1988

Numerical Algorithms For Analysis Of Dynamics Of Ideal Fluid With Free/moving Boundaries

Aleksander Adam Kania

NUMERICAL ALGORITHMS FOR ANALYSIS OF DYNAMICS OF IDEAL FLUID WITH FREE/MOVING BOUNDARIES

by

Aleksander Adam Kania
Faculty of Engineering Science

Submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Faculty of Graduate Studies
The University of Western Ontario
London, Ontario
August 1987

ISBN 0-315-40781-6

ABSTRACT

This thesis proposes and examines various algorithms for analysis of steady ideal fluid capillary flows with free/moving boundary.

For this class of problems, the leading parameter is the capillary number C which for an established flow and fixed geometry of the solution domain is proportional to the ratio of a velocity scale and surface tension. When $C << 1$ the problem can be simplified and is solved using Small Deformation Theory (SDT). Conditions for validity of SDT are identified.

When $C \cong O(1)$ one has to seek a simultaneous solution for two dependant variables, i.e. a flow field and a free surface shape. Depending on the order of linearization of the governing equations one arrives at the Picard Algorithm (linearization of the first order) and the 1-Step Algorithm (of the second order). The latter one provides significantly faster convergence.

All algorithms are based on a finite-difference approximation and the Alternate Direction Implicite (ADI) scheme has been choosen as a method of solution. A thorough study of an algebraic stability of equations of the flow field and the free surface has been carried out. This is supplemented by an analytical and numerical analysis of existence and uniqueness of the solutions to the free surface equation. The Wachspress optimization of relaxation parameters has been used in order to accelerate convergence of the ADI.

Finite differences discretization implemented in the thesis is based on the Hermitian equations, which generated compact differencing schemes of the second and higher order accuracy. In the thesis one can find a rigorous study of the interrelationship between the order of differencing scheme and the rate of convergence of the computed results with grid refining. This rate, called 'grid-convergence order' has been used as the criterion for identification of the minimum dimensionality of the computational grid. It was found that higher order methods require much finer grids than second order methods, to provide desired grid-convergence order.

For higher order method, the new fourth order compact difference estimate (independent of coordinate direction) of a mixed derivative was found. Its application significantly improved the grid-convergence order.

The discussion of the algorithms is supplemented by a physical interpretation of the results obtained for a number of particular cases.

ACKNOWLEDGEMENTS

I wish to express my deepest appreciation to Prof. J.M. Floryan for suggesting these problems and for his guidance and encouragement during the course of the work. Special thanks should be given to Dr. H. Rasmussen and Dr. Ch. Zemach with whom I have had many fruitful discussions concerning this work. Thanks are also due to many members of the Faculty of Engineering Science as well as the Department of Applied Mathematics, both faculty and students, with whom I have had many informal discussions and seminars that benefited my work enormously.

I would like to express my appreciation to the Faculty of Graduate Studies and NSERC for their long standing support during the course of this work.

Lastly, I want to extend my special appreciation to my wife Jola and to my children Anna and John for their sense of humour, patience and support during these years of intensive work.

TABLE OF CONTENTS

NOMENCLATURE

SYMBOL	DESCRIPTION
A_i	functional coefficients of the Newton-Raphson linearization (Eq. (5.6.1)),
A,a	parameters of the initial value problem (5.2.2),
B_i	functional coefficients of the Newton-Raphson linearization (Eq. (5.6.1)),
C	capillary number,
c	order of convergence of the linearization algorithm,
D	characteristic dynamic pressure ($D=\rho V^2$),
D_A	interval of feasible values of parameter A of the initial value problem (5.2.2),
$\mathcal{D}$	physical domain of the flow,
$E_{n,m}f(Gk)$	field f defined on the grid Gk by extrapolation of the numerical results $[f_{ij}]$ computed on grids Gn,Gm,
f , f^n	numerical solution obtained on the grid Gn, of spacing r=1/n,
$\tilde{f}$	exact solution to a given problem,
F(x)	$F(x) = -p_s + C\lvert\underline{V}\rvert^2/2$,
Gk	computational grid of dimension k x k, spatial increment r=1/k,

$G(x,y)$	$G(x,y) = -\sin(\pi x)(1-y/h(x))^3$,
$g(x)$	$g(x)=\int_0^x F(t)dt$,
h_L, h_R	location of the left and right contact points,
$h(x)$	function describing shape of the free surface,
H_i,H_{ij}	Hermitian formulae centered at x_i or (x_i, y_j),
H	$H=\{h(x_R): h(x)$ is a solution to problem (5.2.2)$\}$ = set of values $h(x_R)$ of any possible solution $h(x)$ to (5.2.2),
$H(x)$	auxiliary function defined in Sec.(5.6.5),
HOD	higher order compact differencing scheme,
2-OD	second order compact differencing scheme,
L	length of a slot (see Fig. 2.1).,
M_{ij}	set of indices of pivotal points of a uniform 9-point molecule centered at (x_i, y_j),
p_{ex}	external pressure exerted on the free surface by the outer fluid,
p_s	stagnation pressure of the liquid,
p	static pressure of the liquid,
Q	matrix of a system of linear algebraic equations resulting from the ADI sweep in x- direction,
r	grid spatial increment,
r_n	spatial increment of a grid of dimension n along each coordinate,
T	surface tension,
V	characteristic velocity,

V_s characteristic surface velocity (maximum of the fluid velocity at the free surface, $V = \max(|\underline{V}|^2)$ at $y=h(x)$,

$\underline{V}$ $=[u,v]$ velocity vector,

$\underline{V}_s$ $=[u_s, v_s]$ normalized velocity vector at the free surface ($\underline{V}_s = \underline{V} / V_s$ at $(y=h(x))$, $0 \le x \le 1$),

W matrix of a system of linear algebraic equations resulting from the ADI sweep in z- direction,

w relaxation parameter,

w_{kh}, w_{kv} optimal relaxation parameters of the ADI algorithm; k - ADI iteration counter; h,v indicator of horizontal or vertical direction of the ADI sweep, respectively,

w_ψ, w_f constant relaxation parameters for the stream function and for the free surface equations, respectively,

x .y cartesian coordinates,

x_L, x_R left and right boundary of the interval of arguments for the initial value problem (5.2.2),

z $z=y/h(x)$,

Z matrix of the global iterative scheme (see (3.3.4) - (3.3.5)).

GREEK SYMBOLS	DESCRIPTION
Δ	computational domain,
$\Delta x, \Delta y$	grid spacing along x- and y- coordinate axis,
ψ	stream function,
α	order of grid-convergence of discretization error,
ϵ	parameter of asymptotic expansion,
Ω	optimal set of relaxation factors,
ρ	density of the fluid,
θ	~~relaxation~~ parameter of z-gradient of stream function: $\psi_z(x,h(x))$, $0\leq x\leq 1$ at the free surface,

MATHEMATICAL SYMBOLS	DESCRIPTION
$\subset$	inclusion of sets,
$\in$	belonging to a set,
$\setminus$	subtraction of sets,
$\forall$	general quantifier ('for every...'),
$\exists$	existential quantifier ('there exists...'),
$\emptyset$	empty set.
$\| \cdot \|$	L-2 norm.

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Flows with Free Surface.

There is a variety of problems of a hydrodynamic or thermodynamic nature in which one can encounter capillary phenomena. These by definition are the phenomena in which surface tension on the interphase boundary plays a significant role. Among these we encounter:

- motion of liquids in thin capillaries,
- motion of droplets and bubbles in a liquid,
- formation of surface waves,
- decay of cylindrical jets,
- motion of thin liquid films, and many others.

The presence of the interface between two fluids can affect the motion pattern when either the interface has a finite curvature or when the surface tension is nonuniform. This suggests two possible results of the surface tension presence. In the case of the non-zero curvature, surface tension contributes to the normal forces along the interface. If interfacial tension is not constant, its gradient may act like a shear stress at the interface and can generate flows by itself or can modify the existing motions.

Surface tension is a material property which depends on temperature, composition and electrical potential. Since these factors may change easily, in practice one encounters many physical processes in which surface tension significantly affects the motion of the fluid and the shape of an interface. These are usually highly complex phenomena. Their understanding and description requires determination of dimensionless parameters which indicate dominant physical factors and open the way to formulation of theoretical and experimental models. Analysis of limit values of such parameters may result in specification of some classes of flow phenomena which may be particularly interesting from physical point of view and/or due to simplification of the mathematical model.

For example, when considering free surface flows, we may neglect gravity. It appears then that surface tension is important with essentially no limitations on the configuration scale. Such situation is of interest in simulation of microgravity conditions, which in real life offer the possibility of containerless processing of materials. Containerless processing of liquids or molten materials must be preceded, however, by indepth knowledge of the dynamics of this class of phenomena. In particular, the shape of the bulk fluid under various conditions and its stability have to be known. These requirements are possibly most significant in fueling research on phenomena in which surface tension is one of the major physical factors.

Much of the research dealing with free surface flows has to rely on numerical simulation. Formally, the effect of the surface tension on the hydrodynamics (and possibly on heat and mass transfer) occurs through the boundary conditions on the interface. Since the interface acts very often as the boundary of the computational domain, the applicable numerical methods must be highly accurate at the boundary. Generally three distinct solution methods can be identified (Ryskin and Leal [23]):

- small deformation approach in which the unknown interface is assumed to deviate only slightly from its initial position,
- boundary-integral technique, restricted to Stokes flow or to inviscid irrotational flows,
- universal method suitable for obtaining of the solutions for a general class of problems with finite Reynolds Number.

The last case, which is the most interesting, has captured attention of many researchers, but the results are relatively modest so far.

Whatever the method is, the accuracy of the numerical results is of great importance. As mentioned above, it is on the free surface, i.e. on the boundary, where the most important dynamical or geometrical effects come into play. Moreover, in the case of the surface tension driven flows, high gradients of velocity and/or temperature can be found in the vicinity of the interface. This need for higher accuracy gave rise to a rapid growth in the research on the higher-order differencing schemes (Hirsh [9,10]). Without highly accurate numerical results it is often difficult to get a deeper insight into the flow under investigation. Consequently, a numerical study may be helpful in achieving the desired experimental effects and in interpreting the results. Following this, let us state, that this work is focussed among others, on the accuracy of the computed results. The type of flow that we have selected as a subject of the research, i.e. ideal fluid flow with free surface, has a rather simple analytical description. But its simplicity permits us to identify important aspects of the numerical simulation and submit them to a detailed examination.

1.2 Review of Literature.

Virtually all previous work on problems of capillary flows has been propelled by an intensive experimentation. There is a large number of phenomena with surface tension contributing significantly to the dynamics of the flow. They have given rise to a rather complicated mathematical models lacking general analytical solution. The difficulty in obtaining exact solutions creates motivation for detailed studies of meaningful physical mechanisms, and this leads to the specification of particular classes of flow problems. A detailed study of flows affected by surface tension can be found in works of Levich (e.g. Levich and Krylov [14]), a review of theoretical investigations in this field is to be found in Ostrach [16]. There are three classes of problems which are relevant in this field. These are thermocapillary, diffuso-thermocapillary and thermo-electric flows (Scriven [25]). The literature in this field is vast and ranges from a special class of phenomena like floating-zone process (Hurle and Jakeman [9], or Rybicki and Floryan [20], [21]), to more general works (e.g. Strani and Piva [26] or Levich and Krylov [14]).

When the numerical simulation of flows with a free surface is considered, the existing results are not so impressive. The research effort is directed along either Eulerian or Lagrangian formulation (review paper by Floryan and Rasmussen [8]). As for the Eulerian method, the comprehensive review of the subject can be found in Cheng and Yulin [3]. The Eulerian approach and its numerical implementation is studied in Ryskin & Leal [23,24]. The same authors contributed extensively to the problem of treatment of discretization of the flow domain (Ryskin [22]). Since the flow field equations are to be solved on a well defined domain, there is a problem of dealing with the unknown part of the boundary of the solution domain, i.e. with the free surface.. Most numerical methods for tracking interfaces are arranged either by following the interface as it passes over the reference grid, or by following the changes of the flow domain bounded by the moving free surface (Rasmussen and Salhani [18], Hyman [12]). If we apply the latter method, the best way is to map the flow domain onto the computational region of fixed geometry. This can be accomplished at the expense of supplementing the system of governing equations with some additional equations which originally belonged to boundary conditions. In this way the computational domain is related to the physical flow domain by transformation of variables. This transformation is defined in terms of a function describing the free surface. This technique contributed to an increase of interest in constructing boundary fitted orthogonal curvilinear coordinate systems (Ryskin [22], Floryan [7]).

Another problem that one encounters in the numerical solution, is discretization of the governing equations. There are several methods of accomplishing this, of which we focus on finite differences. A sufficiently general approach is provided by the Hermitian Method, which can be used to generate finite difference approximations of differential formulae with a specified accuracy. This method in its conceptual framework is not a new one, as explained by Collatz [4], but was applied to flow problems quite recently. A short, general presentation of this method is provided by Peyret and Taylor [17]. An indepth, comprehensive study of this method, along with historical background and an extensive list of references can be found in Hirsh [10]. The application of a compact higher order method to the solution of the Navier-Stokes equations is presented in some detail by Elsaeser and Peyret [6] and by Aubert and Deville [1]. The latter publication contains some remarks on an a posteriori estimation of the accuracy of computed results. It is interesting to point out that the accuracy of the computed results is rarely discussed. It certainly deserves more attention, since an a posteriori accuracy estimate is decisive in determination of the dimensionality of a computational grid to avoid erratic variations of the truncation error. It may also supply information helpful in choosing a particular differencing scheme, and in arranging the logical structure of the numerical algorithm.

A numerical solution of a steady free-boundary problem is equivalent to solving an artificial time-dependant problem. So by its very nature it is of an iterative character with advancing flow field/free surface computations. Ryskin and Leal [23] claim that the most effective way of dealing with this kind of problem is the combination of the flow field/free surface iterative scheme with iterative solution of the flow field equation. This solution to the flow equation can be provided by application of the Altenate Direction Implicit (ADI) scheme of Peaceman and Rachford (Richtmayer and Morton [19]). Numerical experimentation shows that separation of these two iterative loops is computationally inefficient. This rules out non-iterative or multigrid procedures for the solution of the flow field equations. Quite recently other schemes have been tested, like an incremental Block - Line - Gauss - Seidel Method (Napolitano and Walters [15]), which can be applied to the general class of Navier - Stokes equations. However, its superiority to the ADI method for free boundary problems has not been demonstrated yet. Another interesting suggestion was made by Botta and Ellenbroek [2] who proposed application of the Successive Over-Relaxation (SOR) Method to the Poisson equation in calculations of the flow with the free surface. This work is limited to the second order discretization only.

The ADI iterative scheme was made more attractive, when opportunity to accelerate convergence was provided by the Wachspress theory of optimal relaxation (Wachspress [28]). It should be said that the discretized model of the free surface flow does not satisfy formal requirements for application of the Wachspress optimization. It was found, however, that for the type of problems that we deal with in this work, the optimization yields good results even when the shape of the interface is far from flat, which is required by the Wachspress theory. Wachspress optimization was developed only for the second order discretization schemes (which may be solved by means of scalar 3-diagonal systems of algebraic equation). There is no report available that deals with application of Wachspress optimization to ADI algorithms for compact higher order discretization schemes.

1.3 Objectives of the Dissertation.

The main objective of this thesis is to develop efficient methods for solving 2-dimensional steady free surface flow problems, with surface tension. The analysis is limited to steady flows of an ideal fluid in a slot, with fixed free surface/solid-walls contact points and constant surface tension. The motion is induced by imposing known velocity distribution at the bottom of the slot. Mathematical formulation of the flow problem leads to the Dirichlet-type Boundary Value Problem with field described by the Laplace equation, and surface tension contribution described by the nonlinear equation of balance of normal forces along the free surface. The methods presented in the thesis can be extended to non-potential or viscous flow problems. Special attention is paid to the analysis of the accuracy of the computed results.

Modelling of the free surface flows places us in a position of having part of the flow boundary undetermined (free surface). This difficulty is circumvented by utilizing a suitable transformation of independent variables, so that the actual computational domain assumes a rectangular shape. This however leads to a more complicated form of the governing equations.

Numerical simulation of the analytical model is being pursued in terms of finite difference discretization. Finite differencing schemes are of the second or higher order. The core of the thesis (Chapters 4 and 5) presents results obtained using 2-nd order differences. To compare both, the higher order differencing and its application is presented in Chapter 6. It will be shown that for the type of problem considered in the thesis, the higher order diferencing scheme yields less reliable results than the traditional second order differencing.

In computational practice it is very important to make a distinction between the two numerical parameters:

- the theoretical estimate of the order of the error of a finite difference approximation
- the rate of convergence of numerical solution, when grid spacing is being decreased.

These two parameters are theoretically of the same value, and any correctly formulated algorithm should possess this property. This property of the proposed algorithms is studied in detail in Chapter 3 and in certain sections of the following chapters.

Discretization of differential expressions in the analytical model yields a system of algebraic equations, which due to its dimensionality is to be solved by an iterative procedure. In this work we apply the ADI iterative scheme (Sec.3.2). ADI is driven by relaxation, or equivalently, by a pseudo-unsteady iterative process.

For the second order discretization, the optimization of relaxation is arranged following the conceptual framework of the Wachspress theory. It was found that even though the algorithm did not satisfy the assumptions of the Wachspress theory, optimized relaxation yielded much faster convergence compared to the pseudo-unsteady (or single-valued relaxation) strategy.

The main point in solving the model problem is simultaneous determination of the flow field and the free surface shape. In Chapter 4 the simplest case is discussed, that is when surface tension dominates all other factors affecting dynamics of the flow at the free surface. This case is described by the Small Deformation Theory which makes possible decoupling of the flow from determination of the free surface. This simplifies the algorithm and establishes a good reference point for more general (Direct) algorithms.

There are two Direct Algorithms discussed in this work (Chapter 5); Picard and 1-Step Algorithms. Both methods are to work in cases where assumptions of Small Deformation Theory are not valid. They are based on the concept of simultaneous solution of the flow and interface equations. In Picard Method (Sec.5.5) the free surface solver is separated from the field solver and the solution is obtained by means of commutative use of both algorithms. This heuristical arrangement has justification in the first-order linearization scheme of the dependent variables involved in field and interface equations. If this scheme is to be of the second order (Newton), we arrive naturally at the so called 1-Step Algorithm (Sec. 6.6). This algorithm treats both variables, i.e. stream function and free surface as implicitly related. The comparison of the Picard and 1-Step Algorithms shows, that the 1-Step Algorithm with appropriate relaxation may yield a much faster convergence of the iterative process.

The mathematical model of the flow contains several physical parameters which may be combined into two dimensionless quantities: stagnation pressure and capillary number. The range of variation of these parameters is carefully studied (Sec. 4.3, 5.4, 5.5.3). Since their magnitude affects the values of coefficients of the algebraic system, they are also ~~imp~~licitly bounded by constraints resulting from algebraic stability criteria.

Chapter 7 presents a number of physical cases of the flow for various combinations of physical parameters. Although the algorithms presented in the preceding chapters were analyzed on a model problem with contact points at the same elevation and uniform external pressure, examples show that the algorithm may be easily extended to flows with fixed but uneven contact points and non-uniform external pressure acting upon the free surface.

CHAPTER 2

STATEMENT OF THE PROBLEM

The aim of this work is to develop algorithms suitable for analysis of flows with free/moving interfaces. A simple model flow problem is selected in order to study and evaluate the performance of the proposed algorithms. The model problem that is described below contains all the essential features of the actual free boundary problems, and yet it retains a considerable simplicity.

Consider a two-dimensional steady motion of an ideal fluid in a configuration shown schematically in Fig. 2.1. The motion is driven by some type of external means which lead to a known velocity distribution at the lower boundary of the solution domain. The upper boundary is formed by an interface with surface tension acting along it. It is assumed that the fluid on the other side of the interface has negligible density and viscosity. The shape of the interface is determined by equilibrium conditions between surface tension forces and forces arising due to pressure jump across the interface. Pressure distribution along the external side of the interface is considered to be known while the pressure along the interior side can be determined from the velocity field through application of the Bernoulli equation.

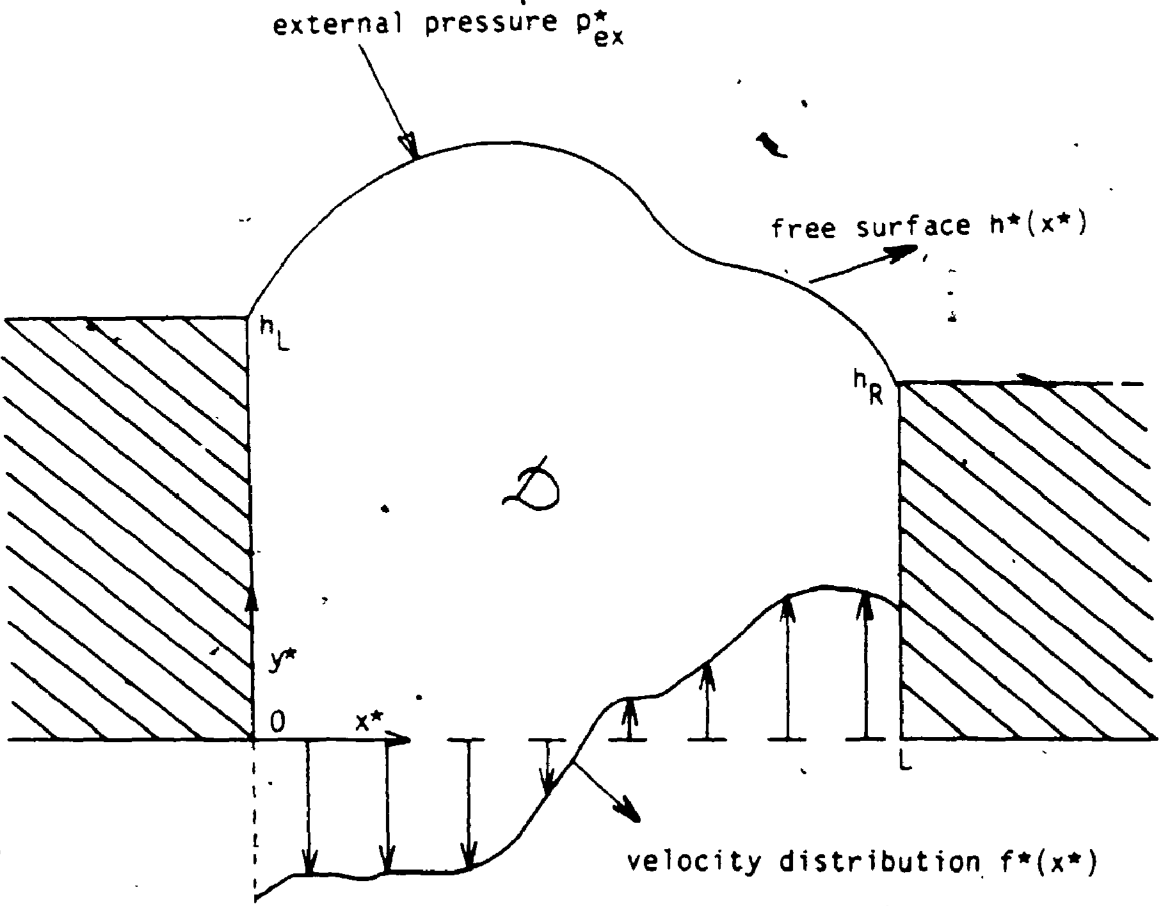

external pressure p^*_{ex}
free surface $h^*(x^*)$
h_L
h_R
$\mathcal{D}$
y^*
0
x^*
L
velocity distribution $f^*(x^*)$

Fig. 2.1 Ideogram of the flow configuration.

It is assumed that there are no body forces present. The problem has the appearance of a standard cavity flow problem with the upper boundary replaced by a free surface.

The motion of the fluid in the cavity is governed by the continuity equation

$$\nabla \cdot \underline{V}^* = 0 \, , \qquad (2.1)$$

assumption of irrotationality

$$\nabla \times \underline{V}^* = 0 \qquad (2.2)$$

and Bernoulli equation

$$p^* + \frac{1}{2} \rho \, \underline{V}^{*2} = p_s^* \qquad (2.3)$$

In the above, $\underline{V}^* = u^*\underline{i} + v^*\underline{j}$ denotes the velocity vector, u^* and v^* are velocity components in the x^* and y^* directions respectively, $\underline{i}$ and $\underline{j}$ denote the unit vectors in the x^* and y^* directions respectively, p^* stands for the static pressure, p_s^* denotes stagnation pressure and ρ is the density of the fluid.

The equations are subject to the following boundary conditions:

$$u^* = 0 \qquad \text{at } x^* = 0 \qquad (2.4)$$

$$u^* = 0 \qquad \text{at } x^* = L \qquad (2.5)$$

$$v^* = f^*(x^*) \qquad \text{at } y^* = 0 \qquad (2.6)$$

$$u^* = v^* h^*_x \qquad \text{at } y^* = h^*(x^*) \qquad (2.7)$$

$$p^*_{ex} - p^* = T\ h^*_{xx}/(1+h^{*2}_x)^{3/2} \qquad \text{at } y^* = h^*(x^*) \qquad (2.8)$$

In the above, T denotes surface tension, $p^*_{ex}(x^*)$ stands for the known external oressure at the interface, $p^*(x^*)$ stands for the interior static pressure at the interface, subscripts x and xx denote d/dx^* and d^2/dx^{*2} respectively, $f^*(x^*)$ stands for the known normal velocity at the bottom of the solution domain and $h^*(x^*)$ denotes the unknown location of the interface. Equations (2.4)-(2.5) and (2.7) describe the no-penetration conditions at the left and right ends of the solution domain and at the interface, respectively, Eq. (2.6) describes normal velocity distribution imposed at the lower boundary by external sources and Eq. (2.8) expresses the balance of forces at the interface in the direction normal to the interface.

Equation (2.3) permits elimination of the static pressure from (2.8), i.e.,

$$p^*_{ex} - p^* - \frac{1}{2}\rho|\underline{V}^*|^2 = Th^*_{xx}/(1+h^{*2}_x)^{3/2} \qquad (2.9)$$

We need to specify the type of contact made by the free surface at the end walls to close the problem. We shall restrict ourselves to the case of fixed contact points, i.e.,

$$h^*(0) = h_L \qquad (2.10)$$

$$h^*(L) = h_R \qquad (2.11)$$

which corresponds to the case of fluid sticking to the sharp edges at the end walls.

We scale the problem by considering length L of a cavity as a lengths scale, maximum of the normal velocity at the bottom V as the velocity scale and T/L as a pressure scale. The pressure scale T/L, which we refer to as the capillary pressure scale, is preferred over the dynamic pressure scale $\rho V^2/2$ since a balance between the capillary forces and pressure jump across the interface must always be maintained. With the above scales the physical variables (marked before by an asterisk) take on dimensionless form and equations and boundary conditions can be written in the following way:

$$\nabla \cdot \underline{V} = 0 \qquad (2.12)$$

$$\nabla \times \underline{V} = 0 \qquad (2.13)$$

$$u = 0 \qquad \text{at } x = 0 \qquad (2.14)$$

$$u = 0 \qquad \text{at } x = 1 \qquad (2.15)$$

$$v = f(x) \qquad \text{at } y = 0 \qquad (2.16)$$

$$u = vh_x \qquad \text{at } y = h(x) \qquad (2.17)$$

$$p_{ex} - p_s - C|\underline{V}|^2/2 = T\, h_{xx}/(1+h_x^2)^{3/2} \qquad \text{at } y = h(x) \qquad (2.18)$$

$$h(0) = h_L \qquad (2.19)$$

$$h(1) = h_R \qquad (2.20)$$

In Eq. (2.18) C is the capillary number defined as

$$C = \rho V^2 L/T \tag{2.21}$$

Equations (2.12)-(2.20) describe our model free-boundary problem. The problem is completely specified if the external pressure distribution $p_{ex}(x)$ at the interface is known, the normal velocity distribution $f(x)$ at the lower end of the cavity is specified and if the locations of the contact points h_L and h_R are prescribed. Given these four quantities we can analyse properties of the solution of the flow problem as a function of two parameters, i.e. stagnation pressure p_s in the interior of the cavity and capillary number C describing conditions at the interface. We should add that the normal velocity distribution $f(x)$ has to satisfy an obvious condition of no net mass flow across the boundary. While $f(x)$, $p_{ex}(x)$, h_L and h_R are in principle arbitrary (with the above qualification), they will be conveniently selected in order to facilitate testing of the proposed numerical algorithms.

The number of unknowns in Eq. (2.12)-(2.20) can be reduced by introducing either stream function or velocity potential. We have elected to work with the stream function formulation, since it leads to the Dirichlet rather than Neumann type boundary conditions; this adds to the simplicity of the model problem to be studied numerically. We introduce a stream function ψ defined as

$$u = \partial\psi/\partial y, \quad v = -\partial\psi/\partial x \tag{2.22}$$

Equation (2.12) is identically satisfied, while (2.13) leads to

$$\nabla^2\psi = 0 \qquad (2.23)$$

subject to the following boundary conditions

$$\psi = 0 \qquad \text{at } x = 0 \qquad (2.24)$$

$$\psi = 0 \qquad \text{at } x = 1 \qquad (2.25)$$

$$\psi = -\int_0^x f(x)dx \qquad \text{at } y = 0 \qquad (2.26)$$

$$\psi = 0 \qquad \text{at } y = h(x) \qquad (2.27)$$

$$p_{ex} - p_s - \frac{1}{2}C|\underline{V}|^2 = h_{xx}/(1+h_x^2)^{3/2} \qquad \text{at } y = h(x) \qquad (2.28)$$

where $|\underline{V}|^2 = (\partial\psi/\partial x)^2 + (\partial\psi/\partial y)^2$ (2.29)

Equation(2.27) expresses the fact that the interface is also a streamline.

Equations (2.23)-(2.29) describe the problem to be solved numerically and the appropriate algorithms are described in Chapters 4-6. The rest of this chapter is devoted to the discussion of the qualitative properties of the model problem.

There are two unknowns involved in the equations (2.23)-(2.28), i.e. field variable $\psi(x,y)$ and boundary variable $h(x)$. The Eq. (2.23)-(2.27) can be therefore interpreted as a system of equations for $\psi(x,y)$ and $h(x)$. It should be noted that while $h(x)$ is explicitly coupled with $\psi(x,y)$ through Eq. (2.28), $\psi(x,y)$ is only implicitly coupled with $h(x)$ through Eq. (2.27). The second coupling occurs because of the dependance of $\psi(x,y)$ on the location of the boudary condition (2.27), i.e. because of the presence of the moving boundary. There are two sources of nonlinearity in the problem The first one, which is explicit, is associated with the nonlinear boundary condition (2.28), and the second one, which is implicit, is associated with the presence of the moving boundary. The second nonlinearity appears explicitly after transformation of variables has been introduced (Chapter 3). It is worth pointing out here that the moving boundary problem is always nonlinear even if the field equation and all the boundary conditions are linear. The principle of superposition is therefore not applicable and the analytical solutions to the problem at hand cannot be found explicitly, except for a certain number of special cases. This justifies the search for an efficient numerical algorithm.

Equations (2.23)-(2.28) have an elliptic character and form together with the contact conditions (2.19)-(2.20), a closed system, i.e. a system which contains a sufficient number of conditions to determine all the unknowns. There exist theoretical questions regarding whether this is a well posed problem and the existence and uniqueness of the solution. We do not address these issues here. Our primary objective is to develop methods to determine the solution of the problem assuming that it exists. We demonstrate in Chapter 5 that solutions do not exist for certain combinations of parameters C and p_s; we propose and implement an algorithm to diagnose such conditions. The only comment that we can make regarding the issue of uniqueness is

(e) $h(0) = h(1) = 1$

Equation (3.2.2.a) is the field equation, Eq. (3.2.2.b) is the boundary condition describing the balance of normal forces at the interface. The above form of the governing equations, i.e. grouping together of the field equation and normal stress boundary condition on one side and the remaining boundary conditions and the contact conditions (3.2.3) on the other side, illustrates very well the character of the problem. The problem consists of a system of two nonlinear coupled equations for ψ and h, with the boundary conditions on the known domain Δ (no undefined "free boundary"). The domain consists of a unit square for the field variable $\psi(x,z)$ and unit interval for the boundary variable h(x). Here the unit interval I (the image of the free surface).is the part of the boundary of the domain . The nonlinearity occurring due to the presence of the moving boundary (unknown h(x)) is explicitly visible.

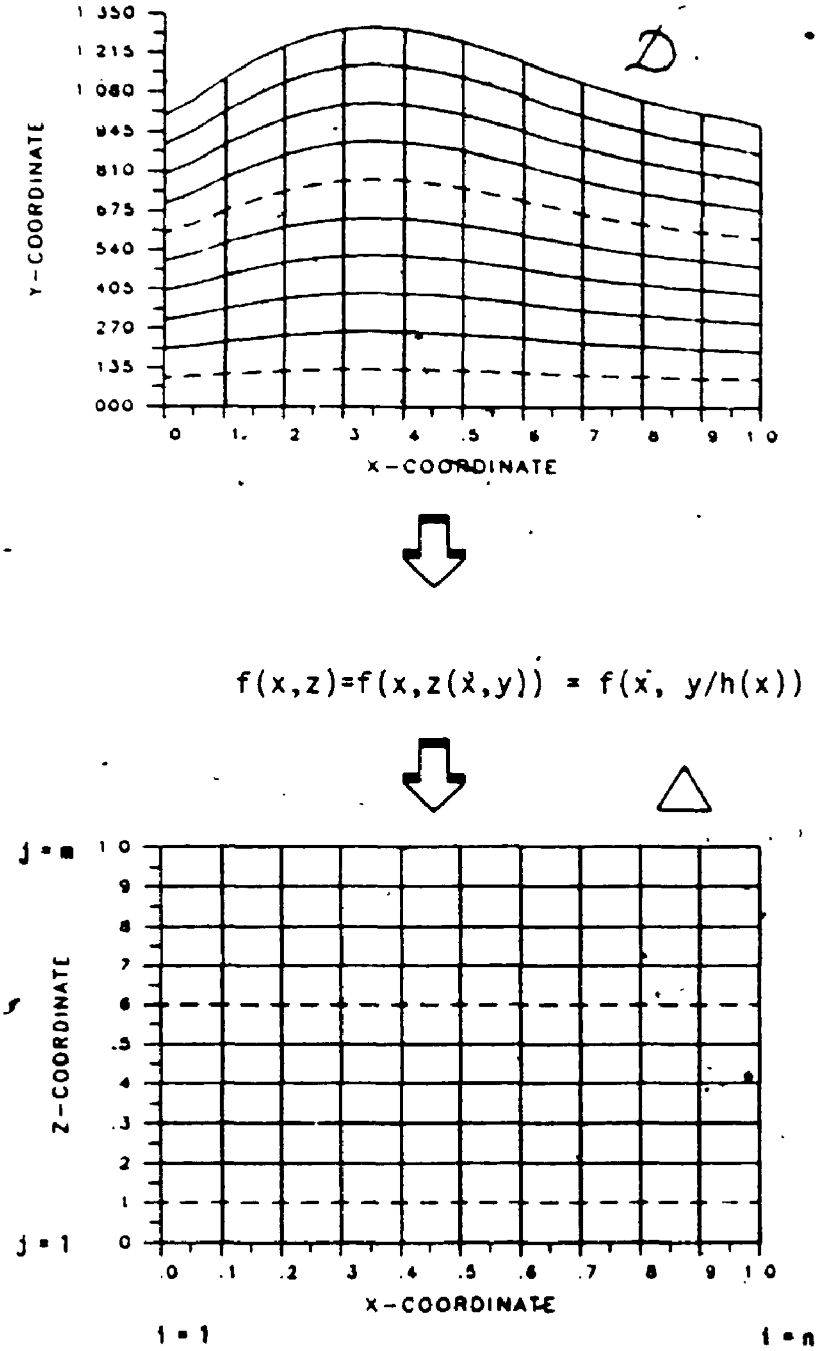
Transformation (3.1) is very simple, and therefore preferable. We must however bear in mind complications resulting from such a mapping. The first one is associated with the mixed derivatives appearing in the governing equation. Another complication is of a geometrical character. The counter-image of a uniform square grid defined on is not uniform in $\mathcal{D}$. It can be easily seen that such a grid is denser towards the free surface in the $\mathcal{D}$ plane, if its curvature is positive and sparser if negative. We might therefore expect a better accuracy of the numerical solution in the case of forces and is unaffected by the dynamic effects associated with the motion of the fluid. We discuss this case in details in Chapter 4. When the dynamic effects are strong as compared to the static effects ($C \rightarrow \infty$, "weak" surface tension), Eq. (2.28) displays a singular limit. We note that in order to have a capillary surface, the right-hand side of the Eq. (2.28) must be retained. An increase of C on the left hand side of Eq. (2.28) has an effect similar to the increase of the pressure jump $|p_{ex}-p_s|$, that has been discussed in the previous paragraph. If C is sufficiently large, the solution of the problem may not exist. It will be shown in Chapter 5 that given a particular $f(x)$ and $p_{ex}(x)$, , the solution exists only in a bounded region in the (p_s,C) parameter space. Since velocity field $\underline{V}$ depends implicitly on p_s and C, the general explicit bounds for (p_s,C) assuring existence of the solution are not available.

CHAPTER 3

ALGEBRAIZATION OF THE PROBLEM

3.1 Treatment of Irregular Geometry.

Since the solution domain described in Chapter 2 has an irregular geometry arising from the presence of the free surface, one has to devise a special procedure to handle such geometries and to track the location of the free surface. We may approach this problem by carrying out the solution directly in the physical domain $\mathcal{D}$ shown on Fig. 2.1, or by having this domain mapped onto a domain of a simpler geometry. We intend to discretize differential equations using finite difference method and, therefore, the issue of appropriate selection of the computation domain is important. If $\mathcal{D}$ is non - rectangular, discretization may require unnecessarily complicated interpolation schemes in order to represent location of the free boundary. Representation of the boundary conditions can be simplified by mapping of $\mathcal{D}$ onto a computational domain Δ,which is taken to be in the form of a cube (rectangle in 2 dimensions). The mapping can be arranged in a variety of ways. One of the most common approaches nowadays is the numerical generation of boundary fitted orthogonal curvilinear coordinate systems (e.g. Ryskin [22]). Numerical coordinate generation is a very effective tool in dealing with a variety of internal or external flows of highly irregular bondaries, like the flow of a gaseous bubble through a fluid, or the flow around an airfoil.



$$f(x,z)=f(x,z(x,y)) = f(x, y/h(x))$$

Fig. 3.1 Ideogram of coordinate transformation.

In our case, however, the solution domain $\mathcal{D}$ has the form of a cavity with only one side (free boundary) having an irregular form, as is shown in Fig. 2.1. This particular geometry permits us to avoid the numerically generated boundary-fitted orthogonal coordinate system through application of a simple analytical mapping shown in Fig. 3.1. It is intended to explore in the present work the capabilities and limitations of such a mapping when applied to the analysis of the moving boundary problems. The mapping as shown in Fig. 3.1 has obvious limitations, i.e.

(i) - it is limited to a single valued function $h(x)$ describing the location of the interface, and

(ii) - it becomes singular when the slope of the interface becomes vertical.

One can however devise other mappings of a similar form to deal with the problem at hand, for example mapping onto a unit circle, or mapping of only a part of the solution domain. These mapping are not general, i.e. they cannot deal with completely arbitrary geometries, however they can deal with a sufficiently large class of problems to make it worthwile to explore problems associated with their numerical implementation. The advantages of such mappings can be summarized as follows:

(i) - they are given analytically,

(ii) - they avoid numerical coordinate generation;

(iii) - they lead to a simpler form of the governing equations when compared to the form appropriate for the general curvilinear coordinate systems.

After the mapping, the computational domain has a form of a unit square. The mapping described in Fig. 3.1 results in the following transformation of independent variables and the relevant derivatives:

$$
\begin{aligned}
&z = y/h(x) \\
&\psi_x = \psi_x + (-zh_x/h)\ \psi_z \qquad (3.2.1) \\
&\psi_y = \psi_z\ /h \\
&\psi_{xx} = \psi_{xx} + (zh_x/h)^2 \psi_{zz} + 2(-zh_x/h)\psi_{xz} + \psi_z\ z(2h_x^2 - h_{xx}h)/h^2 \\
&\psi_{yy} = \psi_{zz}/h^2
\end{aligned}
$$

The governing equations transform to the following form:

$$\psi_{xx} + \psi_{zz}\left(\frac{1 + z^2 h_x^2}{h^2}\right) - \psi \left(\frac{2zh_x}{h}\right) \qquad (3.2.2.a)$$

$$+ \psi_z\ z(2h_x^2 - h_{xx}h)\frac{1}{h^2} = 0$$

$$\frac{h_{xx}}{(1+h_x^2)^{3/2}} = -p_s + \frac{1}{2}\ C\ \left(\frac{h_x^2 + 1}{h^2}\right)\psi_z \qquad (3.2.2.b)$$

with the boundary conditions: (3.2.3)

(a) $\psi(0,z) = 0 \quad 0 \leq z \leq 1$

(b) $\psi(1,z) = 0 \quad 0 \leq z \leq 1$

(c) $\psi(x,1) = 0 \quad 0 \leq x \leq 1$

(d) $\psi(x,0) = 0 \quad 0 \leq x \leq 1$

(e) $h(0) = h(1) = 1$

Equation (3.2.2.a) is the field equation, Eq. (3.2.2.b) is the boundary condition describing the balance of normal forces at the interface. The above form of the governing equations, i.e. grouping together of the field equation and normal stress boundary condition on one side and the remaining boundary conditions and the contact conditions (3.2.3) on the other side, illustrates very well the character of the problem. The problem consists of a system of two nonlinear coupled equations for ψ and h, with the boundary conditions on the known domain Δ (no undefined "free boundary"). The domain consists of a unit square for the field variable $\psi(x,z)$ and unit interval for the boundary variable h(x). Here the unit interval I (the image of the free surface). is the part of the boundary of the domain . The nonlinearity occurring due to the presence of the moving boundary (unknown h(x)) is explicitly visible.

Transformation (3.1) is very simple, and therefore preferable. We must however bear in mind complications resulting from such a-mapping. The first one is associated with the mixed derivatives appearing in the governing equation. Another complication is of a geometrical character. The counter-image of a uniform square grid defined on is not uniform in $\mathcal{D}$. It can be easily seen that such a grid is denser towards the free surface in the $\mathcal{D}$ plane, if its curvature is positive and sparser if negative. We might therefore expect a better accuracy of the numerical solution in the case of concave free surfaces than in the case of convex ones. This is discussed later in Sec. 5.6.5.

3.2 Algebraization of the Model Problem.

The foundation of a numerical solution of a differential equation is a discrete simulation of continuous relations in the given domain Δ. The algebraization of the problem consists of three steps:

(i) - discretization of the computational domain (grid generation),

(ii) - discretization of governing equations,

(iii) - generation of the closed system of algebraic equations simulating the model problem.

The first step in discretization consists of specification of a grid, which is to cover the computational plane . Pivotal points $(x_i, z_j) = (i\Delta x, j\Delta z)$ of the grid system are the points at which we wish the governing equations to be satisfied. The next step of discretization is simulation of derivatives by algebraic formulae. In this thesis we confine ourselves to the finite difference schemes based on Taylor expansion in the vicinity of pivotal points of a grid. These schemes can be more or less sophisticated (e.g. box integration method - Wachspress[28]), but we shall restrict ourselves to the simplest compact schemes with Lagrange residuals of truncated Taylor series of at least $O(\Delta x^2, \Delta z^2)$ order. By "compact" we understand here those schemes, in which Taylor expansion around a fixed point (x_i, z_j) does not involve terms defined at nodal points which would be located outside the circle of radius $(\Delta x^2+\Delta y^2)^{\frac{1}{2}}$ with the centre at (x_i, z_j). Application of compact schemes has two advatages. The first one is associated with the fact that the Taylor expansion is used only in a small neighbourhood $O(\Delta x, \Delta z)$ of a pivotal point (x_i,z_j) and its truncation, therefore, results in a small error. This distinction becomes important when considering the higher order discretization scheme which, in the case of the non-compact method, may require Taylor expansion over neighbourhood $O(2\Delta x, 2\Delta z)$ or bigger. The second advantage consists of the (block) tri-diagonal structure of the algebraic equations resulting from the discretization for which very efficient solution algorithms exist. This is important when considering block relaxation or other implicit solution procedures.

Let for future reference M_{ij} denote the set of indexes of pivotal points belonging to the molecule with centre at (x_i, z_j).

The governing equations are defined on the continuous domain Δ and the unit interval I. Discretization of the governing equations involves a change of the domain from continuous onto the discrete one (pivotal points of grids covering Δ). This results in generation of a system S of (n x m + n) equations, each of them to be satisfied at the nodal points of the grid. These equations have a form of algebraic equations, with values of dependent variable and its derivatives unknown and treated as separate variables. To close the algebraic system, we have to supplement S with additional algebraic equations, so that the new system has a unique solution. Those additional equations relate values of function and its derivatives within the compact molecule. There are several methods of obtaining these relations, like Hermitian, Mehrstellen OCI and others (Collatz [4], Peyret and Taylor [17]). We shall implement only Hermitian methods in the present work.

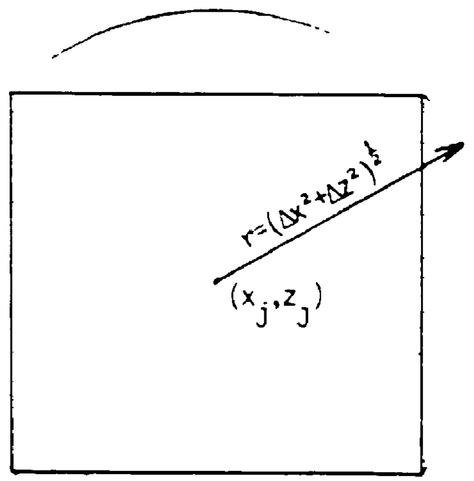


Fig. 3.2 Finite Difference Molecule for Compact Methods.

In the following discussion the symbol f will stand for the dependent variable, i.e. stream function $\psi(x,z)$. The discussion applies also to the function describing interface h(x), which is a function of a single variable.

The Hermitian method considers as unknowns at each grid point the value f_{ij} of the function, and its derivatives. Let us assume, that these unknowns are f_{ij}, f'_{ij} and f''_{ij}, where primes denote derivatives in a fixed direction (either x or z). Our compact molecule will thus reduce to 3 consecutive points. By definition (Peyret and Taylor [17]), a general three - point Hermitian formula for the function, first and second derivatives is given by

$$H_i = \sum_{k=-1}^{1} (a_k f_{i+k} + b_k f'_{i+k} + c_k f''_{i+k}) = 0 \qquad (3.2.4)$$

Here, for simplicity, function and its derivatives are single indexed, which corresponds to the selected direction of differentiation. It is easy to generalize the above definition by extending it to the whole 9-point compact molecule with the centre at (x_i, z_j). In Eq. (3.2.4) coefficients a_k, b_k, c_k are constants which are to be determined by requiring the formula (3.2.4) to be valid with respect to the relationship existing between any analytical function and its two derivatives at three consecutive nodes of a grid. Here we follow a standard practice and make use of Taylor expansion of terms involved in (3.2.4) around the central point of a molecule. When these expansions are substituted into Eq. (3.2.4), we obtain condition

$$\sum_{j=0}^{\infty} F_j \frac{d^j f}{dx^j} = 0 \qquad (3.2.5)$$

where $F_j = F_j(\{a_k, b_k, c_k\})$ with $k=-1,0,1$. If we wished the condition (3.2.4) to be exactly satisfied, we would have to solve the infinite system of equations $F_j=0$, $j=0,1,\ldots$ In practice it is required that only a finite number of these coefficients are equal to zero:

$$F_J = 0 \qquad j = 0,1,\ldots J \qquad (3.2.6)$$

which means that Eq. (3.2.5) and in consequence the Eq. (3.2.4) are satisfied with an accuracy of order $O(\Delta x^{\ell})$, where ℓ is to be examined separately.

The formula (3.2.6) generates a homogenous system of J+1 linear equations with 9 unknowns $\{a_k, b_k, c_k\}_{k=-1}^{1}$. This system has always a trivial solution, which is of no interest for the intended applications. If J+1 < 9, we have an ℓ-parameter family of solutions, $\ell \geq 1$. If J+1 $\geq$ 9, the existence of a non-trivial solution is uncertain, and depends on a particular formulae relating function and its derivatives. It is worth mentioning that there exists a theoretical possibility of obtaining a non-trivial solution of Eq. (3.2.6) for large value of J. This means that it is possible to obtain highly accurate Hermitian formulae based on a nine point molecule. This issue, however, requires a separate investigation and is not pursued here.

Below we present an illustrative example of the implementation of the Hermitian method.

Let us assume that we wish the Hermitian formula to be of a special form;

$$H_i = a_{-1}f_{i-1} + a_0 f_i + a_1 f_{i+1} + b_0 f_i' + c_0 f_i'' \qquad (3.2.7)$$

Equation (3.2.7) is a special form of (3.2.4), with $b_{-1}=b_1=c_{-1}=c_1=0$, i:e. it contains derivatives only at a central point of a molecule. Let us expand f_{i-1} and f_{i+1} into the Taylor series and substitute them into (3.2.7). Assume, that we want J in (3.2.6) to be equal to 3. It can be easily shown that

$$F_0 = a_{-1} + a_0 + a_1$$
$$F_1 = -a_{-1}\ x + a_1\ x + b_0$$
$$F_2 = (a_{-1} + a_1)\Delta x\ /2. \ + c_0$$
$$F_3 = (-a_{-1} + a_1)\Delta x\ /6.$$

If we require (as in (3.2.6)) $F_0 = \ldots = F_3 = 0$, we obtain a system of 4 equations with 5 unknowns, which yields a solution dependent on one parameter, say A. Setting $A = a_{-1}$ leads to the following:

$$H_i(A) = A(f_{i-1} - 2f_i + f_{i+1} - f_i''\Delta x^2) \qquad (3.2.8)$$

It follows from the construction that $H_i(A) = 0$ is accurate up to the order $O(\Delta x^4)$.

If in our example (3.2.7) we substitute J=2 and require $c_k=0$, we would end up with:

$$H_i(B) = B(f_{i+1} - f_{i-1} - f_i' 2\Delta x\ ) \qquad (3.2.9)$$

which is of accuracy $O(\Delta x^3)$.

From (3.2.8) and (3.2.9) we may easily obtain second order accurate finite difference representation of first and second derivatives, which are well known, i.e.

$$f_i'' = (f_{i+1}-2f_1 +f_{i-1})/\Delta x^2 + O(\Delta x^2) \qquad (3.2.10)$$
$$f_i' = (f_{i+1}- f_{i-1})/2\Delta x + O(\Delta x^2)$$

The above example was intended to show that (3.2.8) and (3.2.9), and in consequence (3.2.10), represent only particular solutions to the discretization problem. What makes them important, is the way they affect the algebraic structure of the system S.

The hermitian formula (3.2,7) is a special case of general definition (3.2.4). It makes use of the values of f at all three points (along the selected coordinate) of a linear molecule, and of the first and second derivatives at the central point (x_i, z_j) only. If we drop this restriction, we may derive a whole new family of difference formulae, in particular expressions of the higher order than $O(\Delta x^2)$. We have utilized this approach to construct higher order algorithm. Particular Hermitian formulae of the higher order are listed in the Appendix. It seems that determination of $H_{ij}$of higher order for mixed derivatives is of special interest. Such H_{ij} is spanned on the entire 9-point molecule of Fig.3.1 (see Anex). Performance of the higher order algorithm and its comparison to the second-order method are given in Chapter 6.

The algebraization process of the given analytical problem $\mathcal{A}$ is completed when the computational domain is discretized and analytical problem

$$\mathcal{A} = \mathcal{A}\,(x,z,f\ ,fx,\ldots) = 0 \qquad \text{for } (x,z) \in \Delta \qquad (3.2.11)$$

is replaced by an algebraical problem :

$$\begin{aligned}
&\mathcal{N}_{ij} = \mathcal{A}\,(x_i, z_j, f_{ij}, f_{xij}, f_{zij}\ldots) \\
&H^{1}_{ij} = 0 \qquad\qquad\qquad\qquad (3.2.12) \\
&H^{2}_{ij} = 0 \\
&\ldots \\
&H^{l-1}_{ij} = 0 \\
&i = 1,\ldots,m, \quad j = 1,\ldots,n
\end{aligned}$$

Assume that $\mathcal{N}_{ij}$ is a linear equation (if not, linearization is required) of l-unknowns, $H^{1}_{ij},\ldots H^{l-1}_{ij}$ are Hermitian formulae which are linearly independent. Therefore the analytical problem (3.2.11) is replaced by a system of linear equations. This system consists of (n x m) blocks of the form (3.2.12). Each block is of l equations, where l is a number of dependent variables in the top equation of system (3.2.12).

EXAMPLE 3.1

Let us suppose, for illustration, that we want to solve the flow problem (3.2.2)-(3.2.3) with C=0 and p_s =0. Then the analytical problem is reduced to the Laplace equation for ψ , while free surface is flat (h=1). In this case $\mathcal{A} = \mathcal{A}(x, y, \psi_{xx}, \psi_{yy})$. Algebraization of this system results in a system of linear equations:

$$\mathcal{N}_{ij} = (\psi_{xx})_{ij} + (\psi_{yy})_{ij}$$

$$H^1_{ij} = (\psi_{xx})_{ij} - (\psi_{i,j-1} - 2\psi_{ij} + \psi_{i,j+1})/(\Delta x^2) = 0 \qquad (3.2.13)$$

$$H^2_{ij} = (\psi_{yy})_{ij} - (\psi_{i-1,j} - 2\psi_{ij} + \psi_{i+1,j})/(\Delta y^2) = 0$$

$$i = 1,\ldots m \ , \quad j = 1,\ldots,n$$

If we were to solve our analytical problem with $h \neq 1$, the model equation for the stream function would also show the presence of $\partial^2\psi/\partial x\partial z$ and $\partial\psi/\partial z$, and in consequence, the system would have to be enlarged by two additional Hermitian Formulae: the first one involving mixed, the second one - the z-derivative. Difference approximation of the second order for the mixed derivative can be found for instance in Smith [27]; the higher order difference estimation is explicitly given in Anex.

SEPARABILITY

If the grid consits of $(n\ x\ m)$ points, then the system (3.2.12) consists of $(\ell\ x\ n\ x\ m)$ equations with $(\ell\ x\ n\ x\ m)$ unknowns. It is natural to think about reducing dimensionality of (3.2.12) before attempting a numerical solution. This could be achieved through manipulations with the equations constituting the system. In particular, let us assume that we can find a linear combination of equations within each block of (3.2.12), so that the resulting system reduces to the form

$$\tilde{\mathcal{N}}_{ij} = \mathcal{N}_{ij}(x_i . z_j, f_{ij} , f_{xij}, f_{zij} , f_{xxij}, f_{zzij} . . .)$$

Then we can restructure system (3.2.12) into the form:

$$\tilde{\mathcal{N}}_{ij} = 0 \qquad i=1,\dots n,\ j=1,\dots m \tag{3.2.14}$$

$$\begin{aligned} &H^1_{ij} = 0 \\ &H^2_{ij} = 0 \\ &\dots \\ &H^l_{ij} = 0 \qquad i=1,\dots,n,\quad j=1,\dots,m \end{aligned} \tag{3.2.15}$$

Since when solving (3.2.14)-(3.2.15) we are interested primarily in obtaining estimates of f_{ij}, not of its derivatives, we may attempt to <u>separate</u> (3.2.14) from (3.2.15) and solve just (3.2.14). This separation, if possible, is of great numerical significance, since it yields ℓ-fold reduction of the dimension of the original system (3.2.12).

It can be shown that the systems (3.2.12) supplemented with Hermitian formulae of the <u>second</u> <u>order</u> accuracy of type

$$H_{ij}= H_{ij}(x_i, z_j,\{f\}_{kl},\{\frac{\partial^s f}{\partial x^r \partial z^t}\}_{kl})$$

$$k,l \subset M_{ij}$$

are separable. This is easily seen from analysis of (3.2.12) of Example 3.1. Here, we can eliminate second derivatives using H^1_{ij} and H^2_{ij} and the equivalent equation $\tilde{\mathcal{N}}_{ij}=0$ is of the form:

$$f_{i,j+1}- 4f_{ij} + f_{i,j-1}+ f_{i+1,j}+ f_{i-1,j}= 0$$

As we can see, $\tilde{\mathcal{N}}_{ij}$ contains five unknowns, but since (3.2.14) is made of n x m equations with n x m unknowns f_{ij}, the system (3.2.14) is closed.

Systems (3.2.12) supplemented with Hermitian formulae of accuracy higher than 2, are generally nonseparable. In this case there is no known general algorithm for separation of the system (3.2.12). The separation has been obtained for certain special cases, like boundary layer equations (Hirsh [10]). Some researchers are vigorously investigating the problem of Separability for the class of Navier-Stokes equations (Dennis [5]). For our problem (3.3.2)-(3.3.3) we have not succeeded in separating higher order Hermitian formulae from the field equations.

The above discussion points out the important difference between the 2-nd and higher order methods. 2-nd order methods provide easy Separation, thus considerably reducing computational efforts. In the case of higher order methods Separation is not assured and therefore, the cost of computations may become considerable. This issue will be discussed again in Chapter 6.

3.3 Alternate Direction Implicit Method of Solution.

The discretization of analytical problem $\mathcal{A}$ results in a system of linear equations (3.2.12), which usually is fairly large ((n x m)>> 1). If the system were small, the inversion of the matrix would be the appropriate way to have it solved. Since n x m is large, we must use an iterative scheme. There are many known schemes available, of which we have selected ADI. It is an implicit scheme, hence it is preferred, because of its stability properties. It also leads to tri-diagonal (scalar or block) algebraic systems which can be efficiently solved.

Let us assume that system (3.2.12) has an algebraic form:

$$A \underline{v} = \underline{s}$$

where A is a matrix of coeffitients of the system; $\underline{v}$ is a vector of unknowns and $\underline{s}$ is a vector known from governing equations and boundary conditions. To organize the ADI scheme, we must first subdivide A into the matrices, so that

$$A = Q + W + B \qquad (3.3.1)$$

where B might be a null matrix, matrix Q is related to the horizontal iterative sweep and W to the vertical sweep. A single ADI iteration consists of two half-steps defined as follows:

$$\underline{v}^{k+\frac{1}{2}} = (Q + B + w_{kh} I)^{-1} (\underline{s} - (W - I w_{kh}) \underline{v}^{k})$$

$$\underline{v}^{k+1} = (W + B + w_{kv} I)^{-1} (\underline{s} - (Q - I w_{kv}) \underline{v}^{k+\frac{1}{2}}) \qquad (3.3.2)$$

Index k is the ADI iteration counter, fractional superscript indicate sweep of the algorithm in one direction, which must be followed by a sweep in the orthogonal directon to complete one single iteration. The $w_{k.}$ are relaxation parameters, which generally may vary along with the ADI advancement (subscript k) and may depend on the direction of ADI sweep (subscript h for horizontal direction, v for vertical direction). If we assume B=0, this is the well known Peaceman & Rachford scheme (Richtmeyer & Morton [19]).

Let us return to the problem (3.2.2)-(3.2.3). For simplicity let us assume that we have a fixed interface h(x) (not necessarily constant), and we seek a solution for the stream function. Equation (3.2.2), when discretized, takes on the form of a block:

$$\left\{ \psi_{xx} + \psi_{zz}(1+z^2h_x^2)/h^2 + \psi_z\, z(2h_x^2 - h_{xx}h)/h^2 \right\} \Big/ \text{at } (x_i,\ z_j) = \{\psi_{xz} 2zh_x/h\} \Big/ \text{at } (x_i,\ z_j)$$

$$H^1((j,i-1),\ (j,i),\ (j,i+1)) = 0 \qquad (3.3.3)$$
$$H^2((j,i-1),\ (j,i),\ (j,i+1)) = 0$$
$$H^3((j-1,i),\ (j,i),\ (j+1,i)) = 0$$
$$H^4((j-1,i),\ (j,i),\ (j+1,i)) = 0$$

where H^k, defined before by (3.2.4), have here an abbreviated notation in order to show indexes of the nodes of a specified grid molecule of three consecutive nodes centered around (x_i, z_j). Hermitian formulae H^k are to be independent. H^1 and H^2 relate values of ψ, ψ_x and ψ_{xx} in three consecutive points of a grid in x direction, while H^3 and H^4 relate ψ, ψ_z and ψ_{zz} in three points of a molecule in z direction. Mixed derivative, due to its particular difference representation, is calculated independently of the ADI half-steps (see for instance components of the higher order finite difference approximation of the mixed derivative (A.4) in Anex).

The partition (3.3.1) of a matrix of a system (3.2.12) can be arranged by:

(a) Setting B=0.

(b) To get matrix Q: leave in matrix A only those terms which are associated with ψ, ψ_x , and ψ_{xx}.

(c) To get matrix W: leave in matrix A only those terms which are associated with ψ, ψ_z , and ψ_{zz}.

The partitioning, after some rearrangements, results in a block tri-diagonal structure of matrices $(Q+Iw_{kh})$ and $(W+Iw_{kv})$ of (3.3.2). Tri-diagonal structure results from line-iterative scheme, in which all unknowns along a given line (x or z coordinate) are solved for simultaneously. Since the governing equation is expressed at each pivotal point by algebraic formulae involving values of variables at this point and its right and left neighbouring points, the expressions at left and right neighbours constitute lower and upper diagonal, while expressions at the central point of the molecule constitute the central diagonal. If we apply in (3.3.3) Hermitian formulae of the second order, like (3.2.8) and (3.2.9), we may simplify the matrix structure even further, by Separation. In effect, we get system (3.3.2) of much smaller dimensionality and of scalar tri-diagonal structure. If Separation cannot be achieved, the resulting system of equations (3.3.2) is of 3x3 block tri-diagonal form. Algorithms for solving both kinds of tri-diagonal systems are well known (Thomas Algorithms).

It can be shown, that for the ADI scheme (3.3.2) the iterative error $\underline{e}^k = \underline{v}^k - \underline{\tilde{v}}$ ($\underline{\tilde{v}}$ is exact algebraic solution), is:

$$\underline{e}^k = \prod_{i=1}^{k} Z\, \underline{e}^o \qquad (3.3.4)$$

where

$$Z = (W+w_{kv}I)^{-1}(Q-w_{kv}I)(Q+w_{kh}I)^{-1}(W-w_{kh}I) \qquad (3.3.5)$$

Assuming that Q and W have constant coefficients (for our flow problem this would mean fixed shape of the free surface h(x)), Z is only a function of w_{kv} and w_{kh}. Therefore the question that arises is: how to determine a set of parameters {w} so that the spectral radius of Z is minimized. In other words, how to establish {w} to assure the fastest convergence. This problem has been formulated and succesfully solved by Wachspress [28]. The solution is restricted however to the 'Model Problem Conditions':

(a) The real matrices Q and W are symmetric, and A=Q+W is positive definite,

(b) Q and W commute, i.e. QW = WQ.

These conditions are not in general satisfied by our problem (3.2.2)-(3.2.3). Nevertheless, it was suggested by Wachspress, that even though conditions (a) and (b) are violated, the algorithm that he developed to determine optimal (w_{kv}) and (w_{kh}) may provide relaxation parameters which cause faster convergence compared to the single fixed value of w. The Wachspress algorithm has been implemented in the 2-nd order algorithms presented in this work. Results of a particular study of the effects of optimization are presented in Sec. 4.2.1.

3.4 Analysis of Discretization Error

Finite difference discretization of a differential equation consists of approximating each derivative by a finite difference expression. Every such approximation is biased by the error which depends on grid spacing and on the behaviour of a solution in vicinity of pivotal points of the grid. In consequence the finite difference discretized approximation of a differential equation is characterized by a superposition of all particular errors. Numerical algorithm based on discretization may introduce also some other errors. Therefore we deal with two different issues:

- the error of difference approximation of derivatives which constitute a differential equation,
- the global error of the algorithm solving the discretized equation.

In this section we intend to examine in details the global error, in particular its convergence rate with the grid refinement (grid-convergence property) and its practical implications. We will work out, among others, a criterion for selecting a grid-size to keep the global error dependent on the grid spacing only and we will discuss the possiblity of obtaining highly accurate results through application of the concept of extrapolation.

In any numerical solution of an analytical problem we deal with three distinct mathematical entities:

f_a : analytical solution, which is usually unavailable in an explicit form,

f_N : exact solution of a numerical model of the discretized problem (which is also unknown),

f : actual computed solution, which is a certain approximation of f_a.

We may define discretization error as $[f_N - f_a]$ and global rounding error as $[f_N - f]$.

Any algorithm which is supposed to yield a reliable numerical solution, should posses two properties, i.e. stability and convergence. Stability is assured when cumulative effect of error of all the roundings is negligible, that is, when in the effect of the computational process the difference $[f_N - f]$ is kept small compared to discretization error.

In our approach we model the governing differential equations with a set of finite-difference equations resulting from discretization process. The solution algorithm is convergent by definition if $[f_N - f_a] \longrightarrow 0$ with Δx, $\Delta y \longrightarrow 0$. This property will be called hereafter <u>grid-convergence</u> to avoid confusion with the concept of convergence of the iterative process utilized to solve the system of algebraic equations resulting from discretization. In this section we intend to elaborate on the problem of grid-convergence, which due to its importance requires a separate discussion.

In computational practice stability means, that an algorithm provides some solution; grid-convergence means that solutions computed on grids of decreasing spacing, form a Couchy - convergent sequence with the rate of convergence depending asymptotically on some power-function of the grid spacing. The conjecture is that this sequence is convergent to the exact solution of the analytical problem.

Let us consider now, how we can analyse the above stated grid-convergence property. In the following discussion all arguments will be presented in the context of second order finite difference approximations, but it is understood that they are applicable to approximations of any other order.

As was shown in Sec. 3.2, the difference model of the differential equation is a superposition of difference formulae involving values of function and its derivatives at pivotal points of a grid. For simplicity, let us consider the following operator :

$$\begin{aligned} d(f_{i+1},f_{i-1},f_i',r) &= f'(x_i) - \frac{1}{2r}\,[f(x_{i+1})-f(x_i\ )] \\ &= -\frac{r^2}{6}[f'''(\theta_1)+f'''(\theta_2)] \end{aligned} \qquad (3.4.1)$$

with arguments $\theta_1 \in (x_i,x_{i+1})$ and $\theta_2 \in (x_{i-1},x_i)$, where $x_{i\pm1} = x_i \pm r$. The last formula of the above expression is a superposition of Lagrange residuals of appropriate Taylor expansions of f_{i-1} and f_{i+1} around f_i. If we set r sufficiently small, we can make variations of the third derivatives negligible, and the Langrange residual in (3.4.1) becomes in effect proportional to r^2 This means that

$$\frac{|d(f_{i+1},f_{i-1},f_i' \ , r_1)|}{|d(f_{i+1},f_{i-1},f_i' \ , r_2)|} = \frac{|r_1^2 f'''(\theta_1)|}{|r_2^2 f'''(\theta_2)|} \cong \frac{r_1^2}{r_2^2} \quad . \qquad (3.4.2)$$

Formula (3.4.2) is reflecting proportionality of d to r^2 and leads to the conclusion, that $d \to 0$ as $r \to 0$, which is the grid-convergence condition. Power of the ratio r_1/r_2 is by definition the grid-convergence order of the difference scheme used to approximate derivative in (3.4.1).

We would like to note that:

a) Residual involving r^2 in (3.4.1) is of a local character, since it comes from the Taylor expansion around a fixed point x_i.

b) f_i and f_i' are exact values of a function and its derivative at pivotal points of a grid ($f_i = f(x_i)$).

Locality is not a big concern, in spite of the fact that the higher order derivatives associated with r may vary from point to point. In the limit, as $r \longrightarrow 0$ they approach a constant and their ratio in the vicinity of the fixed point is close to 1. Observation b) however , is a perplexing one, since we never deal with the exact values, but rather with their numerical approximates.. Practically we equate difference formulae like (3.4.1) to zero, consciously entering the error in the form of Lagrange Residual hoping that this error will not be worse, i.e. of a lower order for any of the components of the difference formula. If it were, then by contraposition it would violate the accepted observation that the operator d goes to zero at a quadratic rate. This is an essential statement, since it makes possible a conclusion about the order of function and its derivatives on grounds of the order of the Difference Scheme.

Now we can approach the problem of grid-convergence of the finite-difference algorithm. Let us assume, that in a general case, when f is a function of two variables, the grid consists of (n x n) pivotal points uniformly distributed in a computational domain (i.e. $\Delta x = \Delta z = r_n$). Let f^n be a numerical solution computed on this kind of grid determined by the specification of r_n. A necessary and sufficient condition for the algorithm to be grid-convergent is that

$$[f_a - f_N] \longrightarrow 0 \quad \text{as} \quad r \longrightarrow 0$$

or in a weaker form:

$$[f_a - f^n] \longrightarrow 0 \quad \text{as} \quad r_n \longrightarrow 0 \tag{3.4.3}$$

The difficulty we face at this point is, that we are unable to provide any analytical criterion which, for a particular algorithm, would yield evidence that condition (3.4.3) is satisfied. Therefore we are bound to look for a criterion which is not equivalent to the definition of grid-convergence, but gives sufficiently strong support for the statement that (3.4.3) is valid. At this point we may make use of the concept of the Order of difference scheme and remarks made earlier. The criterion which we can practically accept as evidence of the grid-convergence property is as follows:

$$\frac{[f_a - f^{n_1}]}{[f \;\; - f^{n_2}]} = \frac{(r_{n_1})^{\alpha}}{(r_{n_2})^{\alpha}} \quad \text{as } n_1, n_2 \longrightarrow \infty \qquad (3.4.4)$$

Let us make the following observations here:

a) α is by definition the Order of Grid-Convergence of an algorithm.

b) Formula (3.4.4) says that the numerical solution approaches the exact one as we perform computations on finer and finer grids ($n_i \rightarrow \infty$).

c) Condition (3.4.4) implies that $\{f^n\}$ is the Cauchy convergent sequence, which for practical purposes may be understood as convergent to f_a.

d) The Practical test of (3.4.4) is always carried out on a finite set of grids.

e) Since (3.4.4) is derived from the concept of the order of error of the truncated Taylor expansion around the pivotal point of a grid, it originally applies to the point values of the computed and exact solutions, with norm $[.]$ being a simple modulus. In publications like for example Aubert & Deville [1] we may find this kind, i.e. point-estimation approach. It is understood, however, that global estimation of order, applied to the computational domain is more valuable than the information about the grid-convergence order at few selected pivotal points of a grid.

If we pursue the global estimation, the norm $\|.\|$ might be the Euclidean-type norm on (n x n) dimensional space. α - which is the exponent in (3.4.4) may be of any positive value. In fact, as will be demonstrated, α is about the same as the Order of a differencing scheme implemented in a given algorithm. So, if for some algorithm we apply differences of the second order, then the Order of grid-convergence of this algorithm should also be 2. In Sec. 5.6.5 we present numerical results, which illustrate this statement.

Approximate Estimate

of the Order of Grid-convergence

It is very uncommon in numerical modelling, that an exact analytical solution is known. Practically, however, we might be able to simplify the original problem, so that an exact solution is available, and estimate the order of grid-convergence with the help of (3.4.4). Nevertheless, this will be only a partial test for the original algorithm. If we do not have an analytical solution, the following criterion, known from literature (Aubert & Deville [1]) can be applied:

$$\frac{\|f^{n_1} - f^{n_2}\|}{\|f^{n_2} - f^{n_3}\|} - \frac{r_{n_1}^{\alpha} - r_{n_2}^{\alpha}}{r_{n_2}^{\alpha} - r_{n_3}^{\alpha}} \qquad \text{as } n_1, n_3, n_2 \longrightarrow \infty \qquad (3.4.5)$$

If the above condition is fulfilled, α is by definition the Order of

Grid-Convergence of the algorithm. For clarity, let us assume here, that the computational grid is of equal spacing in x and z direction, $[f^n_{ij}]$ being the matrix of values of function f computed on grid with $\Delta x = \Delta z = r$.

Equation (3.4.5) can be easily referred to (3.4.4) if f degenerates to a scalar and $\|.\|$ is a modulus. In such a case we have equivalence of both conditions provided that a unique analytical solution f exists. In general case, because norm $\|.\|$ is not necessarily additive, we cannot conclude the validity of (3.4.4) from (3.4.5) or vice versa. It will be shown in the following discussion that when some special requirements are imposed on the norm, we may relate directly point-wise order of grid-convergence with its global estimate. In Sec. 4.2.2 we present numerical results, which show experimental (numerical) equivalence of (3.4.4) and (3.4.5). This may authorize us to use (3.4.5) also for cases, where (3.4.4) cannot be applied due to unavailability of an exact solution.

It happens in computational practice, that exponent α changes initially when we move computations from coarser to finer grids. Then it gradually stabilizes and becomes constant. This means that results obtained on coarser grids show uncontrollable discretization error due to the variation of higher derivatives in the Lagrange residual. Stabilization of the value of α means that the grid is fine enough to make Lagrange residuals dependent on r only; therefore, we may assume that the disretization error is proportional to a certain power of r. Another problem is the value of α . It was shown that α should be close to the discretization order. If it is not, it may mean that there are some errors in the algorithm, which may result either from inadequacy of the method or, more trivially, from mistakes of a programmer. This issue is illustrated in examples at the end of this Seotion.

In the following discussion a proper choice of a norm will be analysed. We confine our attention to the L-2 type norm, since we deal with discretized values of the function. These values are defined at pivotal points of a grid Gn which is of dimension (n x n). The L-2 norm may be defined as follows:

$$\| f \|(Gk) = \frac{1}{k^2}\Big(\sum_{(x_i, z_j) \in Gk} f^2_{ij} \Big)^{1/2} \qquad (3.4.6)$$

where Gk is any subgrid of Gn (Gk$\subset$Gn). In particular, Gk may be a single point of a grid Gn and in this case the norm is reduced to a modulus.

Since in determination of the order of the grid-convergence of the algorithm we work with finer and finer grids, we can see that it would be useful to generate grids in such a way, that coarser grids are embedded in finer ones. This seems obvious when looking at (3.4.6). If we consider (3.4.5), we see, that f^{n_1}, f^{n_2} and f^{n_3} are computed on grids with decreasing spacing $r_{n_3} < r_{n_2} < r_{n_1}$. The question arises how to define norms in nominator and denominator on the left hand side of (3.4.5). Certainly, arbitrary application of (3.4.6) may lead to incorrect results. In order to explain this issue, let us introduce the concept of order-consistency of a couple of norms $\|.\|(Gk)$ and. $\|.\|(Gl)$. Before we define this concept assume that $Gn_1 \subset Gn_2 \subset Gn_3$ and also $Gk \subset G_{n_2}$, $Gl \subset G_{n_1}$

Definition

The pair of norms $\|.\|(Gk)$, $\|.\|(Gl)$ is order-consistent if and only if

$$\frac{|f^{n_3}_{ij} - f^{n_2}_{ij}|}{|f^{n_2}_{ij} - f^{n_1}_{ij}|} = \frac{r^{\alpha}_{n_3} - r^{\alpha}_{n_2}}{r^{\alpha}_{n_2} - r^{\alpha}_{n_1}} \Longrightarrow \frac{\|f^{n_3} - f^{n_2}\|(Gk)}{\|f^{n_2} - f^{n_1}\|(Gl)} = \frac{r^{\alpha}_{n_3} - r^{\alpha}_{n_2}}{r^{\alpha}_{n_2} - r^{\alpha}_{n_1}}$$

Now, recalling definition (3.4.6), we may make the following statement:

Proposition 3.1

The pair of norms $\|.\|(Gk)$ and $\|.\|(Gl)$ is order-consistent if and only if the subgrids Gk and Gl are identical.

This is not a trivial proposition, since if we try to make a practical use of the criterion (3.4.5), we may easily make an error by utilizing inconsistent norms: $\|.\|(Gn_2)$ for nominator and $\|.\|(Gn_1)$ for denominator. The above statement demands that subgrids used to define norms of nominator and denominator of formula (3.4.5) must be the same.

The idea of the proof is based on a simple statement that if point-wise grid-convergence is of order α for every single value f_{ij} , so the aggregated grid-convergence should be also of order α .

Evidence of the Proposition can be demonstrated as follows. Let Gk and Gl be subgrids of dimensionality k x k and l x l, respectively. Let us assume (see Def.3.1) that at each point of matrices $[f_{ij}^{n_1}]$, $[f_{ij}^{n_2}]$ and $[f_{ij}^{n_3}]$ we have:

$$\frac{f_{ij}^{n_3}}{f_{ij}^{n_2}} = \frac{r_{n_3}^{\alpha}}{r_{n_2}^{\alpha}} \qquad\qquad \frac{f_{ij}^{n_2}}{f_{ij}^{n_1}} = \frac{r_{n_2}^{\alpha}}{r_{n_1}^{\alpha}}$$

Let us consider first the ratio:

$$\frac{f_{ij}^{n_3}}{f_{ij}^{n_1}} = \frac{r_{n_3}^{\alpha}}{r_{n_1}^{\alpha}}$$

$$f_{ij}^{n_3} - f_{ij}^{n_2}(r_{n_3}/r_{n_2})^{\alpha} = 0$$

$$f_{ij}^{n_3} - f_{ij}^{n_2} + f_{ij}^{n_2} - f_{ij}^{n_2}(r_{n_3}/r_{n_2})^{\alpha} = 0$$

and finally

(a) $$f_{ij}^{n_3} - f_{ij}^{n_2} = f_{ij}^{n_2}[(r_{n_3}/r_{n_2})^{\alpha}-1]$$

By similar argument:

(b) $$f_{ij}^{n_1} - f_{ij}^{n_2} = f_{ij}^{n_2}[(r_{n_1}/r_{n_2})^{\alpha}-1]$$

Suppose the difference ratio is of the form:

$$\frac{[\![f^{n_3} - f^{n_2}]\!](Gk)}{[\![f^{n_2} - f^{n_1}]\!](Gl)}$$

For nominator we have: $[\![f^{n_3} - f^{n_2}]\!](Gk) = \frac{1}{k^2} \left(\sum (f_{ij}^{n_3} - f^{n_3}.)^2 \right)^{\frac{1}{2}} =$

$$= \frac{1}{k^2} \left(\sum (f_{ij}^{n_2}) \left[\left(\frac{r_{n_3}}{r_{n_2}} \right)^{\alpha} - 1 \right]^2 \right)^{\frac{1}{2}} = \frac{1}{k^2} \left[\left(\frac{r_{n_3}}{r_{n_2}} \right)^{\alpha} - 1 \right] \left(\sum (f_{ij}^{n_2})^2 \right)^{\frac{1}{2}}$$

and similarly for denominator. Evaluating ratio of both expressions one gets:

$$\frac{[\![f^{n_3} - f^{n_2}]\!](Gk)}{[\![f^{n_2} - f^{n_1}]\!](Gl)} = \left[\frac{r_{n_3}^{\alpha} + r_{n_2}^{\alpha}}{r_{n_2}^{\alpha} - r_{n_1}^{\alpha}} \right] \left[\frac{l^2 \left(\sum_{Gk} (f_{ij}^{n_2})^2 \right)^{\frac{1}{2}}}{k^2 \left(\sum_{Gk} (f_{ij}^{n_2})^2 \right)^{\frac{1}{2}}} \right]$$

Here we see that the norms in the difference ratio are order-consistent if the expression in square brackets of the last term is equal to 1. In the general case this condition is satisfied only if l = k, which completes the proof.

Conclusions

1. The Proposition 3.1 is valid if we compare the estimated solution f^{n} and the exact one, i.e.

$$\frac{[\![f^{n_3} - f_a]\!](Gk)}{[\![f^{n_2} - f_a]\!](Gk)} = \frac{r^{\alpha}_{n_3}}{r^{\alpha}_{n_2}}$$

2. Gk can be any grid insofar as it can be embedded in Gn_1, Gn_2, Gn_3.

3. In particular, Gk can be a single point. In this case the norm is reduced to the modulus and α^* is a point-wise order of grid-convergence.

The Proposition 3.1 yields practical suggestions on how to obtain the global estimate of order of grid-convergence. As was mentioned earlier, this estimate is of vital importance if we want to make sure that the numerical algorithm provides solutions which are convergent to analytical solution. Practically it may also provide some indication as to how fine the grid should be in order to obtain results which are not biased by an excessive error of discretization. It is often very inconvenient to investigate the order of convergence point by point. In some papers (Elsaesser and Peyret [7]) one can find point - wise order estimation for a few selected points of a grid (e.g. in the centre of the solution domain). Since information about other points is not available, the argument about order of convergence is of little value. Global order estimate on the other hand, provides aggregate information about the function in the entire domain. But if we do not satisfy the conditions imposed by Proposition 3.1, we may end up with wrong conclusions. Because it is not a trivial issue, we want to illustrate it in the example below.

Suppose we want to solve the following boundary-value problem:

$$f'' + 2f' = 4(\pi\cos 2\pi x - \pi^2 \sin 2\pi x) \qquad (3.4.7)$$

$$f(0) = f(1) = 0, \quad x \in [0,1]$$

The second order difference scheme, which would simulate left-hand side differential operator, is of the form:

$$f_{i-1}\left(\frac{1}{r^2} - \frac{1}{r}\right) - \frac{2f_i}{r^2} + f_{i+1}\left(\frac{1}{r^2} + \frac{1}{r}\right) + O(r^2) = f_i'' + 2f_i'$$

The solution was computed on grids G10, G20, G40, G80, where Gk in this case is a grid of dimension k. The respective solutions are denoted as f^{10}, f^{20}, f^{40}, f^{80}. There is an exact solution available: $f_a = \sin(2\pi x)$. Results of the analysis of the accuracy of discretization are presented in Tables 3.1-3.4.

TABLE 3.1 : Pointwise estimation of the order of grid-convergence of discretization error

x_i grids	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
G10/20/40	2.03	2.03	2.04	2.04	2.06	1.85	2.0	2.02	2.02
G20/40/80	2.01	2.01	2.01	2.01	2.02	1.97	2.0	2.00	2.01

TABLE 3.2 : Aggregate estimation of the order of grid-convergence with the use of three grids (Example (3.4.7))

Difference Ratio	$\frac{[f^{10}-f^{20}](G10)}{[f^{20}-f^{40}](G10)}$	$\frac{[f^{20}-f^{40}](G10)}{[f^{40}-f^{80}](G10)}$	$\frac{[f^{20}-f^{40}](G20)}{[f^{40}-f^{80}](G20)}$
α: order estimated	2.03	2.01	2.01

TABLE 3.3 : Aggregate estimation of the order of grid-convergence with the use of three grids (Example (3.4.7)) and norms with inconsistent reference grids .

Difference Ratio	$\frac{\|f^{20}-f^{40}\|(G10)}{\|f^{40}-f^{80}\|(G20)}$	$\frac{\|f^{10}-f^{20}\|(G10)}{\|f^{20}-f^{40}\|(G20)}$	$\frac{\|f^{20}-f^{40}\|(G20)}{\|f^{40}-f^{80}\|(G40)}$
α: order estimated	3.69	3.66	3.58

TABLE 3.4 : Aggregate estimation of order of grid-convergence by reference to the exact solution (Example (3.4.7)).

Difference Ratio	$\frac{\|f^{10}-f_a\|(G10)}{\|f^{20}-f_a\|(G10)}$	$\frac{\|f^{20}-f_a\|(G10)}{\|f^{40}-f_a\|(G10)}$	$\frac{\|f^{40}-f_a\|(G10)}{\|f^{80}-f_a\|(G10)}$
α: order estimated	2.03	2.01	2.00

Comments:

1. Tables 3.1, 3.2, 3.4 yield very similar results. This means that point-wise and aggregate estimation of the order of convergence are numerically equivalent.

2. Exponent α is approaching the value of 2 which is the order

of the applied difference scheme.

3. $\alpha \longrightarrow 2$ as discretization of the domain of the independant variable becomes finer.

4. In Table 3.1 the order at the point x=0.6 is the lowest one. This may indicate that variation of higher derivatives in the Lagrange residual is still considerable. The other suggestion arising from this fact is that results obtained on a grid of r=0.1 may not be sufficiently accurate and one should pursue computations on grids of $r \leq .05$.

4. Table 3.3 shows an essential discrepancy from what we find in Tables 3.1, 3.2 and 3.4. This is due to the fact, that the norms are not order-consistent. This is an evident illustration that one may end up with very misleading results if inaopropriate norms are used.

Applications of the grid-convergence property.

If the computational algorithm is of iterative nature, there is always a question about the criterion of termination of the iterative process. Usually there are two indicators of advancement of iterations. These are: change of the field from one iteration to the next iteration and the value of the residuum (which says how accurately the discretized governing equations are satisfied). Both criteria are expressed as absolute quantities. Ideally, we may iterate as long as these quantities are of the order of computer accunacy. This, however, is very costly, and practically we terminate iterations sooner. To make sure that assumed level of accuracy is sufficient, we may compare results computed on grids of decreasing spacing r. If the grid-convergence order is found to be of expected value, we may accept the assumed limit of iterative process as sufficient.

This criterion can be justified by referring to the Lagrange residual of Taylor expansion. Namely, if the order of the grid-convergence is close to the order of discretization, it means that variation of the higher order derivatives in Lagrange residuals is negligible. If so, further iterations do not have any significant influence on the values of higher derivatives and in consequence on the computed estimates of the function.

As was mentioned before, the analysis of the accuracy of the discretization may be helpful in detecting errors which otherwise ~~may~~ pass unnoticed. These errors may result from inadequacy of the algorithm, or they may occur on the level of the logical structure of the algorithm, or its implementation. Some errors can be fatal, but some are difficult to detect and remain unnoticed, because no evident obtrusion in the algorithm's performance is observed. In what follows we present results of the analysis of accuracy for the problem discussed in the last Example. The only difference is that there is an error introduced by changing a single index in the Thomas Algorithm.

Pointwise order estimate is as follows:

TABLE 3.5 : Pointwise estimation of the order of grid-convergence when computed solution is erroneous .

x_i / grids	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
G10/20/40	.42	.95	.96	.93	.94	.99	1.09	1.22	1.33
G20/40/80	1.22	1.16	1.11	1.07	1.07	1.09	1.14	1.21	1.30

TABLE 3.6 : Aggregate estimation of the order of grid-convergence with the use of three grids (Example (3.4.7)) when computed solution is erroneous.

Difference Ratio	$\frac{[f^{10}-f^{20}](G10)}{[f^{20}-f^{40}](G10)}$	$\frac{[f^{20}-f^{40}](G10)}{[f^{40}-f^{80}](G10)}$	$\frac{[f^{20}-f^{40}](G20)}{[f^{40}-f^{80}](G20)}$
α: order estimated	0.93	1.12	1.12

TABLE 3.7 : Aggregate estimation of the order of grid-convergence with the use of three grids (Example (3.4.7)) and norms with inconsistent reference grids.

Difference Ratio	$\dfrac{[f^{20}-f^{40}](G10)}{[f^{40}-f^{80}](G20)}$	$\dfrac{[f^{10}-f^{20}](G10)}{[f^{20}-f^{40}](G20)}$	$\dfrac{[f^{20}-f^{40}](G20)}{[f^{40}-f^{80}](G40)}$
α: order estimated	2.59	2.71	2.69

Tables 3.5 and 3.6 show much smaller α than the expected value of 2. On the other hand Table 3.5 gives evidence of rising values of α when finer grids come into play. This is the right trend, but in this case it can lead to the false conclusion that all we need to do is to use a finer grid. Definitely it is not a matter of a grid size, since as we may see, computations on a grid of 80 points should be quite sufficient for the function which is the solution to the example problem. This is rather evidence of having an error in the program.

Table 3.7 shows estimate of the order when the norms are inconsistent. These results are higher than the expected value of 2. Theoretically it is not alarming. The discretization error is a result of a superposition of errors, each coming from a distinct difference estimator. Thus the joint discretization error can be lower than the component errors, since some of these component errors, when added together, may vanish. In consequence, the overall error may be of a higher order than expected. Therefore, as we may see, the results shown on Table 3.7 are very misleading. They are computed using an erroneous algorithm, but still, beacause of the application of the inconsistent norms, we may not realize that the numerical estimate of the unknown function is wrong.

Extrapolation.

It is known (Smith [27]) that the knowledge of the order of grid-convergence may lead to a highly accurate approximation of the unknown analytical solution. This may be accomplished by extrapolation provided that we have numerical solutions computed on different grids. Let us briefly state what in this case is understood by extrapolation. Suppose that there are available numerical results on three grids: Gl, Gm, Gn, and for each point of the embedded reference grid Gk, the following is valid:

$$\frac{|f^{l}_{ij} - f^{m}_{ij}|(Gk)}{|f^{m}_{ij} - f^{n}_{ij}|(Gk)} = \frac{r^{\alpha}_{l} - r^{\alpha}_{m}}{r^{\alpha}_{m} - r^{\alpha}_{n}}$$

α is constant and does not depend on x_i, the pivotal point of grid Gk. The above means, that f^{l}_{ij}, f^{m}_{ij} and f^{n}_{ij} lie along a straight line as shown on Fig.3.3 below. Therefore, we conclude that the analytical solution is located on the same line (dotted line) and can be determined as r → 0.

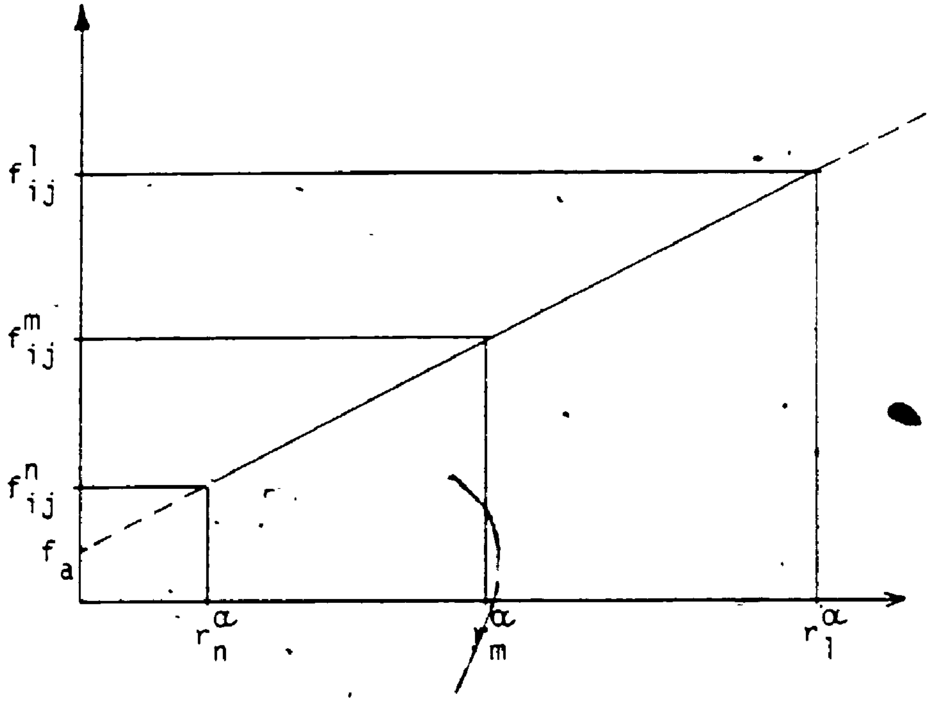


Fig. 3.3 Hypothethical convergence of the computed values to the exact solution with mesh refinement.

Let us now focus on the effectiveness of extrapolation. For illustration we take the boundary value problem defined in example stated by (3.4.7). We take the grid of r = 0.1 as the refernce grid Gk. In Table 3.8 we show results for $x_i = 0 + ir$. The first three rows show the distance of the exact solution from results extrapolated from pairs of grids. In the last row, for comparison, the distance of the solution computed on a very fine grid G80 from the exact solution is shown.

TABLE 3.8 : Distance of the exact solution from the extrapolated one and from the solution obtained on a grid G80.

x_i grids used in extrapolation	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
G10;G20	.12E-3	.24E-3	.31E-3	.31E-3	.22E-3	.99E-4	.19E-4	.89E-4	.81E-4
G20;G40	.76E-5	.15E-4	.19E-4	.19E-4	.14E-4	.60E-5	.13E-5	.56E-5	.50E-5
G40;G80	.47E-6	.93E-6	.12E-5	.12E-5	.85E-6	.37E-6	.85E-7	.35E-6	.31E-6
$\lvert f_i^{80}-f_i^{a}\rvert$(G10)	.36E-3	.64E-3	.73E-3	.6E-3	.3E-3	.61E-4	.34E-3	.43E-3	.32E-3

Among results computed on grids G10, G20, G40, G80, those obtained on the last one are certainly closest to the analytical solution. As is seen from the last row of the table, the error is not larger then 0.1% of the maximum value of $f_a = \sin(2\pi x)$. Yet the extrapolation yields better results for all cases listed in the table. This is the case even with the extrapolation based on results taken from computations on grids G10 and G20.

Let us close this Section by summarizing important developments. In this Section we have examined the concept of Accuracy of Discretization, as expressed in terms of the order of the grid-convergence. We have addressed the problem of practical computations and stressed the requirement of the proper choice of particular norm L-2 to get useful results. Three important reasons for the order estimate were discussed. The first one is the opportunity to detect errors in the algorithm. The second one is the determination of the maximal grid size, which can provide reliable results. Grids sparser than this one, when involved in the order computation, lead to inconsistent results. There is no sharp criterion, though. In the case of example 3.1 (Tab. 3.1) we may have doubts about results obtained on grid G10 (point-wise order estimation); but one may find them acceptable if we look at the aggregate result (global estimation - Tab. 3.2), which is very close to the expected value of 2. Finally, the availability of Extrapolation as means to obtain highly accurate results with moderate cost was discussed. Sample computations for the problem defined by (3.4.7) indicate that there is a great potential for cost-saving and for gaining high accuracy by the application of Extrapolation. Needless to say,. however, the extrapolation can be used only after the examination of the order of grid-convergence has been made, and only those results which were computed on sufficiently fine grids can be utilized.

The discussion has been carried out in the context of the Problem defined by (3.4.7), which is an ordinary differential equation with Dirichlet boundary conditions. It should be noted, however, that the general concepts and remarks are valid for any algorithm based on finite difference discretization and aimed at solving partial as well as ordinary differential equations.

CHAPTER 4

SMALL DEFORMATION THEORY

4.1 Construction of the solution in the asymptotic limit of a small capillary number ($C \to 0$).

Let us recall the mathematical model of the flow that we are interested in:

governing equations:

$$\nabla^2 \psi = 0 \tag{4.1.1}$$

$$\frac{h_{xx}}{(1 + h_x^{\,2})^{3/2}} = - p + C\left(\left(\frac{\partial\psi}{\partial x}\right)^2 + \left(\frac{\partial\psi}{\partial y}\right)^2\right), \tag{4.1.2}$$

boundary conditions for functions $\psi(x,y)$ and $h(x)$: (4.1.3)

$$\begin{array}{ll} \psi = 0 & \text{at } x = 0 \\ \psi = 0 & \text{at } x = 1 \\ \psi = -\cos(2\pi x) & \text{at } y = 0 \\ \psi = 0 & \text{at } y = h(x) \\ h = 1 & \text{at } x = 0 \\ h = 1 & \text{at } x = 1 \end{array}$$

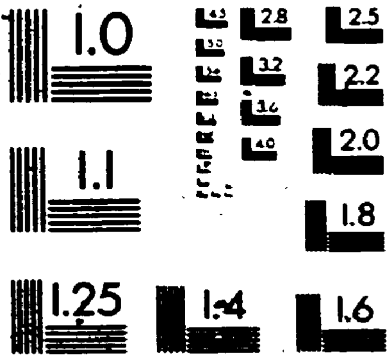
where $[-(\partial\psi/\partial x), (\partial\psi/\partial y)] = \underline{V}$ is the velocity of the fluid at the free surface (Eq.4.1.2).

The solution of (4.1.1) - (4.1.3) consists of determination of the two dependant variables: stream function ψ and function h describing the shape of the free surface. The main feature of the Asymptotic Algorithm is decoupling of the flow field and interface, so that they can be determined separately in a sequence of algorithmical steps. Direct Algorithms on the other hand, will seek to determine the flow and interface simultaneously. The distinction between these two types of algorithms is not merely of a logical character. It will be shown in this Chapter that the Asymptotic Algorithm is applicable to a specific class of flows only, i.e. when surface tension has a decisive influence on the shape of interface ($C \ll 1$).

The asymptotic method is closely related to the physical understanding of the analysed flow phenomenon. Let us assume that parameters C and p_s are fixed. First, let us consider a static case (no motion). Stagnation pressure becomes a static pressure and free surface, if it exists, takes on the form of the segment of a circle (see Sec.2.2). This will happen also with any non-trivial flow field, but infinitely large surface tension. In both cases C is at its limiting value, i.e. C=0. In this limit, the shape of the interface, as can be seen from the system (4.1.1) - (4.1.3) above, decouples from the field equation and is determined from conditions of static equilibrium. The flow field can be determined subsequently by solving the field equations with the predetermined (and fixed) location of the interface.

When $C \ll 1$, the shape of the boundary is dominated by the surface tension and dynamic effects are of the lower order of

2



magnitude. In this approach, therefore, we seek a solution which, to the leading order of approximation, reduces to the case C=0. Let us note, that large surface tension combined with curvature of the free surface, generates a large static pressure component, which dominates the normal forces caused by the motion of the fluid. In consequence, flow of the fluid results in a small deformation of the free surface.

We use asymptotic series expansion of the variables ψ, h and velocity to construct the solution. The velocity term is retained for convenience in order to simplify discussion of the dynamic boundary condition.

$$\psi = \psi_o + \epsilon \psi_1 + O(\epsilon^2)$$

$$h = h_o + \epsilon h_1 + O(\epsilon^2) \tag{4.1.4}$$

$$\underline{V} = \underline{V}_o + \epsilon \underline{V}_1 + O(\epsilon^2)$$

Velocity in Eq.(4.1.4) is the vector $[u(x,h(x)),v(x,h(x))]$. Let us introduce the surface velocity scale defined as

$$V_s = \max \{ (u^2(x,h(x)) + v^2(x,h(x)))^{\frac{1}{2}} \;, x \in [0,1]\}. \tag{4.1.5}$$

The surface velocity of the fluid can be expressed as follows: $\underline{V} = V_s\underline{V}_s$ where $\underline{V}_s$ is scaled surface velocity. Since $\underline{V}$ is of order 1, we can define a parameter ϵ as $CV_s^2/2$.

Relation between C·V and ϵ is an important one. The Small Deformation Theory , which makes use of asymptotic expansions (4.1.4) is valid if results are biased by the error of order $O(\epsilon^2)$. This happens, as will be demonstrated, if C is sufficiently small. On the other hand, this theory is referred to the physical concept of small deformation of the free surface. Since this deformation depends on the value of the Capillary Number C and the velocity of the fluid at the free surface, the scaling (4.1.5) provides the necessary linkage between parameter ϵ of the asymptotic series expansion (4.1.4) and physical parameters of the flow: the capillary number C and the velocity scale V.

Substituting expansions (4.1.4) into equations (4.1.1)-(4.1.3) and equating coefficients of the same power of ϵ, we obtain the following sequence of problems:

Problem of order ϵ^0:

$$\frac{h_{oxx}}{(1 + h_{ox}^2)^{3/2}} + p_s = 0 \qquad (4.1.6)$$

$$\nabla^2\psi_o = 0 \qquad (4.1.7)$$

boundary conditions for functions $\psi_o(x,y)$ and $h_o(x)$: (4.1.8)

$\psi_o = 0$ at $x = 0$

$\psi_o = 0$ at $x = 1$

$\psi_o = -\cos(2\pi x)$ at $y = 0$

$\psi_0 = 0$ at $y = h(x)$

$h_0 = 1$ at $x = 0$

$h_0 = 1$ at $x = 1$

Problem of order C :

$$h_{1xx}\frac{1}{(1+h_{0x}^2)^{3/2}} - h_{1x}\frac{3h_{0xx}h_{1x}}{(1+h_{0x}^2)^{5/2}} = |\underline{V}_{so}|^2 \qquad (4.1.9)$$

boundary conditions:

$h_1 = 1$ at $x = 0$

$h_1 = 1$ at $x = 1$

Equations governing ψ_1 and higher order terms in the asymptotic series (4.1.4) could be derived in a similar way. The present analysis is limited to the determination of the leading order approximation of the flow field and surface deformation only.

In the first problem (of the order ε^0) we solve first for static interface and then for the flow field which would develop if infinite surface tension was imposed. In the second one we solve the equation for correction of the free surface shape h_1. The amplitude of the actual correction εh_1 depends on the magnitude of the model parameter C and the velocity of the fluid along the static interface. Both factors contribute to the value of the new model parameter: ε (see 4.1.5). It is to be remembered, that this algorithm is justified only for small ε. It is one of the objectives of this chapter to find out what is the range of ϵ, for which the method is valid. The answer to this question will be provided through analysis of the quadratic accuracy of the truncated asymptotic series (see (4.1.4)).

The sequence of computational steps leading to the determination of the field and free surface through application of the Small Deformation Theory, will be referred to as the Asymtotic Algoritm. It is schematized in the form of the flow chart given below:

SCHEME OF THE ASYMPTOTIC ALGORITHM

1. Select static pressure p_s and flow across the bottom of the boundary.

2. Determine shape of the static free surface h_0 by solving Eq. (4.1.6).

3. Find the first approximation of the flow in the domain by solving Eq. (4.1.7).

4. Determine surface velocity scale V_s.

5. Find free surface correction h_1 solving Eq. (4.1.9). Define final shape of the free surface as $h = h_0 + \epsilon h_1$.

The material presented hereafter consists of three sections. In the first one (Sec. 4.2) we describe in detail numerical procedures employed in this case; we are interested in the determination of static interface, field convergence characteristics and examination of order of grid-convergence for field and free surface. In Section 4.3 we address the issue of the qualitative effectiveness of the Asymptotic Algorithm. As was explained earlier, the validity of the asymptotic approach depends on the smallness of the C parameter. Therefore the attention is focussed on the criterion that determines whether the parameter is sufficiently small in order for the results to be valid. In Section 4.4 we discuss the property of the Algebraical Stability of the problem. Algebraic Stability requirements will be found to impose constraints on numerical parameters: r - spatial increment of a computational grid, and w - ADI relaxation parameter. In the last Section (4.5) we provide the physical interpretation of results. The discussion there will be focussed on a description of the variations of interface deformation caused by change of capillary number C, stagnation pressure p_s and scale of surface velocity V_s.

4.2 Discussion of the numerical algorithm

4.2.1 Iterative solution of the flow field and interface equations.

As shown on the flowchart of the preceding Section, the first step in the computational procedure is to find static interface h_0. This is done by solving the following equation:

$$\frac{h_{0xx}}{(1 + h_{0x}^2)^{3/2}} = - p_s \qquad (4.2.1)$$

$$|p_s| \;\; 2$$

$$h_0 = 1 \qquad \text{at } x = 0$$

$$h_0 = 1 \qquad \text{at } x = 1$$

If $p_s = 2$, the static surface forms the complete semicircle. In this case h_x is undetermined at the boundaries of the unit interval. Because $\max(|h_x|)$ is approaching infinity as p_s approaches the value ± 2, the numerical integration of Eq.(4.2.1) will be difficult if p_s is in vicinity of ± 2. In the following disucussion the static surface having the shape of the full semicircle (corresponding to $p_s = \pm 2$) will be referred to as the 'reference surface'. For a given case we may relate static surface to the reference one by the percentage of the semicircle area enclosed by the actual static surface curve. This estimate is used as a measure of how close is the given surface to the reference one. For instance 0% will mean, that the given surface is flat, 100%, that it is a full semicircle.

The Eq.(4.2.1) may be easily solved analytically, yielding the solution in form of a segment of a circle with constant curvature equal $-p_s$. However, for the sake of computational completeness, this equation was solved numerically. First it was linearized:

$$h_{oxx}^{n+1} + h_{ox}^{n+1}(h_{ox}^{n}(1 + (h_{ox}^{n})^2)^{\frac{1}{2}} p_s = -p_s(1 + (h_{ox}^{n})^2)^{\frac{1}{2}} \qquad (4.2.2)$$

where superscripts n,n+1 refer to subsequent approximations of the solution in iterative procedure. Equation (4.2.2) is discretized using central differences of the second order, and solved as a boundary value problem using block scheme resulting in 3-diagonal system of algebraic equations. For all cases considered (with $|p_s| < \simeq 1.85$) relaxation does not seem to be necessary to assure convergence. Below we present the table relating value of p_s to the number of iterations required. To avoid ambiguity, the iterations were halted when the change of h_o from iteration to iteration was of the order of machine accuracy. This yielded residuals of the free surface equation of about 1.E-10. Initial guess for the shape of interface was $h_o = 1$.

TABLE 4.1 Number of iterations for the static interface.

P	# its
1.5	15 (13)
-1.5	15 (13)
1.85	25 (17)
-1.85	25 (17).

Two numbers of iterations are presented for each p_s. The first one, when computations were done on a grid with spatial increment r=0.1, and the second one (in brackets) for r=1./60. Number of iterations for intermediate r values behave monotonically. Generally, with p_s increasing, the number of iterations required increases, which is due to stronger nonlinearity effects in Eq. (4.2.2). Generally it takes less iterations to converge on a finer grid.

FIELD CONVERGENCE

As was said in Sec. 3.2 , the iterative procedure is driven by pseudo-time derivative, or relaxation, which are equvalent. It was also said, that we intend to employ an optimal sequence of relaxation parameters, with which we expect the convergence to be fast. There are however conditions, which are sufficient to make Wachspress optimization effective. These conditions correspond to the so called Model Problem assumptions (Sec.3.2). In our case these assumptions are fulfilled in one case only, that is, when p_s= 0. When p_s=0, h is constant, equal to 1., and we can solve the model equation on a physical grid without the necessity of using transformation. But if $p_s = 0$, the transformation becomes non-trivial, the field equation loses its canonical Laplacian form, and the resulting algebraic scheme of ADI procedure no longer satisfies the Model Problem requirements. Practice shows however, that even though we depart from the Model Problem assumptions, we may still expect the optimal set of relaxation parameters to work effectively.

In our computations we determined the optimal sequence of relaxation parameters $w_1, \ldots w_k$ as if the free surface was flat ($h_o=1$). Subsequently, the iterations were performed in cycles, each cycle making use of the same sequence $\Omega = (w_1, \ldots, w_k)$. The following observations have been made:

1. The automatic set-up of relaxation sequence Ω worked for all cases up to $\max\{h_o(x)\}=1.31$ ($\min\{h_o(x)\}=.69$), which is over 60% of the reference static surface ($p_s = \pm 2$).

2. When performing computations on finer and finer grids we might be forced to drop some of the biggest w's to assure convergence of the iterative scheme.

3. In all cases considered, application of the optimal Wachspress sequence Ω (optimal when referred to $p_s = 0$) was much more effective than search for an appropriate single global relaxation factor.

Let us take for example the case of $p_s=1.85$, which is close to the reference surface, and has large gradient h_x compared to the flat interface $p_s = 0$. Assume that we perform computations on grid of spacial increment r=0.1. If we determine a sequence Ω as for the case $p_s = 0$, the iterative procedure with this set Ω is divergent. This is due to the fact that we are far away from the requirements of the Model Problem.* Nevertheless, if the computational cycle is made shorter by cutting off the largest w, the algorithm is convergent, and it takes 89 iterations to get to results compatible with machine accuracy. On the other hand, if we were to rely on a single global relaxation factor (or time increment in a pseudo-unsteady setup), the number of iterations would be much greater. Furthermore, a small departure from the optimal w would yield a rapid increase in the number of iterations required to achieve the same accuracy of results. This makes computations potentially much more expensive. Below there is a graphical presentation of how the number of iterations depends on a choice of a single global relaxation factor (Fig. 4.1). If we applied optimal sequence of relaxation parameters Ω to the case presented on Fig. 4.1, we would obtain convergence after 26 iterations, which results in the saving of 46% of computer time.

To gain a more detailed insight into the effectiveness of the iterative procedure governed by automatically generated sequence Ω, let us consider the Table 4.3. In this table, for each tested grid size, the following information is given: number of iterations, and a set of a priori (p_s=0) computed values of Ω sequence. Computations were carried on up to machine accuracy. Three cases are presented:

A) stagnation pressure p_s= 0 (flat interface)

B) p_s =1.5 (convex static interface of about 40% of elevation when referred to full semicircle).

C) $p_s=-1.5$ concave interface, of the same percentage as above.

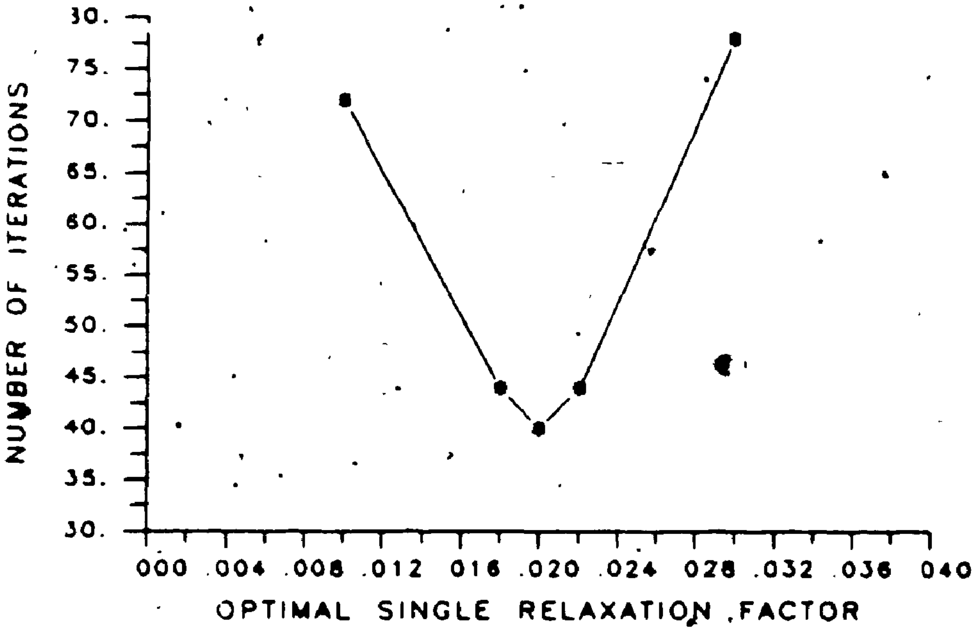


Fig. 4.1 Number of ADI iterations vs relaxation parameter; grid spacial increment r=0.1, $p_s=1.5$.

TABLE 4.3 Number of ADI iterations with optimal relaxation.

r=Δx=Δz	Case A # of its.	Case B # of its.	Case C # of its.	Ω
1/10	22	26	27	.29-2; .80-2; .26-1; .1-1
1/20	22	43	48	.76-3; .22-2; .79-2; .28-1; .82-1
1/30	26	69	62	.33-3; .88-3; .29-2; .96-2; .32-1; .84-1
1/40	26	89 $	89 $	.19-3; .57-3; .21-2; .76-2; .28-1; .82-1
1/60	30	154 $	164 $	.83-4; .23-3; .77-3; .26-2; .91-2; .50-1; .83-1

$: cases in which the largest w was dropped to avoid divergence.

The time required for a single ADI iteration (sweep in x and a sweep in z directions) is as follows (Cyber 825):

TABLE 4.3 CPU time required for a single ADI step with increasing grid dimension.

$\Delta x = \Delta z = r$	1./10.	1./20.	1./30.	1./40.	1./60.
time(sec)	.08	.35	.75	1.4	2.6

It might be interesting to have a look at the history of iterations, as reflected in values of residuals. By residual we understand the value of discretized field operator $\Delta^2\psi = (\Delta_{xx}\psi + \Delta_{zz}\psi)$. It is obvious, that $\Delta^2\psi = 0$ only if ψ is an exact solution of the discretized field equation. The following plots present two cases, i.e. $p_s = 1.5$ and $p_s = -1.5$, each showing the history of residuals for iterations pursued with the best single relaxation factor (see legend) and separately, when sequence Ω of optimal relaxation parameters are employed. Results presented were obtained by computations on grid with r=0.1.

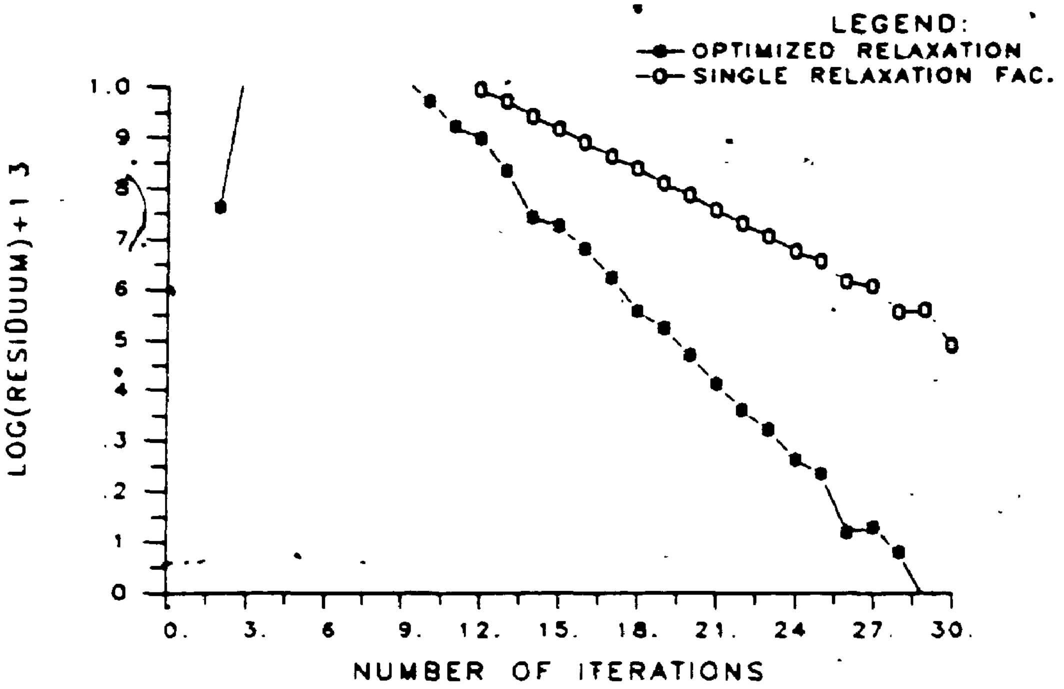


Fig. 4.2 History of residuals of the ADI iterative scheme for $p_s = -1.5$.

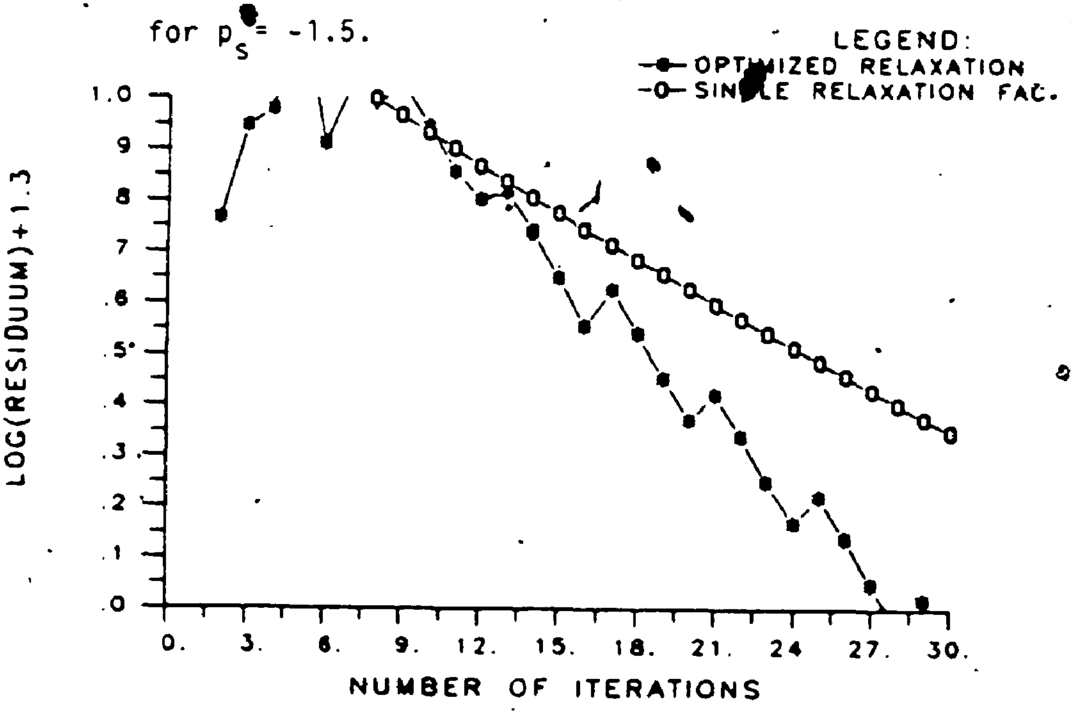


Fig. 4.3 History of residuals of the ADI iterative scheme for $p_s = 1.5$.

4.2.2 - Acuracy of discretization.

As was discussed in Sec. 3.3, discretization of the differential equation always produces an error. This discretization error depends on the spatial increment of the grid r and on the variation of higher derivatives in the neighbourhood of pivotal points. With grid refining this variation is decreased, and the discretization error is dominated by a power of r. With r decreasing, the error decreses to zero. The rate of convergence of error to zero will be referred to, in accordance with convention of Chapter 3, as the order of grid-convergence of the algorithm.

We are going to present the results in two steps. First, the attention will be focussed on a particular case, i.e. when $p_s = 0$. This case has an analytical solution. We will examine the order of grid-convergence by comparing the numerical solution to the exact one, and also by comparing numerical solutions obtained on three grids with decreasing spatial increment r. The purpose of this test is to find out whether both methods yield similar results. Similarity of results is expected to take place in a limit, i.e. when r ⟶ 0. In practice, we always operate on small, but finite values of r. Hence, it depends on the particular difference model how small r should be to to assure that the 3-grid order estimation is valid. Having established the order of convergence, we will address the issue of extrapolation and its effectiveness in providing a highly accurate approximation of the solution.

In the second part of this section, the estimate of the order of grid-convergence of discretization error will be presented for cases with ~~an~~ irregular surface, when $p_s \neq 0$.

The mathematical model of the flow takes on the simplest form when $p_s = 0$. In this case the static interface is flat ($h_o=1$). Since the flow domain is of a regular, square shape, the transformation is trivial and the field equation is in the canonical Laplacian form. With boundary conditions specified as in Sec. 2.1, the field equation has an analytical solution: $\tilde{\Psi} = \sin(\pi x)\sinh(\pi(1-y))/(\pi\sinh(\pi))$. ~~The~~ Table 4.4 (below) shows the distances of computed values from the exact solution and the appropriate estimate of the order of grid-convergence. For the sake of completeness we show data for the function as well as for its second derivative.

TABLE 4.4: Order of grid-convergence of the stream function equation when the analytical solution is known ($p_s = 0$).

Computational grids	$[\psi_o - \tilde{\Psi}]$(G60)	$[\psi_{oxx} - \tilde{\Psi}_{xx}]$(G10)	$[\psi_{oyy} - \tilde{\Psi}_{ozz}]$(G10)
G10	5.4E-5	4.1E-4	4.1E-4
G20	1.4E-5	1.03E-4	1.03E-4
G30	6.0E-6	4.56E-5	4.56E-5
G40	3.4E-6	2.57E-5	2.57E-5
G60	1.5E-6	1.14E-5	1.14E-5

Order of grid-convergence (see formula (3.4.5))

variables / grids	ψ_o	ψ_{oxx}	ψ_{ozz}
G10 : G20	1.99	1.99	1.99
G20 : G30	2.00	2.00	2.00
G30 : G40	2.00	1.99	1.99
G40 : G60	2.02	2.00	2.00

In the above table and in the following discussion, ψ is the numerical estimate of the stream function. Computations were carried out on five grids: G10,...,G60, ψ_{oxx} and ψ_{ozz} are finite difference estimators of respective derivatives of ψ_o . As is seen, ψ_{oxx} and ψ_{ozz} are equally distant from its analytical counterparts. It is an interesting effect, since boundary conditions in x and z direction are quite different. To make sure that results are independent of the method of solution, the computations were repeated using point implicit Gauss-Seidel method, which yielded virtually the same results as the ADI method.

The distances of second derivatives from the appropriate finite difference approximations are, as one may see in Table 4.4, of one order larger than they are for the function. This is due to the fact, that the total error of derivative estimator e_t is the superposition of discretization error e_d and errors of component function estimators e_i:

$$[(\psi_{i-1}+ e_{i-1}) - 2(\psi_i + e_i) + (\psi_{i+1}+ e_{i+1})] + e_d = \tilde{\Psi}_{xx}$$

$$\psi_{xx}+ (e_{i-1}- 2e_i + e_{i+1})/r^2 + e_d = \tilde{\Psi}_{xx}$$

$$e_t = \psi_{xx} - \tilde{\Psi}_{xx} = e_d + (e_{i-1} - 2 e_i + e_{i+1})/r^2$$

Regardless of this difference, the order of grid-convergence is about 2 for function as well as for derivatives. Since the order that has been estimated through computations on grids G10 and G20 is 1.99, it may be reasonable to assume, that results based on these two grids can be used for extrapolation. In fact, extrapolation yields the following;

$$
\begin{aligned}
&[E_{10,20}(\psi) - \tilde{\Psi}](G10) = 2.32\ E\text{-}8\\
&[E_{10,20}(\Delta_{xx}\psi) - \tilde{\Psi}_{xx}](G10) = 2.13E\text{-}6\\
&[E_{10,20}(\Delta_{zz}\psi) - \tilde{\Psi}_{zz}](G10) = 2.13E\text{-}6
\end{aligned}
\qquad (4.2.2)
$$

which shows, that extrapolated field E(Ψ) is better than the field computed on grid of r=1./60 (compare Table 4.4).

Data presented in Table 4.4 were obtained in an idealized situation , that is when we have had an analytical solution available for comparison. In general, we are not in this comfortable position and the grid-convergence order is computed exclusively from numerical solutions. Let us look into Table 4.5. There we may find that the order estimated on grids G10-G20-G40 is 1.95 for Ψ and 2 for Ψ_{xx} and Ψ_{zz} (we still consider the case of p_s=0). Now, using these estimates, we may again look on the quality of extrapolation:

$$
\begin{aligned}
&[E_{10,20}(\psi) - \tilde{\Psi}](G10) = 5.24\ E\text{-}8\\
&[E_{10,20}(\Delta_{xx}\psi) - \tilde{\Psi}_{xx}](G10) = 1.7E\text{-}6\\
&[E_{10,20}(\Delta_{zz}\psi) - \tilde{\Psi}_{zz}](G10) = 2.7E\text{-}6
\end{aligned}
\qquad (4.2.3)
$$

which is worse than the one shown in (4.2.2), but still, extrapolated values are closer to the exact ones than those computed on the very fine grid of r=1/60.

When dealing with capillary flows, we might be interested in the high accuracy of some of the field derivatives at the free surface. Since we discuss extrapolation as a possible method for obtaining more accurate results, one might ask how we should carry out an effective extrapolation for derivatives. Obviously there are two ways of doing this: either by using the scheme $E(\Delta_{xx}\psi)$ or $\Delta_{xx}E(\psi)$. The first one is extrapolation of the derivative estimate, while the second one is expressing derivatives as the appropriate finite difference of extrapolated values $E(\psi)$. Here we confine our attention to the second derivatives. If extrapolation is performed using results obtained on grids G10 and G20, we obtain what follows:

$$\| \Delta_{xx}E(\psi) - \tilde{\Psi}_{xx} \|(G10) = 7.5E-4 \qquad (4.2.4)$$
$$\| \Delta_{zz}E(\psi) - \tilde{\Psi}_{zz} \|(G10) = 7.5E-4$$

$$\| \Delta_{xx}E(\psi) - \Delta_{xx}\tilde{\Psi} \|(G10) = 1.17E-6 \qquad (4.2.5)$$
$$\| \Delta_{zz}E(\psi) - \Delta_{zz}\tilde{\Psi} \|(G10) = 3.6E-6$$

We should recall that $\| E_{10,20}(\psi) - \tilde{\Psi} \| = 5.24$ E-7. This results in high accuracy in (4.2.5). But what we want is accurate estimate of the derivative not of the difference formula $\Delta_{xx}\tilde{\Psi}$. Results shown in (4.2.4) are worse than those in (4.2.3). This is a consequence of truncation error, which for grid G10 may be relatively high. It follows, therefore, that extrapolation of derivatives may yield better results than estimating derivatives using extrapolated stream function values. This observation seems to be of the general character, applicable to any finite-difference algorithm, not only the Asymptotic one. Since computations of the stream function are decoupled from determination of the interface, we may proceed as follows:

a) Make sure that the order of grid-convergence is as expected from the theory.

b) Compute field on two grids, say G10 and G20 and extrapolate $\Delta_z\psi$.

c) Substitute extrapolated $\Delta_z\psi$ into equation (4.1.9) describing correction of the interface.

This procedure may provide a highly accurate estimation of the shape of the interface without the necessity of performing costly computations on fine grids.

IRREGULAR GEOMETRY

The first part of this Section was restricted to the case of the flat static interface (p_s=0). Now attention will be focussed on the general case, when stagnation pressure p_s is not equal to zero. The first observation which is to be made, is that the flow field equation is losing its canonical Laplacian form and has a more complicated structure (4.2.6). This may cause deterioration in the behaviour of the truncation error because of the variable coefficients of the differential equations. In the equation for the stream function, the coefficients are expressions involving function h(x) and its derivatives (see Eq.(4.1.9). Before we comment on this equation, let us present results, which are given in Tables 4.5-4.8. Each table shows the order of grid-convergence of the discretization error. This order is estimated on the basis of results computed on three grids $Gk \subset Gl \subset Gm$. For the sake of completeness we show data not only for the stream function and interface, but also for all derivatives appearing explicitely in the governing equations. Each table has a descriptive head identifying a particular case by specification of p_s, C and ϵ. For all cases ϵ = .05, which, as will be shown in Section 4.3.4 , satisfies assumptions of the Small Deformation Theory. 'Interior' here means that we take into account only values in the interior of the solution domain. When using 2-nd order discretization for Dirichlet-type boundary conditions we compute only the values in the interior of the computational domain.

DESCRIPTION OF TABLES AND COMMENTS

1. The tables present results for the following values of p_s : -1.7,-1.6,-1.5, 0., 0.4, 0.8, 1.3, 1.5. On the graph below we show respective static interfaces from the lowest (p_s=-1.7) to highest (p_s =1.5).

2. For all cases considered, the grid-convergence order for the stream function estimate ψ_o is not worse than 1.95, and in all but one case (p_s=1.5) ψ_{oz} is of order 1.95 or better. We pay attention to ψ_{oz} because this derivative enters the equation determining correction of the shape of the interface.

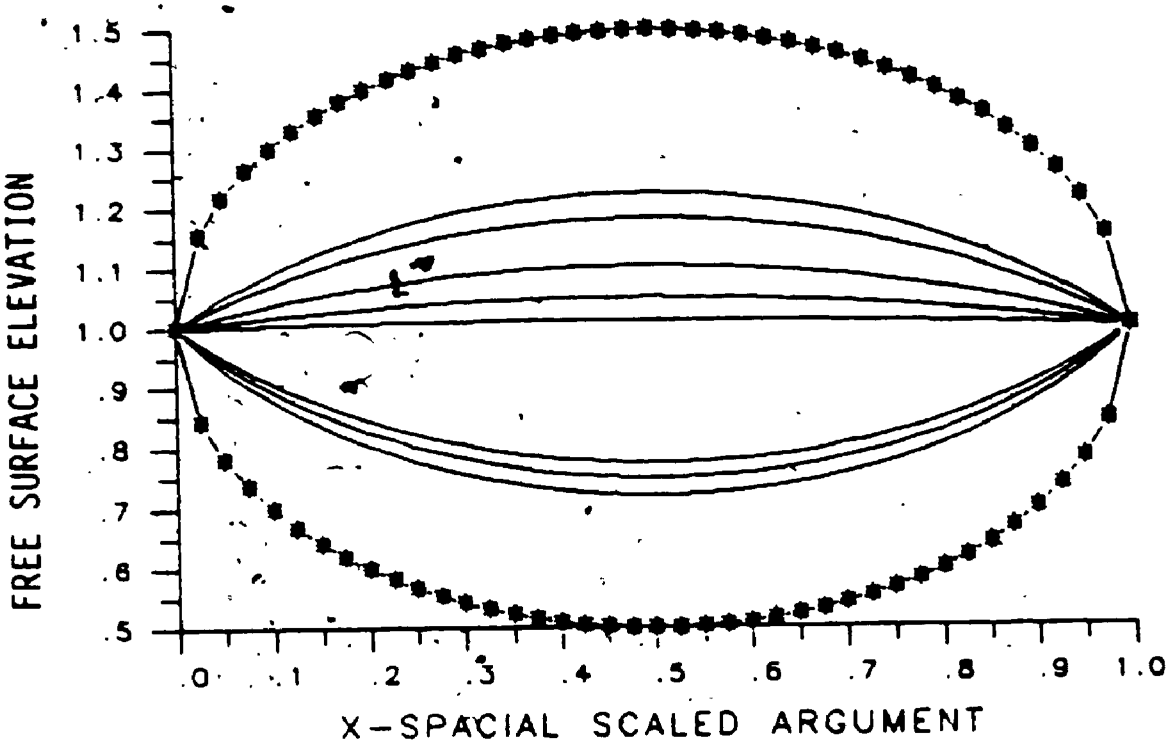


Fig. 4.4 - Static Surface for selected values of p_s = -1.7 -1.6, -1.5, 0., 0.4, 0.8, 1.3, 1.5. Lower and upper star-curves are for p_s= -2, p_s=2.

Therefore a proper order of ψ_{oz} might be decisive in obtaining desired accuracy for the final numerical estimate of the shape of the free surface h(x).

3. Best results were obtained for p_s= 0. The further we go from this static interface, the worse are the results for the fixed triples

of grids Gk, Gl, Gm. This is reflected in particular in worsening of the grid-convergence order for higher derivatives of ψ_0 and h. But, what seems interesting, the deterioration of order is not the same for convex and concave domains. The results for $p_s > 0$ (convex free surface) seem to be worse than for $p_s < 0$ (concave free surface). In fact, for computations on concave domain, denser grids are required to maintain the desired grid-convergence order if the stagnation pressure is $p_s = -1.7$ or lower. On the other hand, when the static surface is convex, this requirement comes into effect when $p_s \geq 1.3$. As one can see, the curvature of the static free surface with $p_s = 1.3$ is much smaller than the curvature of the concave one, when $p_s = -1.7$. Another observation that may be stated is that when considering ψ only, computations with $p_s = -1.5$ or $p_s = -1.6$ provide results which look better (order=2 for all triples of grids), than in the case of $p_s = 0$. This suggests that numerical computations yield results less biased by discretization errors for concave static interface than for the convex ones.. The reason for this is probably transformation of the variables. Regardless of the physical case considered, we always apply uniformly distributed grid spanned on the square domain which is the image of the physical domain through transformation (Fig.3.1).Now, if the inverse transformation is applied to the discretized domain, the respective physical grid is much denser in the vicinity of the free surface for concave interfaces than for the convex ones. This gives some explanation why computations on a convex domain need denser grids to yield the order of grid-convergence comparable to that estimated on concave domains of similar curvature of the free surface.

Another explanation for the better performance of the algorithm on concave domains can be provided by analysis of coefficients of z-derivatives of function Ψ (see (3.2.a)). These coefficients for ψ , Ψ_z , and Ψ_{zz} are as follows:

$$\begin{aligned} c_z(x) &= (2\,h_x^2 - h_{xx}h)(x)\frac{z}{h^2} \\ c_{zz}(x) &= (z^2h_x^2 + 1)(x)\,\frac{1}{h^2} \\ c_{xz}(x) &= -\,\frac{2zh_x}{h} \end{aligned} \qquad (4.2.7)$$

For simplification let us omit mixed derivatives as being computed outside the actual ADI scheme. Consider two geometrically symmetric (symmetry with respect to $p_s=0$, or equivalently, h=1) case $p_s= a$ and $p_s =-a$, $0 < a < 2$. Since this symmetry is reflected in function h(x) and its derivatives, we may state the following:

$$\begin{aligned} c_{zz}(x,a) &= c_{zz}(x,-a) \\ c_z(x,a) &= \frac{2zh_x}{h^2} + \frac{z}{h}\,|h_{xx}| \\ c_z(x,-a) &= \frac{2zh_x}{h^2} - \frac{z}{h}\,|h_{xx}| \end{aligned} \qquad (4.2.8)$$

From the above it follows that the inequality $c_z(x,a) > c_z(x,-a)$ holds at the left and right boundaries of the unit interval [0,1], and hence in their vicinity. This inequality may grow stronger with the increase of the parameter 'a'. Further on, in the process of discretization of (3.2.a), $c_z(x,a)$ is multiplying the Lagrange residual of the discrete approximation of Ψ_z. Therefore, since $c_z(x,a) > c_z(x,-a)$, the variation of the residual is enhanced if $p_s = a$ (convex static free surface), compared to the case $p_s=-a$. This, in turn, may result in a lower estimate of the grid-convergence order for $p_s = a$, compared to $p_s=-a$, if in both cases (convex and concave) the same grids are utilized.

4) Let us examine tables with $p_s=-1.7$, 0.8, 1.3, 1.5. For these cases certain derivatives of Ψ and h(x) are of lower grid-convergence order. It means that in order to have this order increased to the expected value of 2, we would have to perform computations on finer grids than those already employed. We cannot use extrapolation with the grids presently employed.

TABLE 4.5.a Estimates of grid-convergence order for Problem (4.1.6) - (4.1.9) by means of comparing results computed on grids Gk, Gl, Gm;
Case $p_s=-1.5$, $\zeta=2.63$, $\epsilon=0.05$

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.200E+01	.200E+01	.196E+01	.175E+01	.196E+01	.190E+01	.200E+01	.205E+01	.206E+01
$f^{20}-f^{40}-f^{60}$	.200E+01	.200E+01	.196E+01	.190E+01	.196E+01	.196E+01	.200E+01	.200E+01	.200E+01
$f^{30}-f^{40}-f^{60}$	.200E+01	.200E+01	.196E+01	.195E+01	.196E+01	.196E+01	.200E+01	.200E+01	.200E+01

TABLE 4.5.b Estimates of grid-convergence order for Problem (4.1.6) - (4.1.9) by means of comparing results computed on grids Gk, Gl, Gm;
Case $p_s = -1.7$, $C=1.85$, $\epsilon=0.05$

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.195E+01	.200E+01	.195E+01	.175E+01	.190E+01	.170E+01	.205E+01	.215E+01	.220E+01
$f^{20}-f^{40}-f^{60}$	.195E+01	.200E+01	.195E+01	.185E+01	.190E+01	.175E+01	.200E+01	.200E+01	.205E+01
$f^{30}-f^{40}-f^{60}$	.195E+01	.200E+01	.195E+01	.185E+01	.195E+01	.180E+01	.200E+01	.200E+01	.200E+01

TABLE 4.6.a Estimates of grid-convergence order for Problem (4.1.6) - (4.1.9) by means of comparing results computed on grids Gk, Gl, Gm;
Case $p_s=0$, $C=13.8$, $\epsilon=0.05$

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.195E+01	.200E+01	.195E+01	.200E+01	.200E+01	.200E+01	.210E+01	.205E+01	.205E+01
$f^{20}-f^{40}-f^{60}$	.195E+01	.200E+01	.195E+01	.200E+01	.200E+01	.200E+01	.200E+01	.200E+01	.200E+01
$f^{30}-f^{40}-f^{60}$	.200E+01	.200E+01	.200E+01	.200E+01	.200E+01	.200E+01	.200E+01	.200E+01	.200E+01

TABLE 4.6.b Estimates of grid-convergence order for Problem (4.1.6) - (4.1.9) by means of comparing results computed on grids Gk, Gl, Gm;
Case p_s=0.1, C=15.1, ϵ=0.05

	ψ	ψ_x	ψ_z	ψ_{zx}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.196E+01	.200E+01	.195E+01	.200E+01	.200E+01	.200E+01	.215E+01	.205E+01	.210E+01
$f^{20}-f^{40}-f^{60}$	.196E+01	.200E+01	.195E+01	.200E+01	.200E+01	.200E+01	.205E+01	.200E+01	.205E+01
$f^{30}-f^{40}-f^{60}$	.196E+01	.200E+01	.196E+01	.200E+01	.200E+01	.200E+01	.205E+01	.200E+01	.200E+01

TABLE 4.7.a Estimates of grid-convergence order for Problem (4.1.6) - (4.1.9) by means of comparing results computed on grids Gk, Gl, Gm;
Case p_s=0.4, C=20.8, ε=0.05

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.196E+01	.200E+01	.196E+01	.205E+01	.200E+01	.196E+01	.205E+01	.205E+01	.205E+01
$f^{20}-f^{40}-f^{60}$	.196E+01	.200E+01	.196E+01	.200E+01	.196E+01	.196E+01	.200E+01	.200E+01	.200E+01
$f^{30}-f^{40}-f^{60}$	.196E+01	.200E+01	.196E+01	.200E+01	.196E+01	.196E+01	.200E+01	.200E+01	.200E+01

TABLE 4.7.b Estimates of grid-convergence order for Problem (4..1.6) - (4.1.9) by means of comparing results computed on grids Gk, Gl, Gm;
Case p_s=0.8, C=31.2, Є=0.05

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.195E+01	.200E+01	.195E+01	.195E+01	.195E+01	.195E+01	.390E+01	.225E+01	.210E+01
$f^{20}-f^{40}-f^{60}$	.195E+01	.195E+01	.195E+01	.185E+01	.195E+01	.190E+01	.225E+01	.215E+01	.240E+01
$f^{30}-f^{40}-f^{60}$	.195E+01	.195E+01	.195E+01	.185E+01	.195E+01	.190E+01	.185E+01	.215E+01	.245E+01

TABLE 4.8.a Estimates of grid-convergence order for Problem (4.1.6) - (4.1.9) by means of comparing results computed on grids Gk, Gl, Gm;
Case $p_5=1.3$, $C=50.3$, $\varepsilon=0.05$

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.195E+01	.195E+01	.195E+01	.185E+01	.190E+01	.170E+01	.145E+01	.225E+01	.250E+01
$f^{20}-f^{40}-f^{60}$	.195E+01	.195E+01	.195E+01	.165E+01	.180E+01	.150E+01	.170E+01	.210E+01	.310E+01
$f^{30}-f^{40}-f^{60}$	.195E+01	.195E+01	.195E+01	.165E+01	.180E+01	.150E+01	.170E+01	.205E+01	.310E+01

TABLE 4.8.b Estimates of grid-convergence order for Problem (4.1.6) - (4.1.9) by means of comparing results computed on grids Gk, Gl, Gm;
Case p_s=1.5, C=80.6, ϵ=0.05

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.195E+01	.195E+01	.195E+01	.185E+01	.180E+01	.165E+01	.155E+01	.225E+01	.270E+01
$f^{20}-f^{40}-f^{60}$	.195E+01	.190E+01	.195E+01	.175E+01	.150E+01	.135E+01	.170E+01	.205E+01	.285E+01
$f^{30}-f^{40}-f^{60}$	.195E+01	.190E+01	.195E+01	.175E+01	.145E+01	.130E+01	.175E+01	.200E+01	.260E+01

4.3 Qualitative Effectiveness of the Algorithm based on Small Deformation Theory

In this subsection we shall consider the problem of determining the limits of applicability of the Small Deformation Theory. As is seen on the Flowchart (Sec. 4.1), the final shape of interface is

$$h(x) = h_0(x) + (\Delta h)(x) \approx h_0(x) + \epsilon h_1(x) \qquad (4.3.1)$$

Computation of the static free surface deformation is decoupled from the field, i.e. $h_1(x)$ is being determined after the first approximation of flow field is found. Hence the Asymtotic Algorithm depends actually only on a single parameter, i.e., on the assumed stagnation pressure p_s. The algorithm does not use the capillary number C; this information is used only during the last step, when we formulate estimate (4.3.1), where C is implicitly present in ϵ (see Sec. 4.1). Below we will analyse approximation (4.3.1) and try to determine range of ϵ values for which the Small Deformation Theory is valid. Since smallness of ϵ is fundamental, this issue will be discussed first.

Let us recall the dynamic boundary condition:

$$\frac{(h_0 + h)_{xx}}{(1+(h_0+ h)_x^2)^{3/2}} + p_s - (CV_s^2)|\underline{V}|^2 = 0 \qquad (4.3.2)$$

By assuming $\Delta h = .\epsilon h_1 + \epsilon^2 h_2 + \ldots$, $\psi = \psi_0 + \epsilon\, \psi_1 + \ldots$ and applying the appropriate series expansion we obtain the following:

$$\epsilon\left(h_{1xx}\frac{1}{(1+h_{1x}^{2})^{3/2}} - h_{1x}\frac{3h_{oxx}h_{ox}}{(1+h_{ox}^{2})^{5/2}}\right) + \epsilon^{2}\ldots = \epsilon|\underline{V}|^{2} + \epsilon^{2}\ldots \quad (4.3.3)$$

Some terms in this equation were dropped as contributing to the static surface equation, and $C(0.5\ V_s^2)$ was set equal ϵ.

We know very little about the magnitude of terms associated with ϵ^2, ϵ^3, etc. But if we assume that $C << 1$, then due to the assumption that deformation of the static surface is <u>small</u>, we may assume that magnitude of terms multiplying ϵ^k is negligible when compared to ϵ. In effect all terms on both sides of the equation (4.3.3) containing second and higher powers of ϵ can be dropped. Therefore we may feel justified when assuming the final solution to be in the form $h = h_o + \epsilon h_1$ for small ϵ. The above reasoning may be stated formally in the following way:

ϵ^* is small if and only if $h - (h_o + \epsilon h_1)$ is $O(\epsilon^2)$, i.e.

$$\frac{h - (h_o + \epsilon_k h_1)}{h - (h_o + \epsilon_l h_1)} \propto \frac{\epsilon_k^2}{\epsilon_l^2} \quad (4.3.4)$$

for ϵ_k, $\epsilon_l < \epsilon^*$

To give a precise account of how the concept of small ϵ works in our problem, let us start from the table of distances $\|h_d - h\|$, where h is a total free surface shape obtained from the Small Deformation Theory, and h_d is the actual interface computed by a Direct Algorithm (see Chapter 5). Here the Direct Algorithm results are used as reference for values obtained by application of the Small Deformation

Theory.

TABLE 4.9 Distance between the free surface solutions computed by the Asymptotic and Direct Algorithms.
($: Direct Algorithm does not converge)

ϵ / p_s	.5	.4	.3	.2	.1	.05	.025
0.	9.6E-3	5.8E-4	3.1E-4	1.3E-4	3.1E-5	7.6E-6	1.9E-6
1.5	4.2E-3	2.3E-3	1.2E-3	4.8E-4	1.1E-4	2.6E-5	6.4E-6
1.85	3.9E-3	1.9E-3	8.9E-4	3.3E-4	6.8E-5	1.6E-5	3.8E-6
-1.5	3.1E-3	1.7E-3	8.6E-4	3.5E-4	7.9E-5	1.9E-5	4.7E-6
-1.85	$	$	6.6E-3	1.9E-3	3.7E-4	8.2E-5	1.9E-5

Now, let us try to find the biggest ϵ^* for which relation (4.3.4) holds. For the sake of clarity instead of examining (4.3.4) we will look at equivalent formula:

$$\frac{\ln(a_1[h-(h_o+\epsilon_n h_1]) - \ln(a_2[h-(h_o+\epsilon_{n+1} h_1])}{\ln(a_1\epsilon_n) - \ln(a_2\epsilon_{n+1})} \propto 1$$

where subscript 'n' is the index of decreasing sequence of ϵ's, and constants a_1 and a_2 are set arbitrarily for convenient presentation of the numerical results (a_1=e/1.876E-6, a_2=e/6.25E-4). We expect the above ratio to approach a constant proportionality factor as ϵ is decreased. Below we present the aggregate table showing how the proportionality factor converges to a constant value of 1 for different ϵ and p_s.

TABLE 4.10

Small Deformation Theory-

Examination of the criterion (4.3.4) for smallness of parameter ϵ, for selected values of stagnation pressure p_s.

p_s ϵ	0.	1.5	-1.5	1.85	-1.85
0.5	0.	0.	0.	0.	0.
0.4	1.104	1.349	1.361	1.53	5.159
0.3	1.127	1.175	1.184	1.365	43.359
0.2	1.073	1.112	1.126	1.25	1.542
0.1	1.031	1.062	1.063	1.131	1.18
0.05	1.017	1.022	1.028	1.048	1.083
0.025	1.007	1.025	1.008	1.032	1.029

Let us assume that if the proportionality ratio is smaller that 1.1, we accept respective ϵ as sufficiently small. From this we may see that for different cases of stagnation pressure we have a different maximum acceptable ϵ^* . Generally, the bigger the static free surface curvature is, the smaller is acceptable C*, as shown on the following diagram:

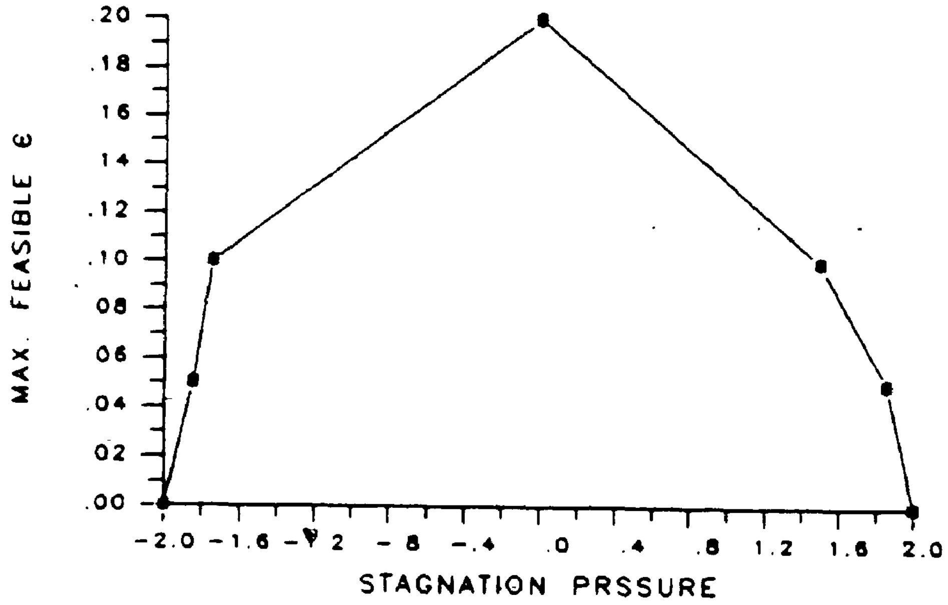


Fig. 4.5 Maximal feasible value ϵ^* of the asymptotic parameter, as a function of a stagnation pressure.

From the Table 4.10 it may be also seen, that the in the case of the concave static surface $(p_s < 0]$ the restriction on ϵ is stronger than in the case of the convex one with the same curvature.

Let us now turn our attention to the problem mentioned earlier, that is the problem of limitations which must be imposed on the model parameters to make the solution consistent with requirements of the Small Deformation Theory.

Let us recall that the structure of the Asymptotic Algorithm requires selecting the stagnation pressure first. Having done this we find static interface and the first approximation to the flow field. From this we may estimate the maximal speed of the fluid along the interface (V_s - which is the surface velocity scale). We are unable to provide an analytical expression for the maximal value of the ϵ parameter. But experimentation documented in Table 4.9 may provide some estimates of acceptable ϵ for a given case. Data presented so far provide basis for the estimate of the upper bound of acceptable ϵ for various values of p_s. Having estimated maximal value of the parameter ϵ*, and knowing surface velocity scale V , we may set upper bound for feasible values of the capillary parameter C. The table below shows experimental estimates of C*, and in consequence, the restriction on the value of C for different stagnation pressures p_s. The above procedure provides bounds on parameters of the physical flow so that an application of the Small Deformation Theory is justified.

TABLE 4.11

Feasible values of capillary number C for Small Deformation Method

p_s	-1.85	-1.5	0.	1.5	1.85
ϵ*	.05	.1	.2	.1	.05
C <	1.25	2.63	27.5	80.6	113.6

Despite the geometrical symmetry (with reference to static surface h=1, or equivalently, $p_s=0$) of the presented cases, bounds for capillary parameters do not display a similar behaviour. Therefore application of the Small Deformation Theory implies stronger bounds on feasible values of the capillary parameter when static free surface is more concave. This constraint is relaxed with the increase of p_s.

4.4 Algebraic stability

When dealing with the equation of flow field, we have confined the discussion to a numerical solution when the static interface is fixed. This follows from the methodology of the Small Deformation Theory. We do not use, however, the exact algebraic solver for the resulting system of algebraic equations, but an approximate iterative scheme of ADI. Therefore it is important to examine the algebraic stability of equations to rule out a possibility that round-off errors could affect the final numerical results. The convenience of pursuing the analysis for the fixed free surface is, that the coefficients of the stream function equation do not depend on the index of the ADI iteration.

Although the static interface in our case is of a circular shape only, we will pursue stability analysis for a sinusoidal shape, which may be generalized to any free surface of a polyharmonic representation.

Let us consider free surface of a harmonic character:

$$h(x) = 1. + A\sin(2\pi x) \tag{4.4.1}$$

The subsequent analysis will impose restrictions on the grid spatial increment r and relaxation factor w, as sufficient condition for the Algebraic Stability of the finite difference model of the stream function equation.

For the 3-diagonal system of equations of the form:

$$-a_j u_{j-1} + b_j u_j - c_j u_{j+1} = d_j \tag{4.4.2}$$

the sufficient conditions to avoid ill-conditioning of Thomas Algorithm are (Smith [27]):

$$\begin{aligned} &\text{(i)} \quad a_j > 0, \quad b_j > 0, \quad c_j > 0 \\ &\text{(ii)} \quad b_j > a_j + c_j \\ &\text{(iii)} \quad b_j > a_{j+1} + c_{j-1} \end{aligned} \tag{4.4.3}$$

If one takes into consideration the discretized form of our field equation, it is known that it generates within the ADI procedure two systems of linear equations. The system constituting the x-ADI sweep has constant coefficients, which trivially satisfy all the above requirements. It is in the z-ADI sweep, where coefficients resulting from change of variables come into play. These coefficients are:

$$a_j = -\frac{z_j}{2rh^2}(2h_x^2 - h_{xx}h) + \frac{1}{r^2}\left(\frac{z_j^2h_x^2 + 1}{h^2}\right)$$

$$b_j = \frac{2}{r^2}\left(\frac{z_j h_x^2+1}{h^2}\right) + w \tag{4.4.4}$$

$$c_j = \frac{z_j}{2rh^2}(2h_x^2 - h_{xx}h) + \frac{1}{r^2}\left(\frac{z_j h_x^2 + 1}{h^2}\right) .$$

Condition (ii) is trivially satisfied, if only the relaxation factor w is positive.

Let us consider requirement (iii). The coefficients in question are as follows:

$$a_j = -\frac{z_j+r}{2rh^2}(2h_x^2 - h_{xx}h) + \frac{1}{r^2}\left(\frac{(z_j+r)^2 h_x^2+1}{h^2}\right)$$

$$B_j = \frac{2}{r^2}\cdot\frac{z_j^2 h_x^2}{h^2} + w \tag{4.4.5}$$

$$c_j = \frac{z_j-r}{2rh^2}(2h_x^2 - h_{xx}h) + \frac{1}{r^2}\left(\frac{(z_j-r)^2 h_x^2+1}{h^2}\right)$$

By adding $a_{j+1} + c_{j-1}$, and recalling that $\Delta z = \Delta x = r$, we obtain:

$$a_{j+1} + c_{j-1} = -\frac{1}{rh^2}(2h_x^2 - h_{xx}h) + \frac{2}{r^2}\left(\frac{z_j h_x^2 + 1}{h^2}\right) + \frac{2h_x^2}{h^2}$$

Therefore the requirement (4.3.3-iii) results in the following:

$$\frac{2h_x^2}{h^2} - \frac{1}{h^2}(2h_x^2 - h_{xx}h) \le w$$

or simpler:

$$\frac{h_{xx}}{h} < w \qquad (4.4.6)$$

which is the condition imposing the lower bound for the relaxation coefficient, or time increment, if we set $\Delta t = 1/w$.

For the function h(x) defined by (4.4.1) this condition requires smaller 1/w with increasing amplitude A. This is shown on the table below:

TABLE 4.12 Constraint on the field relaxation as a function of the amplitude (for harmonic free surface).

Amplit.	.30	.35	.40	.66	.8
1/w <	.059	.048	.038	.013	.006

It is worth noting that whenever $h > 0$ and $h_{xx} \leq 0$ (like all static free surfaces of convex shape), condition (4.4.6) is satisfied and, therefore, requirement (iii) imposes no restriction on the relaxation factor w.

Let us examine now the condition of a_j, b_j, c_j being positive. Certainly, b_j is positive, but examination of a_j and c_j is more involved. As we will see, this condition will lead to a restriction on the spatial increment r.

Let us consider the expression defining a_j (see (4.4.4)):

$$\frac{1}{r^2}\,\frac{z^2h_x^2 + 1}{h^2} \quad - \frac{z\,(\,2\,h_x^2 - h_{xx}h)}{2\,r\,h^2}$$

If $2h_x^2 - h_{xx}h > 0$ then $a_j > 0$; hence this is interesting only when we want to assure that $c_j > 0$. If $2h_x^2 - h_{xx}\,h > 0$, then $c_j > 0$ (see (4.4.4)), but the sign of a_j must be examined.

Examination of $a_j > 0$

Index j is the column index. We want to be sure that $a_j > 0$ for every column, or in the limit, for any argument $x \in [0,1]$. We should also remember that $z_j \in [0,1]$. Consider the interval $J \subset [0,1]$ such that

$$\forall\ (x \in J)(\ 2h_x^2 - h_{xx}h > 0\ \}$$

In general, if $J \neq \emptyset$, it is obvious, that $a_j > 0$, and only condition $c_j > 0$ is left for consideration. But for the function as defined by (4.1.1), J is non-empty which can be easily shown.

As is seen from (4.4.4), coefficient a_j is a function of horizontal and vertical variables: $a_j = a_j(x,z)$. Given the definition of a_j we may state first the following:

Proposition 4.1

$a_j > 0$ for sufficiently small x_j, $j=1,\ldots m$,
where m is a number of columns in z-ADI sweep.

Proof: Let us assume, that for some real B>0 the following holds:

$$|-zh_x^2 + \frac{z}{2}h_{xx}h| > B$$

Let us determine r so that

$$\frac{1}{r}\ B > 0$$

If so, we have the following:

$$\frac{1}{r} + (-zh_x^2 + \frac{z}{2}h_{xx}h) \geq \frac{1}{r} - B > 0$$

On the other hand:

$$\frac{z^2h_x^2+1}{r} - zh_x^2 + \frac{z}{2}h_{xx}h \geq \frac{1}{r} + (-zh_x^2 + \frac{z}{2}h_{xx}h) \geq \frac{1}{r} - B > 0$$

which shows, that having assumed boundedness of the expression in the first step of the proof, we may always find sufficiently small r such that $a_j > 0$ for all j and for $z \in [0,1]$.

In the same way it may be demonstrated that $c_j > 0$ for sufficiently small r.

The above Proposition is the way of assuring $a_j > 0$ provided r is sufficiently small:

$$a_j > 0 \quad \text{iff} \quad r \leq 2\frac{z^2h_x^2 + 1}{z(2h_x^2 - h_{xx}h)} = g(x,z) \qquad (4.4.7)$$

Examination of $g(x,z)$ easily leads to the observation, that for fixed $x_o \in J$, g attains minimum at z:

$$z = \begin{cases} \dfrac{1}{|h_x(x_o)|} & \text{if } [h_x(x_o)] \geq 1 \\ 1 & \text{if } [h_x(x_o)] < 1 \end{cases} \qquad (4.4.8)$$

If minimum is at $z=1/h_x$, we get the following upper bound for r:

$$r \leq \frac{2(1/h_x^2)h_x^2 + 1}{(2h_x^2 - h_{xx}h)/|h_x|} \leq \frac{2(z^2h_x^2 + 1)}{z(h_x^2 - h_{xx}h)} \qquad (4.4.9)$$

and after simplification:

$$r \leq \frac{4|h_x|}{2\,h_x^2 - h_{xx}h} \qquad x \in J, \quad h_x \geq 1. \qquad (4.4.10)$$

If $x \in J$ and $h_x \leq 1$, $g(x,z)$ is minimal if $z=1$. We therefore get:

$$g(x,z) = 2\frac{h_x^2 + 1}{2h_x^2 - h_{xx}h} \geq r \qquad (4.4.11)$$

These results may briefly stated as follows:

Proposition 4.2

To assure $a_j > 0$ it is sufficient to have:

$$r \leq \begin{cases} \dfrac{4|h_x|}{2\,h_x^2 - h_{xx}h} & \text{for } |h_x| \geq 1 \\[2ex] \dfrac{2(h_x^2 + 1)}{2\,h_x^2 - h_{xx}h} & \text{for } |h_x| \leq 1 \end{cases} \qquad (4.4.12)$$

$$h = h_o + \epsilon h_1$$

where h_o is the static interface and h_1 is the correction which is the solution to the following differential equation:

$$h_{1xx} \frac{1}{(1 + h_{ox}^2)^{3/2}} + h_{1x} \frac{3 h_{ox} h_{oxx}}{(1 + h_{ox}^2)^{5/2}} = | \underline{V}_{os} |^2$$

$$h_1 = 0 \quad \text{at} \quad x = 0$$

$$h_1 = 0 \quad \text{at} \quad x = 1$$

Let us recall, that $\epsilon = C(0.5 V_s^2)$, where C is the capillary number, V_s is a maximum of the fluid speed along the free boundary. V_s depends on the stagnation pressure and selected flow across the bottom section of the slot.

The first remark concerning the above equation is that if speed along the free surface is zero, then correction h_1 is zero too. This is obvious, since when there is no flow, the interface has a static character. The second observation is that when assuming positive speed along the interface, we always get non - positive correction h_1. This is not to be wondered at, since with a non-zero velocity field, static pressure decreases compared to stagnation pressure (as can be seen from the Bernoulli equation). Further on, decrease of the static pressure causes the interface to subside below its static position, provided surface tension T is kept constant. This effect is shown in Tables 4.13-4.17 below. The tables are intended to demonstrate how

To assure $c_j > 0$ we need to consider only these values of arguments x, for which $2h_x^2 - h_{xx} h \leq 0$, i.e. $x \in [0,1] \setminus J$. The condition $c_j > 0$ implies:

$$g(x,z) = 2 \frac{z_j^2 h_x^2 + 1}{z(h_{xx} h - 2h_x^2)} \geq r \tag{4.4.13}$$

It may be shown, as before, that $g(x,z)$ reaches its minimum with respect to z if

$$z = \begin{cases} \dfrac{1}{|h_x|} & \text{for } |h_x| \geq 1 \\ 1 & \text{for } |h_x| \geq 1 \end{cases} \tag{4.4.14}$$

Minimization of g(x,z) with respect to z and x yields similar results as before:

$$x \leq \begin{cases} \dfrac{4|h_x|}{h_{xx} h - 2\, h_x^2} & \text{for } |h_x| \geq 1 \\ \dfrac{z(h_x^2 + 1)}{h_{xx} h - 2\, h_x^2} & \text{for } |h_x| \geq 1 \end{cases} \tag{4.4.15}$$

Combihing together formula (4.4.12) and (4.4.15) we arrive at the following statement:

Proposition 4.3

To assure $a_j > 0$ and $c_j > 0$, space increment $\Delta x = \Delta z = r$ of the computational finite difference grid should satisfy the following constraint:

$$r \leq \begin{cases} \dfrac{4|h_x|}{|h_{xx}h - 2\,h_x^2|} & \text{for } |h_x| \geq 1 \\[2ex] \dfrac{2|h_x^2 + 1|}{|h_{xx}h - 2\,h_x^2|} & \text{for } |h_x| \geq 1 \end{cases} \qquad (4.4.16)$$

EXAMPLE 4.2

The table below shows the upper bound for r when both a_j and c_j are required to be positive. Here we refer to function $h(x)$ as defined by (4.4.1)

Amplit.	0.3	0.35	0.40	0.66	0.80
r <	0.13	0.107	0.091	0.046	0.035

It is interesting to have these estimates compared to results obtained from examination of the grid-convergence of discretization error. If A=0.3, from the table above it follows that r should be smaller than 0.13. It means that computations on grid 10x10 are not exposed to algebraic instability. In fact, the grid-convergence order of discretization error when computed on grids 10-20-30, has the value 2.

If A=0.36, the restriction is $r \leq 0.98$ which means that grid 10x10 may be insufficient to provide stability. In fact, the order computed on grids 10-20-30 is 1.85, while when finer grids are used: 20-30-40, the order is 1.95. In the former case it is likely that the lower order is caused by excessive round-off errors of computations on grid 10x10.

EXAMPLE 4.3

Let us return to to the static interface of a circular shape, which is the case for the ideal fluid flow. The equation of this type of free surface has the following form:

$$\frac{h_{xx}}{(1 + h_x^2)^{3/2}} = - p_s$$

Let us examine algebraic stability conditions for this case:

(i) $b_j > a_j + c_j$ if only $w > 0$

(ii) $b_j > a_{j+1} + c_{j-1}$ if $h_{xx} < 0$ (or equivelently $p_s > 0$):

If h_{xx} is positive, (concave static interface), relaxation parameter w must fulfill the following inequality:

$$w > \frac{h_{xx}}{h} \qquad \text{for all } x \in [0,1]$$

This constraint is illustrated in the table below:

TABLE 4.17 : Deformation of the static interface,

when stagnation pressure p_s=-1.85 (C=-0.62).

x_i	0.1	0.2	0.3	0.4	0.5
$h(x_i)$	0.842	0.756	0.703	0.674	0.665
$\Delta h = \epsilon h_1(x_i)$	-3.1E-2	-4.0E-1	-4.61E-1	-4.96E-1	-5.08E-1
curvature of Δh	.21E-0	.39E-1	.23E-1	.23E-1	.24E-1

TABLE 4.18 : Maximum Speed along the Free Surface

(magnitude of scale V_s), for different static prssures.

p_s	0.0	1.5	1.85	-1.5	-1.85
V_s	7.25E-3	1.24E-3	8.80E-4	3.80E-2	8.E-2

It can be seen from the above tables and the following figures (Fig. 4.6.a, 4.6.b), that except for the flat surface, the correction Δh changes fastest near the solid boundary of the interface. This is reflected in the curvature of the correction Δh (see Tables 4.13 - 4.17). Another interesting feature is, that deformations Δh are much greater if the static interface is concave (p_s=-1.5, -1.85) compared to the convex cases (p_s=1.5, 1.85). This may be explained by the fact that for the concave static interface the velocity of the fluid at the free surface is larger (and so is z-gradient of the stream function),

$$h = h_o + \epsilon\, h_1$$

where h_o is the static interface and h_1 is the correction which is the solution to the following differential equation:

$$h_{1xx} \frac{1}{(1 + h_{ox}^2)^{5/2}} + h_{1x} \frac{3\, h_{ox} h_{oxx}}{(1 + h_{ox}^2)^{3/2}} = |\, \underline{V}_{os} \,|^2$$

$h_1 = 0$ at $x = 0$

$h_1 = 0$ at $x = 1$

Let us recall, that $\epsilon = C(0.5\, V_s^2)$, where C is the capillary number, V_s is a maximum of the fluid speed along the free boundary. V_s depends on the stagnation pressure and selected flow across the bottom section of the slot.

The first remark concerning the above equation is that if speed along the free surface is zero, then correction h_1 is zero too. This is obvious, since when there is no flow, the interface has a static character. The second observation is that when assuming positive speed along the interface, we always get non - positive correction h_1. This is not to be wondered at, since with a non-zero velocity field, static pressure decreases compared to stagnation pressure (as can be seen from the Bernoulli equation). Further on, decrease of the static pressure causes the interface to subside below its static position, provided surface tension T is kept constant. This effect is shown in Tables 4.13-4.17 below. The tables are intended to demonstrate how corrections imposed on the static interface depend on initial surface shape h_0. All results are obtained for the same capillary number C=0.62, which satisfies the assumptions of the Small Deformation Theory. It may be demonstrated that with this value of C, for all cases presented below, ϵ is not larger than 0.05, and this value was found to satisfy the criterion for validity of the asymptotic method.

TABLE 4.13 : Deformation of the static interface,
when stagnation pressure $p_s = 0$ (C=0.62).

x_i	0.1	0.2	0.3	0.4	0.5
$h(x_i)$	1.0	1.0	1.0	1.0	1.0
$h = \epsilon h_1$	-5.6E-5	-1.1E-4	-1.56E-4	-1.88E-4	-2.E-4
curvature of Δh	.21E-3	.76E-3	.15E-2	.2E-2	.22E-2

TABLE 4.14: Deformation of the static interface,
when stagnation pressure p=1.5 (C=0.62).

x_i	0.1	0.2	0.3	0.4	0.5
$h(x_i)$	1.092	1.154	1.195	1.218	1.226
$h = \epsilon h_1$	-4.1E-3	-6.3E-3	-7.48E-3	-8.09E-3	-8.28E-3
curvature of Δh	.20E-2	.97E-3	.55E-3	.42E-3	.38E-3

TABLE 4.15 : Deformation of the static interface,
when stagnation pressure p_S=- 1.5 (C= 0.62).

x_i	0.1	0.2	0.3	0.4	0.5
$h(x_i)$	0.908	0.846	0.805	0.782	0.723
$h = \epsilon h_1$	-5.4E-2	-8.9E-2	-1.15E-1	-1.30E-1	-1.37E-1
curvature of Δh	.19E-1	.96E-2	.88E-2	.11E-1	.12E-1

TABLE 4.16 : Deformation of the static interface,
when stagnation pressure p_S= 1.85 (C= 0.62).

x_i	0.1	0.2	0.3	0.4	0.5
$h(x_i)$	1.158	1.244	1.297	1.326	1.335
$h = \epsilon h_1$	-6.9E-3	-8.4E-3	-9.00E-3	-9.30E-3	-9.37E-3
curvature of Δh	.54E-2	.86E-3	.34E-3	.19E-3	.16E-3

TABLE 4.17 : Deformation of the static interface, when stagnation pressure $p_s=-1.85$ ($\epsilon=-0.62$).

x_i	0.1	0.2	0.3	0.4	0.5
$h(x_i)$	0.842	0.756	0.703	0.674	0.665
$\Delta h = \epsilon h_1(x_i)$	-3.1E-2	-4.0E-1	-4.61E-1	-4.96E-1	-5.08E-1
curvature of Δh	.21E-0	.39E-1	.23E-1	.23E-1	.24E-1

TABLE 4.18 : Maximum Speed along the Free Surface (magnitude of scale V_s), for different static prssures.

p_s	0.0	1.5	1.85	-1.5	-1.85
V_s	7.25E-3	1.24E-3	8.80E-4	3.80E-2	8.E-2

It can be seen from the above tables and the following figures (Fig. 4.6.a, 4.6.b), that except for the flat surface, the correction Δh changes fastest near the solid boundary of the interface. This is reflected in the curvature of the correction Δh (see Tables 4.13 - 4.17). Another interesting feature is, that deformations Δh are much greater if the static interface is concave ($p_s=-1.5$, -1.85) compared to the convex cases ($p_s=1.5$, 1.85). This may be explained by the fact that for the concave static interface the velocity of the fluid at the free surface is larger (and so is z-gradient of the stream function), compared to the convex interface. It is of course assumed that the boundary conditions of the system remain unchanged, and that all results presented are referred to the same velocity scale V and length scale L. This implies that fixing the value of capillary number C (C=0.62) is equivalent to fixing the surface tension T for all cases selected for the numerical experiment. The correction Δh shown in tables and on Fig. 4.6 a-b, describes the deformation of the static free surface having constant surface tension, but different flow field. Changes of the flow field are entirely dependent on the shape of the static surface. We may conclude, as a consequence, that the same surface tension allows larger deformation of the free surface when the initial static interface is more concave.

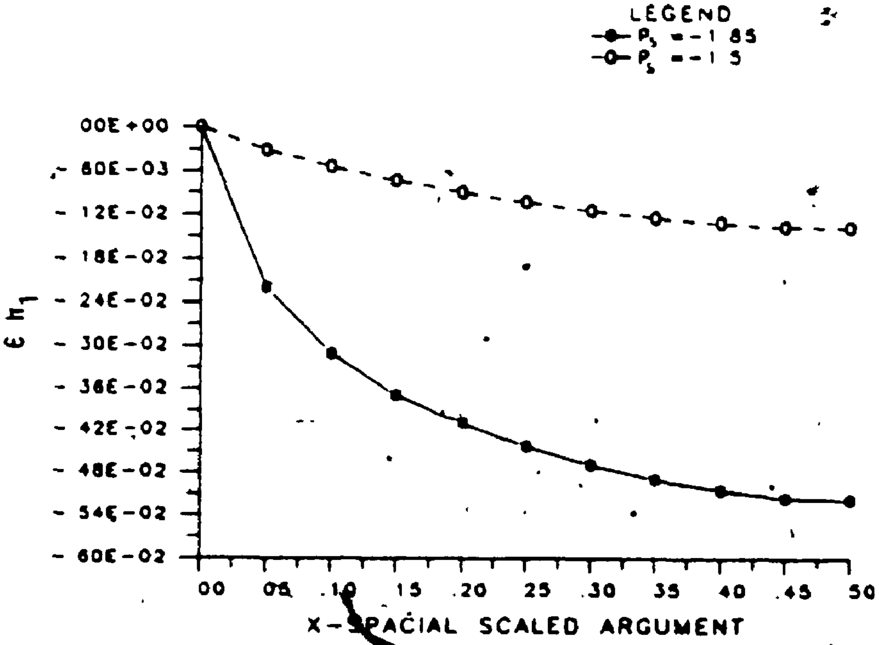


Fig. 4.6.a: First asymptotic correction ϵh_1

(p_s=-1.85, p_s=-1.5)

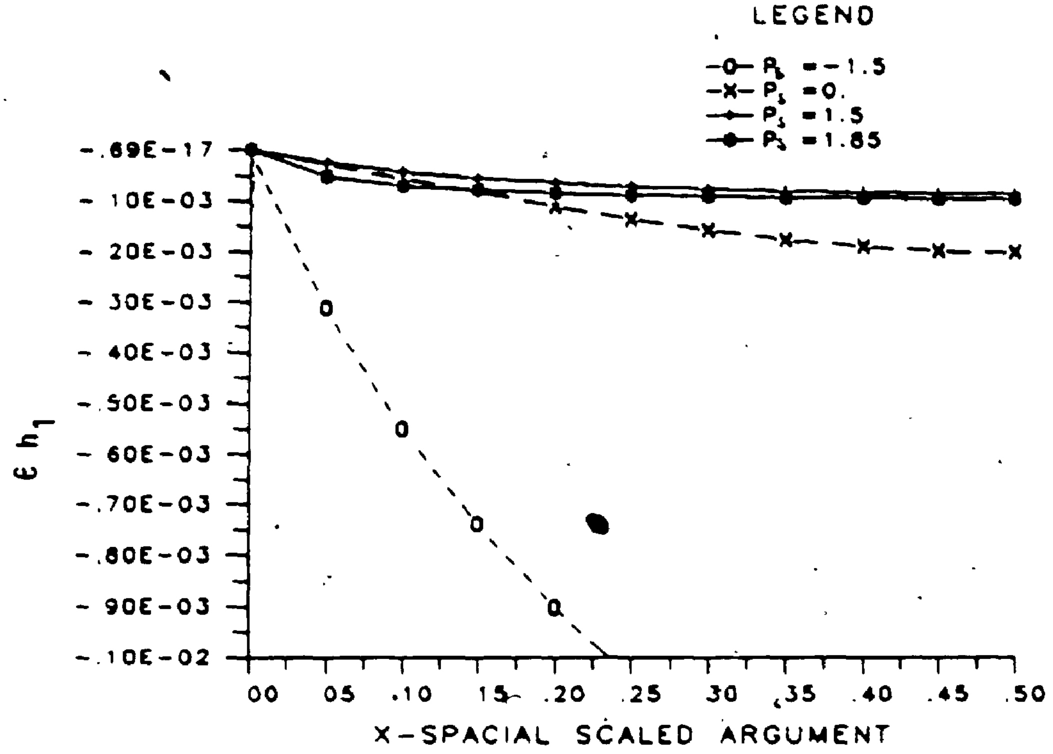


Fig. 4.6.b: First asymptotic correction ϵh_1

(p_s=-1.5, p_s=0, p_s=1.5, p_s=1.85).

CHAPTER 5

DIRECT ALGORITHMS

5.1 Introduction

As was stated in Sec.2.1, the analytical model of the physical flow under consideration, consists of the flow field equation and boundary equations, one of which is the balance of normal forces along a free surface. There are two unknowns in this system: the flow field and the free surface shape. In the preceding chapter, based on the Small Deformation Theory, we have separated the flow field solving algorithm from the one determining the free surface. In the Direct Method we solve the system for both unknowns simultaneously. It makes the computational scheme more complicated, but we evade restriction, which limits the asymptotic algorithm to cases showing only a small deformation of the static free surface.

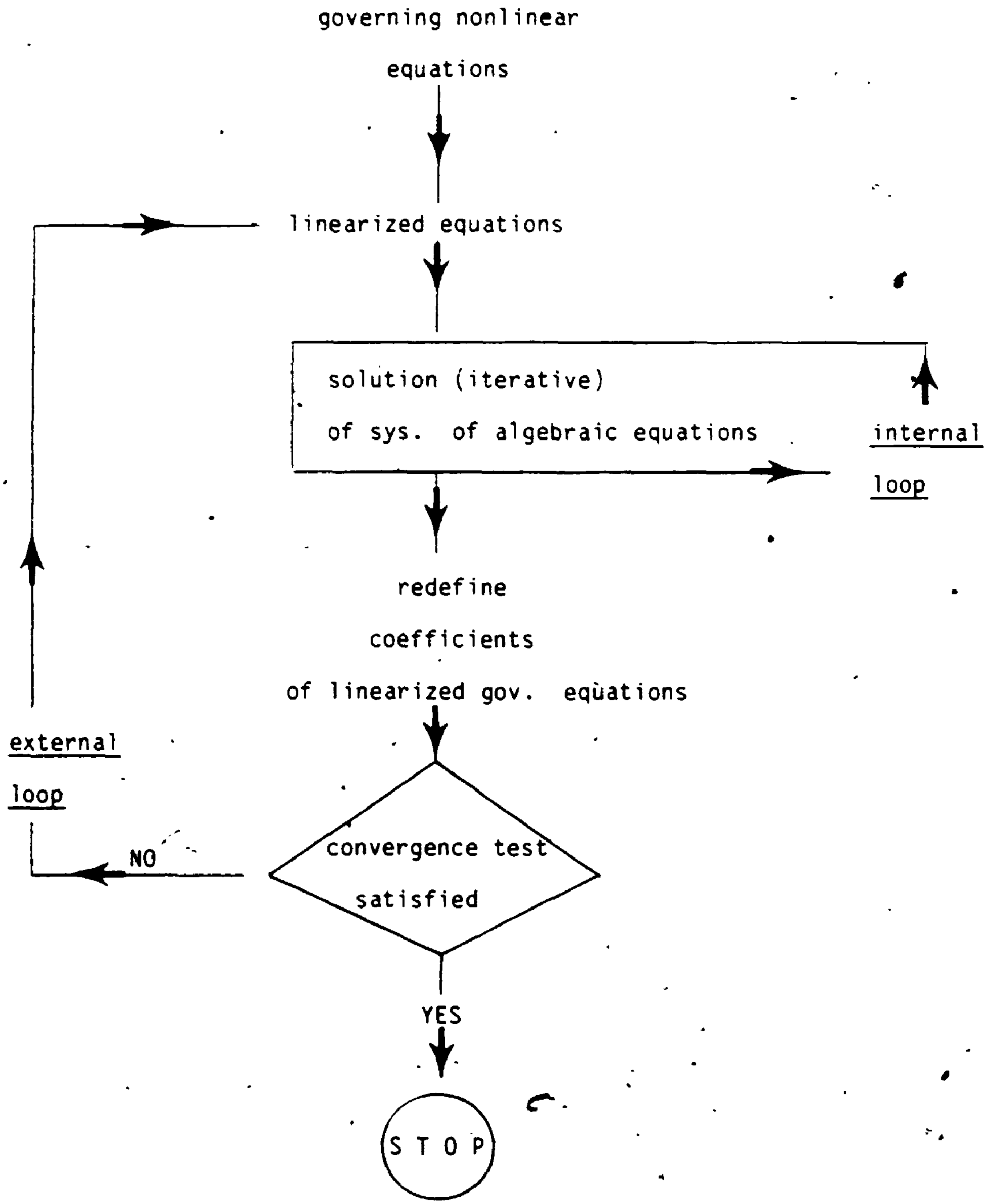
The problem of simultaneous solution for ψ and h, generates the mathematical problem of finding a solution to a system of nonlinear coupled differential equations. Both equations contain two unknown variables. For the sake of clarity, let us recall that the mathematical model of the flow is formulated as presented in Sec. 3.2 by Eq. (3.2.2)-(3.2.3).

If we choose Matrix Algebra as a vehicle to solve the above system, we have to arrange linearization. Therefore the resulting algorithm will be of iterative character. A particular method of linearization gives rise to specific algorithms. We will discuss two of them: the Picard Method (Sec.5.5) where linearization is of the first order, and One-Step Algorithm (Sec.5.6), which comes from the second order, Newton Raphson Linearization of the governing equations.

Apart from non-linearity of the system (3.2.2)-(3.2.3) we have mentioned also another problem: the number of algebraic equations. Since this number is excessive, we will not solve the discretized system by matrix inversion, but instead, by application of the approximating iterative scheme of ADI. This introduces another iterative procedure into the algorithm. As a result, the direct algorithm will consist of 2 iterative loops: the external one for nonlinearities, and the internal one for the approximate solution of the linearized system. These two loops may be logically implemented in a number of ways, and we will discuss this problem when discussing Picard and 1-Step Algorithms. The general scheme of direct algorithms is presented in a flowchart at the end of this section.

As was mentioned, the mathematical model consists of two equations. Whenever the algorithm is found to diverge, it was either by divergence of the flow field, or it was due to divergence of the solution to the equation of dynamic equilibrium at the free surface. The problem with convergence of the field equation is dealt with in subsections devoted to algebraic stability. The equation of the free surface deserves also a special attention and we elaborate on it in the first three sections of this chapter. Section 5.2 discusses the problem of the existence of the solution to this equation. To make that question tractable, we confine ourselves to the assumption that the velocity of the fluid along the free surface is known. The existence of the solution will be found to depend on this velocity, and values of the parameters p_s and C. In Sec.5.3 we elaborate on the numerical implementation of the existence conditions stated analytically in Sec.5.2. Since the existence of the solution of the free surface equation depends on assumed stagnation pressure p_s and capillary number C, we investigate in Sec.5.4 the range of the pair of feasible parameters (p_s, C). It is assumed, that the velocity of the fluid along the free surface is established as if C were set at zero. Therefore estimates of maximal feasible C for given p_s gives experimental upper bounds for the capillary number that assures the existence of the solution. When pursuing computations using Direct Methods it is found that C has to be smaller than the limiting value determined from the above estimation. This is discussed in a following chapter (Ch.7), where it is shown that interface is always beneath the static one, and the velocity of the fluid along the free surface is larger, compared to the flow with the static interface. Therefore, to keep the product $C|\underline{V}|^2$ small enaugh to assure the existence of the solution to the free surface equation, we have to decrease C. More detailed discussion of this topic can be also found in Sec.5.5.

Ideogram of direct algorithm


governing nonlinear equations
linearized equations
solution (iterative) of sys. of algebraic equations
internal loop
redefine coefficients of linearized gov. equations
external loop
convergence test satisfied
NO
YES
S T O P

$$\exists s \in [-2,0]: \forall x \in [x_L, x_R] : (g(x) \in [s, s+2]) \qquad (5.2.5)$$

Therefore the problem (5.2.3) has a solution if and only if (5.2.5) is satisfied. The above condition can be expressed in algorithmical form as a 2-step procedure:

$$\begin{array}{lll} 1)\ \max\{g(x)\} - \min\{g(x)\} < 2 & \text{for } x \in [x_L, x_R] & \\ 2)\ \min\{g(x)\} > -2 & & (5.2.6) \\ \quad \max\{g(x)\} < 2 & \text{for } x \in [x_L, x_R] & \end{array}$$

The above criterion is the necessary and sufficient condition for the existence of the solution to (5.2.2). When referred to our original problem (5.2.1), it is only the necessary condition, since the existence of a solution to (5.2.2) does not imply the existence of the solution to (5.2.1). This observation gives rise to the following:

Proposition 5.1

If (5.2.1) has a solution, then

$$\begin{array}{ll} 1)\ \max\{g(x)\} - \min\{g(x)\} < 2 & \text{for } x \in [x_L, x_R] \\ 2)\ \min\{g(x)\} > -2 & \\ \quad \max\{g(x)\} < 2 & \text{for } x \in [x_L, x_R] \\ \quad \text{where } g(s) = \int_0^s F(x)\,dx & \text{for } s \in [x_L, x_R] \end{array}$$

Now we may go one step further, and ask about the sufficient condition for the existence of a solution to (5.2.1).

unique solution in the interval $I = [x_L, x_R]$. In consequence it will be possible to establish conditions for the existence of the solution to (5.2.1).

Let us note that

$$\frac{h_{xx}}{(1+h_x^2)^{3/2}} = \frac{d}{dx}\left(h_x(1+h_x^2)^{-\frac{1}{2}}\right)$$

If we assume that $F(x)$ is integrable, we define $g(x) = \int_{x_L}^{x} F(x)dx$. Certainly, $g(x_L)=0$. Therefore:

$$g(x) = \int_{x_L}^{x} \frac{h_{xx}}{(1+h_x^2)^{32}} dx = \frac{h_{xx}}{(1+h_x^2)^{\frac{1}{2}}} - \frac{a}{(1+a^2)^{\frac{1}{2}}} \qquad (5.2.3)$$

Let $A = \frac{a}{(1+a^2)^{\frac{1}{2}}}$ by definition. From $a \in (-\infty, \infty)$, the following constraint is imposed on A: $-1 < A < 1$.

When solving (5.2.3), one obtains:

$$\frac{h_x}{(1 + h_x^2)^{\frac{1}{2}}} = g(x) + A$$

which can be re-arranged to the explicit form:

$$h_x = \frac{g(x\,,+A}{(1-(g(x)+A)^2)^{\frac{1}{2}}}$$

Note again, that $h_x(x_L) = \frac{A}{(1+A^2)^{\frac{1}{2}}} = a$

Finally we obtain:

$$h(x) = h(x_L) + \int_{x_L}^{x_R} \frac{g(x)+A}{(1-(g(x)+A^2)^{\frac{1}{2}}}\, dx \tag{5.2.4}$$

This formula yields solution h(x) if and only if there is a value A, $-1 < A < 1$ such that for every $x \in [x_L, x_R]$.the following holds:

$$1 - (g(x) + A)^2 \geq 0$$

or, in other words:

$$\exists A \in [-1,1] : \forall x \in [x_L, x_R] \; (-1-A < g(x) < 1-A)$$

which, after solving the two inequalities:

$$|g(x) + A| < 1$$

$$|A| < 1$$

can be simplified further on to the following form:

$$\exists\, s \in [-2,0]: \forall\, x \in [x_L, x_R] : (g(x) \in [s, s+2]) \qquad (5.2.5)$$

Therefore the problem (5.2.3) has a solution if and only if (5.2.5) is satisfied. The above condition can be expressed in algorithmical form as a 2-step procedure:

1) $\max\{g(x)\} - \min\{g(x)\} < 2 \quad$ for $x \in [x_L, x_R]$

2) $\min\{g(x)\} > -2$ $\qquad$ (5.2.6)

$\max\{g(x)\} < 2 \quad$ for $x \in [x_L, x_R]$

The above criterion is the necessary and sufficient condition for the existence of the solution to (5.2.2). When referred to our original problem (5.2.1), it is only the necessary condition, since the existence of a solution to (5.2.2) does not imply the existence of the solution to (5.2.1). This observation gives rise to the following:

Proposition 5.1

If (5.2.1) has a solution, then

1) $\max\{g(x)\} - \min\{g(x)\} < 2 \quad$ for $x \in [x_L, x_R]$

2) $\min\{g(x)\} > -2$

$\max\{g(x)\} < 2 \quad$ for $x \in [x_L, x_R]$

where $g(s) = \int_0^s F(x)\, dx \quad$ for $s \in [x_L, x_R]$

Now we may go one step further, and ask about the sufficient condition for the existence of a solution to (5.2.1).

Let us consider problem (5.2.2) again. Suppose it has the solution for some real A, say A ∈ D . This solution can be written as follows:

$$h(x) = h(x_L) + \int_{x_L}^{x_R} \frac{g(x) + A}{(1-(g(x)+A^2)^{\frac{1}{2}}} dx \qquad (5.2.7)$$

Let us fix our attention on $x=x_R$ which is the right boundary of the domain of argument x. We may easily notice that $h=h(x_R,A)$ is a continuous function of $A \in D_A$. Let $H = \{ h(x_R,A): A \in D_A \}$ by definition. Now, if h_R of (5.2.1) belongs to H, then certainly, the problem (5.2.1) has a solution. Obviously it can not happen that (5.2.1) has a solution and yet $h_R \notin H$. This provides the basis for the following statement:

<u>Prposition 5.2</u>

If H, which is the set of all possible values of $h(x_R,A)$ $(A \in D_A)$ contains the preassigned boundary value h , then (5.2.1) has a solution.

<u>NOTE</u>.

1) Assuming g(x) is continuous, from condition $-1-A<g(x)<1-A$ we may understand, that D_A is a single interval $D_A=[A_1, A_2]$. It is also easy to show, that $A_2=1-\max\{g(x)\}$, $A_1=-1-\min\{g(x)\}$.

2) Furthermore, continuity of h provided (see (5.2.7), we may

conclude that H is a single interval too.

3) Proposition 2 is based on the idea of sweeping the region $\{(x,y)/ x_L \leq x \leq x_R$, y real$\}$ with integral curves $h(x,A)$ for all feasible A to determine $H = [\min\{h(x_R,A)\} \max\{h(x_R,A)\}]$, and determining whether $h_R \in H$. This is graphicaly displayed on the figure below:

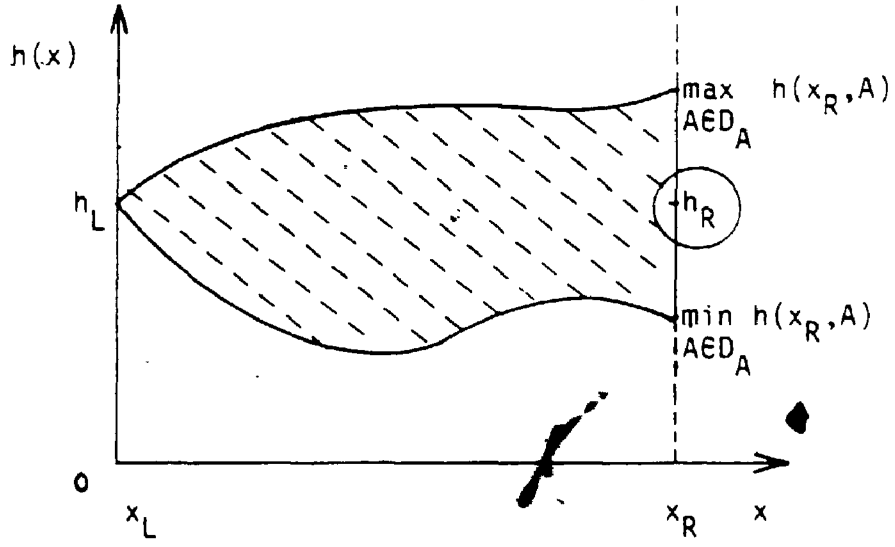


Fig.5.1 Family of solutions to (5.2.2)

From the fact that $h_R \in H$ we conclude that a solution to (5.2.1) exists.

It may also happen that the necessary condition is satisfied, i.e. $D_A \neq \emptyset$ (non-empty), but still (5.2.1) has no solution. This case is presented in Fig.5.2 below.

Having established the problem of the existence of the solution to (5.2.1), we may now address the issue of uniqueness of the solution, if it exists.

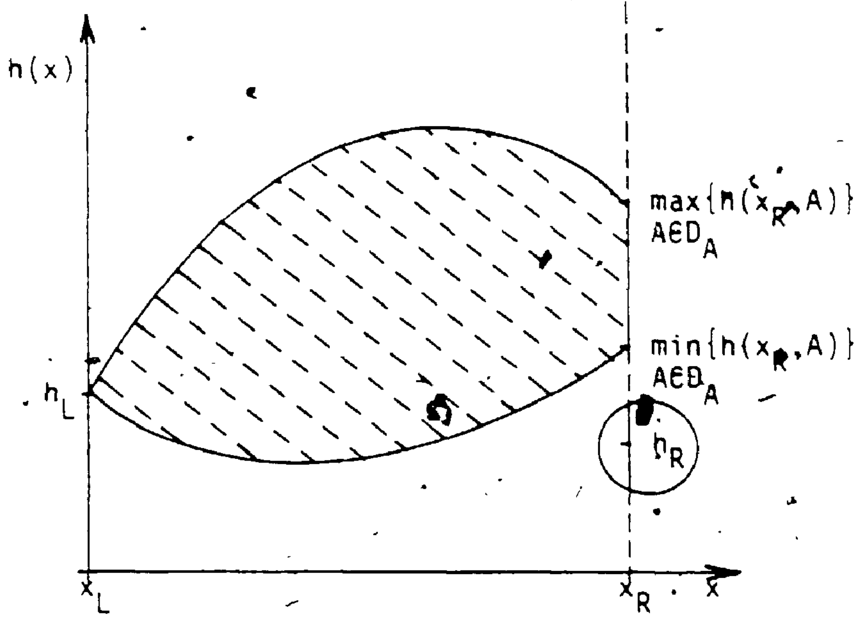


Fig.5.2 Family of solutions to (5.2.2) when there is no solution to (5.2.1)

Proposition 5.3

If the solution of (5.2.1) exists, it is unique.

Proof:

Suppose the solution of (5.2.1) exists. Let this be h(x). Therefore, because $h'(x_L)=a$ for some real a, then h is also a solution to (5.2.2) with the specified initial condition $h'(x_L)=a$. Now we will prove that (5.2.2) has a unique solution.

First, let us notice, that $A = A(a)$ is a monotonic (increasing) function of a:

$$\frac{dA}{da} = \frac{d}{da}\left[\frac{a}{(1+a^2)^{\frac{1}{2}}}\right] = \frac{1}{(1+a^2)^{3/2}} \quad > 0$$

Secondly: Let us note, that for the fixed $x \in [x_L, x_R]$ the ratio $(g(x)+A)/(1-(g(x)+A)^2)^{\frac{1}{2}}$ is an increasing function of A, hence an increasing function of a.

Indeed:

$$\frac{d}{dA}\left[\frac{g(x) + A}{1-(g(x)+A)^2)^{\frac{1}{2}}}\right] = \frac{1+(g(x)+A)}{1-(g(x)+A)^2)^{\frac{1}{2}}} \quad > 0$$

Suppose now, that (5.2.1) has another solution, say t(x). Then either $t'(x_L)=a$ or $t'(x_L)=a_t$, $a_t \neq a$. If $t'(x_L)=a$, then due to monotonicity $t(x)=h(x)$ is a unique solution of (5.2.1).

If $t'(x_L)=a_t \neq a$, it is also a solution to (5.2.2). But due to monotonic dependence of solution of (5.2.2) on initial slope $t'(x_L)=a_t$, the value $t(x_R)$ will not be equal to $h(x_R)=h$. Therefore t is not a solution to (5.2.1). This contradiction completes the proof.

Algorithmical criterion for existence of solution to (5.2.1)

In the course of the numerical integration of the system (3.2.2) - (3.2.3), we have transient estimates of unknown functions, and these estimators are determined on a discrete set of arguments. Therefore we cannot make use of analytical criteria for the existence of a solution. In particular we cannot apply directly Proposition 5.2 as a sufficient condition, since the analytical estimate of interval H is unavailable.

Therefore the logical arguments developed heretofore about the existence of the solution should be interpreted so as to provide the decision algorithm tailored to a numerical procedure for integrating the system of the governing equations.

This decision algorithm will consist of the following steps:

(A) Check $\max\{g(x_i)\}-\min\{g(x_i)\} < 2$

(B) Check $\min\{g(x_i)\} > -2$; $\max\{g(x_i)\} < 2$.

(C) If (A) or (B) is not satisfied, we conclude that (5.2.1) does not have a solution.

(D) Define $D_A=(A_1, A_2)$, where $A_1 = -1-\min\{g(x_i)\}$

$A_2 = 1-\max\{g(x_i)\}$

(E) Discretize D_A so that $D_A=(A_1,...,A_2)$

(F) Determine $H = \{h(x_R,A): A \in D_A\}$ along the formula (5.2.7) performing the numerical integration

$$\int_{x_L}^{x_R} \frac{g(x)+A}{1-(g(x)+A)^2)^{\frac{1}{2}}}dx$$

(G) If for the preassigned value $h(x_R)=h_R$ the following holds:

$$\text{infinum}(H) \leq h_R \leq \text{supremum}(H)$$

then the problem (5.2.1) has a solution.

The decision steps expressed by (C) and (G) are biased by treating continuous problems with discrete variables. There is no other way around, and this risk must be accounted for when performing computations.

5.3 Numerical Implementation of the Existence Condition.

The existence of a solution of the stream function equation (3.2.2) depends on the function describing free surface, as the function and its derivatives constitute coefficients of the stream function equation after arranging the transformation of coordinates. These coefficients must be bounded in order to be tractable by the algorithm solving flow field. The coefficients are functions of h, h_x and h_{xx}. We actually have to focus on h_x. If h_x is bounded, then h is bounded too (on the closed interval $[x_L, x_R]$), and h_{xx} is bounded provided F(x) is bounded, which can be inferred from Eq. (5.2.1):

$$h_{xx} = F(x)(1 + h_x^2)^{3/2} \quad .$$

It was found from numerical experimentation, that if F(x)=const $=-p_s$ and $x_L=0$, $x_R=1$, the Laplace equation solver works for $|p_s| \leq 1.96$, which is acceptable when taking into account the absolute limitation $|p_s|<2$. The numerical parameter, say B, limiting the variability of p_s, may depend on the intermediate and actual surface shape h(x), so in what follows, we will keep it as symbolic quantity B. Having this limitation in mind, we will repeat briefly the results of the previous section to come up finally with a decision rule regarding the existence of a solution to (5.2.1).

We remember that the solution of the interface equation is as stated by (5.2.7). To assure that h(x) does not have singularities, the argument of the square root must be positive. Next, to assure that $h(x)$ is finite, we demand that the integrand, i.e. dh/dx is bounded:

$$\frac{|g(x)+A|}{|(1-(g(x)+A)^2)^{\frac{1}{2}}|} \leq R$$

Analytically, the integrand may be singular without affecting its integrability. The numerical integration however is exposed to major difficulties if the integrand is very large. The above stated condition is to prevent such a situation occuring. This inequality, when solved, yields the following necessary condition of existence of $h(x)$ satisfying (5.2.7):

$$\max\{g(x_i)\} < 1+B$$

$$\min\{g(x_i)\} > -1-B$$

$$\max\{g(x_i)\} - \min\{g(x_i)\} < 2B$$

$$B = \frac{R}{(1+R^2)^{\frac{1}{2}}}$$

provided $A \in (-1,1)$ and $g(x) = \int_{x_L}^{x} F(x)dx$.

This in turn gives rise to the following constraints on A: $A \in [-B-\min\{g(x_i)\}, B-\max\{g(x_i)\}] = [A_1, A_2]$.

We are now on the way to determining the sufficient condition for the existence of the interface boundary value problem.

Let us determine $H = [H_1, H_2]$, which provides the estimate of the interval of possible values of function h at $x=x_R$:

$$H_1 = h_L + \int_{x_L}^{x_R} \frac{g(x)+A_1}{(1-(g(x)+A_1)^2)^{\frac{1}{2}}} dx$$

$$H_2 = h_L + \int_{x_L}^{x_R} \frac{g(x)+A_2}{(1-(g(x)+A_2)^2)^{\frac{1}{2}}} dx \tag{5.3.1}$$

If $h(x_R) \in H$, then we may conclude that the solution h(x) expressed by (5.2.7) does exist.

EXAMPLE 5.1

Let us consider the following boundary value problem:

$$\frac{h}{(1+h_x^2)^{3/2}} = -p_s .$$

$h = 1$ at $x = 0$

$h = 1$ at $x = 1$

$0 \le x \le 1$

We know that the analytical solution is a segment of a circle and it exists if $|p_s| < 2$. We will show in this simple example how our decision rule works.

First, due to the boundedness of p_s, we set B=0.96. Function $g(x)=-p_s x$. Suppose $p_s<0$ $(p_s>0)$. If so, then

$$\max\{g(x)\} = -p_s \qquad (=0)$$

$$\min\{g(x)\} = 0. \qquad (=-p_s<0)$$

Therefore the necessary condition of the existence of the solution is fulfilled, if

$$-p_s < 1+B = 1.96$$

Let us assume for simplicity that $p_s<0$. If so, $-B-\min\{g\} = -B$, $B-\max\{g\} = B+p_s$, that is $A_1 = -B$, $A_2 = B+p_s$. Therefore the range of values of $h(1)$ is $H = [H_1, H_2]$ where H_1 and H_2 are defined along (5.3.1) as below:

$$H_1 = 1 - \left[\frac{(1-B^2)^{\frac{1}{2}}}{p_s} - \frac{(1-(p_s+B)^2)^{\frac{1}{2}}}{p_s}\right]$$

$$H_2 = 1 + \left[\frac{(1-B^2)^{\frac{1}{2}}}{p_s} - \frac{(1-(p_s+B)^2)^{\frac{1}{2}}}{p_s}\right]$$

From the above we can see that $h(1)=1 \in H$, which is the interval symmetrical around the value of 1. This comfortable situation comes from the fact, that function $F(x)$ is constant. In cases like that whenever the necessary condition is satisfied, so is the sufficient condition. It will be found later on, that when $F(x)$ is not constant, this is not true.

5.4 Selection of the Initial Conditions for the Numerical Solutions

As was pointed out, the Direct Iterative Scheme allows us to deal with cases in which C is not necessarily small, as is required when Small Deformation Theory is applied. But it is understood that we can not admit the Direct Algorithm will work for any real C. In the preceding section it was proved, that the existence of an analytical solution of the free surface equation depends on a function F(x), which for our problem is $F(x)=-p_s+C|\underline{V}|^2/2$. Given non-zero velocity along the free surface we may easily see that for sufficiently large C, function f would violate the necessary or/and sufficient condition for the existence of the solution. Therefore for analytical reasons we must have constraints imposed on C. These constraints are bound to be strengthened when using the numerical algorithm, which may produce unfeasible intermediate results. The actual range of feasible values of parameter C is dependent on the following factors:

1. Iterative scheme, i.e. how we arrange alternation of field and interface computations,
2. Parameters of the stream function solving algorithm (relaxation),
3. Iterative approximation of the velocity at the interface,
4. The way we linearize the interface equation,
5. Relaxation applied to the interface equation.

Numerical experiments show, that it is in the interface equation that instability occurs first. This is evidently caused by the intermediate estimation of the surface velocity coupled with large C. It is too complex a task to establish an explicit relation between all the above factors to yield meaningful constraints on C. Therefore, instead of looking for general criterion of feasibility of C, we will try to establish experimental upper bounds for C with regard to the other control parameter, which is stagnation pressure p_s. This estimation will be done for the first approximation of the stream function which is obtained if a static interface is assumed ($C=0$). This is what we might call the start-up constraints on C. Next, keeping the flow field constant, we get determined right hand side function $F(x)$. Now we may ask what is the maximal feasible C so that the interface equation has a solution. In other words, for fixed p_s, and the resulting static flow field with the static free surface, we want to determine the range of C so that the first correction of the static interface does exist. Of course it does not mean, that for this range of C assumed, all subsequent equations for interface, as generated by the iterative procedure, will be solvable. This is why we say that we want to determine the upper bound for C given p_s. The experimental estimation of maximal feasible C (and respective ϵ values) are presented on Table 5.1 below.

TABLE 5.1 Upper bound for the capillary number C as a function of the stagnation pressure value (for the flow with the fixed free surface of constant curvature).

p_s	max C (exper.)	ϵ	max C (theor.)
-1.5	60	0.47	40
-1.	240	1.53	199
-0.5	540	2.67	500
0.	1010	3.71	990
0.5	1750	4.51	1750
1.	3040	5.21	3100
1.5	5650	5.8	6050

The results presented above are accurate up to 1%. The maximal value of C presented in the second column of the table (with respective ϵ) was obtained for each value of p_s by trial and error, which for each case required several computations to eliminate doubt about the convergence or divergence of the free surface solution. The estimations of maximal C given in the fourth column were obtained by application of the criterion of the existence of the solution. Therefore no iterations on the free surface were executed. Practically, for given p_s and the appropriate flow field, we were estimating interval H. From arguments developed in the Section 5.3 we know that if $h(x_R) \in H$, then sufficient condition for the existence of solution is satisfied. With C increasing, the range of H is decreasing. If C is sufficiently large, the value h(1)=1 is no more contained in H. By increasing C further we may come to the point, where H=Ø, that is theoretically we can speak of the inexistence of solution to the initial value problem (5.2.2) and in consequence to the boundary value problem (5.2.1). The reasoning is exemplified on Fig.5.3 for the case of $p_s=0$. Fig.5.3.a shows a convex hull of intervals H=H(C). For a clearer presentation of extreme values of C, we show H intervals on Fig.5.3.b. It can be seen, that when C is about 990, the value of $h(x_R)=h(1)$ is no longer in H (within the convex hull). It is understood then, that (5.2.1) has no solution. For C>1025 intervals H are empty, and we cannot even think about a solution to (5.2.2). These numbers of C which we take as terminal criteria of existence, were obtained by numerical implementation of theoretical criteria. They are therefore biased by numerical errors. It is due to these errors, that in reality experimental maximal C is not exactly as maximal C computed from existence criteria. The biggest discrepancy is seen for strongly concave static interfaces, like for $p_s=-1.5$ (see Table 5.3). This may be explained by the fact that with decreasing p_s, variation of speed along the free surface increases, and so happens to integrals like g(x) or estimates of boundaries to H interval.

It is interesting to note that if we pursue computations of the free surface with prohibitively large C (existence conditions vioparameters p_s and C. Having a complete scheme of the method we will compare the performance of the Picard Algorithm with the Asymptotic one. The last paragraph is devoted to the algebraic stability of the interface equation. Since the algebraic stability of the ADI scheme for the stream function solver has been already accomplished (Sec. 4.4), this will complete the analysis of algebraic good conditioning of the Picard Method.

5.5.2 Linearization scheme of the free surface equation.

Let us remind ourselves that free surface equation is

$$\frac{h_{xx}}{(1 + h_x^2)^{3/2}} = - p_s + \frac{C}{2}\psi_z^2 \frac{(1+h_x^2)}{h^2} = F(x) \qquad \text{(i)}$$

The right hand side of the above formula is assumed to be known. We treat velocity term ($|\underline{V}|^2 = \psi_z{}^2(1+h_x^2)/h^2$) as given from the preceding approximation of the flow field. In what follows we apply the Newton Raphson linearization scheme.

Suppose $\tilde{h}$ is an exact solution to (i). Let h^n be some approximation of $\tilde{h}$. Assume also, that $\tilde{h} = h^n + \delta h$, $\delta h < 1$ ($(\delta h)^n \approx 0$ for n>1). If we put $\tilde{h} = h^n + \delta h$ into (i), we will get:

$$\tilde{h}_{xx} = F(x)(1+(h_x^n+\delta h_x)^2[1+(h_x^n+\delta h_x)^2]^{\frac{1}{2}} \qquad \text{(ii)}$$

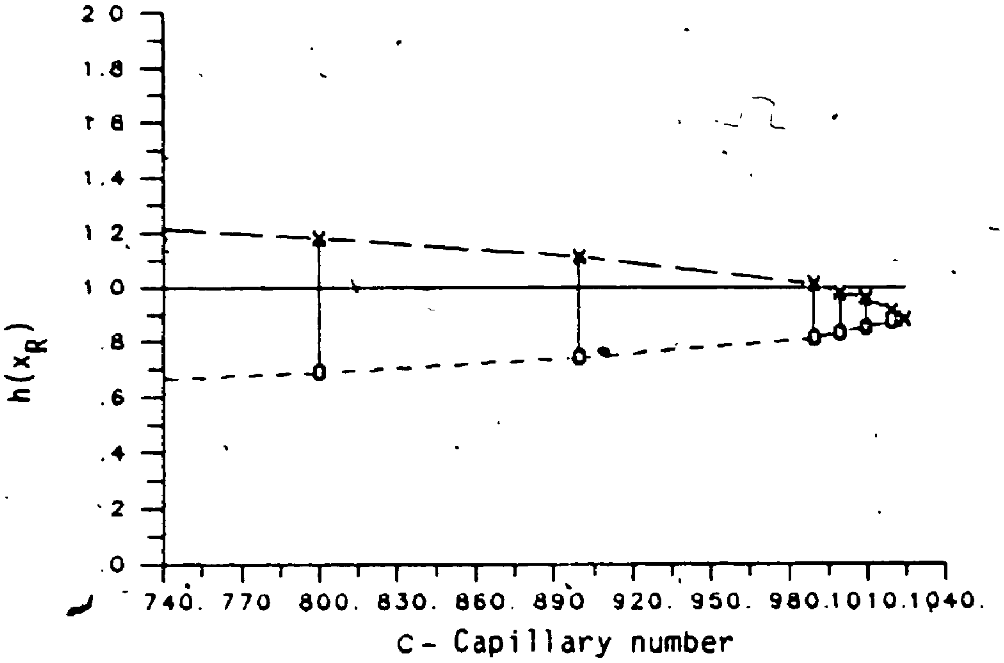


Fig. 5.3.b - Convex hull of H intervals for the Problem(5.2.2); windowing for large values of C.

5.5 Picard Method

5.5.1 Introduction.

The mathematical model of the flow consists of two equations with two unknowns: the stream function and the shape of the free surface. the Picard Method of solving this system is based on treating it as a transient or pseudo unsteady process in which the initial flow field and initial free surface shape are assumed, and with 'time' elapsing, both flow and free surface evolve towards a steady state. If one understands this evolution as the alteration of flow corrections upon interface corrections, this is what historically is known as the Picard Algorithm [8]. The scheme of alternating corrections can be derived more formaly from the analytical point of view. The two equations are nonlinear with respect to two unknown functions ψ and h. As was shown in Sec.5.1, we can separate the flow field equation from the free surface equation by appropriate linearization: in the former one, the free surface is treated as known, while in the latter, the velocity along the interface is assumed to be known. This aprroach disconnects both equations and allows for separate computations on each of them. Nevertheless nonlinearity is still present - it is a free surface equation, which is nonlinear with respect to unknown h(x). Therefore the first topic of this section will be about linearization of the free surface equation. Further on we pass to the problem of establishing the most effective flow field - interface iterative scheme and determination of range of feasible values of parameters p_s and C. Having a complete scheme of the method we will compare the performance of the Picard Algorithm with the Asymptotic one. The last paragraph is devoted to the algebraic stability of the interface equation. Since the algebraic stability of the ADI scheme for the stream function solver has been already accomplished (Sec. 4.4), this will complete the analysis of algebraic good conditioning of the Picard Method.

5.5.2 Linearization scheme of the free surface equation.

Let us remind ourselves that free surface equation is

$$\frac{h_{xx}}{(1+h_x^2)^{3/2}} = -p_s + \frac{C}{2}\psi_z^2\frac{(1+h_x^2)}{h^2} = F(x) \qquad (i)$$

The right hand side of the above formula is assumed to be known. We treat velocity term $(|\underline{V}|^2 = \psi_z{}^2(1+h_x^2)/h^2)$ as given from the preceding approximation of the flow field. In what follows we apply the Newton Raphson linearization scheme.

Suppose $\tilde{h}$ is an exact solution to (i). Let h^n be some approximation of $\tilde{h}$. Assume also, that $\tilde{h} = h^n + \delta h$, $\delta h < 1$ $((\delta h)^n \simeq 0$ for $n>1)$. If we put $\tilde{h} = h^n + \delta h$ into (i), we will get:

$$\tilde{h}_{xx} = F(x)(1+(h_x^n+\delta h_x)^2[1+(h_x^n+\delta h_x)^2]^{\frac{1}{2}} \qquad (ii)$$

By expanding square root into power series:

$$(1+(h_x^n+\delta h_x)^2)^{\frac{1}{2}}=(1+h_x^n)^2)^{\frac{1}{2}}=\delta h_x \frac{h_x^n}{(1+(h_x^n)^2)^{\frac{1}{2}}}+ (\delta h_x) + \ldots$$

and substituting it into (ii), we obtain:

$$\begin{aligned} h_{xx}^n + \delta h_{xx} = {} & F(h_x^n+\delta h_x)\ (1+(h_x^n)^{2\frac{1}{2}}h_x^n + F(1+(h_x^n)^2)^{\frac{1}{2}} \\ & +2F\delta h_x h_x^n(1+(h_x^n)^2)^{\frac{1}{2}} + O(\delta h^2) \end{aligned} \qquad \text{(iii)}$$

Finally, putting $\delta h=h^{n+1}-h^n$ we obtain the following iterative scheme:

$$h_{xx}^{n+1} - 3h_x^{n+1}(Fh_x^n(1+(h_x^n)^2)^{\frac{1}{2}}=F(1-2(h_x^n)^2)\ (1+(h_x^n)^2)^{\frac{1}{2}} \qquad (5.5.1)$$

This linearization scheme theoretically is of the order $O(\delta h^2)$. Obviously, it may be relaxed on grounds of assumption, that $\delta h \to 0$. Therefore, dropping the last term on the right hand side of (iii), we obtain a simplified linear scheme of the form, which also easily could have been obtained heuristically:

$$h_{xx}^{n+1} - h_x^{n+1}(Fh_x^n(1+(h_x^n)^2)^{\frac{1}{2}} = F\ (1+(h_x^n)^2)^{\frac{1}{2}} \qquad (5.5.2)$$

Further simplifications are also possible, but all of them lead to a decrease of the order of the resulting scheme. The highest order is provided by (5.5.1). In fact, when comparing the convergence rate of (5.5.1) with that of (5.5.2), we may find that (5.5.1) provides much faster convergence than (5.5.2). This is shown in the following table:

TABLE 5.2 Experimental comparison of rates of convergence for linearization schemes of the first and second order.

Case param.		# of iterations	
p_s	C	scheme (5.5.2)	scheme (5.5.1)
0.	990	34	7
0.	500	9	4
1.5	5650	53	8
1.5	4000	8	5
-1.5	40	18	5
-1.5	10	13	4

The fast convergence of (5.5.1) is due to keeping all terms of the power series expansion but those with $(\delta h)^n$, $n \geq 2$. This observation leads to the following:

Proposition 5.1

Scheme (5.5.1) provides quadratic convergence, i.e.:

$$\frac{[h^{n+1} - h^n]}{[h^n - h^{n-1}]^2} = O(1)$$

To demonstrate the above statement, let us note that if $h^k = \tilde{h} + \delta h_k$ then (5.5.2) is accurate up to $O(\delta h_k^2)$. Therefore $h^{k+1} = \tilde{h} + O(\delta h_k^2) = \tilde{h} + O(\delta h_{k-1}^4)$ which by induction yields:

$$h^n = h + O(\delta h^{2^{n-1}})$$

In consequence we obtain:

$$\frac{[h^{n+2} - h^{n+1}]}{[h^{n+1} - h^n]^2} = \frac{[(h^{n+2} - h) - (h^{n+1} - h)]}{[(h^{n+1} - h) - (h^n - h)]^2}$$

$$\frac{[(\delta h)^{2^{n+1}} - (\delta h)^{2^n}]}{[(\delta h)^{2^n} - (\delta h)^{2^{n-1}}]^2} = \frac{(\delta h)^{2^n}[(\delta h)^{2^n} - 1]}{(\delta h)^{2^n}[(\delta h)^{2^{n-1}} - 1]} = O(1)$$

as $n \longrightarrow \infty$

To illustrate this, let us look at the three following tables, which present successive iterative approximations of free surface for three different flow fields. Each flow field was computed separately for distinct values of stagnation pressure (p_s=-1.5, 0., 1.5). Values of capillary parameters C were taken as extremal feasible ones for each of distinguished p_s cases. Convergence rate, c, which in limit should be of value 2, is calculated as

$$c = (\ln[h^{n+1} - h^n])/(\ln[h^n - h^{n-1}]).$$

TABLE 5.3 : Order of linearization of the interface equation when $p_s = -1.5$, C = 40.

iteration number	$\|h^n - h^{n+1}\|$	c
1	.21E-1	-
2	.51E-2	1.37
3	.21E-3	1.6
4	.34E-6	1.76
5	.85E-12	1.86

TABLE 5.4 : Order of linearization of the interface equation when $p_s = 0.$, C = 990.

iteration number	$\|h^n - h^{n+1}\|$	c
1.	.72E-1	-
2	.54E-1	1.1
3	.31E-1	1.2
4	.81E-2	1.4
5	.40E-3	1.6
6	.71E-6	1.8
7	.18E-11	1.9

TABLE 5.5 : Order of linearization of the interface equation when $p_s = 1.5$, $C = 5650$.

iteration number	$\lVert h^n - h^{n+1} \rVert$	$\underline{c}$
1	.81E-1	-
2	.54E-1	-
3	.22E-1	-
4	.80E-2	-
5	.16E-2	-
6	.79E-4	1.47
7	.29E-6	1.6
8	.11E-11	1.83

Where c values are not displayed, the corresponding results are irrevelant due to the magnitude of initial δh . It is important, as pointed out in Proposition above, that to get quadratic convergence, or nearly, we have to be close to the ultimate solution ($\delta h << 1$). In real computations, as shown in the tables, an initial guess for h may be fairly far away from the actual solution, therefore at the outset of the iterative process we can not expect c to be close to 2.

For the sake of completeness, we present below the results of the iterative solution of the problem of similar analytical structure as (5.5.1), but with a known analytical solution. This problem has a constitutive equation as follows:

$$\frac{h_{xx}}{(1+h_x^2)^{3/2}} = \frac{-2\pi .\ \sin(2\pi x)}{(1+\cos\ (2\pi x))}$$

$$0 \leq x \leq 1$$

$$h(0)=h(1)=1.$$

The exact analytical solution has the form: $h(x)=1+\sin(2\pi x)/(2\pi)$. Results are shown in the following table:

TABLE 5.6 : Distance between the numerical solution and the exact one; order of convergence of Newton Raphson linearization.

iteration number	$\|h^n - \tilde{h}\|$	$\|h^n - h^{n+1}\|$	c
1	.77E-2	.28E-1	-
2	.26E-3	.75E-2	1.37
3	.276E-3	.42E-3	1.6
4	.277E-3	.10E-5	1.77
5	.277E-3	.56E-11	1.88
6	.277E-3	.59E-14	-

All results presented on tables 5.3 - 5.6 of this section were obtained by computation on a grid of $\Delta x=0.1$.

The distance of the computed value from the exact solution : $\|h^n-\tilde{h}\|$ is a discretization error, which depends on grid size. This error is contrasted with data shown in third column, which shows how results converge to the exact numerical solution of difference equation (with grid spacing fixed). We have included the data of column two to make a clear distinction between the rate of convergence of the iterative process (column 3) and discretization error (column 2). Limit values of column 2 would converge to zero if finer grids were used for computations.

Table 5.6 indicates there were only 6 iterations of the interface equation needed, to obtain results up to computer accuracy. It is worth mentioning, that if we applied the simplified linearization scheme (5.5.2), this would result in an iterative run four times longer.

Conclusions.

In this Section the Newton Raphson linearization of the free surface equation was developed. The resulting scheme (5.5.1) is more involved than the heuristic linearization (5.5.2), but provides much faster convergence of the resulting iterative scheme. The convergence of the iterative procedure was shown to be asymptotically quadratic. All examples presented were referred to the fixed flow field computed separately with preassigned stagnation pressure p_s. So far we were not considering commutative iterations on flow and interface, but focussed attention on several features of the free boundary equation. It was the problem of the existence of solution (Sec.5.2), implementing numerical criteria of solvability of the equation (Sec.

5.3), and finally tuning the algorithm by determining the best linearization scheme. In the preceding Chapter we accomplished the analysis of the algorithm which solved the flow field (part of the Asymptotic Algorithm). In this Chapter we were studying in detail algorithm solving the free surface equation. Now we proceed, and putting these two parts together, analyse the algorithm solving jointly flow field ard interface.

5.5.3 Capabilities and limitations of the algorithm.

As was stated in Sec.5.1, the Picard Method is a particular option of the Direct Algorithm. Its main feature is that the original problem is split into two components: the flow field and the free surface. The actual performance of the Picard Method is commuting of computations on the flow field and interface. This is the iterative procedure, which was discussed in Sec.5.2. Determination of the iterative scheme is the first endeavour, which we will address. Next, we will try to determine the range of feasible parameters p_s and C. This was partially accomplished in Sec.5.4, but for interface only, since the flow field was kept constant. Now we will characterize the range of p_s and C for the general algorithm, solving jointly the flow field and the free surface shape. We will address also the question of how the cost of computations depend on the values of these parameters.

Iterative Scheme.

In the preceding section we found that computations of the free interface equation, if separated from the flow field, were very fast. For the cases presented in Tables 5.3, 5.4, 5.5, the algorithm yielded a highly accurate solution within no more than 7 iterations, while flow field computations take no less than 21 (see Table 4.2 of Section 4.2). It may indicate, that the overall convergence rate is more dependent on the efficiency of the flow field solver, than on the free surface. In Table 4.2 of Sec.4.2 we have listed a number of iterations required to obtain solution for the stream function equation, if the free surface was kept fixed. We may observe, that the rate of convergence of the flow field is essentially the same for any fixed static interface. Therefore, the actual field/interface iterations depend on how we arrange the iterative scheme, and on the initial guess and parameters of the free surface equation. It was found that assuming the initial guess for the stream function ψ=0, the algorithm converged slightly better if initial h was set constant equal to 1 compared with setting h as the appropriate static interface.

As for the iterative scheme, it is difficult to determine the unique procedure which would always yield the fastest convergence. It was suggested in literature (Ryskin & Leal [23]) that the scheme consisting of 1 iteration on the flow field followed by one iteration on the interface is most economical. In our experimentation with the Picard Algorithm this was generally found to be valid. There are, however, situations, in which the other scheme was found to work better. For example, if $p_s=1.5$, and C=550, the one-to-one iterative scheme took 105 iterations to get to the accuracy of the machine. But when we employed two runs on the flow field followed by a single run on the free surface, computations were completed within 81 iterations, which is an improvement of about 20%. These numbers were obtained on the grid of spacing $\Delta x=0.05$. Capillary parameter C=550 is about its maximal value assumed $p_s=1.5$, for which the algorithm is convergent. With the decrease of the value of C, the advantage of the 2/1 scheme over the 1/1 scheme diminished, and at moderate values of C (e.g. C=100), scheme 1/ 1 was more effective. For negative p_s , the 1/1 scheme was always better than any other one.

Domain of Convergence

This concept is referred to feasible values of parameters p_s and C which are decisive in the existence or nonexistence of the solution. It was stated in the chapter devoted to the Asymptotic Algorithm, that the stream function in the domain bounded by the static free surface can be numerically computed for $p_s \in [-1.96,1.96]$, which is very close to the theoretical range of variation of : $p_s \in (-2,2)$. Also, assuming this flow field computed, we found bounds for parameter C so that the interface equation could be numerically solved. These bounds were presented in Sec.5.4. In establishing these bounds, use was made of theoretical conditions of existence of the solution to the equation of free surface. Now, the numerical implementation of these conditions was also placed in the Picard Algorithm. Whenever the algorithm moves from flow field to interface computations, with negligible expenditure of time we check the validity of these conditions. If the necessary or sufficient condition is violated, the execution is halted. This may save a lot of computational time, since the instability effect, like increase of residuals, may be significantly delayed in time. Thus application of this Automatic Stop routine may inform us well ahead that the algorithm is not about to converge.

It was experimentally confirmed, that whenever Automatic Stop gave the warning about violation of the existence criterion, the algorithm eventually was found to be divergent. On the other hand, it was also established, that close to the maximal value of C, the criterion was fulfilled, but the algorithm was yielding nonconverging, oscillating residuals. In these cases sufficient condition for existence was satisfied, but was very close to violation. As an example, let us consider the case, when p_s= -1.5, and C=20, Δx=0.05. The residuals of the stream function equation were oscillating around zero with maximal amplitude of 0.6. The interval H (see formula (5.3.1)) had the lower bound oscillating about 0.9 ± 0.01, while the upper one stabilized at 1.03. It was said, that to have the sufficient condition satisfied, we must have h(1)=1 contained in the interval H. This is provided, but the upper limit of H is so close to 1, that we cannot be certain whether it is coincidental due to numerical error in the implementation of the existence condition, or to inherent oscillatory instability of the flow field - interface numerical solver.

Below we present the table of experimental upper bounds of C (and resultant C), as determined by computations on a grid with space increment $\Delta x = 0.1$:

TABLE 5.7 : Upper bound for the capillary number C as a function of the stagnation prassure value.

p_s	interface equ. only. C	(ϵ)	Picard Algorithm C	(ϵ)	# of its.
-1.5	60	(.47)	19	[illegible]	103
-1.	240	(1.53)	51	(.64)	116
-0.5	540	(2.67)	100	(1.02)	228
0.0	1010	(3.71)	160	(1.07)	72
0.5	1750	(4.51)	260	(1.43)	128
1.	3040	(5.21)	400	(1.78)	350
1.5	5650	(5.8)	600	(1.65)	204

The second column presents for comparison the results coming from computations of interface only. In this case the stream function was kept constant as determined by p_s and C=0. In the Picard Algorithm the stream function is computed along with the interface, and this causes significant decrease in the feasible values of C. It is partially due to the fact that actual interface is always beneath the static one, (see Chapter 7), therefore the actual velocity of the fluid along the free surface is larger than in the case of the static free surface. This in turn causes the term $C|\underline{V}|^2$ to increase. To keep this term in limits imposed by existence conditions, the maximum feasible C must by adjusted downwards.

5.5.4 Algebraic stability of the Picard Method.

The algebraic Stability of the system of equations resulting from the ADI routine was discussed earlier (Sec. 4.4). The Picard Algorithm solves iteratively not only flow field equations, but also the equation of the free surface. Hence the sufficient condition for the algebraic stability of the Thomas Algorithm (for the free surface equation) should also be addressed.

Let us remind ourselves, that the iterative scheme for the free surface equation is

$$wh^{n+1} - h^{n+1}_{xx} + 3h^{n+1}_{x} F(x) h^{n}_{x} [1+(h^{n}_{x})^2]^{\frac{1}{2}} = F(x)[2(h^{n}_{x})^2-1][1+(h^{n}_{x})^2]^{\frac{1}{2}} + wh^{n}$$

where: $F(x) = C\,\psi_z^2\,[(h^n_x)^2+1]/2(h^n_x)^2$

$\psi_z = \psi_z\,(x,z),\ z=h(x)$

w : relaxation parameter ($1/w$ = pseudo-time increment)

Discretization of the differential equation like the one above, leads to the following 3-diagonal system of linear equations:

$$-a_j h_{j-1} + b_j h_j + c_j h_{j+1} = d_j \; ; \qquad j=2,\ldots n-1$$

The first stability condition is that a_j, b_j, c_j are all positive (see in Smith [27]). It can be proven, that this restriction, when referred to equation (5.5.1), yields the following limitation on a spacial grid increment:

$$\Delta x \leq \frac{2}{3|Fh_x|(1+h_x^2)^{\frac{1}{2}}} \qquad (5.5.3)$$

$$x = x_1,\ldots x_n$$

The next condition is, that $b_j > a_j + c_j$. This is satisfied if relaxation w is nonnegative.

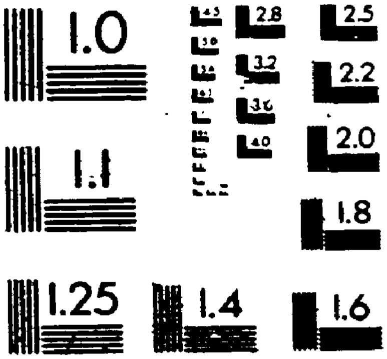
The last condition: $b_j > a_{j+1} + c_{j-1}$ yields constraint on relaxation factor w:

$$w(2\Delta x) > 3h_x(x_{j-1})[1+h_x^2(x_{j-1})]\; F(x_{j-1}) \qquad (5.5.4)$$
$$-3h_x(x_{j+1})[1+h_x^2(x_{j+1})]\; F(x_{j+1}) \qquad j=2,\ldots,n-1$$

EXAMPLE 5.1

For the physical problem with parameters: $p_s = 1.8$, $C=150$, the free surface is shown on Fig.7.4.e of Chapter 7. The constraints calculated in accordance with the above given conditions are as follows:

3



$$r \leq .18$$

$$w > 59 \qquad (1/w) < 0.017 .$$

For another problem, with $p_s = -1.43$, $C=17$, the interface is of concave shape, as on Fig.7.2.c and the restrictions are:

$$r \leq .315$$

$$w > 27 \qquad (1/w) < 0.036$$

The relations discussed above, constitute sufficient condition for algebraic stability. The resulting constraints (5.5.3) and (5.5.4) relate to the algebraic model of the free surface equation. To close the considerations on algebraic stability, the appropriate constraints resulting from the algebraic model for the ADI routine (flow field equation) should be attached. These constraints were derived in Sec.4.4. If w_ψ is to denote relaxation for ψ-field, w_h for the free surface equation, the application of the Picard Method imposes the following constraints to assure algebraic stability:

$$w_h > \frac{3}{2r} h_x(x_{j-1})[1 + h_x^2(x_{j-1})]^{\frac{1}{2}} F(x_{j-1})$$

$$- \frac{3}{2r} h_x(x_{j+1})[1 + h_x^2(x_{j+1})]^{\frac{1}{2}} F(x_{j+1}) \cdot \tag{5.5.5}$$

$$w_\psi > \frac{h_{xx}}{h}$$

$$w_h > 0 \qquad \text{for } j = 2,\ldots,n-1$$

$$r \leq \frac{2}{3|h_x A|[1 + h_x^2]^{\frac{1}{2}}} \qquad \text{for } x = x_T, \ldots x_n$$

$$r \leq \begin{cases} \dfrac{4|h_x|}{|h_{xx}h - 2h_x^2|} & \text{for } |h_x| \geq 1 \\ \dfrac{2|1 + h_x|}{|h_{xx}h - 2h_x^2|} & \text{for } |h_x| \leq 1 \end{cases}$$

EXAMPLE 2

Let us consider physical flows with fixed physical stagnation pressure $p_s^*=0.1$ and the dynamic characteristic pressure $D=\rho V^2= 38$, but variable surface tension T. The table below shows estimates of r and w for 3 different values of T. The actual free surface (which is convex in this case) is plotted on Fig. 7.1.a,b,c of Chapter 7.

TABLE 5.8 : Stability constraints with surface tension decrease, for a convex solution domain ($p_s^*=.1$, D=38).

			interface	flow field
T	$w_h >$	$w_\psi >$	$r \leq$	$r \leq$
1.00	.42	.042	-	-
.10	2.66	.42	-	-
.048	88.00	-	.44	.13

For comparison we present below the table of constraints for flows of distinct geometry (concave interface). As before, pressure and dynamic parameters are kept constant ($p_s^* = -1$, D = 12), while surface tension T is being changed. The shapes of interface for the respective flows are shown pn Figures: 7.2.a,b,c.

TABLE 5.9 : Stability constraints with surface tension decrease, for a concave solution domain.

			interface	flow field
T	$w_h >$	$w_\psi >$	$r \leq$	$r \leq$
1.00	5.35	1.33	-	-
.80	12.66	2.61	.55	-
.70	27.	4.49	.31	.13

5.5.5 Comparison of performance of the Asymptotic and Direct Algorithms.

This comparison may be performed for only those values of parameters for which both algorithms are valid. It is the Asymptotic Algorithm, which is more restrictive in choice of parameters p_s and C. Hence, quite naturally the Picard Algorithm has one decisive advantage over the Asymptotic Algorithm in this respect. The second criterion is the rate of convergence of the iterative process. Below a table is shown, in which we produce the number of iterations required to get to machine accuracy for both algorithms. C is assumed sufficiently small, so that $\epsilon \leq 0.05$, which we recall was the general restriction for Asymptotic Algorithm if $-1.5 \leq p_s \leq 1.5$. The resultant values of C for Picard Algorithm may be slightly different, since we are unable to predict the final value of velocity $\underline{V}$ before completion of the computations of the stream function.

TABLE 5.10 : Comparison of the number of iterations for Asymptotic and Picard Algorithms, r = 0.1.

	Asympt. Alg.	Picard Alg.		
p_s	# of its (ϵ=0.05)	# of its	C	ϵ
-1.5	27	27	4.0	0.048
0.0	22	21	13.8	0.053
1.5	26	29	50.0	0.048

TABLE 5.11 : Comparison of the number of iterations for Asymptotic and Picard Algorithm, r = 0.05.

	Asymptotic Alg.	Picard Alg.		
p_s	# of its (ϵ=0.05)	# of its	C	ϵ
-1.5	48	46	4.0	0.048
0.0	22	21	13.8	0.053
1.5	43	39	50.0	0.049

TABLE 5.12 : Comparison of the number of iterations for Asymptotic and Picard Algorithm, r = 0.025

	As. Alg.	Picard Alg.		
p_s	# of its (ϵ=0.05)	# of its	C	ϵ
-1.5	89	90	4.0	0.049
0.0	26	25	13.8	0.053
1.5	89	105	50.0	0.05

As can be seen, the rate of convergence of the two algorithms is practically the same and this similarity does not change when the grid step size r is changed. Therefore alterations of flow field and interface iteration, as in the Picard Method, do not affect the speed of convergence. As for the time cost, we show in the table below the CPU time required for the single iteration on flow field and interface:

TABLE 5.13 : CPU (CYBER 825) time per single iteration on the equation of free surface and one full ADI iteration.

	grid spacing r			
	0.1	0.05	0.025	0.0167
flow field	.08	.35	1.4	3.4
free surf.	.003	.006	.013	.02

When using the Picard Algorithm, generally we perform as many iterations on the flow field equation as on the interface. Compared to the Asymptotic Algorithm, this implies additional costs of multiple iterations on the free surface. From the Table above we see however, that this cost is practically negligible, thus the Asymptotic Algorithm is just marginally superior in this aspect to the Picard Method. This cannot offset however the strong limitations imposed on the Asymptotic Algorithm by the requirement of using only small values of parameter ζ (ϵ).

5.6 Direct 1-Step Algorithm

5.6.1 Linearization of the governing equations.

In the presentation of the performance of the Direct Algorithm, we referred heretofore to the Picard Method. This method is based on separating the flow field and interface equations as distinct algorithmical units. In consequence, we can iterate independently on the flow field or the interface, transmitting information from stream function solver to interface solver or vice versa, whenever the assumed iterative cycle was completed. The Picard Method is convenient also because of the other reason: the analysis of the resulting algorithm can be broken down into the analysis of the stream function solver (discussed in Chapter 4) and interface solver, which was accomplished in this chapter. Heuristically however, the Picard Method is the simplification of a more general approach, which we called 1-Step Algorithm. Its conceptual framework was presented in Sec. 5.1. Let us remind ourselves, that in the 1-Step Algorithm, both equations of the mathematical model (flow field and interface equations) are treated as an inseparable pair inasmuch as use is made of them in determining two variables: stream function ψ and free surface shape h. In what follows, we will present the algorithm for solving the mathematical model using the 1-Step Method. First we will apprroach the problem of non-linearities, and in the following sub-section the resulting ADI algorithm will be formulated.

Let us recall the governing equations :

the flow field equation

$$\psi_{xx} + \psi_{zz}\left[\frac{zh_x}{h}\right]^2 \text{[illegible]}_{xz}\frac{-2zh_x}{2} + \psi_z \frac{z}{h^2}(zh_x^2 - h_{xx}h) = 0 \qquad \text{(i)}$$

The equation resulting from balance of forces along the free surface

$$\frac{h_{xx}}{(1+h_x^2)^{3/2}} = - p_s + \frac{C}{2}\psi_z^2 \frac{1+h_x^2}{h^2} \qquad \text{(ii)}$$

Linearization of the Free Surface Equation

The simplified form of (ii) was analysed in the case of the Picard Method (Sec. 5.5.2). Simplification consisted of treating the right hand side as known function F(x). This was justified in the context of the Picard Method, but at present we will not follow this idea, since we treat (ii) as an equation of two variables: h and ψ .

In addition, we may notice an increase of nonlinearities, which are to be found in term F(x). In an attemt to linearize equation (ii) let us assume, that:

$$\tilde{h} = h + \delta h$$
$$\tilde{h}_x = h_x + \delta h_x$$
$$\tilde{h}_{xx} = h_{xx} + \delta h_{xx}$$
$$\tilde{\psi}_z = \psi_z + \delta\psi_z$$

where dashed are exact solutions, h, h_x, h_{xx} and ψ are approximations to the exact solutions with accuracy of δ terms (which are assumed to be small). Having term of the form $f + \delta f$ substituted into (ii), and expanding power expressions into power series, we get:

$$h_{xx} + \delta h_{xx} = A_0 + A_1(\delta h) + A_2(\delta h_x) + A_3(\delta\psi_z) + O(\delta^2) \qquad (5.6.1)$$

where: $$A_0 = -p_s(1 + h_x^2) + \frac{C}{2}\frac{(1+h_x^2)^{5/2}}{h^2}\psi_z^2$$

$$A_1 = -\frac{C(1+h_x^2)^{5/2}}{h^3}\psi_z$$

$$A_2 = C\psi_z^2 h_x(1 + h_x^2)^{3/2}\frac{5}{2h^2} - 3p_s h_x(1 + h_x^2)^{\frac{1}{2}}$$

$$A_3 = \frac{C(1+h_x^2)^{5/2}}{h^2}\psi_z$$

Linearization of the Field Equation:

As before, assume that we are given some approximation ψ of the exact solutions $\tilde{\psi}$ of (i), and that it is within small distance from $\tilde{\psi}$. Let us put the expression of the form $\tilde{\psi} = \psi + \delta\psi$ into (i):

$$\psi_{xx} + \delta\psi_{xx} + (\psi_{zz} + \delta\psi_{zz})\frac{z^2(h_x + \delta h_x)^2 + 1}{(h + \delta h)^2} +$$

$$(\psi_{xz} + \delta\psi_{xz})\frac{-2z(h_x + \delta h_x)}{h} +$$

$$(\psi_z + \delta\psi_z)[2(h_x + \delta h_x)^2 - (h_{xx} + \delta h_{xx})(h + \delta h)]\frac{2}{(h + \delta h)^2} = 0$$

Expanding power terms into series, and dropping all terms containing products of increments (products are assumed to be of negligible magnitude), we arrive to the following equation:

$$\psi_{xx} + B_1\psi_{zz} + B_2\psi_{xz} + B_3\psi_z + \delta\psi_{xx} + B_1\delta\psi_{zz} + B_2\delta\psi_{xz} + B_3\psi_z + B_4\delta h + B_5\delta h_x + B_6\delta h_{xx} + O(\delta^2) = 0 \tag{5.6.2}$$

where $B_1 = \dfrac{z^2h_x^2 + 1}{h^2}$

$$B_2 = \frac{-2zh_x}{h}$$

$$B_3 = \frac{2z}{h^2}\left[(h_x^2) - h_{xx}h\right]$$

$$B_4 = -2\psi_{zz}\frac{z^2h_x^2 + 1}{h^3} - \psi_{xz}\frac{2zh_x}{h^2} + \psi_z\ z(h_{xx}h - 4h_x^2)\frac{1}{h^3}$$

$$B_5 = 2\psi_{zz}\frac{z^2h_x}{h^2} - 2\psi_{xz}\frac{z}{h} + \psi_z\frac{4zh_x}{h^2}$$

$$B_6 = -\psi_z\frac{z}{h}$$

Linearized Iterative Scheme

Equations (5.6.1) and (5.6.2) were developed by the use of heuristic assumption: $\tilde{\psi} = \psi + \delta\psi$. Since we actually do not know $\tilde{\psi}$, nor $\delta\psi$, we put instead approximate relation: $\psi^{n+1} = \psi^n + \delta\psi$ and $h^{n+1} = h^n + \delta h$, which define the iterative algorithm for solving the original problem. Using this formulation, we may put (5.6.1) and (5.6.2) in the following way:

Flow field equation:

$$\begin{aligned}&\psi_{xx}^{N} + B_1^N\psi_{zz}^{N+1} + B_2^N\psi_{xz}^{N+1} + B_3^N\psi_{z}^{N+1} + \\ &B_4^N h^{N+1} + B_5^N h_x^{N+1} + B_6^N h_{xx}^{N+1} + \\ &-B_4^N h^{N+1} - B_5^N h_x^{N+1} - B_6^N h_x^{N+1} = 0\end{aligned} \tag{5.6.3}$$

Interface equation:

$$h_{xx}^{N+1} - A_1^N h^{N+1} - A_2^N h_x^{N+1} - A_3^N \psi_z^{N+1} = A_0^N - A_1^N h^N - A_2^N h_x^N - A_3^N \psi_z^N \tag{5.6.4}$$

This is a system of linear partial differential equations with respect to ψ and h, and their derivatives, N being the index of linearization iterations; A_i^N, B_i^N are defined as in (5.6.1) and (5.6.2) with their arguments taken from the N-th iterative step.

Now we will briefly refer to the effectiveness of this linearization scheme, and we will try to estimate the rate of its convergence. The concept of order of the convergence of the linearization scheme was presented earlier in Sec. 5.5.2, so let us only mention that it informs us about the rate of convergence of a particular scheme. By definition, c is an order of the iterative scheme, if:

$$\frac{[f^{N+1} - f^N]}{[f^N - f^{N-1}]^c} = O(1)$$

Since the linearization was arranged according to the Newton Raphson method, which is of accuracy of $O(\delta^2)$, we may expect c to be close to the value of 2. In the table below we show results for the scheme (5.6.3) and (5.6.4). The solution for each linearization step is assumed to be obtained by inversion of the matrix (in fact sequence of ADI iterations of the internal loop, as described in Sec. 5.1). The parameters of the exemplified problem are: $p_s = -1.5$, $C = 5$, computations were done on the grid of $r = 0.1$.

TABLE 5.14 Performance of the joint linearized iterative scheme.

	FIELD		INTERFACE
N	$\|\psi^{N+1}-\psi^{N}\|$	residuum	$\|h^{N+1}-h^{N}\|$
2	.012	.075	.048
3	.0022	.077	.012
4	2.1E-4	.002	2.2E-3
5	3.0E-6	.49E-3	9.0E-5
6	6.6E-9	.30E-10	2.4E-7
7	3.5E-14	.10E-12	1.5E-12

TABLE 5.15 : Order of convergence of the iterative scheme of linearization.

N	2	3	4	5	[illegible]
FIELD	1.39	1.38	1.5	1.48	1.64
RESID.	-	-	2.42	1.96	-
INTERF.	1.46	1.38	1.52	1.63	1.78

As we see, with advancing N, c approaches the value of 2. The best result is for residuum (on the flow field). This is an important feature, since in computational practice we want to bring the residual value to zero, that is to approach the exact numerical solution as closely as possible with the computed solution. Some c values are not shown, either due to computer accuracy, or large c's in the initial iterations. Here again, as in Sec. 5.5.2, we see very fast convergence. It takes 7 iterations of this highly nonlinear system to get down to a solution wihich is of order of machine accuracy.

5.6.2 Alternate Direction Implicite (ADI) solution procedure.

In the preceding paragraph we focussed on the effectiveness of the linearization. If we had a one-step solver for the system of linear equations generated by the finite-difference model of (i) and (ii), it would take only about 7 iterations on nonlinearities to achieve a solution. But we do not have this solver due to the dimensionality of the problem, and we have to use the approximate iterative solution (internal loop, see Sec. 5.1). This approximate solution scheme is provided in our case by the ADI algorithm.

In the example referred to in Tables 5.14, 5.15, it took up to 30 ADI iterations to increase the linear iteration index N by one. Overall performance was about 150 ADI/linearization iterations, which exceeds several times the number of iterations required by the Picard Method (28 iterations for the same parameters). This means that the scheme in which the ADI loop is entirely nested within the external linearization loop, is not economical and must be arranged so that the ADI and external loops overlap. This was found to be much more cost-effective.

For the sake of clarity and as a reference for the final algorithm, we show below the linearization and the nested ADI scheme setup.

Assume we have performed N loops of linearization and i - ADI steps in an attempt to estimate variables for the N+1 step. Therefore the next ADI step is defined as follows:

z - ADI sweep (5.6.5)

for the flow field:

$$-B_1^N\psi_{zz}^{i+\frac{1}{2}} - B_3^N\psi_z^{i+\frac{1}{2}} - B_4^N h^{i+\frac{1}{2}} =$$

$$\psi_{xx}^i + B_2^N\psi_{xz}^{i+\frac{1}{2}} + B_5^N(h_x^i - h_x^N) + B_6^N(h_{xx}^i - h_{xx}^N) - B_4^N h^N$$

for the interface :

$$A_1^N h^{i+\frac{1}{2}} + A_3^N \psi_z^{i+\frac{1}{2}} =$$

$$h_{xx}^i - A_2^N(h_x^i - h_x^N) - A_0^N + A_1^N \underline{h_1^N} + A_3^N \psi_z^N$$

x - ADI sweep

for the flow field:

$$\psi_{xx}^{i+1} = - B_1^N \psi_{zz}^{i+\frac{1}{2}} - B_3^N \psi_z^{i+\frac{1}{2}} - B_2^N \psi_{xz}^{i+1} - B_4^N h^{i+\frac{1}{2}}$$

$$-B_5^N(h_x^i - h_x^N) - B_6^N(h_{xx}^i - h_{xx}^N) + B_4^N h^N$$

for the interface:

$$h_{xx}^{i+1} - A_1^N h^{i+1} - A_2^N h_x^{i+1} =$$

$$A_0^N - A_1^N h^N - A_2^N h_x^N + A_3^N(\psi_z^{i+\frac{1}{2}} - \psi_z^N)$$

The above model is an ADI implementation of the scheme given by (5.6.3) and (5.6.4). Now, if we want to break down the internal nesting of ADI and make ADI / linearization loops overlap, we may accomplish this by changing in the scheme (5.6.5) index N into ADI index retarded by one full or half-step. The algorithm defined in this way; was found to be twice as fast compared to performance of (5.6.5), but about the same rate slower when referred to the Picard Method. Since it was expected that the 1-Step Algorithm should be superior to the Picard Method, we tried to modify the implementation to improve the rate of convergence. The actual shape of the ADI /

linearization scheme that we propose is as follows:

z - ADI sweep (5.6.6)

for the flow field:

$$-B_1^i\psi_{zz}^{i+\frac{1}{2}} - B_3^i\psi_z^{i+\frac{1}{2}} - B_4^i h^{i+\frac{1}{2}} =$$

$$\psi_{xx}^i + B_2^i\psi_{xz}^{i+\frac{1}{2}} - B_4^i h^i$$

for the interface:

$$A_3^i\psi_z^{i+\frac{1}{2}} + A_1^i h^{i+\frac{1}{2}} = -A_0^i + h_{xx}^i + A_3^i\psi_z^i - A_1^i h^i$$

x - ADI sweep

for the flow field:

$$\psi_{xx}^{i+1} = B_1^{i+\frac{1}{2}}\,\psi_{zz}^{i+\frac{1}{2}} + B_2^{i+\frac{1}{2}}\,\psi_{xz}^{i+\frac{1}{2}} + B_3^{i+\frac{1}{2}}\,\psi_z^{i+\frac{1}{2}}$$

for the interface:

$$h_{xx}^{i+1} - A_1^{i+\frac{1}{2}}\,h^{i+1} - A_2^{i+\frac{1}{2}}\,h_x^{i+1} =$$

$$- A_0^{i+\frac{1}{2}} + A_3^{i+\frac{1}{2}}\,(\psi_z^{i+\frac{1}{2}} - \psi_z^i)$$

If one compares the scheme (5.6.6) to the original linearized formulation (5.6.1) and (5.6.2), one may see, that in (5.6.6) we actually make use of these increments o, which are associated with dependent variables at the current ADI sweep, and of increments oψ_z in the free surface equation. All other increments are dropped. Therefore we may lose theoretical o accuracy at a single ADI half step, but we are close to maintaining this accuracy if complete ADI iteration is considered. The only increment which does not appear in (5.6.6) is δh_x.

In the table below we compare the rate of convergence of the Picard and 1-Step Algorithms:

TABLE 5.16 : Comparison of the rate of convergence (iterations number) for Picard and 1-Step Algorithms.

parameters: C p_s	18 -1.5	10 -1.5	100 1.5	500 1.5
Picard Algorithm	71	31	32	60
1-Step Algorithm	62	35	27	39

For the convex flow domain ($p_s > 0$), the 1-Step Algorithm is definitely faster. As for the negative p_s, this advantage is not so obvious. C = 18 is close to the extremal feasible value of C (for p_s = -1.5), hence the advantage of the 1-Step Method might be seen as a result of a better algebraical structure, but for smaller C (like C=10), we generally observe the Picard method yielding faster convergence. The question about possible further improvements of the 1-Step Algorithm is addressed in the following paragraph.

5.6.3 Relaxation strategy.

As was mentioned before, the scheme provided by (5.6.6) is the fastest ADI algorithm. If we think about the overall improvement of the algorithm, we do not think about the ADI, but rather about the rate of information transfer in frames of the general ADI / linearizatton algorithm. This is relaxation, i.e. damping or amplifying some factors. It was found in the Asymptotic Method study, that the flow field equation is converging fast, even for large p_s. Since we observe an essential slowdown for Direct Algorithms, we may attribute this to the presence of the interface equation in the the numerical model. In what follows we refer to the examination of two features: relaxation of the computed interface and damping / amplifying of the term $\delta\psi_z$ in the free surface equation. Results are exemplified for two cases - Case 1 ; $p_s=-1.5$, C=18, Case 2 : $p_s= 1.5$, C=500.

A) Relaxation of interface is of the form:

$$h_{new}^{i+\frac{1}{2}}=h^{i} + w(h^{i+\frac{1}{2}}- h^{i})$$

It was found, that effectiveness of w depends on p_s. For negative p_s it is desirable to have small overrelaxation, for positive - underrelaxation. For example, for Case 1 the best is w=1.1, yielding 57 iterations, which is 8% improvement compared to 62 (see Table 5.16). In Case2, for w=0.9 we have 5% improvement (37 iterations versus 39, when w=1.).

B) Speed factor. The stream function field affects free surface equation (5.6.1) by coefficients A_k and the term $A_3\delta\psi_z$. This is one of the memory terms in the iterative procedure, and it decreases to zero with the iterations advancing. It was observed, that multiplying this term by $o\psi_z$ may essentially change the convergence rate. For Case 1 there was an improvement by 40% (37 iterations vs 62), but for Case 2 it brought slowdown by 48%. Certainly this is connected with the magnitude of stream function gradient at the interface. ψ_z is much bigger in Case 1 ($\psi_z \leq 0.179$), than in Case 2 ($\psi_z \leq 0.059$), but in both cases , $\psi_z < 1$. This seems to suggest the application of underrelaxation, which could depend on p_s . Underrelaxation was arranged by multiplying the term $A_3\delta\psi_z$ by a relaxation constant, say θ. It was found, that for Case1, θ=0.1 brought about a 48% gain (32 iterations vs 62 when θ=1), and for Case2 θ=0.9 resulted in a slight improvement (37 vs 39).

The two relaxations A) and B), when applied jointly, yielded results very close to those of B). Therefore the 1-Step Algorithm, when relaxation is applied, yields definitely faster convergence, as shown on the table below:

TABLE 5.17 : Comparison of the rate of convergence for the Picard and 1-Step Algorithms with relaxation.

parameters: G	18	500
p_s	-1.5	1.5
Picard Algorithm	71	60
1-Step Algorithm	32	37
% improvement	41%	38%

All results presented in this paragraph come from computations on grid 10x10. The relaxation factors for interface if finer grids are used may be essentially different. Morover, it was found that if w=1, θ=1, we may have problems with stability on finer grids. They can be overcome by underrelaxation on the interface, as was checked. Therefore although it seems that relaxation of $\delta\psi_z$ is more effective computationally, the interface relaxation is decisive in terms of providing numerical stability of the algorithm. It is worth mentioning that this instability depends on a particular iterative scheme. For example in Case 1, r=0.05, the algorithm defined by (5.6.5) is stable, but (5.6.6) with w=1, θ=1 is unstable unless w is decreased to 0.5. This suggests that the gains in rate of convergence, as shown on Table 5.17 are to be paid for by appropriate relaxation of the interface equation to avoid numerical instability.

5.6.4 Differences in implementation of the Picard and 1-Step Algorithms.

Conceptual distinction and performance differences between these two methods were addressed earlier. Also imlementation for each of them was discussed in appropriate paragraphs. To close this section on Direct Methods, we want to expose differences in the implementation of these two algorithms. They are as follows:

A) - In the Picard Method, the interface equation is solved with respect to only one variable, i.e. free surface function h(x), the flow field equation with respect to 'function ψ. There is no difference between the two methods in the implementation of the x-ADI scheme. It comes into play in the z-ADI scheme, as the 1-Step Algorithm solves both equations with respect to two unknown functions h and ψ. This difference can be seen in formula (5.6.6). Here, for a single comlumn (fixed x_i), apart from the ψ variable we have present unknown $h(x_i)$ in the flow field eqation. This spoils the three - diagonal structure of the resulting system of algebraic equations. To make this system closed, we must have one equation attached to the system. This is the interface equation, in which $h(x_i)$ and $\psi_z(x_i)$ are unknowns. The algebraic structure of the system is therefore as follows:

$$\begin{bmatrix} b_1 & c_1 & \cdots & \cdots & \cdots & q_1 \\ a_2 & b_2 & c_2 & \cdots & \cdots & q_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & a_n & b_n & q_n \\ \cdots & \cdots & \cdots & e_n & a_n & b_{n+1} \end{bmatrix} \circ \begin{bmatrix} \psi_{i1} \\ \psi_{i2} \\ . \\ . \\ . \\ \psi \\ h_i \end{bmatrix} = \begin{bmatrix} d_1 - A_1\psi_{11} \\ d_2 \\ . \\ . \\ . \\ d_n - c_n\psi_{i,n+1} \\ d_{n+1} \end{bmatrix} \qquad (5.6.7)$$

where for i-th column (x=0 +ir) and line index k; i,k=1,...,n:

$$a_k = \frac{B_1(k,i)}{r^2} + \frac{B_3(k,i)}{2\,r}$$

$$b_k = -B_4(k,i) + 2\,B_6(k,i)/(r^2)$$

$$a_k = \frac{B_1(k,i)}{r^2} + \frac{-B_3(k,i)}{2\,r}$$

$$q_k = -B_4(k,i) + 2\,B_6(k,i)/r^2$$

$$d_k = \psi_{xx} + B_1(k,i)h_i - B_6(k,i)(2h_i)/r^2 + B_5(k,i)h_{xi} + B_2(k,i)\psi_{xz}$$

Coefficients B=B(k,i) as defined earlier in (5.6.2) can be here treated as functions of geometrical arguments defined on computational grid , thus B(k,i)=B(0.+kr , 0.+ir).

The last equation of the above system (5.6.7) is the algebraical formulation of the free surface equation, so in accordance with (5.6.6), we have:

$$e_{n+1} = -A_3(i)/2r$$

$$a_{n+1} = \frac{2A_3(i)}{r}$$

$$b_{n+1} = -A_1(i) - \frac{2}{r^2}$$

$$d_{n+1} = A_0(i) - A_1(i)h_i - A_3(i)\psi_{zi} + 1.5\frac{\psi_{n+1,i}}{r^2} - \frac{h_{i+1} + h_{i-1}}{r^2}$$

coefficients A=A(i) are defined by (5.6.1) and are exclusively functions of the column index i.

As was mentioned before, the resulting system of algebraic equations is no longer of a 3-diagonal structure, but this can be dealt with easily. It is more important that apart from solving the interface equation along with the X-ADI sweep, we solve it in in the z-ADI sweep too. This two-step free surface solving procedure can be looked at as a kind of Predictor (from z-ADI sweep) Corrector (x-ADI sweep) method. It was found, that if horizontal iteration on the interface (corrector) is removed from the algorithm, the overall iterative scheme provides a convergence rate that is several times slower. This comes from the fact that vertical solver of the interace (Predictor) is essentially the point-wise method, which is not quite appropriate for a boundary value problem. On the other hand, if we remove the Predictor, and use the Corrector, i.e. line-iterative scheme for h, the convergence rate is very much the same as that for the Picard Method. It is the joint application of Predictor - Corrector, which essentially accelerates convergence, at least for positive values of p_s.

B) - The second difference between these two methods comes from linearization. In the Picard Method the flow field equation is linear with respect to function ψ, hence it is $O(\delta)$ with respect to unknown h. In the interface equation we arrange $O(\delta^2)$ linearization of the curvature term only, so it is again linearization of global order $O(\delta)$. In the 1-Step Algorithm we linearize the flow field and interface equations with regard to ψ and h in the same instant. Although in the final form we do not make use of all incremental expressions in x and z sweeps of ADI, we have the increment $\delta\psi_z$, which when appropriately relaxed, may cause essential acceleration of convergence. This is an important advantage of the 1-Step Method over the Picard Method. It may happen, however, that when C is close to its extremal feasible value (with resp. to p_s), we may encounter stablility problems. As we reported, these can be dealt with by suitable relaxation of interface solution h(x).

5.6.5 Accuracy of discretization.

In this chapter we have been concerned hitherto with several aspects of implementation which would yield an effective numerical algorithm for the solution of the ideal fluid flow. When discussing the Picard and 1-Step Methods we were interested in the rate of convergence of the iterative scheme. This was found to depend on the linearization and ADI-relaxation. Another problem was determination of the range of feasible parameters p_s and C. We addressed also the question of the existence of a solution to the free surface equation of our model problem. All these points were dealing with the availability of a numerical solution. Now we put forth the question of the quality of numerical results. As was discussed in Seq. 3.3, the efficient tool to pursue this question is the analysis of the grid-convergence of the error of discretization. This analysis was applied already in the case of the Asymptotoic Method. We estimated the order of convergence of the ψ field with the interface being assumed as known (static interface). Now, when we deal with Direct Methods, we are interested in the estimation of the order of convergence for both - interface and flow field. This is a much different situation when compared to the Asymptotic case, since equations for both variables are non-linear due to stream function -

free surface coupling. This difference, coming from nonlinearities, raises again the question of the validity of the convergence order estimation based on results obtained on a sequence of 3 different grids (Sec. 3.3). Therefore to make sure that the 3-grid estimator is of value for Direct Methods, we will start from order estimation for the mathematical model with the known analytical solution, similar in structure to our system describing ideal fluid flow. In a further subsection we will examine results for the original system of equations.

Analysis of model with known analytical solution.

Let us consider the following function:

$$G(x,y) = \frac{1}{\pi} \sin(\pi x)(1 + \frac{y}{h})^3 \qquad (i)$$

$$H(x) = H_1(x) + H_2(x) \qquad (ii)$$

where: $H_1(x) = 1 - s(\frac{1}{p_s^2} - 0.25)^{\frac{1}{2}} + s\,[\frac{1}{p_s^2} - (x-0.5)^2]^{\frac{1}{2}}$, $s = \mathrm{sgn}(p_s/|p_s|)$

$H_2(x) = 0.05\sin(2\pi \cdot x)$

Function $H_1(x)$ is a segment of a circle of radius $1/|p_s|$, $x \in [0,1]$, which is the counterpart of the static interface in the original model. Function H_2 (x) is a small harmonic disturbance superimposed on H_1.

Let us formulate the problem:

Find function $\psi(x,y)$ and $h(x)$, so that:

$$\nabla^2\psi = \nabla^2 G -$$

$$\frac{h_{xx}}{(1 + h_x^2)^{3/2}} - \frac{C}{2}\left[\left(\frac{\partial\psi}{\partial x}\right)^2 + \left(\frac{\partial\psi}{\partial y}\right)^2\right] = \frac{H_{xx}}{(1 + H_x^2)^{3/2}} \qquad (5.6.8)$$

hold for $x \in [0,1]$, $y \in [1,h(x)]$, and:

$\psi = 0$ at $x = 0$

$\psi = 0$ at $x = 1$

$\psi = \cos(\pi x)$ at $y = 0$

$\psi = 0$ at $y = h(x)$

$h = 1$ at $x = 0$

$h = 1$ at $x = 1$

Let us note that the above problem is defined in the same terms as the original physical problem. We have the elliptic equation for the field (though it is not Laplace, but Poisson-type), ordinary differential equation for the free boudary $h(x)$, and boundary conditions, which are precisely the same, as in the formulation of the model an for ideal fluid flow. The above problem has an analytical solution, which is (i) and (ii).

Let us remind ourselves, that α is an order of grid-convergence if (Sec. 3.3):

$$\frac{[f^{k_1} - \tilde{f}](Gn)}{[f^{k_2} - \tilde{f}](Gn)} \propto \left|\frac{r_{k_1}}{r_{k_2}}\right|^{\alpha} \qquad (iii)$$

where $\tilde{f}$ is an exact analytical solution (ψ or h), $[\![.]\!](Gn)$ is L-2 norm defined on a subgrid Gn of grids Gk_1 and Gk_2, r_k is spatial increment of grid of dimension k x k. Let us discuss Tables 5.18, 5.19, 5.20. Each table consists of two parts. The first one lists $[\![f^k - \tilde{f}]\!](G10)$, the second shows the respective values of α, which is the order of convergence of numerical solution to the analytical one with the grid refinement. Three table cover three cases: 'static free surface' H_1 being convex (p_s =1.5), flat (p_s=0.) and concave (p_s=-1.5). All 3 cases show α=2, as expected, beginning from results computed on grid 10x10.

Tables 5.21, 5.22, 5.23 show results obtained from the same computed fields, but using reference estimation of the grid-convergence order. That is, assuming $\tilde{f}$ is unavailable, we approximate (iii) by (iv):

$$\frac{[\![f^{k_1} - f^{k_2}]\!](Gk)}{[\![f^{k_2} - f^{k_3}]\!](Gk)} \propto \frac{|r_k^{\alpha} - r_{k_2}^{\alpha}|}{|r_{k_2}^{\alpha} - r_{k_3}^{\alpha}|} \qquad (iv)$$

These three tables show again results very close to α=2. This means that for the type of problem we consider here, formula (iii) can be approximated with (iv) starting from grid 10x10.

Analysis of model with unknown analytical solution.

Let us return to the numerical model of the ideal fluid flow. Order estimation is presented for the same stagnation pressure values as tested above: $p_s=1.5$, $p_s=0.$, $p_s=-1.5$. Results are displayed on Tables 5.24, 5.25, 5.26. Each specified case is identified by a couple of parameters: p_s and C. It is worth mentioning though, that for the problem (5.6.6) the value of C is of small importance. It is due to the fact, that $(\partial\psi/\partial x)^2+(\partial\psi/\partial y)^2$ goes to zero as the grid is made finer. Therefore the coupling of (i) and (ii) is dominated by the surface → interface influence, while the field affects equation (ii) by the relatively small and in limit zero-valued term $C((\partial\psi/\partial x)^2+(\partial\psi/\partial x)^2)/2$. It is not the same in our original problem, where velocity does not vanish on the surface. Therefore C is of importance as a factor amplifying or damping the influence of velocity on the shape of the interface. Large value of $C=\rho V^2L/T$ (see Sec. 2.2) in physical interpretation means a small surface tension. Evidently, this makes free surface more sensitive to the distribution of velocity along the free boundary. Conversely, small C means strong surface tension and in consequence the interface is expected to be less dependent on the motion pattern along the free surface.

In Sec. 5.5.3 we have addressed the problem of bounds on the value of C for fixed p_s. These bounds were established on grounds of the existence condition for interface equation. Now, we can put the question, whether the order of convergence is sensitive to the value of capillary number C.

Let us consider case with $p_s=1.5$. It was found, that the range of feasible C, when $p_s=1.5$, is $0 \leq C \leq 600$. On Table 5.24 we have listed values of α for the flow field function ψ its derivatives and for free surface function along with its first derivative and curvature of the interface. Curvature is pointed out as a factor directly dependant on the balance of forces at the free surface. We see that with grid refinement, we get better results for ψ, h and curvature. As for derivatives of field function ψ, we are apt to think that the results are not satisfactory. This is comment on the case with p_s =1.5 and C is=100; in this C is far from its limit of feasiblity (which is 600). Now, let us look at Tables 5.25 and 5.26 which contain results for p_s=1.5 as before, but C=200 and C=400, resp. We can see gradual deterioration of order α, in particular for ψ, h, h_x and curvature. It means that to provide results which would show the second order of grid-convergence, we should either perform computations on finer grids than those listed in the tables, or restrict the model to smaller values of C. Let us mention by the way, that if C=100 (p_s =1.5), the asymptotic parameter ϵ is 0.11, i.e. only twice as big as ϵ for which the Asymptotic Method can be applied. This comparison may suggest that to obtain valuable results (α close to 2) for larger C ($\epsilon > 0.05$), we may use the Direct Method, but the choice of the computational grid is not a trivial problem.

Deterioration of the order of convergence for convex domains was already observed in the case of the Asymptotic Method. It was suggested, that this poor behaviour may be the result of the transformation of variables. The physical grid as a counter image of the transformation becomes sparser close to free surface, if p_s>0.

Therefore Lagrange residuals in difference approximations may by more difficult to stabilize. If they are multiplied by large C, they may contribute even more to the deterioration of the order of grid-convergence unless we use very fine grids.

These were comments on the case of $p_s=1.5$. When we look at Table 5.27, $p_s= 0.$, C=100 ($\epsilon=0.47$), and table 5.28, where $p_s=-1.5$ and $\epsilon=10$ ($\epsilon=0.133$), we see that order estimators converge to the desired value of 2. What is more, ϵ asymptotic parameters are larger compared to the convex case, and C are closer to the maximum feasible values. Therefore the order of grid-convergence depends on the parameters of the problem. It was shown that for convex physical domains it may require much denser grids to obtain the desired order of grid-convergence, than for concave ones.

TABLE 5.18 Distance $[\![f^k - \tilde{f}]\!]$(G10) for the problem (5.6.8) and estimates of the grid-convergence order, obtained through comparison of the computed values and the exact solution.

Case p_s=0, C=100.

	ϕ	ϕ_x	ϕ_z	ϕ_{xz}	ϕ_{xx}	ϕ_{zz}	h	h_x	h_{xx}
$[\![f^{10} - \tilde{f}]\!]$	.257E-04	.349E-03	.286E-03	.122E-02	.664E-03	.606E-03	.387E-03	.203E-02	.145E-02
$[\![f^{20} - \tilde{f}]\!]$	.627E-05	.878E-04	.719E-04	.301E-03	.167E-03	.148E-03	.950E-04	.503E-03	.348E-03
$[\![f^{30} - \tilde{f}]\!]$	.277E-05	.391E-04	.320E-04	.134E-03	.742E-04	.655E-04	.421E-04	.223E-03	.154E-03
$[\![f^{40} - \tilde{f}]\!]$	.156E-05	.220E-04	.180E-04	.751E-04	.417E-04	.368E-04	.237E-04	.126E-03	.866E-04
$[\![f^{60} - \tilde{f}]\!]$	.691E-06	.978E-05	.800E-05	.334E-04	.186E-04	.163E-04	.105E-04	.558E-04	.384E-04
$f^{10}: f^{20}: \tilde{f}$	2.04	1.99	1.99	2.01	1.99	2.03	2.03	2.01	2.06
$f^{20}: f^{30}: \tilde{f}$	2.01	2.00	2.00	2.00	2.00	2.01	2.01	2.00	2.01
$f^{30}: f^{40}: \tilde{f}$	2.01	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
$f^{40}: f^{60}: \tilde{f}$	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00

TABLE 5.19 Distance $\| f^k - \tilde{f} \|(G10)$ for the problem (5.6.8) and estimates of the grid-convergence order obtained through comparison of the computed values and the exact solution.

Case p_s=-1.5, C=10.

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$\| f^{10} - \tilde{f} \|$	.133E-04	.413E-03	.269E-03	.101E-02	.905E-03	.213E-03	.863E-03	.246E-02	.469E-02
$\| f^{20} - \tilde{f} \|$	.348E-05	.104E-03	.675E-04	.253E-03	.227E-03	.534E-04	.217E-03	.601E-03	.112E-02
$\| f^{30} - \tilde{f} \|$	.156E-05	.460E-04	.300E-04	.113E-03	.101E-03	.238E-04	.965E-04	.266E-03	.493E-03
$\| f^{40} - \tilde{f} \|$	.878E-06	.259E-04	.169E-04	.635E-04	.567E-04	.134E-04	.543E-04	.149E-03	.276E-03
$\| f^{60} - \tilde{f} \|$	.391E-06	.115E-04	.751E-05	.282E-04	.252E-04	.596E-05	.241E-04	.663E-04	.123E-03
$f^{10}; f^{20}: \tilde{f}$	1.94	2.00	2.00	1.99	2.00	1.99	1.99	2.03	2.07
$f^{20}: f^{30}: \tilde{f}$	1.98	2.00	2.00	2.00	2.00	1.99	2.00	2.01	2.02
$f^{30}: f^{40}: \tilde{f}$	1.99	2.00	2.00	2.00	2.00	2.00	2.00	2.01	2.01
$f^{40}: f^{60}: \tilde{f}$	1.99	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.01

TABLE 5.20 Distance $\|f^{k} - \tilde{f}\|(G10)$, for the problem (5.6.8) and estimates of the grid-convergence order obtained through comparison of the computed values and the exact solution.

Case p_s=1.5, C=400.

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$\|f^{10} - \tilde{f}\|$	.398E-04	.318E-03	.307E-03	.131E-02	.642E-03	.982E-03	.107E-02	.229E-02	.426E-02
$\|f^{20} - \tilde{f}\|$	.953E-05	.811E-04	.768E-04	.319E-03	.163E-03	.234E-03	.230E-03	.589E-03	.105E-02
$\|f^{30} - \tilde{f}\|$	.420E-05	.362E-04	.341E-04	.141E-03	.730E-04	.103E-03	.991E-04	.263E-03	.478E-03
$\|f^{40} - \tilde{f}\|$	.236E-05	.204E-04	.192E-04	.793E-04	.411E-04	.579E-04	.551E-04	.149E-03	.272E-03
$\|f^{60} - \tilde{f}\|$	.105E-05	.906E-05	.853E-05	.352E-04	.183E-04	.257E-04	.243E-04	.662E-04	.122E-03
$f^{10}: f^{20}: \tilde{f}$	2.06	1.97	2.00	2.04	1.97	2.07	2.21	1.96	2.01
$f^{20}: f^{30}: \tilde{f}$	2.02	1.99	2.00	2.01	1.99	2,02	2.08	1.98	1.94
$f^{30}: f^{40}: \tilde{f}$	2.01	2.00	2.00	2.01	1.99	2.01	2.04	1.99	1.97
$f^{40}: f^{60}: \tilde{f}$	2.00	2.00	2.00	2.00	2.00	2.01	2.02	2.00	1.98

TABLE 5.21 Estimates of the grid-convergence order for the problem (5.6.8) obtained through comparison of results computed on grids Gl, Gl, Gm; Case p_s=0, C=100.

	ϕ	ϕ_x	ϕ_z	ϕ_{xz}	ϕ_{xx}	ϕ_{zz}	h	h_x	h_{xx}
$\epsilon^{10}-\epsilon^{20}-\epsilon^{40}$	.200E+01	.195E+01	.196E+01	.200E+01	.195E+01	.200E+01	.200E+01	.200E+01	.205E+01
$\epsilon^{20}-\epsilon^{40}-\epsilon^{60}$	.200E+01	.195E+01	.195E+01	.200E+01	.195E+01	.200E+01	.200E+01	.200E+01	.200E+01
$\epsilon^{30}-\epsilon^{40}-\epsilon^{60}$	.200E+01	.195E+01	.196E+01	.200E+01	.195E+01	.200E+01	.200E+01	.200E+01	.200E+01

TABLE 5.22 Estimates of the grid-convergence order for the problem (5.6.8) obtained through comparison of results computed on grids Gl, Gl, Gm; Case p_s=-1.5, C=10.

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
f^{10}-f^{20}-f^{40}	.190E+01	.195E+01	.195E+01	.195E+01	.195E+01	.195E+01	.195E+01	.200E+01	.205E+01
f^{20}-f^{40}-f^{60}	.195E+01	.195E+01	.195E+01	.195E+01	.195E+01	.195E+01	.195E+01	.200E+01	.200E+01
f^{30}-f^{40}-f^{60}	.195E+01	.195E+01	.195E+01	.195E+01	.195E+01	.195E+01	.195E+01	.200E+01	.200E+01

TABLE 5.23 Estimates of the grid-convergence order for the problem (5.6.8) obtained through comparison of results computed on grids Gl, Gl, Gm; Case p_s=1.5, C=400.

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.205E+01	.196E+01	.196E+01	.200E+01	.196E+01	.205E+01	.225E+01	.195E+01	.205E+01
$f^{20}-f^{40}-f^{60}$	.200E+01	.195E+01	.196E+01	.200E+01	.196E+01	.200E+01	.205E+01	.195E+01	.190E+01
$f^{30}-f^{40}-f^{60}$	.200E+01	.195E+01	.196E+01	.200E+01	.195E+01	.200E+01	.205E+01	.196E+01	.196E+01

TABLE 5.24 Estimates of the grid-convergence order for the problem (3.2.2) -(3.2.3) obtained through comparison of results computed on grids Gl, Gl, Gm;

Case p_s =1.5, C=100.

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	curv.
$f^{10}-f^{20}-f^{40}$	.185E+01	.195E+01	.195E+01	.185E+01	.185E+01	.160E+01	.135E+01	.230E+01	.365E+01
$f^{20}-f^{40}-f^{60}$	.190E+01	.190E+01	.195E+01	.170E+01	.160E+01	.135E+01	.165E+01	.210E+01	.210E+01
$f^{30}-f^{40}-f^{60}$	.190E+01	.190E+01	.195E+01	.170E+01	.155E+01	.135E+01	.170E+01	.205E+01	.205E+01

TABLE 5.25 Estimates of the grid-convergence order for the problem (3.2.2) -(3.2.3) obtained through ~~results~~ results computed on grids Gl, Gl, Gm;

Case p_s=1.5, C=200.

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	curv.
$f^{10}-f^{20}-f^{40}$	.175E+01	.195E+01	.196E+01	.180E+01	.190E+01	.165E+01	.150E+01	.250E+01	.320E+01
$f^{20}-f^{40}-f^{60}$	.185E+01	.195E+01	.196E+01	.165E+01	.170E+01	.140E+01	.165E+01	.230E+01	.155E+01
$f^{30}-f^{40}-f^{60}$	.185E+01	.195E+01	.196E+01	.165E+01	.165E+01	.140E+01	.105E+01	.220E+01	.220E+01

TABLE 5.26 Estimates of the grid-convergence order for the problem (3.2.2) -(3.2.3) obtained through comparison of results computed on grids Gl, Gl, Gm;

Case p_s =1.5, C=400.

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	curv.
$f^{10}-f^{20}-f^{40}$	.135E+01	.205E+01	.200E+01	.175E+01	.230E+01	.205E+01	.280E+01	.255E+01	.250E+01
$f^{20}-f^{40}-f^{60}$	.165E+01	.200E+01	.200E+01	.160E+01	.200E+01	.170E+01	.305E+01	.245E+01	.300E+01
$f^{30}-f^{40}-f^{60}$	.170E+01	.200E+01	.200E+01	.165E+01	.195E+01	.165E+01	.310E+01	.240E+01	.295E+01

TABLE 5.27 Estimates of the grid-convergence order for the problem (3.2.2) -(3.2.3) obtained through comparison of results computed on grids Gl, Gl, Gm;

Case p_s=0, C=100.

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	curv.
$f^{10}-f^{20}-f^{40}$	.195E+01	.195E+01	.195E+01	.195E+01	.195E+01	.195E+01	.200E+01	.195E+01	.180E+01
$f^{20}-f^{40}-f^{60}$	.200E+01	.195E+01	.195E+01	.200E+01	.195E+01	.200E+01	.190E+01	.190E+01	.180E+01
$f^{30}-f^{40}-f^{60}$	.200E+01	.195E+01	.195E+01	.200E+01	.195E+01	.200E+01	.190E+01	.190E+01	.185E+01

TABLE 5.28 Estimates of the grid-convergence order for the problem (3.2.2) -(3.2.3) obtained through comparison of results computed on grids Gl, Gl, Gm;

Case p_s=-1.5, C=10.

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	curv.
$f^{10}-f^{20}-f^{40}$	.200E+01	.200E+01	.195E+01	.175E+01	.190E+01	.185E+01	.200E+01	.205E+01	.140E+01
$f^{20}-f^{40}-f^{60}$	.200E+01	.200E+01	.195E+01	.185E+01	.195E+01	.190E+01	.200E+01	.200E+01	.165E+01
$f^{30}-f^{40}-f^{60}$	.200E+01	.200E+01	.195E+01	.190E+01	.195E+01	.190E+01	.200E+01	.200E+01	.170E+01

In solving the analytical model using a finite-difference simulation, we always wonder how distant the actual numerical solution is from the exact analytical one. This distance is decisive in terms of the validity of the algorithm. Since the distance cannot be given explicitly, we may try to investigate the character of convergence of numerical results to the exact solution with $r \rightarrow 0$. When we use second order finite difference approximations, we have grounds to expect that starting from some small-r, the numerical solutions form a sequence which is convergent in accordance with formula (iv) above. Here we arrive at the concept of extrapolation. We gave the background for this idea in Sec. 3.3. In Table 5.29 we show distances of extrapolated values from the exact solution (problem 5.6.8). Extrapolation was performed for two cases: p_s=-1.5, C=10 and p_s=1.5 and C=400.

Let us consider case p_s=1.5, C=400. In the first column we have distances of the exact solution and solutions obtained by extrapolation of variables computed on grids 10x10 and 20x20. If we compare the distance of the exact solution from that computed on grid G60 (Table 5.20) we can see that extrapolated values are closer to the exact solution than those from grid G60. This is an important observation, since it suggests that we may considerably decrease the time of computations, by running the algorithm on coarser grids and then extrapolationg results. This operation is justified, however, only if there is confirmed information on the order of grid-convergence, and grids involved in extrapolation fall within the range of dimension required to prove this order.

When speaking about extrapolation as a cost-saving vehicle, let us mention that the time-gain resulting from extrapolation in place of computing on fine grids is considerable. For the example formulated by (5.6.8):

a) total time required to get solution on grid G10 is 1.62 secs.. (24 iter.)

b) total time required to get solution on grid G20 is 10.8 secs.. (36 iter.)

Joint time cost of obtaining extrapolated results : about 13 secs..

c) total time required to get solution on grid G60 is 320 secs. (320 iter.)

On top of the cost saving, comes higher accuracy of extrapolated values, as we mentioned above. This can be seen in the two following columns of Table 5.29 showing the distance of the exact solution from values extrapolated from finer grids. The results are far superior to those obtained on grid G60.

The same conclusions may be applied to the second case, i.e. p_s=-1.5, C=10, which is presented in the lower part of Table 5.29.

Table 5.29 with resulting comments and conclusions apply directly to the simplified case (5.6.8), when exact solution to the problem is available. It may be shown, however, that it is possible to have some information about the distance of numerical solution from the analytical one, even though the latter one is not available. We will stay for a while with the example (5.6.8). Let us suppose we have done computations on grids G10,G20,...G60. Let us fix as a reference f^{60} which are values computed on grid G60. Now, let $E_{k,l}(f)$ be an extrapolation of values of f computed on grids Gk, and Gl. If we take a sequence

$$[E_{10,20}(f) - f^{60}]$$
$$[E_{20,40}(f) - f^{60}]$$
$$[E_{30,60}(f) - f^{60}]$$

we may observe (Table 5.30), that it is fast converging to some constant. This limit constant may be understood as an estimate of the distance of f^{60} from the unknown analytical solution. Table 5.30 lists this sequence for several variables. This table is referred to problem (5.6.6), therefore we may examine validity of the statement formulated above by comparison to the exact solution. Although each of the sequences is short, they show rapid stabilization. What is more, when we consider the distance of f^{60} from the exact solution, we see that this is very close to $[E_{30,60}(f) - f^{60}]$. The value of this norm is therefore a good estimate of how far from the exact solution is the result f^{60}.

The same effect can be observed in Tables 5.31a,b,c, which are referred to our original model of an idal fluid flow. Here, as for problem (5.6.6), we observe fast stabilization of the sequence for each variable. Assuming $E_{30,60}(f)$ is close to the ultimate solution, we may treat $[E_{30,60}(f) - f^{60}]$ as an estimation of the distance of f^{60} from the exact unknown solution. Detailed listing of the $[E_{k,l}(f) - f^{60}]$ is presented in Tables 5.31.a,b,c. For the sake of completeness we show data not only for ψ and h, but also for their derivatives and curvature of the free surface. Table 5.31.a is referred to the case p_s=1.5, C=100, Table 5.32.b to the case p_s=0, C=100 and Table 5.31.c to the case p_s =-1.5, C=10. Other cases discussed in preceding section, like C=200, or C=400, are not taken here into account, since within the range of grids used in this work, the grid-converence is not of the desired order. Therefore we can not arrange extrapolation for these cases.

TABLE 5.29 Distance of extrapolated values from the exact solution $[E_{k,1}(f) - \tilde{f}]$ for p_s=1.5, C=400; p_s=-1.5, C=10 (example problem (5.6.8)

variables	$[E_{10,20}(f)-\tilde{f}]$	$[E_{20,40}(f)-\tilde{f}]$	$[E_{40,60}(f)-\tilde{f}]$
p_S= 1.5, C = 400			
ψ	.653E-6	.414E-7	.821E-8
ψ_x	.382E-5	.275E-6	.570E-7
ψ_z	.367E-5	.228E-6	.456E-7
ψ_{xz}	.139E-4	.106E-5	.233E-6
ψ_{xx}	.189E-4	.138E-5	.332E-6
ψ_{zz}	.163E-4	.118E-5	.268E-6
h	.543E-4	.364E-5	.731E-6
h_x	.138E-3	.877E-5	.174E-5
h_{xx}	.834E-3	.526E-4	.105E-4
p = -1.5, C = 10			
ψ	.374E-6	.240E-7	.477E-8
ψ_x	.182E-5	.124E-6	.252E-7
ψ_z	.130E-5	.823E-7	.163E-7
ψ_{xz}	.831E-5	.682E-6	.148E-6
ψ_{xx}	.100E-4	.788E-6	.168E-6
ψ_{zz}	.521E-5	.361E-6	.760E-7
h	.598E-5	.366E-6	.722E-7
h_x	.327E-4	.204E-5	.404E-6
h_{xx}	.113E-3	.665E-5	.130E-5

TABLE 5.30 Distance of extrapolated values from the solution computed on grid G60: $[E_{k,1}(f) - f^{60}]$; (Cases p_s=1.5, C=400; p_s=-1.5, C=10, example prob. (5.6.8)).

p_s= -1.5, C = 10	$[E_{10,20}(f)-f^{60}]$	$[E_{20,40}(f)-f^{60}]$	$[E_{40,60}(f)-f^{60}$
variables			
ψ	.362E-6	.379E-6	.389E-6
ψ_x	.114E-4	.115E-4	.115E-4
ψ_z	.733E-5	.749E-5	.750E-5
ψ_{xz}	.275E-4	.281E-4	.282E-4
ψ_{xx}	.262E-4	.252E-4	.252E-4
ψ_{zz}	.748E-5	.591E-5	.595E-5
h	.230E-4	.241E-4	.241E-4
h_x	.107E-3	.825E-4	.813E-4
h_{xx}	.248E-3	.155E-3	.151E-3
p_s= 1.5, C = 400			
ψ	.163E-5	.108E-5	.105E-5
ψ_x	.779E-5	.892E-5	.903E-5
ψ_z	.914E-5	.953E-5	.853E-5
ψ_{xz}	.472E-4	.359E-4	.353E-4
ψ_{xx}	.223E-4	.180E-4	.182E-4
ψ_{zz}	.408E-4	.266E-4	.259E-4
h	.762E-4	.275E-4	.249E-4
h_x	.138E-3	.787E-4	.804E-4
h_{xx}	.793E-3	.140E-3	.145E-3

TABLE 5.31.a Distance of extrapolated values from the solution computed on grid G60: $[E_{k,l}(f) - f^{60}]$; (Cases p_s=1.5, C=100; example prob. (3.2.2) - (3.2.3)).

variables	$[E_{10,20}(f)-f^{60}]$	$[E_{20,40}(f)-f^{60}]$	$[E_{30,60}(f)-f^{60}]$
ψ	.433E-6	.111E-5	.119E-5
ψ_x	.766E-5	.623E-5	.657E-5
ψ_z	.174E-4	.194E-4	.195E-4
ψ_{xz}	.976E-4	.244E-4	.291E-4
ψ_{xx}	.590E-4	.171E-4	.236E-4
ψ_{zz}	.100E-3	.192E-4	.307E-4
h	.226E-4	.685E-5	.111E-4
h_x	.191E-3	.629E-4	.538E-4
Curv.	.413E-3	.631E-4	.311E-4

TABLE 5.31.b Distance of extrapolated values from the solution computed on grid G60: $[E_{k,l}(f) - f^{60}]$; (Cases p_s=0, C=100; example prob. (3.2.2) - (3.2.3)).

variables	$[E_{10,20}(f)-f^{60}]$	$[E_{20,40}(f)-f^{60}]$	$[E_{30,60}(f)-f^{60}]$
ψ	.138E-5	.151E-5	.151E-5
ψ_x	.985E-5	.987E-5	.991E-5
ψ_z	.105E-4	.114E-4	.115E-4
ψ_{xz}	.135E-4	.111E-4	.111E-4
ψ_{xx}	.143E-4	.142E-4	.146E-4
ψ_{zz}	.944E-5	.950E-5	.928E-5
h	.698E-5	.578E-5	.624E-5
h_x	.305E-4	.342E-4	.369E-4
Curv.	.158E-3	.101E-3	.122E-3

TABLE 5.31.c Distance of extrapolated values from the solution computed on grid G60: $[\![E_{k,l}(f) - f^{60}]\!]$; (Cases p_s=-1.5, C=10; example prob. (3.2.2) - (3.2.3)).

variables	$[\![E_{10,20}(f)-f^{60}]\!]$	$[\![E_{20,40}(f)-f^{60}]\!]$	$[\![E_{30,60}(f)-f^{60}]\!]$
ψ	.172E-5	.166E-5	.165E-5
ψ_x	.205E-4	.196E-4	.196E-4
ψ_z	.719E-5	.710E-5	.711E-5
ψ_{xz}	.586E-4	.235E-4	.264E-4
ψ_{xx}	.532E-4	.326E-4	.339E-4
ψ_{zz}	.254E-4	.125E-4	.136E-4
h	.186E-4	.152E-4	.150E-4
h_x	.152E-3	.966E-4	.944E-4
Curv.	.142E-3	.475E-4	.717E-4

CHAPTER 6

HIGHER ORDER METHODS

The flow of an ideal fluid, as defined mathematically in Sec. 2.1 is determined by the flow field and the shape of the free surface. Since the flow pattern is coupled with geometry of the free surface, the accuracy of the numerical solution is of primary importance. High accuracy is desirable in particular in the vicinity of the free surface. In finite difference discretization, accuracy of the numerical solution depends primarily on the accuracy of the differencing scheme, i.e. on the magnitude of the truncation error of discretization. For the fixed grid-size parameters, the truncation error depends on the order of Lagrange residuals resulting from the Taylor-series approximations. In Chapter 3 we have discussed discretization of the mathematical model of the flow and have formulated discretized models of the second and higher order of accuracy. In Chapters 4-5 we have discussed algorithms implemented algebraically by discretization of the 2-nd order. In this chapter we shall discuss the higher order methods. First we will point out differences in implementation between second and higher order methods. Next, having a particular differencing scheme, truncation error and the rate of its convergence to zero will be examined. Finally we will compare 2-nd and higher order methods in terms of their cost-efficiency.

All computations using higher order differencing schemes have been carried out using the Picard Algorithm (see Sec.5.5).

6.1 Description of Implementation.

There are three major differences in implementation of the higher order discretization schemes when compared to the 2-nd order methods. These are:

(a) Algebraic ADI procedure,

(b) implementation of the boundary conditions,

(c) iterative strategy and selection of relaxation coefficients.

We shall discuss each of these issues separately.

(a): In Sec. 3.2 we have defined the algebraical model of the analytical problem as:

$$\begin{aligned} \mathcal{N}_{ij} &= 0 \\ H^{1}_{ij} &= 0 \\ &\vdots \\ H^{k-1}_{ij} &= 0 \\ i &= 1,\dots,n \\ j &= 1,\dots,m \end{aligned}$$

where $\mathcal{N}_{ij}$ is the governing equation specified in a pivotal point (x_i,z_j) of a grid and H^{1}_{ij} are independent Hermitian formulae. It was shown in Sec. 3.2 that for second order discretization we may arrange separation of a modified governing equation from the associated

Hermitian formulae (see (3.14)). In consequence; systems of algebraic equations defining x and z ADI sweeps are in a scalar 3-diagonal form. In the case of higher order Hermitian formulae, we are unable to arrange the separation, and the ADI procedure is defined in terms of systems of algebraic equations of 3x3 block tri-diagonal structure. Therefore the dimensionality of the systems resulting from higher order discretization is much higher, and this is the first disadvantage of the method.

(b) Higher order Hermitian formulae (like (A.1) and (A.2) in Appendix) provide an implicit relation between the values of the function and its derivatives at 3 consecutive points of a grid molecule. Therefore, when setting up a system of algebraic equations, we have to consider the first and second partial derivatives on the boundary as unknowns. The required boundary relations will be expressed by the Pade difference formula (A.3 in Appendix), which is of the order 3. This in turn may cause a reduction in the overall accuracy of the algorithm. Here we are at a point of particular importance, since high accuracy at the boundary is of major importance in numerical simulation of flows with free surface. There also used to be a problem with accuracy due to mixed derivatives difference estimators. This was, however, successfully solved in this work by developing planar estimates of the 4-th order of accuracy on the compact molecule in the interior of the domain as well as at the boundary (see appropriate Hermitian formulae A.4, A.5 and A.6 in

Appendix).

(c) ADI is driven by pseudo-time derivative, or equivalently, by relaxation. It has been shown in Sec. 4.2 (Figs. 4.2 and 4.3) that in the case of the 2-nd order method, application of Wachspress optimization yielded a considerable acceleration of the iterative process. Unfortunately, this optimization has been developed only for the separated systems, that is for a case when resulting algebraic systems are of a scalar 3-diagonal structure. For higher order methods we can apply only a single relaxation factor, and this makes the iterative process considerably longer. In addition, we have to rely on a trial & error procedure in order to determine the optimal value of the relaxation factor.

All three factors listed above, contribute to the worsening of the performance of the higher order algorithms. Below we provide a quantitative comparison of the second and higher order methods.

6.2 Comparison of Performance of Higher and 2-nd Order Methods.

If we operate on the same grid, of say (n x m) points, the higher order discretization (HOD) algorithm requires more storage locations than that of the 2-nd order (the 2-OD). When applying the ADI scheme, HOD needs about (6 x n x m) storage locations (for function, its first and second partial derivatives). When using 2-OD, only (2 x n x m) memory units are required. This difference becomes important, when using HOD on fine grids, like 100x100, which may exceed memory availability on a medium-size computer like CYBER-825. The above estimate of storage required is certainly very approximate. It does not take into account, for instance, the intermediate memory demand to store arrays resulting from the Thomas algorithm operation. We may take these inaccuracies as insignificant, though.

Algebraic structure resulting from applying a particular differencing scheme also affects the number of elementary computer operations (operation count). It is known (Hirsh,1975) that on a (n x m) grid:

for 2-OD - ops = (12 x n x m)

for HOD - ops = (216 x n x m)

which means, that HOD requires about 18 times more operations to complete two ADI sweeps, than the 2-OD algorithm. In reality, however, this difference can be substantially decreased by optimization of the Thomas Algorithm for block 3-diagonal systems. In the Table below we present CPU time (CYBER-835) for a single ADI iteration (2 sweeps) required by HOD or 2-OD method:

TABLE 6.1 - CPU time (seconds) for a single ADI iteration Comparison of second and higher order algorithms.

grid spacing / algorithm	0.1	0.05	0.025	0.0167
2-OD	0.036	0.152	0.634	1.45
HOD	0.124	0.463	1.79	4.

The first observation is that in both cases, time is proportional to the number of points in a grid. Next, we may observe, that time required for a single grid point is for 2-OD about 4.2E-4 secs., and for HOD about 1.2E-3 secs., which makes HOD performance worse by only a factor of 3. Let us remember that those particular relations come from optimization of Thomas Algorithm by simplifying expressions wherever we have zero in 3x3 blocks resulting from Hermitian formulae (A.1), (A.2) and (A.3) or governing equation.

There is also another comparison to be made, i.e. iterative effectiveness of algorithms based on discretization of the second, or higher order. The rate of the convergence of the iterative scheme depends on the relaxation coefficient. In the case of 2-OD we employed Wachspress optimization. In Sec.4.2.1 (Figs. 4.2 and 4.3) we have compared the rate of iterative convergence with and without optimization for algorithm based on 2-OD. It was demonstrated that optimization yielded much faster convergence. Since the Wachspress optimization is unavailable for ADI block 3-diagonal systems, iterations of HOD method are driven by a single relaxation parameter, (optimized by trial and error procedure). This, however, takes many more iterations when compared to the optimized second order algorithm. For example, if stagnation pressure is zero, C=100, and computations are carried out on the grid 60x60, HOD algorithm requires 275 iterations to reduce residue down to machine accuracy, while the 2-OD method yields convergence after 30 iterations.

In this Section we discussed 3 criteria of algorithm performance:

a) memory storage requirements,

b) time of a single ADI iteration,

c) number of iterations of ADI scheme.

It was shown, that the 2-OD algorithm is superior to that of HOD in all three aspects. The only one remaining is the accuracy of a computed solution. This topic is discussed in the following section. It will be shown, that should the rigorous analysis of truncation error be performed, we may apply extrapolation of results obtained by the 2-OD algorithm and achieve high accuracy with much smaller cost compared to the performance of HOD. For instance, in the case presented in Table 6.4 (next Section) we may see that extrapolation from the 2-nd order results obtained on grids 10x10 and 20x20 is more accurate than results computed on grid 10x10 using the HOD algorithm. Moreover, the overall time required for 2-nd order computations with extrapolation included, was about 8 secs., while for the HOD algorithm it was 19 seconds.

6.3 Analysis of the discretization error.

The concept of discretization error was explained in Sec. 3.3. It was shown, that the accuracy of discretization (Sec. 3.2) is reflected in the rate of convergence of the truncation error to zero (grid-convergence with $\Delta x, \Delta z \longrightarrow 0$). This rate of grid-convergence associated with the order of discretization, has been examined thoroughly for the second order methods (see Sec. 4.2.2, 5.6.5). It was stated theoretically (Sec. 3.3), that the grid convergence order depends on a differencing scheme. It was shown, that the use of a differencing scheme of the order 2, yielded second order ($O(r^2)$) grid-convergence of discretization error.

When applying higher order discretization, we make use of the Hermitian formulae which are not of the same accuracy (see Appendix). First of all, the Pade formula ((A.3) in Appendix) is $O(r^3)$, while difference expressions for the interior grid points are of higher accuracy (of the order $O(r^5)$ and $O(r^6)$, resp.). Secondly, the expressions for the interior points are also of uneven accuracy: (A.1) is $O(r^5)$ and (A.2) is $O(r^6)$, while differences for mixed derivatives for the interior and boundary are $O(r^4)$. Therefore it is not quite clear what the aggregated order of the resulting algorithm will be.

We may speculate, however, that the overall discretization error is dominated by the lowest accuracy element. Taking only the interior of the computational domain into account, this error should be dominated by residuals of mixed derivatives estimates, which are of $O(r^4)$. On the other hand, our algorithm is implicitly utilizing the Pade formula of the 3-d order accuracy on the boundary. Therefore, we may expect the overall order to drop below $O(r^4)$, since the error of the differences containing the boundary points may affect the accuracy of computations in the interior of the domain.

The method of estimating the order of discretization error has been described in Sec. 3.3. Since this method is valid asymptotically, i.e. as $r \longrightarrow 0$, we feel that it is necessary to check whether for our type of algorithm (HOD), the numerical approximate estimation of the order of grid-convergence is valid. In other words we want to make sure that the asymptotic justification of the approximate order estimation is valid. In order to do that, we solve the problem which was stated by (5.6.8) of Sec. 5.6.5. This problem has a known analytical solution:

$$G = (1 - y/h(x))^3 \sin(\pi x)/\pi \tag{6.1}$$

$$H(x) = H_1(x) + H_2(x)$$

where H_1 is a segment of a circle of curvature p_s,

H_2 is small harmonic deformation

At this point we are interested in estimating the order of truncation error by direct reference to a known solution. Later we will carry out the comparison of the order of discretization estimated by reference to the exact analytical solution (formula (iii) of Sec. 5.6.5) with estimation based on L-2 distance of results computed on three consecutive grids ((iv) of the same section).

When comparing the numerical solution to the analytical one, we will consider the three following cases:

(A) - computational domain Δis almost flat (H_1=1) : Tables 6.2 a,b,c,

(B).-Δ is convex (curvature of H_1 is 1.5) : Tables 6.3 a,b,c,

(C) -Δ is concave (curvature of H_1 is -1.5) : Tables 6.4 a,b,c,

Each of the cases is represented by three tables. Tables 'a' and 'b' show order estimation when interior, or interior and boundary grids points, respectively, are taken into account. Each of these tables consist of two parts. The upper one list $[f^k - \tilde{f}](Gk)$, i.e. L-2 distance between the field computed on a grid Gk, and exact solution $\tilde{f}$. The lower one shows estimation of grid-convergence order using results obtained on 2 consecutive grids when referenced to the exact solution. Tables 'c' show order estimation when the exact solution is ignored and numerical results computed on three grids of increasing density are involved.

Conclusions are as follows:

(1) For all 3 cases (A),(B) and (C), with grid refining, the order

of the truncation error is approaching the value of 4, but with some exceptions it is considerably lower than 4.

(2) Comparing tables 'a' and 'b' with 'c' we may see that relative order estimation (tables 'c') show underestimation, and is not stablized for the range of grid sizes we tested. This variability means that truncation error is still influenced by large changes of higher derivatives in the Lagrange residuum. The numerical solution is usually meant to be valid if discretization error depends primarily on some power of a grid constant r. In view of this principle, our results, except for the case (A) are not satisfactory for any of the grids tested. If we take results for the same case, obtained by the algorithm of the second order (Tables 5.20, 5.21 and 5.23), we see that the variability of the truncation error is vanishing starting from grid 10x10. This is confirmed by uniformity in the estimation of the grid convergence order also for the finer grids. In view of this argument, the second order method yields results of better numerical quality.

Due to the unstabilized order of convergence we cannot apply extrapolation. Therefore in order to get results of high accuracy we must compute on fine grids.

(3) A comparison of tables 'a' and 'b' can inform by how much computations of boundary derivatives affect overall grid-convergence order. One can observe that there are negligible differences except for ψ_{xx}, h_x, and h_{xx}, which show higher order if their boundary values are dropped. This is due to high gradients of these functions at the boundary.

TABLE 6.2.a Estimation of $\|f^n - \tilde{f}\|$ and the resulting estimates of the grid-convergence order (interior of the solution domain) for the problem (5.6.8), HOD scheme. Upper boundary of the solution domain is flat.

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$\|f^{10} - \tilde{f}\|$	.365E-06	.384E-05	.102E-05	.963E-05	.325E-04	.104E-04	.193E-04	.224E-03	.157E-03
$\|f^{20} - \tilde{f}\|$	.286E-07	.242E-06	.769E-07	.607E-06	.223E-05	.655E-06	.168E-05	.156E-04	.110E-04
$\|f^{30} - \tilde{f}\|$	.615E-08	.475E-07	.164E-07	.125E-06	.439E-06	.128E-06	.367E-06	.313E-05	.219E-05
$\|f^{40} - \tilde{f}\|$	.205E-08	.150E-07	.543E-08	.403E-07	.139E-06	.413E-07	.122E-06	.995E-06	.694E-06
$\|f^{60} - \tilde{f}\|$	.420E-09	.296E-08	.111E-08	.817E-08	.274E-07	.838E-08	.252E-07	.197E-06	.137E-06
$f^{10}: f^{20}: \tilde{f}$	3.67	3.99	3.73	3.99	3.87	3.99	3.52	3.84	3.84
$f^{20}: f^{30}: \tilde{f}$	3.79	4.02	3.81	3.91	4.00	4.02	3.75	3.96	3.98
$f^{30}: f^{40}: \tilde{f}$	3.82	4.01	3.84	3.92	4.01	3.94	3.83	3.98	4.00
$f^{40}: f^{60}: \tilde{f}$	3.91	4.01	3.91	3.94	4.00	3.93	3.89	3.99	4.00

TABLE 6.2.b Estimation of $\| f^n - \tilde{f} \|$ and the resulting estimates of the grid-convergence order (interior and boundary of the solution domain) for the problem (5.6.8), HOD scheme. Upper boundary of the solution domain is flat.

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$\| f^{10} - \tilde{f} \|$	.366E-06	.509E-05	.233E-05	.294E-04	.396E-04	.366E-04	.193E-04	.216E-03	.128E-03
$\| f^{20} - \tilde{f} \|$	.286E-07	.342E-06	.122E-06	.190E-05	.224E-05	.132E-06	.168E-05	.158E-04	.899E-05
$\| f^{30} - \tilde{f} \|$	.615E-08	.703E-07	.257E-07	.360E-06	.438E-06	.216E-06	.367E-06	.323E-05	.179E-05
$\| f^{40} - \tilde{f} \|$	.206E-08	.226E-07	.842E-08	.113E-06	.140E-06	.695E-07	.122E-06	.103E-05	.568E-06
$\| f^{60} - \tilde{f} \|$	.420E-09	.456E-08	.172E-08	.229E-07	.280E-07	.137E-07	.252E-07	.207E-06	.112E-06
$f^{10} : f^{20} : \tilde{f}$	3.67	3.90	4.26	3.95	4.15	4.80	3.52	3.77	3.84
$f^{20} : f^{30} : \tilde{f}$	3.79	3.90	3.84	4.11	4.02	4.45	3.75	3.92	3.98
$f^{30} : f^{40} : \tilde{f}$	3.82	3.94	3.87	4.02	3.95	3.94	3.83	3.96	4.00
$f^{40} : f^{60} : \tilde{f}$	3.91	3.95	3.92	3.94	3.97	4.00	3.89	3.97	4.00

TABLE 6.2.c Numerical estimation of the grid-convergence order for the problem (5.6.8), HOD scheme. Upper boundary of the domain is flat.

	BOUNDARY INCLUDED								
	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.365E+01	.385E+01	.430E+01	.124E+01	.415E+01	.485E+01	.350E+01	.375E+01	.380E+01
$f^{20}-f^{40}-f^{60}$	.375E+01	.390E+01	.380E+01	.410E+01	.395E+01	.425E+01	.375E+01	.390E+01	.395E+01
$f^{30}-f^{40}-f^{60}$	.375E+01	.390E+01	.385E+01	.405E+01	.395E+01	.410E+01	.380E+01	.395E+01	.395E+01
	INTERIOR ONLY								
	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.365E+01	.395E+01	.370E+01	.400E+01	.385E+01	.395E+01	.350E+01	.380E+01	.380E+01
$f^{20}-f^{40}-f^{60}$	.375E+01	.400E+01	.380E+01	.390E+01	.400E+01	.395E+01	.375E+01	.395E+01	.395E+01
$f^{30}-f^{40}-f^{60}$	.375E+01	.400E+01	.380E+01	.390E+01	.400E+01	.390E+01	.380E+01	.395E+01	.395E+01

TABLE 6.3.a Estimation of $\|f^n - \tilde{f}\|$ and the resulting estimates of the grid-convergence order for the problem (5.6.8), HOD scheme. Upper boundary of the solution domain has curvature 1.5.

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$\|f^{10} - \tilde{f}\|$	.639E-06	.586E-05	.214E-05	.178E-04	.100E-03	.166E-04	.117E-03	.869E-03	.450E-02
$\|f^{20} - \tilde{f}\|$	.953E-07	.484E-06	.261E-06	.188E-05	.550E-05	.189E-05	.129E-04	.538E-04	.275E-03
$\|f^{30} - \tilde{f}\|$	.268E-07	.125E-06	.728E-07	.458E-06	.103E-05	.503E-06	.320E-05	.103E-04	.523E-04
$\|f^{40} - \tilde{f}\|$	.103E-07	[illegible]	.279E-07	.162E-06	.322E-06	.188E-06	.115E-05	.321E-05	.162E-04
$\|f^{60} - \tilde{f}\|$	.240E-08	.102E-07	.647E-08	.351E-07	.634E-07	.436E-07	.260E-06	.626E-06	.315E-05
$f^{10}: f^{20}: \tilde{f}$	2.77	3.60	3.04	3.24	4.19	3.13	3.18	4.01	4.03
$f^{20}: f^{30}: \tilde{f}$	3.08	3.33	3.15	3.48	4.13	3.26	3.43	4.07	4.09
$f^{30}: f^{40}: \tilde{f}$	3.31	3.50	3.34	3.62	4.05	3.43	3.56	4.06	4.07
$f^{40}: f^{60}: \tilde{f}$	3.60	3.71	3.60	3.77	4.01	3.60	3.67	4.03	4.04

TABLE 6.3.b Estimation of $\|f^n - \tilde{f}\|$ and the resulting estimates of the grid-convergence order (interior and boundary of the solution domain) for the problem (5.6.8), HOD scheme. Upper boundary of the solution domain has curvature 1.5.

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$\|f^{10} - \tilde{f}\|$	.639E-06	.624E-05	.310E-05	.323E-04	.145E-03	.451E-04	.117E-03	.105E-02	.693E-02
$\|f^{20} - \tilde{f}\|$	.963E-07	.445E-06	.371E-06	.281E-05	.144E-04	.260E-05	.129E-04	.968E-04	.684E-03
$\|f^{30} - \tilde{f}\|$	.268E-07	.113E-06	.104E-06	.749E-06	.361E-05	.696E-06	.320E-05	.235E-04	.169E-03
$\|f^{40} - \tilde{f}\|$	.103E-07	.414E-07	.394E-07	.261E-06	.131E-05	.252E-06	.115E-05	.841E-05	.612E-04
$\|f^{60} - \tilde{f}\|$	.240E-08	.958E-08	.919E-08	.592E-07	.300E-06	.586E-07	.260E-06	.191E-05	.140E-04
$f^{10}: f^{20}: \tilde{f}$	2.77	3.81	3.06	3.52	3.33	4.12	3.18	3.44	3.34
$f^{20}: f^{30}: \tilde{f}$	3.08	3.38	3.15	3.26	3.42	3.25	3.43	3.49	3.44
$f^{30}: f^{40}: \tilde{f}$	3.31	3.48	3.36	3.67	3.53	3.54	3.56	3.57	3.54
$f^{40}: f^{60}: \tilde{f}$	3.60	3.61	3.59	3.66	3.63	3.59	3.67	3.65	3.64

TABLE 6.3.c Numerical estimation of the grid-convergence order for the problem (5.6.8), HOD scheme. Upper boundary of the solution domain has curvature 1.5.

	BOUNDARY INCLUDED								
	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.275E+01	.385E+01	.310E+01	.355E+01	.330E+01	.420E+01	.310E+01	.340E+01	.330E+01
$f^{20}-f^{40}-f^{60}$	.305E+01	.335E+01	.315E+01	.335E+01	.340E+01	.330E+01	.345E+01	.350E+01	.345E+01
$f^{30}-f^{40}-f^{60}$	.320E+01	.340E+01	.325E+01	.365E+01	.345E+01	.350E+01	.350E+01	.350E+01	.350E+01
	INTERIOR ONLY								
	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.275E+01	.365E+01	.305E+01	.325E+01	.415E+01	.310E+01	.310E+01	.400E+01	.400E+01
$f^{20}-f^{40}-f^{60}$	.305E+01	.330E+01	.315E+01	.345E+01	.410E+01	.325E+01	.345E+01	.405E+01	.405E+01
$f^{30}-f^{40}-f^{60}$	.320E+01	.340E+01	.320E+01	.355E+01	.405E+01	.335E+01	.350E+01	.405E+01	.405E+01

TABLE 6.4.a Estimation of $\|f^n - \tilde{f}\|$ and the resulting estimates of the grid-convergence order (interior of the solution domain) for the problem (5.6.8), HOD scheme. Upper boundary of the solution domain has curvature -1.5.

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$\|f^{10} - \tilde{f}\|$	.377E-05	.186E-04	.100E-04	.464E-04	.161E-03	.553E-04	.117E-03	.869E-03	.450E-02
$\|f^{20} - \tilde{f}\|$	.434E-06	.189E-05	.116E-05	.473E-05	.119E-04	.608E-05	.129E-04	.538E-04	.275E-03
$\|f^{30} - \tilde{f}\|$	.109E-06	.449E-06	.293E-06	.114E-05	.255E-05	.151E-05	.320E-05	.103E-04	.523E-04
$\|f^{40} - \tilde{f}\|$	.392E-07	.157E-06	.106E-06	.403E-06	.852E-06	.542E-06	.115E-05	.321E-05	.162E-04
$\|f^{60} - \tilde{f}\|$	.898E-08	.349E-07	.242E-07	.899E-07	.180E-06	.123E-06	.260E-06	.626E-06	.316E-05
$f^{10}: f^{20}: \tilde{f}$	3.12	3.30	3.11	3.29	3.75	3.18	3.18	4.01	4.03
$f^{20}: f^{30}: \tilde{f}$	3.41	3.54	3.41	3.60	3.80	3.44	3.43	4.07	4.09
$f^{30}: f^{40}: \tilde{f}$	3.55	3.66	3.54	3.62	3.81	3.56	3.56	4.06	4.07
$f^{40}: f^{60}: \tilde{f}$	3.63	3.71	3.63	3.70	3.83	3.65	3.67	4.03	4.04

TABLE 6.4.b Estimation of $\| f^n - \tilde{f} \|$ and the resulting estimates of the grid-convergence order (interior and boundary of the solution domain) for the problem (5.6.8), HOD scheme. Upper boundary of the solution domain has curvature -1.5.

	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$\| f^{10} - \tilde{f} \|$	.377E-05	.131E-04	.131E-04	.573E-04	.165E-03	.750E-04	.117E-03	.105E-02	.693E-02
$\| f^{20} - \tilde{f} \|$	.434E-06	.133E-05	.146E-05	.547E-05	.163E-04	.730E-05	.129E-04	.968E-04	.684E-03
$\| f^{30} - \tilde{f} \|$	.109E-06	.319E-06	.364E-06	.120E-05	.403E-05	.176E-05	.320E-05	.235E-04	.169E-03
$\| f^{40} - \tilde{f} \|$	.392E-07	.112E-06	.131E-06	.425E-06	.146E-05	.627E-06	.115E-05	.841E-05	.612E-04
$\| f^{60} - \tilde{f} \|$	.898E-08	.249E-07	.299E-07	.935E-07	.333E-06	.141E-06	.260E-06	.191E-05	.140E-04
$f^{10} : f^{20} : \tilde{f}$	3.12	3.30	3.16	3.39	3.34	3.36	3.18	3.44	3.34
$f^{20} : f^{30} : \tilde{f}$	3.41	3,53	3.43	[illegible]	3.44	3.51	3.43	3.49	3.44
$f^{30} : f^{40} : \tilde{f}$	3.55	3.64	3.56	3.62	3.54	3.58	3.56	3.57	3.54
$f^{40} : f^{60} : \tilde{f}$	3.63	3.71	3.65	3.73	3.64	3.68	3.67	3.65	3.64

TABLE 6.4.c Numerical estimation of the grid-convergence order for the problem (5.6.8), HOD scheme. Upper boundary of the solution domain has curvature -1.5.

BOUNDARY INCLUDED								
	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$r^{10}-r^{20}-r^{40}$	.305E+01	.325E+01	.310E+01	.335E+01	330E+01	.335E+01	.310E+01	.340E+01	.330E+01
$r^{20}-r^{40}-r^{60}$	.340E+01	.350E+01	.340E+01	.365E+01	.345E+01	.350E+01	.345E+01	.350E+01	.345E+01
$r^{30}-r^{40}-r^{60}$	.350E+01	.360E+01	.350E+01	.355E+01	.350E+01	.355E+01	.350E+01	.350E+01	.350E+01
INTERIOR ONLY									
	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$r^{10}-r^{20}-r^{40}$	.305E+01	.325E+01	.305E+01	.325E+01	.370E+01	.315E+01	.310E+01	.400E+01	.400E+01
$r^{20}-r^{40}-r^{60}$	.340E+01	.355E+01	.340E+01	.350E+01	.380E+01	.345E+01	.345E+01	.405E+01	.405E+01
$r^{30}-r^{40}-r^{60}$	.350E+01	.360E+01	.350E+01	.355E+01	.380E+01	.350E+01	.350E+01	.405E+01	.405E+01

In Sec. 5.6.5 we have examined the concept of extrapolation of results computed on different grids. This analysis was pursued on grounds of the 2-OD method. Now we can compare results of the extrapolation with those obtained by the use of HOD method. For convenience;.we collect data in TABLE 6. at the end of the Section. The distance of an exact solution from extrapolated values for the 2-OD method was displayed before in Table 5.29. The Case Codes of Table 6.4 are to be understood as follows:

- I,II,II,IV are for the HOD method. They present the distance of exact solution from the ones computed on grids G10, G20, G40, G60 with space increments respectively: r=1/10, 1/20, 1/40 and 1/60.
- V,VI,VII are for the 2-OD method. Here we have distances of the exact solution from the field obtained by pairwise extrapolation of the results computed on grids of r=1/10 and r=1/20; r=1/20, r=1/40 and r=1/40, r=1/60.

As is shown, the results obtained by the HOD algorithm on grid 10x10 are worse than those coming from extrapolation from grids 10x10 and 20x20. Similarly, extrapolation from grids 40x40 and 60x60 is closer to the exact solution than higher order results computed on grid 40x40. From this comparison, one may see that extrapplation is a very efficient tool to get highly accurate results. The second observation is that even though it might seem that application of the higher order method is a more straightforward way of obtaining highly accurate results, we must acknowledge that in view of experiments presented heretofore, the second order method with extrapolation is a much cheaper way of getting accurate results.

For the sake of completeness, we present below 2 tables (6.5, 6.6) presenting order estimations for HOD computations of our original physical flow with free surface, As we can notice, the increase of grid density causes an increase of the order of discretization error. This suggests that the differencing scheme (see Appendix) yields 4-th order (or close to) convergence. Unfortunately, variability of the order of grid-convergence estimates (with grids refining), makes extrapolation unfeasible in the range of grid spacing examined in this work.

TABLE 6.5 Numerical estimation of the grid-convergence order for an ideal flow problem (3.2)-(3.3), HOD scheme. Parameters: $p_s = 1.5$, $C = 100$.

	BOUNDARY INCLUDED								
	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.275E+01	.385E+01	.310E+01	.355E+01	.330E+01	.420E+01	.310E+01	.340E+01	.330E+01
$f^{20}-f^{40}-f^{60}$	.305E+01	.335E+01	.315E+01	.335E+01	.340E+01	.330E+01	.345E+01	.350E+01	.345E+01
$f^{30}-f^{40}-f^{60}$	.320E+01	.340E+01	.325E+01	.365E+01	.345E+01	.350E+01	.350E+01	.350E+01	.350E+01
	INTERIOR ONLY								
	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.275E+01	.365E+01	.305E+01	.325E+01	.415E+01	.310E+01	.310E+01	.400E+01	.400E+01
$f^{20}-f^{40}-f^{60}$	.305E+01	.330E+01	.315E+01	.345E+01	.410E+01	.325E+01	.345E+01	.405E+01	.405E+01
$f^{30}-f^{40}-f^{60}$	.320E+01	.340E+01	.320E+01	.355E+01	.405E+01	.335E+01	.350E+01	.405E+01	.405E+01

TABLE 6.6 Numerical estimation of the grid-convergence order for an ideal flow problem (3.2)-(3.3), HOD scheme. Parameters: $p_s = -1.5$, $C = 10$.

	BOUNDARY INCLUDED								
	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.305E+01	.325E+01	.310E+01	.335E+01	.330E+01	.335E+01	.310E+01	.340E+01	.330E+01
$f^{20}-f^{40}-f^{60}$	.340E+01	.350E+01	.340E+01	.365E+01	.345E+01	.350E+01	.345E+01	.350E+01	.345E+01
$f^{30}-f^{40}-f^{60}$	.350E+01	.360E+01	.350E+01	.355E+01	.350E+01	.355E+01	.350E+01	.350E+01	.350E+01
	INTERIOR ONLY								
	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
$f^{10}-f^{20}-f^{40}$	.305E+01	.325E+01	.305E+01	.325E+01	.370E+01	.315E+01	.310E+01	.400E+01	.400E+01
$f^{20}-f^{40}-f^{60}$	.340E+01	.355E+01	.340E+01	.350E+01	.380E+01	.345E+01	.345E+01	.405E+01	.405E+01
$f^{30}-f^{40}-f^{60}$	.360E+01	.360E+01	.350E+01	.355E+01	.380E+01	.350E+01	.350E+01	.405E+01	.405E+01

TABLE 6.7 Distance from exact solution when curvature of the upper surface = 1.5; comparison of the extrapolated results obtained by the 2-OD method with results computed by the HOD method.

Case code	ψ	ψ_x	ψ_z	ψ_{xz}	ψ_{xx}	ψ_{zz}	h	h_x	h_{xx}
I	.640E-06	.590E-05	.210E-04	.180E-04	.100E-03	.170E-04	.120E-03	.870E-03	.450E-02
II	.940E-07	.480E-06	.260E-06	.190E-05	.550E-03	.190E-05	.130E-04	.540E-04	.280E-03
III	.100E-07	.460E-07	.280E-07	.160E-06	.320E-06	.190E-06	.120E-05	.320E-05	.160E-04
IV	.240E-08	.100E-07	.650E-08	.350E-07	.630E-07	.440E-07	.260E-06	.630E-06	.320E-05
V	.650E-06	.380E-05	.370E-06	.140E-04	.190E-04	.180E-04	.540E-04	.130E-03	.830E-03
VI	.410E-07	.280E-06	.230E-06	.110E-05	.140E-05	.120E-05	.360E-05	.880E-05	.530E-04
VII	.820E-08	.570E-07	.460E-07	.230E-06	.330E-06	.270E-06	.730E-06	.170E-05	.110E-04

CHAPTER 7

PHYSICAL INTERPRETATION OF THE RESULTS

In this Chapter we intend to show how the free surface depends on parameters of the flow. Because the shape of the free surface is interrelated with the resulting static pressure along the free boundary, we will examine the variation of this pressure when flow parameters vary. We confine ourselves to the investigation of the effects of changing the stagnation pressure p_s^*, the external pressure p_{ex}^* , the velocity scale V, the surface tension T and the length scale L. As in Chapter 2 asterisks are used to denoye dimensional quantities. We use in this chapter dimensional parameters, because they make physical interpretation of the results easier.

In the first section we assume that the external pressure acting upon the free surface is zero. In the following section we present results for nonuniform external pressure and when liquid-solid contact points are of nonequal hight.

7.1 Case of zero external pressure.

The original form of the Dynamic Boundary Condition is:

$$p^* = p_{ex}^* - T \frac{h_{xx}^*}{(1 + (h_x^*)^2)^{3/2}} \qquad (7.1)$$

p* is the static pressure at the free surface, exerted from the interior of the fluid. Let us assume that the outer pressure p* is zero. We know from the Bernoulli equation that the following is valid in the whole domain (boundary included):

$$p^{*} + \frac{\rho}{2}|\underline{V}|^{2} = p_{s}^{*} = \text{const} \tag{7.2}$$

where p_s^* is the stagnation pressure. The stagnation pressure is equal to static pressure if there is no motion of the fluid. After having appropriate scaling done (Chapter 2), and (7.2) put into (7.1), we get:

$$-\frac{h_{xx}}{(1 + h_{x}^{2})^{3/2}} = p^{*} - C|\underline{V}|^{2}/2 \tag{7.3}$$

where $p_s = p_s^* L/T$ is dimensionless stagnation pressure, C is the capillary number and V^2 is the square of the fluid speed along the free boundary.

Let $D=\rho V^2/2$ by definition, so $C=DL/T$. D is the characteristic dynamic pressure. Let $P=p_s - Cv^2/2$, which we will call the characteristic static pressure along the free surface. From (7.3) it follows, that actual shape of the free surface depends on the distribution of P(x) along the free surface. Let us make few statements about P as a function of parameters p_s^*, L, T and D.

Proposition 7.1.

a) $P \leq p_s^* L/T$.

b) P is a decreasing function of T.

c) P is a decreasing function of dynamic pressure D.

d) P is an increasing function of p_s^*.

e) P is an increasing function of L.

REMARK:

Property of monotonicity should be understood as a functional monotonicity, i.e. if 'a' is some parameter of the function P, then: P is monotonic with regard to 'a' in interval I if and only if

$$a_1 \lessgtr a_2 \Rightarrow \{ [P(a_1) \lessgtr P(a_2)] \Rightarrow \forall x \in I\ [P(a_1)(x) \lessgtr P(a_2)(x)]\}$$

All the statements of the above Proposition are easy to demonstrate on grounds of formal definition of P(x). But each of these points have an interesting physical interpretation with reference to the free surface equation: $P = -h_{xx}/(1 + h_x^2)^{3/2}$. This interpretation is presented below.

(a) $P = p - C|\underline{V}|^2/2 \leq p_s = p_s^* \frac{L}{T}$

This inequality is visualized by the fact that the dynamic (actual) free surface is always either equal to or is located beneath the static surface generated by the stagnation pressure $p_s = p_s^* L/T$. This feature can be observed on Figures 7.1 ÷ 7.4. Therefore the fluid motion, and resulting dynamic pressure, results in subsidence of free surface when compared to its static shape.

b) P is decreasing function of T .

Let us consider cases presented on Fig. 7.1.a, 7.1.b, 7.1.c. Stagnation pressure p_s^* is set constant as 0.1, similarly the dynamic pressure D=38, but the surface tension is set in decreasing sequence (T=1.0, 0.1, 0.048). According to the statement b) of the proposition, decrease of T is causing increase of P. The pressure P consists of stagnation and dynamic components. Let us discuss them one by one. Nondimensional stagnation pressure $p_s = p_s^*L/T$ is rising nonlinearly as 1/T, and so is the curvature of the static interface. In what follows, the static interface, which occures when dynamic effects are absent, will be referred to as 'shadow' interface, since what we physically have, is only dynamic free surface. In all three cases (Figures 7.1.a, 7.1.b, 7.1.c) decrease of T is causing the capillary number to change at the rate of 1/T. Therefore, when T is being decreased, the actual dynamic interface is rising, but nevertheless it is trailing behind the 'shadow' one (which was demonstrated in the statement (a)).

An interesting case is shown on Fig. 7.1.c. Here, surface tension is so small that at the given p_s^*, a static free surface would not exist, i.e. the liquid would be pushed out of the cavity. However, because of the presence of dynamic effects, overall pressure distribution is such that the capillaray surface may exist. It was found though, that further decrease of the surface tension (T<0.048) would cause the dynamic free surface to break up.

Let us consider now the case of the negative stagnation pressure.

Assume $p^*_s=-1.$, D=12, while T is variable, with values 1., 0.8, 0.7. This case is shown on Fig. 7.2.a, 7.2.b, 7.2.c. With decreasing T, the 'shadow' free surface lowers down as $p_s = p^*_s L/T$ is being decreased ($p^*_s<0$). Dynamic interface, as was stated in a), is located beneath the 'shadow' interface. Here again one can observe an interesting phenomenon (Fig. 7.2.c), when T is too small to maintain dynamic free surface, and liquid is being sucked into the cavity. However, if the motion of the fluid was stopped, the pressure P would increase sufficiently to provide existence of the free surface.

c) P is decreasing function of dynamic pressure D.

In this point we analyse the physical results of increasing dynamic pressure $D=\varrho V^2/2$, or physically, quadratic increase of velocity of the fluid. Let us fix $p^*_s=1.$, and T=1. At the outset, we assume D=0, i.e. no motion, which results in the buidup of static free surface.- Next we set D=300 and D=380. This is shown on Fig. 7.4.a. Increase of D causes free surface to subside gradually. It is worth to mention, that increase of D to the value 382 would result in collapse of the free surface (liquid is sucked into the cavity).

d) P is increasing function of p^*_s.

Let us assume, that all parameters except p^*_s are kept constant, i.e. D=150, T=1. Fig. 7.4.a - 7.4.e display cases with p^*_s ranging from negative values to high positive ($p^*_s \in \{ -0.1, 0., 0.75, 1.5, 1.85\}$). Actual, i.e dynamic free surface is seen to move gradually upwards.

The same with 'shadow' interface. The interesting thing is, that with p_s^* increasing, the distance between dynamic and 'shadow' interface is diminishing. This means, that the deformation of static interface due to the fluid motion, depends on the velocity scale, but also on the value of the stagnation pressure. The higher p_s^*, the bigger must be the velocity scale to bring free surface to the proportional deformation. Establishing of the exact relation between p_s^* and D in this aspect is difficult, since the static pressure P depends also on the fluid speed along the interface, which is not known a priori.

The last case we discuss in this section is the liquid surface in different geometries (variable length scale L), all other parameters (T, D, p_s^*) kept constant. For simplicity, let D=0, i.e. we will investigate only static liquid free surface of variable span. Having T and p_s^* fixed, we find that moving apart contact points of the liquid with solid walls, causes the curvature of the meniscus to increase. If the contact points are too far away from each other (e.g. L=3), there is no meniscus of the circular shape, which would have the preassigned curvature (=Lp_s^*/T) It means the surface tension T is unable to counteract the internal pressure p_s^* and the liquid bridge is pushed out of the cavity. For the example presented on 7.5, the maximum feasible L is 2.5.

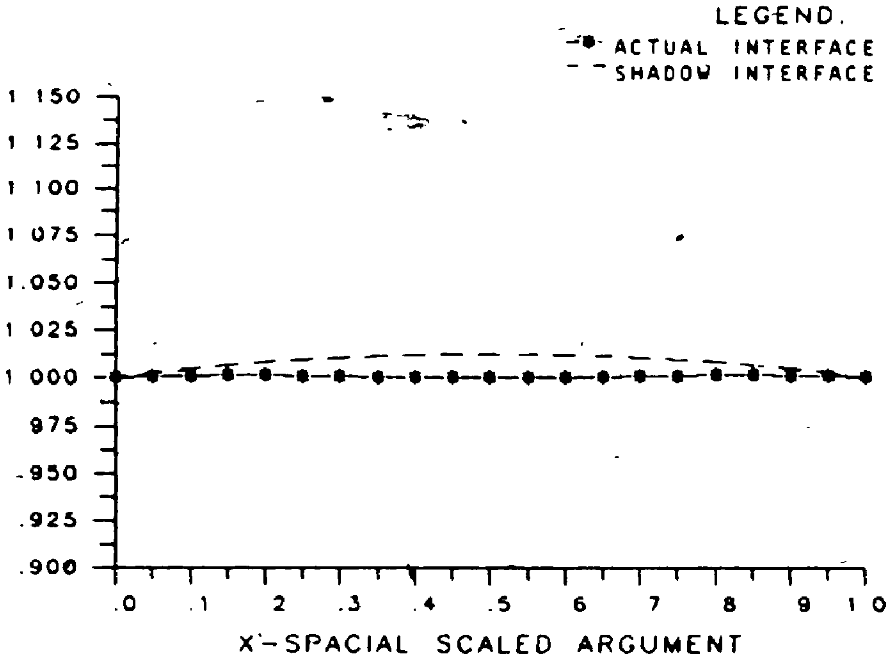


FIGURE 7.1.a Free surface shape when T=1.0 (ρ_s^*=0.1, D=38)

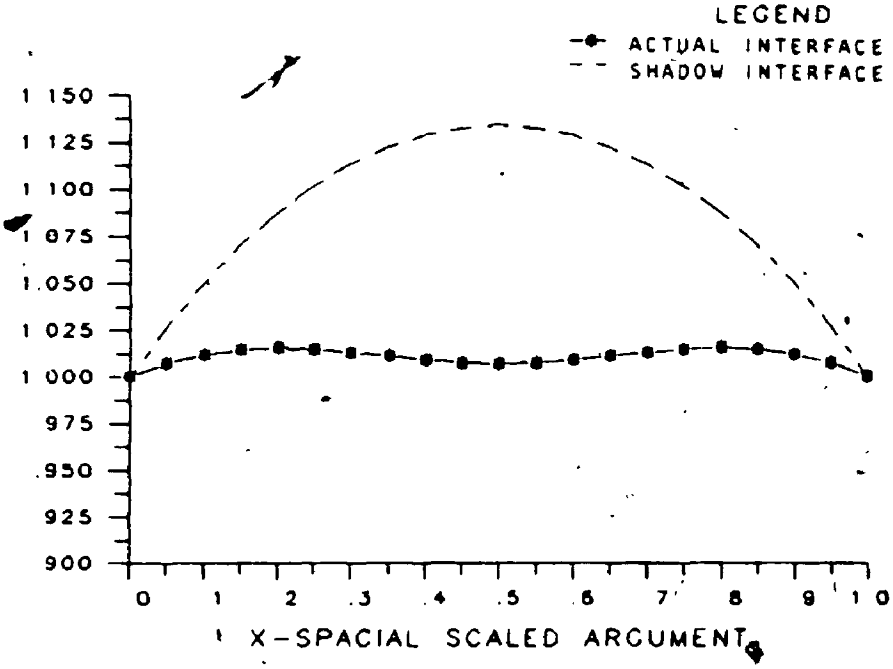


FIGURE 7.1.b Free surface shape when T=0.1 (p_s^*=0.1, D=38)

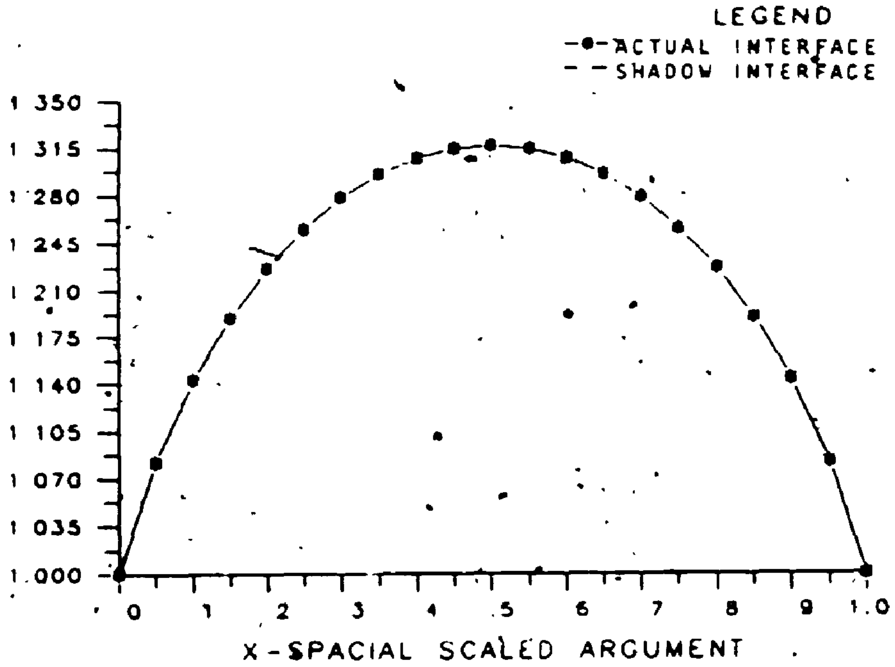


FIGURE 7.1.c Free surface shape when T=0.048 (p_s^*=0.1, D=38)

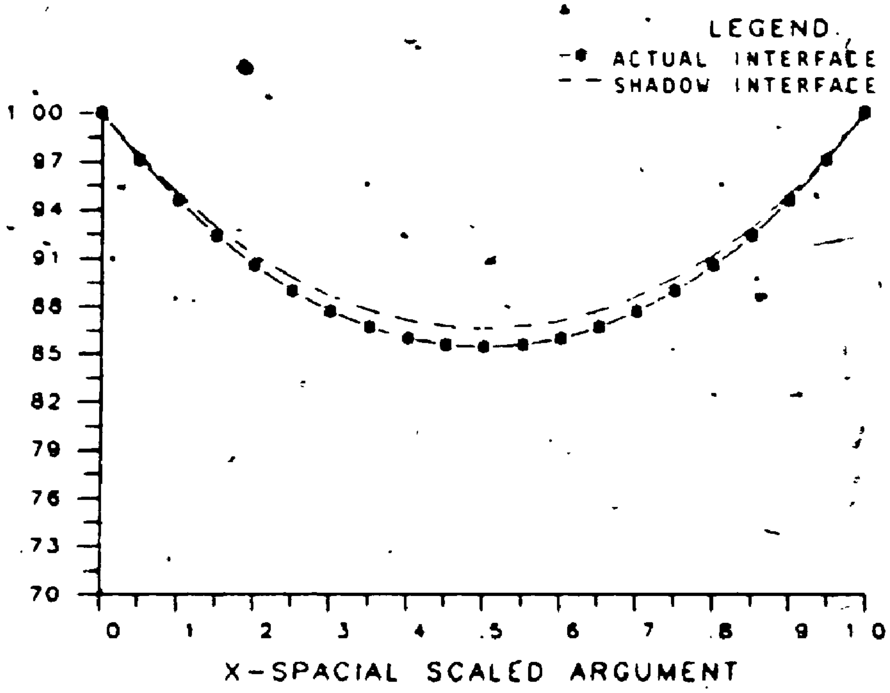


FIGURE 7.2.a. Free surface shape when T=1.0 (p_s^*=-1, D=12).

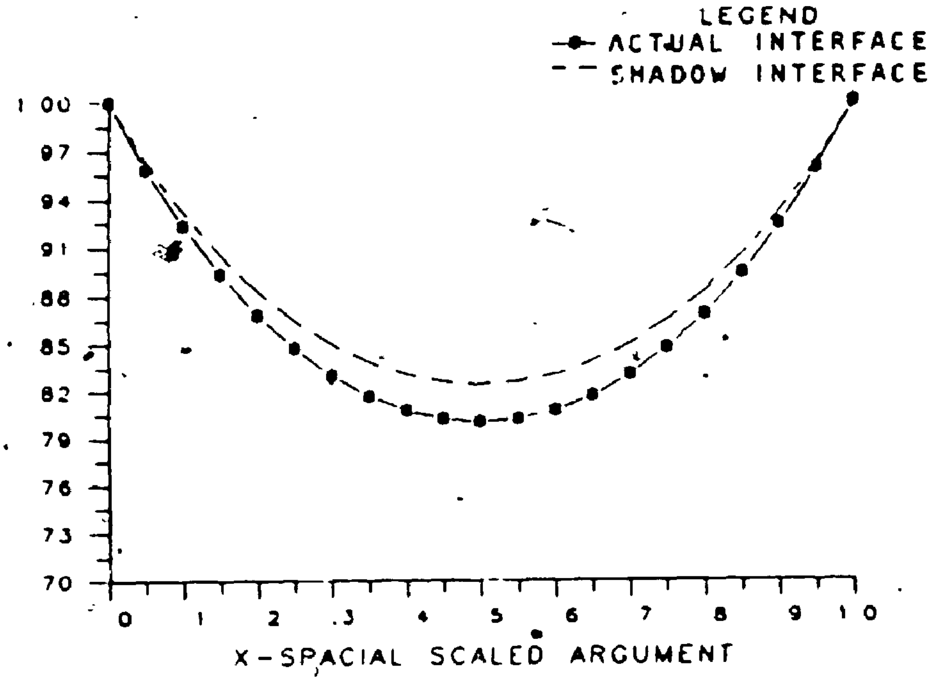


FIGURE 7.2.b Free surface shape when T=0.8 (p^*_s=-1, D=1)

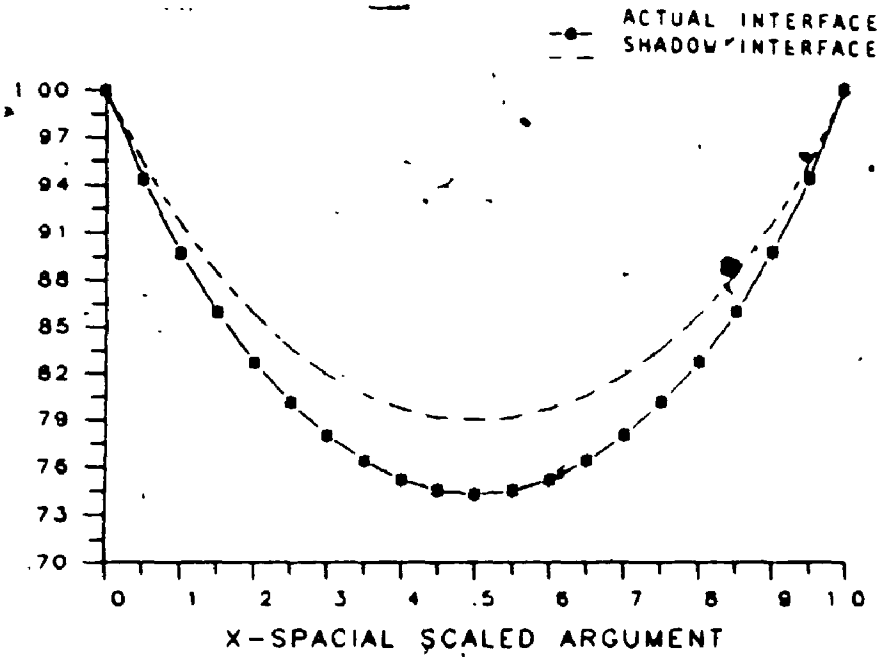


FIGURE 7.2.c Free surface shape when T=0.7 (p_s^*=-.1, D=12).

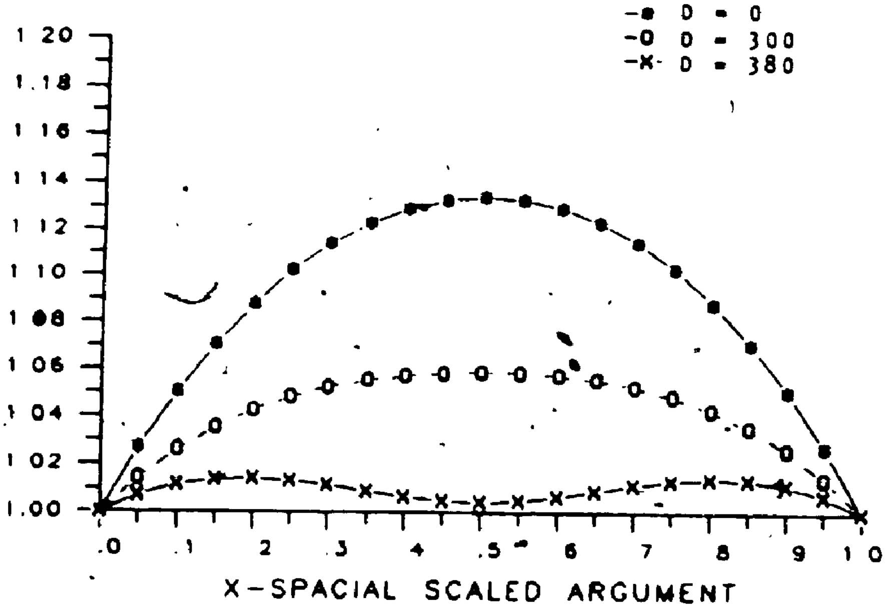


FIGURE 7.3 Free surface shape for variable D=0., 300, 380, (p_s^*=1, T=1.)

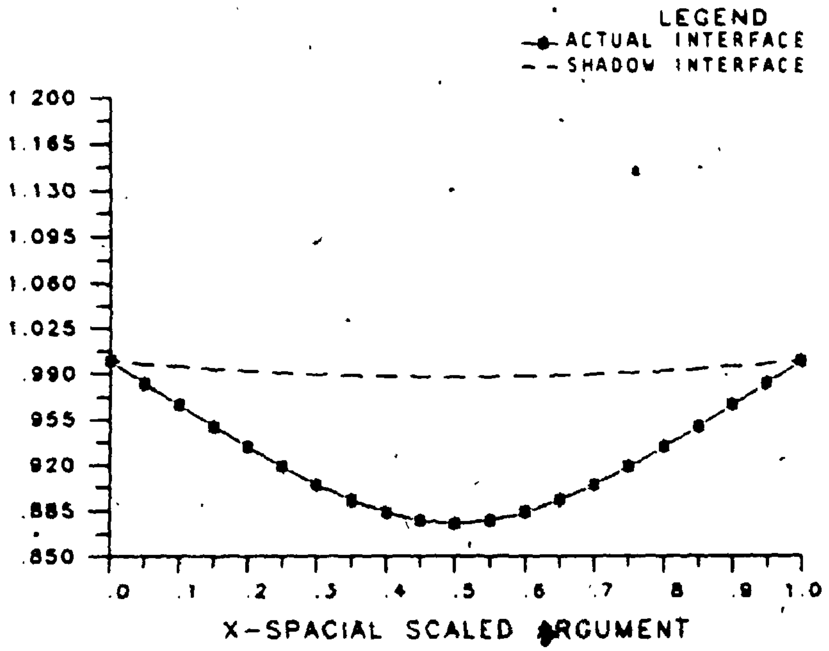


FIGURE 7.4.a Free surface shape when p_s^*=-.1, (D=150, T=1.).

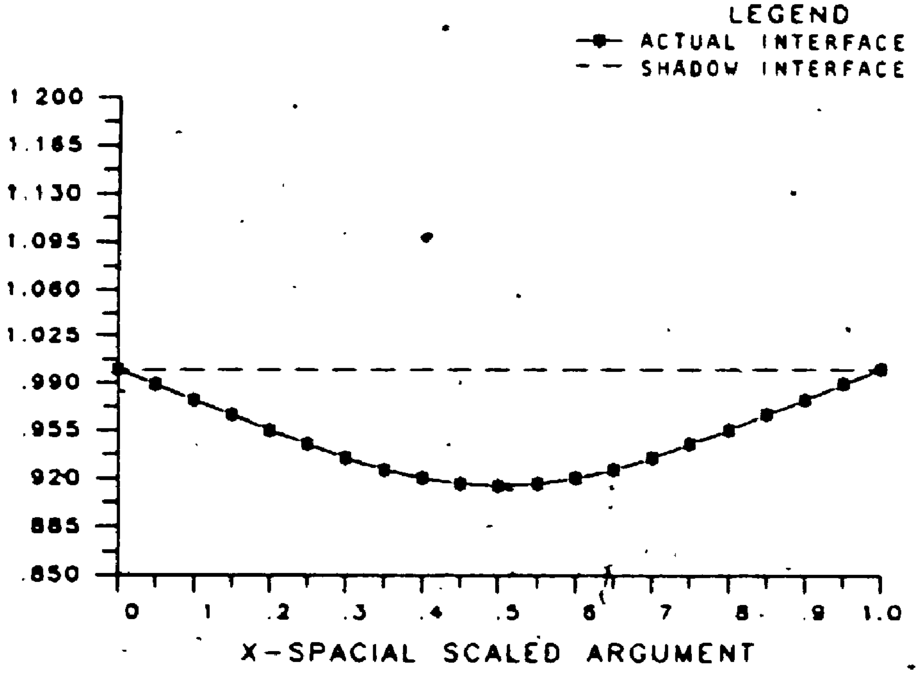


FIGURE 7.4.b Free surface sbape when $p_s^* = .0$, (D=150, T=1.).

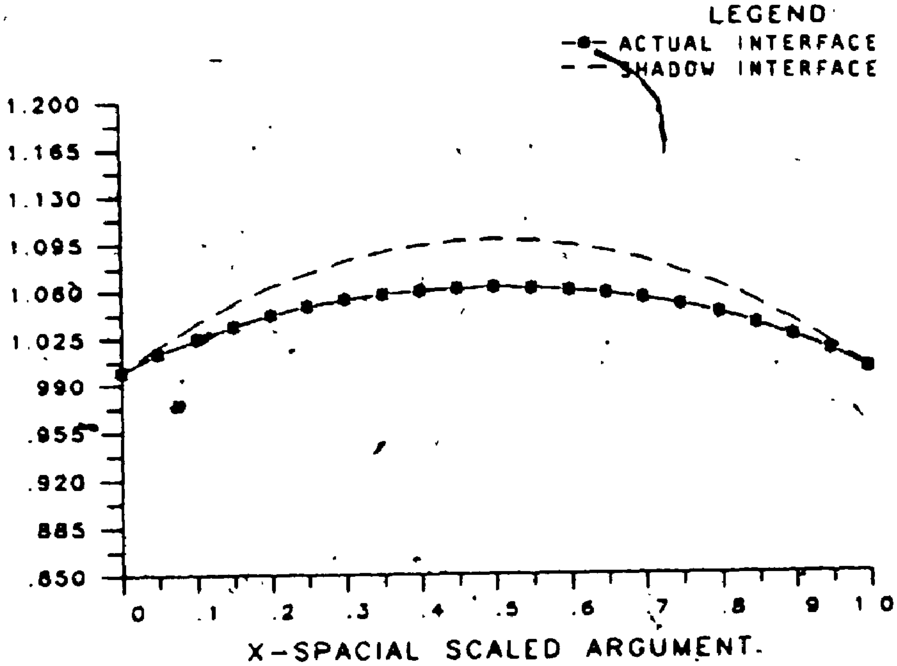


FIGURE 7.4.c Free surface shape when p_s^*=0.75, (D=150, T=T.).

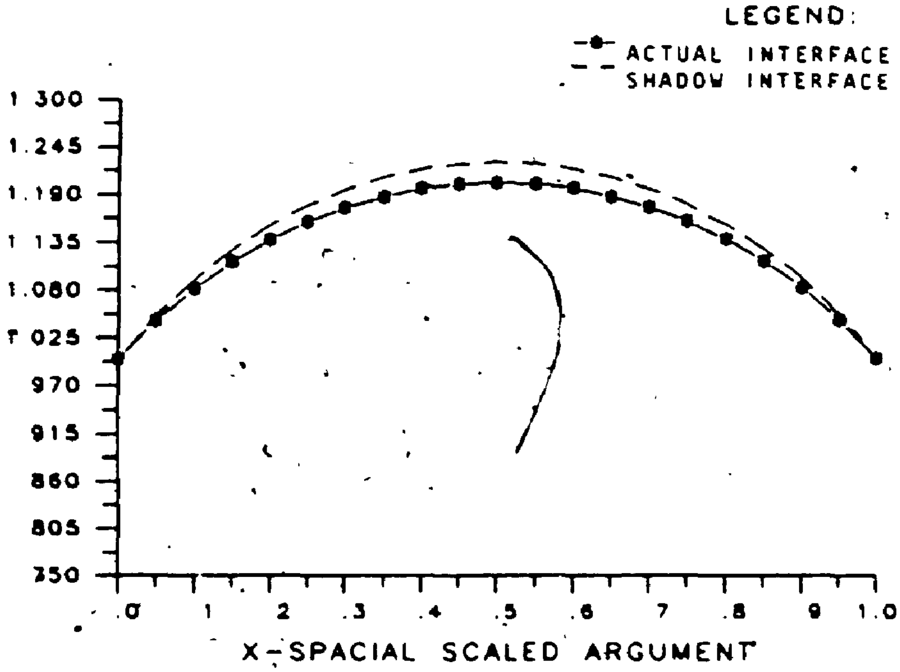


FIGURE 7.4.d Free surface shape when p_s^*=1.5, (D=150, T=1.).

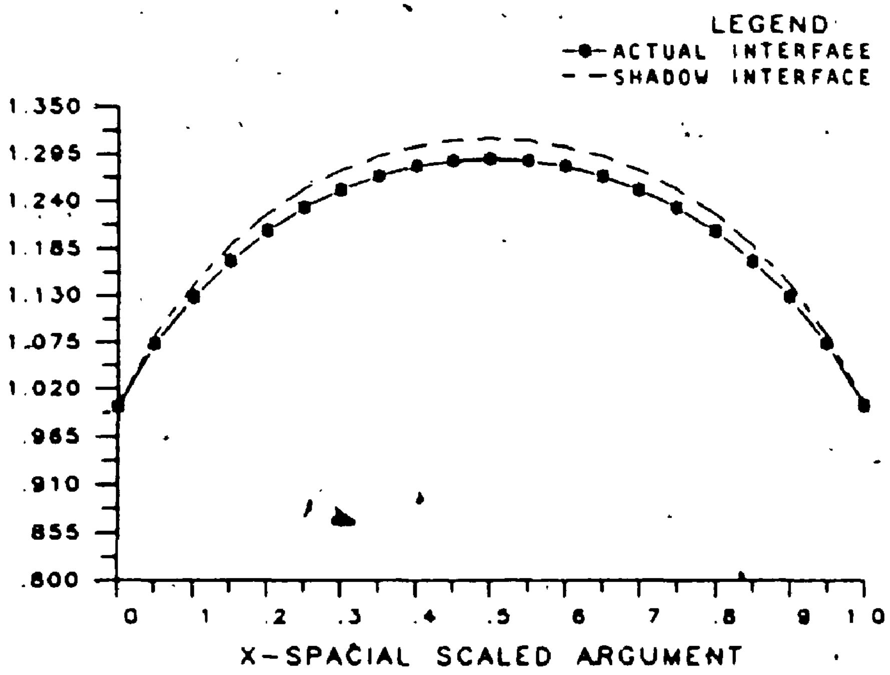


FIGURE 7.4.e Free surface shape when $p_s^*=1.8$, (D=150, T=1.).

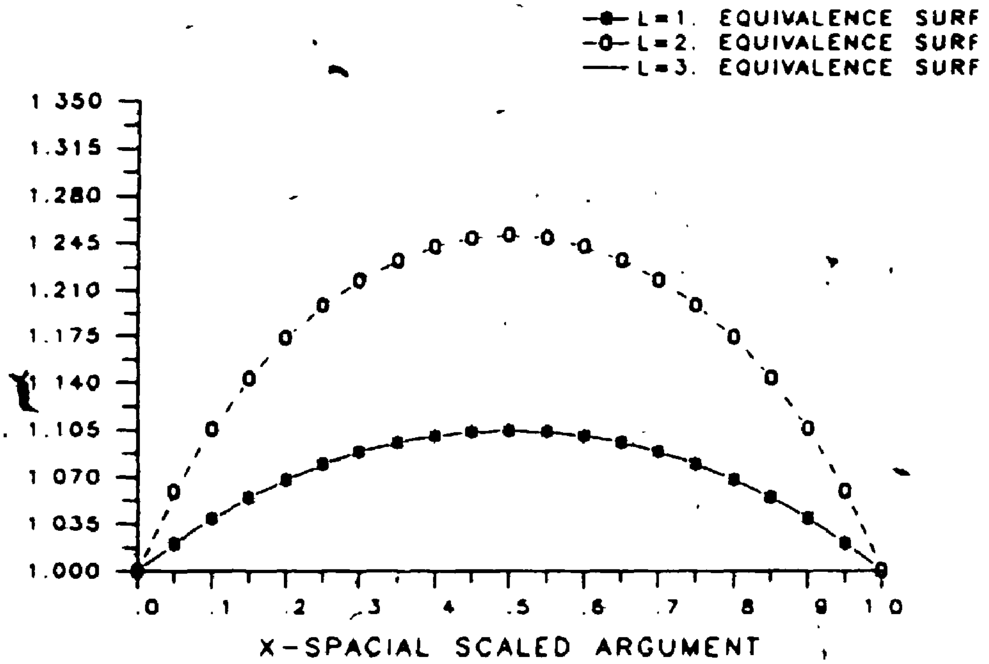


FIGURE .7.5 Free surface shape with variable length scale L=1, 2, 3, (p_S^*=0.8, D=0., T=1.).

7.2 Case of variable external pressure.

If the external pressure along the free surface is not constant, than the interface equation takes on the following form:

$$\frac{h_{xx}}{(1 + h_x^2)^{3/2}} = \frac{-p_s^* L}{T} + \frac{p_{ex}^* L}{T} + \frac{(L \rho_s V^2)|\underline{V}|^2}{2\ T} \qquad (7.4)$$

Despite the fact, that p_{ex}^* is a linear term on the right hand side of (7.4), it certainly has nonlinear effect on the final shape of the free surface. This problem was partially covered in the preceding paragraph, when we discussed the outcome of the variation of the dynamic scale $D=\rho_s V^2/2$ (see point (d)). In two examples which follow (Fig. 7.6 and 7.7) the same external pressure p_{ex}^* =0.1sin(2.[illegible]) is combined with different flows:

(1) : p_s^*=-1., D=12., T=0.8 (Fig. 7.6). The deformation introduced by p_{ex}^* is very small. In fact the stagnation pressure is of one order higher then p_{ex}^*.

(2) : p_s^*=0.1, D=38., T=0.1 (Fig 7.7). The effect of p_{ex}^* ~~is~~ much larger compared to the preceding case. This time p_{ex}^* is of order of p_s^*. In addition, both cases differ in the values of the surface tension T. T is much smaller in case (2). It is certainly affecting dynamic pressure term, and in case (2) the actual interface is far away from the 'shadow' free surface, when p_{ex}^* is 0. The static interfaces for cases (1) and (2) are plotted for comparison on Fig. 7.2.b and 7.1.b.

Fig. 7.8 demonstrates the free surface of the montionless fluid with uneven elevation of the contact points (h(0)=1., h(1)=0.75). The stagnation pressure p_s^* is assumed zero. Since no body forces are present, the static surface is flat. The plot shows the deformation of the static free surface under the action of the external pressure $p_{ex}^*=0.2\sin(2\pi x)$ with decreasing surface tension (T=1., T=0.75).

Fig. 7.9 presents free surface of the field with non-zero flow, with zero and non-zero outer pressure. It can be observed, that decrease of surface tension may cause disruption of the free surface due to large h_x at the right boundary.

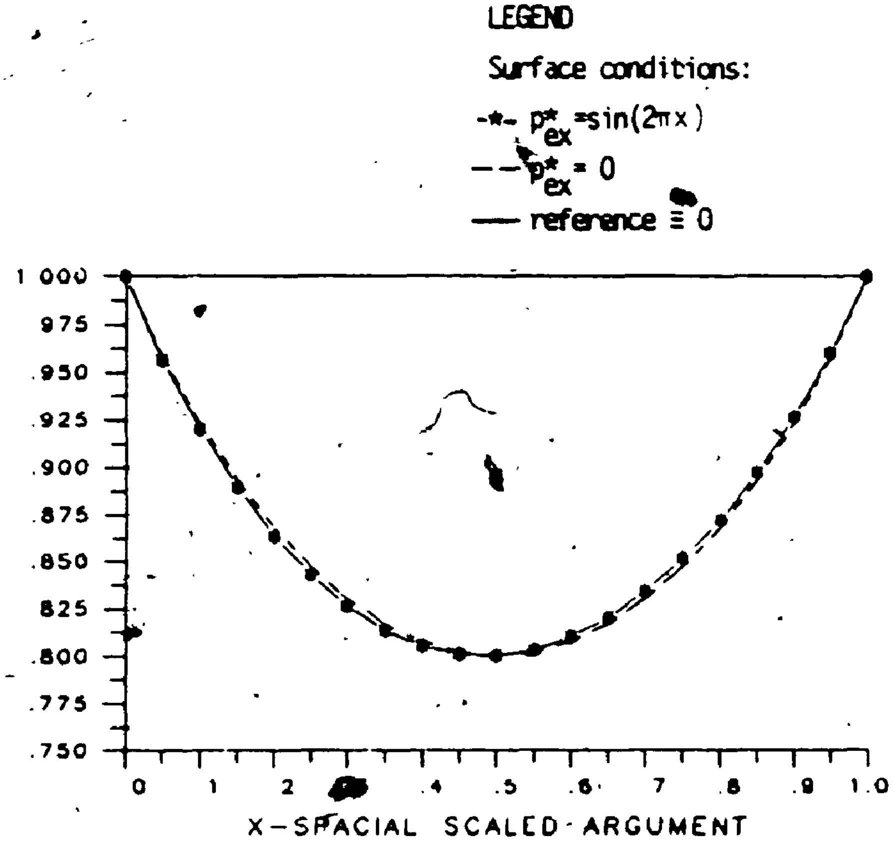
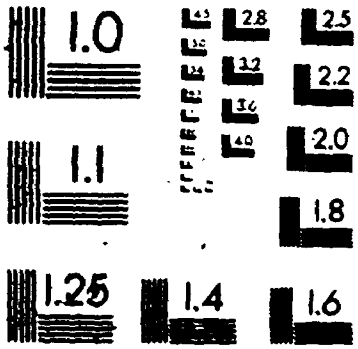


FIGURE 7.6 Free surface when subject to variable external pressure p^*_{ex}=0.1sin(2πx), with other parameters kept constant ($\bar{p}^*_s$=-1., D=12, T=0.8).

4 of/de 4



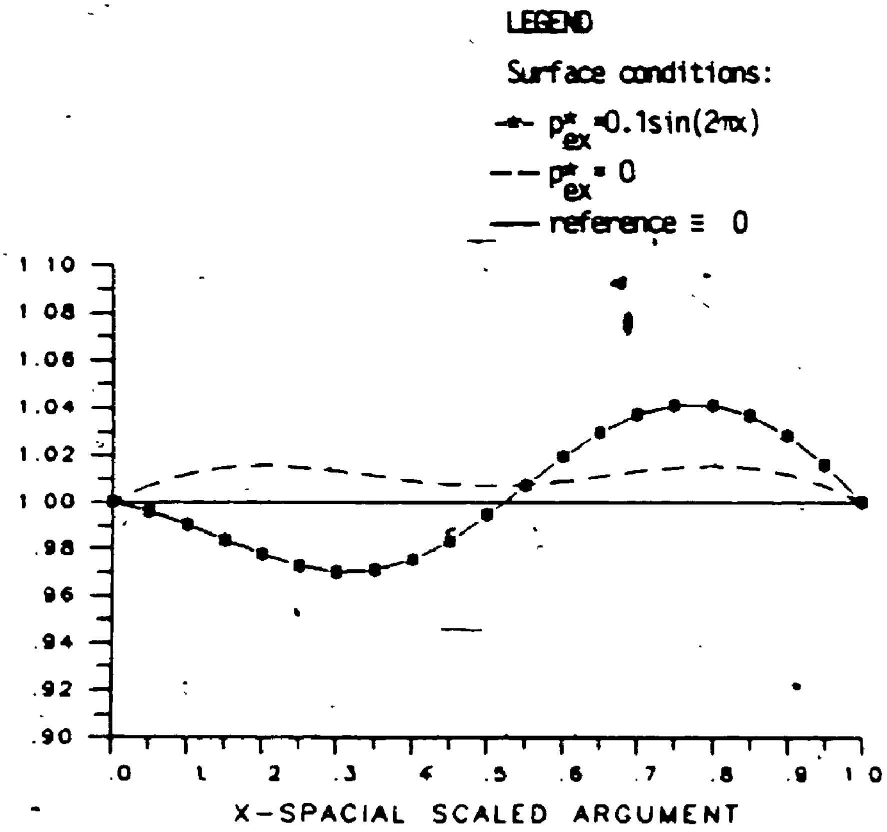


FIGURE 7.7 Free surface for zero and variable external pressure p^*_{ex}=0.1sin(2πx) with other paramenters kept constant (p^*_s=0.1, D=38, T=0.1).

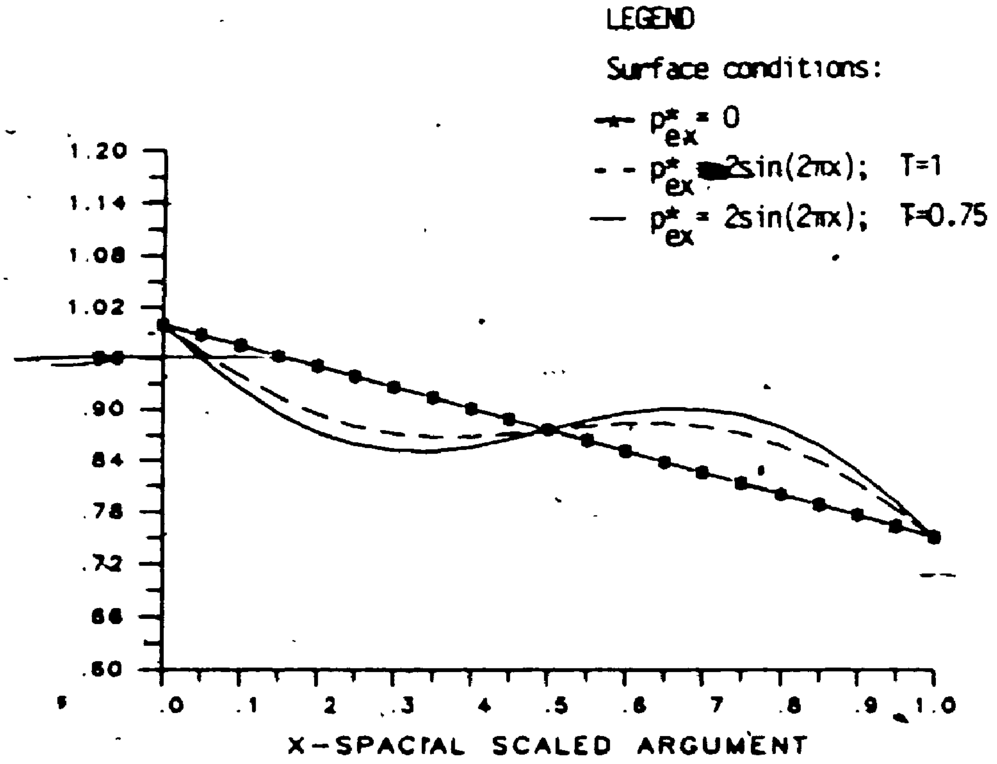


Figure 7.8 Static free surface with variable external pressure with contact points at different elevations, when surface tension decreases (p^*_{ex}=2sin(2πx), p^*_s=0, D=0, T=1., T=0.75).

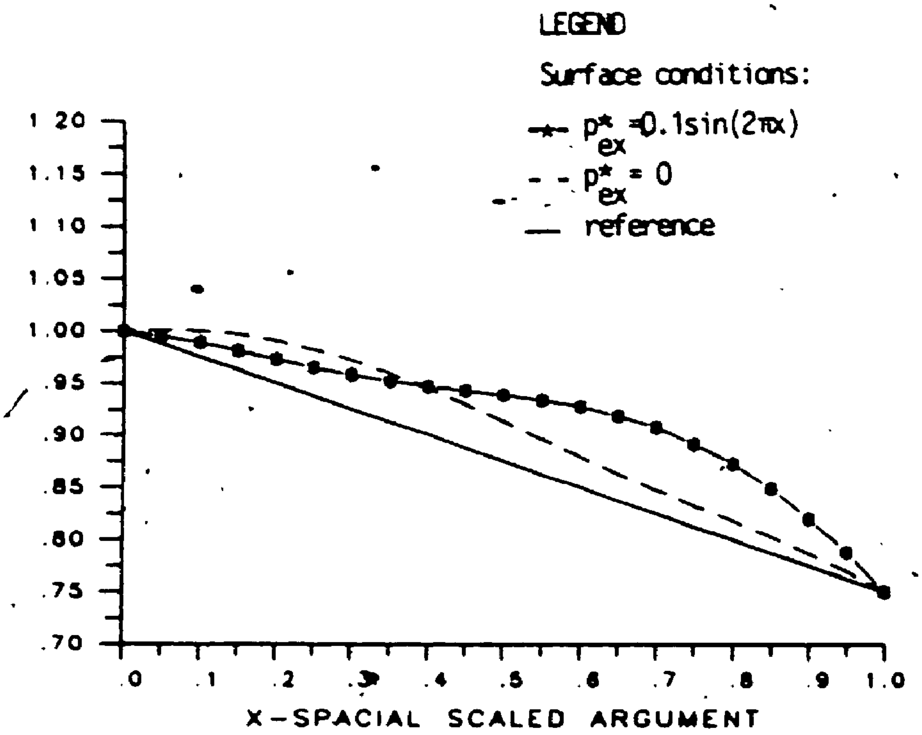


Figure 7.9 Free surface with contact points at different elevations, comparison of effect of constant and variable external pressures: $p^*_{ex}=0$, $p^*_{ex}=0.1\sin(2\pi x)$, ($p^*_s=0.1$, D=16, T=0.1).

CONCLUSIONS

The research presented in the thesis is focussed on developing algorithms for analysis of dynamics of an ideal fluid with free/moving boundaries. Particular attention was paid to the methods of providing high computational efficiency and verification of the accuracy of the numerical solution. The ideal fluid has been chosen because of simplicity of the governing equation; yet, it was complicated enaugh to retain all essential features of a typical moving boundary problem.

A computational model of the governing equation is of iterative character. During the iteration process, the flow domain evolves towards the final form. At each psedo-time level this area is being transformed onto the unit square which constitutes the actual computational domain. The transformation x=x, z = y/h(x) (see Sec. 3.1) was the simplest one to implement. It seems however, that this particular transformation is responsible for the drop of the accuracy of results for flows with the large convex meniscus (see Sec.5.6.5). For the concave meniscus, conversely, the accuracy was improved. Should the boundary fitted orthogonal mapping been applied, this assymetry would probably be removed. The transformation of the coordinate system resulted also in a change of the form of the governing equations, including introduction of mixed derivatives and variable coefficients into the flow equations. The presence of mixed derivatives might pose a problem in higher order differencing approximation, but this was successfully resolved by developing compact planar 4-th order mixed derivative estimates.

Discretization of the analytical model resulted in a system of algebraic equations. It was decided to solve it iteratively using the ADI method. The two types of algorithms were analysed: Asymptotic and Direct. The Asymptotic Algorithm, based on the Small Deformation Theory, is designed to solve flows with expected small dynamic deformation of the static free surface; such situation is expected to occur when surface tension dominates over the dynamic effects. The formal criterion of smallness of deformation was formulated (Sec. 4.3) and constraints on flow parameters were given, so that the class of flows which could be solved through application of the Asymptotic Algorithm was determined.

Applicability of the Direct Algorithms is not restricted to small interface deformation. 'Direct' means that the numerical calculations are carried out jointly for the flow field and interface equations. The nonlinear equation for the free surface is treated with particular attention. In Sec. 5.2 and 5.3 conclusions regarding the existence and uniqueness of the solution to this equation are described. The study of the algebraic stability of the overall ADI scheme (Sec. 4.4 and 5.5.4) supplement these conditions to determine the constraints on applicability of the algorithm. In Sec. 5.5.3 the range of feasible parameters of the model problem is analysed. This provides the information about capabilities and limitations of the algorithm.

There are two Direct Algorithms presented: Picard and 1-Step method. The difference between the two is in the order of linearization and, therefore, this distinction is reflected in iterative effectiveness. The 1-Step algorithm, which results from the 2-nd order linearization of the governing equations is shown to be significantly faster for all physical cases considered. Another contribution to the iterative performance was provided by acceleration of the ADI solving procedure (for all algorithms involved), which was accomplished through optimization of relaxation (following the Wachspress algorithm).

Particular attention was paid to the accuracy of the numerical results. As we discretize the analytical model using finite difference approximation, we focus on the behaviour of the truncation error. Principles of the analysis are presented in Sec. 3.3. It was found that the 2-nd order discretization for the type of flows analysed in this work provided results which have relative grid-convergence of the 2-nd order, as expected. Due to the smoothness of the grid-convergence with the grid refining, the extrapolation of results computed on coarser grids is fully justified. Extrapolation was found to yield highly accurate results. When discretization with higher order differences was used (Chapter 6), the the grid-convergence rate of the results computed on finer and finer grids approached the expected value of 4, but it was not as smooth as in the case of the 2-nd order differencing. Hence extrapolation was found to be unjustified for the grids used in this work. It was shown finally, that it takes much less computer time and memory resources to obtain results of a certain accuracy using second order algorithm with extrapolation, compared to the 4-th order algorithm.

The detailed analysis was carried out for a simple physical model: uniform external pressure and fixed contact points of equal elevation. These restrictions, however, are not relevant, and it is shown in Chapter 7 that we may use the same algorithms to deal with flows with non-uniform external pressure along the free surface, and/or uneven elevation of the contact points.. In the same chapter, a number of theoretical properties of the free surface is derived by analysing the dynamic boundary condition and investigating effects of the physical parameters of the flow. All these results are confirmed by computations and are presented graphically. Among the most interesting features is the existence of the free surface which would not have existed should the flow be halted. In such cases it is the dynamic pressure that contributes to the decrease of the effective static pressure permitting the surface tension to maintain the normal force balance at the free surface.

APPENDIX

FINITE DIFFERENCE IMPLEMENTATION OF COMPACT HIGHER ORDER METHOD

Assume, that the computational domain is covered by a grid of square cells with spatial increments $\Delta x = \Delta z = r$. Let the function to be determined on pivotal points of a grid be denoted by 'u', and its first and second derivative by 'F' and 'S', respectively. For simplicity of notation, we will use single indexation (except for mixed derivative). This will not cause any ambiguity, since in the ADI method, one of coordinate directions (column or row index) is fixed. The discretization scheme is limited to the Dirichlet type boundary conditions.

A-1 Hermitian formulae involving 1-st and 2-nd derivatives

(i) in the interior of the domain (Hirsh [10]):

$$\frac{1}{r}\,(u_{j+1}-u_{j-1}) - 2\left(\frac{1}{6}F_{j+1} + \frac{2}{3}F_j + \frac{1}{6}F_{j-1}\right) = \frac{1}{90}h^4 u^{v} \qquad (A.1)$$

$$\frac{24}{r^2}(u_{j+1}-2u_j+u_{j-1}) - \frac{9}{r}(F_{j+1}-F_{j-1}) +$$

$$+ S_{j+1} - 8S_j + S_{j-1} = \frac{h^6}{2520}u^{viii} \qquad (A.2)$$

(ii) on the boundary (Pade formula):

$$\frac{1}{r^2}(u_{j+1}-u_j) - \frac{1}{2r}(F_{j+1}+F_j) + \frac{1}{12}(S_{j+1}-S_j) = \frac{h^3}{720}u^V \qquad (A.3)$$

A-2 Mixed derivatives.

It is a common practice to derive Hermitian formulae in one dimension. It is understandable that these formulae being linear may be superimposed in order to approximate expressions in the governing partial differential equations. This eventually may lead to separation of all difference expressions and in consequence to the reduction of dimensionality of the algebraic system. The presence of mixed derivatives makes the discretization more involved, though. There are suggestions of utilizing formula (A.1), in which we might set $u=f_x$ and $F=u_z=f_{xz}$ (Aubert and Deville [1]). This however is not a unique formulation and was found by numerical experimentation not to be satisfactory. When mixed derivatives were estimated alternatively by z-differentiation of x-derivatives, followed by x - differentiation of z-derivatives (successive application of (A.1)), the convergence was found to be very slow and oscillatory. The reason for that is probably uneven distribution of respective Lagrange residuals in formulae utilized in x- and z- directions.

To make the mixed derivatives estimation independent of the direction, we have solved the Hermitian formula defined on the entire 9-point compact grid molecule. The results are described below. Derivatives are denoted by the appropriate subscripts and the nodes are denoted by letter symbols shown on the sketch below.

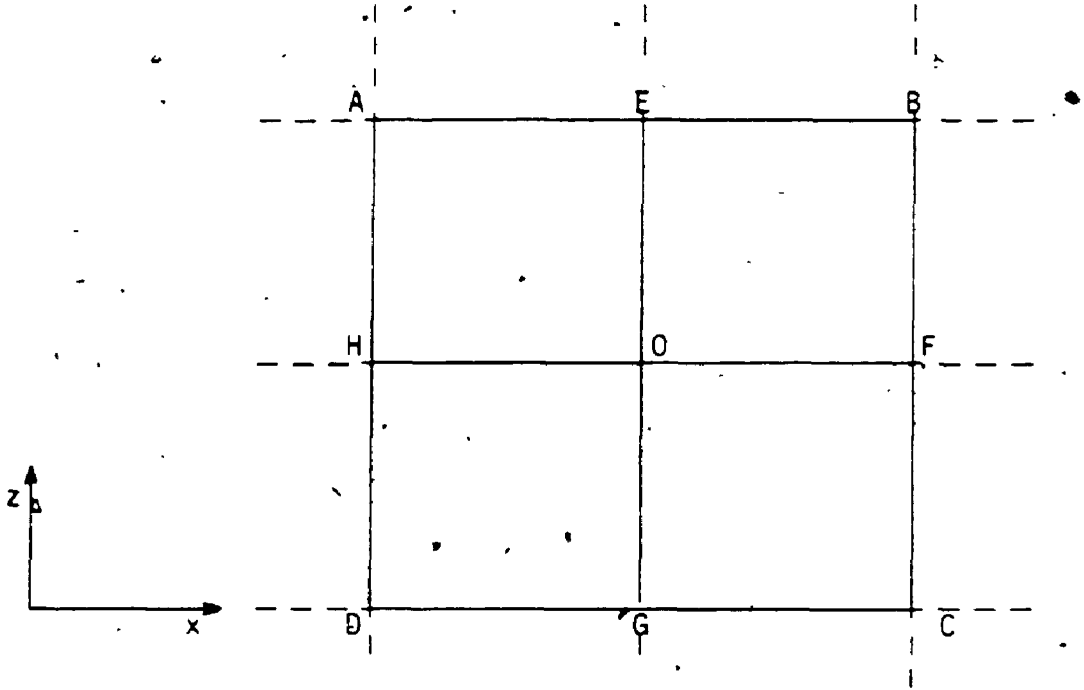


Fig. A1 : 9-point compact planar molecule.

For the interior of the domain we have:

$$\begin{aligned}(u_{xz})_O &- \frac{1}{20r^2}(-u_A + u_B - u_C + u_D) \\ &- \frac{1}{20r}(-u_{xA} - u_{xB} + u_{xC} + u_{xD} + 6u_{xE} - 6u_{xG}) \\ &- \frac{1}{20r}(u_{zA} - u_{zB} - u_{zC} + u_{zD} + 6u_{zF} - 6u_{zH}) \leq O(r^3)\end{aligned} \qquad (A.4)$$

On the left and right boundaries:

$$(u_{xz})_H - \frac{3}{r}(u_{zF} - u_{zH}) + (u_{xz})_F + 4(u_{xz})_O \leq O(r^3) \qquad (A.5.a)$$

$$(u_{xz})_F - \frac{3}{r}(u_{zF} - u_{zH}) + (u_{xz})_H + 4(u_{xz})_O \leq O(r^3) \qquad (A.5.b)$$

On the upper and lower boundaries:

$$(u_{xz})_E - \frac{3}{r}(u_{xG} - u_{xE}) + (u_{xz})_G + 4(u_{xz})_0 \leq 0(r^3) \qquad (A.6.a)$$

$$(u_{xz})_G - \frac{3}{r}(u_{xG} - u_{xE}) + (u_{xz})_E + 4(u_{xz})_0 \leq 0(r^3) \qquad (A.6.b)$$

In particular, (A.5) estimates mixed derivatives if the edges AHD or BFC of the molecule belong to the boundary. (A.6) covers cases when either AEB or DGC are on the boundary. There were also other boundary adjusted estimates of a mixed derivative found, but those listed above have been found to work satisfactorily.

Formulae (A.4), (A.5) and (A.6) were found by solving appropriate Hermitian expressions on the asumption that the resulting difference formula is at least of the order 3 (see Sec. 3.2). It can be checked however, that all three formulae are of the order 4. For clarity we will demonstrate this in detail.

Let us consider the function:

$$u(x,y) = \sin(3x - y^2)$$

Differentiating u in pivotal points of a molecule centered around (x,y)=(4.,4.) and putting appropriate values into (A.3), we get the following sequence of residuals:

grid spacial increment r_i	residual R_i
1/10	1.8E-3
1/20	1.15E-4
1/40	7.2E-6
1/80	4.5E-7

Obviously residuals are going to zero as r is being decreased. If we relate the ratio of this convergence to r, we make use of the relation (compare definition (3.4.4) of the grid-convergence order):

$$\frac{R_i}{R_{i+1}} \propto \frac{r_i^{\alpha}}{r_{i+1}^{\alpha}}$$

Solving it with respect to 'α', we get:

i	α
1	3.97
2	4.0
3	4.0

which demonstrates that (A.3) is of the order 4. The same can be shown for the boundary estimations (A.4) and (A.5).

A-3 Block 3-diagonal structure.

Mixed derivatives are computed point by point, prior to each half-step of ADI. If Hermitian formulae are not used to eliminate any of the variables from the governing equations, and the governing equations contain first and second partial derivatives, than at each grid point we have 3 equations (having ADI direction specified). For the interior point of a grid we have thus generated three blocks (forming the lower, central and upper block-diagonal) of the following structure:

$$\left[\begin{array}{ccc|ccc|ccc} & & & & \vdots & & & & \\ & & & X & X & X & & & \\ -1/r & -1/3 & 0 & 0 & -4/3 & \sigma & 1/r & -1/3 & 0 \\ 24/r & 9/r & 1 & -48/r & 0 & -8 & 24/r & -9/r & 1 \\ & & & & \vdots & & & & \end{array}\right] \circ \begin{bmatrix} \vdots \\ u_{j-1} \\ F_{j-1} \\ S_{j-1} \\ u_j \\ F_j \\ S_j \\ u_{j+1} \\ F_{j+1} \\ S_{j+1} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ X \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \end{bmatrix}$$

First line is filled with terms (X) of the governing equation. Lines second and third come from (A.1) and (A.2).

The block structure for the point at the lower (left) boundary is as follows:

$$\left[\begin{array}{ccc|ccc} X & X & X & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ -1/r & -1/2r & -1/12 & 1/r & -1/2r & 1/12 \\ & & \cdot & & & \\ & & \cdot & & & \\ & & \cdot & & & \\ & & \cdot & & & \\ & & \cdot & & & \\ & & \cdot & & & \end{array}\right] \circ \left[\begin{array}{c} u_1 \\ F_1 \\ S_1 \\ u_2 \\ F_2 \\ S_2 \\ \cdot \\ \cdot \\ \cdot \end{array}\right] = \left[\begin{array}{c} X \\ u_1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{array}\right]$$

and the block structure for the upper (right) boundary:

$$\left[\begin{array}{cccc|ccc} & & & \cdot & & & \\ & & & \cdot & & & \\ & & & \cdot & & & \\ & & & \cdot & & & \\ & & & \cdot & & & \\ & & & \cdot & & & \\ \cdot\;\cdot & .\;0 & 0 & 0 & X & X & X \\ & 0 & 0 & 0 & 1 & 0 & 0 \\ & -1/r & -1/2r & -1/12 & 1/r & -1/2r & 1/12 \end{array}\right] \circ \left[\begin{array}{c} u_1 \\ \cdot \\ \cdot \\ u_{n-1} \\ F_{n-1} \\ S_{n-1} \\ u_n \\ F_n \\ S_n \end{array}\right] = \left[\begin{array}{c} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ X \\ u_n \\ 0 \end{array}\right]$$

First line of the block contains terms (X) of the governing equation, the third one is the Pade boundary formula of the third order (see (A.3)), and the second row comes from the boundary condition imposed by the model. In the above scheme the Dirichlet boundary conditions have been implemented, but it is obvious that the Neuman boundary conditions can also be implemented in a straightforward manner.

REFERENCES

[1] Aubert,X., and Deville,M., 'Steady Viscous Flows by Compact Differences in Boundary Fitted Coordinates' J. Comp. Phys. 49 (1983).

[2] Botta,E.F.F., and Ellenbroeck,M.H.M.,'A Modified SOR Method for Poisson Equation in Unsteady Free Surface Flow Calculations', J.Comp. Physics, 60 (1985).

[3] Cheng,S.I., and Yulin,Lu., 'An Eulerian Method for Transient Nonlinear Free Surface Wave Problems', J. Comp. Phys., 62 (1986).

[4] Collatz,L., The Numerical Treatment of Differential Equations (Springer Verlag 1966).

[5] Dennis,S.C.R., 'Compact Explicite Finite Difference Approximations to the Navier Stokes Equations' Ninth International Conference on Numerical Methods in Fluid Dynamics, Lecture Notes in Physics 218 (Springer Verlag 1985).

[6] Elsaesser,E., and Peyret;R., 'Methodes Hermitiennes pur la Resolution Numerique des Equation de Navier Stokes', Comptes Rendu du Congres International sur les Methodes Numeriques dans les Sciences de l'Ingenieur (Paris 1978).

[7] Floryan,J.M., 'Conformal - Mapping - Based Coordinate Method for Channel Flows', J. Comp. Phys. (1985).

[8] Floryan,J.M., and Rasmussen,H., 'Numerical Methods for Viscous

Flows with Free/Moving Boundaries', to appear.

[9] Hirsh,R.S., 'Higher Order Accurate Difference Solutions of Fluid Mechanics Problems by a Compact Differencing Technique', J. Comp. Phys. 19 (1975).

[10] Hirsh,R.S.,'Higher Order Approximations in Fluid Mechanics; Compact to Spectral', Computational Fluid Mechanics, Von Karman Institute For Fluid Mechanics, Lecture Series 1983-04.

[11] Hurle,D.T.J., and Jakeman,E. 'Introduction to the Techniques of Crystal Growth', PCH, vol.2, 4 (1981).

[12] Hyman,J.M.,'Numerical Methods for Tracking Interfaces, Physica, 12D (1984).

[13] Isaacson,E., and Keller,H.B., Analysis of Numerical Methods (John Wiley & Sons 1966).

[14] Levich,V.G., and Krylov,V.S., 'Surface Tension Driven Phenomena', Adv. in Fluid Mechanics, Vol.1 (1969).

[15] Napolitano,M., and Walters,R.W., 'An Incremental Block - Line - Gauss - Seidel Method for the Navier Stokes Equations' (AIAA 23rd Aerospace Sciences Meeting Jan.14-17 1985).

[16] Ostrach,S.,Convection Phenomena of Importance for Materials Processing in Space' in Material Sciences in Space with Applications to Space Processing, Progress in Astronautics and Aeronautics, Vol. 52 (ed. Steg L.) AIAA (1977).

[17] Peyret,R., and Taylor,Th.D.,Computaional Methods for Fluid Flow (Springer Verlag 1983).

[18] Rasmussen,H., and Salhani,D.S., 'Unsteady Porous Flow with a

Free Surface', IMA Journal of Applied Math., 27 (1981).

[19] Richtmyer,R.D., and Morton,K.W., Difference Methods for Initial Value Problems (John Wiley & Sons 1967)

[20] Rybicki,A., Floryan,J.M, 'Fluid Dynamical Aspects of the Floating Zone Process', Proceedings of the Tenth Canadian Congress of Applied Mechanics (1985).

[21] Rybicki,A., and Floryan,J.M., 'Thermocapillary Convection in Liquid Bridges', Bull. Amer. Phys. Soc., 30 (1985).

[22] Ryskin,G., 'Orthogonal Mapping', J. Comp. Phys. 50 (1983).

[23] Ryskin,G, and Leal,L.G.,'Numerical Solution of Free Boundary Problems in Fluid Mechanics. Part 1: The Finite Difference Technique', J. Fluid Mech. vol.148 (1984)

[24] Ryskin,G, and Leal,L.G.,'Numerical Solution of Free Boundary Problems in Fluid Mechanics. Part 2: Boyancy - Driven Motion of a Gas Bubble through a Quiescent Liquid', J. Fluid Mech. vol.148 (1984)

[25] Scriven,L.I., 'Drops and Bubbles: Their Science and the Systems They Model', Proc. International Colloquium on Drops and Bubbles, (edits. D.J.Collins, M.S.Plesset, M.M.Saffren), California Institute of Technology and Jet Propulsion Lab., (August 1974).

[26] Strani,M., and Piva,R.,'Computational Models for Convective Motions Induced at Fluid Interfaces', Proc. 7 Int. Conference on Numerical Methods in Fluid Dynamics (Stanford 1980).

[27] Smith,G.D.m Numerical Solution of Partial Differential Equa-

tions: Finite Difference Methods (Oxford Univristy Press 1978).

[28] Wachspress,E.L., Iterative Solution of Elliptic Systems and Applications to the Neutron Diffusion Equations of Reactor Physics (Prentice Hall 1966).