification and hence the growth of the instability is less for alternative II for this direction of **k**.
From this kind of computations the following properties of the scheme according to alternative II can be found:

For directions of **k** not along the coordinate axes, the absolute value of the greatest eigenvalue for alternative II is closer to unity than the greatest absolute value according to (6.6) for alternative I. Furthermore, the imaginary component of the eigenvalues associated with the transverse components of the solution is reduced in the directions away from the coordinate axes.

An interesting consequence of this reduction of the imaginary component is that the error in the convecting velocity, made by the spatial discretisation process (see section 4.2.4), is to a certain extent compensated for by the error in the time integration. While the imaginary component of both eigenvalues associated with the transverse directions is reduced, the real component is somewhat increased in one direction and somewhat decreased in the other. Thus, in the transient phase of the solution, some positive and negative numerical viscosity is added to the explicit process by the intermixing with the ADI integration.

Another result is that the largest eigenvalue will be reduced automatically, if the explicit stability limit is locally exceeded. Along the coordinate axes, the ADI process does not influence the explicit steps, but for these directions of the wave number vector the computation is in effect one dimensional. Thus, the practical stability of the total scheme will be close to the 1-D limit.

6.4 Further Comparison between Alternative I and II

The properties of the two alternative scheme's can also be illustrated in terms of the "ADI error term" R as defined by (5.20) :

Alternative I leads to the expression :

$$(1 + R + D)\ \psi^{n+1} = (1 + R)\ (1\text{-}iCD)\ \psi^{n} \qquad (6.10)$$

while alternative II results in :

$$(1 + R + D)\ \psi^{n+1} = (1 + R - iCD)\ \psi^{n} \qquad (6.11)$$

In these expressions CD again contains the convection diffusion terms while D represents the remaining terms of the Navier-Stokes equations.

To compare the two expressions (6.10) and (6.11) we rewrite (6.11) as :

$$(1 + R + D)\ \psi^{n+1} = (1 + R)\ (1 - (1+R)^{-1} iCD) \qquad (6.11a)$$

Instead of the term iCD in (6.10) we have $(1+R)^{-1}iCD$ in (6.11a), which represents the influence of the implicit ADI part of the computation on the explicit part.

Another point of interest is revealed if we write (6.10) and (6.11) in delta-form :

Alternative I :

$$(1 + R + D)\ (\psi^{n+1} - \psi^{n}) = (-\ D\ -\ iCD\ -\ iCD.R)\ \psi^{n} \qquad (6.10b)$$

Alternative II :

$$(1 + R + D)\ (\psi^{n+1} - \psi^{n}) = (-\ D\ -\ iCD)\ \psi^{n} \qquad (6.11b)$$

While the second alternative (6.11b) converges to the exact stationary solution $(D + iCD)\ \psi = 0$, the converged solution of the first alternative (6.10b) still contains an error term iCD.R. While this does not make (6.11) inconsistent (R goes to zero with Δt), it makes the final stationary solution dependent on, among others, the time step and the compressibility factor α.

From the arguments given in this chapter it will be clear that in our opinion the second alternative (4.29) is to be preferred.

7. TURBULENCE

7.1 Introduction

Although the effect of turbulence on the flow is of utmost importance we will in this thesis restrict ourselves to very simple turbulence models. A turbulence model is needed for the closure of the Navier-Stokes equations (2.1-2.2). These equations are theoretically valid for turbulent flow but existing numerical solution techniques cannot as yet deal with the highly irregular small scale turbulent motion. To solve the equations for turbulent flow in a manner as discussed in this thesis would involve extremely fine grids and corresponding small time steps, leading to prohibitive computing times.

Instead one solves the Navier-Stokes equations for the mean values of the flow variables. As will be explained in section 7.2 the coefficient of the viscosity term in this case no longer represents the molecular viscosity ν but rather the flow dependent "turbulent viscosity" ν_t. The relation between the flow-variables proper (p, u, v and w) and ν_t forms the above mentioned closure problem. It necessitates some turbulence model. Many of these models exist and are used in engineering practice. Still, there is much difference of opinion about the applicability of the various models. The problem of turbulence modelling can be regarded as a separate problem and in this thesis only the most simple form of turbulence model, the constant eddy viscosity, is employed.

In section 7.2 the principles, on which the idea of turbulent or eddy viscosity is based, are discussed and in section 7.3 some useful formulae for the a priori determination of the eddy viscosity are given.

7.2 Turbulence Models

7.2.1 Theoretical Background

Most turbulence models presently in use are based on the following two concepts:

a. The separation of the turbulent motion in a mean and a fluctuating part.

b. The translation of the extra terms that arise from the non linearity of the original equations into "viscous" terms.

The first concept is due to Reynolds(1895), the second to Boussinesq(1877, historically unconnected to Reynolds' research. Boussinesq tried to describe simply the greater resistance felt by turbulent flow compared to laminar flow as an artificial viscosity effect).

We start with the Navier-Stokes equation (2.1) where ν now represents the molecular viscosity coefficient. According to Reynolds the flow variables are considered to consist of a mean and a fluctuating part :

$$\begin{aligned} V &= \bar{V} + v' \\ p &= \bar{p} + p' \end{aligned} \qquad (7.1)$$

After insertion of the righthand sides of (7.1) into the Navier-Stokes equation, the ensemble average of this equation is taken. As a result of this operation one gets an equation for the mean values of the flow variables that is identical to the original N.-S. equation for the linear terms. Due to the non-linearity of the convection term an additional term appears:

$$\overline{(\mathbf{V}\ \text{grad}).\mathbf{V}} = \overline{((\bar{\mathbf{V}}+\mathbf{v}')\ \text{grad}).(\bar{\mathbf{V}}+\mathbf{v}')} = (\bar{\mathbf{V}}\ \text{grad}).\bar{\mathbf{V}} + \overline{(\mathbf{v}'\text{grad}).\mathbf{v}'}$$

This additional term $\overline{(\mathbf{v}'\text{grad}).\mathbf{v}'}$ is in most textbooks written in its components :

$$\begin{aligned} &\partial/\partial x(\overline{u'^2}) + \partial/\partial y(\overline{u'v'}) + \partial/\partial z(\overline{u'w'}) \\ &\partial/\partial x(\overline{u'v'}) + \partial/\partial y(\overline{v'^2}) + \partial/\partial z(\overline{v'w'}) \\ &\partial/\partial x(\overline{u'w'}) + \partial/\partial y(\overline{v'w'}) + \partial/\partial z(\overline{w'^2}) \end{aligned} \qquad (7.2)$$

or in the Ricci notation :

$$\partial/\partial x_j(\overline{u_i'u_j'})$$

In this notation $\mathbf{V} = (u_1,u_2,u_3)^T$ and the spatial ordinates : x_1,x_2,x_3.
i and j assume the values 1,2 and 3 while repeated appearance of an index in a term means summation over the three values of that index (Einstein convention).

The averaged products in (7.2) can be interpreted as stresses and are consequently known as the Reynold stresses. As such they can be combined with the viscosity term in the original N.-S. equation :

$$\partial u_i/\partial t + {}^1/\rho\partial p/\partial x_i + u_j\partial u_i/\partial x_j - \partial/\partial x_j(\tau_{ij}/\rho) = 0 \qquad (7.3)$$

$$\text{where } \tau_{ij}/\rho = \nu\ \partial u_i/\partial x_j - \overline{u_i'u_j'}$$

The terms $-\overline{u_i' u_j'}$ represent the unknown correlations between the fluctuating motions and to be able to use equation (7.3) their values must be related to the mean flow variables. This relation forms the closure problem mentioned in section 7.1.

7.2.2 Eddy Viscosity

In numerical flow computation the relation between the turbulent stresses $-\overline{u_i' u_j'}$ and the mean flow variables is commonly taken from Boussinesq's assumption that the turbulent stresses (i.e. the second term in the expression for τ_{ij}/ρ in (7.3)) can be expressed in analogy to the viscous stress as the product of a coefficient ν_t and the local velocity gradient. This coefficient is called the "turbulent viscosity" or "eddy viscosity" coefficient. Because its value is flow dependent and thus spatially varying we can no longer employ the simple viscosity term as in (2.1). Instead, the expression for the eddy viscosity now consists of two terms for each velocity component :

$$\partial/\partial x_j(\tau_{ij}/\rho) = \partial/\partial x_j[(\nu + \nu_t)(\partial u_i/\partial x_j + \partial u_j/\partial x_i)] \qquad (7.4)$$

Generally, the value of ν is negligible compared to ν_t, except in the boundary layer along the wall where the turbulence decreases to zero. In our model, where the boundary layer is represented by a logarithmic wall function (see section 8.3), the molecular viscosity coefficient can be omitted from (7.4).

Because of the Boussinesq assumption the problem of turbulence modelling is shifted from the unknown Reynold stresses to an unknown eddy viscosity. Most turbulence

models in use today try to describe the distribution of this eddy viscosity over the computational domain.

A well known formula that links the local eddy viscosity directly to the local flow conditions is the mixing length formula (Prandl,1925) :

$$\nu_t = l^2 |\partial u/\partial y| \qquad (7.5)$$

with l = a mixing length

and $\partial u/\partial y$ the crossflow gradient

The use of this formula shifts the problem of describing the eddy viscosity to that of the mixing length. For the determination of the mixing length (i.e. the spatial scale of the eddies) formulas exists for several situations. For instance in the neighbourhood of a solid boundary the mixing length is often taken as (von Karman,1930) :

$$l = \kappa\, y \qquad (7.6)$$

with $\kappa = 0.4$

and y = distance to the wall

Remark :

If (7.6) and (7.5) are used to express ν_t in the equation for the wall shear stress :

$$\tau_w = \rho\, \nu_t\, du/dy$$

the solution of this differential equation results in the "law of the wall" :

$$u(y) = u^*/\kappa \ \ln(y/y_0)$$

with $u^* = \sqrt{(\tau_w/\rho)}$ (or wall friction velocity)
and y_0 = integration constant related to the bottom roughness

Another, more general formula is due to Buleev (see Rodi-,1984) :

$$l = \kappa/\pi \iint(^1/d)d\Omega \qquad (7.7)$$

with d = distance to the wall
and $\iint(..)d\Omega$ = integration over the cross section of the flow domain

The most advanced turbulence model based on the eddy viscosity concept is the κ-ϵ model. In this model the eddy viscosity is a function of the turbulent energy κ and the rate of dissipation ϵ. Both these quantities are treated as substances that are generated, transported and diffused by the flow. Because not all these phenomena can be formulated exactly, new unknown correlations appear, for which assumptions must be taken. This results in a set of 5 constants, that are assumed to be valid for a wide range of flow situations. A more extensive description can be found in Rodi (1984).

7.3 The Constant Eddy Viscosity Model

In the present stage of development we do not as yet consider it meaningful to employ an advanced turbulence model. Most turbulence models are based on the local velocity gradient. And, while the grid density may be sufficient to describe the general flow field, the gradients, especially near singular points where the velocity changes rapidly, are represented not accurately enough to be used as a basis for a turbulence model. If such a model is based on the local gradients only, as in the mixing length model, one may correct for the coarseness of the grid by locally adapting the value of the mixing length, but this presumes knowledge about the development of shear layers in the model.

Thus, in a flow model as is described here one quickly reaches the point where a turbulence model is more refined than the flow model itself. Of course, with a simple eddy viscosity model the responsibility for the right modelling of turbulence effects is shifted towards the user.

The most simple turbulence model assumes the eddy viscosity equal to a constant over the computational domain. This would be acceptable in flow situations where the turbulence can be considered homogeneous, for instance in regions where the flow is accelerating everywhere because of a decreasing cross section.

A more general model can be obtained by allowing the eddy viscosity to be a spatially varying but known quantity. This involves a certain a priori knowledge of the flow. This knowledge can be based on experience or even on trial and error calculations with different eddy viscosity distributions. Useful formulas are given by Lean and Weare (1979) for two cases of turbulence generation :

free shear layer generated turbulence :

$$\nu_t = {}^1/_2 \, U_0 \, x/R \qquad (7.8)$$

with U_0 = free stream velocity
x = downstream distance to separation point
R = numerical constant = 288

bottom generated turbulence :
(depth averaged value)

$$\nu_t = 0.08 \, u^* \, d \qquad (7.9)$$

with $u^* = \sqrt{(\tau_w/\rho)}$
d = depth

remark : Lean and Weare use in formula (7.9) a numerical constant 0.16 but introduce a factor $^1/_2$ in their relation between u* and the mean velocity. Consequently we use for the numerical constant the value 0.08. A theoretical derivation of this same formula leads to a numerical factor 0.067 or $\kappa/6$.

Another formula for the turbulence generated in a free shear layer is given by Prandl (see Rodi,1984) :

$$\nu_t = C\ y\ |U_{max} - U_{min}| \qquad (7.10)$$

with C = a constant = 0.01 - 0.026
and y = layer width (or distance between U_{max} and U_{min}.

Remark :

If formula (7.8) and (7.10) are combined, we find a relation between the streamwise development x and lateral development y of the shear layer. Equating ν_t in (7.8) and (7.10) gives :

$$^1/_2\ U_0\ x/R = C\ y\ U_0$$

Taking the values R=300 and C=0.01 (the latter value is suggested in Rodi,1984 for plane shear layers) we get for the relation y/x :

$$y/x = 6$$

This can be considered as a measure of the reattachment length after a backstep and it agrees well with the usual value of about 7.

8. BOUNDARY TREATMENT

8.1 Introduction

In a practical numerical model the boundary treatment is of extreme importance because the flow is driven, and thus completely determined, by the boundary conditions. Boundaries can be divided in active boundaries like inflow and outflow boundaries and passive ones like solid walls and symmetry planes.

It is advantageous to make a further distinction. If we consider the Navier-Stokes equations as a combination of the wave equation and the convection-diffusion equation (see section 4.4.1), we can give, in the numerical approximation, separate boundary conditions for these two parts of the full N.-S. equations. The two sets of boundary conditions should not conflict with each other.

For the wave equation either the velocity component normal to the boundary or the pressure must be specified at all boundaries. At the inflow usually the normal velocity component is specified and at the outflow the pressure distribution. The reason for this is given below. If the outflow boundary is chosen far enough downstream of the area of interest, a uniform pressure distribution, representing a hydrostatic pressure distribution, can be given in most cases.

On solid walls as well as on symmetry planes we can simply impose a zero normal velocity, so a pressure condition is never needed in this case.

For the convection diffusion equation, because of its parabolic character, the normal and tangential components of the velocity must be given on all boundaries. However, for

high Reynolds numbers steep velocity gradients will appear at the outflow boundary, if the boundary condition does not match the value of the velocity that is convected towards the outflow boundary. Thus, a very small grid size is necessary to represent the analytical solution. This effect, known as the numerical outflow boundary layer, is controlled by the cell Reynolds number :

$$Re_c = V_c \Delta x / \nu$$

It can be shown (Vreugdenhil,1989) that for the explicit time integration with central differences (FTCS) the cell Reynolds number Re_c must be less than 2 to prevent an oscillatory solution near the outflow boundary. This severe restriction on the grid size can be removed by specifying a zero second derivative for all velocity components on the outflow boundary, because this neutralizes the viscosity term.

Thus, we do not want to specify the velocity itself at the outflow boundary. Instead we use in the wave equation part of the solution process a pressure condition on the outflow, together with the above mentioned second derivatives of the velocity components.

On solid boundaries unfortunately we must specify a zero normal velocity, but here the problem is less severe, as the convecting velocity normal to the wall is generally small. Furthermore, in most cases a slip condition is given on the wall (see section 8.3) which diminishes the gradients of the tangential velocity components near the wall.

In table 8.1 the boundary conditions that are used in the applications of chapter 9 are summarized. In this table we use the notation V_n for the velocity component normal to the boundary and V_t for the tangential component.

	wave equation	convection-diff. equation
inflow	V_n given	V_n and V_t given
outflow	uniform pressure	$\partial^2 V_n/\partial n^2=0$, $\partial^2 V_t/\partial n^2=0$
solid wall	$V_n=0$	$V_n=0$, V_t= slip
symmetry	$V_n=0$	$V_n=0$, $\partial V_t/\partial n=0$

table 8.1 Boundary Conditions

As can be observed in table 8.1 there are no conflicting boundary conditions for the two parts of the computational procedure.

The second derivative of V_n that is specified for the convection diffusion part of the equations is formally extraneous for the wave equation, but no problems were encountered in the computations in this respect.

At solid walls and symmetry planes the boundary conditions are effectuated by means of a slip factor (see section 8.3.3). They are treated implicitly, i.e. the derivatives are evaluated and used in the same time step. In contrast, the second derivatives at the outflow boundary are taken explicitly : they are evaluated on the basis of the velocities of the old time step. For the computation of stationary flow this difference is not important.

Another class of problems is connected with the use of a regular grid. The main disadvantage of the regular finite

difference grid is its inability to follow the physical boundaries other than in a "staircase"-like manner. For inflow and outflow boundaries this is in many cases less important because the modeller is often free in his choice of the exact location of those boundaries, viz. they can often be taken along grid lines. Closed boundaries, however, are generally dictated by the physical structure one wants to model so in this case it is important that the grid can accurately represent the shape of the physical boundary. In 2-D models one can sometimes circumvent the problem by applying a very fine grid, but in 3-D applications this would lead to a prohibitive amount of gridpoints.

A common method to solve this problem is the use of generalized, or boundary fitted, coordinates. In this method the physical space is mapped or transformed to a simple rectangular computational domain. This tranformation is such that the physical boundaries coincide with coordinate lines in the computational domain, removing the need for complicated boundary conditions. The method works well if the curvilinear grid can be chosen along streamlines of the flow, but if recirculating flows are present, this is generally impossible. In such cases care must be taken that transformation errors do not degrade the accuracy of the flow computation. Thus, this method is especially useful in flows around streamlined bodies (e.g. airfoils), but less so when the boundaries exhibit sharp corners. Application of the method in practical engineering use involves complicated grid generation programs and the storage of much geometrical data. Nevertheless many reports of succesfull application exist in literature (for instance: Kwak,1989).

Still, in our opinion the advantages of a simple rectangular grid do prevail over its disadvantages, if one aims toward minicomputer applications. Furthermore, possibilities

exist to improve the approximation of closed boundaries. By using a finite volume approach for the gridcells adjacent to the boundary, in a method known as the porosity method or fractional area/volume method (Hirt,1984), the above mentioned disadvantages can be overcome to a great extent. This method will be explained in section 8.2.

Another problem inherent to the regular grid is that it is generally too coarse to represent boundary layers, unless one is again prepared to use an enormous amount of grid points. A well-known solution to this problem is the use of wall functions. This involves a slip condition on the wall. To impose such a condition on a staircase-boundary requires some ingenuity. This problem will be dealt with in section 8.3.

8.2 The Porosity Method

8.2.1 Description of the Boundary Problem

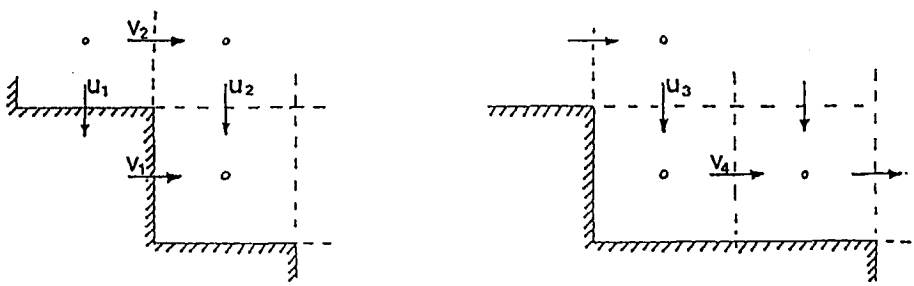
Because we use a simple regular finite difference grid we cannot accurately describe, without further measures, solid boundaries other than those that coincide with cell boundaries. Oblique or curved boundaries must be represented in a "staircase" manner as indicated in fig. 8.1. This representation has several drawbacks.

In the general case the influence of the staircase-boundary will extend over several grid distances away from the boundary. In order to represent the flow along the wall more or less accurately a fine grid must be used.

A more fundamental disadvantage is the impossibility to specify a free slip condition on a general oblique boundary. Some authors (Weare,1979) maintain that a staircase-boundary

always results in a no-slip condition. Usually the following argument is given :

Velocity component V_1 in fig. 8.1a must be set to zero to satisfy the condition of no mass flow through the wall. Accordingly component V_2 always "feels" a no-slip condition.



a: 1:1 b: 1:2

staircase boundaries

fig. 8.1

This problem can be partly solved by considering V_1 in fig. 8.1a as a non-zero tangential component in diffusion terms of the convection diffusion equation and as a zero normal component in the flow terms in the continuity equation. In this way it is possible to construct, at least for the stationary case, a free slip condition on a boundary line that makes an angle of 45 degrees with the grid lines as in fig. 8.1a.

But in the situation in fig. 8.1b, where the boundary intersects the grid in the ratio 1:2 the use of different boundary conditions for the different terms is not sufficient to attain a free slip condition. In order to satisfy continuity, velocity component V_4 must have the same value as U_3, while the free slip condition should result in $V_4 = 2*U_3$. Thus, the value of V4 will be too low and even when the boundary condition for the diffusion terms is set to free slip, some wall friction will be felt by the flow.

8.2.2 Finite Volume Approach

To counteract the difficulties mentioned in the foregoing section, we decided to use a finite volume approach to the grid cells adjoining the solid boundaries. In a finite volume method the finite difference equations are written in flux form:

$$\partial\psi/\partial t + \partial F/\partial x + \partial G/\partial y + \partial H/\partial z = 0 \qquad (8.1)$$

where $F(\psi)$, $G(\psi)$ and $H(\psi)$ are fluxes through the cell-faces

Equation (8.1) is integrated over the cell using Green's theorem :

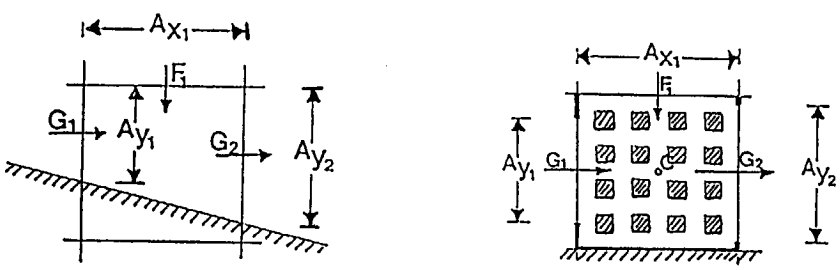
$$\partial/\partial t \iiint \psi \, dV + \iint F \, dS_x + \iint G \, dS_y + \iint H \, dS_z = 0 \qquad (8.2)$$

If ψ is taken constant over the cell volume V and the fluxes are taken constant over their respective surfaces S_x, S_y and S_z, (8.2) simplifies to :

$$\partial/\partial t(\psi V) + F_2Ax_2 - F_1Ax_1 + + G_2Ay_2 - G_1Ay_1 + H_2Az_2 - H_1Az_1 = 0 \qquad (8.3)$$

with F_1 , Ax_1 ,etc as indicated in fig. 8.2a

In the case of the equation of continuity the fluxes are obtained by multiplication of the velocity components with the appropriate cell surfaces. In this form the equation is automatically mass conserving on a staggered grid where the velocity components are situated on the cell faces.



a: finite volume b: porosity

fig. 8.2

For the equations of motion the formulation is more complicated because it is not possible, in the staggered grid, to define a cell (or finite volume) in such a way that all

necessary velocity components and pressures are available at the cell surface. Thus, interpolations are necessary to formulate the fluxes in this case.

8.2.3 The Porosity Formulation

A method that is closely associated with the finite volume formulation is the so-called porosity method. In this method the equations are formulated in flux form, as in the finite volume method. The only difference with the finite volume method proper is that the geometrical data are based on the regular grid rather than on the real boundary. In the finite volume formulation the partly cut off-cell would be represented as in fig. 8.2a, with the fluxes situated in the center of the "open" part of the cell faces. In the porosity formulation (fig. 8.2b) this same cell takes the position of the full cell. The respective surfaces, however, are given an "open" area conforming to the corresponding cut-off area in fig. 8.3a. Likewise the volume of the cell is reduced to that of the cut-off cell, but the cell center remains at the regular grid point C. The name "porosity" reflects the fact that the cell is not cut off by the boundary, dividing it in a open and solid part, but made porous by a blockage evenly distributed over the cell.

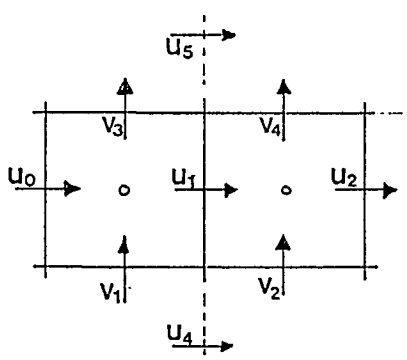
If the quantity, needed to formulate the flux, is directly available on the cell face, the porosity formulation is just as accurate as the finite volume method. If interpolation is necessary, the fact that the geometrical locations of the concerned values are not correct may affect the result. The velocity fluxes on oblique cell faces are always zero because of the solid boundary so the fact that

the flux should be perpendicular to the cell surface is not important here.

The advantage of the porosity method over the finite volume method lies in the fact that in the former method the grid remains regular. Thus, the amount of geometrical data to be retained is reduced to the "porosities" of the cells. This means that, for the cells adjoining the solid boundary, six surface areas and one volume per cell have to be stored with the grid data.

In the calculations in chapter 9 of this thesis the porosity method is used for the formulation of the equation of continuity and for the formulation of the convection terms in the equations of motion. Application to the continuity equation is straightforward conform fig. 8.2b.

The porosity formulation for the equations of motion is more complicated because not all necessary quantities are known on the cell faces. They have to be approximated by interpolations. As an example we consider the u-equation. The computational cell is now situated centered around U_1.



double cell for equation of motion

fig. 8.3

Because we do not want to store a separate cell system with its own volumes and surfaces, we choose as a control volume a double-cell consisting of two "continuity" cells as indicated in fig. 8.3.

In terms of equation (8.1) the fluxes in the u-equation of motion are expressed as :

$$\begin{aligned} F_x &= gh + u^2 - \nu\, \partial u/\partial x \\ G_x &= \quad uv - \nu\, \partial u/\partial y \\ H_x &= \quad uw - \nu\, \partial u/\partial z \end{aligned} \tag{8.4}$$

The pressure term gh poses a difficulty because the pressure is not specified on the solid boundary (The fact that this is not necessary is one of the advantages of the artificial compressibility method). In our computational scheme the pressure term is approximated with regular finite differences.

The friction terms are also difficult to incorporate in the porosity formulation. Here the fact that the true geometrical data are not available, i.e. the distance of the cell centre to the solid boundary is not known, hinders an accurate description. This problem is dealt with in section 8.4.

As a result of the above mentioned considerations our version of the porosity method can be expressed as follows :

$$\partial/\partial t \iiint u\, dV + \iint u^2\, dS_x + \iint uv\, dS_y + \iint uw\, dS_z = \iiint \left(-g\partial h/\partial x + \partial/\partial x_j(\tau_{ij}/\rho) \right) dV \tag{8.5}$$

where for the viscosity term the expression (7.4) is used.

Equation (8.5) shows that in the expression on the left-hand side momentum is conserved over the control volume V. The righthand side represents the pressure and friction forces. They are expressed in the non-conserving finite difference formulation.

The separate terms of the lefthand side of (8.5) are calculated according to :

$$\partial/\partial t \iiint u \, dV = (u^{n+1} - u^{n}) * V$$

with V = open cell volume

$$\iint u^2 \, dS_x = Ax_2 * U_2 * U_2 - Ax_0 * U_0 * U_0$$

$$\iint uv \, dS_y = (Ay_3 + Ay_4) * (V_3 + V_4) * (U_5 + U_1)/2 - (Ay_1 + Ay_2) * (V_1 + V_2) * (U_4 + U_1)/2$$

$$\iint uw \, dS_z = (Az_3 + Az_4) * (W_3 + W_4) * (U_7 + U_1)/2 - (Az_1 + Az_2) * (W_1 + W_2) * (U_6 + U_1)/2$$

according to the nomenclature in fig. 8.3

Treatment of the v- and w- equations of motion is similar.

8.3 Wall Friction

8.3.1 Introduction

The physically correct boundary condition of viscous flow is no-slip. This means that all three components of the flow velocity should be zero at the solid boundary. However, in the boundary layer along a solid wall the turbulence decreases to zero towards the wall so the turbulent viscosity coefficient ν_t also tends to zero. At the wall the

velocity gradient of the flow component parallel to the wall is determined solely by the molecular viscosity coefficient ν. In the case of a high Reynolds number the gradient in the region close to the wall is very steep and for its proper representation a very fine grid would be necessary. This can be avoided by using a parametric representation for the velocity in the boundary layer, commonly known as a wall function. For turbulent flows this function generally describes a logarithmic profile over the boundary layer and is used to relate the velocity gradient in the first grid-point away from the wall to the physically required no-slip boundary condition at the wall.

From a computational point of view the Dirichlet boundary condition of no slip is replaced by a Neumann type boundary condition, prescribing the gradient of the velocity parallel to the wall. By means of the wall function this gradient is a function of the velocity in the first grid point away from the wall.

8.3.2 The Wall Function

At the wall the turbulence is suppressed and the flow is completely determined by the molecular viscosity of the fluid. For hydraulically smooth walls a laminar viscous sub-layer exists very close to the wall. A little further away from the wall the flow gradually becomes turbulent. This turbulent boundary layer may encompass the whole flow region. In most engineering applications we have walls that are hydraulically rough and only a turbulent boundary layer exists at the wall. In the region close to the wall the tangential component of the velocity has a very steep gradient. To avoid a very small grid size to accomodate these

gradients we do not use a no-slip boundary condition at the wall proper, but instead a boundary condition based on the "law of the wall" (Rodi,1980, see also section 7.2).

The associated theory is based on the assumption that the value of the shear stress τ_w at the wall can be considered constant for the steep gradient part of the turbulent boundary layer, or, in our case, for the region between the wall and the first grid point away from the wall. For simplicity we first consider a 2-D region with u the tangential velocity and y the perpendicular distance to the wall.

The turbulent viscosity coefficient ν_t is expressed with Prandl's mixing length formula (7.5) :

$$\nu_t = l^2 \, \partial u/\partial y$$

where u = the flow velocity along the wall
and y = the distance to the wall

According to (7.6) the mixing length l is related to the distance y by :

$$l = \kappa \, y$$

with κ = von Karman's constant (0.4)

which expresses the notion that the turbulent eddies grow linearly with their distance to the wall.

With constant shear stress τ_w over the turbulent boundary layer we have :

$$\tau = \rho \, (\kappa \, y \, \partial u/\partial y)^2 \approx \tau_w$$

or :

$$\partial u/\partial y = \frac{\sqrt{(\tau_w/\rho)}}{\kappa\, y} \qquad (8.6)$$

The expression $\sqrt{(\tau_w/\rho)}$ is commonly designated the wall friction velocity u*.

The solution of the differential equation (8.6) gives the wall function :

$$u(y) = \frac{u^*}{\kappa} \ln(y/y_0) \qquad (8.7)$$

where y_0 is an integration constant empirically related to the wall friction coefficient by :

$$y_0 = k_n / 33$$

with k_n = roughness parameter of Nikuradse

8.3.3 The Slip Formulation

From (8.7) we can express the wall shear stress τ_w (i.e. the shear stress that the wall imposes on the fluid body) as function of the velocity in the first grid point away from the wall y_1 :

$$\tau_w = \rho\, \kappa^2 u(y_1)^2 / \ln^2(y_1/y_0) \qquad (8.8)$$

In this first point y_1 the influence of the wall is "felt" by the diffusion term of the Navier-Stokes equations :

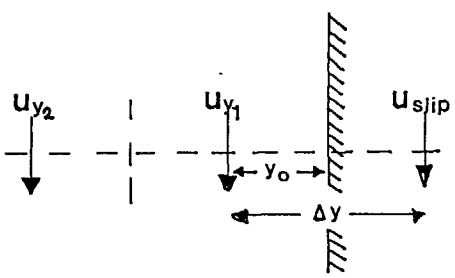
$$\partial(\tau_{xy}/\rho)/\partial y = \partial(\nu_t \partial u/\partial y)/\partial y$$

or in finite differences :

$$\nu_{t2}\frac{u(y_2)-u(y_1)}{\Delta y} - \nu_{t1}\frac{u(y_1)-u(slip)}{\Delta y} \qquad (8.9)$$

where y_2 = the second grid point away from the wall

See also fig. 8.4.



slip formulation
fig. 8.4

In order for the wall friction to exert the proper shear stress on the fluid we must equate the rightmost term in (8.9) to the wall shear stress (8.8) divided by ρ :

$$\nu_t \frac{u(y_1)-u(slip)}{\Delta y} = \kappa^2\, u(y_1)^2/\ln^2(y_1/y_o) \qquad (8.10)$$

or :

$$u_{slip} = (1 - a_{sl}\, u_{(y_1)})\, u_{(y_1)} \qquad (8.11)$$

$$\text{with } a_{sl} = \frac{\kappa^2 \, \Delta y}{\nu_t \ln^2(y_1/y_0)} \; [m^{-1}s]$$

Remark :

The value of ν_t in (8.11) neutralizes the value of the turbulent viscosity in the apropriate diffussion term. In this way the wall shear stress has its proper value even if the coefficient of turbulent viscosity, as used in the equations, is not physically correct near the wall.

If we assume that locally the velocity varies only normal to the wall, we can translate the theory developed above to the more general case of a solid boundary that intersects the grid at an arbitrary angle. Then, the shear stress τ_w becomes a vector parallel to the wall. It will have the direction of the local velocity (whose component normal to the wall is zero) and (8.8) now must be written as :

$$\tau_w = \rho \, \kappa^2 / \ln^2(d_1/d_0) \; |V_1| \; V_1$$

(8.10) becomes :

$$\nu_t \frac{V_1 - V_0}{\Delta n} = \kappa^2 \; |V_1| / \ln^2(d_1/d_0) \; V_1$$

where V_1 = the velocity vector at a distance d_1 from the wall

and n = the coordinate normal to the boundary

Expressing Δn in its direction cosines the slip velocity V_0 is given by :

$$V_{slip} = (1 - a_{sl}|V_1|) \cdot V_1 \qquad (8.12)$$

with $(..)_1$ the velocity in the first "live" grid point away from the wall

$$\text{and } a_{sl} = \frac{\kappa^2}{\nu_t \ln^2(d_1/d_0)} \Delta x_i \cos \gamma_i \qquad (8.13)$$

where $\Delta x_i = \Delta x, \Delta y, \Delta z$
and γ_i = the angle between the normal and the corresponding coordinate direction.

In formula (8.13) the value of a_{sl} depends on d_1 and on the direction of differencing, i.e. the value of j in the tensor element τ_{ij}, that is represented. While d_1 generally can be replaced by its mean value, the factor $\Delta x_i \cos \gamma_i$ must be given its correct value to prevent the influence of the "staircase" boundary on the flow pattern, especially if a substantial wall friction is specified on this boundary.

The use of (8.12) and (8.13) in the computational scheme will be discussed in section 8.4.2.

8.3.4 Vorticity Defect of the Wall Function Formulation

In many computational methods wall functions are used to model the turbulent boundary layer. It is useful to point out a possible source of error connected to the above

described use of wall functions or partial slip boundary conditions.

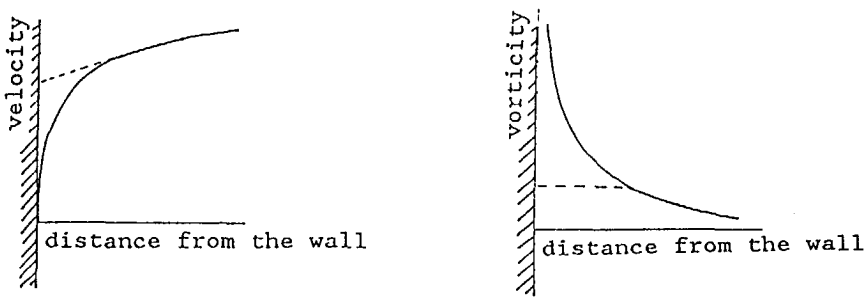
To illustrate the problem we will consider a turbulent flow parallel to a solid wall. Note that in the following we do not consider the turbulent motion of the fluid but the averaged values of the flow variables as used in the computational method.

In the neighbourhood of the wall the vorticity of the flow can be expressed as :

$$\omega = \partial u/\partial y$$

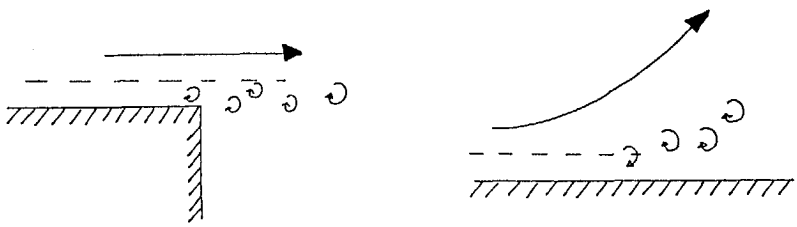
with u and y as in fig. 8.4

In fig. 8.5 the velocity- and vorticity distribution as function of the distance y from the wall is indicated.



a: velocity b: vorticity

fig. 8.5



a: backstep b: separating flow

fig. 8.6

As seen in fig. 8.5b the vorticity decreases sharply from a high value at the wall to a low value further away from the wall. The use of a wall function to generate a boundary condition is equivalent to the approximation of the functions in fig. 8.5 by the dashed lines in the region of the boundary layer. Thus, part of the wall vorticity is neglected. In the situation considered so far this is no problem, but if the main flow separates from the wall, for instance as in the situations depicted in fig. 8.6a and b, the neglected amount of vorticity should be convected into the computational domain. How this should be done computationally is a complicated matter beyond the scope of this thesis, but the modeller should be aware of the possible error made by the use of the boundary conditions as described here.

8.4 Implementation in the Computational Scheme

8.2.1.Porosity

As noted in section 8.2.3 application of the porosity method involves the storage of the volume and surface area's of the grid cells adjacent to the solid walls in the model. It is possible to design a complicated addressing system and store the necessary data for the boundary cells only.

We chose a simpler system where the data for all cells are kept in storage, but in an efficient manner using one single word of storage per grid cell. Per grid cell four integer values (volume 0-999, 3 area's 0-99) are stored. A special code is used to indicate a cell that is completely open (or completely closed) in which case the formulation as given in (8.5) can be simplified.

A separate program generates the necessary geometrical data from user specified contour planes describing the structure(s) to be modelled. See appendix A.

8.4.2 Viscosity and Wall Function

The viscosity coefficient in (4.1) is used to describe the turbulent viscosity of the flow. Thus, the possibility must exist to specify a spatially varying value.

The wall friction also may differ for different sections of the solid walls in the model.

To accomodate these needs a separate array is used to store both the viscosity coefficients and, for grid cells near the boundaries, the slip factors (8.13). Because a computational point may be connected to other "active" points

as well as to the boundary, per grid cell three values, representing either a viscosity coefficient or a slipfactor, must be stored. Using the same storage technique as described in the preceding section for the grid data, these three values are stored in one memory word.

While the value of the turbulent viscosity coefficient, at least theoretically, is independent of direction, the value of the slipfactor a_{sl} that must be specified is different for each coordinate direction as explained in section 8.3.3. However, in the applications described in chapter 9, notably for the oblique walls in the second example, this difference in direction is not made, because the present form of the grid-generating program (appendix A) cannot as yet generate the necessary directional information.

If the value of the viscosity coefficient depends on the local velocity gradient (for instance, if one uses the mixing length formula (7.5)) the array containing those values is updated a few times during the computation.

Including the arrays mentioned in this and the preceding section, seven arrays in all are needed during the flow computation : four for the primary flow variables u, v, w and h, one for the geometrcal data, one for the viscosity/slip coefficients and an auxilary array. This last array permits a faster computational scheme, but it can be omitted if storage capacity is a problem.

8.4.3 Free Surface

Most applications, for which the computational method decribed in this thesis was developed, involve open channel

flow. This means that the method must provide a way to simulate a free surface. Formally this surface is to be determined during the computation, involving a spatially varying grid system or a technique, known as MAC (marker and cell, Harlow and Welch,1965). If the variation in surface elevation is small a simpler simulation of the free surface is often permitted: the so-called rigid lid approximation. One replaces the free surface by a rigid lid on which a varying pressure is permitted. The boundary condition given is that of zero normal flow and free slip.

The method of artificial compressibility permits another simple solution of the problem of simulating the free surface. If in the surface layer of grid cells the continuity equation according to (4.1) is replaced by its SWE counterpart (2.11), a top layer with a variable vertical dimension is created. Variations of the water level less than the vertical grid size can thus be represented in the difference scheme. However, the gain in accuracy is small because of the small amount of fluid mass involved. Moreover the advantage of a linear continuity equation is lost. In the applications in chapter 9 the rigid lid approximation is used in all cases.

9. APPLICATIONS

9.1 Introduction

9.1.1 Short Description of Numerical Examples

In this chapter we present three applications of the computational method developed in the foregoing text. As a first example a theoretical configuration is treated : a harbor situated in the outer bend of a river. This problem is used to demonstrate a possible difference between the 2-D SWE, and the full 3-D solution. The artificial compressibility method is very easily adapted to a 2-D SWE problem, so the comparison can be done with one and the same computer program. For simplicity we use a constant depth in both the 2-D and the 3-D computation, although a variable bathymetry could be accomodated with only a few changes in the 2-D version of the program. A rigid lid approximation (i.e. the free surface is simulated by a fixed, free slip boundary) is used in both computations.

The second and third applications concern real structures. They represent each a different class of problems. In the second application, where the flow through the intake structure of a small hydro-power station is computed, the flow accelerates and no back flow occurs. In the third example the flow decelerates and regions of back flow are present. This application numerically simulates the flow through a gate in the proposed storm surge barrier in the Rotterdam Waterway. Both computations could be compared with measurements from physical scale models.

9.1.2 Determination of Computation Parameters

Computation parameters can be divided in physical parameters, i.e. the dimensions of the structure to be modelled, the velocity V and the eddy or turbulent viscosity ν_t, and the numerical parameters, i.e. the grid sizes Δx, Δy, Δz, the time step Δt and the propagation speed of the artificial pressure wave $\sqrt{(g/\alpha)}$. These parameters can be combined into several important nondimensional numbers that determine the behaviour of the numerical solution.

From the physical parameters we obtain the turbulent Reynolds number :

$$Re_t = V\ L\ /\nu_t$$

where L is a characteristic length

The classical Reynolds number determines the ratio between the inertial and viscous forces acting on the fluid. Similarly the turbulent Reynolds number determines the ratio between the inertial forces in terms of the mean flow velocities and the forces generated by the eddy viscosity, i.e. the representation of the turbulent stresses.

Other combinations result in :

a: The Courant numbers : $u\Delta t/\Delta x$, $v\Delta t/\Delta y$, $w\Delta t/\Delta z$

This Courant number is based on the flow velocity. As explained in chapter 6 the stability of our numerical scheme depends on the 1-D Courant number rather than on the full 3-D Courant number. In our computations we have to deal with the maximum of

the three Courant numbers cited above.

b: The Courant number : $\sqrt{(g/\alpha)}\ \Delta t/\Delta s$

with Δs = the smallest grid size

This number is associated with the artificial pressure wave. Because this wave travels in all directions the smallest grid size is the determining one.

c: The diffusion dominated stability limit :
(hereafter called "diffusion limit")

$$2\nu_t\left(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}+\frac{1}{\Delta z^2}\right)\Delta t$$

Here we must take the full 3-D form.

d: The convection dominated stability numbers :
(hereafter called "convection parameter")

$$u^2\Delta t/2\nu_t,\ v^2\Delta t/2\nu_t,\ w^2\Delta t/2\nu_t$$

Again the maximum of these three numbers is the determining entity.

Because the convection-diffusion terms of the Navier-Stokes equations are treated explicitly in our numerical scheme, the Courant number based on the flow velocities and the diffusion limit should always be smaller than unity, because otherwise the computation will be unstable for every Fourier component of the solution. While the latter may be close to unity, the Courant number should always stay well

below unity. In our experiments its value was about 0.3 or less.

The Courant number based on the pressure wave propagation velocity may take values greater than unity, because this part of the computation is performed implicitly. As explained in section 4.5 the propagation velocity should not be chosen too large.

According to the von Neumann stability analysis (see section 5.4) the convection parameter should be less than unity. However, it was found in the experiments that stable computations were possible with the maximum of the three numbers taking values of 2 to 3. The reason for this may be the fact that this parameter is associated with the lower range of wave numbers.

In some cases another number is important as well : the cell Reynold number :

$$Re_c = V\Delta s/\nu_t$$

where V is the convecting velocity in the direction of grid size Δs.

As mentioned in section 8.1 this number plays a part in the behaviour of the solution near the outflow boundary. If the cell Reynolds number attains values larger than 2, wiggles will appear in the solution. This problem is eliminated by specifying a zero second derivative boundary condition at the outflow boundary. However, in some cases the wiggles associated with the cell Reynolds limit occur at other locations in the flow field. In some situations, e.g. near a stagnation point, a high velocity gradient may exist which

cannot be resolved by the spatial grid. An example of such a situation is found in the first application (section 9.2.4).

In a practical computation one starts by determining the grid sizes. If we disregard the problem just mentioned, the choice of the grid size is governed by two factors : the smallest eddy to be resolved, or the smallest (double) radius of curvature that must be followed by the flow, and the available computer memory space.

An estimation of the smallest resolvable eddy can be made on the basis of the spatial error mentioned in section 4.2.4. If we assume that this eddy is built up from Fourier components with a wave length equal to the diameter of the eddy, and if we require that the spatial error does not exceed 10%, it can be calculated from the theory in section 4.2.4 that there should be at least 8 grid lengths across an eddy.

Having determined the grid size, the time step is chosen on the basis of the non dimensional parameters mentioned under a, b, c and d above.

For the time step of the ADI part of the computation a value can be chosen on the basis of the Courant number (b) associated with the propagation of the artificial pressure wave. Because the ADI part of the computation is unconditionally stable this Courant number may exceed unity, but for reasons mentioned in section 4.5 it should not be taken too high. Its value is not very critical. In our experiments we give the ADI Courant number a value of 2 to 3.

The propagation speed $\sqrt{(g/\alpha)}$ is taken up to 5 times the maximum flow velocity. To limit the velocities in the transient phase, a computation is generally started with a higher propagation speed. In this way the initial pressure disturbance is spread rapidly over the computational domain.

If the model is somewhat converged, the propagation speed, or rather the compressibility factor α, is set to its final value.

9.2 Harbor in Outer River Bend

9.2.1 Grid System

In this and the following sections we will use the words-:"north","south","east" and "west" to indicate the four sides in the horizontal plane of the computational domain. In the vertical direction the words "surface" and "bottom" are used.

The basic grid system consists of a block of 40 X 40 X 11 grid cells. Normal to each cell face a velocity component is located in the center of that face, while the pressure, or rather the piezometric height, is situated in the center of each cell. In each cell the pressure in the center and the velocity components on the southern, eastern and downward face are associated with each other, i.e. they are indicated with the same values of the grid numbers i, j and k. The cell (1,1,1) forms the upper north-west corner of the block, while u(i), v(j) and w(k) point south-, east- and downward, respectively. See also fig. 4.1. The outer layer on each side of the computational block (including the upper surface) consists of dummy cells. They are used to specify the in- and outflow boundaries. Thus, we have a block of 38 x 38 x 9 possible active cells. By closing off (parts of) these cells the desired geometry is obtained. By using the porosity formulation mentioned in section 8.2, cells can be partially blocked and the boundaries do not have to coincide with grid planes.

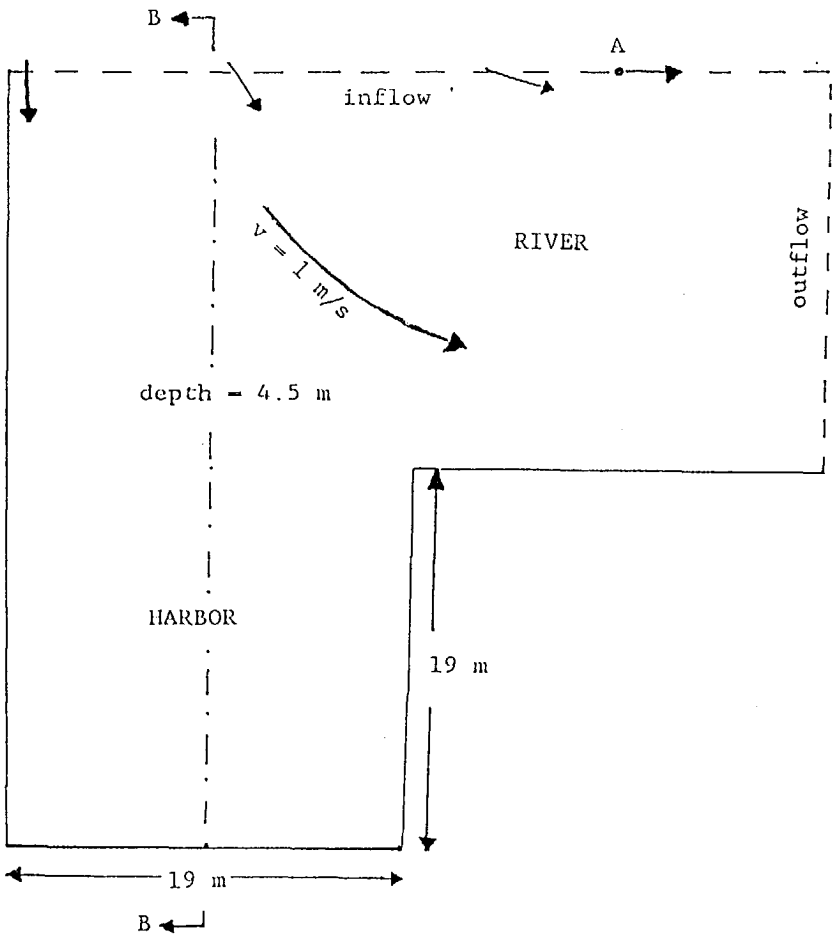


fig. 9.1 : Harbor in river bend

In the model we are considering in this section the desired geometry is obtained by closing a block of 20 x 20 x 11 cells in the south-east corner, which results in a geometry that can be considered as a square basin situated at the outer bend of a channel. See fig. 9.1. For simplicity, a constant depth is assumed.

For both the 2-D and 3-D model the horizontal grid size was chosen as $\Delta x = \Delta y = 1$ m., resulting in a basin with dimensions 19 x 19 m. The depth is 4.5 m.

9.2.2 The 2-D Model

The 2-D and 3-D computation work both with the grid system described in the preceding section, but in the 2-D computation only one layer of grid cells is used. Because the uppermost layer is used to specify the free slip conditions representing the free surface (see also section 8.4.3) the second layer is the active one. In the 2-D computation the third layer is used to specify the bottom friction. This is done by using the "vertical" viscosity terms in the u- and v- equations to simulate the bottom friction by means of the slip formulation discussed in section 8.3.3. This formulation allows a quadratic dependence of the bottom friction on the velocity conform the SWE friction terms.

The SWE bottom friction terms usually are given as :

$$g \frac{u \sqrt{(u^2+v^2)}}{C^2 D} \qquad \text{in the x-direction and}$$

$$g \frac{v \sqrt{(u^2+v^2)}}{C^2 D} \qquad \text{in the y-direction}$$

where g = acceleration of gravity
C = friction coefficient of Chezy
D = depth

The finite difference approximations of the viscosity terms in the u- and v-equation of the system (4.1) for a free slip surface condition and a partial slip bottom condition according to formula (8.12) result in :

$$\nu_t \, a_{sl} \frac{u \sqrt{(u^2+v^2)}}{\Delta z^2} \qquad \text{in the x-direction and}$$

$$\nu_t \, a_{sl} \frac{v \sqrt{(u^2+v^2)}}{\Delta z^2} \qquad \text{in the y-direction.} \qquad (9.1)$$

By taking

$$a_{sl} = g\,\Delta z^2/(C^2 D\,\nu_t) \qquad (9.2)$$

the standard 3-D formulation can be used for the 2-D computation. Because we use a rigid lid approximation, the variation in water level does not appear in the continuity equation, which for constant depth is then reduced to $\nabla.V = 0$. Thus, Δz enters the 2-D formulation in (9.2) only. It can be treated as a numerical factor, relating a_{sl} to ν_t.

The bottom friction in the model is determined by the depth D and the value of the Chezy coefficient C. In the present computation the following values where used :

$D = 4.5$ m
$C = 50\ m^{1/2}/s$

The remaining computation parameters are listed below:

- Inflow at north boundary :

$$u = V_m \cos\left(\frac{j-2}{28} \times \frac{\pi}{2}\right) \quad \text{for } j=2,30 \quad ; \quad u=0 \quad \text{for } j=31,40$$
$$v = V_m \sin\left(\frac{j-2}{28} \times \frac{\pi}{2}\right) \quad \text{for } j=2,30 \quad ; \quad v=V_m \quad \text{for } j=31,40$$

with V_m = depth averaged velocity = 1. m/s
and j = grid number (west to east)

This results in an inflow with a direction varying linearly from due south at the west boundary to due east eastwards of point A in fig. 9.1.

- Outflow at east boundary :

  a uniform pressure distribution h=0

  $\partial^2 u/\partial x^2 = 0$ and $\partial^2 v/\partial x^2 = 0$

- Horizontal viscosity coefficient :

$$\nu_t = 0.025 \text{ m}^2/\text{s}$$

  This value is based on formula (7.9). The value of u* in this formula is obtained according to :

$$u^* = \sqrt{g} \cdot V_m/C = 0.063 \text{ m/s} \qquad (9.3)$$

- Horizontal slip factor $= 0.15 \text{ m}^{-1}\text{s}$

  This value simulates an arbitrary roughness of the river banks.

- Vertical slip factor

  To obtain this value we must chose an arbitrary value for Δz. The value for a_{sl} then follows from (9.2). In this case we took :

$$\Delta z = 2.4 \text{ m}, \quad a_{sl} = 0.2 \text{ m}^{-1}\text{s}$$

- Time step for the convection-diffusion part of the computation :

$$\Delta t_{cd} = 0.125 \text{ s}$$

  This choice results in the following values of the nondimensional computation parameters :

Maximum Courant number : $\frac{u\,\Delta t}{\Delta x} = \frac{1.\ x\ 0.125}{1.} = 0.125$

Diffusion limit : $2\nu_t(1/\Delta x^2+1/\Delta y^2)\Delta t =$
$0.05\ x\ (1/1+1/1)\ x\ 0.125 = 0.0125$

Convection parameter : $u^2/2\nu_t\Delta t = 1/0.05\ x\ 0.125 = 2.5$

The time step is determined by the convection parameter.

- Time step for the ADI part of the computation :

$$\Delta t_{adi} = 0.5\ s$$

- Compressibility factor $\alpha = 0.2\ m^{-1}$

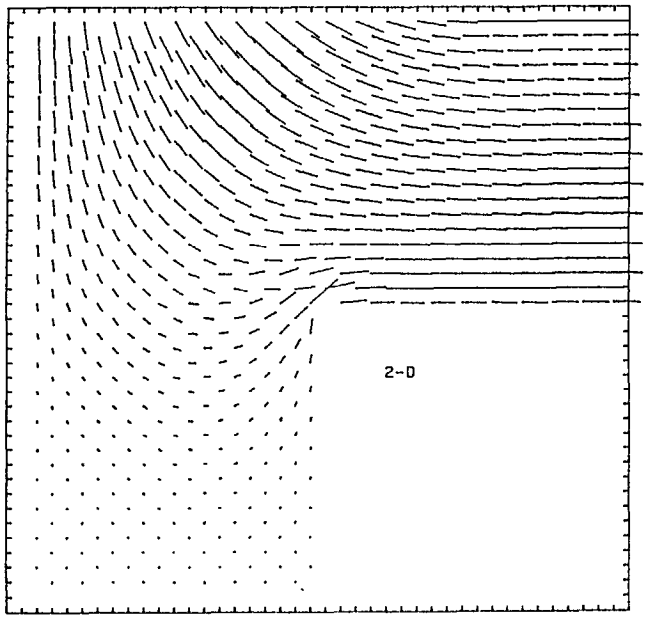
The last two figures result in a Courant number for the ADI process :

$$Courant_{adi} = 2.25$$

with a propagation speed of the artificial pressure wave of 4.5 m/s

The result of the 2-D computation is given in fig. 9.2. The velocity vectors in this and the following plots are drawn in the cell center, by averaging the velocity components on opposite cell faces. A velocity vector is drawn for every other cell in the horizontal plane.

From the figure it can be observed that the primary flow penetrates the whole of the basin. No eddy is generated.



2-D depth averaged computation

fig. 9.2

9.2.3 The 3-D Model

The 3-D model employs the same grid system as the 2-D model, except that now all layers are used. The vertical grid size regains its proper meaning :

$$\Delta z = 0.5 \text{ m}$$

Because there are 9 active grid cells in the vertical this results in a depth of 4.5 m., the same as in the 2-D case.

To create a flow situation that is as much as possible identical to the situation in the 2-D computation, the same in- and outflow conditions are chosen. Of course, in the 3-D model a vertical distribution of the inflow must be given. This distribution is obtained from the logarithmic formula:

$$V(z) = v^*/\kappa \ \ln(z/z_0)$$

with $v^* = 0.063$ m/s as in the 2-D model
and $z_0 = 0.0045$ m.

The value of z_0 is related to the Chezy coefficient used in the 2-D model with the formula :

$$C = (\sqrt{g}/\kappa) \ \ln(D/ez_0)$$

The remaining parameters are chosen identical to the 2-D case :

$$\nu_t = 0.025 \ m^2/s$$

$$a_{sl}(\text{bottom}) = 0.2 \ m^{-1}s$$

$$a_{sl}(\text{side walls}) = 0.15 \ m^{-1}s$$

$$\Delta t_{cd} = 0.125 \ s$$

$$\Delta t_{adi} = 0.5 \ s$$

$$\alpha = 0.2 \ m^{-1}$$

Because now three dimensions are involved and because the surface velocity is slightly higher than the mean velocity used in the 2D computation, the non dimensional parameters are different. The determining number, i.e. the convection parameter becomes :

$$V^2\Delta t/2\nu_t = 1.08^2 \times 0.125 \ / \ 0.05 = 2.9$$

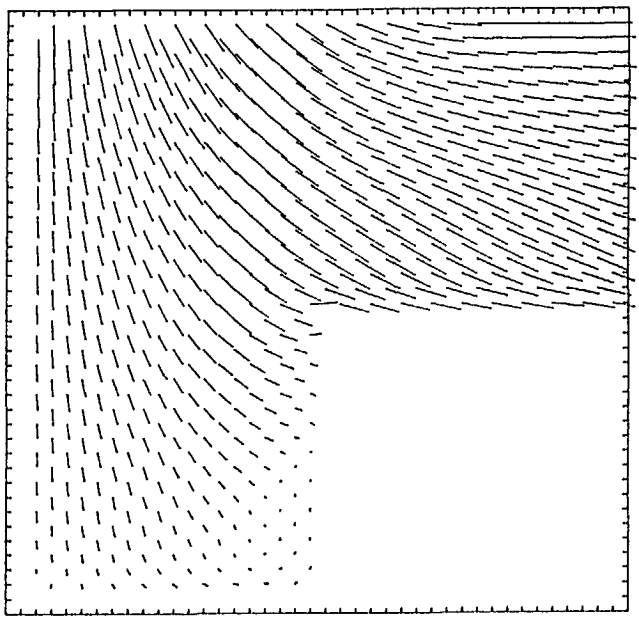


fig. 9.3 a: surface velocities

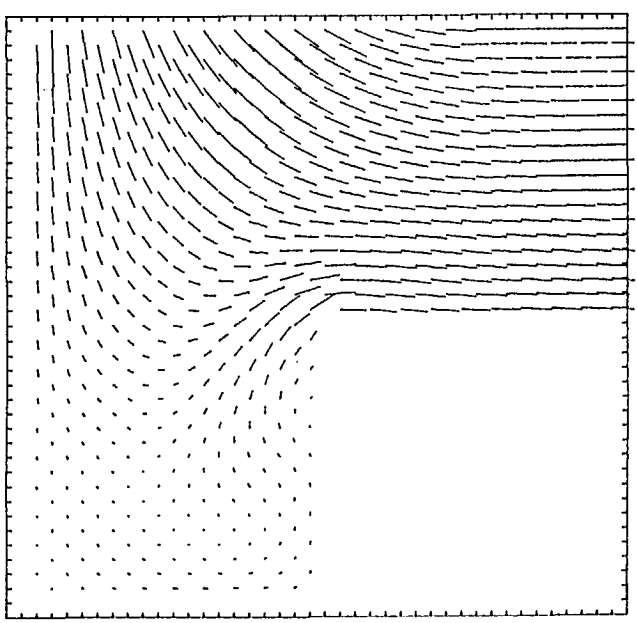


fig. 9.3 b: mid depth velocities

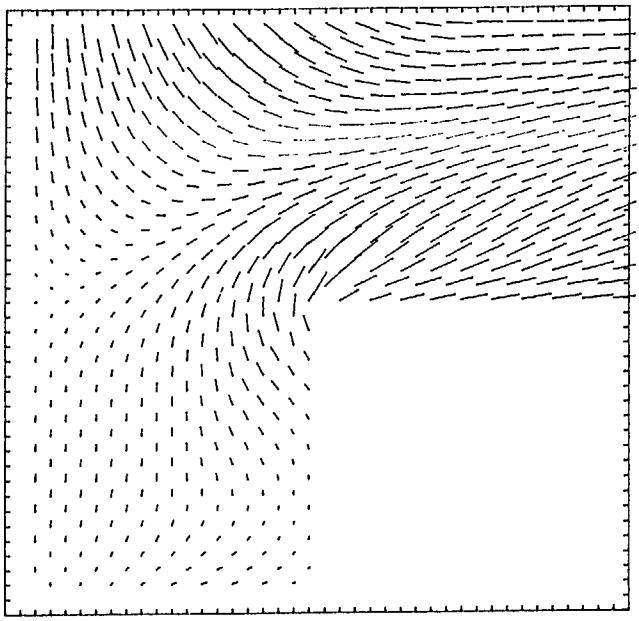


fig. 9.3 c: bottom velocities

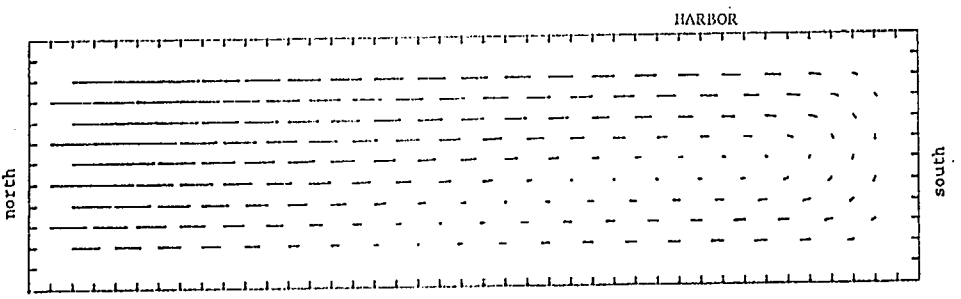


fig. 9.4 a: vertical cross section west wall

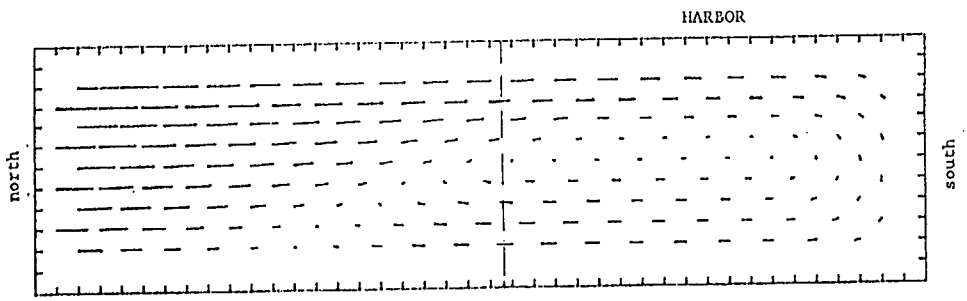


fig. 9.4 b: vertical cross section B-B

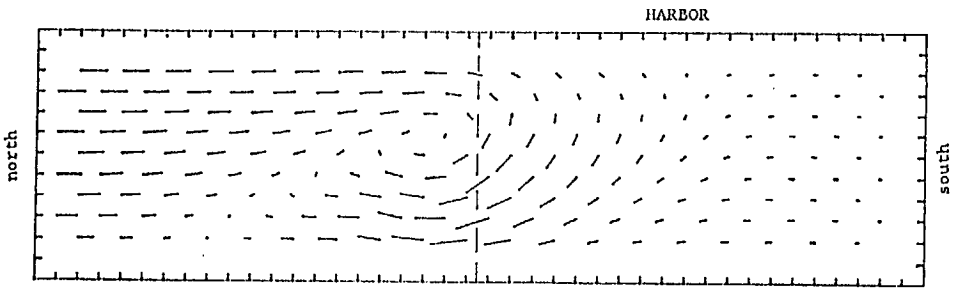


fig. 9.4 c; vertical cross section east wall

The result of the 3-D computation is given in fig. 9.3a,b,c. In this figure the horizontal velocity is plotted on three depth levels : 0.25 m(a), 2.25 m(b) and 3.75 m(c) from the surface. Fig. 9.4 gives the velocity vectors in three vertical cross sections: one along the west boundary (fig. 9.4a), one along the line B-B as indicated in fig. 9.1 (fig.9.4b) and one along the east boundary of the harbor (fig. 9.4c). From this figures a strong helical movement is visible.

About 600 ADI iterations and 2400 convection-diffusion iterations were needed for convergence, i.e. to reach a steady state. Expressed in cpu-time this means :

2.5 hours on IBM 3083
17 hours on Harris 1000

On the IBM machine, using 4-byte words, the computer program (without arrays) demands 250 Kbyte of memory. The arrays containing the flow variables and parameters take another 500 Kbyte for the dimensions used in this case.

Remark :

In the method of artificial compressibility, the time axis of the computation has no physical meaning, but some relation with the physics of the flow exists : The computational time (i.e. number of iterations x Δt) must at least exceed the time needed for the boundary conditions to spread their influence through the computational domain. See section 4.5.

9.2.4 Discussion

Comparing the results from the 2-D and the 3-D computation the conclusion can be drawn that a calculation of sediment transport based on the 2-D computation would differ considerably from one that is based on the 3-D flow field. In the 3-D result almost no flow enters the harbor in the lower half of the water body where most of the sediment is concentrated. Near the bottom a strong outflow exists at the east side of the entrance. The difference is caused by a strong vertical eddy as shown in fig. 9.4.

In order to compare the results of the 2- and 3-D computation, we computed the depth average of the 3-D computation. The result of this averaging is given in fig. 9.5. There is an interesting difference between the 2-D and the averaged 3-D result, which becomes visible when the 2-D field is subtracted from the 3-D averaged field. A weak horizontal eddy (the velocity vectors in fig. 9.5 are

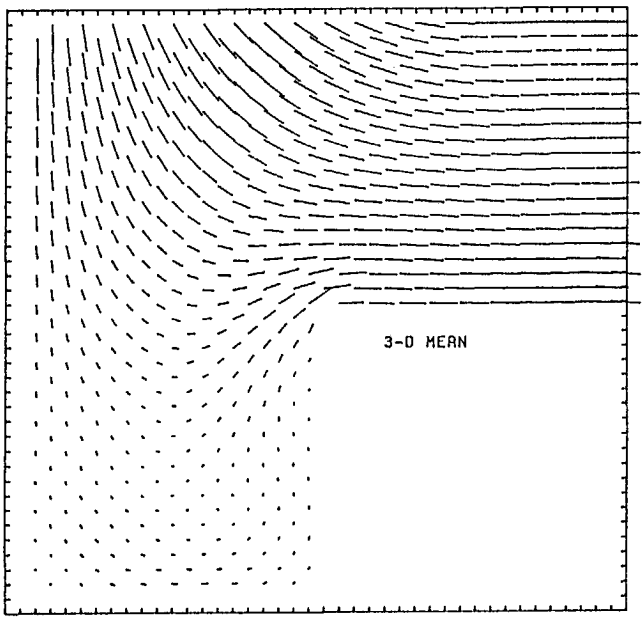


fig. 9.5 Depth averaged 3-D computation

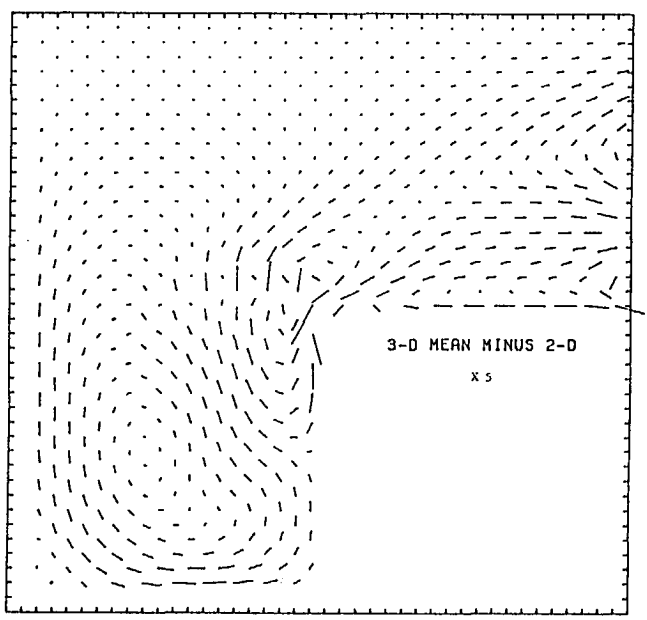


fig. 9.6 Difference 3-D(averaged) - 2-D (X5)

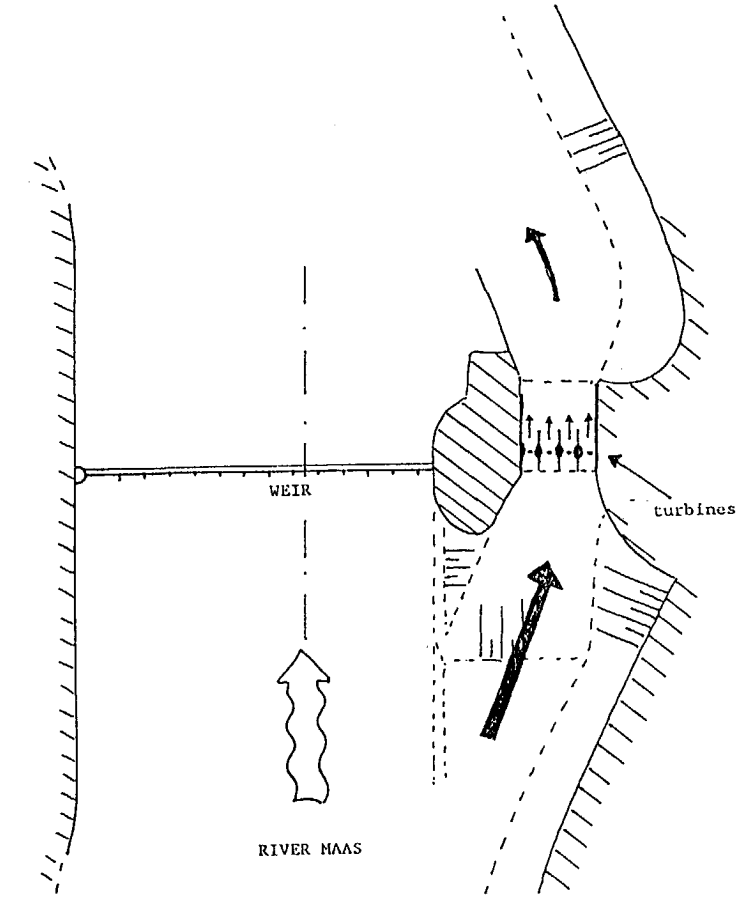
multiplied by 5) shows up, indicating that in the 3-D computation the penetration of the primary flow is stronger than in the 2-D computation.

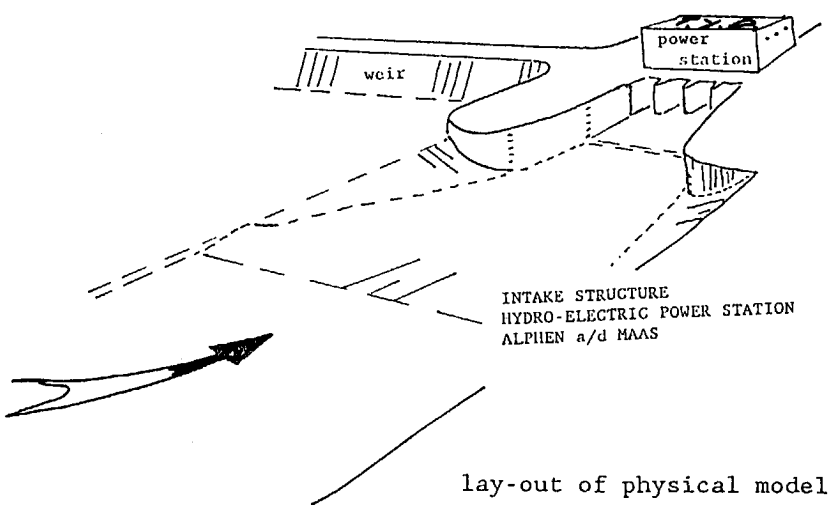
Some trouble was experienced near the east side of the harbor entrance. In earlier computations, wiggles appeared here during the transient stage of the computation because a stagnation point is formed where the flow hits the wall. Upstream of this point the cell Reynolds number limit probably is violated. This kind of wiggles cannot be reduced by decreasing the time step, as they are part of the exact solution of the difference equations. The wiggles were transported slowly towards the outflow boundary but did not disappear in the steady solution. The problem was overcome by locally increasing the value of the eddy viscosity in the vicinity of the stagnation point by 20%. In fig. 9.6 it can be seen that near the east side of the entrance, where the flow leaves the harbor a reduced amount of wiggling still exists.

9.3 Intake Structure Hydro-Power Station

9.2.1 Introduction

The purpose of the numerical experiment described in this section is twofold. Firstly, we want to examine how well the model reproduces the measured flow pattern. Because we aim at the use of this model on small computer installations, acceptable results should be obtained without an excessive amount of grid points. Secondly, the possibility must exist to model complicated configurations without too much effort. This means that the calculation of the volumes and areas of the boundary cells in connection with the porosity method





lay-out of physical model

fig. 9.7

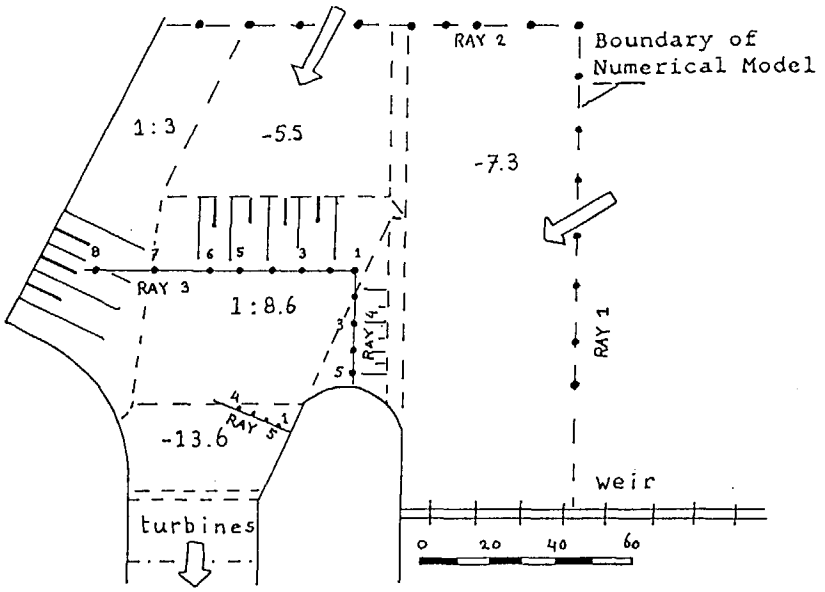


fig. 9.8 Numerical model

should be done automatically. A special "grid generating" program was developed for this purpose.

In the following application both aspects could be tested.

In the past several weirs have been constructed in the river Maas, mainly to control the water levels in periods of low discharge. Lately small hydro-power stations are being build parallel to these weirs. Because of the small difference in water level an optimal design of the intake structure is important in order to make a maximum use of the available head.

To test the numerical model a comparison is made with a physical model of the intake structure designed for the power station next to the weir near Alphen a/d Maas. This physical model (scale 1 : 40, undistorted) was available at Delft Hydraulics and measurements could be obtained for the purpose of the validation of the numerical model. A lay-out

of the physical model is shown in fig. 9.7, while in fig. 9.8 the part of this model that was simulated numerically is sketched. In this figure the rays along which the measurements were taken are indicated. Ray 1 and 2 coincide with the boundaries of the numerical model.

From the physical model velocity measurements (absolute value and horizontal direction averaged over 2 minutes) were obtained with a micro propeller on several depths per location. The numerical model covers only a small part of the physical model which incorporates upstream and downstream river sections. In a regular numerical design process a larger 2-D model could be used to furnish the boundary values of the 3-D model. In the present case these boundary values were taken from the measurements along the outer rays (ray 1 and 2) in fig. 9.8. These rays thus form the north and east boundary of the model, which covers an area of 160 x 160 m.

The measurements along the rays 3, 4 and 5 (fig. 9.8) are compared to the results of the numerical computation.

In the following we mean by "inflow" and "outflow" in in- and outflow of the numerical model. The outflow of the model is in the real design the inlet to the turbines. This inlet is divided in four equal sections each leading to a separate turbine. This part of the design is not incorporated in the numerical model.

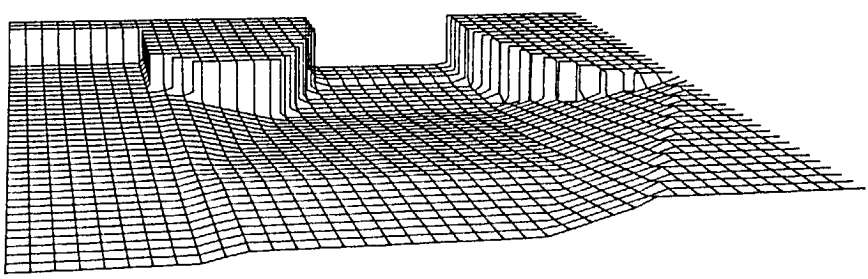
9.3.2 Grid System

The horizontal grid size was determined on the basis of the curvature of the pier at the east side of the outflow. The rule of 8 grid lengths given in section 9.1.2 leads to a grid size of about 4 m. Because of the large difference in

depth (minimum depth: 6.5 m, maximum: 13.5 m.) the vertical grid size was taken 1.5 m. in order to have a minimal amount of vertical points in the shallowest region. The number of vertical points thus becomes 11, including the dummy cells.

As the total number of grid points was limited by the memory capacity of the available mini computer (Harris 1000), we ended up with a basic rectangular block of grid points consisting of 40 x 40 x 11 points. With the horizontal grid size of 4 m. this leads to an model area of 160 x 160 m.

The resulting schematisation is rather coarse but the purpose of this test is also to investigate to what extent the methods discussed in the last chapter can compensate for this coarseness.



bottom topography
fig. 9.9

Within the basic block of grid points the actual flow domain is bounded by user specified planes. A special program calculates the intersections with the grid lines and the "porosity factors" for each grid cell that is intersected. Per cell a wetted volume and three wetted areas are

determined. Details of this procedure are given in appendix A. The perspective view of this schematisation is shown in fig. 9.9.

9.3.3 Boundary Conditions and Computation Parameters

The horizontal outline of the numerical model is given in fig. 9.8. On the north and east boundary the velocity components are specified according to the measurements in the physical model. Measurements were available on every fourth grid point along the north and east boundary, each for two depth levels. The remaining boundary values were obtained by linear interpolating. When necessary, data were extrapolated towards the bottom using a roughly logarithmic fitting.

A locally varying viscosity coefficient was used based on formula (7.9) and the local depth. In each horizontal location its value was constant over the vertical. Thus :

$$\nu_t = 0.08\ u^*\ D$$

with D = local depth
and $u^* = 0.063$ m/s

Further computation parameters were :

- Outflow :

  a uniform pressure distribution h=0
  $\partial^2 u/\partial x^2 = \partial^2 v/\partial x^2 = \partial^2 w/\partial x^2 = 0$

  The assumption of a uniform pressure distribution can be justified by the reasoning that the design

is aimed at an even distribution of the flow across the turbines. If this goal is reached, the pressure distribution will be more or less hydrostatic.

- Wall roughness :

  Because of the spatially varying ν_t, the value of a_{sl} depends on the local viscosity according to formula (8.13) with d_0=0.0045 m.

- Time step :

  The time step was chosen on the basis of the convection parameter. It was found that, to avoid trouble near the boundaries, this limit should be chosen somewhat lower than with the first application. The reason is that the use of the porosity method has a destabilizing effect on the computation. Stable computations could be performed with :

$$\Delta t = 0.2 \text{ s}$$

With these values we get the following non dimensional parameters for the convection-diffusion part of the computation :

Courant number : $u\Delta t/\Delta x$ or $v\Delta t/\Delta y = \frac{1 \times 0.2}{4.} = 0.05$

Diffusion limit : $2\nu_t(1/16+1/16+1/2.25)\ 0.2 = 0.0072$

This value is based on the lowest value of ν_t, which occurs at the location of the smallest depth.

Convection parameter : $u^2/2\nu_t = 1.54$

Here we take the value of ν_t, where the maximum ve-

locity occurs. This is around ray 5 in the outflow channel.

The compressibility factor is chosen as 0.3, resulting in a propagation speed of 5.7 m/s. The time step for the ADI part of the computation is :

$$\Delta t = 0.4 \text{ s}$$

resulting in a Courant number associated with the pressure wave of :

$$\text{Courant}_{adi} = \frac{5.7 \times 0.4}{1.5} = 1.5$$

based on the smallest grid size (Δz = 1.5 m).

About 800 ADI iterations and 1600 convection-diffusion iterations were necessary to reach steady state.

9.3.4 Results and Comparison to Measurements

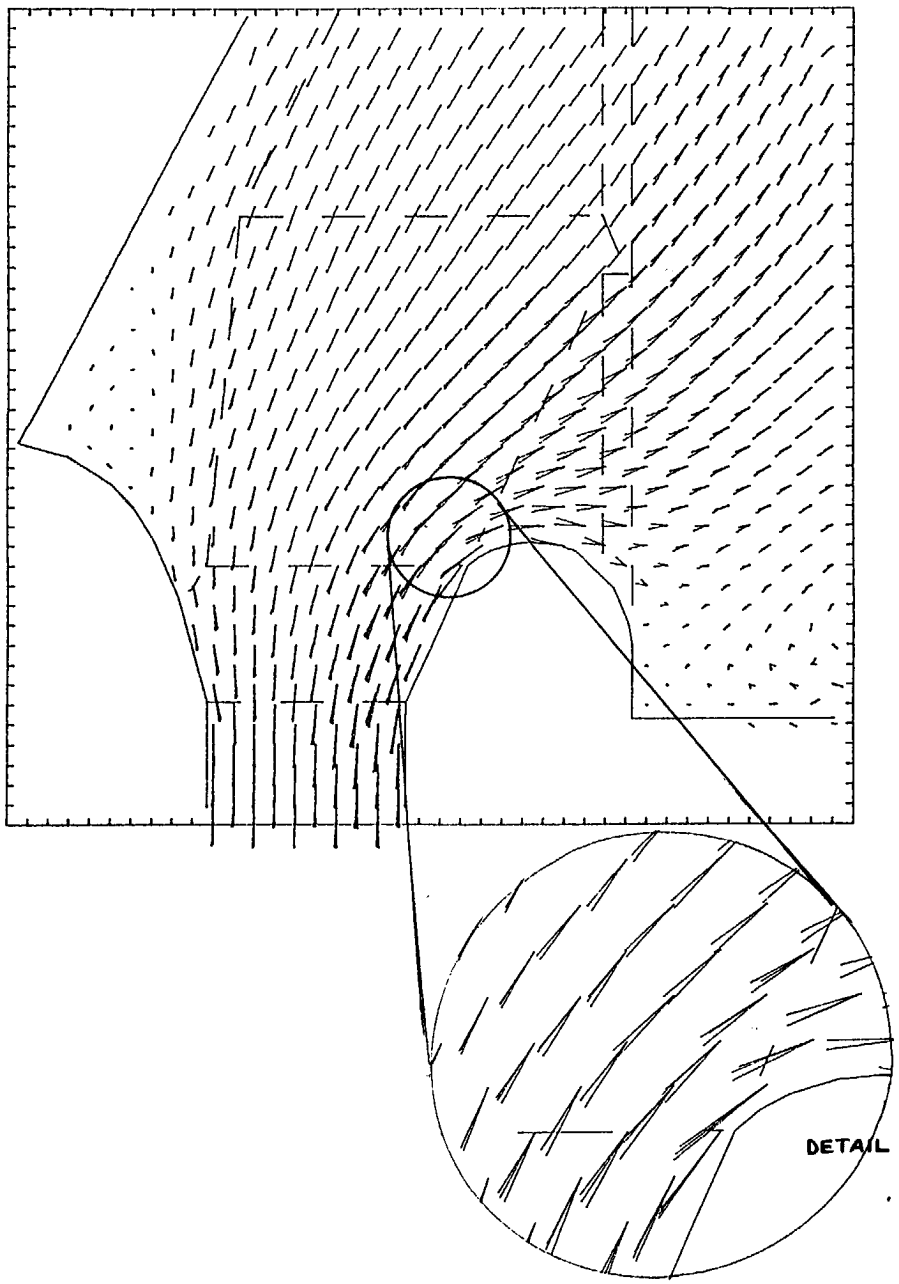
The result of the computation is given in fig. 9.10. In this figure the horizontal velocities are given for the surface-, middle- and bottom-layer in a single plot. It can be observed that the flow has a double helical movement in the outflow channel, i.e. towards the turbine entrances. This, and the bottom step in front of the entrance redistributes the flow in such a way that an even distribution of the flow between the turbines is obtained. In table 9.1 the distribution of the flow between the four separate turbine openings is given and compared to the measured values.

Turbine opening no:	4	3	2	1
Measured :	26.1	22.4	25.5	26.0
Calculated :	23.9	23.7	24.7	27.7

Distribution of outflow in %

table 9.1

As mentioned in section 9.3.1 measurements were available for comparison along rays 3, 4 and 5 (fig. 9.8). They consisted of the horizontal absolute velocities and directions on positions along the rays, as indicated in fig. 9.8. In every position velocity and horizontal direction were measured at at least two depth levels. Ray 3 and 4 were chosen in such a way that direct comparison with the numerical model was possible. In the case of ray 5 some interpolation of the numerical results was necessary. The result of the comparison is presented in the graphs in fig. 9.11. In this figure the measured velocity is multiplied by a factor $\sqrt{40}$ in order to obtain the prototype values. The angles are given in degrees in respect to a nort-south line (south = 0^0, east = 90^0).



surface-, mid depth- and bottom flow

fig. 9.10

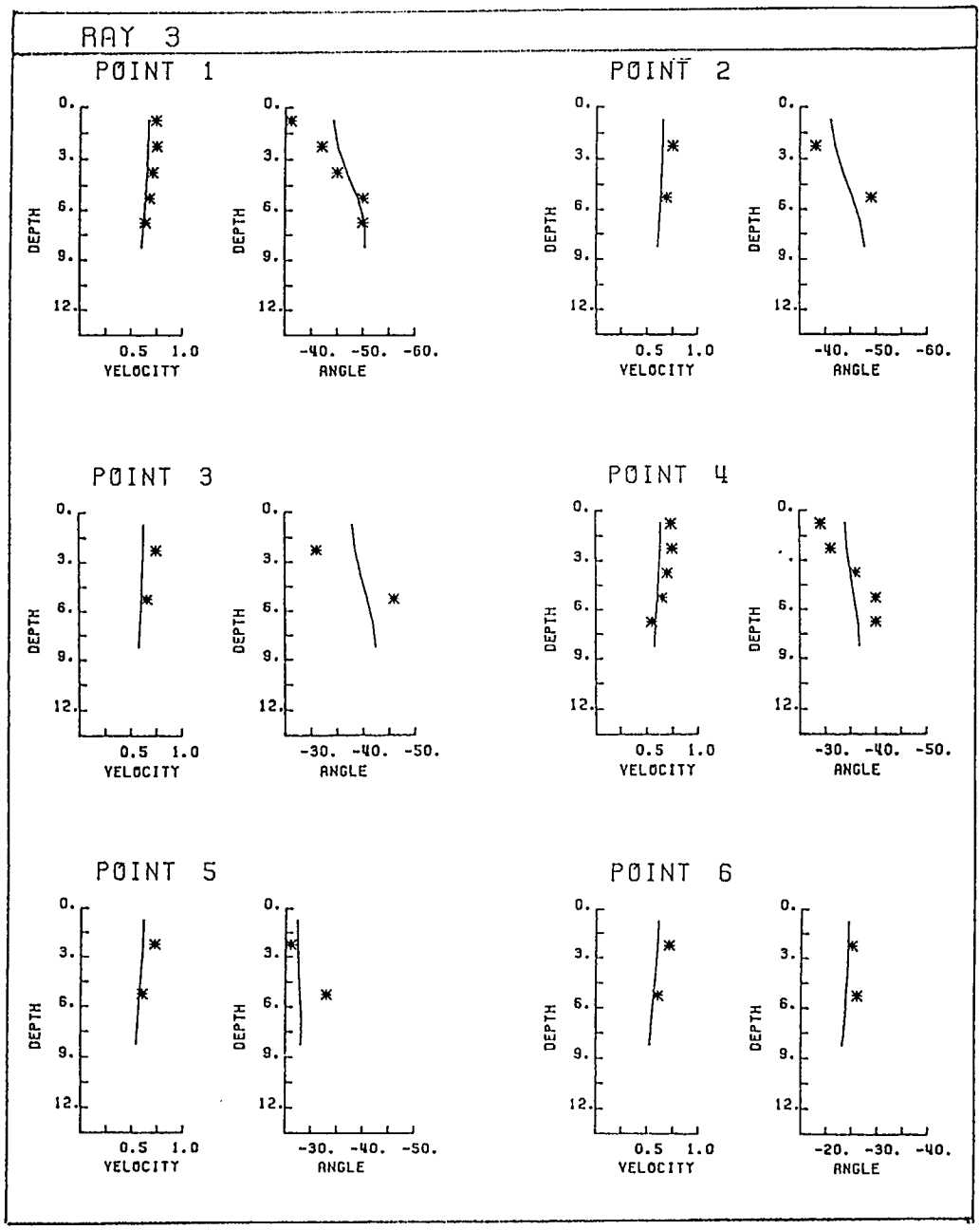


fig. 9.11 (ray 3, points 1-6)

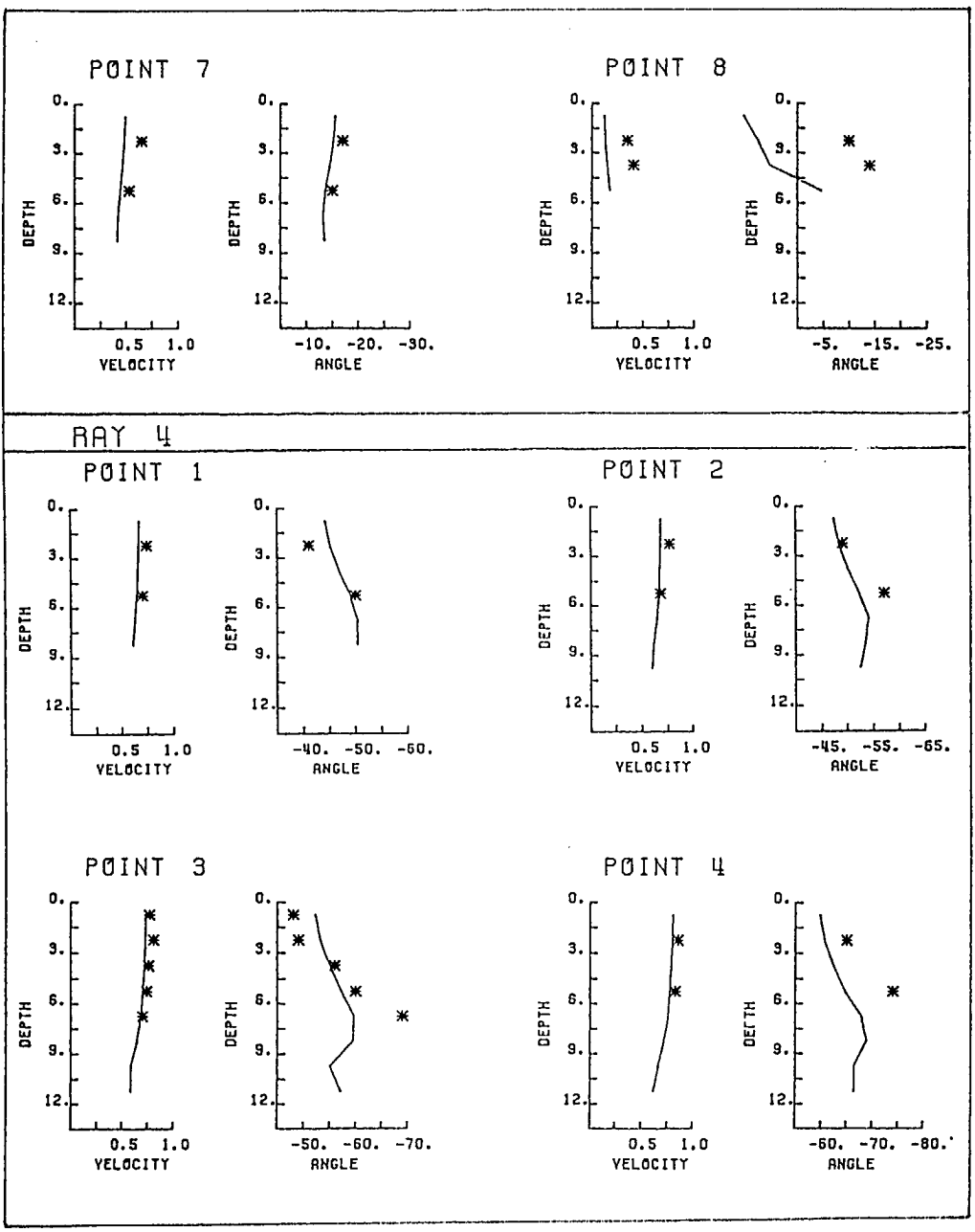


fig. 9.11 (ray 3, points 7-8 ; ray 4, points 1-4)

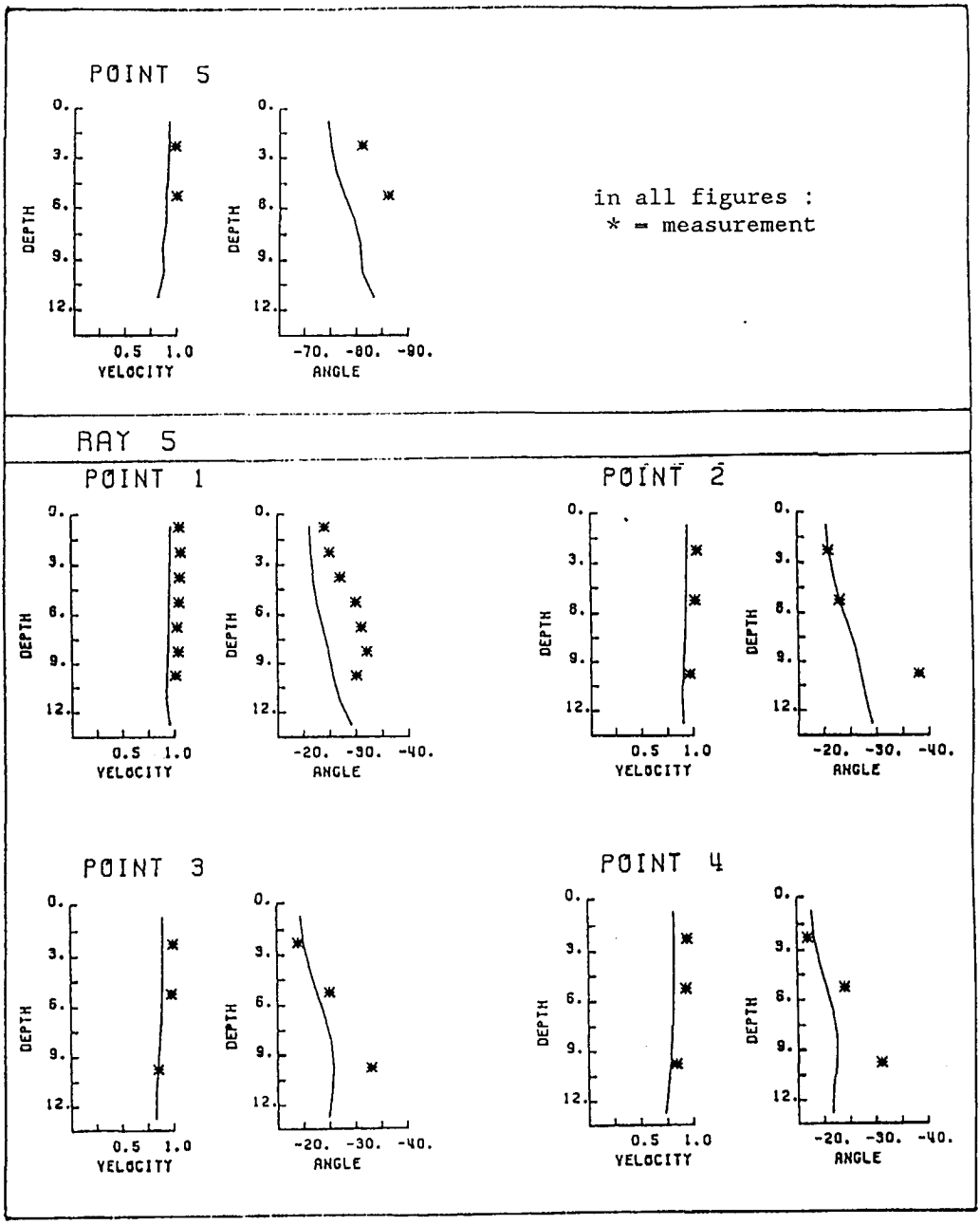


fig. 9.11 (ray 4, point 5 ; ray 5, points 1-4)

9.3.5 Discussion

The computed values agree reasonably well with the measurements. The absolute velocity and the direction of the flow, if averaged over the vertical, have about the right value, but it can be seen in the graphs (fig. 9.11) that the variation over the vertical is generally larger in the physical model.

From table 9.1, where the horizontal distribution over the four outflow sections is given, it appears that in the physical model a greater fraction of the fluid follows the west bank, resulting in a concentration at the west side of the outflow channel. In fig. 9.10 an eddy is visible on the west bank. While a weak eddy also was apparent in the physical model, this eddy may be too strong in the numerical model resulting in the effect mentioned above. The large deviation in point 8 on ray 3 also indicates that here the calculated and measured flow field do not agree. Note, that this point is situated near the same eddy.

Other calculations were performed. In these calculations a spatially constant viscosity coefficient was used. With a value of ν_t about equal to the lowest value used in the computation described above (i.e. ν_t=0.035) the eddy on the west bank disappeared. However, the distribution over the outflow sections hardly changed. Also the graphs as in fig. 9.11 stayed practically the same except in point 8 (ray 3) where much better agreement was found.

In a third calculation a higher value of the bottom friction was used : a_{sl}= 0.45 $m^{-1}s$ with ν_t= 0.05 m^2/s. This resulted in a better agreement between the measured and calculated absolute velocities, but the calculated directions showed much less vertical variation although the vertical average remained acceptable.

Some influence of the compressibility factor was found on the flow pattern near ray 5. If the value of α=0.1 (which was used to start the computations) was not changed into a greater value, the influence of the "staircase" boundary became visible in the flow field along the oblique wall, notwithstanding the porosity formulation.

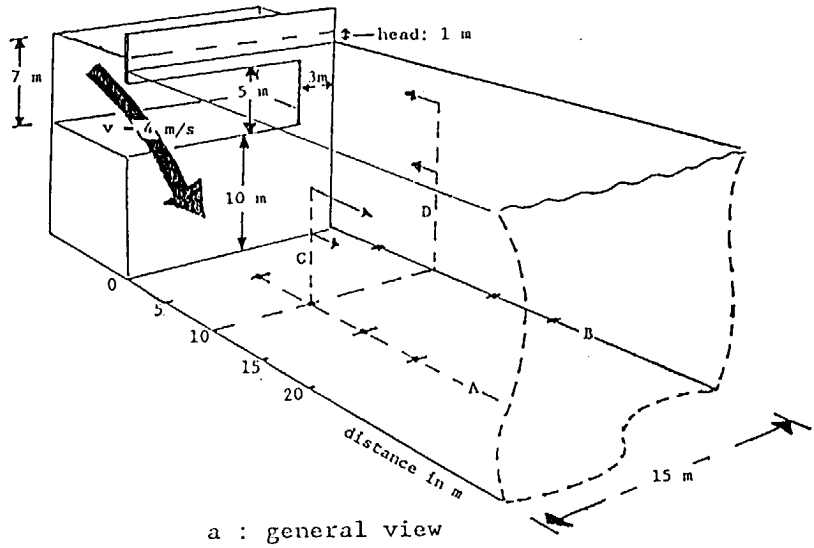
9.4 Gate in Storm Surge Barrier

9.4.1 Introduction

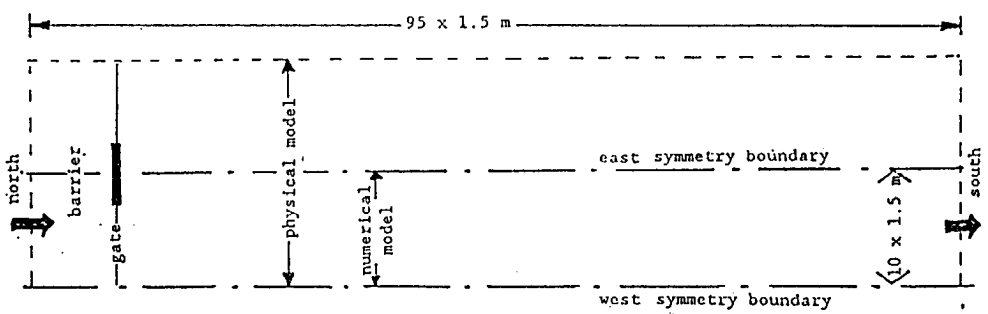
The model presented here concerns a detail of a much larger design for the removable storm surge barrier that is to be constructed in the Rotterdam Waterway. In this design the barrier is equipped with a large number of gates to discharge the river flow during low tide without having to open the barrier itself. To study the flow through these gates a physical scale model (1 : 30, undistorted) was constructed representing the space between the vertical symmetry planes through the middle of two subsequent gates. This study offered a good opportunity to test our numerical model.

The numerical model covered half the scale model, using the vertical mid-plane of the scale model as a symmetry boundary.

A general view of the model is given in fig. 9.12a, while fig. 9.12b shows the horizontal plan. The west boundary is formed by the symmetry plane in the middle of the gate, the east boundary is situated on the symmetry plane between two subsequent gates. The barrier and gate are situated at the north end of the model. Because the direction of the jet through the gate is unknown a priori, we could not take the gate itself as the inflow boundary. As indicated in



a : general view



b : horizontal plan

fig. 9.12

fig. 9.12, the inflow boundary was chosen 8 grid sizes upstream from the aperture. This had an interesting consequence connected to the use of the artificial compressibility method. In other applications the calculation could be started with the full boundary values and the internal field variables initially set to zero. Here, this procedure caused very high velocities in the aperture during the transient phase of the calculation. These velocities made the calculation unstable, resulting in overflow. The reason for this behaviour can be explained as follows : Upstream of the restriction caused by the gate aperture the

fluid is compressed, i.e. the (negative) divergence builds up. This results in a high pressure gradient across the gate during the transient phase, causing the high velocities mentioned above.

The solution of this problem was either to apply the boundary conditions gradually, i.e. starting with a zero inflow and slowly increase its value, or to chose initially a low value of the compressibility factor α. When the solution has converged to a more or less stationary flow, the value of α can be increased.

9.4.2 Grid System

The gate opening is situated between 2 and 7 m below the upstream water level. The head loss over the barrier is 1 m, while the downstream depth = 16 m. The vertical grid size for this application was chosen on the basis of the vertical eddy that was expected to develop below the gate opening. A grid size Δz = 1 m results in 10 grid points between the sill and the bottom. To limit the total number of grid points, a larger horizontal grid size was selected : $\Delta x = \Delta y$ = 1.5 m.

Originally the number of cells in the main stream direction was chosen as 65 but it was found that, in order to be able to specify a hydrostatic pressure distribution at the outflow boundary, this boundary should be placed further downstream. The resulting grid system thus consists of a block of 95 X 12 X 19 grid cells, corresponding to the width of the model of 15 m.

The water surface must be specified upstream as well as downstream from the gate aperture. If all other coefficients are chosen correctly one should find a pressure condition

along the upper boundaries that corresponds with the free surface condition, i.e. the calculated values of the piezometric height up- and down-stream of the gate should agree with the specified difference in surface level (fig. 9.13).

9.4.3 Boundary Conditions and Computation Parameters

The inflow velocity is given as 2.10 m/s. From this it follows that the velocity in the gate opening will be around 4 m/s. This velocity is taken as a basis for the determination of the eddy viscosity. Because it is difficult to specify a distribution a priori, a spatially constant value was chosen. Applying formula (7.9), using the downstream depth (-16 m.) and a friction velocity v*=0.25 m/s the value of for the eddy viscosity becomes :

$$\nu_t = 0.35 \ m^2/s$$

According (8.13), taking z_0 = 0.0045 m we find for the bottom slip factor a_{sl}:

$$a_{sl} = 0.01 \ m^{-1}s$$

Along the free surface and along the two side walls a free slip (a_{sl}= 0) condition is specified. Along the walls of the gate structure a slip factor of 0.01 m^{-1}s is taken.

Remark :

The viscosity coefficient can also be based on formula (7.10) for a free shear layer. Taking $U_{max} - U_{min}$ = 4 m/s a value of 5 m for the shear layer width results in the same

value for ν_t as found above. The value of d is somewhat arbitrary. Theoretically, the shear layer width is zero at the separation point, increasing downstream reaching finally a figure of the order of the water depth (15m.).

Because of the high velocity a small time step must be chosen :

$$\Delta t_{cd} = 0.1 \text{ s}$$

resulting in the following non dimensional computation parameters :

$$\text{Courant}_{cd} : \frac{U\,\Delta t}{\Delta x} = \frac{4. \times 0.1}{1.5} = 0.32$$

Diffusion limit : $2\nu_t(1/\Delta x^2+1/\Delta y^2+1/\Delta z^2)\Delta t =$

$$0.7 \times (1/2.25+1/2.25+1/1) \times 0.1 = 0.13$$

Convection parameter : $u^2/2\nu_t = 16/0.7 \times 0.1 = 2.3$

As stated in section 9.4.1 the compressibility factor is given a low value until the the flow has an almost zero divergence. Computations are performed with $\alpha = 0.1\ m^{-1}$ for about 1000 iterations. Then the value of α is increased to 0.2 and the computation is continued for another 2000 iterations. For the ADI part of the computation the same time step is used as for the convection-diffusion part. This results in a Courant number for the pressure wave of :

$$\text{Courant}_{adi} = \sqrt{(g/\alpha)}\Delta t/\Delta z = \frac{10 \times 0.1}{1} = 1$$

for the first part of the computation and :

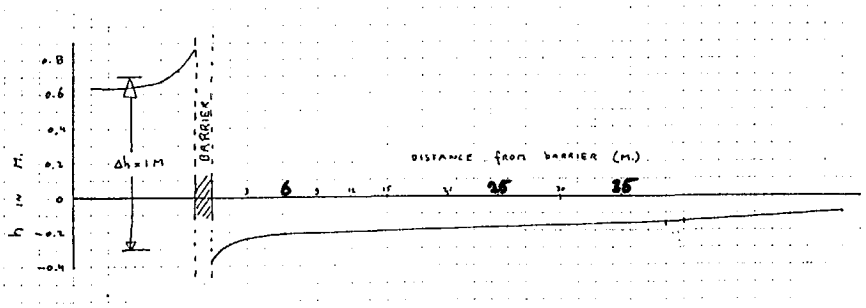
$$Courant_{adi} = 0.7 \qquad \text{for the final part.}$$

In the final part the propagation velocity is about 7 m/s

Because the great length of the computational domain and the small amount of wall friction (three free slip walls) the total number of iterations was rather large. It took a great number of iterations before the pressure wave subsided. Possibly a smaller initial value of the compressibility factor α would have been more efficient. At the other hand it is difficult to acertain when steady state is reached, because the velocity in the secondary flow regions changes very slowly. Thus, a lot of iterations are needed to make sure the solution is indeed stationary. The present computation was continued for about 10000 iterations. It must be noted however that, after some change in a boundary condition or other parameter, one just continues the computation without starting anew with a zero field of flow variables. Thus, subsequent computations, even if small changes in the geometry are made, generally take fewer iterations than the first computation.

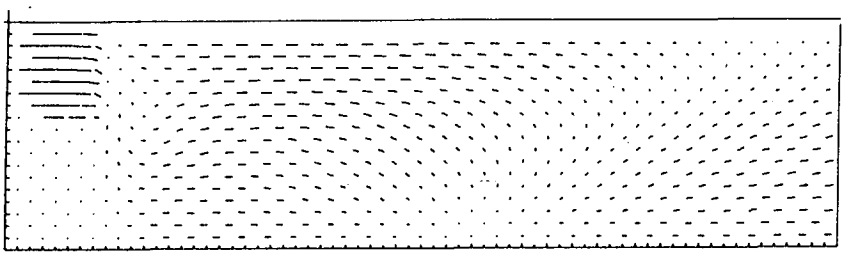
9.4.4 Results and Comparison to Measurements

In fig. 9.14 the velocity field is given in some horizontal and vertical cross sections parallel to the main stream direction. A jet is formed that is directed downward and to the west symmetry boundary. It hits the bottom about 25 m downstream (in the physical model this distance is less : 15 m), and turns upward towards the other side of the model. This helical movement is clearly seen in the cross



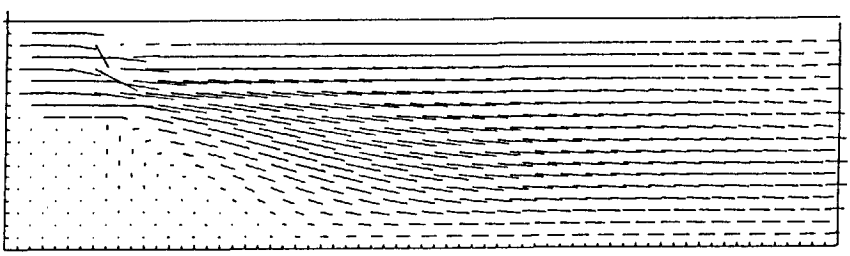
surface pressure (piezometric height)

fig. 9.13



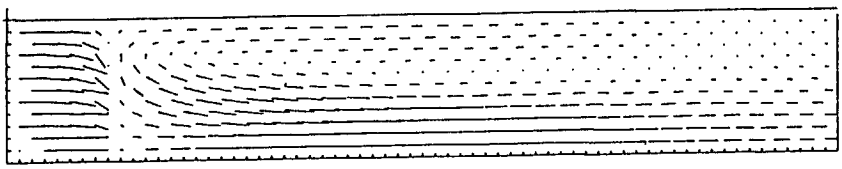
vertical cross section at east boundary

fig. 9.14a



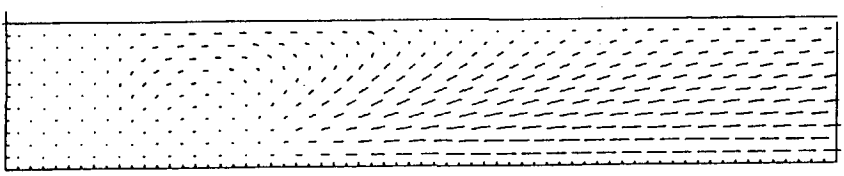
vertical cross section at west boundary

fig. 9.14b



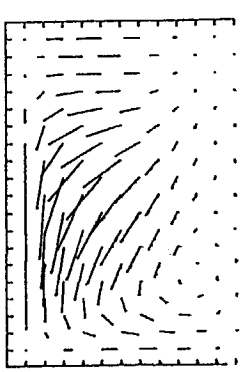
horizontal cross section at down stream surface

fig. 9.14c

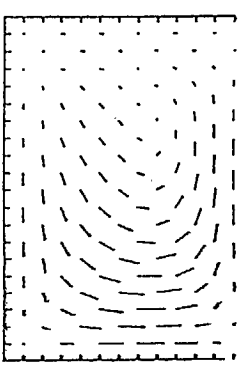


horizontal cross section at bottom

fig. 9.14d



a: 15m from barrier



b: 40m from barrier

fig. 9.15 cross stream cross sections

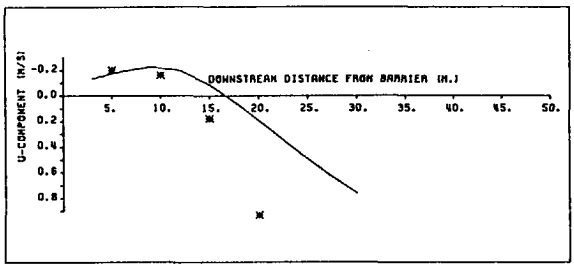
sections perpendicular to the main stream direction (fig. 9.15). As a result, a large region of back flow appears behind the closed section of the barrier. A vertical eddy exists below the gate opening. A remnant of this same movement is discernible in the vertical streamwise cross section along the east boundary (fig. 9.14).

In fig. 9.13 the piezometric head along the surface at the western boundary (i.e. the center line of the gate) is depicted. Across the gate a loss of head of about 1 m. is computed, which agrees with the difference in surface level chosen a priori. The variation in surface pressure up- and downstream from the gate are not represented by the geometry of the model. This is a consequence of the adopted rigid lid approximation.

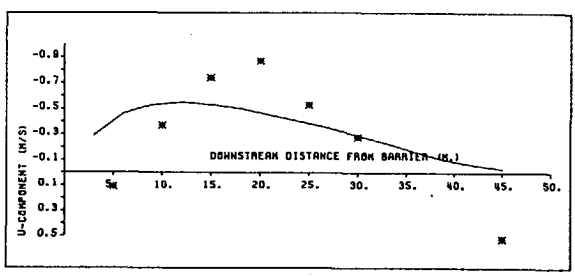
In the physical scale model velocities were measured along two streamwise rays near the bottom as indicated in fig. 9.12a. In two places a vertical distribution was obtained. Only the u-component (= main stream direction) was measured. Measurents were performed with a micro propellor instrument, electronically averaged over 100 s in each direction.
The measured velocities are compared with the calculated values in fig. 9.16.

9.4.5 Discussion

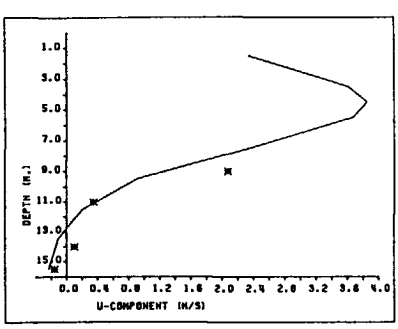
A controversial element in this kind of computation is the simple approach to turbulence. In the present computation a constant value of the eddy viscosity for the whole computational domain is used. A more sophisticated application of the constant eddy viscosity model is possible, for instance on the basis of a formula like (7.8),



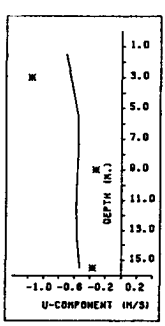
a: u-component along horizontal line A



b: u-component along horizontal line B



c: u-component along vertical line C



d: u-component along vertical line D

Comparison of measured and computed velocities (x-component) in m/s.

fig. 9.16

in which the eddy viscosity is related to the distance from the separation point. The main problem here is that, because of the low value of the eddy viscosity near this point, the specified distribution probably will have a large influence on the direction of the jet. This makes the choice of the correct distribution a difficult task. We believe that in a numerical model that is to used for engineering purposes the eddy viscosity distribution either should be generated automatically, i.e. by some thrustworthy turbulence model, or on the basis of simple, objective rules.

In the present case the jet will be diffused too strongly, due to the high value of the viscosity coefficient near the separation point. As a result the velocities in the regions of back flow are too low. Still, even with the simple turbulence model used, the location of those regions is reproduced fairly well by the numerical model.

10. CONCLUSIONS

In this thesis a computational model for three dimensional stationary fluid flow is developed based on the full 3-D Navier-Stokes equations. The efficiency of the proposed numerical method makes it feasible to perform 3-D flow computations, in which the flow in the direct surroundings of a hydraulic construction is simulated, on mini-computers or even PC's.
Three-dimensional simulation of this kind of flows is important because, as is demonstrated by the first example in chapter 9, substantial differences may occur between the 2-D depth averaged solution and the full 3-D solution.
Especially in calculations where the flow solution is used to determine the transport of sediment or contaminants, this difference has great impact on the final results.

By using the method of artificial compressibility the incompressible Navier-Stokes equations are transformed into a variant of the compressible equations. This makes it possible to employ finite difference methods very similar to the methods used for the solution of the 2-D SWE equations. These methods are very efficient because they use ADI on a regular staggered finite difference grid. We believe that the use of wall functions in the manner as described in chapter 8 of this thesis in combination with the porosity method of Patankar e.a. eliminates the disadvantages of such a grid system sufficiently to reach results that are acceptable for engineering purposes. Because of the computational efficiency of a method based on these principles, such a method can be used on relatively small computers. This kind of computation becomes of practical interest to engineers if

the cpu-time on a mini-computer does not exceed 15 hours, i.e. programs can be run overnight.

The method of artificial compressibility is not basically different from the more common method of pressure correction, so in principle the proposed numerical scheme can be used for dynamical calculations with minor changes in the program.

Because the alternating direction implicit (ADI) method is an important part of the solution procedure, attention is given to the extension of this method to 3-D. It is concluded that for hyperbolical problems the stability properties of the 3-D ADI method are fundamentally different compared to the 2-D version of this method. This difference is graphically demonstrated using the von Neumann stability theory. In our application the convection terms in the Navier-Stokes equations are generally dominant. Hence they exhibit a hyperbolical character.

While in 2-D it is possible to design an unconditionally stable ADI method (in the linear sense), this is not possible in 3-D. By treating the convective terms of the Navier-Stokes equations separately from the ADI process, the ADI part regains its stability properties. It is shown that the order in which the different operations are performed affects the stability properties of the overall computational procedure. In actual computations using the numerical scheme developed in this thesis, stable computations could be performed with a time-step two to three times larger than the theoretical limit for explicit time integration.

The turbulence is modelled in a simple way using a predefined distribution of the eddy viscosity . The use of

more advanced turbulence models intentionally was not incorporated in the present study. While for some flows, especially converging flows, a simple model may be sufficient, for more complicated flows it is difficult to estimate the viscosity distribution a priori and a better turbulence model is required. In diverging, and thus decelerating flow, regions of recirculation may appear. With the present simple turbulence modelling, the computed velocities in such regions will generally be too low. It should be noted, however, that even an advanced turbulence model like the k-ϵ model is not always able to reproduce such flows properly.

Because of the simple turbulence model, in most cases the eddy viscosity coefficient will be too high in the vicinity of a solid wall. To compensate for this fact a partial slip is specified as a boundary condition at solid walls. By a special use of the logarithmic wall function, the slip formulation is applied in such a way that the correct value (in the sense of the mixing length theory) of the bottom shear stress is obtained, independent of the value of the eddy viscosity. Thus, as an extreme example, if one should specify a large value of the eddy viscosity throughout the computational domain, the results of the 3-D computation will approach the 2-D horizontal solution with the right bottom friction. As shown in chapter 9 the same formulation can indeed be used to simulate the bottom friction in a 2-D application of the numerical model.

In one application (section 9.3) it was found that the use of an eddy viscosity, which had the same value everywhere in the model, gave virtually identical results as a computation using a spatially varying value based on the local depth. From this it could be concluded that, lacking a

trustworthy turbulence model, the use of a spatially constant value of the eddy viscosity is the most objective representation of the turbulence in the present computational method.

References

Beam, R. M. and Warming, R. F., An Implicit Factored Scheme for the Compressible Navier-Stokes Equations, AiAA J., Vol 16, No 4, 1978

Beam, R. M. and Warming, R. F., An Implicit Fin. Diff. Algorithm for Hyperbolic Systems in Conservation-Lwa Form J. Comp. Phys., Vol.22,1976

Brian, P. L. T., A Fin. Difference Method of Higher Order Accuracy for the Solution of Three-Dimensional Transient Heat Conducting Problems, Am. Inst. of Chem. Engineering J., Vol 7,no 3, 1961

Briley, W. R. and McDonald, H., On the Structure and Use of Linearized Block Implicit Schemes, J. Comp. Physics, 34, 1980

Chang, J. L. C. and Kwak, D., On the Method of Pseudo Com pressibility for Numerically Solving Incompressible Flows AAIA paper no. 84-0252, Reno, Nevada, 1984

Chorin, A. J., A Numerical Method for Solving Incompressible Viscous Flow Problems, J. of Comp. Physics, Vol 2, 1967

Chorin, A. J., The Numerical Solution of the Navier-Stokes Equations for an Incompressible Fluid, Bull. Am. Math. Soc., Vol 73, 1967

Dick, E. and Desplanques, D., Fin. El. Sol. of Steady N.-S. Eq. for Lam. Recirc. Flow with an Accelerated Pseudo Transient Method, Proc. 3th Int. Conf. Num. Meth. in Lam. and Turb. Flow, Seattle, 1983

Douglas, J., On the Num. Integration of $\partial^2 u/\partial x^2 + \partial^2 v/\partial y^2 = \partial u/\partial t$ by Implicit Methods, J. Soc. Indust. Appl. Math. (SIAM), Vol 3, No.1, 1955

Douglas, J. and Rachford, H. H., On the Numerical Solution of Heat Conducting Problems in Two and Three Space Variables, Trans. of the Am. Math. Soc., Vol 82, 1956

Douglas, J. and Gunn, J. E., A General Formulation of Alternating Direction Implicit Methods, Part I, Parabolic and Hyperbolic Problems, Numerische Math., Vol 6, 1964

Dronkers, J. J., Tidal Computations in Rivers and Coastal Waters, North Holland Publ. 1964 p.388.

Friedrichs, K. O., Symmetric Hyperbolic Lin. Differential Equ., Comm. Pure and Appl. Math, Vol 7, 1954

Hansen, W., Theorie zur Errechnung des Wasserstandes und der Stromungen in Randmeeren nebst Anwendungen, Tellus 8, 1956

Harlow, F. H., Shannon, J. P. and Welch, J. E., Liquid Waves by Computer, Science 149, 1965

Harlow, F. H. and Welch, J. E., Num. Calc. of Time Dependent Viscous Incompress. Flow of Fluid with Free Surface, Physics of Fluids, Vol. 8, No.12, 1965

Hartwich, P. M., Hsu, C. H., Liu, C. H., Vectorizable Impicit Algorithms for the Flux Diff. Split, 3-D Navier-Stokes Equations, Symp. on Parallel Processor Appl. in Fluid Mech. ASME Fl. Engg. Conference, Cincinatti, 1987

Hindmarsh, A. C., Gresho, P. M. and Griffiths,D. F., The Stab. of Expl. Euler Time-Integration for Certain Fin. Diff. Approx. of the Multi-Dim. Advection-Diffusion Equation, Int. J. Num. Meth. in Fluids, Vol 4, 1984

Hirt, C. W., Heuristic Stability Theory for Fin. Diff. Equ., J. of Comp. Phys., Vol 2, 1968

Hirt, C. W., Num. Methods for Bluff Body Aerodynamics, von Karman Institute for Fluid Dynamics, Lecture series, 1984-06,1984

Kuipers, J. and Vreugdenhil, C. B., Calculations of 2-D Horizontal Flow, Delft Hydraulics Report S 163-I, 1973

Kwak, D., Computation of Viscous Incompressible Flows, CFD von Karman Insitute for Fluid Dynamics, Lecture Series 1989-04, 1989

Leendertse, J. J., Aspects of a Computational Model for Long

Period Water Wave Propagation, Thesis TH-Delft 1967

Leendertse, J. J., A New Approach to 3-D Free-Surface Flow Modeling, Rand Corp. Report R-3712-Neth/RC, 1989

Peaceman, D. W. and Rachford, H. H., The Num. Sol. of Parabolic and Elliptic Diff. Equations, J. Soc. Indust. Appl. Math. (SIAM), Vol 3, No.1, 1955

Peyret, R. and Taylor, T. D., Computational Methods for Fluid Flow, Springer-Verlag, 1983

Praagman, N., Num. Sol. of the Shallow Water Equations by a Finite Element Method, Thesis, TU-Delft, 1979

Richtmyer, R. D. and Morton, K. W., Difference Methods for Initial Value Problems, Interscience Publ., 1967

Rodi, W., Turbulence Models and their Appl. in Hydraulics, IAHR, 1984

South, J. C., Recent Advances in Comp. Transonic Aero-dynamics, AIAA paper 85-0366, 1985

Stelling, G. S., On the Construction of Computational Methods for Shallow Water Flow Problems, Thesis, TU-Delft, 1983

Steger, J. L. and Kutler, P., Implicit Fin. Diff. Procedures for the Computation of Vortex Wakes, AIAA J., Vol 15, No.4 1977

Thom, A. and Apelt, C. J., Field Comp. in Engg. and Physics, Van Nostrand Co., 1961

Vichenevsky, R. and Bowles, J. B., Fourier Analysis of Num. Approx. of Hyperbolic Equations, SIAM studies in Appl. Math., Philadelphia, 1982

Vreugdenhil, C. B., Computational Hydraulics, An Introduction, Springer Verlag, 1989

Warming, R. F. and Hyett, B. J., The Modified Equation Approach to the Stability and Accuracy Analysis of Fin. Diff. Methods, J. Comp. Phys, Vol 14, 1974

Appendix A

Grid Generation

Computation of the wetted areas and volumed of those cells that are intersected by the solid boundaries

1. A grid block is defined containing the volume of fluid and solid mass that is to be modelled. This grid block consist of cells. Point (1,1,1) designated the upper north west corner of the upper north west cell in the block. The grid lines coincide with the cell ribs. (In the flow calculation, this grid system is shifted over half a grid size in all directions. The grid lines then connect the cell centers. The point (1,1,1) of the grid generation phase becomes the point (-0.5,-0.5,-0.5) in the flow calculation phase)

2. The horizontal plane is divided in areas by specifying lines. Each area so formed, receives an unique number. This number either refers to a "bottom plane" z=ax+by+c or to a file containing a bottom topography. In this way for every horizontal coordinate (x,y) or vertical grid line (i,j) a depht value can be found by the program.

3. In each vertical grid line (i,j) the depth is determinated by z(i,j)=a*i+b*j+c, where a,b and c are a set of coefficients for the area in which the point (i,j) is located. This depth is recorded in an array RZ(i,j,k) according to k= int(z) and RZ(i,j,k)= amod(z,1). In the same operation "dry" cell corners are flagged in an array CH.

4. Along j-lines the locations RZ(i,j,k) are connected by straight lines. Intersections with j-lines are recorded in RY(i,j,k) as done in operation 3 for RZ.

The same is done for the i-direction. Intersections with i-lines are recorded in RX(i,j,k).

At this point all geometrical data is recorded in the three arrays RX, RY and RZ. All dry upper-north-west cell corners are flagged in array CH(i,j,k).

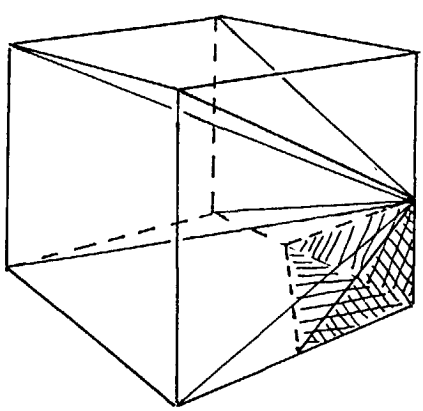


fig. A.1

5. For the three upstream faces of every cell (upstream for positive flow along i-, j-, and k-lines) the wetted area is computed from the information in the CH, RX, RY and RZ arrays. This area is expressed as a percentage (100% is a completely open face).

6. For each cell the wetted volume is obtained by the following method (fig. A.1) :

a. A single intersection as found in operation 3 and 4 is chosen. If no intersection is found on the ribs of a cell a cell corner is chosen. For each cell face the volume is computed of the pyramid formed by the wetted area of this face and the chosen intersection as top. In this way per cell six pyramids (two with a zero volume) are found, all having the same point as top. The six volumes added together are taken as the total wetted volume of the cell. Again this volume is expressed as a percentage.

7. Cells that are completely "open" (wetted area of all 6 faces and wetted volume = 100%) are flagged. For those cells a simpler calculation procedure can be done.

8. Finally, for every cell the wetted volume and the wetted area of the three downstream faces are stored in code (3 digits for the volume and 3 x 2 digits for the areas plus a single digit flag) in a single memory word in array ICH(i,j,k). All other arrays are released.

SAMENVATTING

Dit proefschrift behandelt de numerieke simulatie van stationaire drie-dimensionale onsamendrukbare stroming. Hiertoe worden de volledige 3-D Navier-Stokes vergelijkingen opgelost, waarbij elke ruimte dimensie op dezelfde wijze behandeld wordt. De gebruikte methode, de kunstmatige samendrukbaarheid van Chorin, maakt het mogelijk een gelijksoortig rekenschema te gebruiken als dat wat reeds veelvuldig wordt toegepast op de ondiep water vergelijkingen. Deze vergelijkingen zijn op te vatten als een speciale vorm van de 2-D Navier-Stokes vergelijkingen voor samendrukbaar medium. Een veel gebruikt numerieke oplossings methode is die van Leendertse. Dit schema lost de vergelijkingen op met centrale plaats differenties op een regelmatig eindig differentie rooster. Voor de tijds-integratie wordt een ADI methode gebruikt, waarbij de convectie termen min of meer expliciet worden behandeld.

Voor de simulatie van de 3-D onsamendrukbare stroming kan een dergelijk schema worden gebruikt wanneer dit tot 3-D wordt uitgebreid. Een belangrijk deel van dit proefschrift handelt over deze uitbreiding (hoofdstuk 4, 5 en 6). Hoewel het stabiliteitsprobleem als zodanig reeds bekend is, wordt in deze studie het fundamentele verschil in stabiliteits eigenschappen tussen 2-D en 3-D ADI schema's verduidelijkt.

De inpassing van de expliciet behandelde convectie termen in het totale schema kan op verschillende manieren. Dit blijkt van invloed op het gedrag van de numerieke oplossing.

Teneinde toepassing op reele situaties mogelijk te maken wordt een zeer eenvoudig turbulentie model ingevoerd, door de turbulente viscositeit a priori bekend te veronderstellen. Hoewel op deze manier redelijke resultaten zijn bereikt, zou een betrouwbaar en efficient turbulentie model de huidige rekenwijze sterk verbeteren. In zijn huidige vorm is de methode geschikt om op mini-computer installaties gebruikt te worden.

In een drietal voorbeelden wordt de methode op zijn bruikbaarheid getest. Behalve in gebieden met sterke neer vorming, stemmen de gemeten en berekende waarden redelijk met elkaar overeen.