AN ABSTRACT OF THE THESIS OF

Jörg Kuhnert for the degree of Master of Science in Mathematics presented on

November 30, 1995 . Title: Numerical Computation of Free Surface Water Flow .

Abstract approved: Redacted for privacy

Ronald B. Guenther

The present thesis is a report of investigations on a model for computing the water flow in the wave tank of the OSU Wave Laboratory.

A wave board, established at one end of the 100 m long, 4 m wide and 6 m deep wave tank, produces water waves by periodically moving back and forth. The resulting free surface water flow will be modeled physically. This physical model will be transformed into a mathematical model which is implemented on a computer. The goal is to approximately compute the velocity distribution of the fluid modeled as a linear combination of basis functions. Certain well known mathematical tools such as the Galerkin method or the idea of finite elements are employed.

The theoretical developments are done for the three-dimensional case. However, numerical experiments are performed in two dimensions only.

Two different types of basis functions will be introduced and investigated. Their advantages, disadvantages and capability to compute realistic results will clearly be shown and demonstrated in practice.

Several problems occured during the development of the computation algorithms. These problems are outlined in this thesis, and several solutions are suggested. For example, a method how to deal with rising and falling water surface at a vertical boundary will be presented.

The results of five numerical test runs will be exhibited, underlining what the theory predicts.

Numerical Computation of Free Surface Water Flow

by

Jörg Kuhnert

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Completed November 30, 1995
Commencement June 1996

Master of Science thesis of Jörg Kuhnert presented on November 30, 1995

APPROVED:

Redacted for privacy

Major Professor, representing Mathematics

Redacted for privacy

Chair of Department of Mathematics

Redacted for privacy

Dean of Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Redacted for privacy

Jörg Kuhnert, Author

ACKNOWLEDGMENT

Hereby I would like to thank Professor Guenther, who was not only my advisor in this research project, but also an excellent guide during the time I spent at Oregon State University. I appreciate the close cooperation with Dr. Guenther which I consider to be of great personal benefit.

Moreover, I would like to thank all professors, who taught classes I took. I am deeply impressed by their excellent performance and patient work. Several of the ideas presented in their lectures are incorporated in this thesis.

Thanks to the team of the Mathematics Department Office. They do a great job!

Last but not least I would like to emphasize the great work of Katherine Socha (who checked the whole thesis and made the thesis sound English), of Alisa Hunt, Robert Quelette and Kinga Farkas (who all played a necessary part in combining the mathematical ideas of this thesis with a proper English).

TABLE OF CONTENTS

Appendix: Disk

LIST OF FIGURES

Numerical Computation of Free Surface Water Flow

Chapter 1: Introduction

The present thesis deals with the research that I have undertaken during my enrollment as a graduate student in the Department of Mathematics at Oregon State University.

Based on already existing and well known physical models in fluid mechanics, the topic of my research was to develop and investigate a mathematical model and a related numerical method to solve for the velocity distribution of a fluid (water) in one of the wave tanks of the OSU wave laboratory. Thus, the aim was to simulate the motion of the water in the tank.

A wave tank is made to simulate natural processes that deal with water waves in order to make measurements and thereby to understand what processes accompany the habitual appearance of a wave.

Computing the velocity field of a fluid flow is a very old, as well as a very difficult, topic. Whenever a fluid is flowing in any application (cars, airplanes, medical instruments, etc.), engineers are immediately interested in knowing the velocity and pressure distributions, since these are necessaray for calculating stresses arising from the fluid acting on a solid body as well as for computations concerning the 'energyflux' related to the fluid flow. Stresses and energy have a huge informative value for engineers who construct new machines and instruments, and it is finally the stresses and the flow of energy that must be known before a new application can be set in motion.

Further, the simulation of the velocity distribution in the wave tank, is a means of calculating the stresses acting from the water on the ground of the wave tank and thereby comprehending the water-ground interaction. This interaction takes place in nature (at ocean or river shores) at every second of earth's life and for ages people have tried to change this interaction to their advantage by constructing dams, wave breakers etc. On one hand, people are interested in knowing effects of the changes prior to establishing them. On the other hand, people would like to know the basic processes taking place when water and ground interact. Thus we recognize many practical applications for the simulation of the waterflow in the wave tank. This was, indeed, a very good motivation to make any effort in approaching the final aim.

The wave tank I dealt with was about 100 meters long and 4 meters wide. Its depth was approximately 6 meters. The ground of the wave tank can be adapted to the particular shapes researchers like to investigate. At one end of the wave tank there is a wave board that generates water waves by moving back and forth. The wave board can be controlled such that different kinds of waves can be produced.

The following picture gives a little sketch of the wave tank.

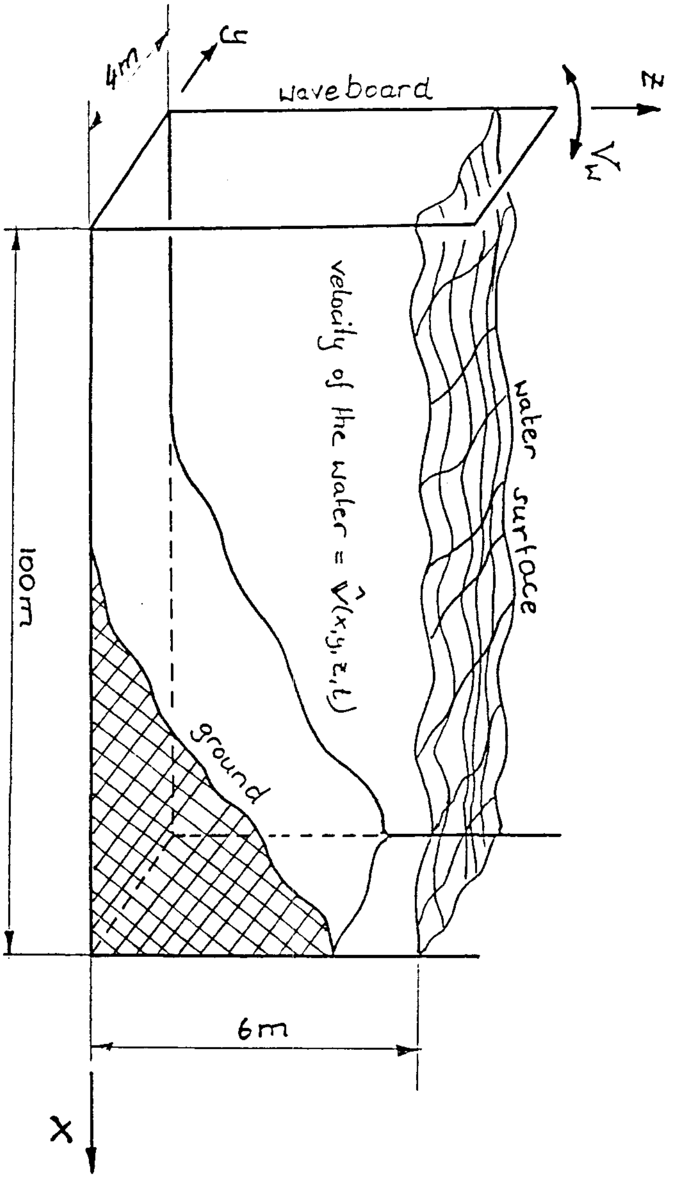


figure 1.1: Sketch of the wave tank in three dimensions

The task for my thesis can be formulated simply.
Suppose we know the time dependent motion of the wave board. Moreover, suppose we know the velocity distribution of the water at an initial point of time (namely exactly at the time when we start our observations). Then the goal is to approximately compute the velocity distribution over all the water in the wave tank and the motion of the surface at any time after the wave board started to move.

In order to solve this problem, we have to deal with the physical model for incompressible free surface fluid flow which we will introduce in chapter 2. A brief look over this chapter shows that we need to consider a set of three partial differential equations representing the law of conservation of mass in a moving fluid, the law of conservation of momentum acting on a fluid particle (Navier-Stokes equations) and the formulation of the surface motion.

Our first approach to solving these differential equations was to employ a finite difference method. The idea of finite differences consists in approximating the partial derivatives belonging to a differential equation by finite differences of the form

$$\begin{aligned}
\frac{\partial f}{\partial x}(x_i, y, z) &\approx \frac{f(x_{i+1}, y, z) - f(x_{i-1}, y, z)}{x_{i+1} - x_{i-1}}, \\
\frac{\partial f}{\partial y}(x, y_j, z) &\approx \frac{f(x, y_{j+1}, z) - f(x, y_{j-1}, z)}{y_{j+1} - y_{j-1}}, \\
\frac{\partial f}{\partial z}(x, y, z_k) &\approx \frac{f(x, y, z_{k+1}) - f(x, y, z_{k-1})}{z_{k+1} - z_{k-1}}.
\end{aligned}$$

$f(x, y, z)$ is any function defined in the region where the partial differential equations are defined. Applying finite differences requires establishing a grid of discrete points with possibly small distances to the neighbor points for good approximation of the

derivatives. In order to apply a finite difference scheme, we studied the very educational doctoral thesis of Dr. Wilhelm Heinrichs [5] on spectral multigrid methods for the Navier-Stokes equations. We thank Dr. Heinrichs for immediately supplying his thesis to support our research.
Applying a difference scheme requires transforming and normalizing the region of the fluid flow into a rectangular box and rewriting all differential equations for the transformed region [2]. For our problem, the boundaries of the region of the fluid flow are the very curvy free surface and the ground as well as the vertical walls. Thus, the transformation to be performed considerbly complicates our differential equations since one of the normalization parameters is the height of the surface itself. However, this height is not known in advance and changes as time increases. Therefore, the transformation results in highly nonlinear equations. Unfortunately, Dr. Heinrichs' work does not contain ideas how to deal with problems in free surface flow.

Furthermore, we reviewed the literature dealing with fluid dynamics. Thereby, we made three observations.

First : Papers about research on numerical methods for three dimensional flow involving the Navier-Stokes equations are hardly available. A relatively rich source for this topic are the journals 'Computer methods in applied mechanics and engineering' and 'Computers and fluids'. Unfortunately, numerical computation of free surface flow by solving the Navier-Stokes equations was not found.

Second : A very important part of literature on numerical methods for free surface flow is to be found in the oceanography section. There are interesting papers about three dimensional free surface flow, for instance the collection of papers on ship hydrodynamics, water waves, and asymtotics of Fritz Ursell [14] or the book about ocean wave modelling by the SWAMP Group [12]. We notice that free surface flow in oceanography is considered as potential flow. That is one assumes that the fluid flow has zero vorticity. This implies that the fluid flow is non viscous. The differential equations simplify considerably and become linear. For example engineers in ship building employ the concept of potential flow for the calculation of the ship's wave resistance [6]. However, our intention in the present problem is computing the stresses acting on solid bodies caused by the fluid motion. Hence, we cannot assume a non-viscous (potential) flow. So, the work on potential flow is, unfortunately, not helpful for us.

Third : There is some literature in oceanography where even stresses are involved in the mathematical considerations. This is done by dealing with mean values of stresses over the length of one wave as for instance in Paul Martinez' and John Harbaugh's book on simulation of nearshore environments [8]. There exist already software packages for computing the wave propagation involving mean values of stresses acting on the free water surface and the ground. We may not employ these algorithms for our problem since we require a high resolution of our results.
Other papers on numerical methods in free surface flow, for instance [7], employ the shallow water model. This model is based on neglecting velocity changes in vertical direction. Again, we are unable to employ this idea since our wave tank is very deep compared to the length of the waves, so we clearly have no shallow water.

This whole situation was the motivation for us to come up with completely new ideas. With our methods, we attempt to compute three dimensional free surface water flow by numerically solving the Navier-Stokes equations. As the literature research shows, computations of this kind were not performed before, so we can assume that our numerical program represents a new quality in computational fluid mechanics.

In the following we introduce some symbols that are used most often in this thesis:

$\hat{\mathbf{v}}(x, y, z, t)$ or simply $\hat{\mathbf{v}}(t)$ ... a vector indicating the velocity field of the fluid throughout the wave tank

$\mathbf{v}(x, y, z, t)$ or simply $\mathbf{v}(t)$ ... a vector indicating an approximation to $\hat{\mathbf{v}}(t)$ of the form

$$\hat{\mathbf{v}}(t) \approx \mathbf{v}(t) = V + \sum_{i=1}^{N} b_i(t) \cdot \mathbf{e}^{(i)}$$

$V = (V_x, V_y, V_z)^T$ ... a function satisfying certain conditions at the boundary of the wave tank $\partial\Omega$

$\mathbf{e}^{(i)} = (\mathbf{e}_x^{(i)}, \mathbf{e}_y^{(i)}, \mathbf{e}_z^{(i)})^T$ ... a set of linearly independent basis functions for $i = 1 \ldots N$ having the property $\mathbf{e}^{(i)} \equiv 0$ at the boundary $\partial\Omega$

N ... number of basis functions (also number of weight functions) used

$\mathbf{w}^{(k)} = (\mathbf{w}_x^{(k)}, \mathbf{w}_y^{(k)}, \mathbf{w}_z^{(k)})^T$ ... a set of linearly independent weight functions for $k = 1 \ldots N$ having the property $\mathbf{w}^{(k)} \equiv 0$ at the boundary $\partial\Omega$

ρ ... the density of the fluid (in our case water)

g ... the gravity constant

U_0 ... the generalized velocity

T ... the generalized time

L ... the length of the wave tank

H ... the global height of the wave tank

$A = \frac{L}{2H}$ ... one half the ratio of length to height

$\xi = \frac{x}{L}$ ... the nondimensionalized component of length pointing in x-direction

$\eta = \frac{z}{H}$ ... the nondimensionalized component of length pointing in z-direction

$Gr(x)$ ... a function that represents the height of the ground of the wave tank above the point $z = 0$

$gr(x) = \frac{Gr(x)}{H}$ ... the nondimensionalized ground function

ν ... the viscosity of the fluid

$Re = \frac{L \cdot U_0}{\nu}$ ... the Reynolds number

$St = \frac{L}{U_0 \cdot T}$ ... the Strouhal number

$Fr = \frac{L \cdot g}{U_0^2}$ ... the Froude number

$V_W(y, z, t)$ ... the velocity describing the motion of the wave board

$V_0(t)$ ... the velocity of the wave board in the height H

I_{gr} ... an interval [-1,a] on the $\xi - axis$ where special conditions are satisfied

I ... the identity matrix

$\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$ or $(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3})$

$\chi I_{E^{(i)}}$ ... characteristic function of the interval $I_{E^{(i)}}$ where the function $E^{(i)}$ is non-zero

Chapter 2: Physical Model and Mathematically Modelling the Problem

2.1 Physical Model and Boundary Conditions

The present problem of fluid mechanics (as explained in Chapter 1) is physically uniquely described by the Navier-Stokes-equations (law of conversation of momentum) and the continuity equation (law of conservation of mass) and the conditions at the boundary $\partial\Omega$ which must be known at every time. These three criteria are called the physical model of the problem and have the following forms:

1.) Navier-Stokes-equations:

$$\frac{\partial \hat{\mathbf{v}}}{\partial t} + (\hat{\mathbf{v}} \cdot \nabla) \cdot \hat{\mathbf{v}} = \nabla \cdot \hat{\mathbf{A}} \tag{2.1}$$

where $\hat{\mathbf{A}}$ is a tensor due to pressure, viscosity and gravity forces:

$$\hat{\mathbf{A}} = \begin{pmatrix} -\frac{p}{g} + 2\nu\frac{\partial \hat{u}}{\partial x} - \rho z & \nu(\frac{\partial \hat{u}}{\partial y} + \frac{\partial \hat{v}}{\partial x}) & \nu(\frac{\partial \hat{w}}{\partial x} + \frac{\partial \hat{u}}{\partial z}) \\ \nu(\frac{\partial \hat{u}}{\partial y} + \frac{\partial \hat{v}}{\partial x}) & -\frac{p}{g} + 2\nu\frac{\partial \hat{v}}{\partial y} - \rho z & \nu(\frac{\partial \hat{v}}{\partial z} + \frac{\partial \hat{w}}{\partial y}) \\ \nu(\frac{\partial \hat{w}}{\partial x} + \frac{\partial \hat{u}}{\partial z}) & \nu(\frac{\partial \hat{v}}{\partial z} + \frac{\partial \hat{w}}{\partial y}) & -\frac{p}{g} + 2\nu\frac{\partial \hat{w}}{\partial z} - \rho z \end{pmatrix}$$

Remark: The tensor $\hat{\mathbf{A}}$ is valid in this form only if the gravity points exactly in negative z-direction (which is satified in our case).

2.) Continuity equation:

$$\nabla \cdot \hat{\mathrm{v}} = 0 \quad . \tag{2.2}$$

The equation in this form is valid only for incompressible fluids (which is satisfied in our case) and must be satisfied throughout Ω.

3.) Boundary conditions:

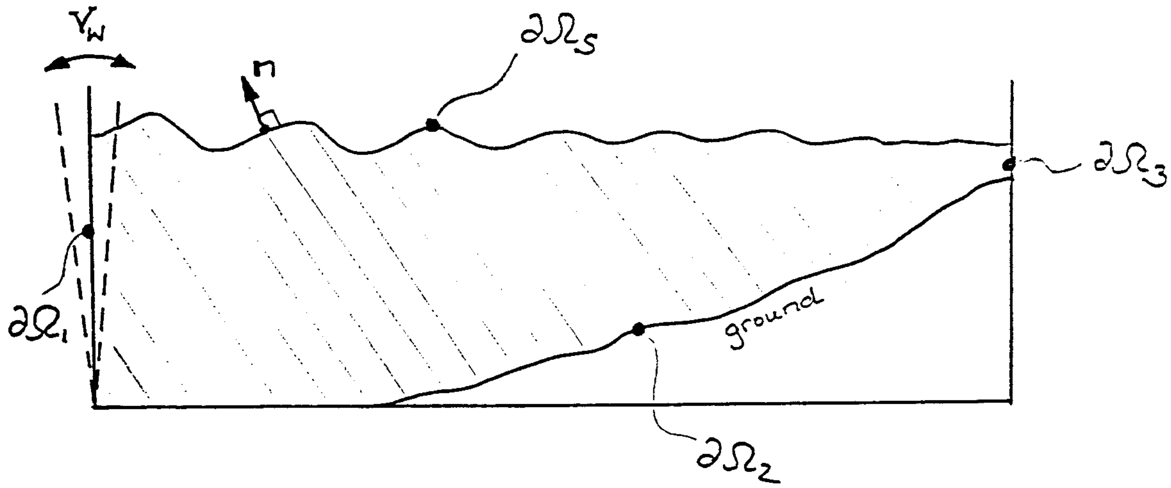


figure 2.1: Sketch of the wave tank in two dimensions

The above picture is a sketch of the wave tank in two dimensions where the boundaries are given special symbols:
$\partial\Omega_s$ denotes the surface,
$\partial\Omega_1$ denotes the vertical (left) wall were the wave board is located,
$\partial\Omega_2$ denotes the ground of the wave tank,
$\partial\Omega_3$ denotes the right end of the wave tank.

Consider $\partial\Omega_s$:
To determine a boundary condition for the surface, one needs to know the forces that act from the medium above the water (air) on the water surface. These are mainly dependent on the state of motion of the air. A means of considering the air's action on the surface is to consider it with respect to the normal to the surface line.
If there are no forces acting on the surface (i.e. there is no significant wind) then at least the condition

$$\hat{\mathbf{T}} \cdot \mathbf{n} = \mathbf{0} \tag{2.3}$$

must hold at $\partial\Omega_s$, where $\hat{\mathbf{T}}$ is the so called stress tensor and is given by

$$\hat{\mathbf{T}} = \begin{pmatrix} -\frac{p}{g} + 2\nu\frac{\partial\hat{u}}{\partial x} & \nu(\frac{\partial\hat{u}}{\partial y} + \frac{\partial\hat{v}}{\partial x}) & \nu(\frac{\partial\hat{w}}{\partial x} + \frac{\partial\hat{u}}{\partial z}) \\ \nu(\frac{\partial\hat{u}}{\partial y} + \frac{\partial\hat{v}}{\partial x}) & -\frac{p}{g} + 2\nu\frac{\partial\hat{v}}{\partial y} & \nu(\frac{\partial\hat{v}}{\partial z} + \frac{\partial\hat{w}}{\partial y}) \\ \nu(\frac{\partial\hat{w}}{\partial x} + \frac{\partial\hat{u}}{\partial z}) & \nu(\frac{\partial\hat{v}}{\partial z} + \frac{\partial\hat{w}}{\partial y}) & -\frac{p}{g} + 2\nu\frac{\partial\hat{w}}{\partial z} \end{pmatrix} .$$

If we take into account the wind acting on the surface then $\hat{\mathbf{T}} \cdot \mathbf{n} \neq \mathbf{0}$!

In this case we have

$$\hat{\mathbf{T}} \cdot \mathbf{n} = \mathbf{s}(x, y) \tag{2.4}$$

where $\mathbf{s}(x, y)$ must be known at every time and could be the result of preceding fluid dynamical computations concerning the air flow adjacent to the surface.

Consider $\partial\Omega_1$:
The fluid flow at $\partial\Omega_1$ is defined by the motion of the wave board. Let $V_W(y, z, t)$ be the velocity of the wave board; then,

$$\hat{\mathbf{v}} = V_W(y, z, t) \qquad \textit{throughout} \;\; \partial\Omega_1 \;\; . \tag{2.5}$$

The fluid cannot have any velocity different from $V_W(y, z, t)$ since the Stokes no slip condition forces the fluid to stick on the board.

Consider $\partial\Omega_2$:
Again, the Stokes no slip condition requires that the fluid sticks on the ground. So, the equation

$$\hat{\mathbf{v}} = \mathbf{0} \tag{2.6}$$

must be satisfied everywhere at $\partial\Omega_2$.

consider $\partial\Omega_3$:
Again, the Stokes no slip condition requires that the fluid sticks on the right wall. So, the equation

$$\hat{\mathbf{v}} = \mathbf{0} \tag{2.7}$$

must be satisfied everywhere at $\partial\Omega_3$.

2.2 Differential Equation describing the Motion of the Surface

The differential equation that decribes the motion of the surface results from the assumption that the surface moves with the same velocity as the set of particles being instantly on the free surface.

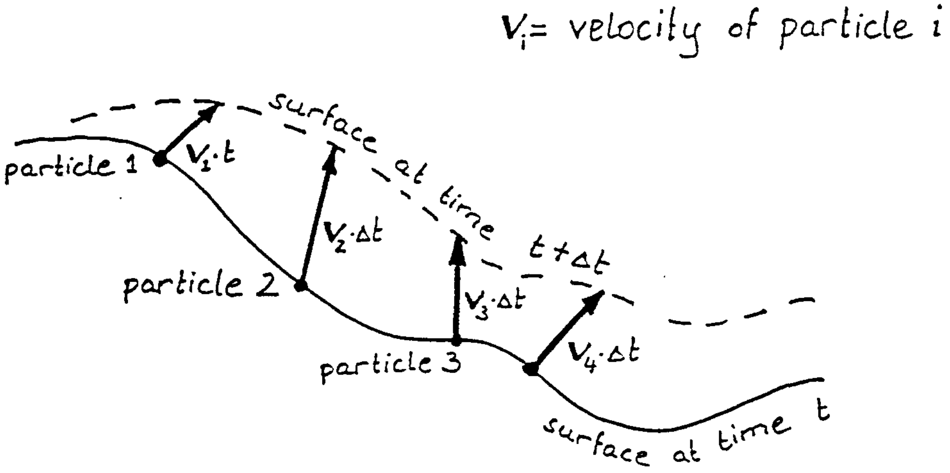


figure 2.2: Model of the surface motion

So, we have the differential equation

$$\frac{d\mathbf{s}}{dt} = \hat{\mathrm{v}}_{(particle)} \tag{2.8}$$

where $\mathbf{s} = (\mathrm{s}_x, \mathrm{s}_y, h)^T$ represents a point on the surface.

Equation 2.8 actually consists of the three differential equations

$$\frac{ds_x}{dt} = \hat{u} \ , \tag{2.9}$$

$$\frac{ds_y}{dt} = \hat{v} \ , \tag{2.10}$$

$$\frac{dh}{dt} = \hat{w} \ . \tag{2.11}$$

For numerically solving for the surface motion, the equation (2.11) is the most important: if we consider the height of the surface h as a function in x and y, then equation (2.11) gives us the partial differential equation

$$\frac{\partial h}{\partial t} + \hat{u}\frac{\partial h}{\partial x} + \hat{v}\frac{\partial h}{\partial y} = \hat{w} \ . \tag{2.12}$$

This is the primary differential equation that we used in our numerical applications for computing the motion of the surface.

2.3 Transformation into a Mathematical Model

2.3.1 Transforming Navier Stokes equations and Continuity equations

The physical model as introduced in the previous sections is not ready yet for direct use to compute the velocity distribution in the tank. In the following, we shall, based on the physical model, develop a mathematical model that allows us to compute approximations for the velocity distribution.

Ansatzfunction and appropriate requirements

We do not attempt to compute the exact function $\hat{\mathbf{v}}(t)$; rather we try to find an approximation $\mathbf{v}(t)$ in the sense that

$$\hat{\mathbf{v}}(x,y,z,t) \approx \mathbf{v}(x,y,z,t) = V(x,y,z,t) \ + \sum_{i=1}^{N} b_i(t) \cdot \mathbf{e}^{(i)}(x,y,z) \tag{2.13}$$

where

1.) $\mathbf{e}^{(1)} \ldots \mathbf{e}^{(N)}$ is a set of linearly independent basis functions. Each of the basis functions has to satisfy the conditions

$$\nabla \cdot \mathbf{e}^{(i)} = 0 \qquad \textit{throughout } \Omega \tag{2.14}$$

$$\mathbf{e}^{(i)} = 0 \qquad \textit{at } \partial\Omega_1 \ , \ \partial\Omega_2 \ , \ \partial\Omega_3 \tag{2.15}$$

2.) $V(x, y, z, t)$ has to satisfy the condition

$$\nabla \cdot V = 0 \tag{2.16}$$

and

V must satisfy the explicitly given boundary conditions at $\partial\Omega_1$, $\partial\Omega_2$ and $\partial\Omega_3$ as they were stated in section 2.1 , (2.17)

3.) $b_i(t)$ are factors that need to be computed by the numerical method.

At this point we realize that, no matter which values the b_i's take on, any function of the form $\mathbf{v}(t) = V(x, y, z, t) + \sum_{i=1}^{N} b_i(t) \cdot \mathbf{e}^{(i)}(x, y, z)$ (Ansatzfunction (2.13)) already satisfies $\nabla \cdot \mathbf{v} = 0$ (condition (2.2)) as well as the boundary conditions at $\partial\Omega_1$, $\partial\Omega_2$ and $\partial\Omega_3$. However, the boundary conditions at $\partial\Omega_s$ may not be satisfied immediately. Please refer to the next section where the conditions at $\partial\Omega_s$ are involved in the computations.

Using Galerkins Method for evaluating the numbers b_i

Since continuity and the conditions at $\partial\Omega_1$, $\partial\Omega_2$ and $\partial\Omega_3$ are already satisfied by cleverly choosing the Ansatzfunction, the remaining task is to use the Navier Stokes equations to evaluate the b_i's in the Ansatzfunction (2.13) as well as to take care of the conditions at the surface.

We will use Galerkins method to obtain an expression for the b_i's and we shall see that, if we use a weak formulation, the Ansatzfunction obeys the boundary conditions at $\partial\Omega_s$.

Galerkins method, applied to our problem:
Suppose there is a set of N linearly independent weight functions $\mathbf{w}^{(k)}$ with $k = 1...N$, and lets furthermore require that these weight functions satisfy the conditions

$$\nabla \cdot \mathbf{w}^{(k)} = 0 \qquad \textit{throughout } \Omega \ , \tag{2.18}$$

$$\mathbf{w}^{(k)} = 0 \qquad \textit{at } \partial\Omega_1 \ , \ \partial\Omega_2 \ , \ \partial\Omega_3 \ . \tag{2.19}$$

Galerkins method then states that the Navier Stokes equations need not be exactly satisfied for $\mathbf{v}$, but that the equation

$$\int_\Omega \mathbf{w}^{(k)T} \cdot (\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \cdot \mathbf{v})\, d\Omega = \int_\Omega \mathbf{w}^{(k)T} \cdot (\nabla \cdot \mathbf{A})\, d\Omega \tag{2.20}$$

is true for $k = 1...N$.

We should note that the matrix $\mathbf{A}$ has the same meaning as matrix $\hat{\mathbf{A}}$, but $\mathbf{A}$ is computed using the approximative velocity $\mathbf{v}$ instead of $\hat{\mathbf{v}}$.

All further modifications of equation (2.20) are performed in order to obtain an expression for $\mathbf{v}$ (i.e. an expression for the b_i's in Ansatz (2.13)).

We use the identity

$$\nabla \cdot (\mathbf{A}^T \cdot \mathbf{w}) = \mathbf{w}^T \cdot (\nabla \cdot \mathbf{A}) + \sum_{i,j} \mathbf{A}_{ij} \frac{\partial \mathbf{w}_i}{\partial x_j} \tag{2.21}$$

to rewrite (2.20) as follows :

$$\int_\Omega \mathbf{w}^{(k)^T} \cdot \frac{\partial \mathbf{v}}{\partial t}\, d\Omega + \int_\Omega \mathbf{w}^{(k)^T} \cdot ((\mathbf{v} \cdot \nabla) \cdot \mathbf{v})\, d\Omega = \int_\Omega \nabla \cdot (\mathbf{A}^T \cdot \mathbf{w}^{(k)})\, d\Omega - \int_\Omega (\sum_{i,j} \mathbf{A}_{ij} \frac{\partial \mathbf{w}_i^{(k)}}{\partial x_j})\, d\Omega \tag{2.22}$$

Now, using Gauss' divergence theorem, from (2.22) it follows that

$$\int_\Omega \mathbf{w}^{(k)^T} \cdot \frac{\partial \mathbf{v}}{\partial t}\, d\Omega + \int_\Omega \mathbf{w}^{(k)^T} \cdot ((\mathbf{v} \cdot \nabla) \cdot \mathbf{v})\, d\Omega = \int_{\partial\Omega} \mathbf{w}^{(k)^T} \cdot (\mathbf{A} \cdot \mathbf{n})\, d(\partial\Omega) - \int_\Omega (\sum_{i,j} \mathbf{A}_{ij} \frac{\partial \mathbf{w}_i^{(k)}}{\partial x_j})\, d\Omega \tag{2.23}$$

Furthermore using the fact that $\mathbf{A} = \mathbf{T} - gz \cdot \mathbf{I}$, (2.23) then looks like this:

$$\begin{gathered} \int_\Omega \mathbf{w}^{(k)^T} \cdot \tfrac{\partial \mathbf{v}}{\partial t}\, d\Omega + \int_\Omega \mathbf{w}^{(k)^T} \cdot ((\mathbf{v} \cdot \nabla) \cdot \mathbf{v})\, d\Omega \;= \\ \int_{\partial\Omega} \mathbf{w}^{(k)^T} \cdot (\mathbf{T} \cdot \mathbf{n})\, d(\partial\Omega) \;-\; \int_{\partial\Omega} gz \cdot (\mathbf{w}^{(k)^T} \cdot \mathbf{n})\, d(\partial\Omega) - \int_\Omega (\sum_{i,j} \mathbf{A}_{ij} \tfrac{\partial \mathbf{w}_i^{(k)}}{\partial x_j})\, d\Omega \end{gathered} \tag{2.24}$$

$\mathbf{T}$ is the same as $\hat{\mathbf{T}}$ except that we use $\mathbf{v}$ instead of $\hat{\mathbf{v}}$ to compute it.

At this point, we are able to incorporate the boundary condition (2.3) and the condition (2.19) set forth to the weight functions, which yield

$$\int_{\partial\Omega} \mathbf{w}^{(k)^T} \cdot (\mathbf{T} \cdot \mathrm{n})\, d(\partial\Omega) = 0 \ .$$

Hence, (2.24) has now the following form.

$$\int_{\Omega} \mathbf{w}^{(k)^T} \cdot \frac{\partial \mathbf{v}}{\partial t}\, d\Omega + \int_{\Omega} \mathbf{w}^{(k)^T} \cdot ((\mathbf{v} \cdot \nabla) \cdot \mathbf{v})\, d\Omega = - \int_{\partial\Omega} gz \cdot (\mathbf{w}^{(k)^T} \cdot \mathrm{n})\, d(\partial\Omega) - \int_{\Omega} (\sum_{i,j} \mathbf{A}_{ij} \frac{\partial \mathbf{w}_i^{(k)}}{\partial x_j})\, d\Omega \tag{2.25}$$

Another way of expressing $\mathbf{A}$ is $\mathbf{A} = -\frac{p}{\rho} \cdot \mathbf{I} + \mathbf{ST} - gz \cdot \mathbf{I}$, which would (based on (2.25)) lead to the following equation.

$$\begin{aligned} &\int_{\Omega} \mathbf{w}^{(k)^T} \cdot \tfrac{\partial \mathbf{v}}{\partial t}\, d\Omega + \int_{\Omega} \mathbf{w}^{(k)^T} \cdot ((\mathbf{v} \cdot \nabla) \cdot \mathbf{v})\, d\Omega \ = \\ &- \int_{\partial\Omega} gz \cdot (\mathbf{w}^{(k)^T} \cdot \mathrm{n})\, d(\partial\Omega) \ + \ \int_{\Omega} (\sum_{i,j} \tfrac{p}{\rho} \cdot \mathbf{I}_{ij} \tfrac{\partial \mathbf{w}_i^{(k)}}{\partial x_j})\, d\Omega \ - \\ &\int_{\Omega} (\sum_{i,j} \mathbf{ST}_{ij} \tfrac{\partial \mathbf{w}_i^{(k)}}{\partial x_j})\, d\Omega \ + \ \int_{\Omega} (\sum_{i,j} gz \cdot \mathbf{I}_{ij} \tfrac{\partial \mathbf{w}_i^{(k)}}{\partial x_j})\, d\Omega \end{aligned} \tag{2.26}$$

This equation is equivalent to

$$\begin{aligned}
&\int_{\Omega} \mathbf{w}^{(k)^T} \cdot \tfrac{\partial \mathbf{v}}{\partial t}\, d\Omega \;+\; \int_{\Omega} \mathbf{w}^{(k)^T} \cdot ((\mathbf{v} \cdot \nabla) \cdot \mathbf{v})\, d\Omega \;= \\
&-\int_{\partial\Omega} gz \cdot (\mathbf{w}^{(k)^T} \cdot \mathbf{n})\, d(\partial\Omega) \;+\; \int_{\Omega} \tfrac{p}{\rho} (\nabla \cdot \mathbf{w}^{(k)})\, d\Omega \;- \\
&\int_{\Omega} (\textstyle\sum_{i,j} \mathbf{ST}_{ij} \tfrac{\partial \mathbf{w}_i^{(k)}}{\partial x_j})\, d\Omega \;+\; \int_{\Omega} gz\, (\nabla \cdot \mathbf{w}^{(k)})\, d\Omega
\end{aligned} \tag{2.27}$$

With requirement (2.18), we obtain finally

$$\int_{\Omega} \mathbf{w}^{(k)T} \cdot \frac{\partial \mathbf{v}}{\partial t}\, d\Omega + \int_{\Omega} \mathbf{w}^{(k)T} \cdot ((\mathbf{v} \cdot \nabla) \cdot \mathbf{v})\, d\Omega \;=\; -\int_{\partial\Omega} gz \cdot (\mathbf{w}^{(k)^T} \cdot \mathbf{n})\, d(\partial\Omega) \;-\; \int_{\Omega} (\sum_{i,j} \mathbf{ST}_{ij} \frac{\partial w_i}{\partial x_j})\, d\Omega \tag{2.28}$$

Note, that we now eliminated the terms containing the pressure p. This is of importance, since $\mathbf{v}$ is the only unknown remaining in the equation. Hence, we try to find an expression for the b_i's.

Thus, we incorporate Ansatz (2.13) and rearrange the order of the terms to obtain

$$\begin{gathered}
\sum_{i=1}^{N} \tfrac{\partial b_i}{\partial t} \int_{\Omega} \mathbf{w}^{(k)^T} \cdot \mathbf{e}^{(i)}\, d\Omega \;+\; \sum_{i=1}^{N} b_i \int_{\Omega} \mathbf{w}^{(k)^T} \cdot ((V \cdot \nabla) \cdot \mathbf{e}^{(i)})\, d\Omega \;+ \\
\sum_{i=1}^{N} b_i \int_{\Omega} \mathbf{w}^{(k)^T} \cdot ((\mathbf{e}^{(i)} \cdot \nabla) \cdot V)\, d\Omega \;+\; \sum_{i=1}^{N} b_i \int_{\Omega} (\sum_{n,m} (\mathbf{ST}_{\mathbf{e}^{(i)}})_{nm}\, \tfrac{\partial \mathbf{w}_n{}^{(k)}}{\partial x_m})\, d\Omega \\
= \\
-\int_{\Omega} \mathbf{w}^{(k)^T} \cdot \tfrac{\partial V}{\partial t} \;-\; \int_{\Omega} \mathbf{w}^{(k)^T} \cdot ((V \cdot \nabla) \cdot V)\, d\Omega \;- \\
\int_{\Omega} (\sum_{n,m} (\mathbf{ST}_V^{(i)})_{nm}\, \tfrac{\partial \mathbf{w}_n{}^{(k)}}{\partial x_m})\, d\Omega \;-\; \int_{\Omega} gz\, (\mathbf{w}^{(k)^T} \cdot \mathbf{n})\, d\Omega \;- \\
\sum_{i,j}^{N} b_i \cdot b_j \int_{\Omega} \mathbf{w}^{(k)^T} \cdot ((\mathbf{e}^{(i)} \cdot \nabla) \cdot \mathbf{e}^{(j)})\, d\Omega
\end{gathered} \tag{2.29}$$

where $\mathbf{ST} = \mathbf{ST}_{\mathrm{e}} + \mathbf{ST}_V$.

Here,

$$\mathbf{ST}_{\mathrm{e}}{}^{(i)} = \begin{pmatrix} 2\nu\frac{\partial \mathbf{e}_x^{(i)}}{\partial x} & \nu(\frac{\partial \mathbf{e}_x^{(i)}}{\partial y} + \frac{\partial \mathbf{e}_y^{(i)}}{\partial x}) & \nu(\frac{\partial \mathbf{e}_z^{(i)}}{\partial x} + \frac{\partial \mathbf{e}_x^{(i)}}{\partial z}) \\ \nu(\frac{\partial \mathbf{e}_x^{(i)}}{\partial y} + \frac{\partial \mathbf{e}_y^{(i)}}{\partial x}) & 2\nu\frac{\partial \mathbf{e}_y^{(i)}}{\partial y} & \nu(\frac{\partial \mathbf{e}_y^{(i)}}{\partial z} + \frac{\partial \mathbf{e}_z^{(i)}}{\partial y}) \\ \nu(\frac{\partial \mathbf{e}_z^{(i)}}{\partial x} + \frac{\partial \mathbf{e}_x^{(i)}}{\partial z}) & \nu(\frac{\partial \mathbf{e}_y^{(i)}}{\partial z} + \frac{\partial \mathbf{e}_z^{(i)}}{\partial y}) & 2\nu\frac{\partial \mathbf{e}_z^{(i)}}{\partial z} \end{pmatrix}$$

and

$$\mathbf{ST}_V = \begin{pmatrix} 2\nu\frac{\partial V_x}{\partial x} & \nu(\frac{\partial V_x}{\partial y} + \frac{\partial V_y}{\partial x}) & \nu(\frac{\partial V_z}{\partial x} + \frac{\partial V_x}{\partial z}) \\ \nu(\frac{\partial V_x}{\partial y} + \frac{\partial V_y}{\partial x}) & 2\nu\frac{\partial V_y}{\partial y} & \nu(\frac{\partial V_y}{\partial z} + \frac{\partial V_z}{\partial y}) \\ \nu(\frac{\partial V_z}{\partial x} + \frac{\partial V_x}{\partial z}) & \nu(\frac{\partial V_y}{\partial z} + \frac{\partial V_z}{\partial y}) & 2\nu\frac{\partial V_z}{\partial z} \end{pmatrix}$$

are the parts of $\mathbf{ST}$ due to the functions $\mathrm{e}^{(i)}$ and V .

The next step is to approximate $\frac{\partial b_i}{\partial t}$. We define a certain timestep Δt and discrete time points t_m with

$$t_m = t_0 + m \cdot \Delta t \quad . \tag{2.30}$$

t_0 is the initial point of time where we start our computations. We denote furthermore

$$b_i^{(m)} = b_i(t_m) \quad . \tag{2.31}$$

Using (2.31), we approximate $\frac{\partial b_i}{\partial t}$ by

$$\frac{\partial b_i}{\partial t}(t_m) = \frac{b_i^{(m+1)} - b_i^{(m)}}{\Delta t} \quad . \tag{2.32}$$

Equation (2.32) provides the oppotunity to rewrite (2.29) as an implicit scheme for the unknowns $b_i^{(m+1)}$ ($k = 1...N$), provided that all $b_i^{(m)}$ are known from the previous time step computations:

$$\begin{gathered}
\sum_{i=1}^{N} b_i^{(m+1)} \Big[\tfrac{1}{\Delta t} \int_\Omega \mathbf{w}^{(k)^T} \cdot \mathbf{e}^{(i)} \, d\Omega \; + \; \int_\Omega \mathbf{w}^{(k)^T} \cdot ((V \cdot \nabla) \cdot \mathbf{e}^{(i)}) \, d\Omega \; + \\
\int_\Omega \mathbf{w}^{(k)^T} \cdot ((\mathbf{e}^{(i)} \cdot \nabla) \cdot V) \, d\Omega \; + \; \int_\Omega \Big(\sum_{n,m} (\mathbf{ST}_{\mathbf{e}}{}^{(i)})_{nm} \, \tfrac{\partial \mathbf{w}_n{}^{(k)}}{\partial x_m} \Big) \, d\Omega \Big] \\
= \\
\sum_{i=1}^{N} b_i^{(m)} \Big(\tfrac{1}{\Delta t} \int_\Omega \mathbf{w}^{(k)^T} \cdot \mathbf{e}^{(i)} \, d\Omega \Big) \; - \; \int_\Omega \mathbf{w}^{(k)^T} \cdot \tfrac{\partial V}{\partial t} \; - \\
\int_\Omega \mathbf{w}^{(k)^T} \cdot ((V \cdot \nabla) \cdot V) \, d\Omega \; - \; \int_\Omega \Big(\sum_{n,m} (\mathbf{ST}_V^{(i)})_{nm} \, \tfrac{\partial \mathbf{w}_n{}^{(k)}}{\partial x_m} \Big) \, d\Omega \; - \\
\int_\Omega gz \, (\mathbf{w}^{(k)^T} \cdot \mathbf{n}) \, d\Omega \; - \; \sum_{i,j}^{N} b_i^{(m+1)} \cdot b_j^{(m+1)} \int_\Omega \mathbf{w}^{(k)^T} \cdot ((\mathbf{e}^{(i)} \cdot \nabla) \cdot \mathbf{e}^{(j)}) \, d\Omega
\end{gathered} \tag{2.33}$$

In order to make this huge equation readable, we denote the matrix **L** by

$$\begin{gathered}
\mathbf{L}^{(k,i)} = \\
\tfrac{1}{\Delta t} \int_\Omega \mathbf{w}^{(k)^T} \cdot \mathbf{e}^{(i)} \, d\Omega \; + \; \int_\Omega \mathbf{w}^{(k)^T} \cdot ((V \cdot \nabla) \cdot \mathbf{e}^{(i)}) \, d\Omega \; + \\
\int_\Omega \mathbf{w}^{(k)^T} \cdot ((\mathbf{e}^{(i)} \cdot \nabla) \cdot V) \, d\Omega \; + \; \int_\Omega \Big(\sum_{n,m} (\mathbf{ST}_{\mathbf{e}}{}^{(i)})_{nm} \, \tfrac{\partial \mathbf{w}_n{}^{(k)}}{\partial x_m} \, d\Omega
\end{gathered} \tag{2.34}$$

The vector **R** is denoted by

$$
\begin{gathered}
\mathbf{R}^{(k)} = \\
\sum_{i=1}^{N} b_i^{(m)} \left(\tfrac{1}{\Delta t} \int_{\Omega} {\mathbf{w}^{(k)}}^T \cdot \mathbf{e}^{(i)} \, d\Omega \right) \; - \; \int_{\Omega} {\mathbf{w}^{(k)}}^T \cdot \tfrac{\partial V}{\partial t} \; - \\
\int_{\Omega} {\mathbf{w}^{(k)}}^T \cdot ((V \cdot \nabla) \cdot V) \, d\Omega \; - \; \int_{\Omega} \Big(\sum_{n,m} (\mathbf{ST}_V^{(i)})_{nm} \, \tfrac{\partial {\mathbf{w}_n}^{(k)}}{\partial x_m} \Big) \, d\Omega \; - \\
\int_{\Omega} gz \, ({\mathbf{w}^{(k)}}^T \cdot \mathbf{n}) \, d\Omega \; - \; \sum_{i,j}^{N} b_i^{(m+1)} \cdot b_j^{(m+1)} \int_{\Omega} {\mathbf{w}^{(k)}}^T \cdot ((\mathbf{e}^{(i)} \cdot \nabla) \cdot \mathbf{e}^{(j)}) \, d\Omega
\end{gathered}
\tag{2.35}
$$

and the vector **b** by :

$$
\mathbf{b}^{(m)} = \begin{pmatrix} b_1^{(m)} \\ \\ \vdots \\ \\ b_N^{(m)} \end{pmatrix} \tag{2.36}
$$

With these abbreviations we can write equation (2.33) simply as

$$
\mathbf{L} \cdot \mathbf{b}^{(m+1)} = \mathbf{R} \tag{2.37}
$$

At this point, we have gained our aim to derive an expression for the b_i's in the Ansatzfunction (2.13). Thus we can completely determine the approximate velocity distribution throughout the wave tank.

Taking into account that the vector **R** is dependent on $\mathbf{b}^{(m+1)}$, we realize that equation (2.37) is an implicit time scheme for the Navier-Stokes-equations for the problem introduced in Chapter 1.

Conclusion : We realize that $\mathbf{L}^{(k,i)}$ and $\mathbf{R}^{(k)}$ depend only on the choice of the basis and weight functions and on the parameter Δt. So we could compute $\mathbf{L}^{(k,i)}$ and $\mathbf{R}^{(k)}$

if we knew the region Ω at the time t_{m+1}. Thus the conclusion is that the surface must be known prior to solving for the coefficients $\mathbf{b}_i^{(m+1)}$.

The next section shows how to compute the surface at time t_{m+1} (thereby computing the region Ω at t_{m+1}).

2.3.2 Surface

Fixed x-y-grid for discretization

We have already seen that the differential equation modelling the motion of the surface is

$$\frac{\partial h}{\partial t} + \hat{u}\frac{\partial h}{\partial x} + \hat{v}\frac{\partial h}{\partial y} = \hat{w} \quad . \tag{2.38}$$

In the following, we shall derive an time-implicit difference scheme which approximates the motion of the surface.
In order to do that, we establish a rectangular x-y-grid that covers the x-y-plane of the wave tank. We call the discrete grid points (x_i, y_j) where $i = 0 \ldots M_f + 1$ and $j = 0 \ldots N_f + 1$. $M_f + 2$ is the number of intervals in x-direction, $N_f + 2$ is the number of intervals in y-direction:

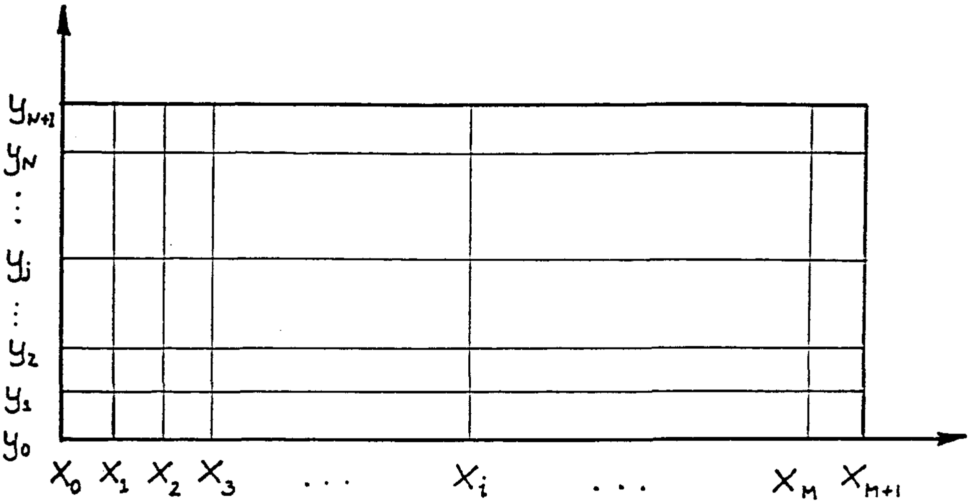


figure 2.3: x-y-grid for discretization of the surface

We approximate $\frac{\partial h}{\partial t}$, $\frac{\partial h}{\partial x}$ and $\frac{\partial h}{\partial y}$ by

$$\frac{\partial h}{\partial t} \approx \frac{h(t_{m+1}, x_i, y_j) - h(t_m, x_i, y_j)}{\Delta t} \quad ,$$

$$\frac{\partial h}{\partial x} \approx \frac{h(t_{m+1}, x_{i+1}, y_j) - h(t_{m+1}, x_{i-1}, y_j)}{x_{i+1} - x_{i-1}} \quad ,$$

$$\frac{\partial h}{\partial y} \approx \frac{h(t_{m+1}, x_i, y_{j+1}) - h(t_{m+1}, x_i, y_{j-1})}{y_{j+1} - y_{j-1}} \quad .$$

Thus, from the original differential equation (2.38), we obtain

$$\tfrac{h(t_{m+1},x_i,y_j)-h(t_m,x_i,y_j)}{\Delta t} \; +$$

$$u(t_{m+1}, x_i, y_j) \; \tfrac{h(t_{m+1},x_{i+1},y_j)-h(t_{m+1},x_{i-1},y_j)}{x_{i+1}-x_{i-1}} \; +$$

$$v(t_{m+1}, x_i, y_j) \; \tfrac{h(t_{m+1},x_i,y_{j+1})-h(t_{m+1},x_i,y_{j-1})}{y_{j+1}-y_{j-1}} \tag{2.39}$$

$$=$$

$$w(t_{m+1}, x_i, y_j)$$

This gives us finally the time implicit difference scheme for the surface:

$$\tfrac{\Delta t}{x_{i+1}-x_{i-1}} \, u(t_{m+1}) \, h(t_{m+1}, x_{i+1}, y_j) \;-\; \tfrac{\Delta t}{x_{i+1}-x_{i-1}} \, u(t_{m+1}) \, h(t_{m+1}, x_{i-1}, y_j) \; +$$

$$\tfrac{\Delta t}{y_{j+1}-y_{j-1}} \, v(t_{m+1}) \, h(t_{m+1}, x_i, y_{j+1}) \;-\; \tfrac{\Delta t}{y_{j+1}-y_{j-1}} \, u(t_{m+1}) \, h(t_{m+1}, x_i, y_{j-1}) \; +$$

$$h(t_{m+1}, x_i, y_j) \tag{2.40}$$

$$=$$

$$h(t_m, x_i, y_j) \; + \; \Delta t \, w(t_{m+1}, x_i, y_j)$$

Advantages of this scheme based on the fixed grid :

Since the x-y-grid-points are fixed, we can imagine the approximation of the surface by table tennis balls that swim on the surface and are allowed to move up and down (in z-direction) only. This has the advantage that the table tennis balls do not drift away and we do not necessarily have to add new balls as we would have to in the case of moving grid if a gap between any two balls or the distance to the boundary grows too large. Moreover, we do not have to remove balls from the surface in case they come too close together. This means that setting up this scheme on a computer is very simple and is basically the implementation of a hyperbolic differential equation which is quite well understood.

Disadvantages :

1.) The scheme above provides only $M_f \cdot N_f$ equations for the $(M_f+2)*(N_f+2)+4$ unknowns ($h(t_{m+1}, x_i, y_j)$ for $i = 0...M_f+1$ and $j = 0...N_f+1$). The unknowns that cannot be calculated are the table tennis balls being directly at the boundary $\partial\Omega$ of the wave tank. So, the values for those points must be obtained by another method, for example by assuming that the slope formed by the surface close to the boundary is nearly constant in a sufficiently small neighborhood of $\partial\Omega$. In this case, if the shape of the surface at the boundary looks like the following picture

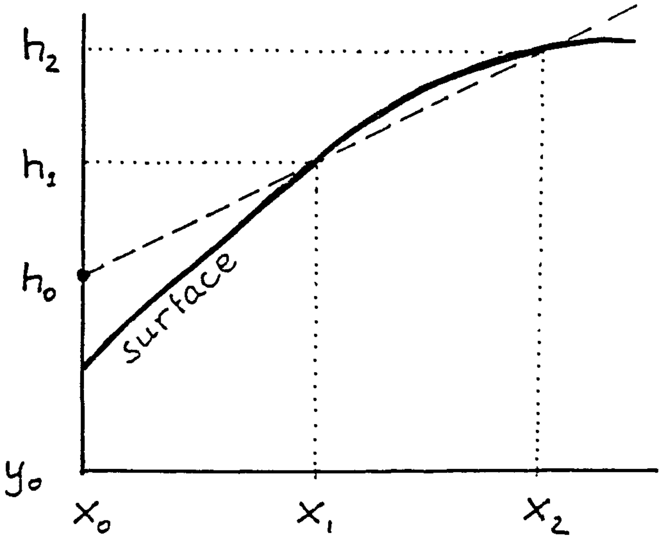


figure 2.4: Example of a certain surface shape

then we can make the assumption that

$$\frac{h(x_1) - h(x_0)}{x_1 - x_0} = \frac{h(x_2) - h(x_1)}{x_2 - x_1} \ . \tag{2.41}$$

2.) If the surface of the fluid becomes very steep (for example when a wave breaks) then this scheme fails and numerical instabilities occur in higher and higher spikes growing on the surface. It is very difficult to prevent these instabilities, since the mechanism for these occurences is quite simple: the steeper the surface, the larger

the distance between adjacent table tennis balls, hence the approximation of the surface at steep points is less accurate and so the probability of making large errors rises and the spikes probably grow larger. So the process is self supporting. One mechanism to prevent this would be to inject more table tennis balls (i.e. to establish more grid points) when the surface becomes sufficiently steep. However, then the whole algorithm would lose its simplicity.
Moreover, breaking waves tend to form overhanging shapes, but the present scheme is unable to calculate an overhanging wave formation.

3.) The scheme above, viewed as a system of linear equations, gives an $(M \cdot N) \times (M \cdot N)$-Matrix which might be inconveniently large depending on how many table tennis balls are chosen. The benfit is that the matrix is extremely sparse such that we can use algorithms for sparse matrices.

Nevertheless, for our test runs we used mainly this surface scheme due to its easy handling.

Floating x-y-grid-points

We remember that the original differential equation for the surface (2.8) was

$$\frac{d\mathbf{s}}{dt} = \mathbf{v}$$

We now approximate the surface as a set of N_m points

$$\mathbf{s}_i = \begin{pmatrix} s_{ix} \\ s_{iy} \\ h_i \end{pmatrix}$$

We recognize furthermore, that (2.8) is an ordinary differential equation, and we use the implicit Euler method to set up the scheme

$$\frac{d\mathbf{s}_i}{dt} \approx \frac{\mathbf{s}_i(t_{m+1}) - \mathbf{s}_i(t_m)}{\Delta t} \approx \mathbf{v}(\mathbf{s}_i(t_{m+1})) \quad .$$

Note that Δt is the same as the Δt used in scheme (2.33).

The above equation gives us

$$\mathbf{s}_i(t_{m+1}) = \Delta t \cdot \mathbf{v}(\mathbf{s}_i(t_{m+1})) + \mathbf{s}_i(t_m) \quad , \tag{2.42}$$

which is easily recognized as an implicit Euler scheme for the above ordinary differential equation.

Advantages of this scheme :

1.) For the N_m unknown surface points, the scheme provides exactly N_m different equations. So there is no need to find other conditions for the table tennis balls on the boundary. Moreover, these N_m equations are decoupled, one could solve every single of them without regarding the other $N_m - 1$ equations, which means that we do not have to handle a huge $N_m \times N_m$-matrix.

2.) We will be able to deal even with overhanging wave formations. In our test runs using this kind of surface model, no numerical instabilities occured.

Disadvantages :

We have the situation that (mainly at the left end of the wave tank where the wave board oscillates) the table tennis balls drift considerably and disappear behind the boundary or the gap between the balls and the wall could grow too large. It needs a lot of effort to handle these occurances. Most of all, adding new points to the surface when the gap becomes too large is a problem that is not well understood yet.

For this reason we use the following idea. If the surface looks like the picture below,

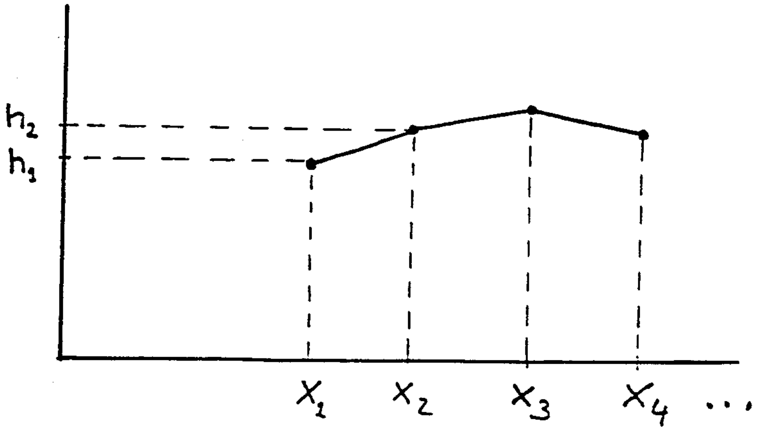


figure 2.5: Surface discretization at the boundary

then we add points in the gap between the last point x_1 and the boundary. The new points are on a line which has the same slope as the line drawn from the point x_2 to x_1.

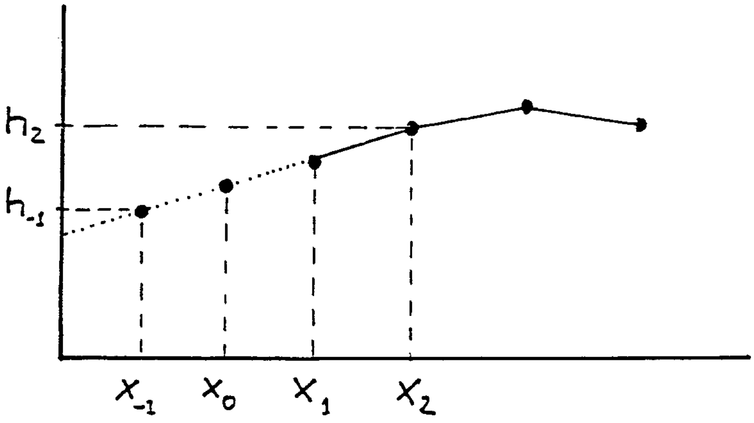


figure 2.6: Injection of new discrete points

One example of solutions where the idea of floating grid points has been employed, is exhibited in chapter 6.

Conclusion : At the end of section 2.3.1 we concluded that the coefficients $b_i^{(m+1)}$ can be evaluated only if the region Ω is known at the time t_{m+1}.

The summarizing conclusion of the present section is that the surface at t_{m+1} (i.e. Ω at t_{m+1}) cannot be computed unless the velocity $\mathbf{v}$ at time t_{m+1} (i.e. the coefficients

$\mathbf{b}_i^{(m+1)}$) are known. This is a contradiction in itself, and the way out of this is the establishment of the following iteration algorithm.

2.3.3 Iteration algorithm for the velocity and the surface at the time t_{m+1}

```
1.) define $\epsilon_s$ (break condition for surface iteration)
2.) define $\epsilon_v$ (break condition for velocity iteration)
3.) define initial surface $\mathbf{s}^{(0)}$
4.) define initial velocity $\mathbf{v}^{(0)}$
5.) define initial time $t_0$
6.) set $m = 0$ (counter for discrete timesteps)
LOOP (over all time steps)
    7.) Compute $t_{m+1} = t_m + \Delta t_m$
    8.) guess a new surface $\mathbf{s}_{(0)}^{(m+1)}$ for time $t_{m+1}$ or let simply $\mathbf{s}_{(0)}^{(m+1)} = \mathbf{s}^{(m)}$
    9.) guess a new velocity $\mathbf{v}_{(0)}^{(m+1)}$ for time $t_{m+1}$ or let simply $\mathbf{v}_{(0)}^{(m+1)} = \mathbf{v}^{(m)}$
    10.) set $i = 0$ (counter for iteration steps)
    WHILE $\frac{\| \mathbf{s}_{(i)}^{(m+1)} - \mathbf{s}_{(i-1)}^{(m+1)} \|}{\| \mathbf{s}_{(i)}^{(m+1)} \| + \| \mathbf{s}_{(i-1)}^{(m+1)} \|} > \epsilon_{\mathbf{s}}$ or $\frac{\| \mathbf{v}_{(i)}^{(m+1)} - \mathbf{v}_{(i-1)}^{(m+1)} \|}{\| \mathbf{v}_{(i)}^{(m+1)} \| + \| \mathbf{v}_{(i-1)}^{(m+1)} \|} > \epsilon_{\mathbf{v}}$

        11.) set $i = i + 1$

        12.) iterate scheme (2.37) using $\mathbf{s}_{(i-1)}^{(m+1)}$ as integration boundary to obtain
             a new velocity $\mathbf{v}_{(i)}^{(m+1)}$

        13.) iterate scheme (2.40) using $\mathbf{v}_{(i)}^{(m+1)}$ as velocity to obtain
             a new surface $\mathbf{s}_{(i)}^{(m+1)}$

    END WHILE

    14.) set $\mathbf{v}^{(m+1)} = \mathbf{v}_{(i)}^{(m+1)}$ and save

    15.) set $\mathbf{s}^{(m+1)} = \mathbf{s}_{(i)}^{(m+1)}$ and save

    16.) set $m = m + 1$

END LOOP
```

2.3.4 Restriction of the Mathematical Model to 2 Dimensions and Nondimensionalization

Although the models (2.33) and (2.40) / (2.42) may be used in 3 dimensions, we will make a restriction of these schemes to the dimensions x and z only. Here are several reasons for this step:

1.) Due to their unavoidable complexity, these algorithms run very slowly on a computer, even in the limited two dimensions case. Computation of the three dimensions case would take far too long.

2.) Involving the y-dimension is not reasonable for some basic test runs of the algorithms, since one can assume that in several cases the change of velocity in y-direction is negligible in comparison to the other dimensional directions. This is due to the fact that the wave tank is very narrow. However, we took into account that there are cases where the y-direction is not negligible at all, namely when crosswaves are occuring. That is why the basic mathematical model was established for 3 dimensions.
Computing in 3 dimensions would be possible with much more powerful computer equipment. Refer to chapter 4 for more detailed information.

Please note that from now on all considerations are done in the 2-dimensional case. We neglect the y-direction.

Nondimensionalization and Characteristic Numbers

Nondimensionalization is a standard tool useful for the study of problems in fluid mechanics. This technique generalizes the problem, and all necessary parameters characterizing the problem will be contained in the so called characteristic numbers (as we shall see in this section). Whenever two different problems have the same characteristic numbers and the same nondimensionalized boundary conditions, then the problems are similar and need not to be considered seperately.

For our problem, we introduce the usual nondimensionalizations.

For the variables with the dimension 'length' we choose

$$\begin{aligned} \xi &= \frac{2}{L}x - 1 \quad , \\ \eta &= \frac{z}{H} \quad , \\ gr(\xi) &= \frac{Gr(x(\xi))}{H} \quad . \end{aligned}$$

For the variables with the dimensions 'velocity' we choose

$$\begin{aligned} \bar{\mathbf{e}}^{(i)} &= \frac{\mathbf{e}^{(i)}}{U_0} \quad , \\ \bar{V} &= \frac{V}{U_0} \quad , \end{aligned}$$

where U_0 is a characteristic velocity.

Finally, for the variables with the dimension 'time' we choose

$$\tau = \frac{t}{T}$$

where T is a characteristic time.

We establish furthermore the characteristic numbers

$$Re = \frac{L \cdot U_0}{\nu} \quad ,$$

$$St = \frac{L}{U_0 \cdot T} \quad ,$$

$$Fr = \frac{g \cdot L}{U_0^2} \quad ,$$

$$A = \frac{L}{2H} \quad .$$

In the following section we shall see how these numbers fit into the existing mathematical model.

Transformation of the scheme (2.33) into a nondimensionalized scheme

Using the nondimensionalizations introduced in the previous section, one obtains the following expressions for all the single terms used in equation (2.33). We exhibit these equations here since it is exactly this set of equations that is implemented in the computer program and the integrals appear exactly under the same number in the source code.

1.)

$$\frac{1}{\Delta t}\int\limits_{\Omega} \mathbf{w}^{(k)^T} \cdot \mathbf{e}_x^{(i)}\, d\Omega \;=\; U_0^2 H\,(\frac{St}{\Delta_T}\int\limits_{\bar{\Omega}} (\,\mathbf{w}_x^{(k)}\,\bar{\mathbf{e}}_x^{(i)} \;+\; \mathbf{w}_z^{(k)}\,\bar{\mathbf{e}}_z^{(i)}\,)\, d\xi\, d\eta) \tag{2.43}$$

2.)

$$\begin{gathered}\int\limits_{\Omega} \mathbf{w}^{(k)^T} \cdot ((V \cdot \nabla) \cdot \mathbf{e}^{(i)})\, d\Omega \;= \\ U_0^2\, H\, \int\limits_{\bar{\Omega}} (\mathbf{w}_x^{(k)}\,\bar{V}_x\,\frac{\partial \bar{\mathbf{e}}_x^{(i)}}{\partial \xi} \;+\; \mathbf{w}_z^{(k)}\,\bar{V}_x\,\frac{\partial \bar{\mathbf{e}}_z^{(i)}}{\partial \xi} \;+\; A \cdot \mathbf{w}_x^{(k)}\,\bar{V}_z\,\frac{\partial \bar{\mathbf{e}}_x^{(i)}}{\partial \eta} \;+\; A \cdot \mathbf{w}_z^{(k)}\,\bar{V}_z\,\frac{\partial \bar{\mathbf{e}}_z^{(i)}}{\partial \eta}) d\xi d\eta\end{gathered} \tag{2.44}$$

3.)

$$\begin{gathered}\int\limits_{\Omega} \mathbf{w}^{(k)^T} \cdot ((\mathbf{e}^{(i)} \cdot \nabla) \cdot V)\, d\Omega \;= \\ U_0^2\, H\, \int\limits_{\bar{\Omega}} (\mathbf{w}_x^{(k)}\,\bar{\mathbf{e}}_x^{(i)}\,\frac{\partial \bar{V}_x}{\partial \xi} \;+\; \mathbf{w}_z^{(k)}\,\bar{\mathbf{e}}_x^{(i)}\,\frac{\partial \bar{V}_z}{\partial \xi} \;+\; A \cdot \mathbf{w}_x^{(k)}\,\bar{\mathbf{e}}_z^{(i)}\,\frac{\partial \bar{V}_x}{\partial \eta} \;+\; A \cdot \mathbf{w}_z^{(k)}\,\bar{\mathbf{e}}_z^{(i)}\,\frac{\partial \bar{V}_z}{\partial \eta}\,)\, d\xi d\eta\end{gathered} \tag{2.45}$$

4.)

$$\begin{gathered}\int\limits_{\Omega} (\sum_{n,m} (\mathbf{ST}_{\mathbf{e}}{}^{(i)})_{nm}\,\frac{\partial \mathbf{w}_n{}^{(k)}}{\partial x_m})\, d\Omega\,) \;= \\ U_0^2\, H\, (\frac{1}{Re}\int\limits_{\bar{\Omega}} (2\,\frac{\partial \mathbf{w}_x^{(k)}}{\partial \xi}\,\frac{\partial \bar{\mathbf{e}}_x^{(i)}}{\partial \xi} \;+\; A\,\frac{\partial \mathbf{w}_x^{(k)}}{\partial \eta}\,\frac{\partial \bar{\mathbf{e}}_z^{(i)}}{\partial \xi} \;+\; A^2\,\frac{\partial \mathbf{w}_x^{(k)}}{\partial \eta}\,\frac{\partial \bar{\mathbf{e}}_x^{(i)}}{\partial \eta} \;+ \\ \frac{\partial \mathbf{w}_z^{(k)}}{\partial \xi}\,\frac{\partial \bar{\mathbf{e}}_z^{(i)}}{\partial \xi} \;+\; A\,\frac{\partial \mathbf{w}_z^{(k)}}{\partial \xi}\,\frac{\partial \bar{\mathbf{e}}_x^{(i)}}{\partial \eta} \;+\; 2A^2\,\frac{\partial \mathbf{w}_z^{(k)}}{\partial \eta}\,\frac{\partial \bar{\mathbf{e}}_z^{(i)}}{\partial \eta})\, d\xi\, d\eta\,)\end{gathered} \tag{2.46}$$

5.)

$$\begin{gathered}\int\limits_{\Omega} (\sum_{n,m} (\mathbf{ST}_V^{(i)})_{nm}\,\frac{\partial \mathbf{w}_n{}^{(k)}}{\partial x_m})\, d\Omega\,) \;= \\ U_0^2\, H\, (\frac{1}{Re}\int\limits_{\bar{\Omega}} (2\,\frac{\partial \mathbf{w}_x^{(k)}}{\partial \xi}\,\frac{\partial \bar{V}_x^{(i)}}{\partial \xi} \;+\; A\,\frac{\partial \mathbf{w}_x^{(k)}}{\partial \eta}\,\frac{\partial \bar{V}_z^{(i)}}{\partial \xi} \;+\; A^2\,\frac{\partial \mathbf{w}_x^{(k)}}{\partial \eta}\,\frac{\partial \bar{V}_x^{(i)}}{\partial \eta} \;+ \\ \frac{\partial \mathbf{w}_z^{(k)}}{\partial \xi}\,\frac{\partial \bar{V}_z^{(i)}}{\partial \xi} \;+\; A\,\frac{\partial \mathbf{w}_z^{(k)}}{\partial \xi}\,\frac{\partial \bar{V}_x^{(i)}}{\partial \eta} \;+\; 2A^2\,\frac{\partial \mathbf{w}_z^{(k)}}{\partial \eta}\,\frac{\partial \bar{V}_z^{(i)}}{\partial \eta})\, d\xi\, d\eta\,)\end{gathered} \tag{2.47}$$

6.)

$$\int_{\Omega} \mathbf{w}^{(k)^T} \cdot \frac{\partial V}{\partial t} = U_0^2 \, H \, (St \int_{\bar{\Omega}} (\mathbf{w}_x^{(k)} \frac{\partial \bar{V}_x}{\partial \tau} + \mathbf{w}_z^{(k)} \frac{\partial \bar{V}_z}{\partial \tau}) \, d\xi \, d\eta) \tag{2.48}$$

7.)

$$\int_{\Omega} \mathbf{w}^{(k)^T} \cdot ((V \cdot \nabla) \cdot V) \, d\Omega = \\ U_0^2 \, H \int_{\bar{\Omega}} (\mathbf{w}_x^{(k)} \, \bar{V}_x \frac{\partial \bar{V}_x}{\partial \xi} + \mathbf{w}_z^{(k)} \, \bar{V}_x \frac{\partial \bar{V}_z}{\partial \xi} + A \cdot \mathbf{w}_x^{(k)} \, \bar{V}_z \frac{\partial \bar{V}_x}{\partial \eta} + A \cdot \mathbf{w}_z^{(k)} \, \bar{V}_z \frac{\partial \bar{V}_z}{\partial \eta}) d\xi d\eta \tag{2.49}$$

8.)

$$\int_{\partial \Omega} gz \, (\mathbf{w}^{(k)^T} \cdot \mathbf{n}) \, d(\partial \Omega) = U_0^2 \, H \, (Fr \int_{\bar{\Omega}} \mathbf{w}_z \, d\xi \, d\eta) \tag{2.50}$$

9.)

$$\int_{\Omega} \mathbf{w}^{(k)^T} \cdot ((\mathbf{e}^{(i)} \cdot \nabla) \cdot \mathbf{e}^{(j)}) \, d\Omega = \\ U_0^2 \, H \int_{\bar{\Omega}} (\mathbf{w}_x^{(k)} \, \bar{\mathbf{e}}_x^{(i)} \frac{\partial \bar{\mathbf{e}}_x^{(j)}}{\partial \xi} + \mathbf{w}_z^{(k)} \, \bar{\mathbf{e}}_x^{(i)} \frac{\partial \bar{\mathbf{e}}_z^{(j)}}{\partial \xi} + A \cdot \mathbf{w}_x^{(k)} \, \bar{\mathbf{e}}_z^{(i)} \frac{\partial \bar{\mathbf{e}}_x^{(j)}}{\partial \eta} + A \cdot \mathbf{w}_z^{(k)} \, \bar{\mathbf{e}}_z^{(i)} \frac{\partial \bar{\mathbf{e}}_z^{(j)}}{\partial \eta}) \, d\xi \, d\eta \tag{2.51}$$

Notice that these nine equations allow us to operate completely in the nondimensionalized model. We need to know only the characteristic numbers Re, St, Fr, A and the nondimensionalized boundary conditions which together would uniquely determine the problem.

In our computer application, we neither compute the matrix $\mathbf{L}^{(k,i)}$ nor the vector $\mathbf{R}^{(k)}$, but we compute

$$\bar{\mathbf{L}}^{(k,i)} = \frac{1}{U_0^2 \, H} \cdot \mathbf{L}^{(k,i)}$$

and

$$\bar{\mathbf{R}}^{(k)} = \frac{1}{U_0^2 H} \cdot \mathbf{R}^{(k)} ,$$

which are their appropriate nondimensionalized forms.

Thus, scheme (2.37) appears as

$$\bar{\mathbf{L}}^{(k,i)} \cdot \mathbf{b}^{(m+1)} = \bar{\mathbf{R}}^{(k)} . \tag{2.37/2}$$

It is exactly equation (2.37/2) which is implemented in our computer application.

Nondimensionalization for the schemes of the surface motion

In nondimensionalized notation, the scheme for the surface motion in the two dimensional case with fixed x-y-grid (2.39) appears as follows.

$$\begin{aligned} \tfrac{1}{St}\,\tfrac{\Delta\tau}{\Delta\xi_i}\,\bar{u}(\tau_{m+1},\xi_i)\cdot\bar{h}(\tau_{m+1},\xi_{i+1}) \;-\; \tfrac{1}{St}\,\tfrac{\Delta\tau}{\Delta\xi_i}\,\bar{u}(\tau_{m+1},\xi_i)\cdot\bar{h}(\tau_{m+1},\xi_{i-1}) \;+\; \bar{h}(\tau_{m+1},\xi_i) \;= \\ \bar{h}(\tau_m,\xi_i) \;+\; \tfrac{A}{St}\,\Delta\tau\,\bar{w}(\tau_{m+1},\xi_i) \end{aligned} \tag{2.52}$$

Moreover, the scheme for the floating grid (2.42) has the following nondimensionalized appearance,

$$\begin{aligned} \bar{\mathrm{s}}_{ix}(\tau_{m+1}) &= \tfrac{1}{St}\,\Delta\tau\,\bar{u}(\bar{\mathrm{s}}_i,\tau_{m+1}) \;+\; \bar{\mathrm{s}}_{ix}(\tau_m) \\ \bar{h}_i(\tau_{m+1}) &= \tfrac{A}{St}\,\Delta\tau\,\bar{u}(\bar{\mathrm{s}}_i,\tau_{m+1}) \;+\; \bar{h}_i(\tau_m) \end{aligned} \tag{2.53}$$

Again, concerning the surface motion, we recognize that we can work completely under nondimensionalized notation.

Chapter 3: Numerical Implementation

In this chapter we will discuss the numerical implementation of the algorithms derived in chapter 2 and we will give an answer to the question which kind of basis functions ($\mathbf{e}^{(i)}$'s) and weight functions ($\mathbf{w}^{(k)}$'s), and what kind of the special function V do we have to choose in order to obtain realistic results.

The main problem is obvious:
the best estimate $\mathbf{v}(t)$ for the function $\hat{\mathbf{v}}(t)$ would be

$$\hat{\mathbf{v}}(t) \approx \mathbf{v}(t) = V(t) + \sum_{i=1}^{\infty} b_i(t) \cdot \mathbf{e}^{(i)}$$

This, viewed as a numerical Ansatz, is indeed impossible since we can deal with only a finite number of basis functions. Therefore, it is wise to choose the functions V, $\mathbf{e}^{(i)}$ and $\mathbf{w}^{(k)}$ such that the numerical computations yield the most realistic output.

Moreover, a mathematical method is not ideal for use as a numerical application if the algorithms consume too much time. The integrals that we have to compute are very lengthy and may be very complicated if the surface or the functions V, $\mathbf{e}^{(i)}$ and $\mathbf{w}^{(k)}$ are complicated. So, the goal is to choose these functions in order to solve for the occouring integrals in the least time consuming manner.

These two aspects may contradict each other. Here, we exhibit a suggestion for assembling them in a clever way.

3.1 What do Basis Functions and Weight Functions look like

3.1.1 Basis Functions

We refer to what was already said in chapter 2. There we stated that the basis functions have to satisfy the two conditions (2.14) and (2.15). Equation (2.14) written in another way gives

$$\frac{\partial \mathbf{e}_x^{(i)}}{\partial x} + \frac{\partial \mathbf{e}_z^{(i)}}{\partial z} = 0 \quad .$$

Using the nondimensionalization introduced in chapter 2, we obtain

$$\frac{\partial \bar{\mathbf{e}}_x^{(i)}}{\partial \xi} + A \frac{\partial \bar{\mathbf{e}}_z^{(i)}}{\partial \eta} = 0 \quad . \tag{3.1}$$

Equation (2.15) states that

$$\bar{\mathbf{e}}^{(i)} = 0 \quad at \ \partial\Omega_1 \ , \ \partial\Omega_2 \ , \ \partial\Omega_3 \tag{3.2}$$

which means

$$\begin{aligned} \bar{\mathbf{e}}^{(i)}(\ \xi = -1 \ , \ \eta \ ) &= 0 \ , \\ \bar{\mathbf{e}}^{(i)}(\ \xi \ , \ \eta = gr(\xi) \ ) &= 0 \ , \\ \bar{\mathbf{e}}^{(i)}(\ \xi = 1 \ , \ \eta \ ) &= 0 \ . \end{aligned} \tag{3.3}$$

Based on the conditions (3.1) and (3.3), we define a function $E^{(i)}$, given by

$$E^{(i)} = (\xi^2 - 1)^2 (\eta - gr(\xi))^2 \cdot G_b^{(i)}(\xi) \cdot H_b^{(i)}(\eta) \quad , \tag{3.4}$$

and use this function to establish $\bar{e}^{(i)}$ by

$$\begin{pmatrix} \frac{\partial E^{(i)}}{\partial \eta} \\ \\ -\frac{1}{A}\frac{\partial E^{(i)}}{\partial \xi} \end{pmatrix} = \begin{pmatrix} \bar{e}_x^{(i)} \\ \\ \bar{e}_z^{(i)} \end{pmatrix} = \bar{e}^{(i)} \quad . \tag{3.5}$$

1.) It is easy to verify that condition (3.1) is satisfied.

2.) Furthermore, we easily verify that the factor $(\xi^2 - 1)(\eta - gr(\xi))$ belongs to both $\bar{e}_x^{(i)}$ and $\bar{e}_z^{(i)}$, which is sufficient to satisfy condition (3.3) . For this factor makes $\bar{\mathbf{e}}^{(i)}$ zero at $\eta = gr(\xi)$ (denoting the ground), at $\xi = -1$ (denoting the left boundary) and at $\xi = 1$ (denoting the right boundary).

3.1.2 Weight Functions

Again, we refer to what was already said in chapter 2. There we stated that the basis functions have to satisfy the two conditions (2.18) and (2.19). Equation (2.18) written in another way gives

$$\frac{\partial \mathbf{w}_x^{(k)}}{\partial x} + \frac{\partial \mathbf{w}_z^{(k)}}{\partial z} = 0 \quad .$$

Using the nondimensionalization introduced in chapter 2, we obtain

$$\frac{\partial \mathbf{w}_x^{(k)}}{\partial \xi} + A\frac{\partial \mathbf{w}_z^{(k)}}{\partial \eta} = 0 \quad . \tag{3.6}$$

Equations (2.19) states that

$$\mathbf{w}^{(k)} = 0 \qquad at \;\; \partial\Omega_1 \;,\; \partial\Omega_2 \;,\; \partial\Omega_3 \tag{3.7}$$

which means

$$\begin{aligned} \mathbf{w}^{(k)}(\;\xi = -1 \;,\; \eta\;) &= 0 \;, \\ \mathbf{w}^{(k)}(\;\xi \;,\; \eta = gr(\xi)\;) &= 0 \;, \\ \mathbf{w}^{(k)}(\;\xi = 1 \;,\; \eta\;) &= 0 \;. \end{aligned} \tag{3.8}$$

Based on the conditions (3.6) and (3.8) we define a function $W^{(k)}$

$$W^{(k)} = (\;\xi^2 - 1\;)^2(\;\eta - gr(\xi)\;)^2 \cdot G_w^{(k)}(\xi) \cdot H_w^{(k)}(\eta) \tag{3.9}$$

and use this function to establish $\mathbf{w}^{(k)}$:

$$\begin{pmatrix} \frac{\partial W^{(k)}}{\partial \eta} \\ \\ -\frac{1}{A}\frac{\partial W^{(k)}}{\partial \xi} \end{pmatrix} = \begin{pmatrix} \mathbf{w}_x^{(k)} \\ \\ \mathbf{w}_z^{(k)} \end{pmatrix} = \mathbf{w}^{(k)} \quad . \tag{3.10}$$

1.) It easy to verify that condition (3.6) is satisfied.

2.) Furthermore, we easily verify that the factor $(\;\xi^2 - 1\;)(\;\eta - gr(\xi)\;)$ belongs to both $\mathbf{w}_x^{(k)}$ and $\mathbf{w}_z^{(k)}$, which is sufficient to satisfy condition (3.8) . It is this factor that makes $\mathbf{w}^{(k)}$ zero at $\xi = -1$ (left boundary), at $\eta = gr(\xi)$ (on the ground), and at $\xi = 1$ (right boundary).

3.1.3 The functions $G_b^{(i)}$, $H_b^{(i)}$, $G_w^{(k)}$ and $H_w^{(k)}$

In general, the four functions $G_b^{(i)}$, $H_b^{(i)}$, $G_w^{(k)}$ and $H_w^{(k)}$ introduced in sections 3.1.1 and 3.1.2 can be anything. The only requirement is that the sets $G_b^{(i)}$ and $H_b^{(i)}$ make the set $E^{(i)}$ be a set of linearly independent functions. Similarly, the sets $G_w^{(k)}$ and $H_w^{(k)}$ have to be chosen such that the set $W^{(k)}$ is a set of linearly independent functions.

Choosing polynomials for $G_b^{(i)}$, $H_b^{(i)}$, $G_w^{(k)}$ and $H_w^{(k)}$

One possibility is to choose polynomials of certain degrees for the functions $G_b^{(i)}$, $H_b^{(i)}$, $G_w^{(k)}$ and $H_w^{(k)}$, because clever choice of the polynomials easily yields a set of linearly independent functions.

In order to succeed in this matter, we follow three steps:

1.) split i into two separate numbers i_ξ and i_η .

2.) split k into two separate numbers k_ξ and k_η .

3.) use these numbers to choose
$G_b^{(i)}(\xi)$ to be a polynomial in ξ of i_ξ-th degree
$H_b^{(i)}(\eta)$ to be a polynomial in η of i_η-th degree
$G_w^{(k)}(\xi)$ to be a polynomial in ξ of k_ξ-th degree
$H_w^{(k)}(\eta)$ to be a polynomial in η of k_η-th degree

Patterns for splitting i and k into separate numbers :

For our experiments with the polynomial-Ansatz we used two different patterns of splitting i as well as k into two separate numbers.

In the following, these two patterns will be introduced and named for the purpose of further reference.

First pattern (psI) :

k , i	$\Longrightarrow$	k_ξ, i_ξ	k_η, i_η
1		0	0
2		1	0
3		0	1
4		2	0
5		1	1
6		0	2
7		3	0
8		2	1
9		1	2
10		0	3
11		4	0
12		3	1
$\vdots$		$\vdots$	$\vdots$

Second pattern (psII) :

k , i	$\Longrightarrow$	k_ξ, i_ξ	k_η, i_η
1		0	0
2		1	0
3		0	1
4		1	1
5		2	0
6		2	1
7		0	2
8		1	2
9		2	2
10		3	0
11		3	1
12		3	2
$\vdots$		$\vdots$	$\vdots$

Use of i_ξ, k_ξ, i_η and k_η for defining the functions $E^{(i)}$ and $W^{(k)}$:

For our experimental investigations, we employed the following four patterns which make use of the split numbers i_ξ , i_η , k_ξ , k_η in order to define $E^{(i)}$ and $W^{(k)}$.

First pattern (ppI) :

$$G_b^{(i)} = \xi^{i_\xi} \quad H_b^{(i)} = (\eta - gr(\xi))^{i_\eta}$$

$$G_w^{(k)} = \xi^{k_\xi} \quad H_w^{(k)} = (\eta - gr(\xi))^{k_\eta}$$

Second pattern (ppII) :

$$G_b^{(i)} = \cos(i_\xi \arccos(\xi)) \quad H_b^{(i)} = (\eta - gr(\xi))^{i_\eta}$$

$$G_w^{(k)} = \xi^{k_\xi} \quad H_w^{(k)} = (\eta - gr(\xi))^{k_\eta}$$

Third pattern (ppIII) :

$$G_b^{(i)} = \xi^{i_\xi} \quad H_b^{(i)} = (\eta - gr(\xi))^{i_\eta}$$

$$G_w^{(k)} = \cos(k_\xi \arccos(\xi)) \quad H_w^{(k)} = (\eta - gr(\xi))^{k_\eta}$$

Fourth pattern (ppIV) :

$$G_b^{(i)} = \cos(i_\xi \arccos(\xi)) \quad H_b^{(i)} = (\eta - gr(\xi))^{i_\eta}$$

$$G_w^{(k)} = \cos(k_\xi \arccos(\xi)) \quad H_w^{(k)} = (\eta - gr(\xi))^{k_\eta}$$

At this time, please note that the functions $H_b^{(i)}$ and $H_w^{(k)}$ are always of the form

$$H_b^{(i)} = (\eta - gr(\xi))^{i_\eta}$$

and

$$H_w^{(k)} = (\eta - gr(\xi))^{k_\eta} \quad .$$

The reason for this is the employment of a convenient and fast method for numerically computing the integrals introduced in equations (2.43) through (2.51). A detailed explanation of this method is to be found in the section 3.3 'Numerical Integration'.

Choosing finite elements for $E^{(i)}$ and $W^{(k)}$

If we relax our requirement that $E^{(i)}$ and $W^{(k)}$ have to be polynomials in ξ and η, then we are able to define functions $E^{(i)}$ that are different from zero only in certain intervals, $[a_E^{(i)}, c_E^{(i)}]$, and functions $W^{(k)}$ that are different from zero only in certain intervals, $[a_W^{(k)}, c_W^{(k)}]$. Hence we require that

$$\bigcup_i^N [a_E^{(i)}, c_E^{(i)}] = [-1, 1] \quad ,$$

$$\bigcup_k^N [a_W^{(k)}, c_W^{(k)}] = [-1, 1] \quad ,$$

and that adjacent functions should preferably overlap each other.

This has several advantages:

1.) Only overlapping pairs (k,i) of basis and weight functions contribute to the matrix $\bar{\mathbf{L}}^{(k,i)}$ (that is, pairs (k,i) which satisfy

$$[a_E^{(i)}, c_E^{(i)}] \cap [a_W^{(k)}, c_W^{(k)}] \neq \emptyset \quad),$$

and we can say that $\bar{\mathbf{L}}^{(k,i)} = 0$ for all other pairs (k,i) of basis and weight functions. If the functions are chosen cleverly, the matrix will be extremely sparse.

2.) The integrations concerning a pair (k,i) must be performed only for the interval $[a_E^{(i)}, c_E^{(i)}] \cap [a_W^{(k)}, c_W^{(k)}]$ and not for the whole length of the wave tank. This saves time, and smaller intervals require less computing time.

3.) The functions modelling the flow at the left end of the wave tank can be chosen such that they are decoupled from the functions modelling the fluid flow at the right end of the wave tank. This might be of huge advantage, because our test runs employing polynomials showed considerable disturbances at the right boundary($\partial\Omega_3$) as soon as the action started at the left boundary ($\partial\Omega_1$). We know that this is not true in praxis.

One disadvantage is that many more basis functions are required in order to obtain reasonable results than were required when polynomials were used.

Since the topic of finite elements is very extensive, we have established a separate chapter for the related considerations. Please refer to chapter 5 for detailed explanations on the finite element idea.

3.2 What does $\bar{V}$ look like ?

Our considerations concerning the special function $\bar{V}$ are based on the two assumptions

$$\frac{dgr}{d\xi}(\xi = -1) \;=\; 0 \tag{3.11}$$

and

$$\frac{dgr}{d\xi}(\xi) \;=\; very\;\; small \qquad \forall\, \xi \,\epsilon\, I_{gr} \;=\; [-1, a] \tag{3.12}$$

where a is a number between -1 and 1.

These two assumptions on $gr(\xi)$ are very realistic. Assumption (3.11) states that the ground at the wave board should be flat. Clearly, this is satisfied in the wave tank. Requirement (3.12) states that the ground does not vary too much in a certain region I_{gr} adjacent to the wave board. The wave tank staisfies this condition, too.

The requirements on $\bar{V}$ itself are outlined in chapter 2. Conditions (2.16) and (2.17) state that the equation

$$\frac{\partial \bar{V}_x}{\partial \xi} \;+\; A\,\frac{\partial \bar{V}_z}{\partial \eta} \;=\; 0 \tag{3.13}$$

has to be satisfied and that

$$\bar{V}\; must\; satisfy\; the\; conditions\; at\; the\; boundaries\; \partial\Omega_1\;, \partial\Omega_2\; and\; \partial\Omega_3\;. \tag{3.14}$$

For example, the function

$$\bar{V} = \begin{pmatrix} (\eta - gr(\xi))(\frac{1}{4}\xi^3 - \frac{3}{4}\xi + \frac{1}{2}) \\ \\ -\frac{1}{A}[(\eta - gr(\xi))^2(\frac{3}{8}\xi^2 - \frac{3}{8}) - (\eta - gr(\xi))\,gr'(\xi)(\frac{1}{4}\xi^3 - \frac{3}{4}\xi + \frac{1}{2})] \end{pmatrix} \cdot \frac{V_0(t)}{U_0} \qquad (3.15)$$

would satisfy the two conditions (3.13) and (3.14) . Here, $V_0(t)$ is the velocity of the wave board at the height H. We see that at $\xi = -1$ (denoting the left boundary) the velocity $\bar{V}$ is equal to the velocity of the wave board $\bar{V}_W$, since it is easily verified that $\bar{V}_z = 0$ and that $\bar{V}_x$ rises linearly as η increases.

The functions $\bar{V}_x$ and $\bar{V}_z$, evaluated at $\eta = 1$, are plotted in the figure below. For this plot, we set $gr(\xi) = 0$.

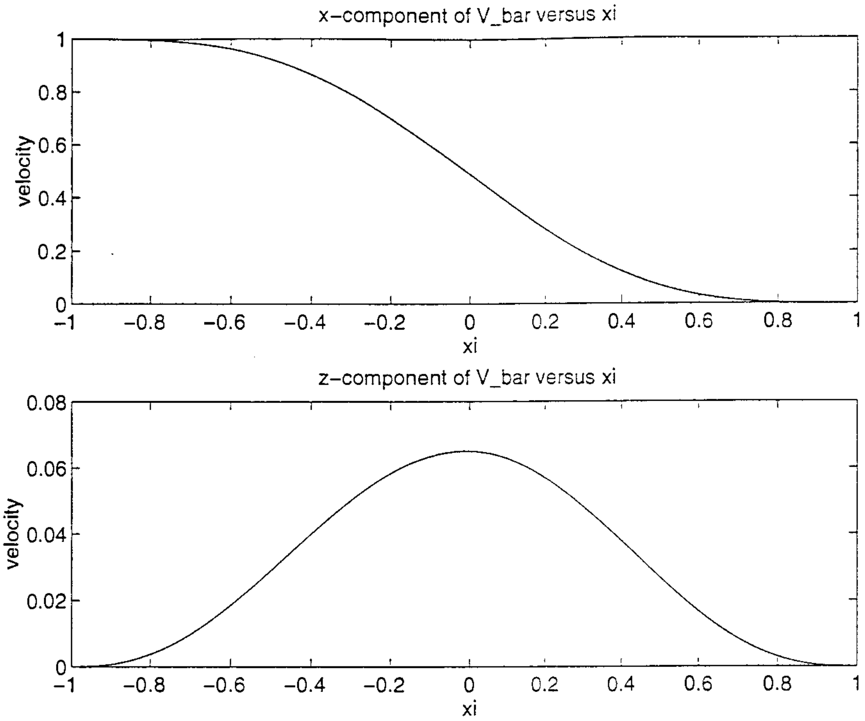


figure 3.1: Function V evaluated at the surface

The following observation shows that (3.13) is not the best way to set up $\bar{V}$:

Suppose the initial velocity of the water throughout the wave tank is equal to zero and the wave board starts to move at the present instant. Performing one time step and using the polynomial model, we computed a velocity distribution after one time step as shown below for $\eta = 1$.

Using 25 basis functions:

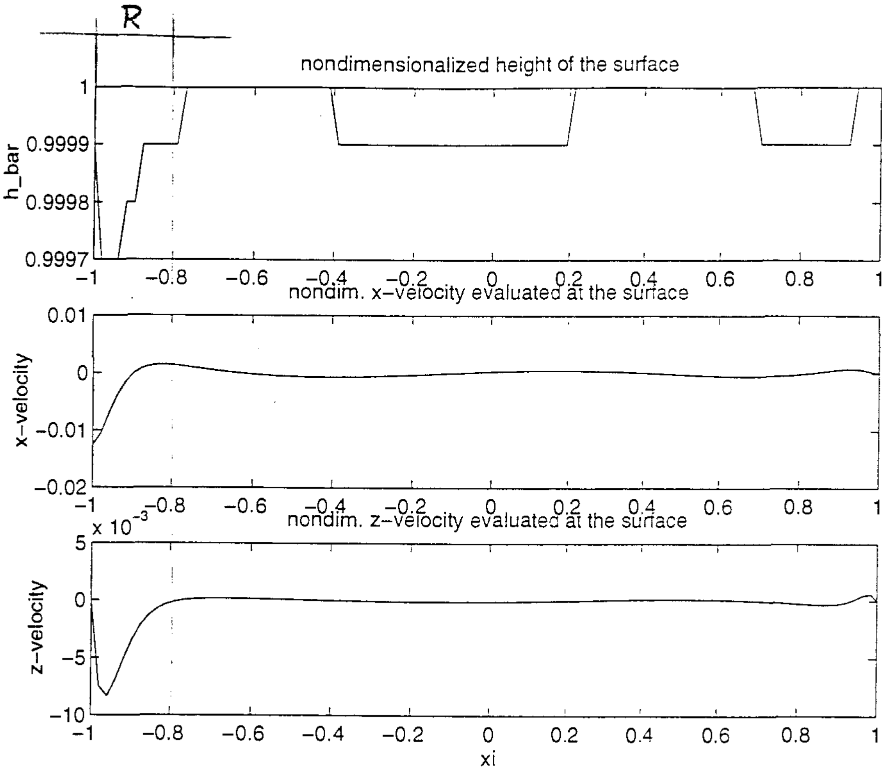


figure 3.2: Completion of one time step - 25 basis functions

Using 36 basis functions:

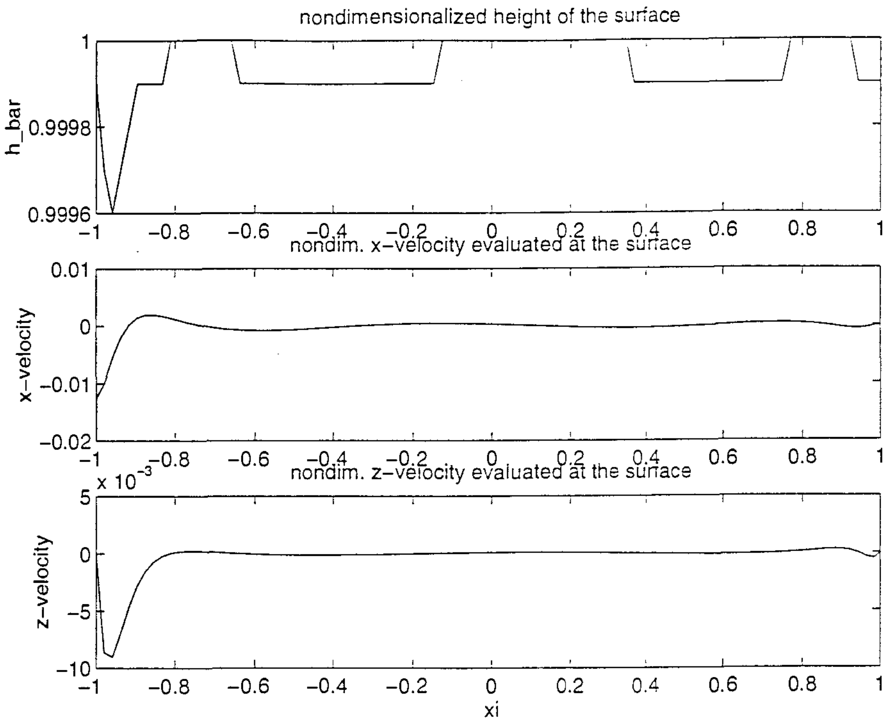


figure 3.3: Completion of one time step - 36 basis functions

Using 49 basis functions:

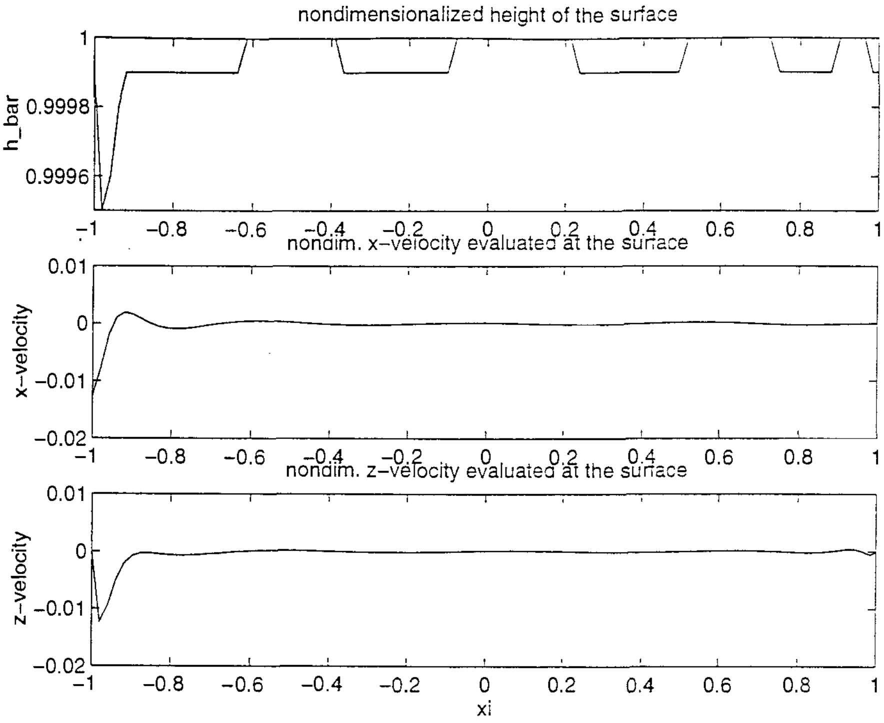


figure 3.4: Completion of one time step - 49 basis functions

We see that the region where the fluid has nonnegligible velocity is at the left boundary of the wave tank (where the board moves). When more basis and weight functions are used, this region is squeezed the closer to the left wall. So we conclude that it would be wise to choose a function V that looks similar to the velocity shape shown, for instance, in the attempt with 49 basis and weight functions.

Following this idea, we obtained convincing results. On one hand, using the function in (3.15) for V and using 25 basis functions, the model was too weak to compute waves that actually run down the wave tank, unless the wave board was stopped. Rather, it happend that the waves, produced by a forward movement of the wave board, were completely swallowed by the subsequent backward movement. The reason is that the region R marked in figure 3.2 was too broad. Even with 36 basis functions we obtained almost the same disaster - no wave running down the tank; and 49 basis functions consumed so much computation time that we did not succeed in calculating any result for this number of basis functions.

On the other hand, if we choose a function V that looks like the following picture (velocities evaluated at $\eta = 1$) , then the model was able to produce waves which actually were running down the wave tank without being swallowed by the subsequent backward movement of the wave board.

All our results, exhibited in chapter 6, were computed by using this special choice of V.

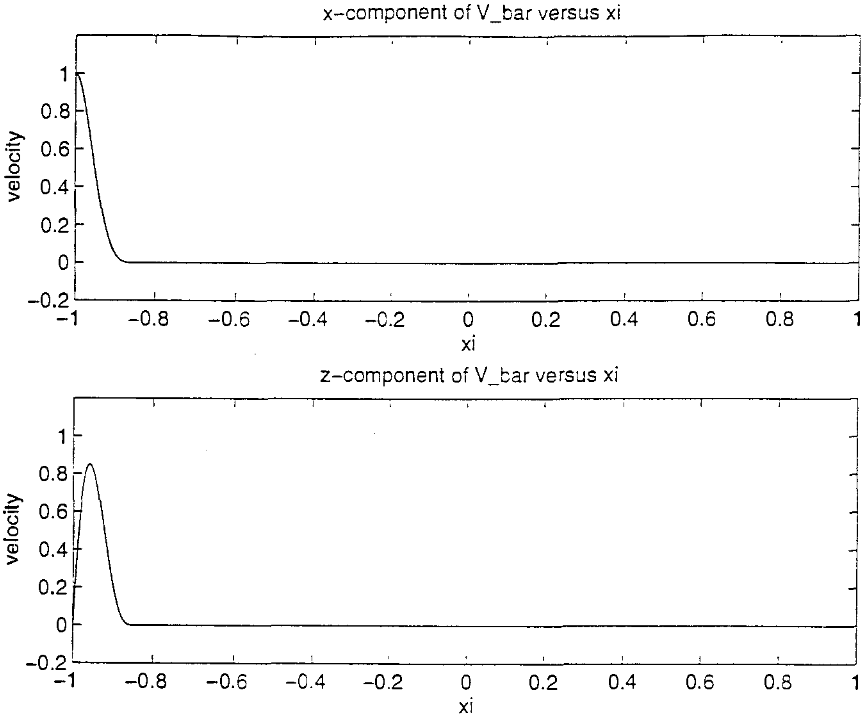


figure 3.5: A more sophisticated kind of V

Thus, a more general set-up for the function $\bar{V}$ would be

$$\bar{V} = \begin{pmatrix} (\eta - gr(\xi))\, f(\xi) \\ \\ -\frac{1}{A}\left[\, \frac{1}{2}(\eta - gr(\xi))^2 f'(\xi) \; - \; (\eta - gr(\xi))\, gr'(\xi)\, f(\xi) \right] \end{pmatrix} \cdot \frac{V_0(t)}{U_0} \, . \tag{3.16}$$

We demand that the function $f(\xi)$ satisfy four requirements.

$$f(\xi) = 0 \quad iff \quad \xi \notin I_{gr} \tag{3.17}$$

$$f(\,\xi = -1\,) = 1 \tag{3.18}$$

$$f'(\,\xi = -1\,) = 0 \tag{3.19}$$

$$f(\xi) \ \textit{must be continuous} \tag{3.20}$$

In combination with condition (3.12), condition (3.17) amounts to neglecting the term $(\eta - gr(\xi))\,gr'(\xi)\,f(\xi)$ in the function (3.16), since either $gr'(\xi)$ is very small (due to (3.12)) or $f(\xi) = 0$ (due to (3.17)).

This, finally, leads to the general setup for V :

$$\bar{V} = \begin{pmatrix} (\eta - gr(\xi))\,f(\xi) \\ \\ -\frac{1}{A}\,\frac{1}{2}(\eta - gr(\xi))^2\,f'(\xi) \end{pmatrix} \cdot \frac{V_0(t)}{U_0} = \begin{pmatrix} \bar{V}_x \\ \\ \bar{V}_z \end{pmatrix} \tag{3.21}$$

Here we see that a power of the term $(\eta - gr(\xi))$ is a factor of both $\bar{V}_x$ and $\bar{V}_z$. In section 3.3, we will see that this is a necessary property of the function $\bar{V}$ in order to employ a fast algorithm for computing complicated integrals.

3.3 Numerical Integration

According to section 2.3.3, we see that we have to solve several integrals over the whole region Ω. The fact that Ω is time dependent (it changes with every time step) requires recomputing all these integrals after each time step. This makes the whole algorithm very slow, since it is the computation of the integrals that is most time consuming.

Thus, it is very important to determine how to solve the concerned integrals efficiently.

3.3.1 Analytical solution of the occuring integrals

Solving the integrals analytically would require
1.) knowing the surface in terms of an algebraic function in ξ and
2.) obtaining integrands that are analytically integrable for all integrals exhibited in section 2.3.3 . However, refering to section 3.3, our modelling of the surface is done by discrete points and not by algebraic functions. Moreover, even if we use ordinary polynomials for the functions $E^{(i)}$ and $W^{(k)}$, the integrands become complicated rather than analytically solvable. Thus, numerical solution of the integrals seems to be a much more convenient way than to perform the integrations by hand.

3.3.2 A simple way to solve occurring integrals numerically

The easiest way to solve the integrals numerically would be to divide Ω by a grid into little rectangles. For instance, by the following:

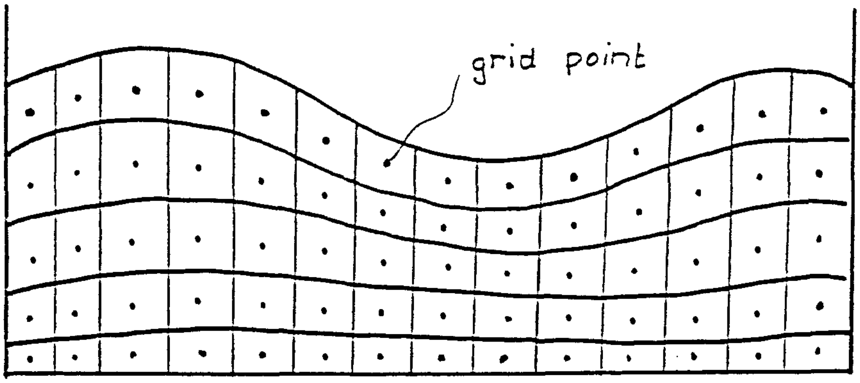


figure 3.6: Grid for computing the Riemann sum

Then the numerical integration would consist of the two steps:

1.) compute the values of the integrand at every grid point and multiply it by the area of the little rectangle surrounding it.

2.) sum up over all little rectangles.

This method is very slow and it would consume too much time if one would like to

compute the integrals very accurately.
Here a convincing example. One can imagine that if the wave length of the waves is 2 meters then the distance of the grid points need to be at least 0.5 meters in order to compute reasonable approximations for the integrals. This would mean we had 200 grid points in x-direction and maybe 20 gridpoints in z-direction, which would give a total of 4000 grid points distributed over all the wave tank. Employment of (psI) and (ppI) with 25 basis functions caused a computation time of one hour on a SUN SPARC 5 station for one time step (ignoring that this blew up the allocated memory considerably).

3.3.3 A fast solver for the occurring integrals

In order to develop a new method to solve the integrals, we considered the integrands very carefully. Taking into account the properties of the general set-up for $\bar{V}$ as discussed at the end of section 3.2 and taking furthermore into account that the functions $H_b^{(i)}$ and $H_w^{(k)}$ have the form

$$H_b^{(i)} = (\eta - gr(\xi))^{i_\eta}$$

and

$$H_w^{(k)} = (\eta - gr(\xi))^{k_\eta} ,$$

then we noticed that all integrals mentioned in section 2.3.3 have the form:

$$\int_\Omega D(\xi) \cdot (\eta - gr(\xi))^c \, d\eta \, d\xi \quad , \tag{3.22}$$

where $D(\xi)$ is a function that depends only on the choice of the functions $G_b^{(i)}$ and $G_w^{(k)}$. The exponent c varies from integral to integral, too. If we know c, then the integral (3.14) can be simplified as follows:

$$\begin{aligned} \int_\Omega D(\xi) \cdot (\eta - gr(\xi))^c \, d\Omega &= \int_{\xi=-1}^{1} \int_{\eta=gr(\xi)}^{\bar{h}(\xi)} D(\xi) \cdot (\eta - gr(\xi))^c \, d\eta \, d\xi \\ &= \int_{\xi=-1}^{1} \tfrac{1}{c+1} D(\xi) \cdot (\bar{h}(\xi) - gr(\xi))^{c+1} \, d\xi \end{aligned} \tag{3.23}$$

This allows us to reduce the two dimensional grid mentioned in the beginning of this section to a one dimensional grid along the x-axis only.

Now we can gain the same accuracy (which needed 4000 grid points in the two dimensional case) with only 200 grid points. This increases the speed of the computations by, roughly, a factor of 20.

Chapter 4: Several Problems

In this chapter, we will exhibit some problems that occured while investigating our algorithms.

One problem arose due to the nature of fluid flow: that is the phenomenon of the water climbing up and down the wall when the fluid is strongly excited by the wave board. This problem and a suggested solution are explained in detail in section 4.1 .

A second problem occurred due to the numerical algorithms themselves. The condition number of the matrix $\bar{\mathbf{L}}^{(k,i)}$ requires a smaller number of basis functions than we would like. This very annoying problem is outlined in section 4.2 .

A third problem, concerning the speed of the numerical computations, is presented in section 4.3 .

4.1 A Global Problem

4.1.1 Description of the Problem

Let us, again, consider the differential equation for the surface motion (compare with chapter 2):

$$\frac{\partial h}{\partial t} + u\frac{\partial h}{\partial x} + v\frac{\partial h}{\partial y} = w \quad .$$

For the two dimensional case (which is the only one to be considered here), this equation reduces to

$$\frac{\partial h}{\partial t} + u\frac{\partial h}{\partial x} = w \quad , \tag{4.1}$$

which is valid for $\partial\Omega_s - \partial\Omega_1 - \partial\Omega_3$. It cannot be valid at the points $\partial\Omega_s \cap \partial\Omega_1$ and $\partial\Omega_s \cap \partial\Omega_3$ where the surface intersects with the other parts of the boundary.

Here are two examples which illustrate this fact:

1.) Let us assume that equation (4.1) is valid even at the point $\partial\Omega_s \cap \partial\Omega_1$. Let us furthermore assume that the wave board does not move. The boundary conditions set forth by the Stokes no slip condition require

$$u = 0 \quad and \quad w = 0 \quad at \;\; \partial\Omega_1 \quad .$$

So, the surface (if (4.1) were valid at $\partial\Omega_1$) would actually be described by the equation

$$\frac{\partial h}{\partial t} + 0\frac{\partial h}{\partial x} = 0$$

which means that

$$\frac{\partial h}{\partial t} = 0 \quad .$$

So,

$$h = constant$$

The illustration of this result is that the surface does not move up and down at the wall.

This, indeed, contradicts our daily experience. For example, when swimming in a swimming pool, we usually notice a visible rise and fall of the surface directly at the wall at the instant when waves (caused by our swimming) gently hit against the wall.

2.) Now let us assume the wave board moves. If the differential equation (4.1) were valid at the point $\partial\Omega_s \cap \partial\Omega_1$, then (4.1) would reduce to

$$\frac{\partial h}{\partial t} = -(V_W)_x \frac{\partial h}{\partial x}$$

at the point $\partial\Omega_s \cap \partial\Omega_1$ (since w is zero directly at the board and u is equal to the x-component of V_W).

Now imagine that $(V_W)_x > 0$ (meaning the wave board moves the water forward) and the surface close to $\partial\Omega_1$ looks like the picture below.

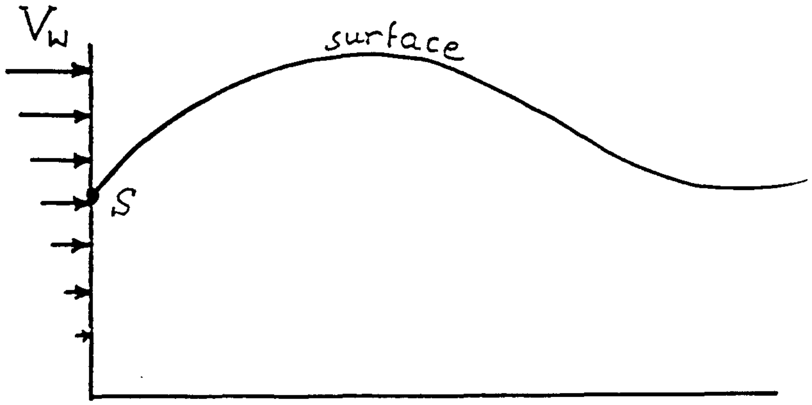


figure 4.1: Exampled state of motion

From the picture we conclude that $\frac{\partial h}{\partial x} > 0$. According to our differential equation above, we clearly have

$$\frac{\partial h}{\partial t} < 0$$

and thus the surface would fall at $\partial\Omega_1$.

But again, this contradicts our daily experience. Carefully observing the wave tank in action, we notice that in this situation the water is more likely to rise than to fall. However, if differential equation (4.1) were valid at $\partial\Omega_1$ then a rising surface in that situation would be impossible.

4.1.2 What is the mechanism for rising and falling surface at the boundary?

Since we deal here with a viscous fluid, the Stokes no slip condition is indeed valid for $\partial\Omega_1$ and $\partial\Omega_3$. So, the instantaneous velocity of the fluid particles at the wall is actually zero. But also, arbitrarily close to the wall, the velocity in z-direction (w) must be very large. Otherwise a rise or fall of the surface would not be possible. So, it must be true that the term $\frac{\partial w}{\partial x}$ close to the wall is extremely large. (FACT 1)

Another fact is that the angle α as shown in the picture

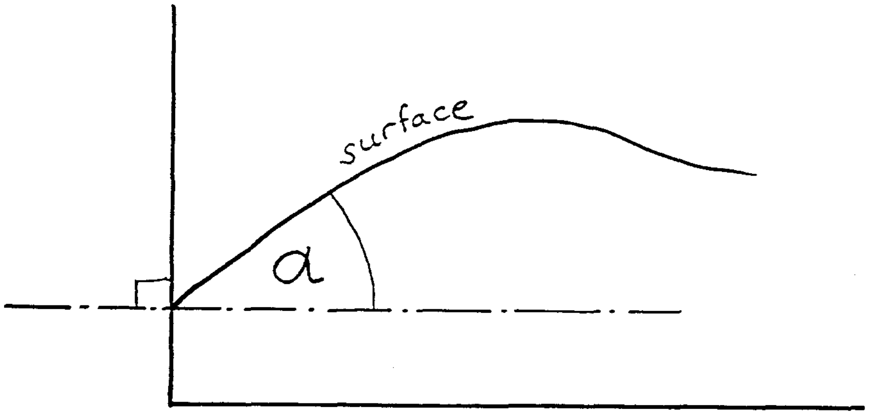


figure 4.2: Wetting angle

cannot be larger than a certain value α_w, the so-called wetting angle, which is due to adhesive and cohesive forces at the wall. (FACT 2)

(FACT 1) and (FACT 2) lead to the hypothesis that the rising and falling of the surface depends mainly on the adhesive and cohesive forces and on the fact that the vorticity of the fluid does not vanish at the wall.

So. the mechanism of surface motion is the following. If α rises quickly because of (FACT 1) and even exceeds the wetting angle α_w then, due to (FACT 2), the surface at the wall must either rise or fall in order to make α smaller. We notice that this surface motion is possible due to the huge vorticity at the wall.

4.1.3 How to model the rise and fall mechanism numerically

For the numerical programming, we employ the same mechanism that nature does. After each time step the value of the angle α will be calculated. If this value becomes strictly larger than the wetting angle α_w, we rise or lower the surface point at the wall until the angle is equal to the wetting angle.

Using the polynomial model for the function $E^{(i)}$ with only a few basis functions (for example 25 or 36 basis functions), one observes quickly that the term

$$\frac{\partial w}{\partial x} = \frac{\partial V_z}{\partial x} + \sum_{i=1}^{N} b_i \frac{\partial \mathbf{e}_w^{(i)}}{\partial x}$$

is not large enough to model rising or falling surface at the wall. That means we need to support this process somewhat by introducing more sophisticated functions $E^{(i)}$ (refer to equation (3.4)) and $W^{(k)}$ (refer to equation (3.9))

$$E^{(i)} = (\ \xi^{a} - 1\ )^{2}(\ \eta - gr(\xi)\ )^{2} \cdot G_{b}^{(i)}(\xi) \cdot H_{b}^{(i)}(\eta) \tag{4.2}$$

and

$$W^{(k)} = (\ \xi^{a} - 1\ )^{2}(\ \eta - gr(\xi)\ )^{2} \cdot G_{w}^{(k)}(\xi) \cdot H_{w}^{(k)}(\eta) \quad . \tag{4.3}$$

Please note the power of ξ, 'a': we require that a must be an even number. The larger the number a, the larger is the term $\frac{\partial w}{\partial x}$ at the wall in our computed results, and the more realistic is the computed surface while using only few basis functions. Note that we did not change the model for computing $\mathbf{v}$. The use of the Galerkin method remains undisturbed by the introduction of a.

We used $a = 20$ for our test runs with few basis functions. If one uses more basis functions then the number a is of less importance, since then the large number of basis functions is flexible enough to produce a sufficently large $\frac{\partial w}{\partial x}$ at the boundary. For instance, this is the case when we employ the finite element model with about 500 basis functions.

Conclusion: The surface correction mechanism produces very realistic results. chapter 6 ('Results') demonstrates the difference in the numerical outputs for test runs with and without surface correction. For the results without surface correction one can observe the phenomena discussed at the beginning of this section. For the results with surface correction one can notice the rise and fall of the surface.

4.2 A special problem: Matrix condition number of $\bar{\mathbf{L}}^{(k,i)}$

Let us consider the following graph. It shows the condition number of the matrix $\bar{\mathbf{L}}^{(k,i)}$ in the 2-norm versus the number of basis functions N. The values were computed for a flat surface and for realistic numbers Re, St, Fr, and A. The type of basis and weight functions used is the polynomial model (ppIV) and the associated pattern of splitting i and k is (psI).

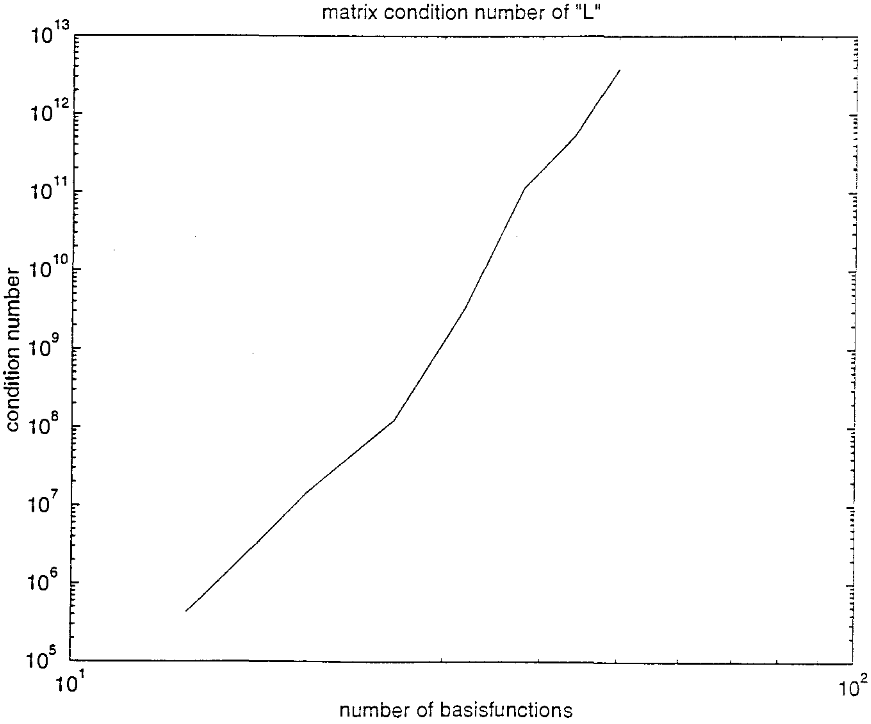


figure 4.3: Matrix condition number for polynomial model

It is easy to recognize that the condition number of the matrix rapidly increases as the number of basis function increases. The condition number is 10^{12} when 50 basis functions are used.

However, we know from standard numerical algebra that reliable solutions of a linear system of equations

$$\mathbf{A} \cdot x = \mathrm{b}$$

are obtained only if the condition number is sufficiently small. In our case, we implemented our program in MATLAB, which works with 8-byte-real numbers. Thus, a condition number of 10^{12} is very large. Thus, we can expect that the solution for the system $\mathbf{A} \cdot x = \mathrm{b}$ has only three reliable decimal places. MATLAB itself does not perform a Gaussian elimination to obtain the solution x if the matrix condition number of $\mathbf{A}$ is larger than 10^{14}.

Therefore, we are not able to use more than 70 basis functions if the polynomial model (refer to section 3.1.3) is employed. However, there might be cases where one needs to use more basis functions, for instance if the waves are very short and are distributed over a significant part of the wave tank.

For these cases, there are two possible solutions:

1.) Use real numbers with more than 8 byte memory size. This can be done on special kinds of powerful workstations, but it will probably require more computation time. This, indeed, does not give us any information about the convergence of the polynomial Ansatz. So, using more basis functions does not actually imply that the results become better.

2.) Search for other types of reasonable basis functions. We tried to introduce another type of basis functions (finite elements) in order to work around the matrix condition problem. Refer to chapter 5, where the idea of the finite elements is introduced in detail. There, we shall consider the matrix condition numbers again and come up with more convenient results.

4.3 A problem due to hardware and software

We already mentioned that the algorithms are computationally very expensive. For example, using the finite element model (which will be presented in chapter 5) one time step requires approximately $2 \cdot 10^9$ floating point operations, depending on how many iteration steps are necessary to complete a whole time step.

A workstation that performs $2 \cdot 10^7$ floating point operations per second would take 100 seconds to perform one time step. This means, with this computer we would perform 36 timesteps per hour. For computing realistic results the time steps should be small (about 0.01 ... 0.05 seconds). Thus, for a simulation of 30 seconds, 1500 time steps have to be performed. With the mentioned computational speed of the workstation, the whole simulation process would take two days!

We implemented the program in MATLAB. If one intends to apply WHILE- or FOR-loops in MATLAB, then MATLAB itself drastically reduces the rate of operations per second. Even if we searched patiently for ways to increase the speed of the program, we could not avoid using WHILE-loops. This resulted in a rate of appoximately 10^6 floating point operations per second on a SUN SPARC station 5. So, MATLAB dropped the rate by a factor of 10 to 20. For the example above we would have to wait for about a month!

Moreover, in MATLAB, to avoid WHILE-loops and the loss of speed, one has to program in matrix- or vector-form. For instance, in order to perform the numerical integration of the function $f(x)$ over the interval $[\,a, b\,]$, one often applies the Riemann-sum

$$\int_a^b f(x)\,dx \;\approx\; \sum_{i=1}^{N} f(x_i)\cdot(\,x_i - x_{i-1}\,)$$

In MATLAB, however, it is not wise to implement this Riemann-sum as a loop. To increase the speed, one has to create a vector $\mathbf{v}_1$ of length N with all the values $f(x_i)$ and a vector $\mathbf{v}_2$ of length N with all the values $(\,x_i - x_{i-1}\,)$ and, in order to compute the Riemann-sum, calculate the scalar product $\mathbf{v}_1^T \cdot \mathbf{v}_2$.

In fact, this method is relatively fast, but it is very memory extensive. In general, the rule 'the faster a MATLAB-program the more memory must be allocated' is true. One can imagine, how much computer memory will be required to compute all integrals mentioned in chapter 2. From time to time, our experimental runs allocated the whole memory of a workstation, which made either the program break down or other machine users completely frustrated about the slowness of their processes. Nevertheless, this was the only way to make the program run fast enough.

One way out of this situation could be implementing the whole algorithms in FORTRAN or in C++. This would increase the speed condiderably, but one might lose the overview of the whole program. One of the important advantages of MATLAB is the ability to program in a very compact manner. For our purposes, this property of MATLAB was of considerable importance.

Chapter 5: A Finite Element Model

Considering the results exhibited in chapter 6, one remarks that using polynomials for the functions $E^{(i)}$ and $W^{(k)}$ causes several nonnegligible problems, two of which are:

1.) As discussed in chapter 4, we cannot use more than 50 basis functions. But we can imagine a case where more than 50 basis functions would be necessary: for instance, when the wave length of the water waves is about 2 meters, and waves are distributed throughout the wave tank, then we have 50 waves in the wave tank. At the surface the sign for the velocity in the z-direction would switch about 100 times. This kind of velocity-field is impossible to be modeled using only 50 basis functions! So we see that the use of only 50 basis functions is too restrictive. We need more freedom.

2.) Again referring to the results in chapter 6, we notice that the velocity of the water at the right end of the wave tank is disturbed immediately after the wave board starts to move at the left end. These disturbances grow steadily as time increases until they become nonnegligible in certain examples. These disturbances are due to the properties of the polynomials and, in particular, to the fact, that the supports of the polynomials are not locally bounded. However, these disturbances do not coincide with our practical experience; we observe that the water motion in the wave tank is such that the right end is only slightly disturbed (almost invisible motion of the water!) when the wave board starts to move.

Conclusion : We should make an effort to find basis functions that are of finite support and that provide a convenient matrix condition number for even a large number of basis functions.

5.1 The Idea

In order to resolve problem 2.) mentioned above, we will use basis functions with locally bounded support. We introduce the function $E^{(i)}$ given by

$$E^{(i)} = (\xi^{\alpha} - 1)^2 \, (\eta - gr(\xi))^2 \cdot cos[\frac{\pi}{E}(\xi + 1 - offset \cdot E - (i_\xi - 1) \cdot E)] \cdot (\eta - gr(\xi))^{i_\eta} \cdot \chi_{I_{E(i)}} \quad (5.1)$$

where

$E = \frac{2}{nb_x - 1}$ is one half of the length of the intervals $I_{E(i)}$ where the functions $E^{(i)}$ are non-zero.

$I_{E(i)} = [-1 + offset \cdot E + (i_\xi - 1) \cdot E - E \, , \, -1 + offset \cdot E + (i_\xi - 1) \cdot E + E]$
is the interval where $E^{(i)}$ is non-zero.

nb_x is the number of different intervals $I_{E(i)}$ that belong to the current set of basis functions.

$offset$ is a parameter for translating the basis functions along the x-axis. This parameter has no importance yet, but later on we will discover its usefulness.

i_ξ and i_η are (as already mentioned in chapter 3) results of spliting the number i in two separate numbers.

i is the index of the particular basis function under consideration.

Again, as already done in chapter 3, we set up the basis functions as

$$\begin{pmatrix} \frac{\partial E^{(i)}}{\partial \eta} \\ \\ -\frac{1}{A}\frac{\partial E^{(i)}}{\partial \xi} \end{pmatrix} = \begin{pmatrix} \bar{\mathrm{e}}_x^{(i)} \\ \\ \bar{\mathrm{e}}_z^{(i)} \end{pmatrix} = \bar{\mathrm{e}}^{(i)} \quad . \tag{5.2}$$

For the purpose of illustration, here is a plot of the basis functions for $offset = 0$, $nb_x = 15$, $i_\xi = 1...nb_x$ and $i_\eta = 1$:

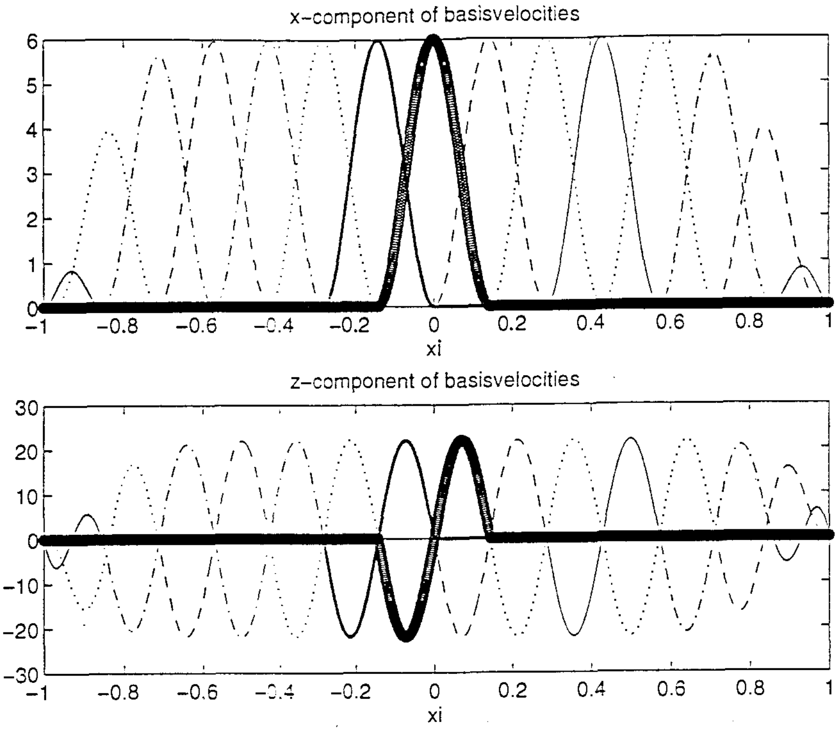


figure 5.1: Basis Functions according to model (5.1)

We see that the interval $I_{E^{(i)}}$ of a certain basis function is divided into two even pieces by the overlapping basis functions.

Likewise, we could set up the function $E^{(i)}$ as

$$E^{(i)} = (\xi^{\alpha} - 1)^2 \, (\eta - gr(\xi)\,)^2 \cdot cos[\,\frac{\pi}{E}(\xi + 1 - offset \cdot E - \frac{2}{3}(i_{\xi} - 1) \cdot E\,)] \cdot (\,\eta - gr(\xi)\,)^{i_{\eta}} \cdot \chi_{I_{E^{(i)}}} \quad (5.3)$$

where

$E = \frac{3}{nb_x - 2}$ is one half of the length of the interval $I_{E^{(i)}}$ where the function $E^{(i)}$ is non-zero.

$I_{E^{(i)}} = [-1 + offset \cdot E + \frac{2}{3}(i_\xi - 1) \cdot E - E\ ,\ -1 + offset \cdot E + \frac{2}{3}(i_\xi - 1) \cdot E + E]$
is the interval where $E^{(i)}$ is non-zero.

All other parameters as described above.

This would result in the following plot of the basis functions (again for $offset = 0$, $nb_x = 15$, $i_\xi = 1 ... nb_x$ and $i_\eta = 1$).

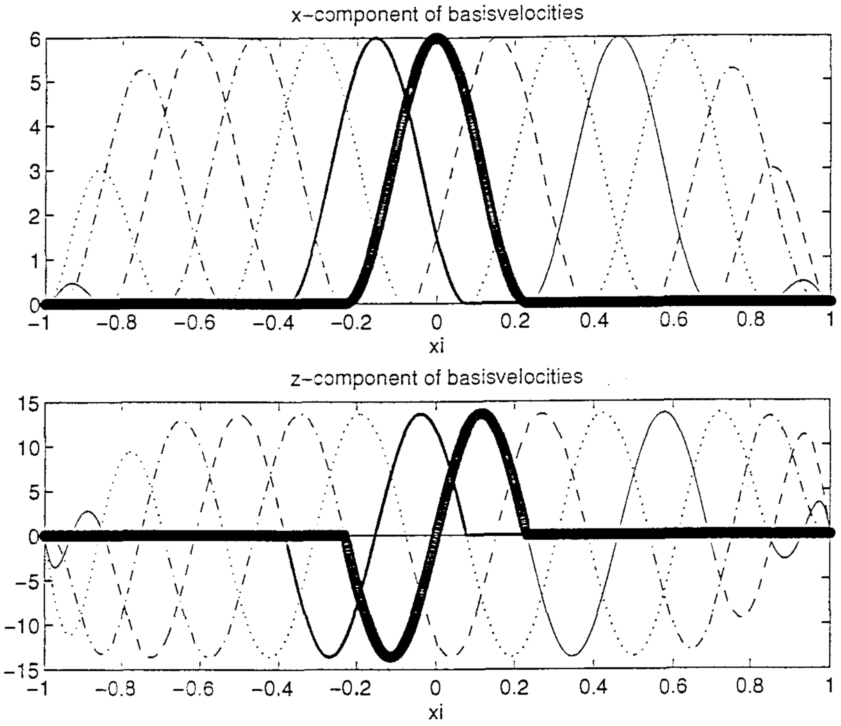


figure 5.2: Basis Functions according to model (5.3)

Here we see that each interval $I_{E(i)}$ is divided into three even pieces by the overlapping basis functions.

In the same fashion as (5.1) or (5.3), we could define

$$E^{(i)} = (\xi^{\alpha} - 1)^2 \, (\eta - gr(\xi)\,)^2 \cdot cos[\frac{\pi}{E}(\xi + 1 - offset \cdot E - \frac{2}{6}(i_{\xi} - 1) \cdot E)\,] \cdot (\,\eta - gr(\xi)\,)^{i_{\eta}} \cdot \chi_{I_{E(i)}} \quad (5.4)$$

Here,

$E = \frac{6}{nb_x - 5}$ is one half of the length of the certain interval $I_{E^{(i)}}$ where the function $E^{(i)}$ is non-zero.

$I_{E^{(i)}} = [-1 + offset \cdot E + \frac{2}{6}(i_\xi - 1) \cdot E - E\ ,\ -1 + offset \cdot E + \frac{2}{6}(i_\xi - 1) \cdot E + E]$
is the interval where $E^{(i)}$ is non-zero.

Below, we exhibit a plot of the basis functions (5.4) (again for $offset = 0$, $nb_x = 15$, $i_\xi = 1 \ldots nb_x$ and $i_\eta = 1$) .

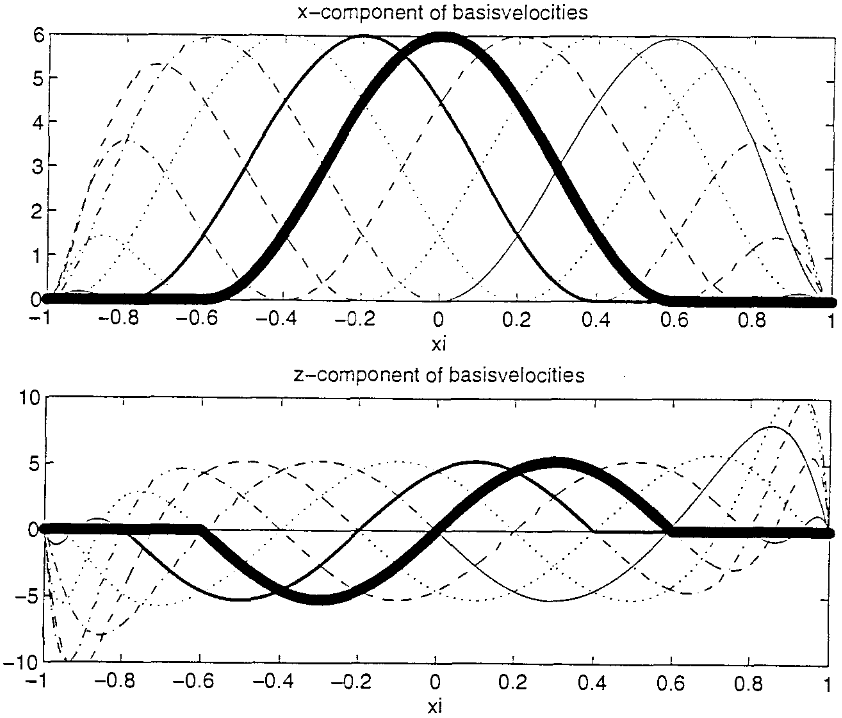


figure 5.3: Basis Functions according to model (5.4)

We see that each interval $I_{E^{(i)}}$ is divided into six even sub-intervals by the overlapping functions.

For all our further considerations, we define

$$W^{(k)} = E^{(i)} \qquad iff \quad i = k \quad . \tag{5.5}$$

Furthermore we introduce the following two numbers.

nb_x is the number of different intervals $I_{E^{(i)}}$ that cover the whole length of the wave tank, and

nb_z is the highest power that can be taken by the $(\eta - gr(\xi))$-term in the equation for $E^{(i)}$.

With these definitions, we may determine a pattern for splitting the numbers i and k into two separate numbers:

$$\begin{aligned} i_\xi &= int(\tfrac{i-1}{nb_z}) + 1 \\ i_\eta &= i - nb_z \cdot i_\xi \\ k_\xi &= int(\tfrac{k-1}{nb_z}) + 1 \\ k_\eta &= k - nb_z \cdot k_\xi \end{aligned} \tag{5.6}$$

At this point, note that the number N of basis functions (weight functions) must be

$$N = nb_x \cdot nb_z \tag{5.7}$$

Thereby, the setting of the functions $E^{(i)}$ and $W^{(k)}$ for the finite element model is completed. We notice that the functions in (5.1), (5.3), and (5.4) are twice differentiable with respect to ξ and η which is sufficient for our purposes.

5.2 Further Investigations on the Models (5.1), (5.3), and (5.4)

One of the problems we had with the polynomial model was the considerably large matrix condition number of the matrix $\bar{\mathbf{L}}^{(k,i)}$. Now, we wish to study how the condition numbers behave when using the models (5.1), (5.3), and (5.4).

In the following pictures, we plotted the matrix condition number versus the number nb_x, always using $nb_z = 5$.

In the plot below we employed (5.1).

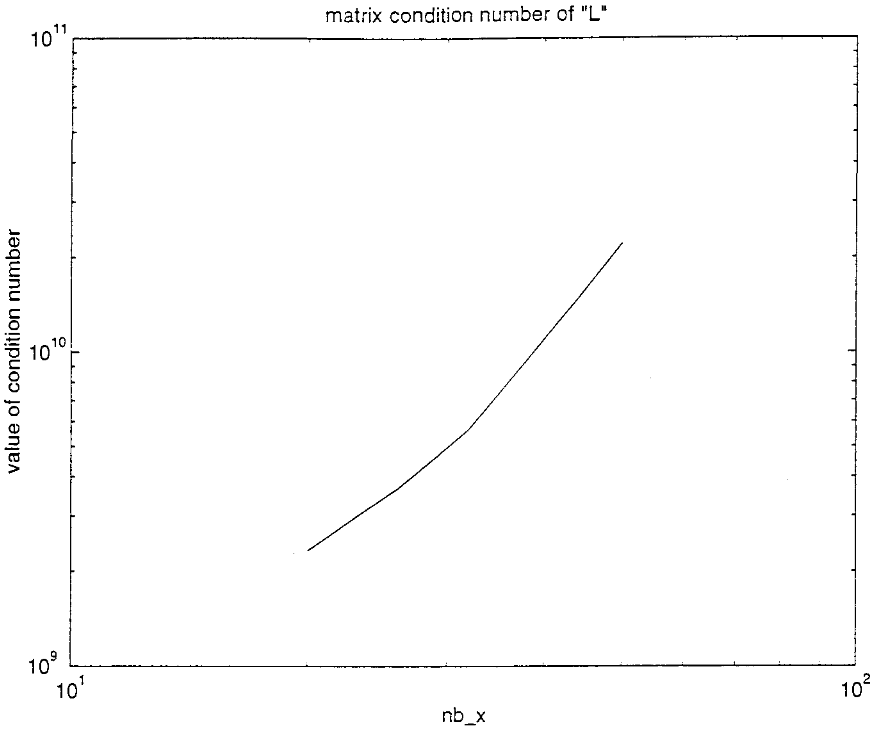


figure 5.4: Matrix condition numbers for model (5.1)

In the next plot we employed (5.3).

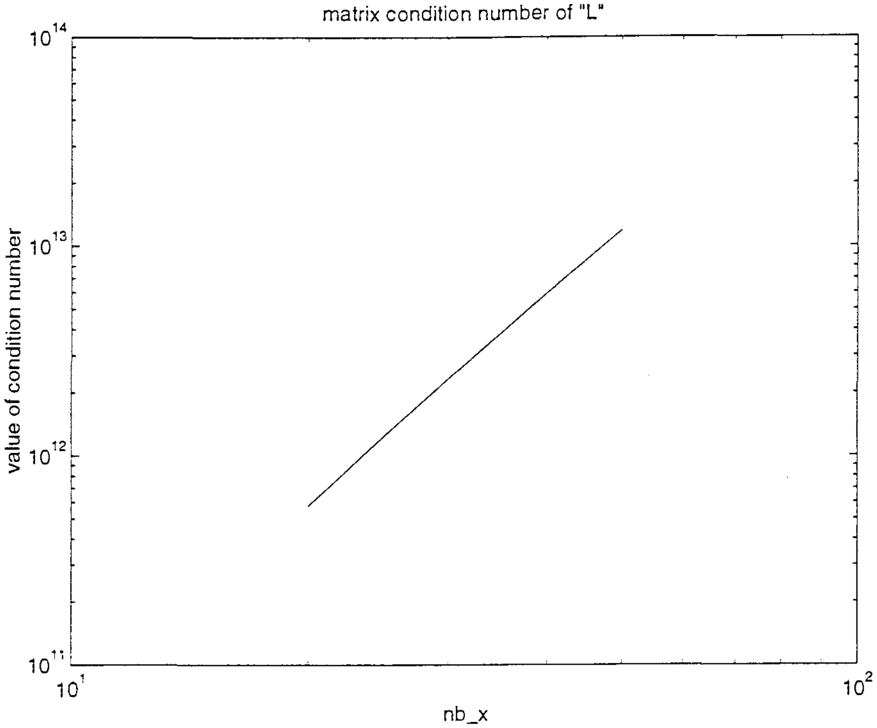


figure 5.5: Matrix condition numbers for model (5.3)

In the next plot we employed (5.4).

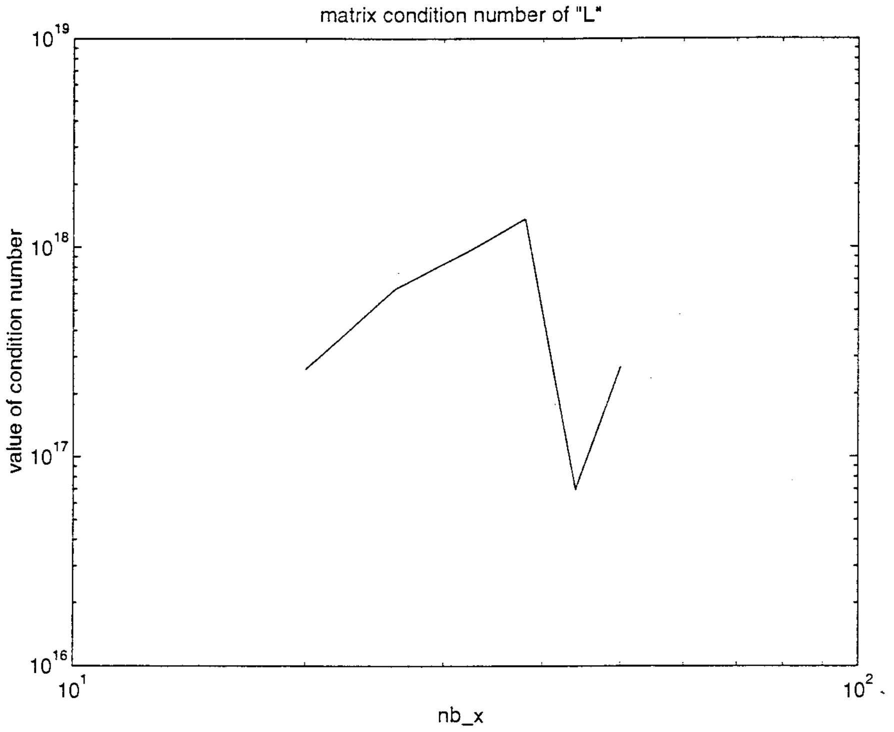


figure 5.6: Matrix condition numbers for model (5.4)

We recognize that model (5.4) would not work for our purposes because the matrix condition number of $\bar{\mathbf{L}}^{(k,i)}$ is too large. However, it could work if we could use 16 byte real numbers.

But (5.1) and (5.3) show suitable numbers up to $nb_x = 100$. So, using 8-byte-real numbers, we can work with up to 500 basis functions! Compared to the polynomial model, this is an improvement by a factor of 10, which means that we can model waves of 2 meters length distributed over the whole length of the wave tank.

5.3 Some Problems with the Finite Element Set-up

Suppose we had $nb_x = 20$ and $nb_z = 1$. Suppose furthermore we had $b_i = i$. Using these assumptions, the following picture shows a plot of the function

$$\bar{\mathbf{v}} = \sum_{i=1}^{N} b_i \, \bar{\mathbf{e}}^{(i)} ,$$

evaluated at $\eta = 1$.

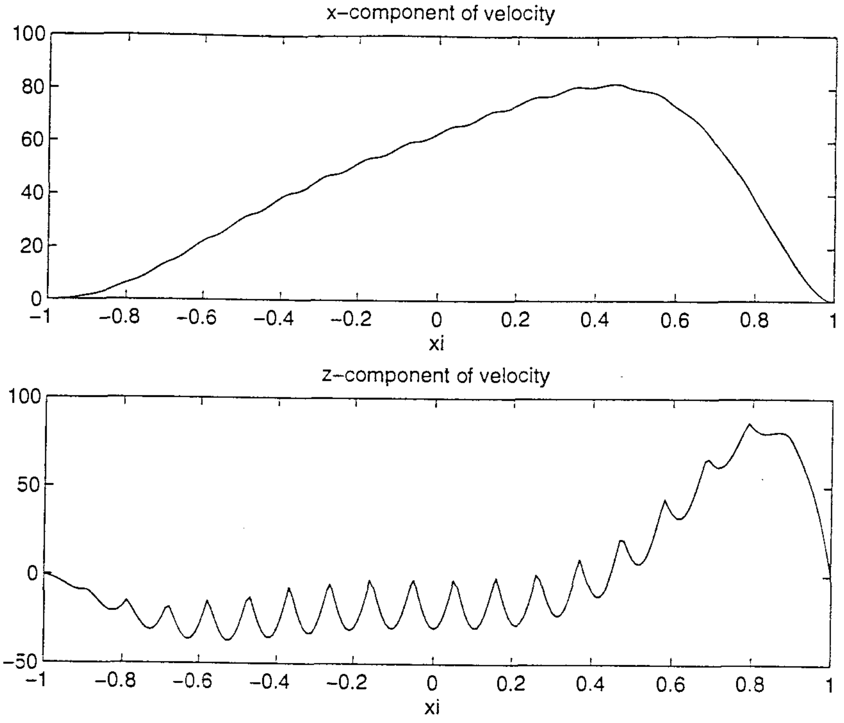


figure 5.7: Example for velocity distribution for model (5.1)

We observe the 'corners' that appear in the z-component of the velocity. These spikes are very large and grow sometimes larger if the velocity $\bar{\mathbf{v}}$ becomes more sophisticated. Some of our experiments with this model (5.1) showed that the final results are not reliable.

A plot using model (5.3) for the same function would show that the corners became smaller, and finally, a plot of the same function using model (5.4) looks like this:

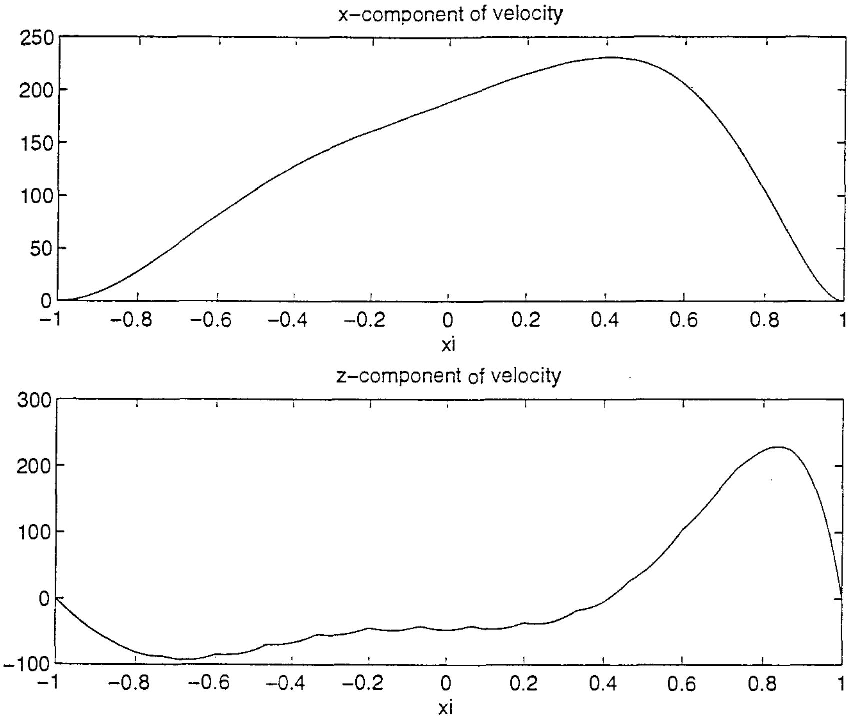


figure 5.8: Example for velocity distribution for model (5.4)

We see that the 'corners' have almost disappeared. The z-component of the velocity seems to be smooth enough such that model (5.4) can be used as a reliable model.

The problem now is the following: on one hand, we cannot use model (5.4) because the matrix condition number is far too large for the model to be employed for our algorithms. On the other hand, models (5.1) and (5.3) cause such large corners that the computations do not give good results.

The way out : A relatively smooth solution (like the one provided by (5.4)) using models (5.1) and (5.3) can be obtained by the following idea.

If there is a set of n approximate solutions to the problem (call them $\bar{\mathbf{v}}_1 \ldots \bar{\mathbf{v}}_n$) then the mean value of these approximate solutions

$$\bar{\mathbf{v}} = \frac{1}{n} \sum_{i=1}^{n} \bar{\mathbf{v}}_i \tag{5.8}$$

must be an approximate solution, too. We make use of this in the following sense.

Using models (5.1) or (5.3), we compute n different approximate solutions to the problem and find (according to (5.8)) the mean value of all these solutions, thereby hoping that the annoying 'corners' will be smoothed out. To find n approximate solutions, we perform the same computations for n different sets of basis functions, which vary only in the parameter $offset$. So we compute one approximate solution using a set of basis functions, then we translate this set of basis functions by a small distance and compute another approximate solution and so forth. The parameter $offset$ will be cleverly chosen such that the 'corners' become smaller as more basis functions are chosen.

Example : Suppose we employ model (5.1). Suppose furthermore we want to compute three different approximate solutions $\bar{\mathbf{v}}_1$, $\bar{\mathbf{v}}_2$, $\bar{\mathbf{v}}_3$. Then these three solutions are computed using three sets of basis functions with the offsets

$$\begin{aligned} offset_1 &= 0 \ , \\ offset_2 &= \frac{1}{6} \ , \\ offset_3 &= \frac{2}{6} \ , \end{aligned}$$

respectively. While each of the three sets of basis functions causes large corners in the approximate solutions $\bar{\mathbf{v}}_1$, $\bar{\mathbf{v}}_2$ and $\bar{\mathbf{v}}_3$, their mean value

$$\bar{\mathbf{v}} = \frac{1}{3} \sum_{i=1}^{3} \bar{\mathbf{v}}_i$$

smoothes out those corners, and thus the mean value solution has the quality of model (5.4). The smoothing of the corners occurs because the three sets of basis functions overlap each other in the same way as the basis functions in model (5.4) .

However, there are still small 'corners' in the computed solution even if the quality of (5.4) is reached. For those examples exhibited in chapter 6 which were computed using the finite element model, one can easily see the ugly 'corners' in the surface which are due to the 'corners' in the z-component of the basis functions rather than to numerical instabilities.

5.4 Increasing Speed by Parallel Computing

When studying the algorithm for the finite element model carefully, one will notice that the whole process is easily parallelizable.

This is a fact that we would like to illustrate here by considering the steps to be completed for computing the matrix $\bar{\mathbf{L}}^{(k,i)}$. Here is an example: Suppose we use model (5.1) and, moreover, suppose we have $nb_x = 10$ and $nb_z = 5$, so that we have 50 basis- and weight functions. Remember that $W^{(k)} = E^{(i)}$ iff $i = k$.
To compute $\bar{\mathbf{L}}^{(k,i)}$ we need to perform the steps below.

Step 1:
1.a) take the first five weight functions ($k = 1,2,3,4,5$) (since their intervals of support $I_{E^{(k)}}$ coincide) and determine the indices of all basis functions overlapping this subset of weight functions (here $i = 1,2,3,4,5$).
1.b) compute the entries of the matrix $\bar{\mathbf{L}}^{(k,i)}$ for the values $k = 1,2,3,4,5$ and $i = 1,2,3,4,5$.

Step 2:
2.a) take the next five weight functions ($k = 6,7,8,9,10$) (again, the intervals $I_{E^{(k)}}$ coincide) and determine the indices of all basis functions overlapping this subset of weight functions (here $i = 1,2,3,4,5,6,7,8,9,10$).
2.b) compute the entries of the matrix $\bar{\mathbf{L}}^{(k,i)}$ for the values $k = 5,6,7,8,9,10$ and $i = 1,2,3,4,5,6,7,8,9,10$.

and so on until

Step 10:
10.a) take the next five weight functions ($k = 46,47,48,49,50$) (again, the intervals $I_{E^{(k)}}$ coincide) and determine the indices all of basis functions overlapping this subset of weight functions (here: $i = 46,47,48,49,50$).

10.b) compute the entries of the matrix $\bar{\mathbf{L}}^{(k,i)}$ for the values $k = 46, 47, 48, 49, 50$ and $i = 46, 47, 48, 49, 50$.

Each of the steps outlined above is very time consuming; and it would take a very long time to perform them each after another in a FOR-loop on a computer with no parallel computing utilities. However, we see that those ten steps are not linked to one another; so the matrix entries can be produced separately, which means they can be computed at the same time if a parallel computer is available. Moreover, the computing time from step to step does not vary much since the same number of floating point operations is required for each step (except for the case when the weight functions are at the boundary; in our example this concerns steps 1,2,9,10). So the process is almost perfectly parallelizable.

In the example above, we could increase the speed by a factor of at most ten if we had a computer with at least ten parallel processors.

For more realistic applications we would use $nb_x = 50...100$ (for instance example 4 and 5 in chapter 6 were computed by using $nb_x = 50$ and $nb_z = 5$). In such a case the speed could be increased easily by factor nb_x provided that a computer with at least nb_x parallel processors is available.

If N_{pr} denotes the number of processors in the computer, then the speed of the computations can possibly be increased by the factor

$$F_{inc} = \frac{parallel\ speed}{sequential\ speed} = \frac{nb_x}{int(\frac{nb_x - 1}{N_{pr}}) + 1} \qquad (5.9)$$

We see that this factor increases as nb_x increases (which means more basis functions could cause a larger factor).

There are, indeed, more possibilities to parallelize the whole computations. For instance, each of the different integrals (2.43) .. (2.51) may be calculated independently from the others, so it may be possible to compute these integrals in parallel. Unfortunately, the problem is that the number of floating point operations might differ considerably from integral to integral; so, in this case, the parallelization would be less effective.

The ideas above have not been set into practice yet. Obviously, setting up the algorithms of this thesis on a parallely working machine requires first to have the machine and then to determine the parallelization that would work best on the machine.

Chapter 6: Results

In this chapter, we exhibit five of the workstation computed results.

1.) The first is an example employing the polynomial model introduced in chapter 3.

2.) The second is an example employing the moving grid surface model. Again, the basis functions were polynomials.

3.) The third example impressively demonstrates the problems of the Stokes no slip condition for the boundary. This result was produced without applying the mechanism for surface correction.

4.) Examples four and five were produced by using the finite element model introduced in chapter 5.

For all examples, we used realistic values for Re, St, Fr and A. The length of the wave tank was assumed to be $L = 100\,m$ and the height of the water was assumed to be $H = 6\,m$. Furthermore, we used $gr(\xi) = 0$ for all ξ.

While the first three examples demonstrate the large number of problems with employing the polynomial model, they also demonstrate that the whole model is able to work and to deliver realistic results. We exhibit these polynomial results since they were the first results we obtained and they motivated further investigations.

The last two examples demonstrate the capacity of the finite elements method to compute results close to reality. At least, our imagination should tell us that the solutions as computed in example 4 or 5 could happen in nature. Unfortunately, the examples do not demonstrate the amount of computation time each required.

The results for each example show a sequence of graphs, each after a certain number of time steps. For each time step, we plotted three graphs:

1.) The first (upper) one represents the shape of the surface in nondimensionalized coordinates.
2.) The second one shows two curves. The solid curve represents the x-component of the nondimensionalized velocity, evaluated at the surface. The dashed curve represents the z-component of the nondimensionalized velocity, evaluated at the surface.
3.) The third (lower) plot shows the tracks of certain molecules of the water as they moved from the beginning of the computations.

The horizontal axis of all three plots is the ξ-axis.

6.1 Example 1

For this example, we started with $\mathbf{v} = 0$ (motionless water) and let the wave board move back and forth periodically, beginning with a small amplitude and increasing it up to a maximum value.

basis and weight functions used: polynomial model
number of basis functions: 36
pattern of splitting i and k: (psI)
frequency of the wave board: one cycle in 2 seconds
max. velocity of the wave board at height $6\,m$: $2\,\frac{m}{s}$
generalized velocity $U_0 = 10\,\frac{m}{s}$
timestep $\Delta t \;=\; 0.02\,s$
plotted results every 50 timesteps (one plot after each second), beginning with timestep 50
We employed the idea of rising and falling surface at the boundary.

Remarks:

The waves that were produced in this example do not look very realistic. This could have two reasons:

1.) The waves almost break, but the surface model does not allow breaking waves.
2.) The resolution of the polynomial model is not high enough to match the (in nature complicated) velocity field and produces a more or less intermediate result.

Furthermore, we observe that the disturbances of velocity at the right boundary of the wave tank grow steadily as time increases and become nonnegligible in appearance after a certain amount of time. Also, the slight disturbances in the middle of the tank grow larger as time increases, and by the last timestep their magnitude exceeds that of the waves.

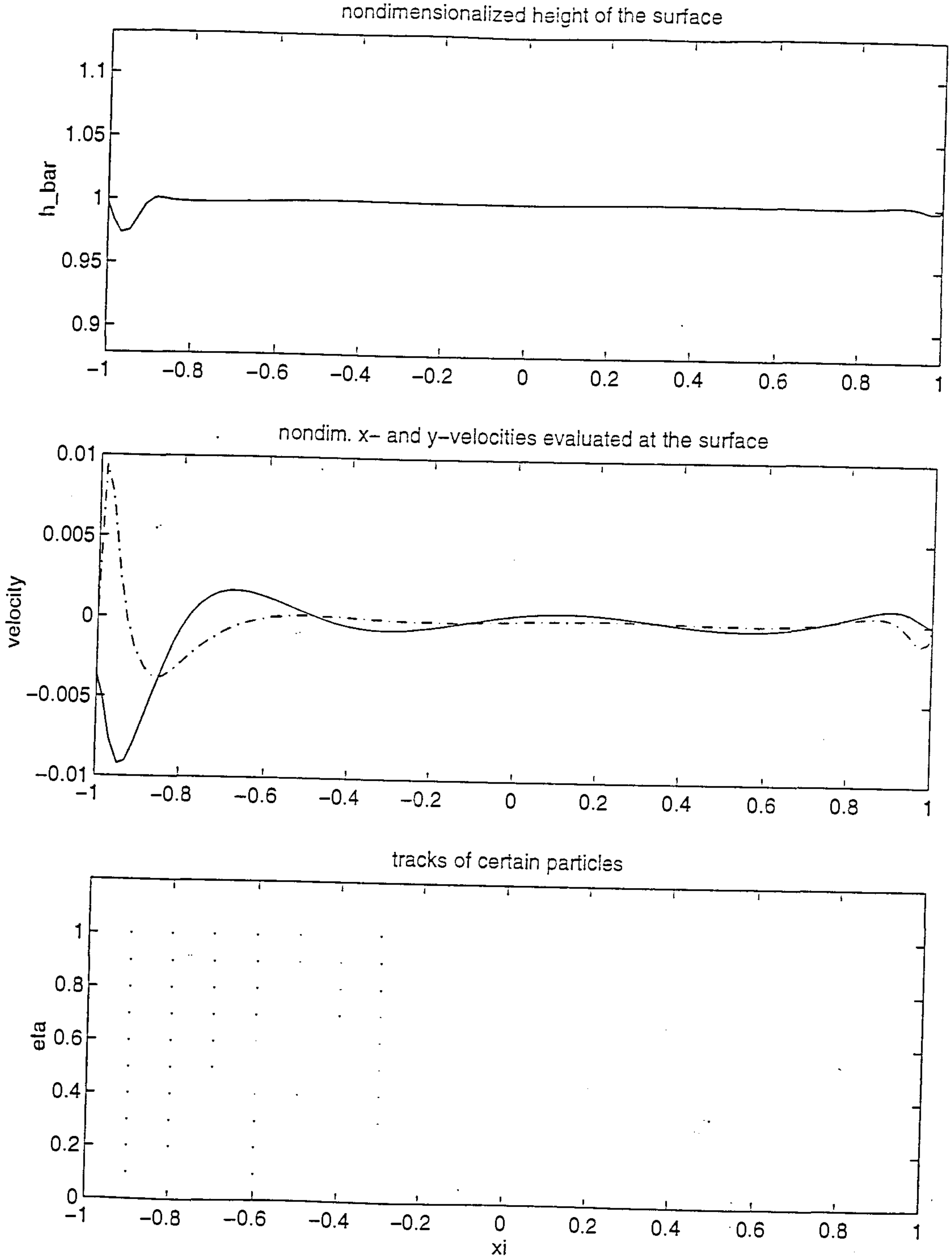


figure 6.1: Example 1, results after 50 timesteps

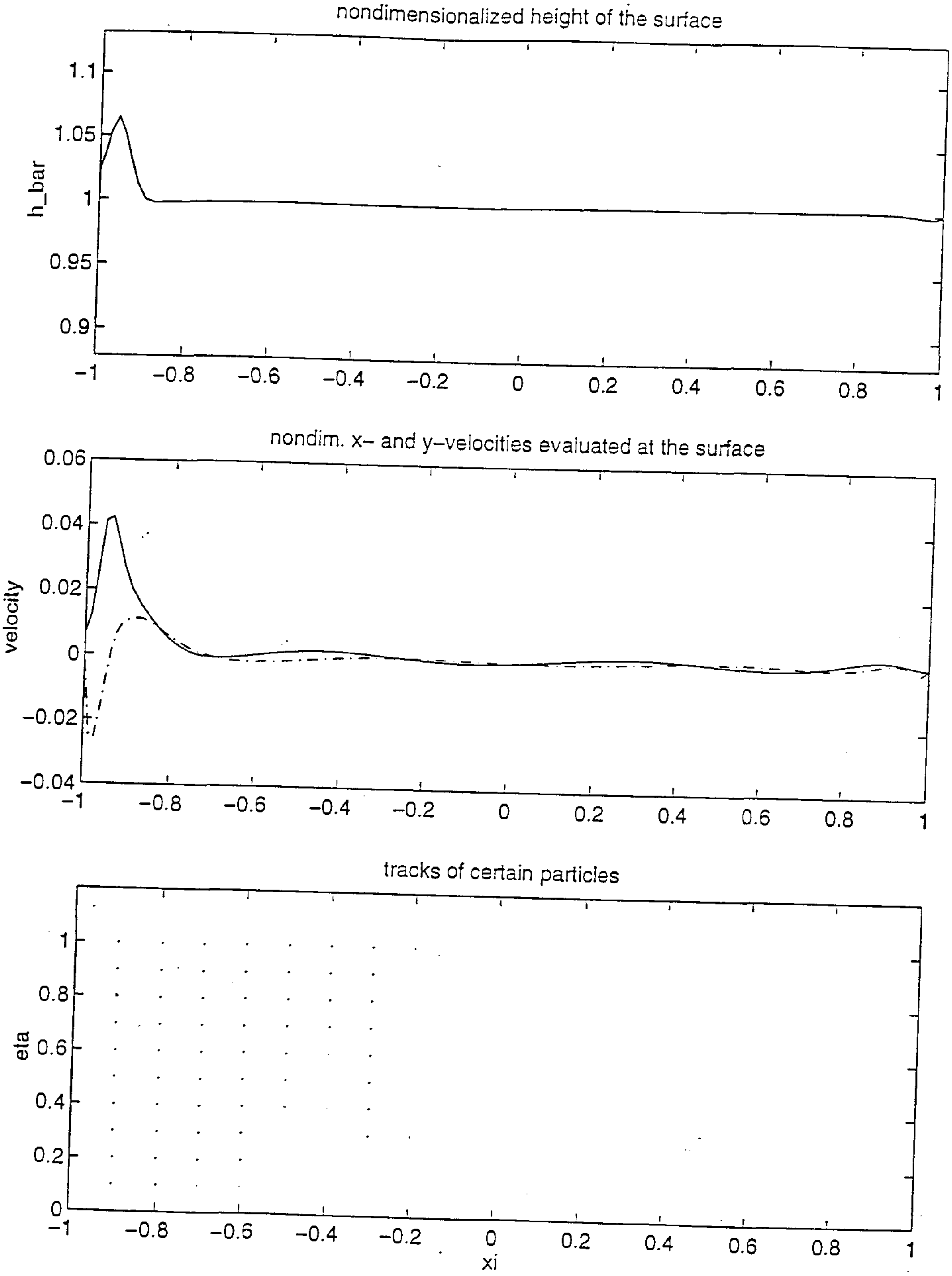
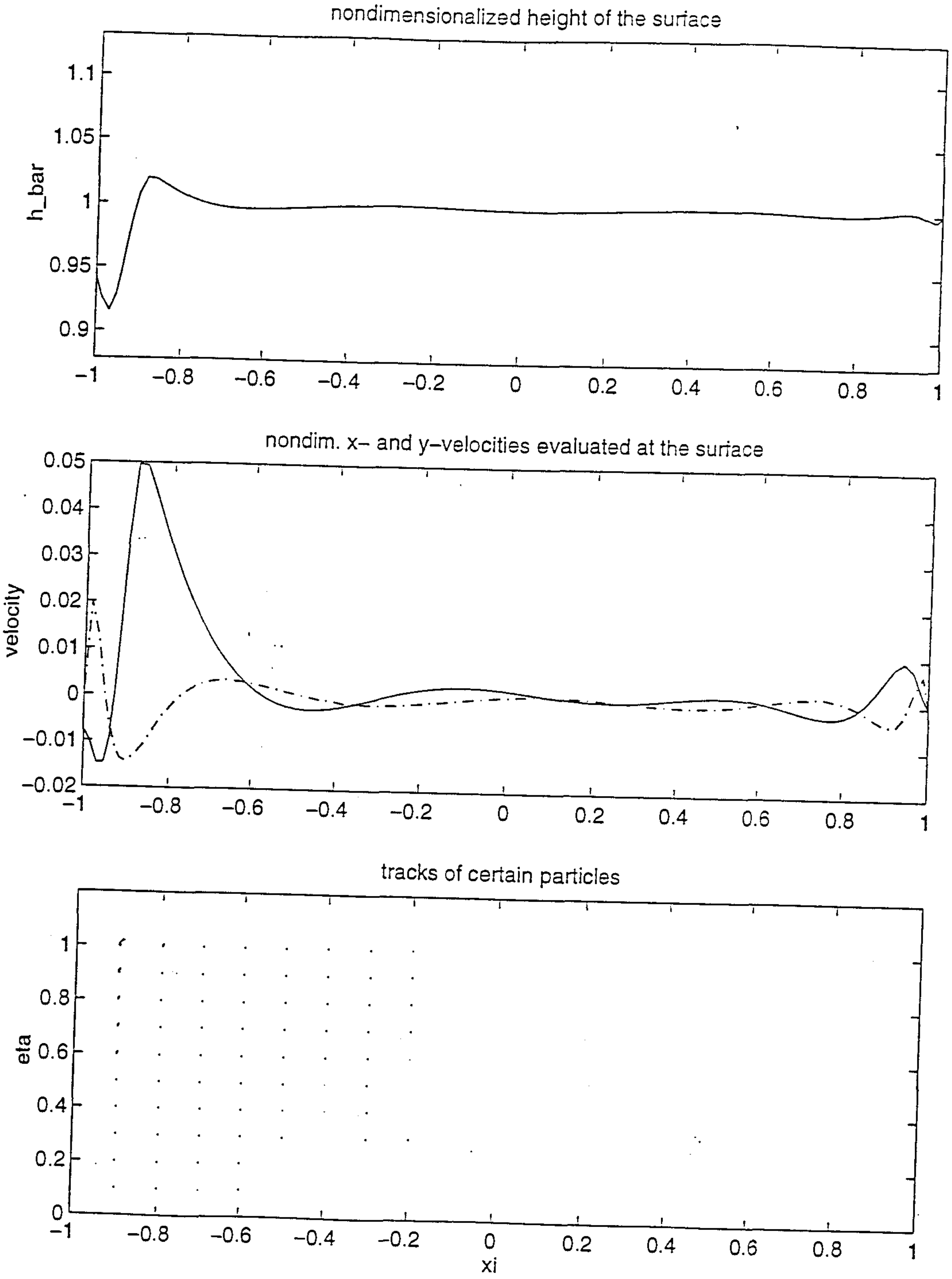


figure 6.1 cont.: Example 1, results after 100 timesteps



nondimensionalized height of the surface

nondim. x– and y–velocities evaluated at the surface

tracks of certain particles

figure 6.1 cont.: Example 1, results after 150 timesteps

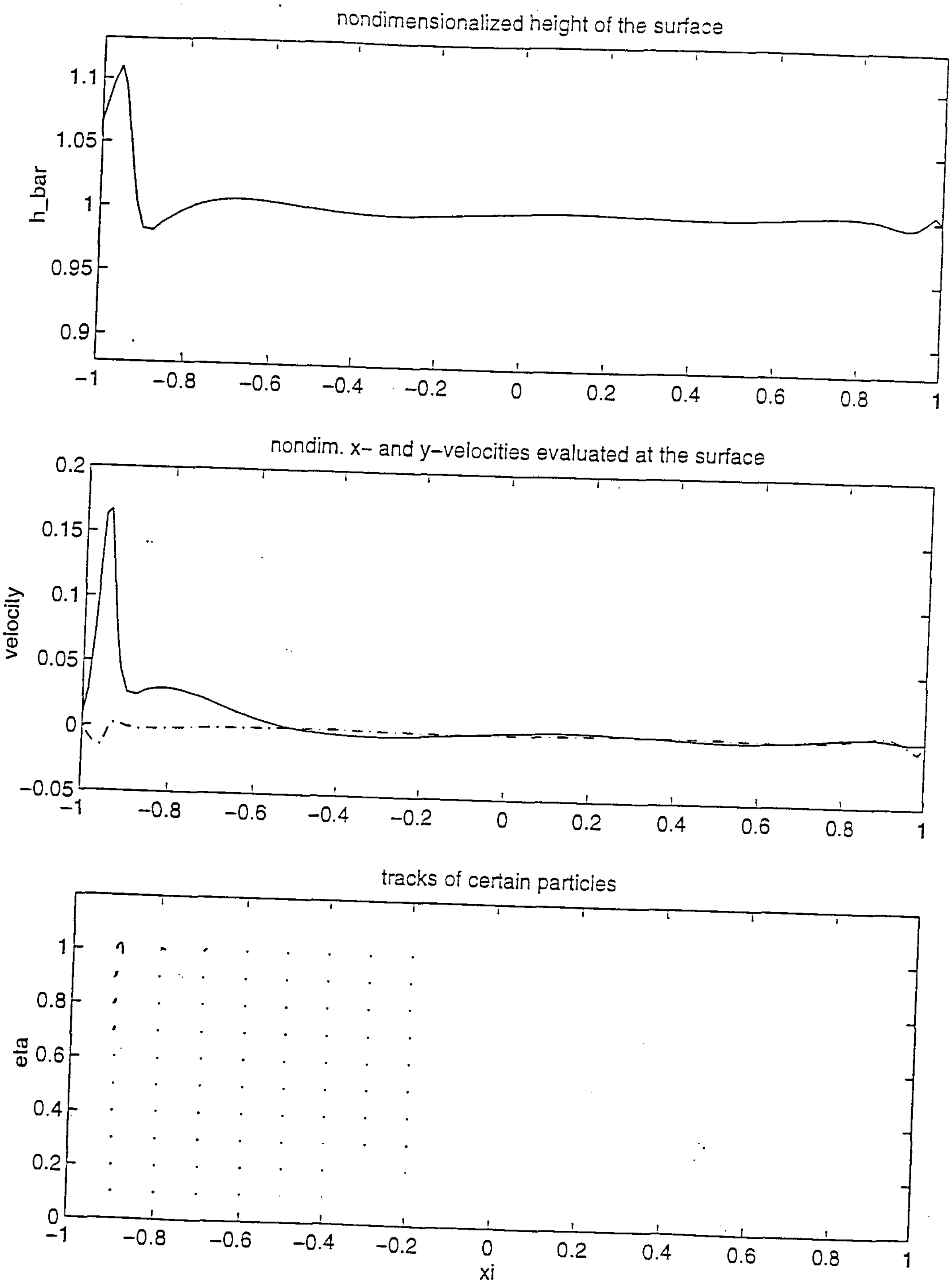
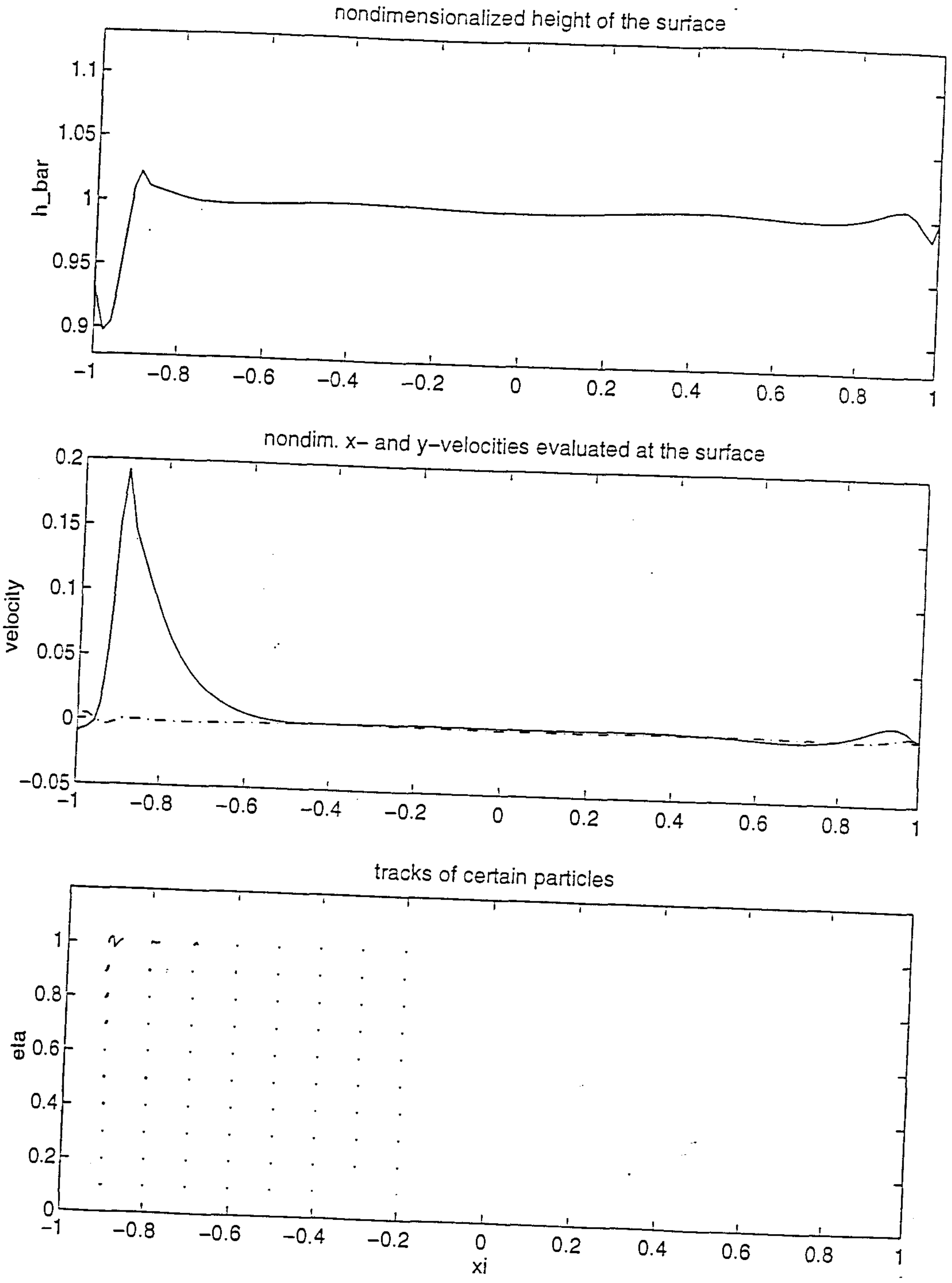


figure 6.1 cont.: Example 1, results after 200 timesteps

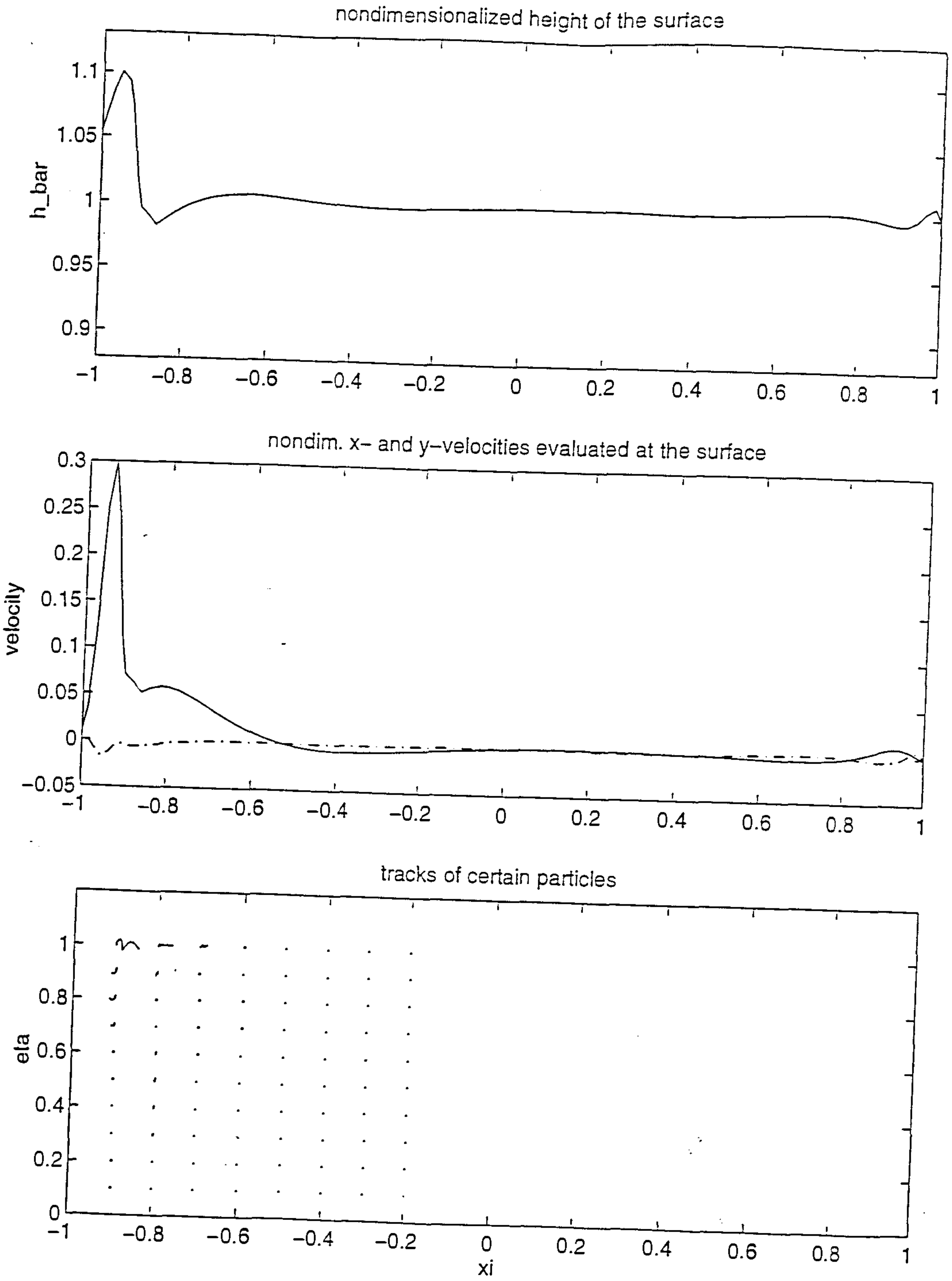


nondimensionalized height of the surface

nondim. x- and y-velocities evaluated at the surface

tracks of certain particles

figure 6.1 cont.: Example 1, results after 250 timesteps



nondimensionalized height of the surface

nondim. x- and y-velocities evaluated at the surface

tracks of certain particles

figure 6.1 cont.: Example 1, results after 300 timesteps

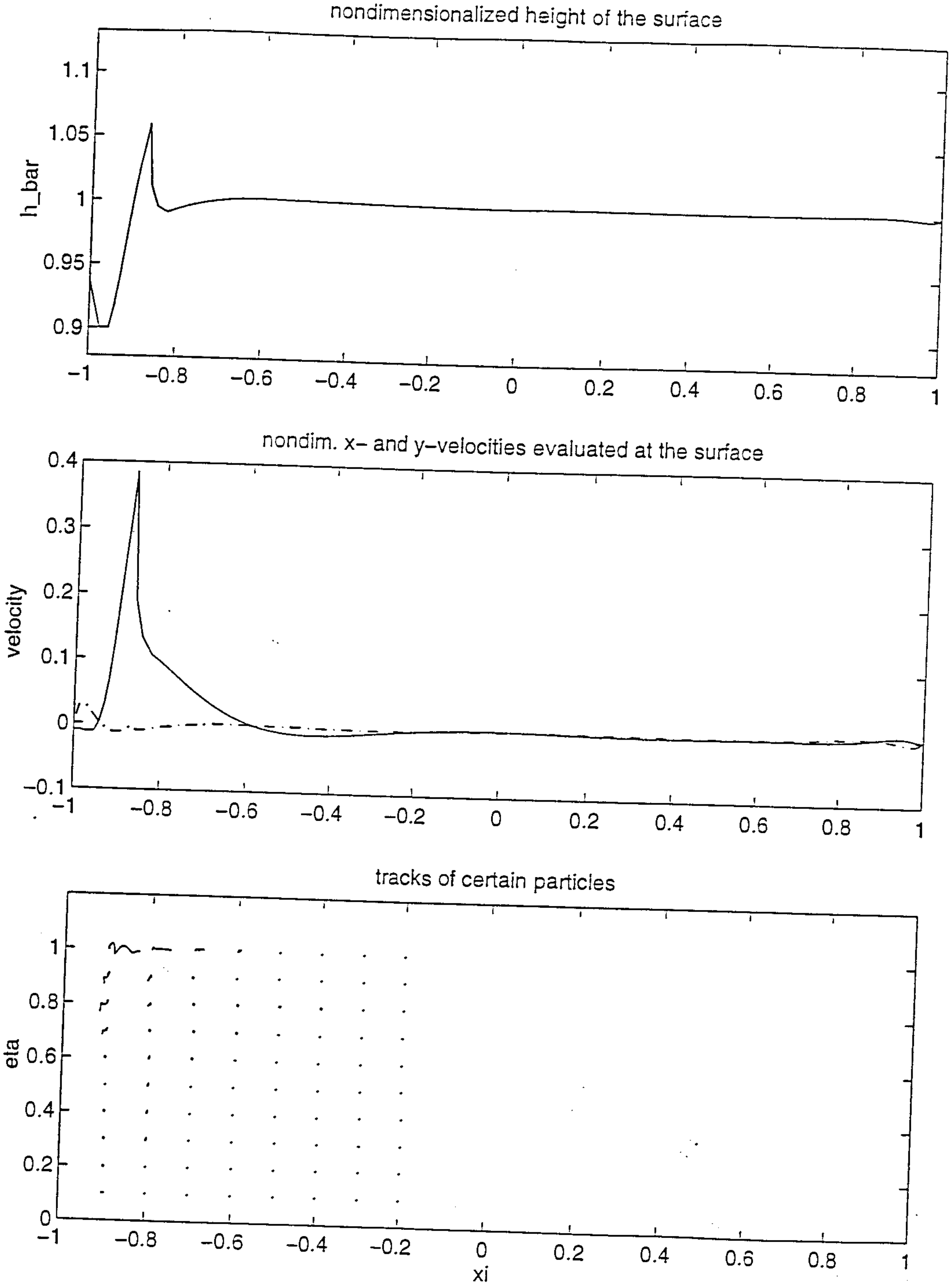


figure 6.1 cont.: Example 1, results after 350 timesteps

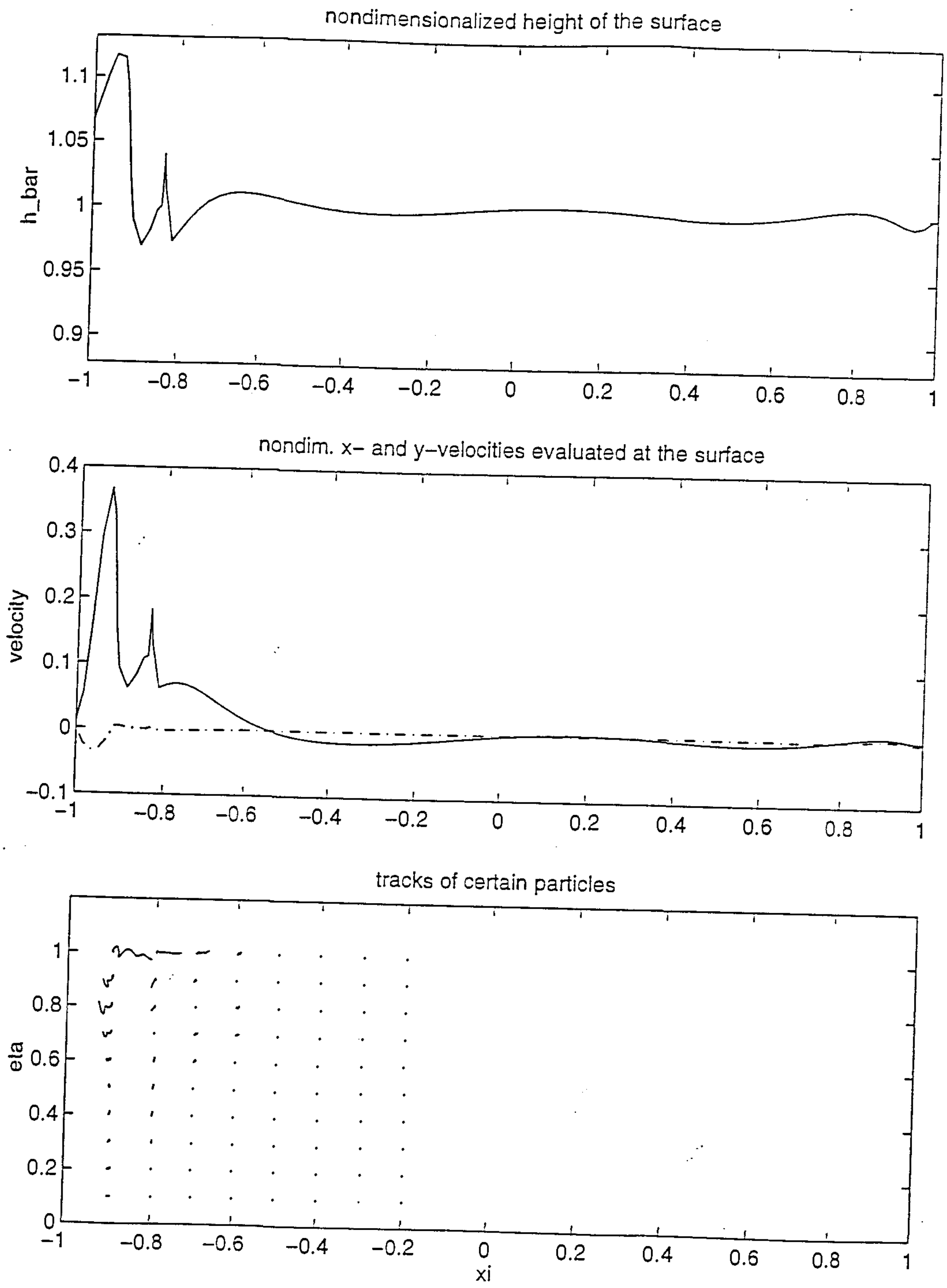
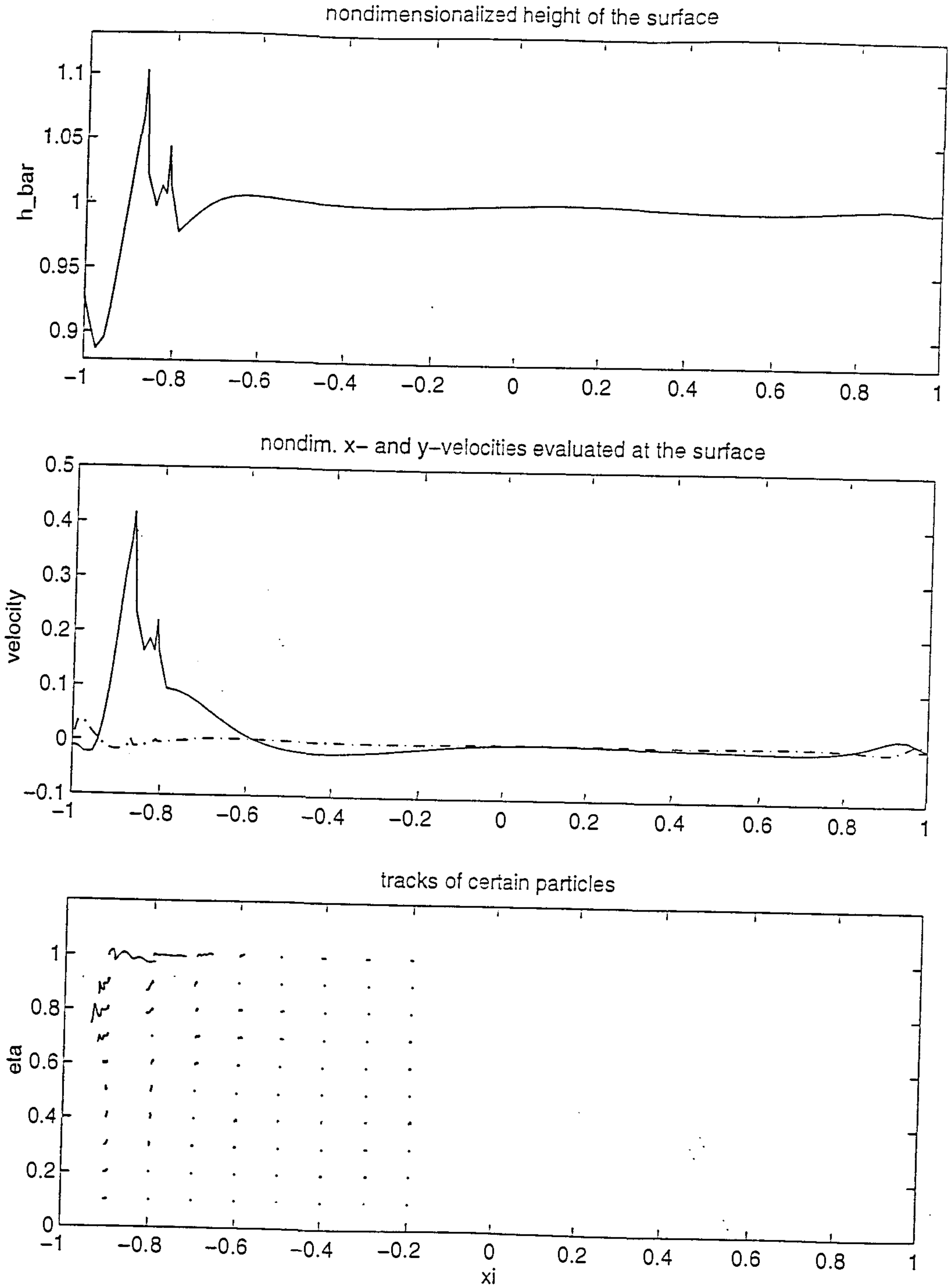


figure 6.1 cont.: Example 1, results after 400 timesteps


nondimensionalized height of the surface

nondim. x– and y–velocities evaluated at the surface

tracks of certain particles

figure 6.1 cont.: Example 1, results after 450 timesteps

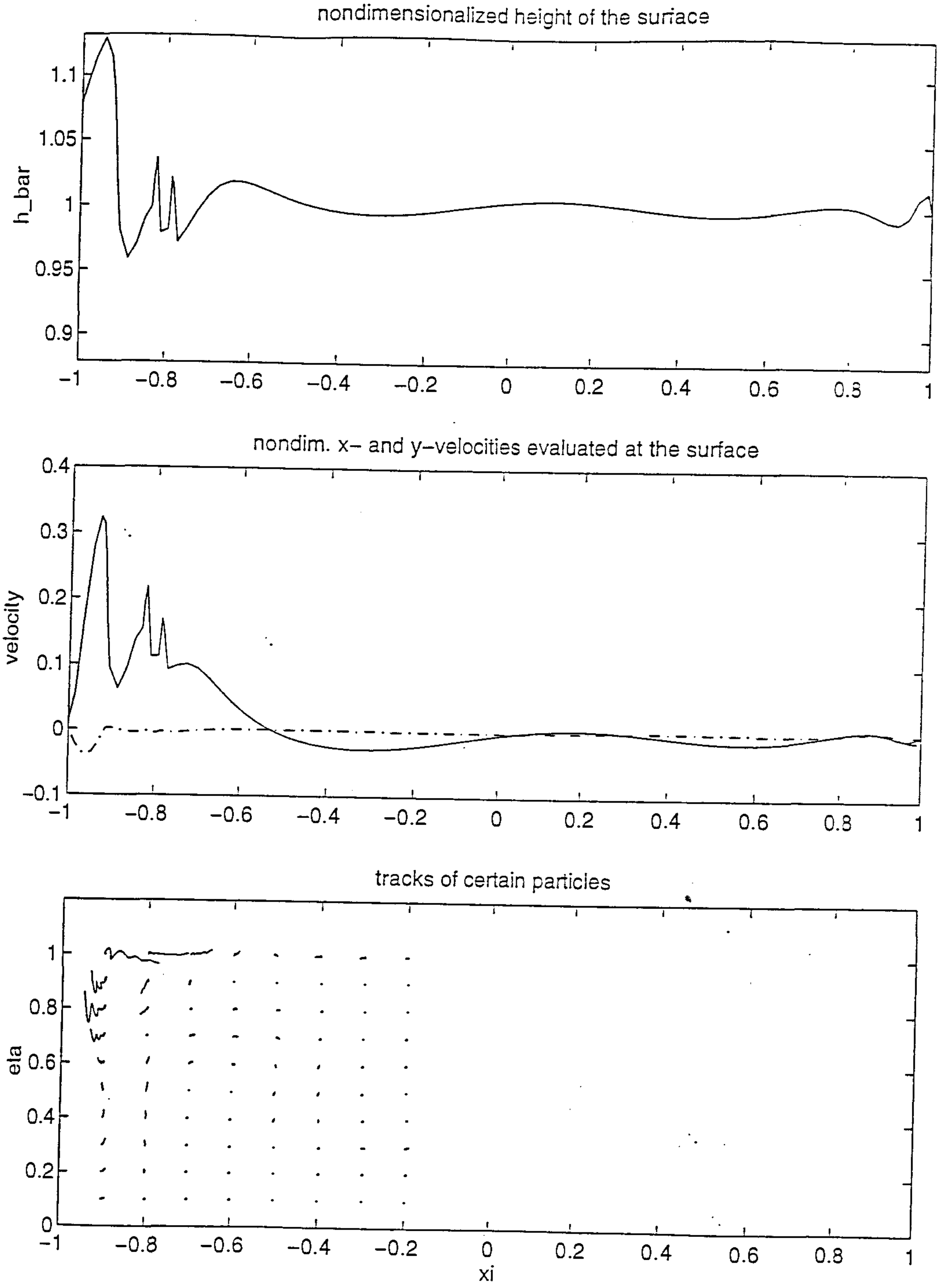
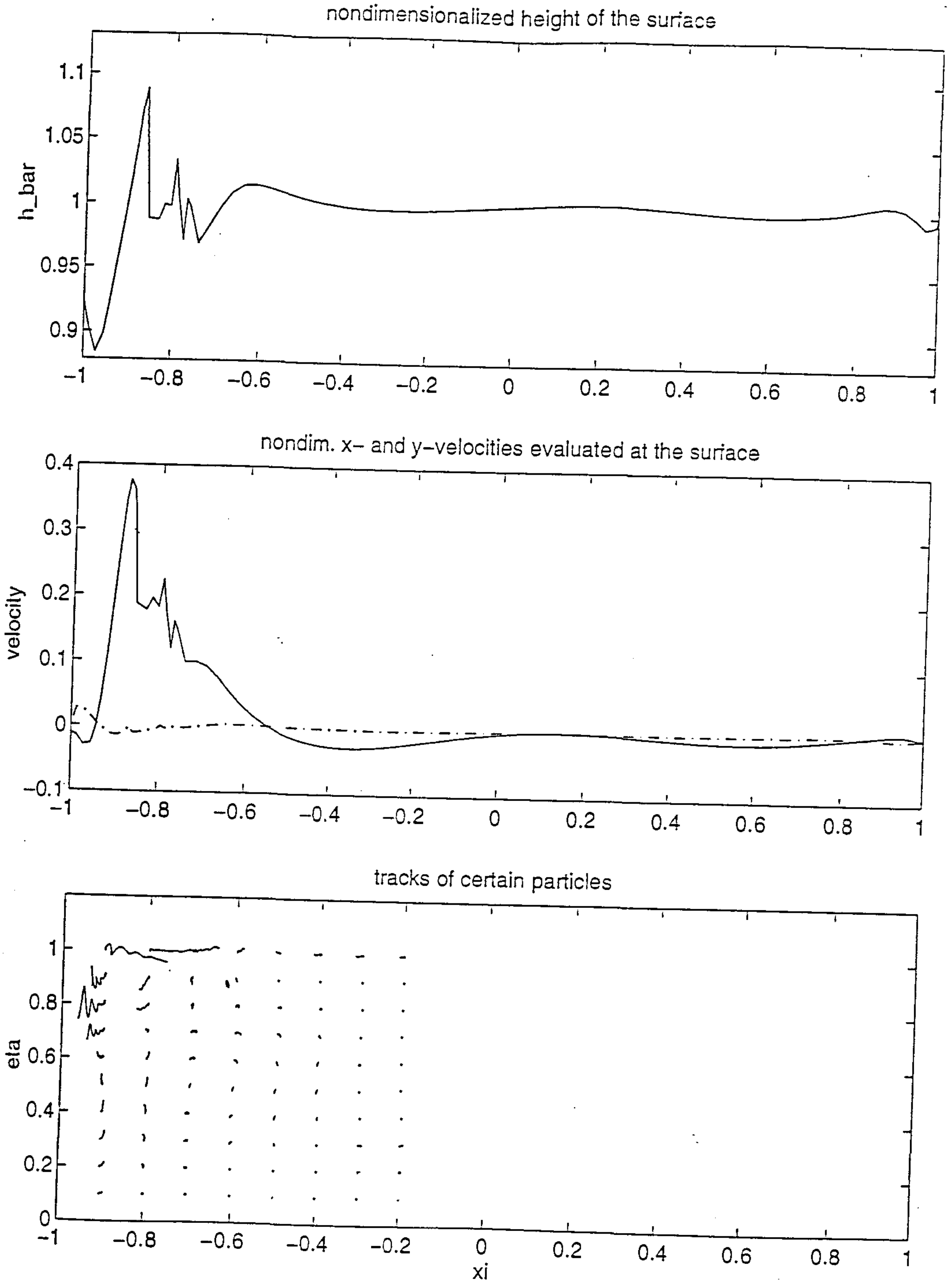


figure 6.1 cont.: Example 1, results after 500 timesteps


nondimensionalized height of the surface

nondim. x- and y-velocities evaluated at the surface

tracks of certain particles

figure 6.1 cont.: Example 1, results after 550 timesteps

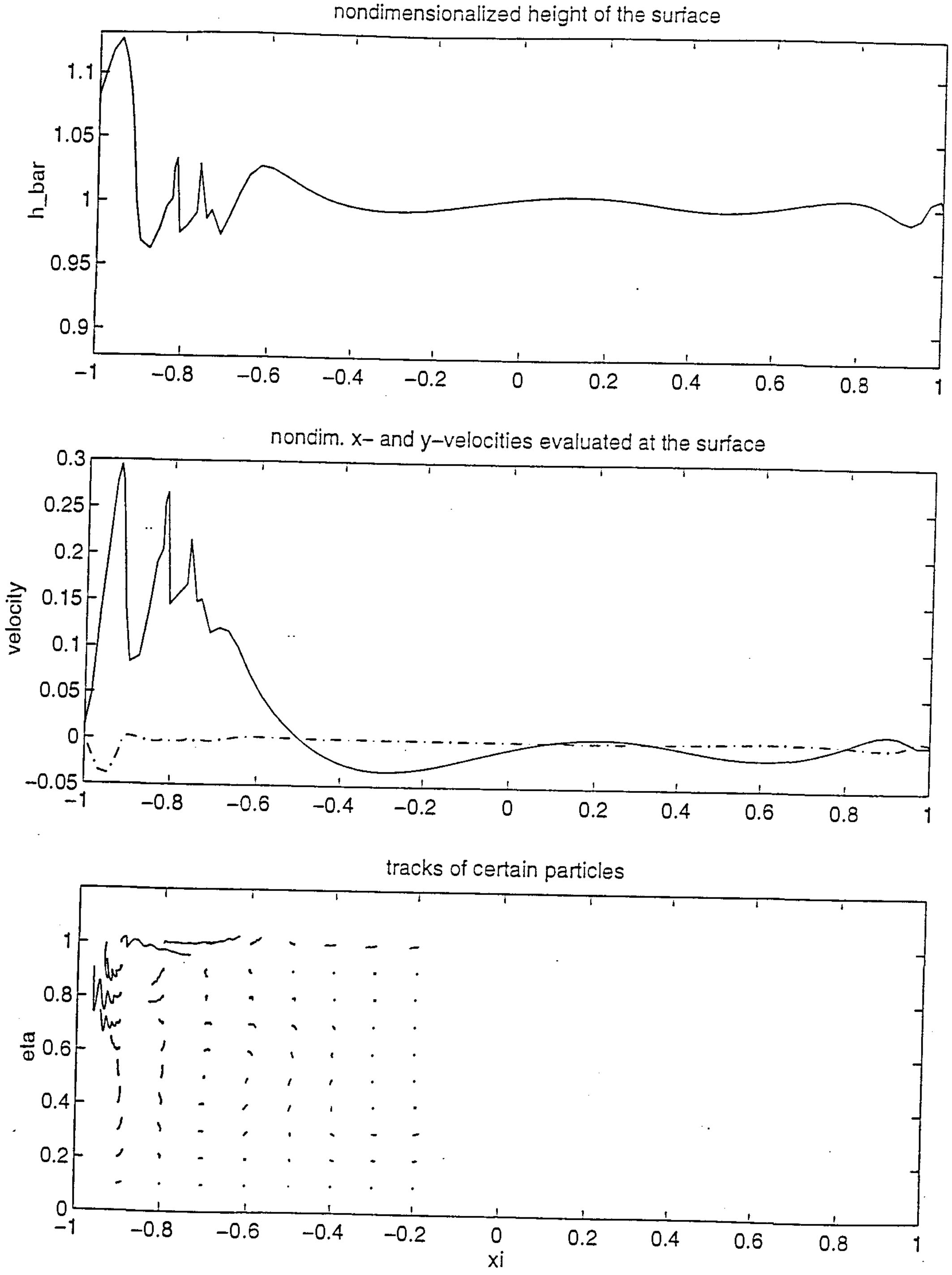


figure 6.1 cont.: Example 1, results after 600 timesteps

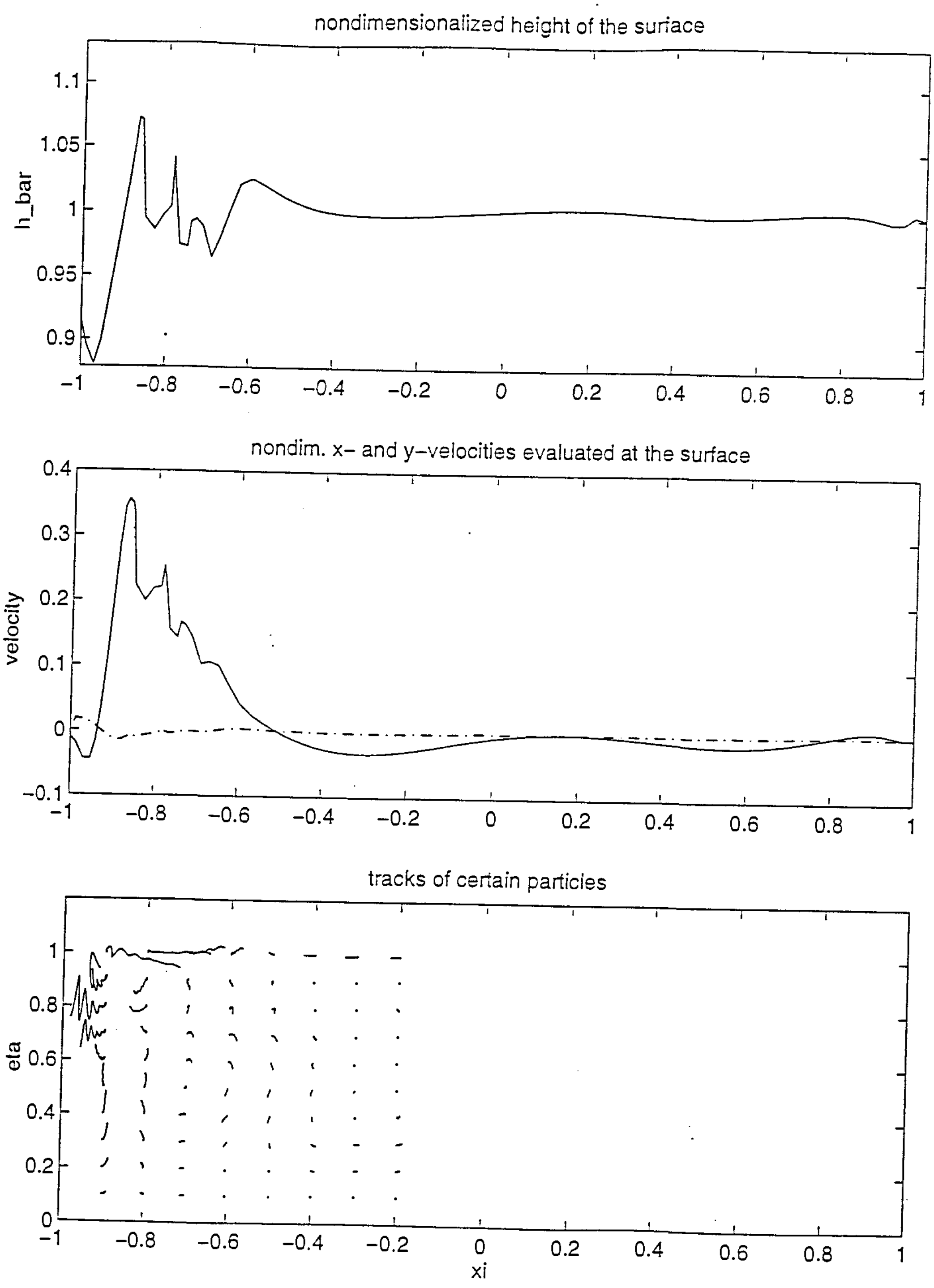


figure 6.1 cont.: Example 1, results after 650 timesteps

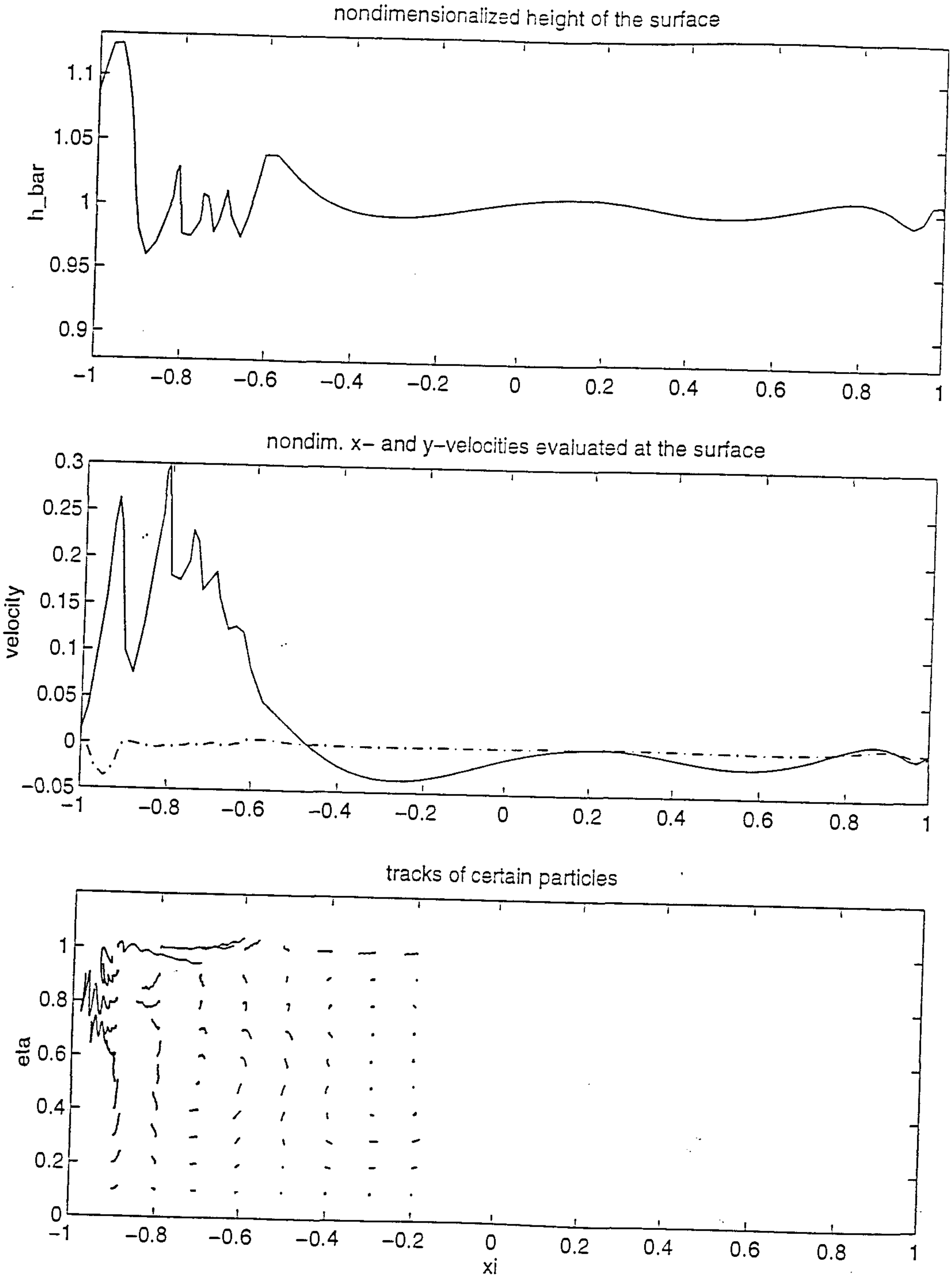


figure 6.1 cont.: Example 1, results after 700 timesteps

6.2 Example 2

Example 2 was produced under the same assumptions as experiment 1. The only difference is that we used the moving grid surface model instead of the fixed grid model.

Remarks: We see clearly that computing "overhanging" waves is possible with this kind of surface model. We see also that, from a certain point of time on, the breaking waves do not look realistic. This is due to the weakness of the polynomial model and its insufficient flexibility. However, no computations using the moving surface model combined with the finite elements were performed. Thus, we cannot compare, and it is impossible to judge, how powerful the moving surface model really is.

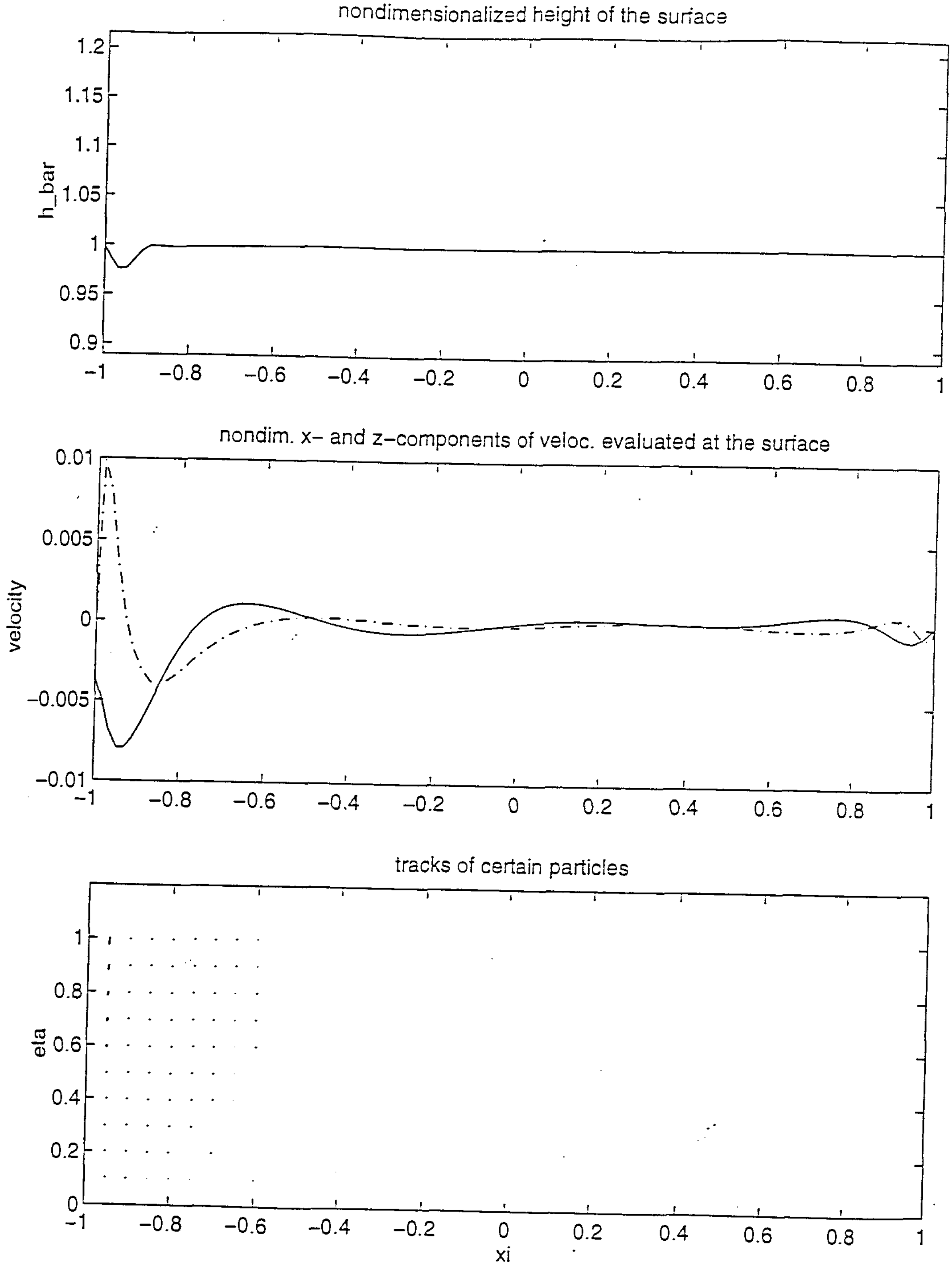


figure 6.2: Example 2, results after 50 timesteps

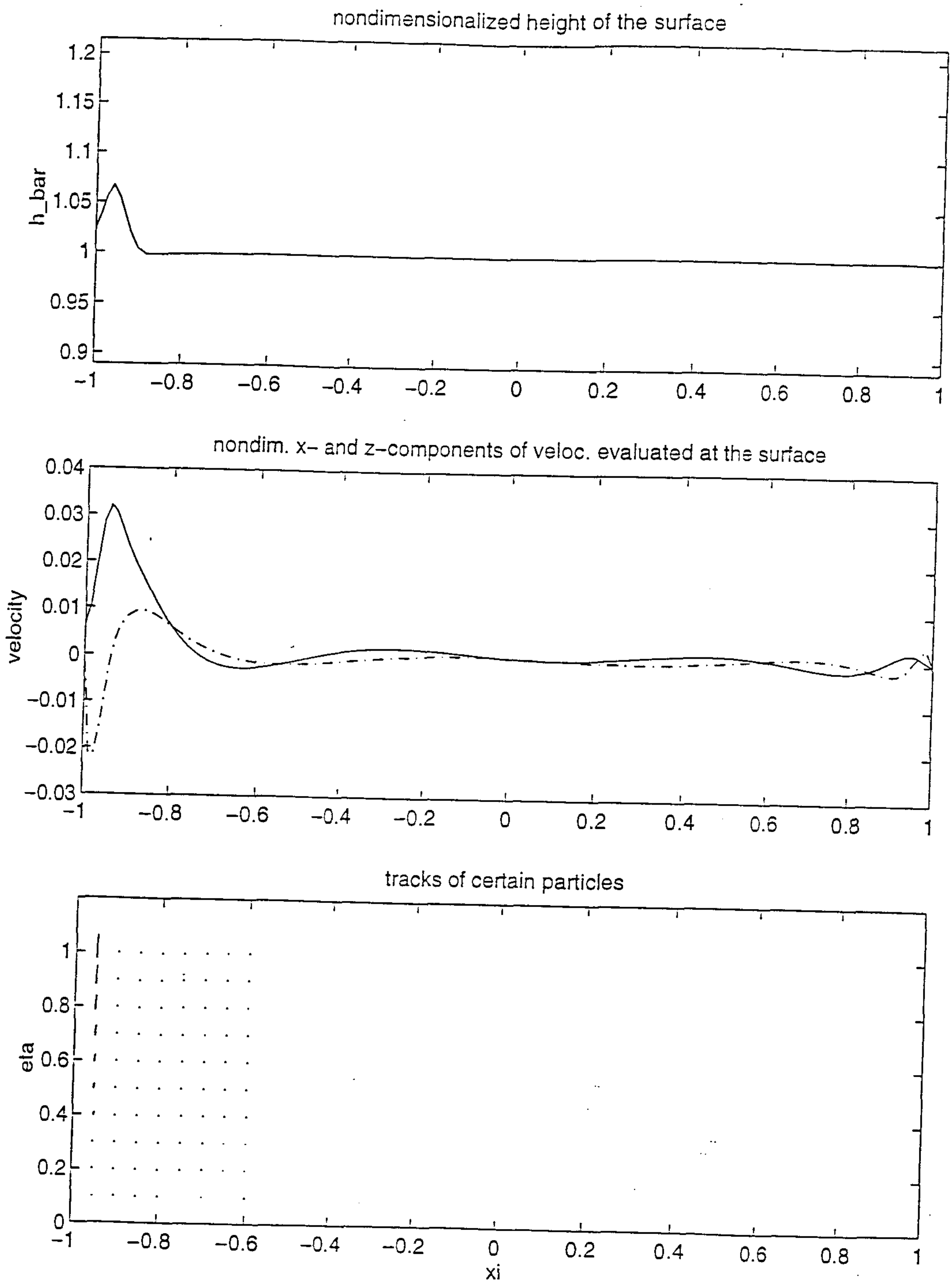


figure 6.2 cont.: Example 2, results after 100 timesteps

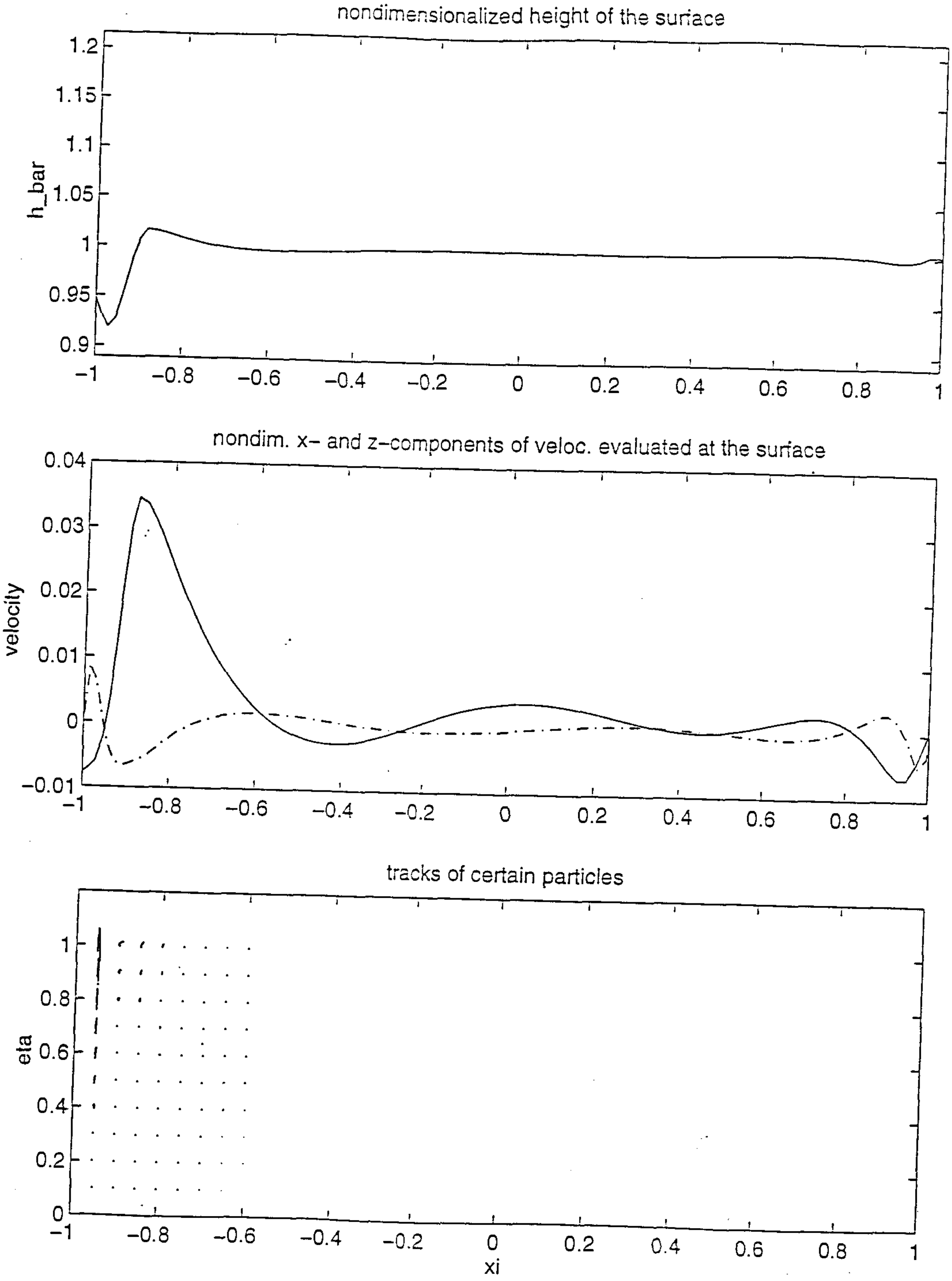
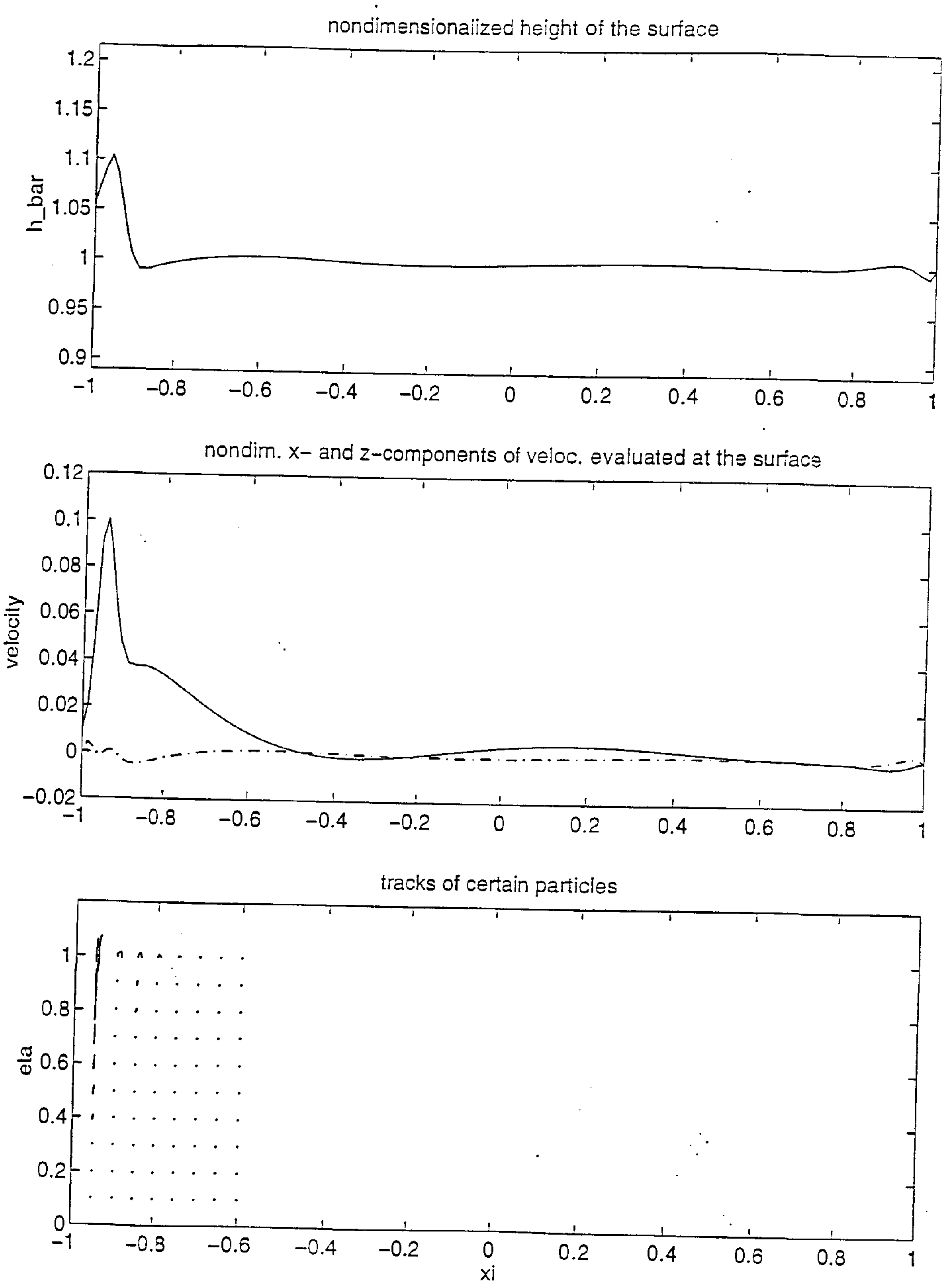


figure 6.2 cont.: Example 2, results after 150 timesteps



nondimensionalized height of the surface

nondim. x– and z–components of veloc. evaluated at the surface

tracks of certain particles

figure 6.2 cont.: Example 2, results after 200 timesteps

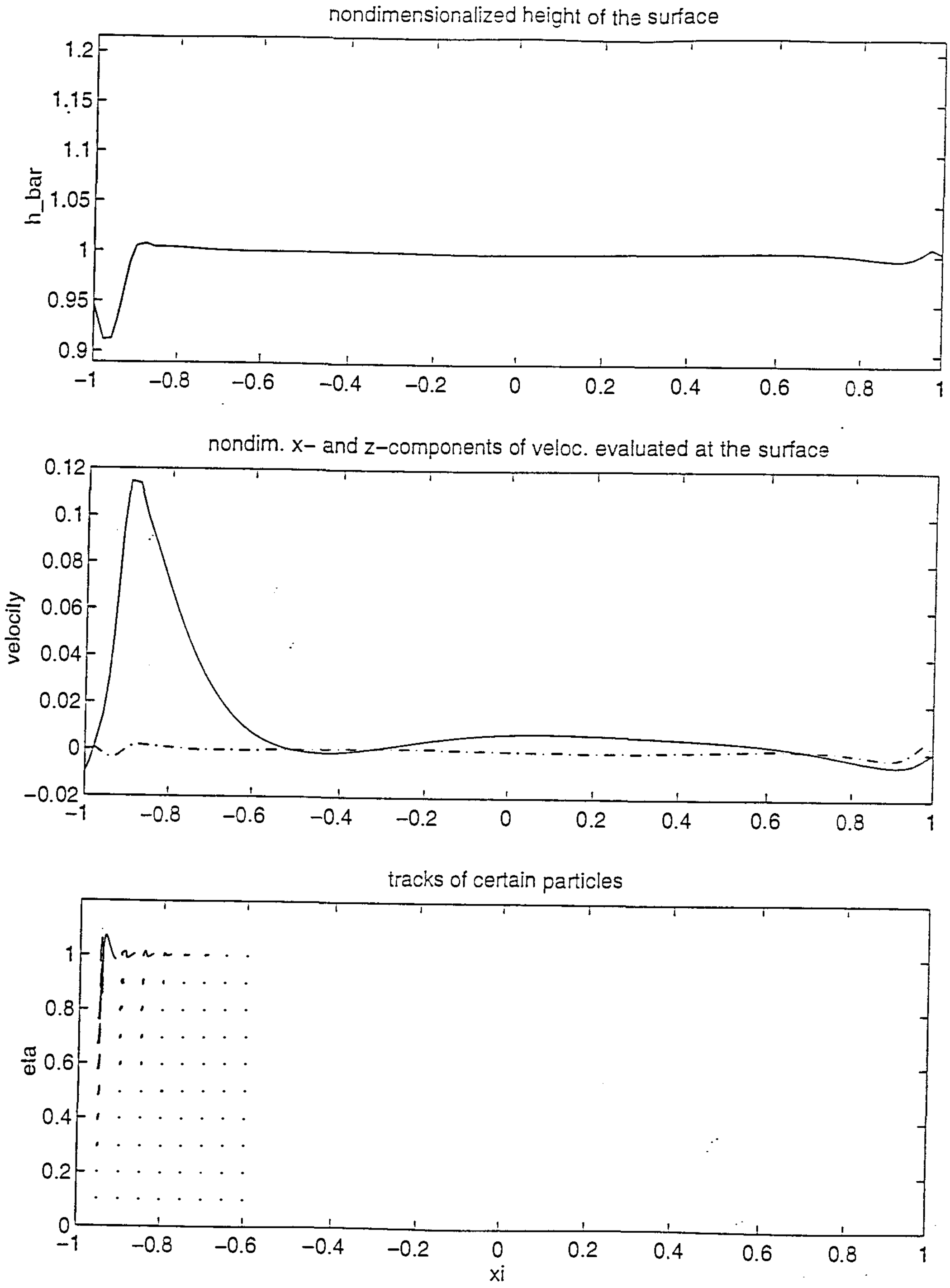


figure 6.2 cont.: Example 2, results after 250 timesteps

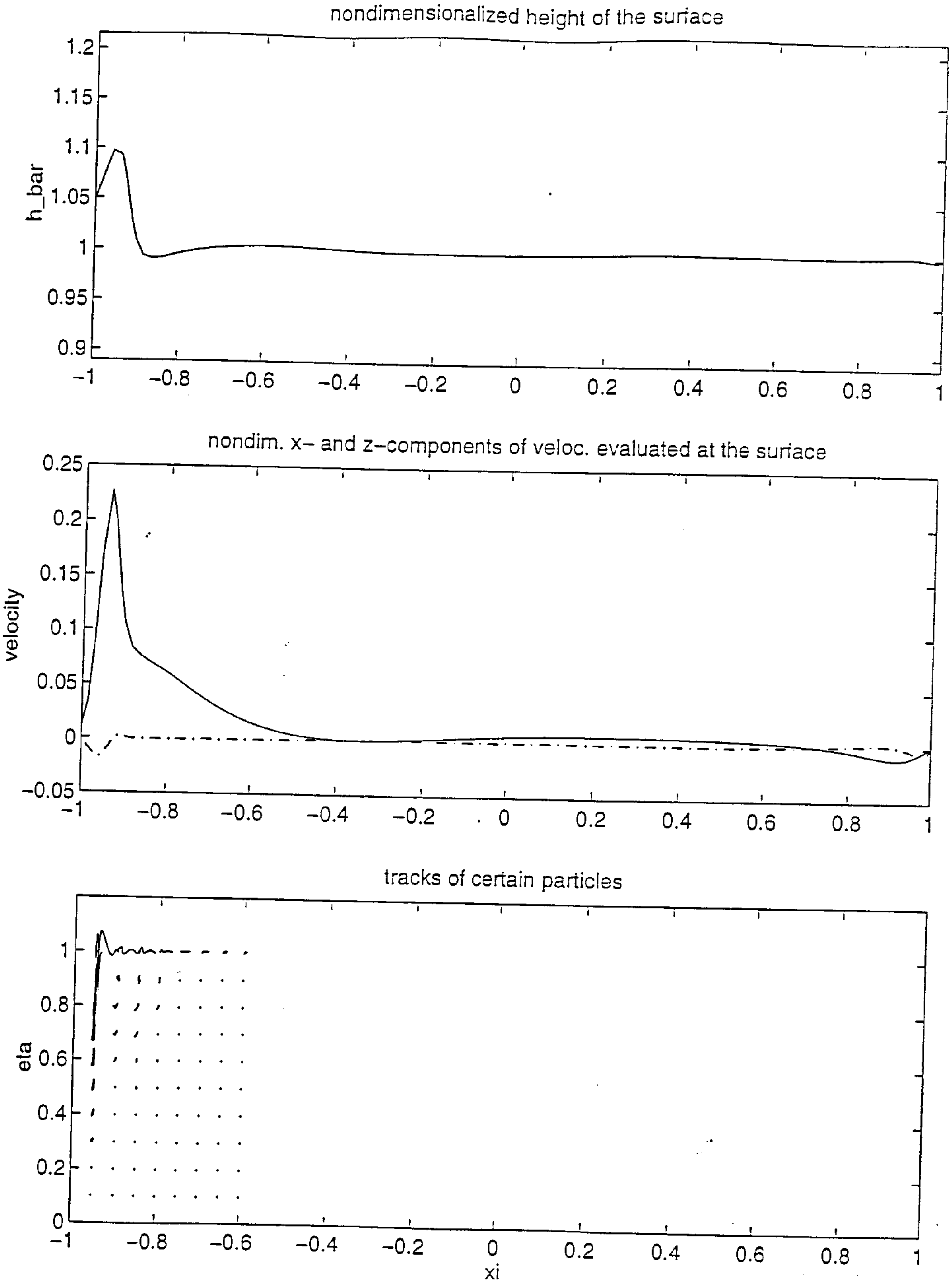


figure 6.2 cont.: Example 2, results after 300 timesteps

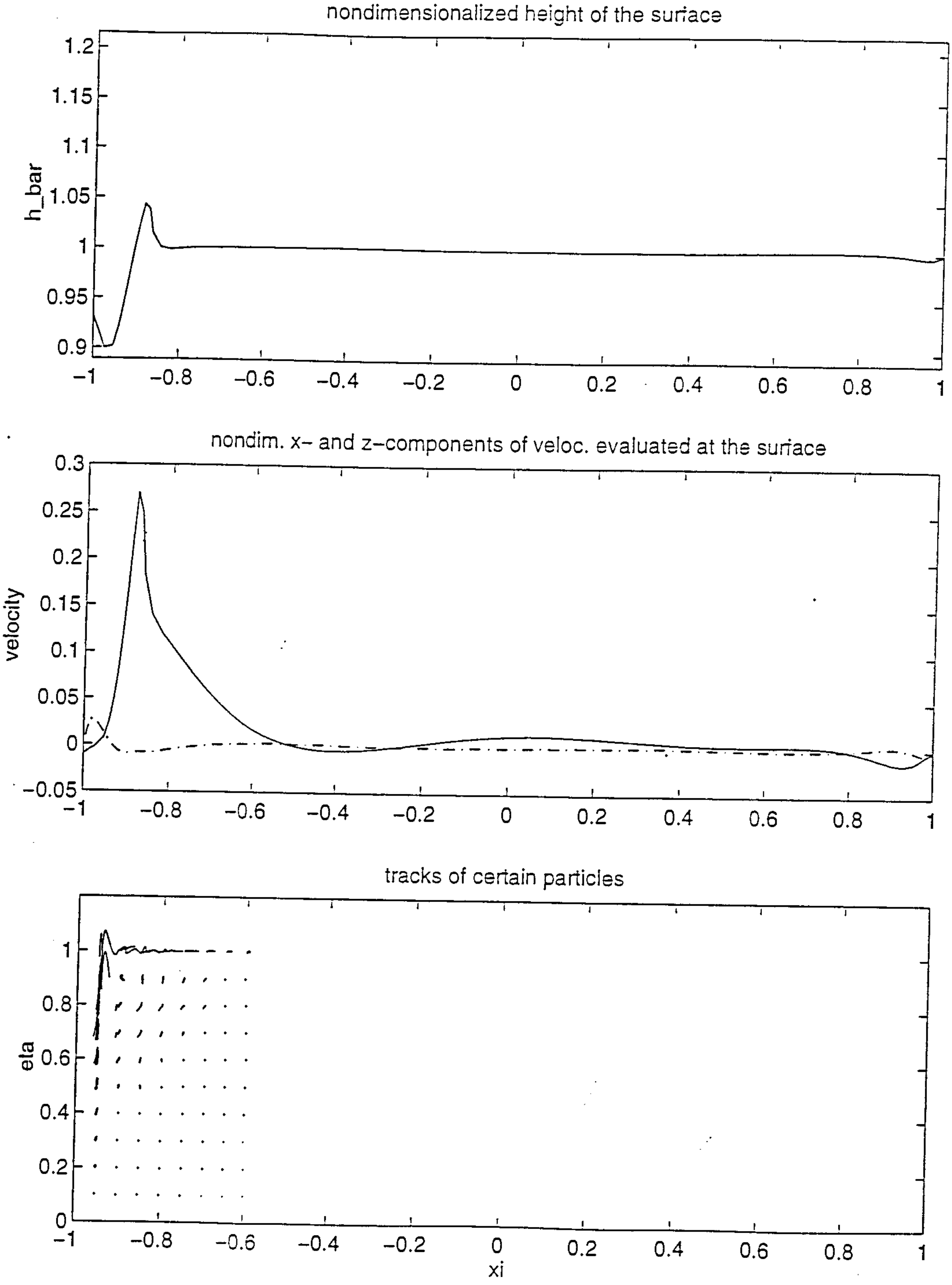
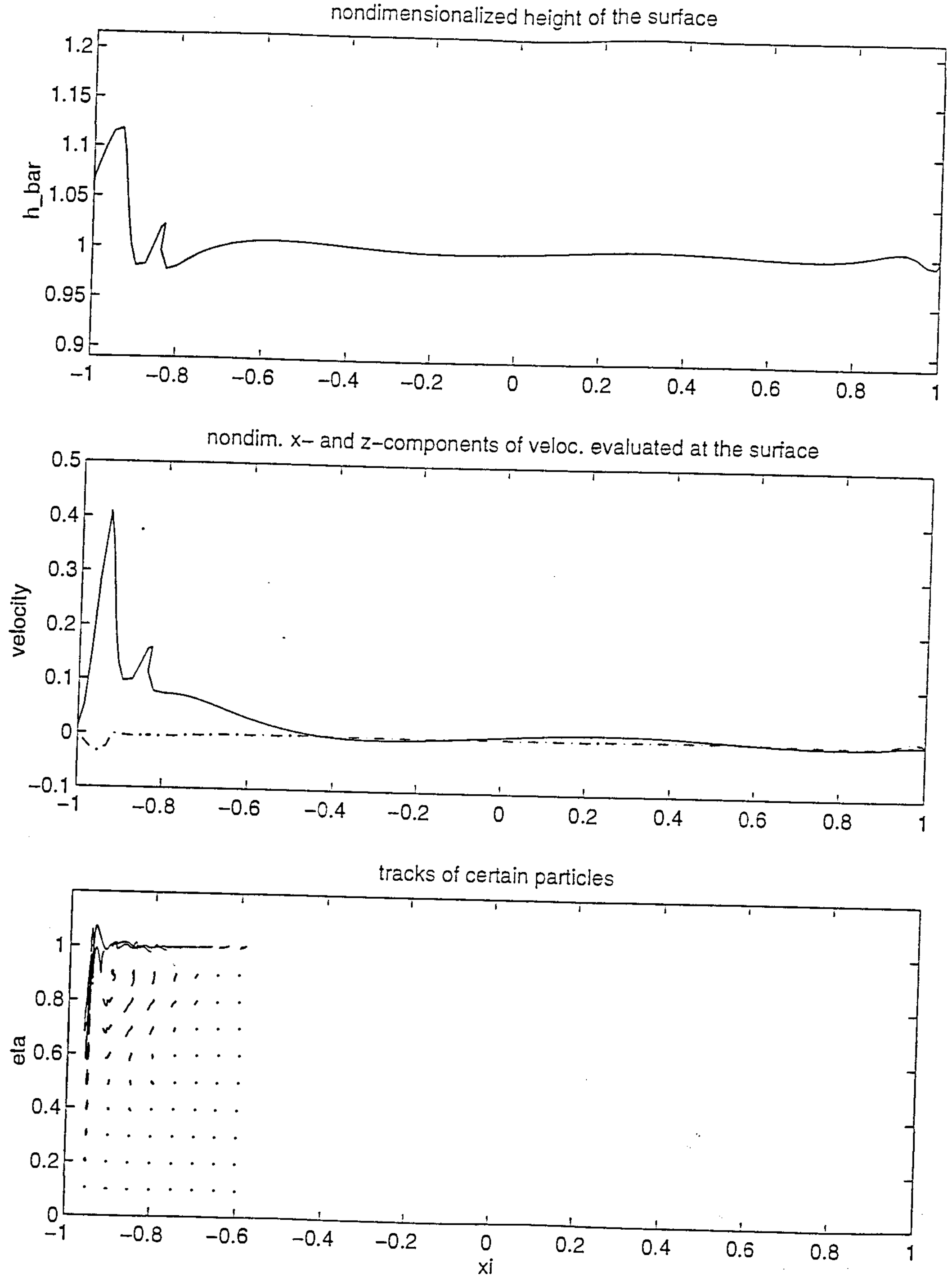


figure 6.2 cont.: Example 2, results after 350 timesteps



nondimensionalized height of the surface

nondim. x– and z–components of veloc. evaluated at the surface

tracks of certain particles

figure 6.2 cont.: Example 2, results after 400 timesteps

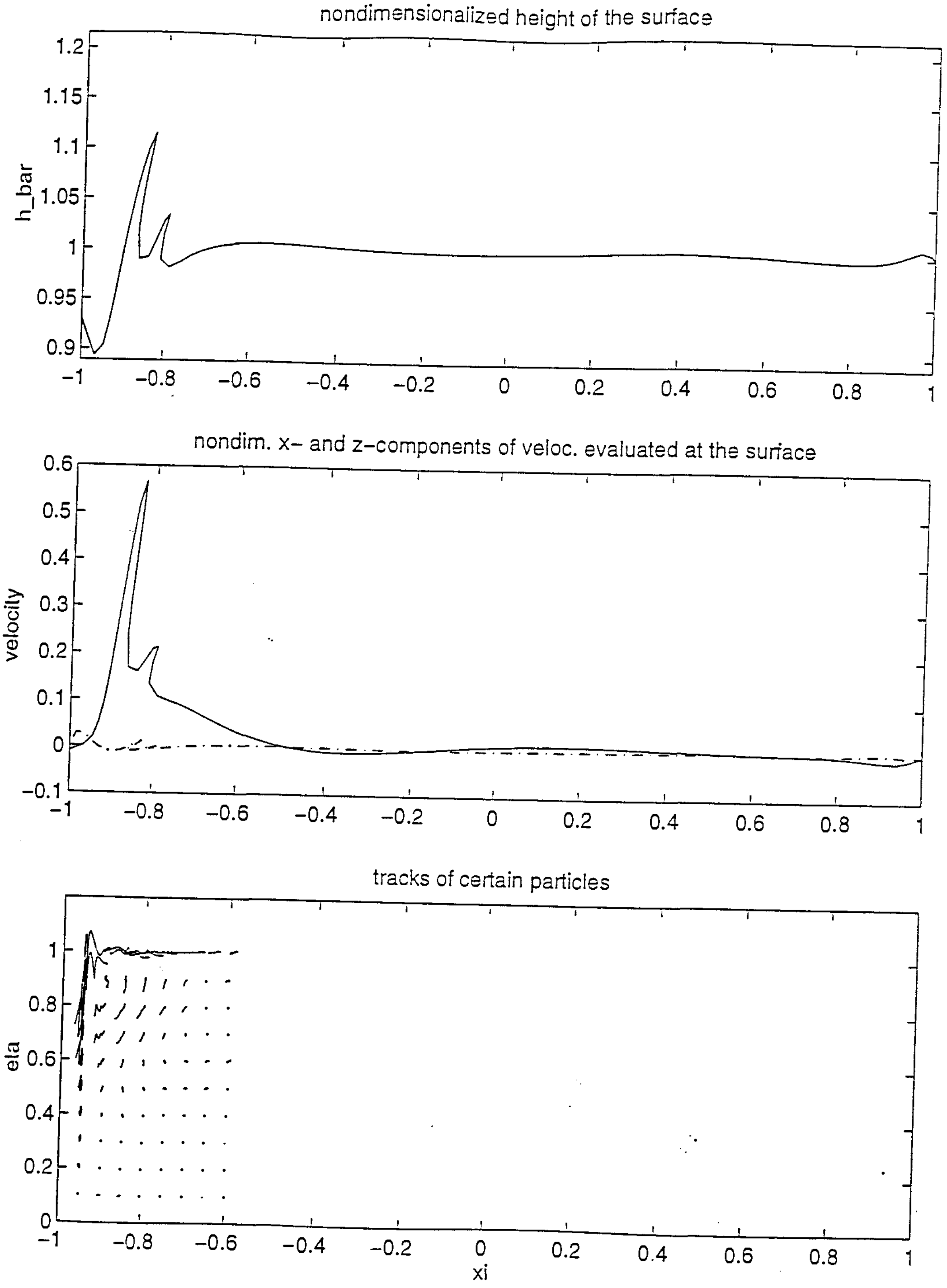


figure 6.2 cont.: Example 2, results after 450 timesteps

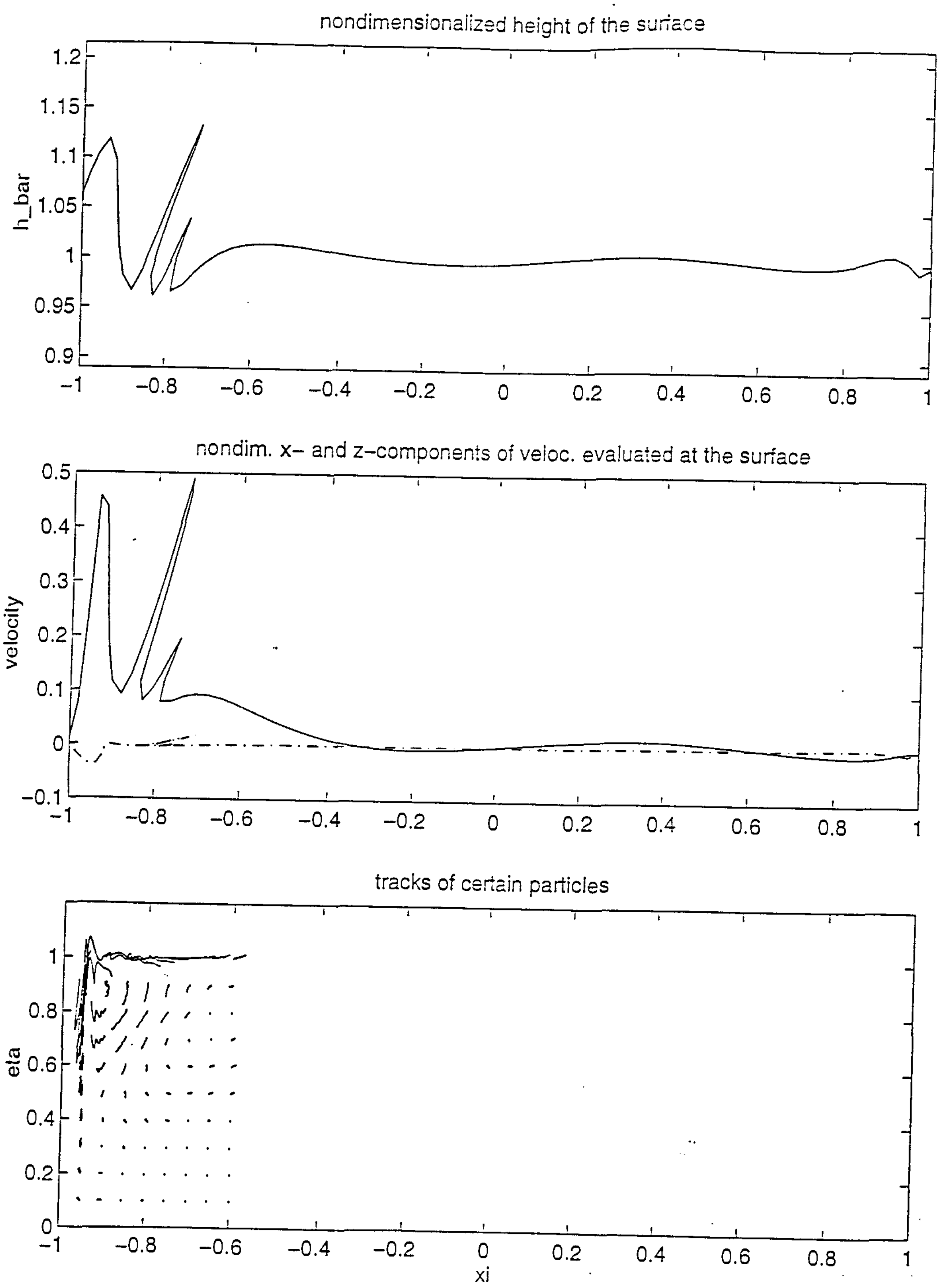


figure 6.2 cont.: Example 2, results after 500 timesteps

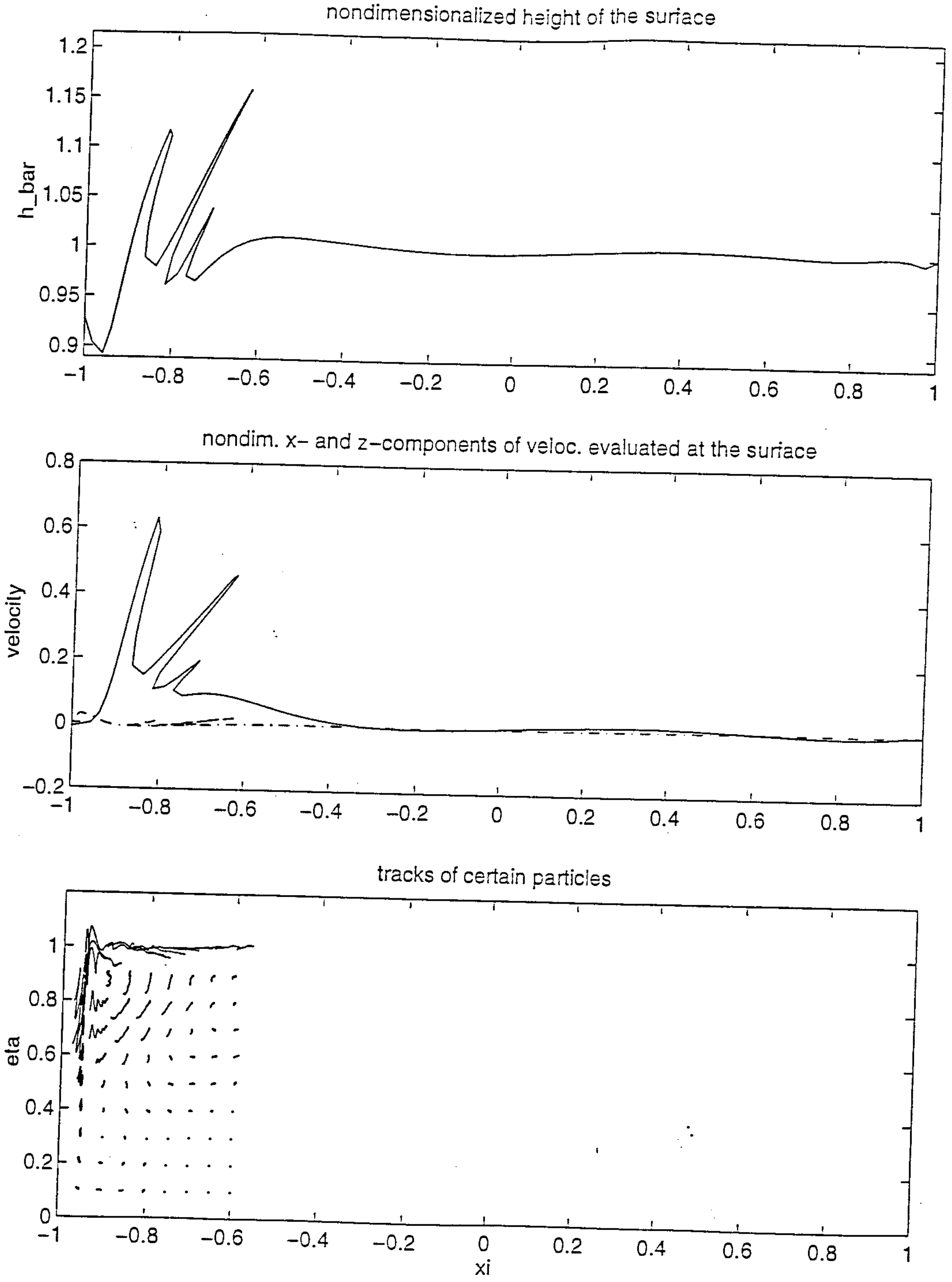


figure 6.2 cont.: Example 2, results after 550 timesteps

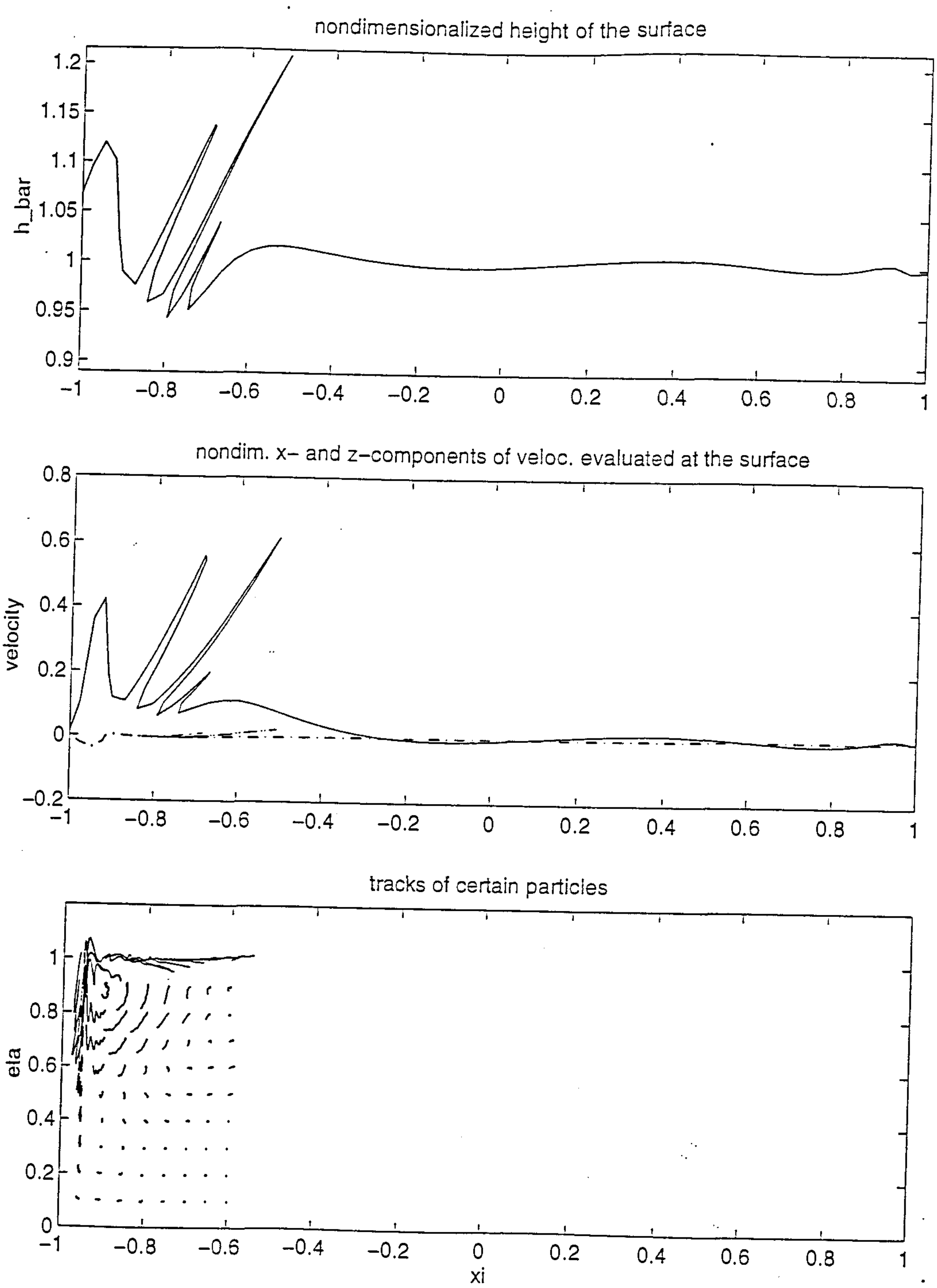


figure 6.2 cont.: Example 2, results after 600 timesteps

6.3 Example 3

Example 3 was produced under the same assumptions as experiment 1. The difference from example 1 is that we eliminated the surface correction mechanism introduced in chapter 4. This has a huge effect on the surface behavior close to the boundary. The differences between example 1 and example 3 can be detected simply by comparing the plots.

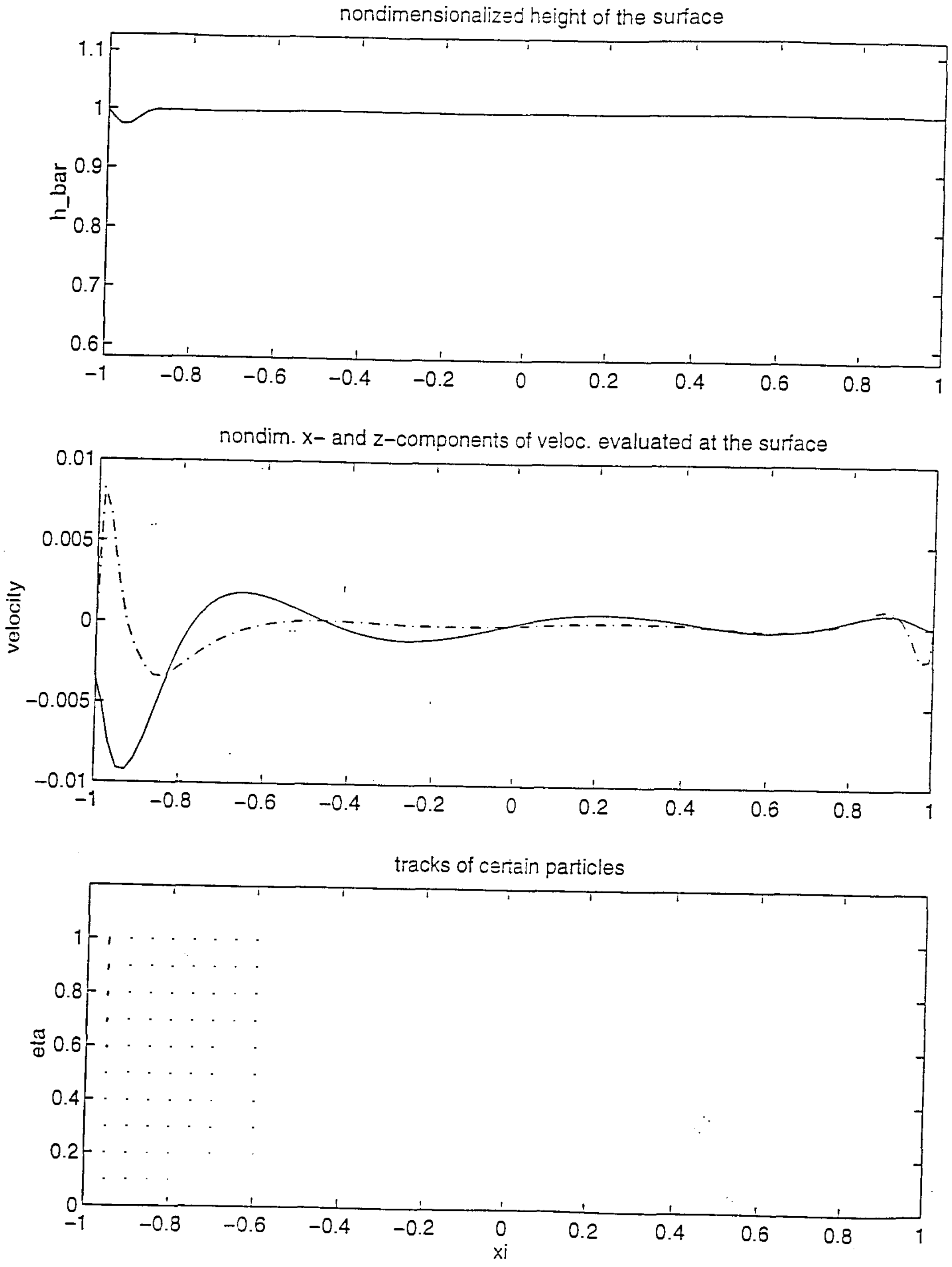
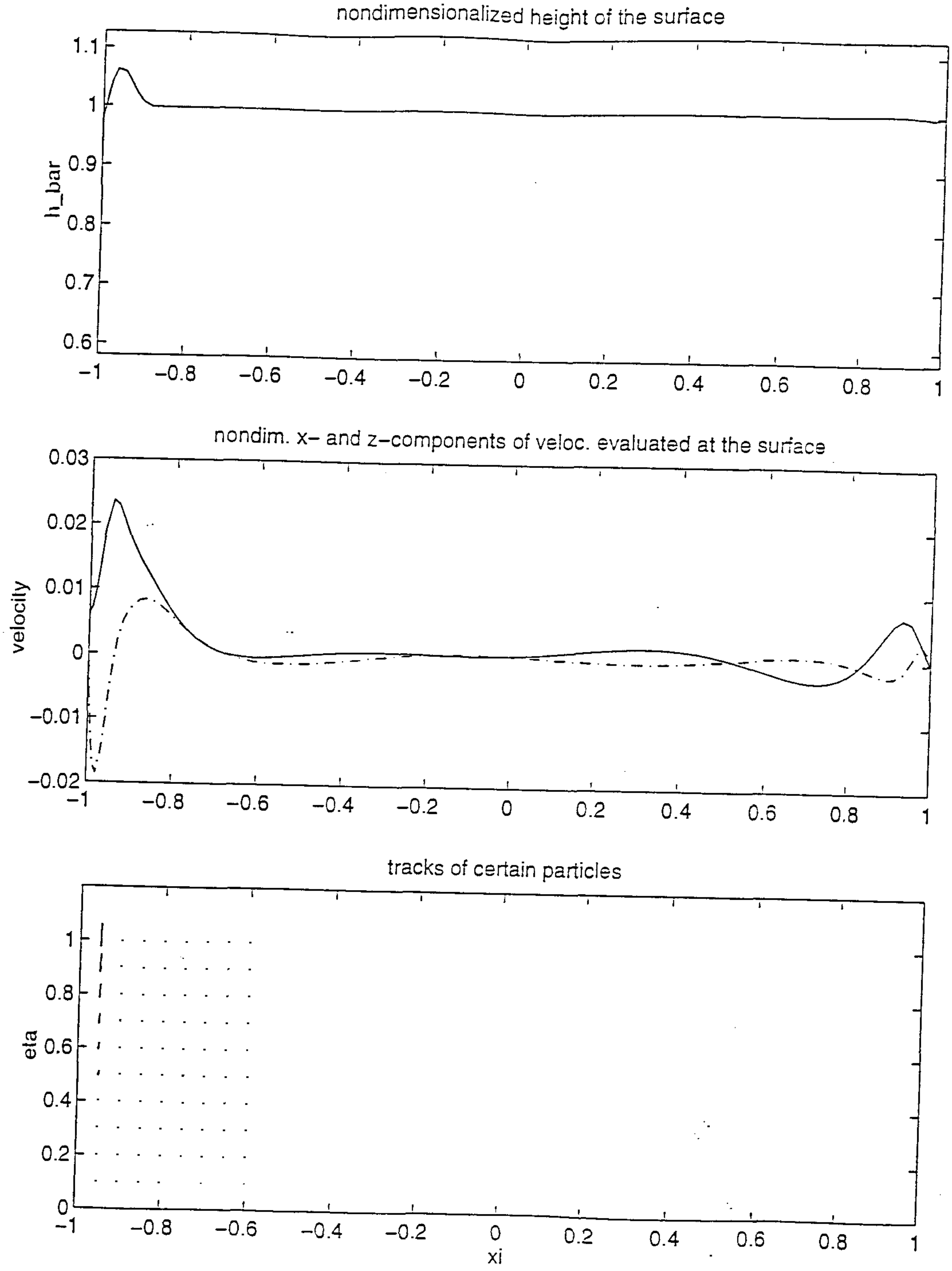


figure 6.3: Example 3, results after 50 timesteps



nondimensionalized height of the surface

nondim. x- and z-components of veloc. evaluated at the surface

tracks of certain particles

figure 6.3 cont.: Example 3, results after 100 timesteps

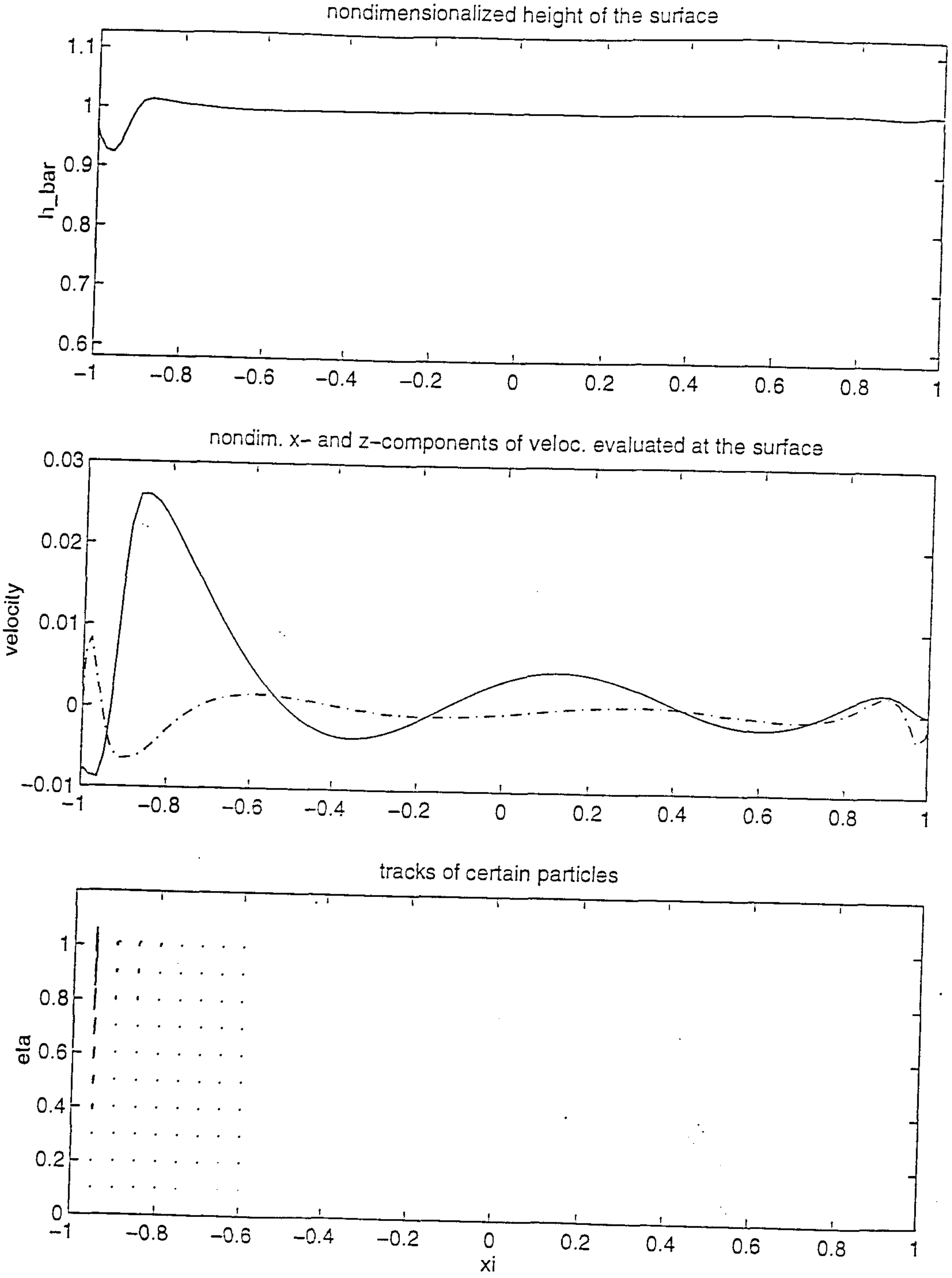
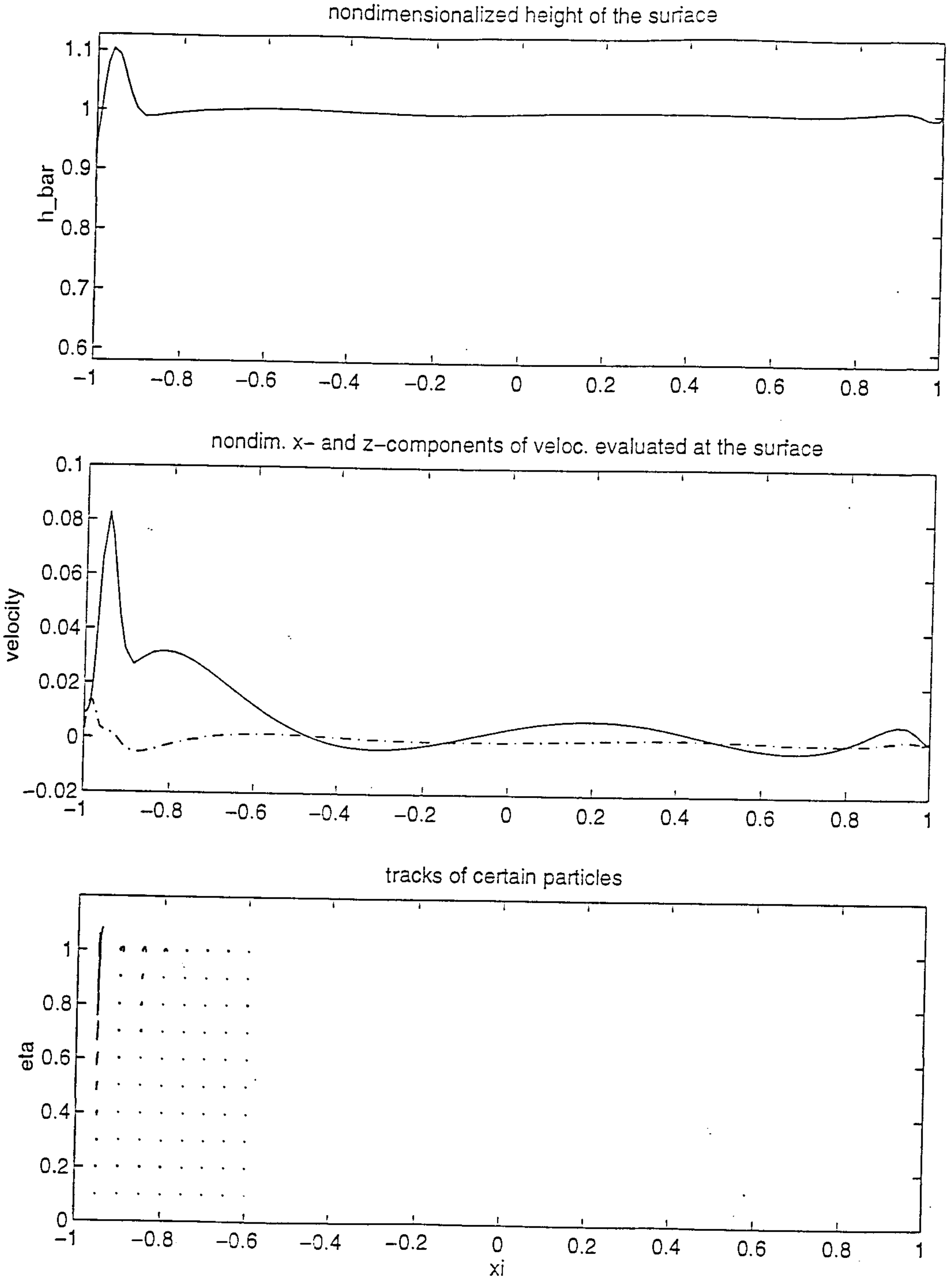


figure 6.3 cont.: Example 3, results after 150 timesteps



nondimensionalized height of the surface

nondim. x- and z-components of veloc. evaluated at the surface

tracks of certain particles

figure 6.3 cont.: Example 3, results after 200 timesteps

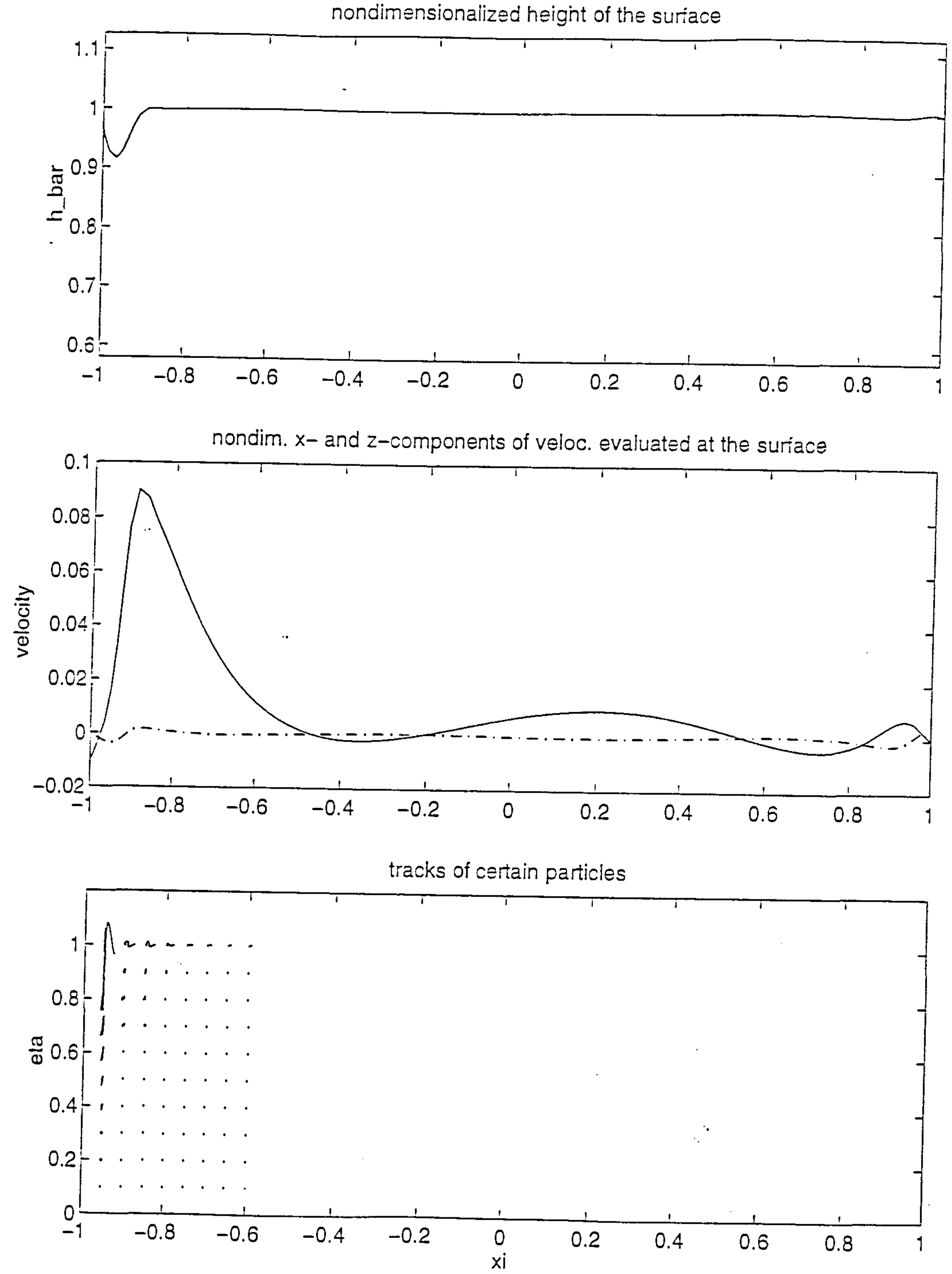
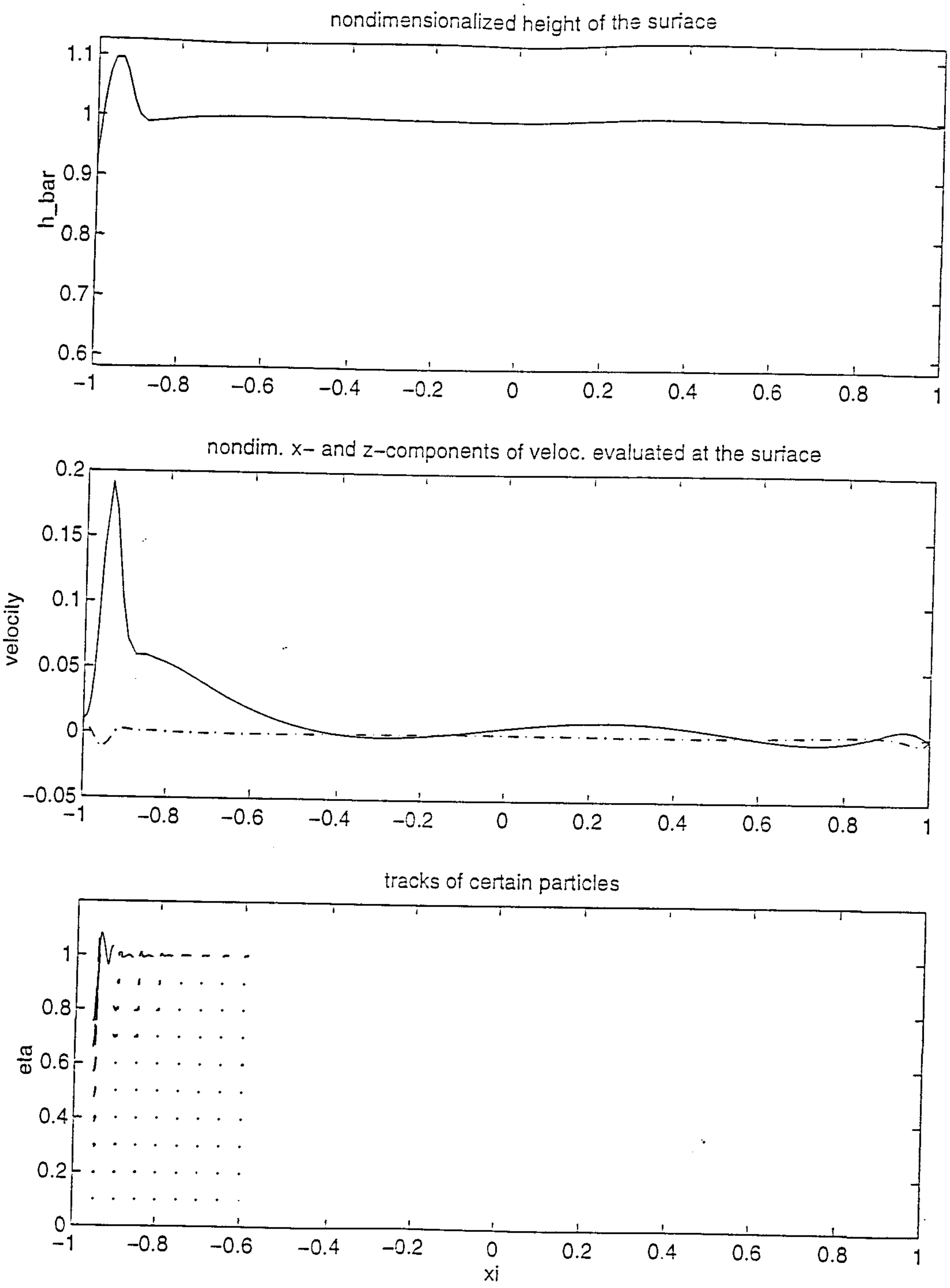


figure 6.3 cont.: Example 3, results after 250 timesteps


nondimensionalized height of the surface

nondim. x– and z–components of veloc. evaluated at the surface

tracks of certain particles

figure 6.3 cont.: Example 3, results after 300 timesteps

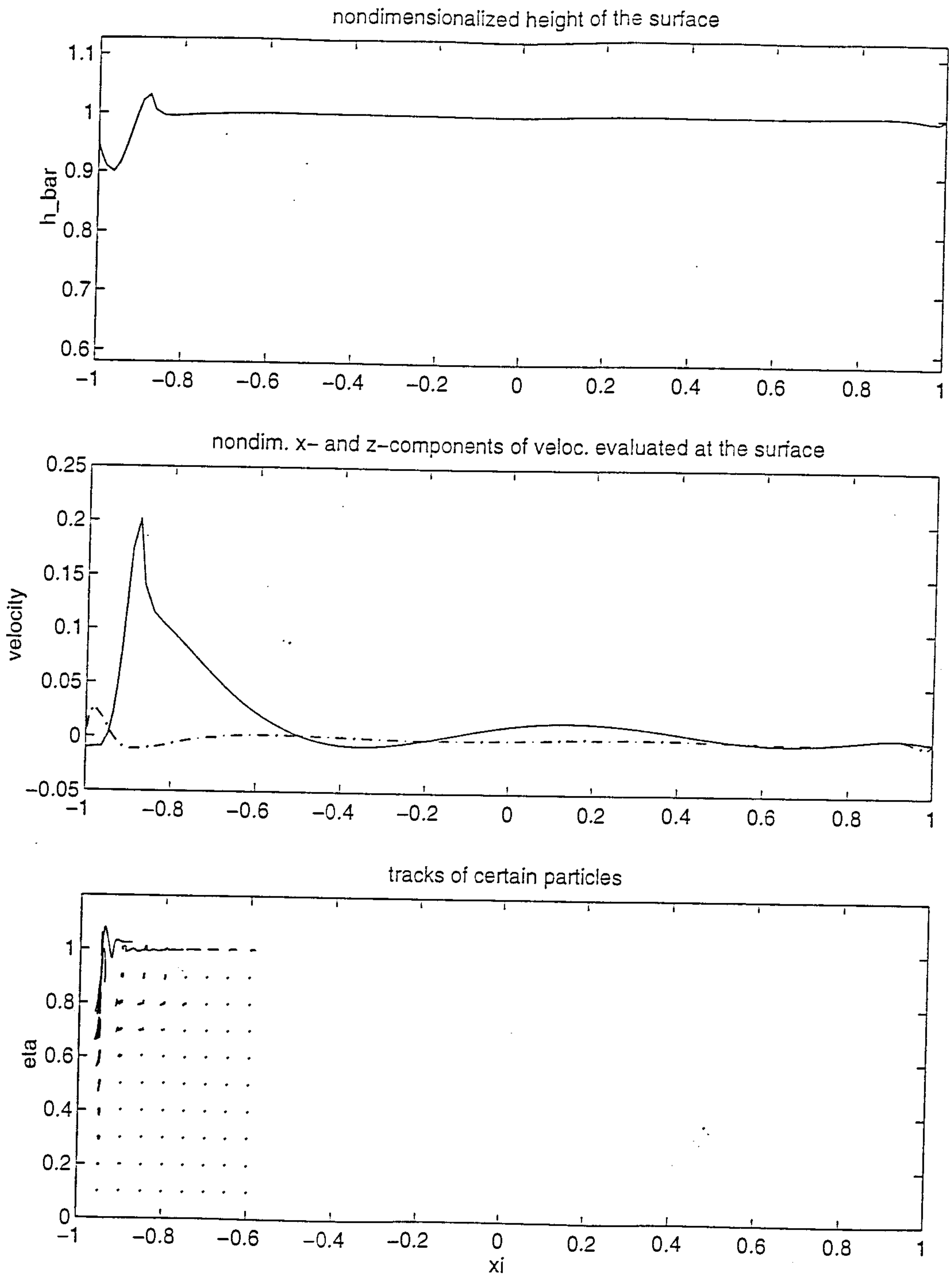


figure 6.3 cont.: Example 3, results after 350 timesteps

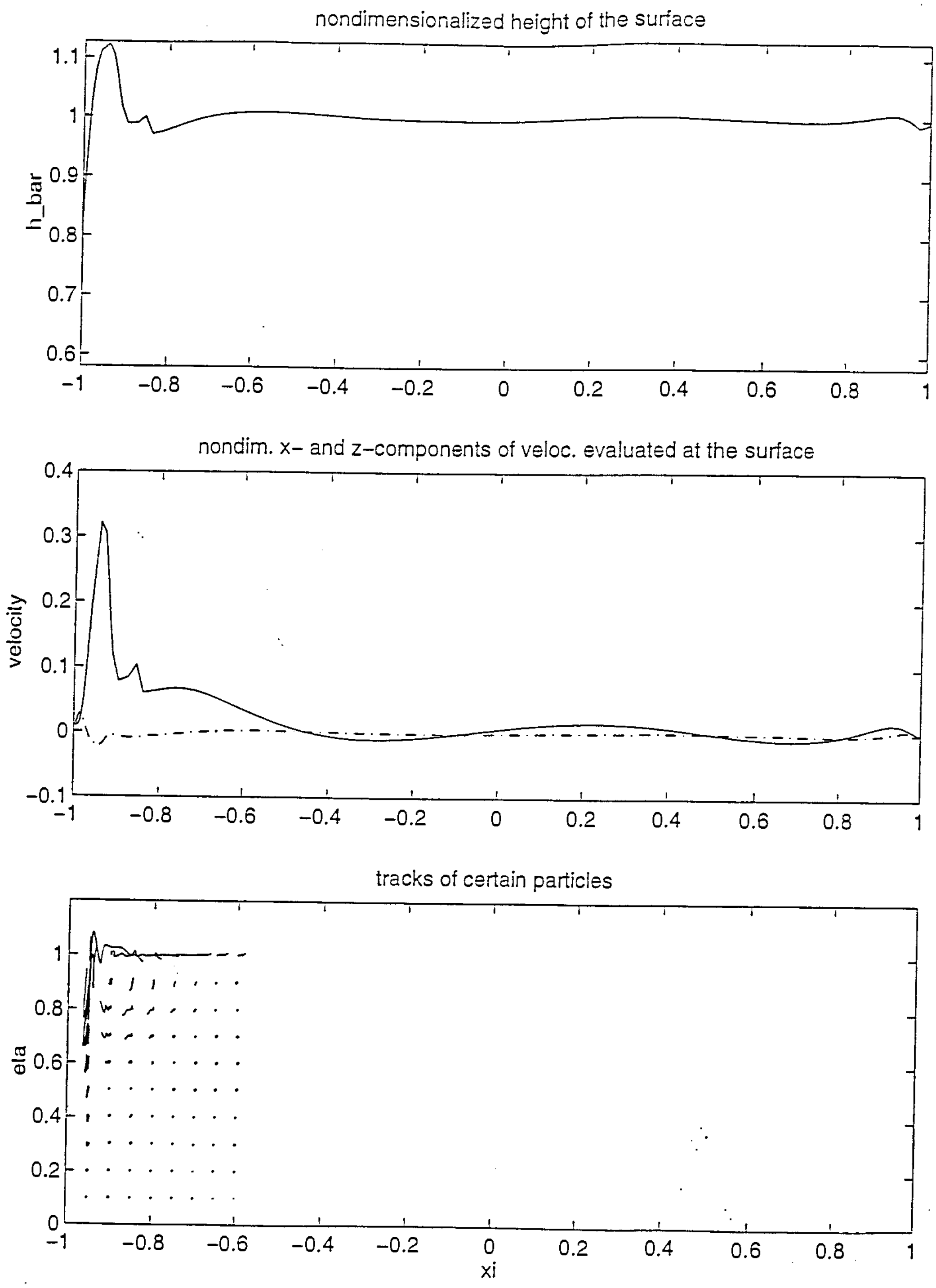


figure 6.3 cont.: Example 3, results after 400 timesteps

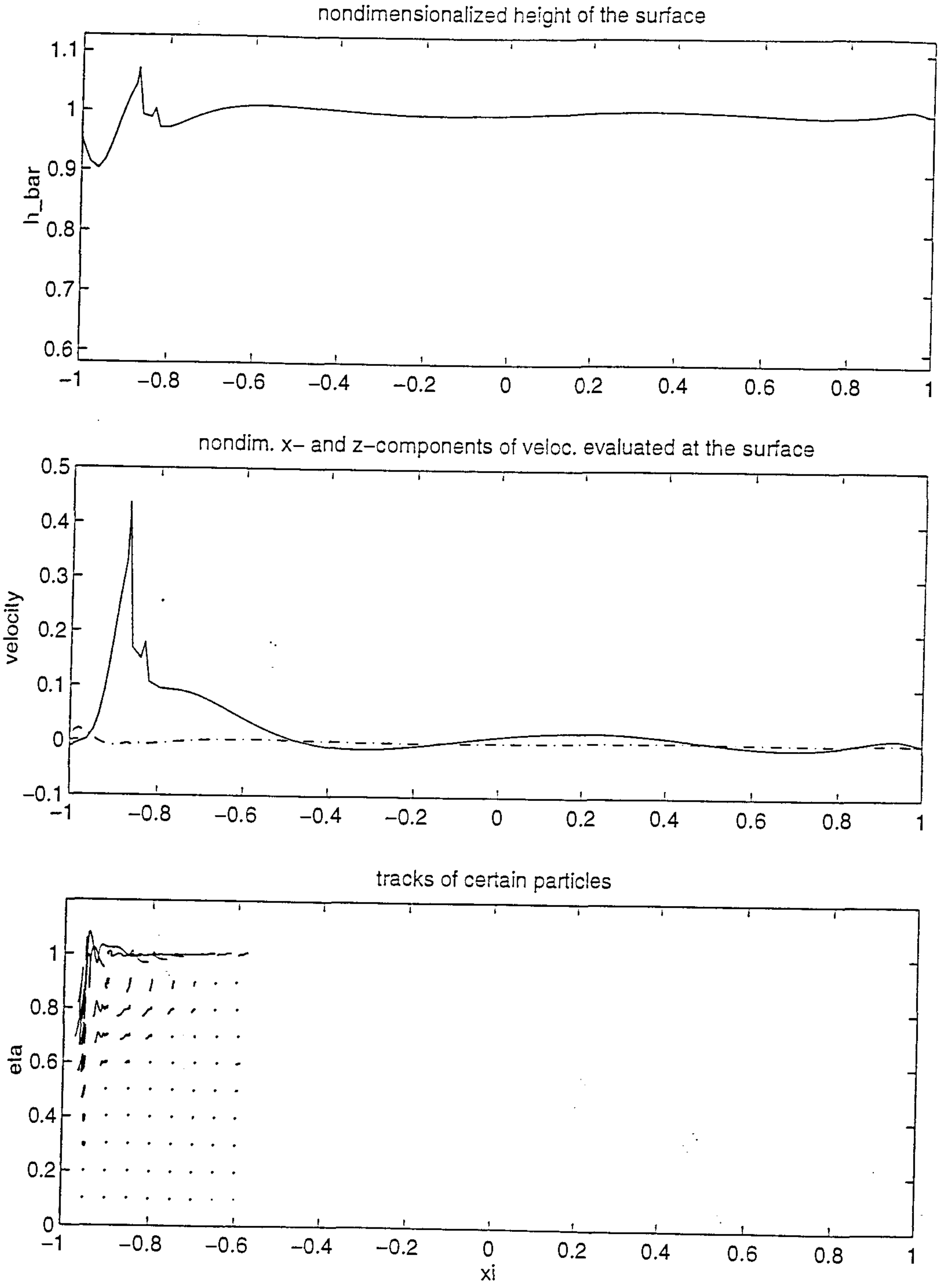
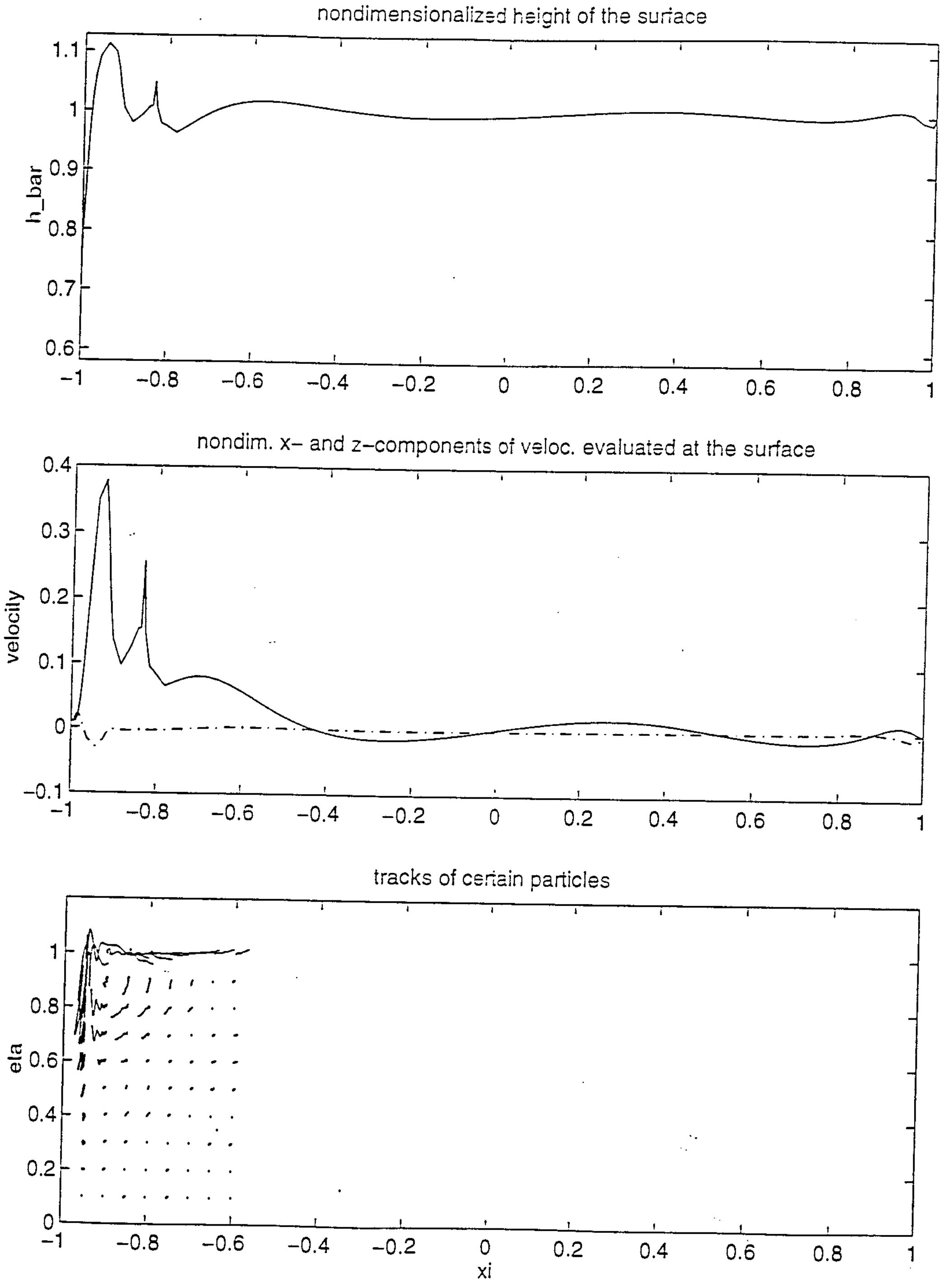


figure 6.3 cont.: Example 3, results after 450 timesteps



nondimensionalized height of the surface

nondim. x– and z–components of veloc. evaluated at the surface

tracks of certain particles

figure 6.3 cont.: Example 3, results after 500 timesteps

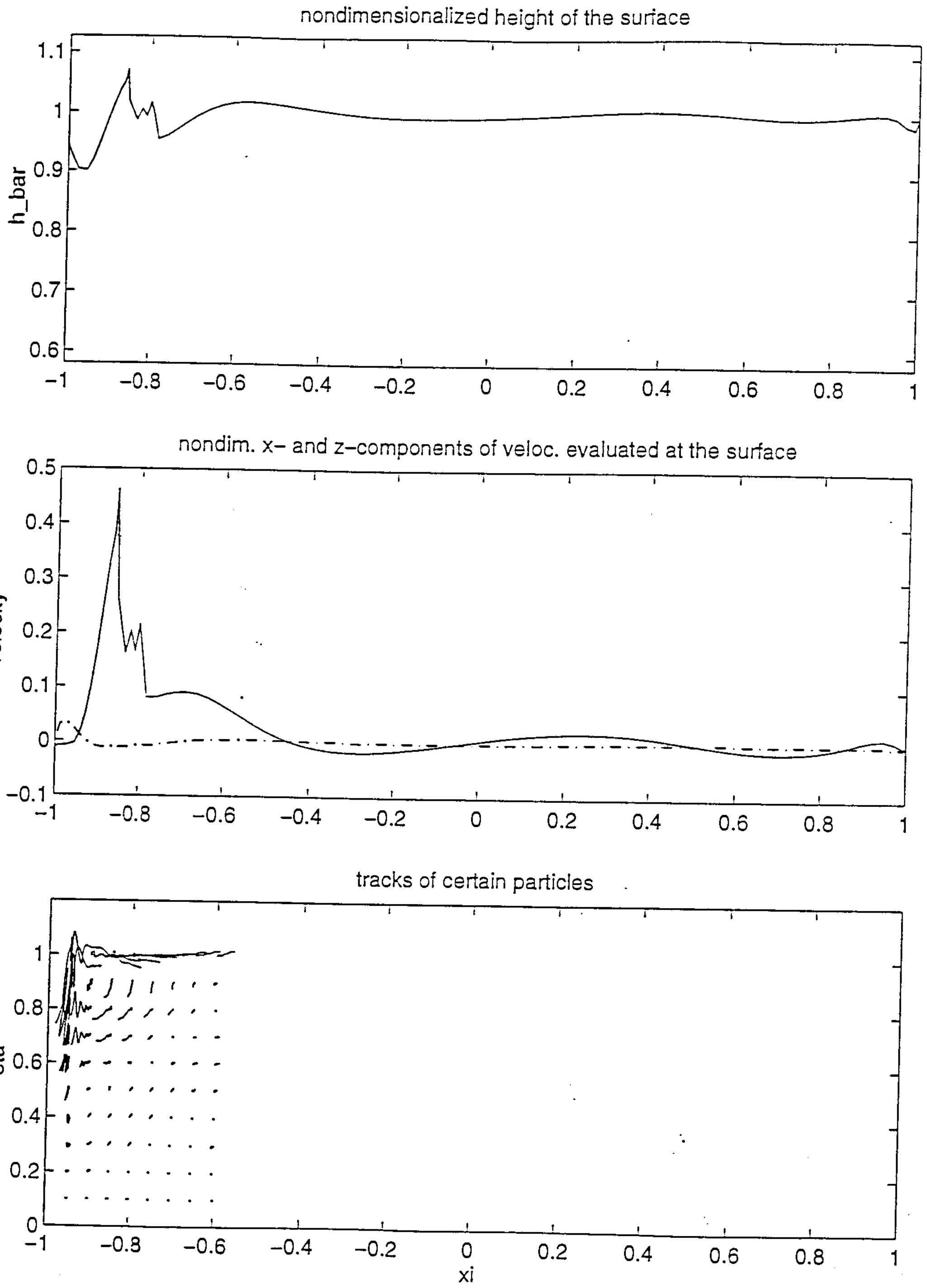
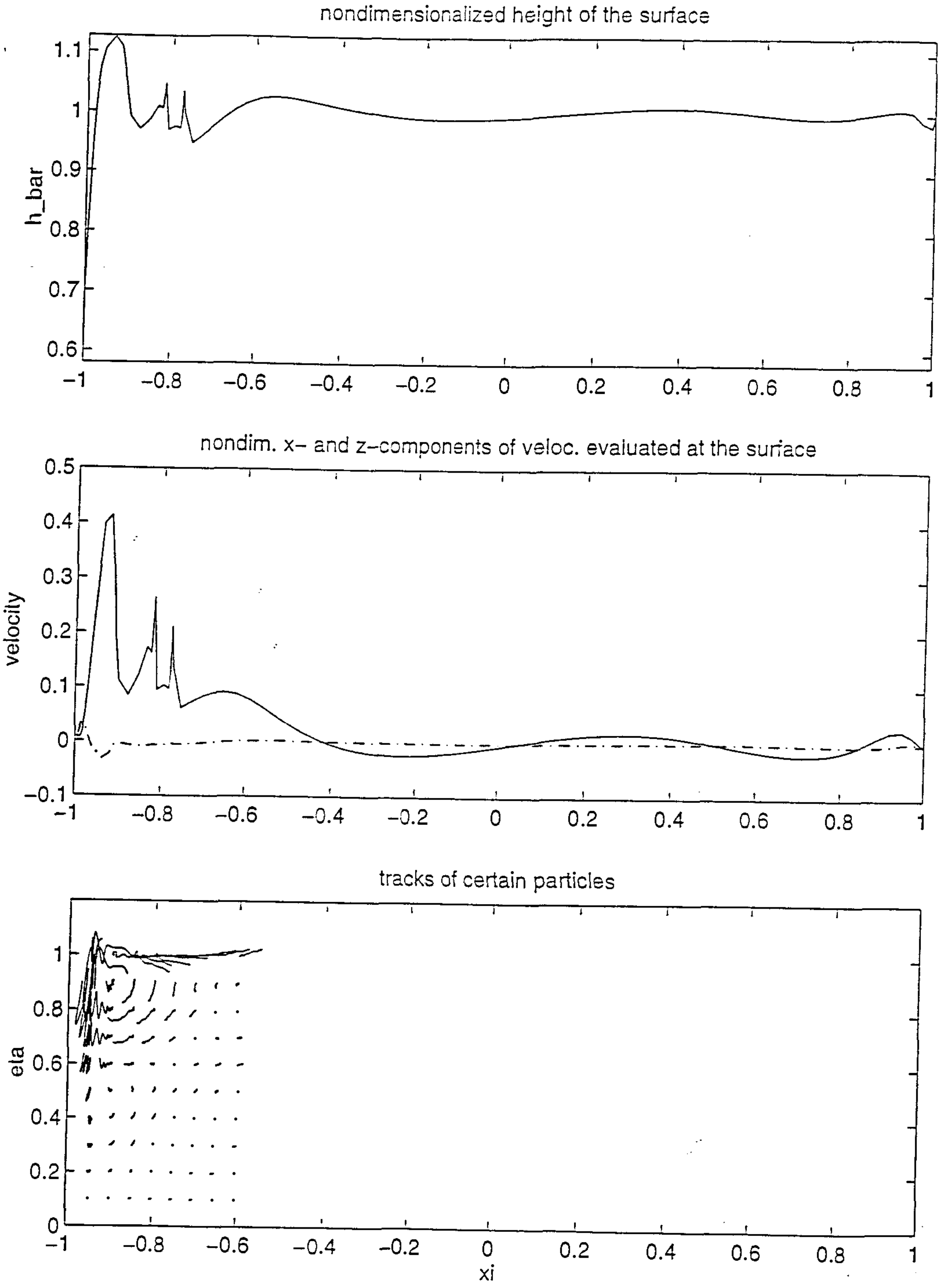


figure 6.3 cont.: Example 3, results after 550 timesteps


nondimensionalized height of the surface

nondim. x- and z-components of veloc. evaluated at the surface

tracks of certain particles

figure 6.3 cont.: Example 3, results after 600 timesteps

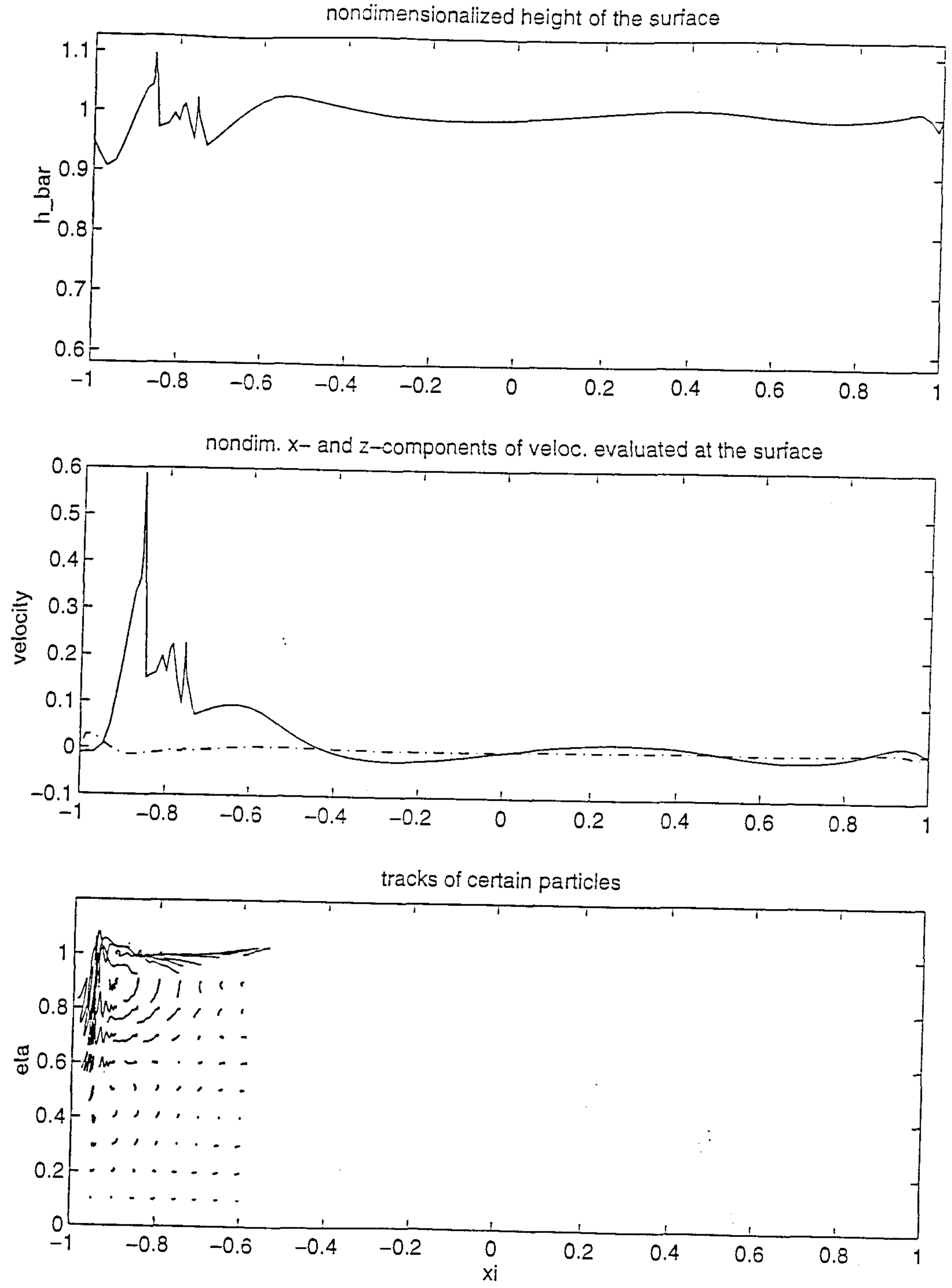
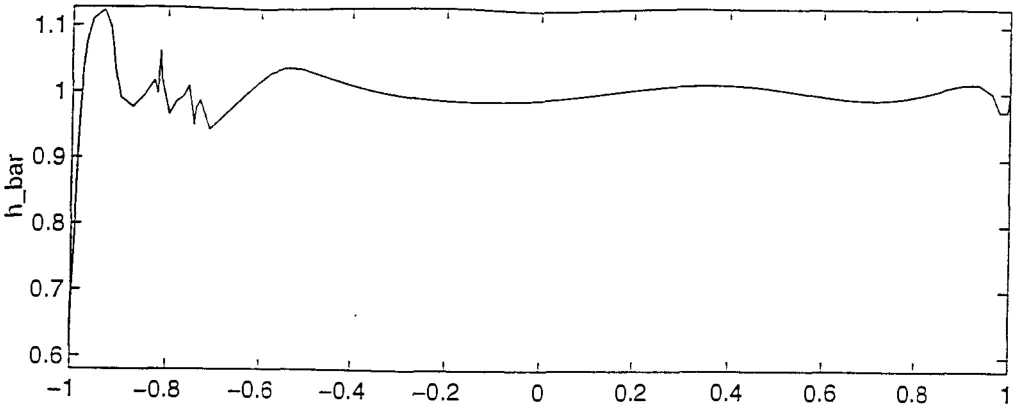
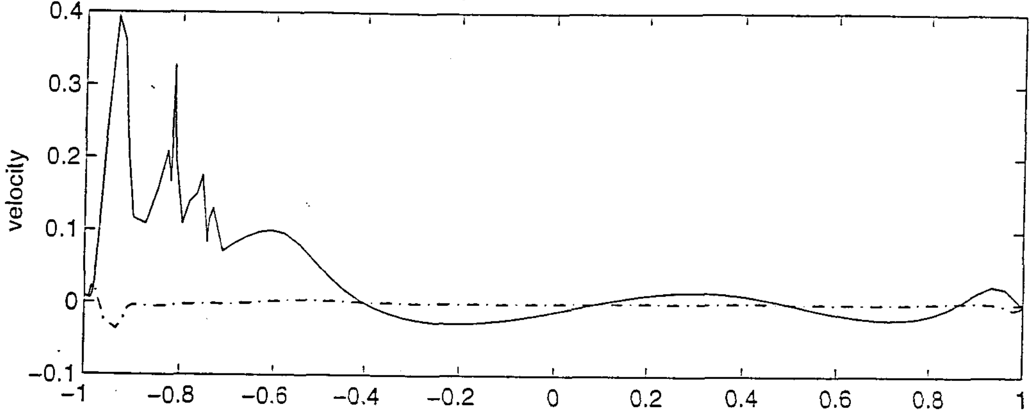


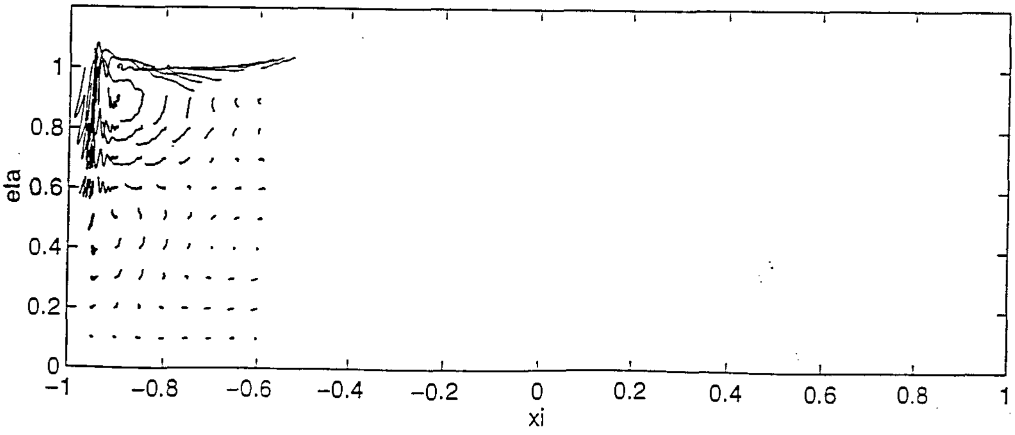
figure 6.3 cont.: Example 3, results after 650 timesteps



nondimensionalized height of the surface



nondim. x− and z−components of veloc. evaluated at the surface



tracks of certain particles

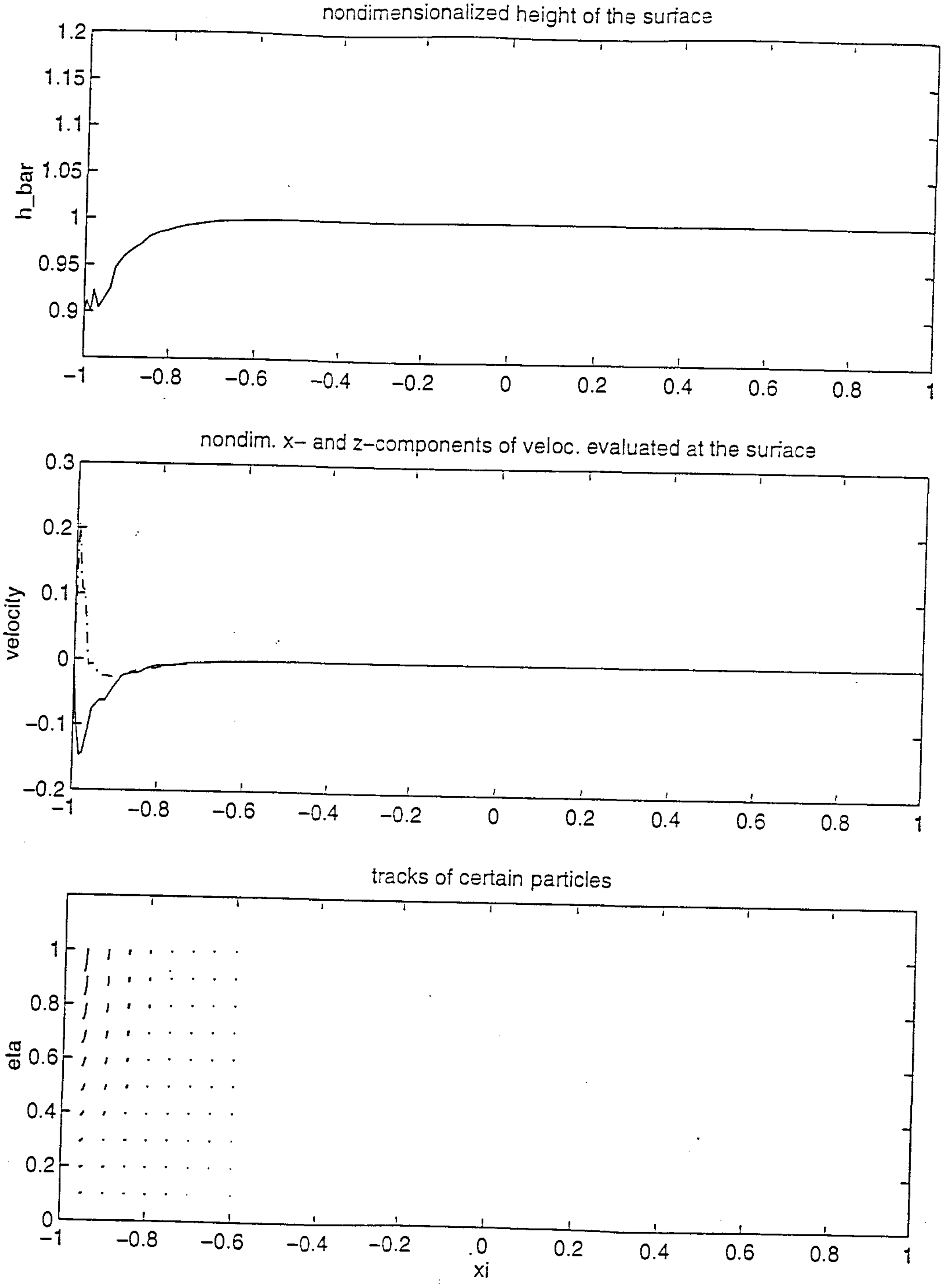
figure 6.3 cont.: Example 3, results after 700 timesteps

6.4 Example 4

This example demonstrates the high capability of the finite element model as it was intoduced in chapter 5. The initial condition is $\mathbf{v} = 0$. The wave board completes a whole cycle (thereby giving the water a large push) and then stops.

basis and weight functions used: finite element model
pattern of finite elements used: equation (5.3)
number of computed approximate solutions (according to section 5.3): 2
frequency of the wave board: one cycle in 2 seconds
max. velocity of the wave board at height of $6\,m$: $2\,\frac{m}{s}$
$nb_x \; = \; 50$
$nb_z \; = \; 5$
$U_0 \; = \; 10\,\frac{m}{s}$
timestep $\Delta t \; = \; 0.02\,s$
plotted results every 50 timesteps (one plot after each second), beginning with timestep 50
We employed the idea of rising and falling surface at the boundary.

Remark : From time to time, we notice little disturbances (little "corners") in the surface and in the velocity field. These disturbances are due mainly to the "corner"-problem discussed in section 5.3 and are not due to numerical instabilities. However, they could cause numerical instabilities as it will be seen in example 5 of this chapter. Nevertheless, we see that the results already are much improved compared to the first three examples.



nondimensionalized height of the surface

nondim. x– and z–components of veloc. evaluated at the surface

tracks of certain particles

figure 6.4: Example 4, results after 50 timesteps

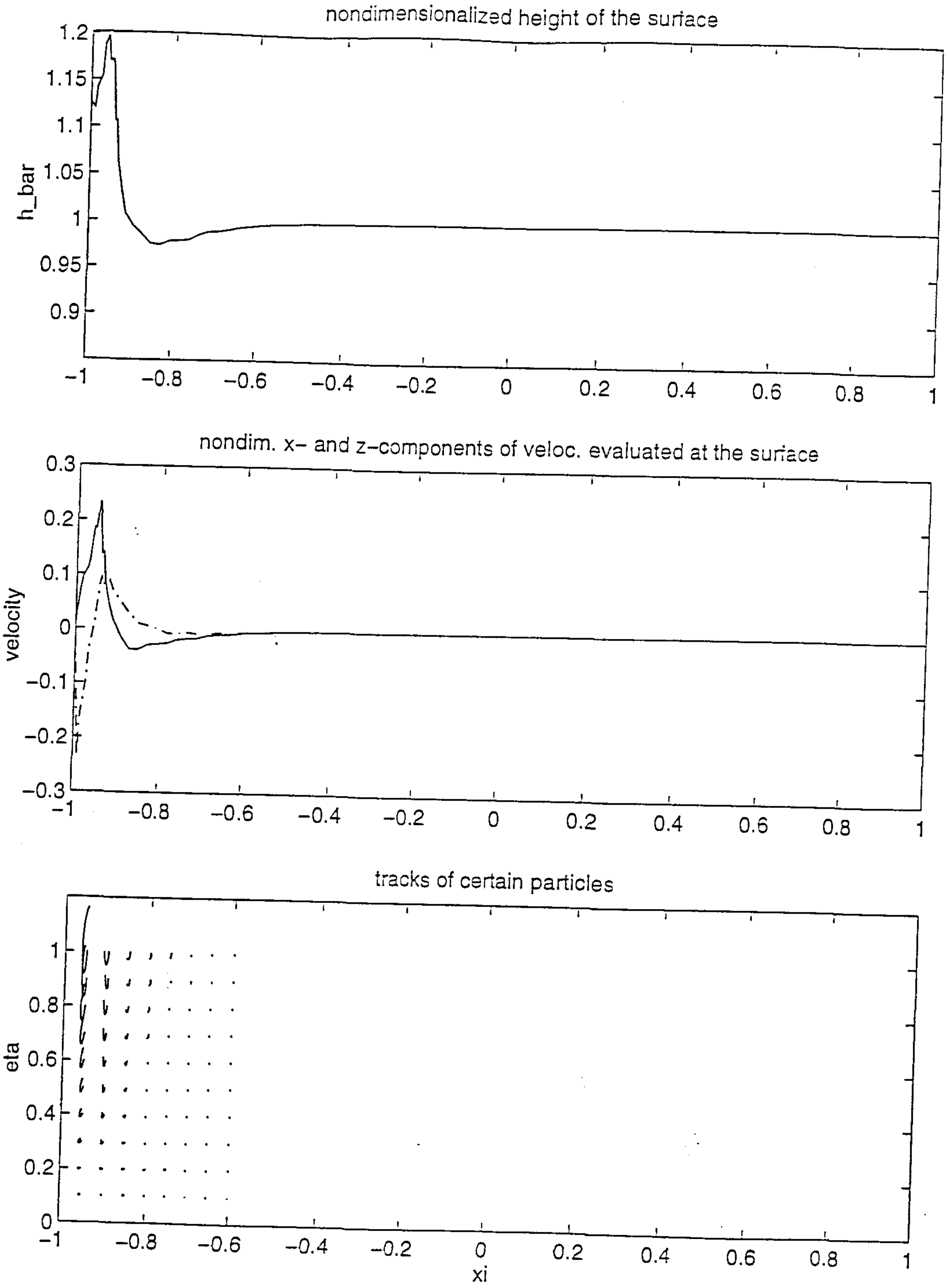


figure 6.4 cont.: Example 4, results after 100 timesteps

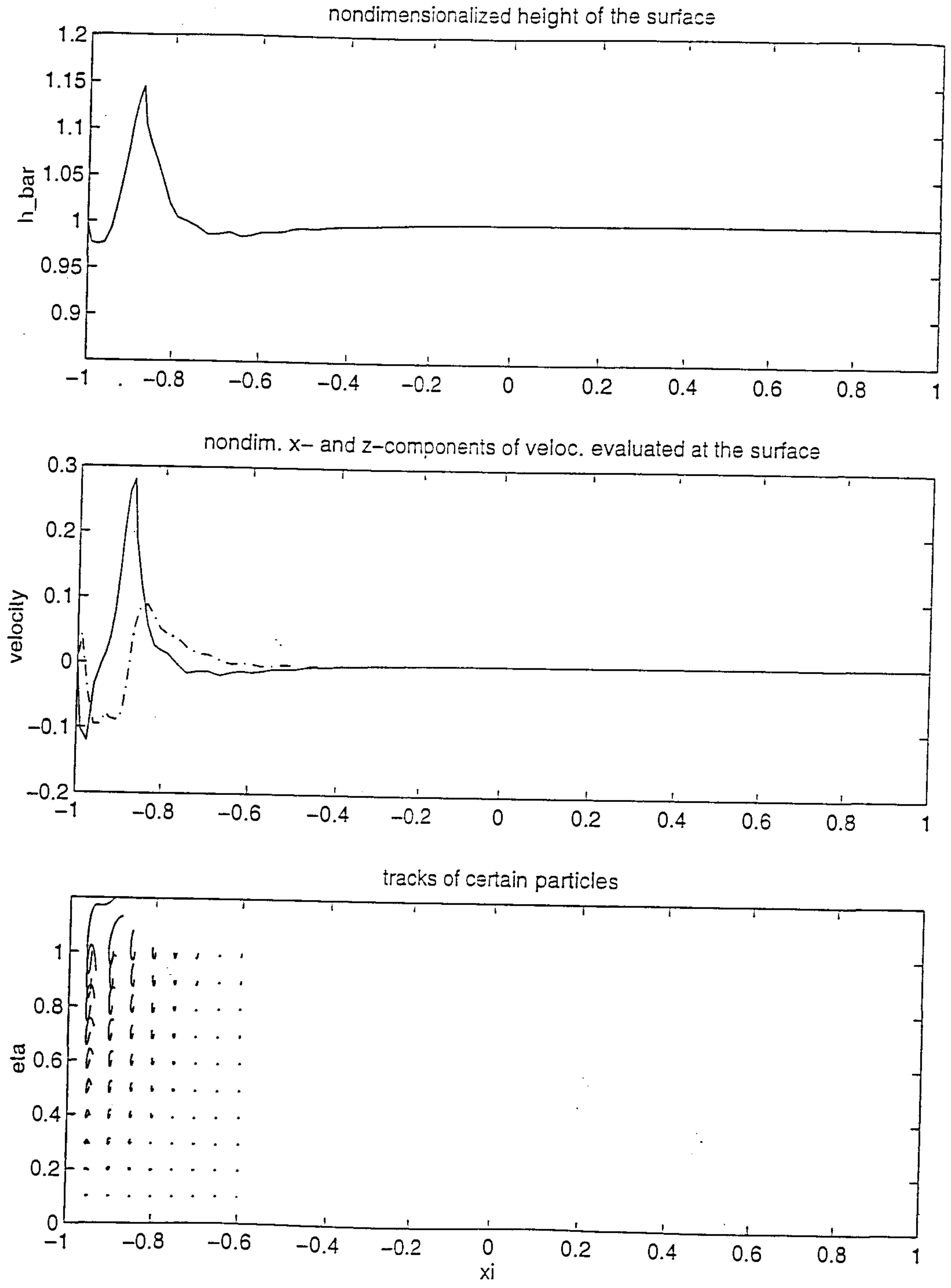


figure 6.4 cont.: Example 4, results after 150 timesteps

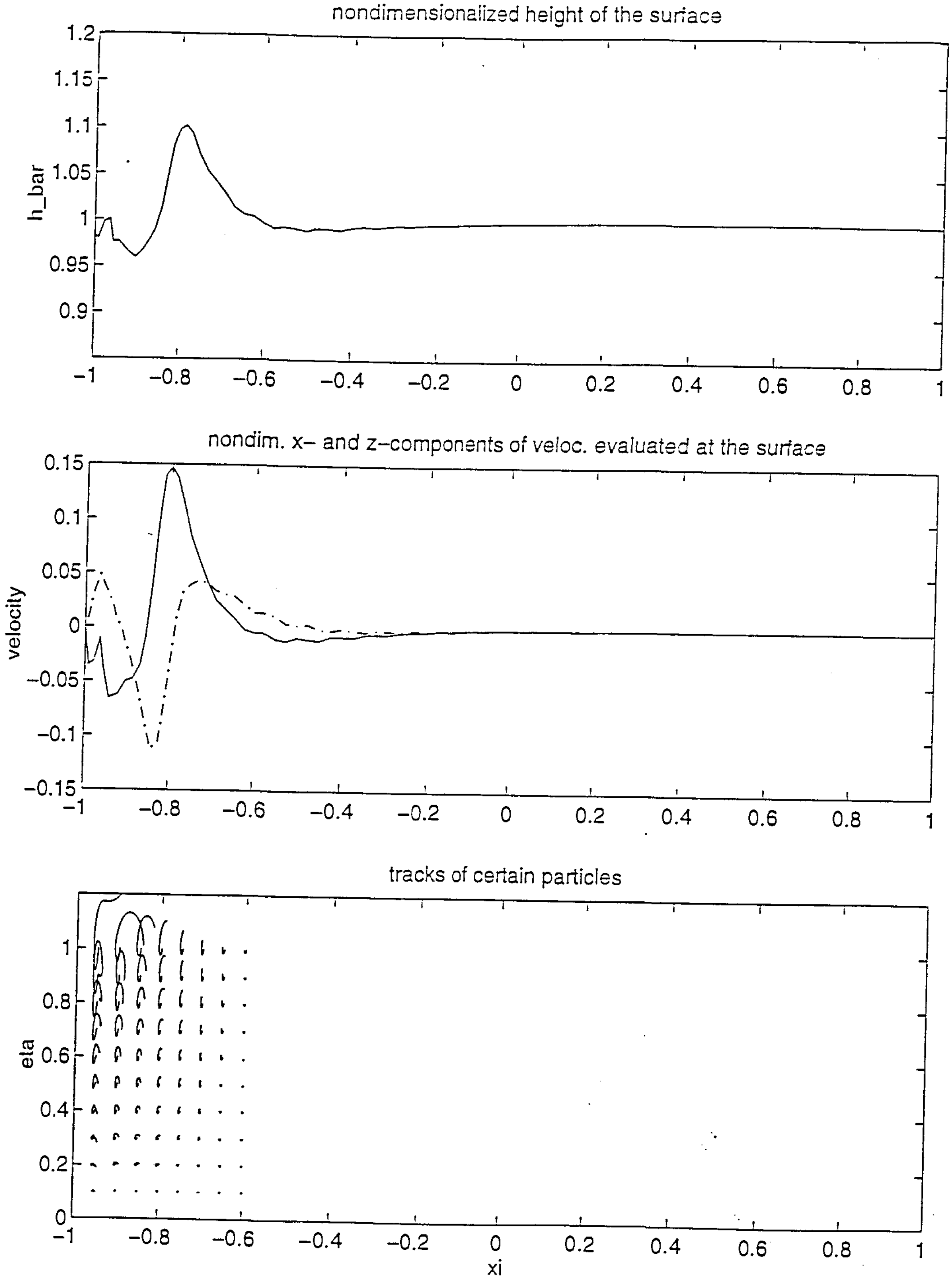


figure 6.4 cont.: Example 4, results after 200 timesteps

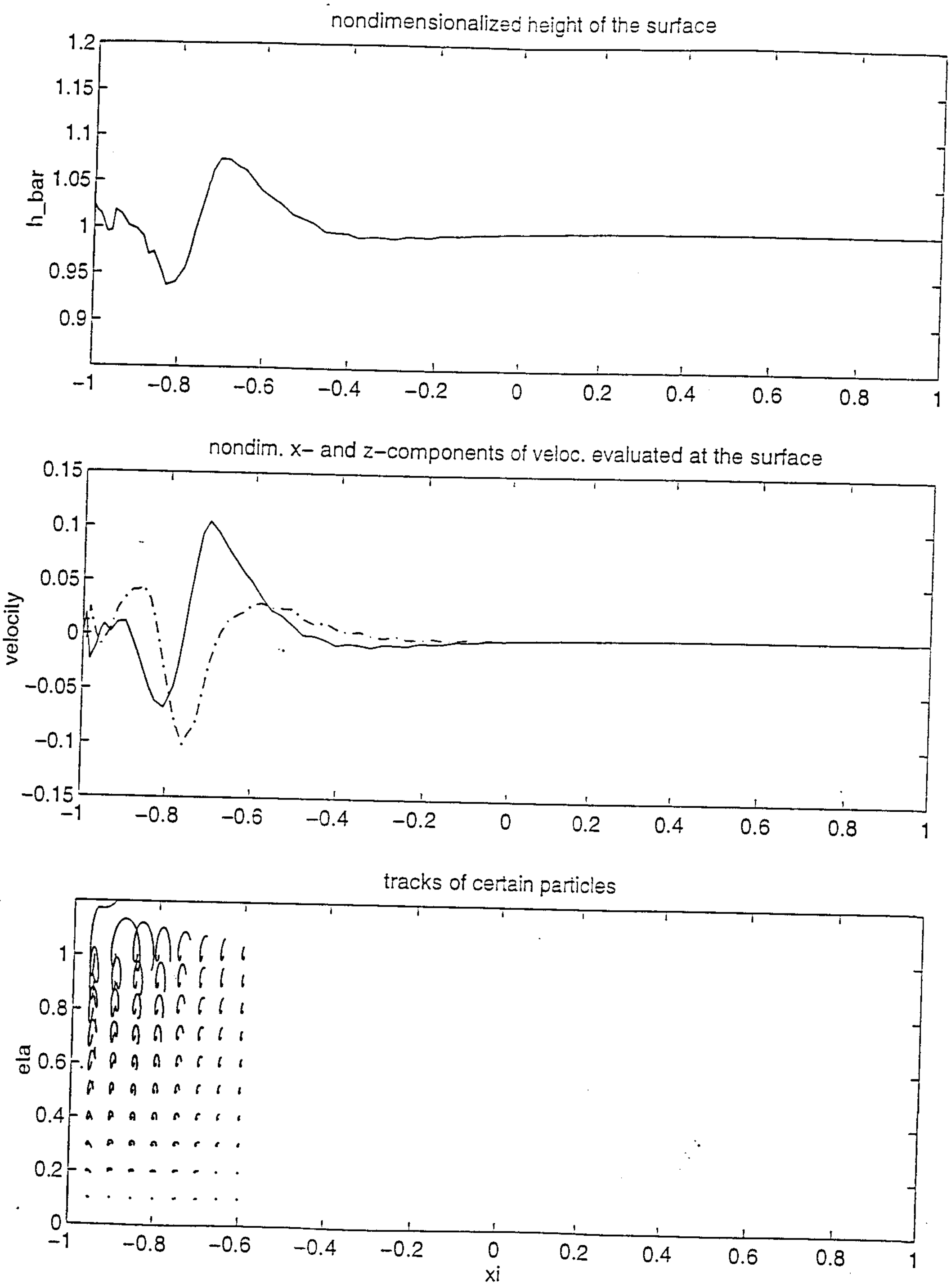


figure 6.4 cont.: Example 4, results after 250 timesteps

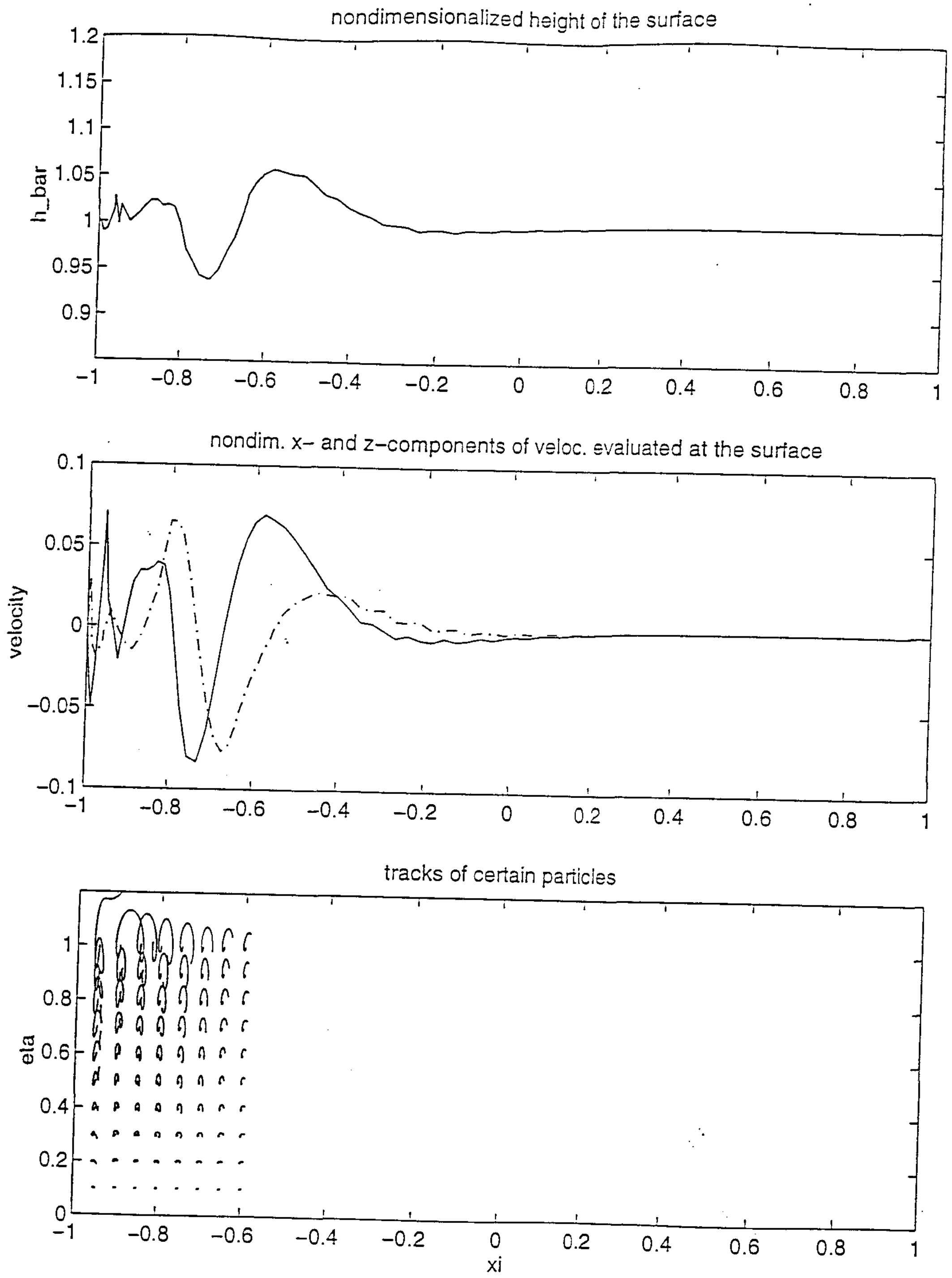
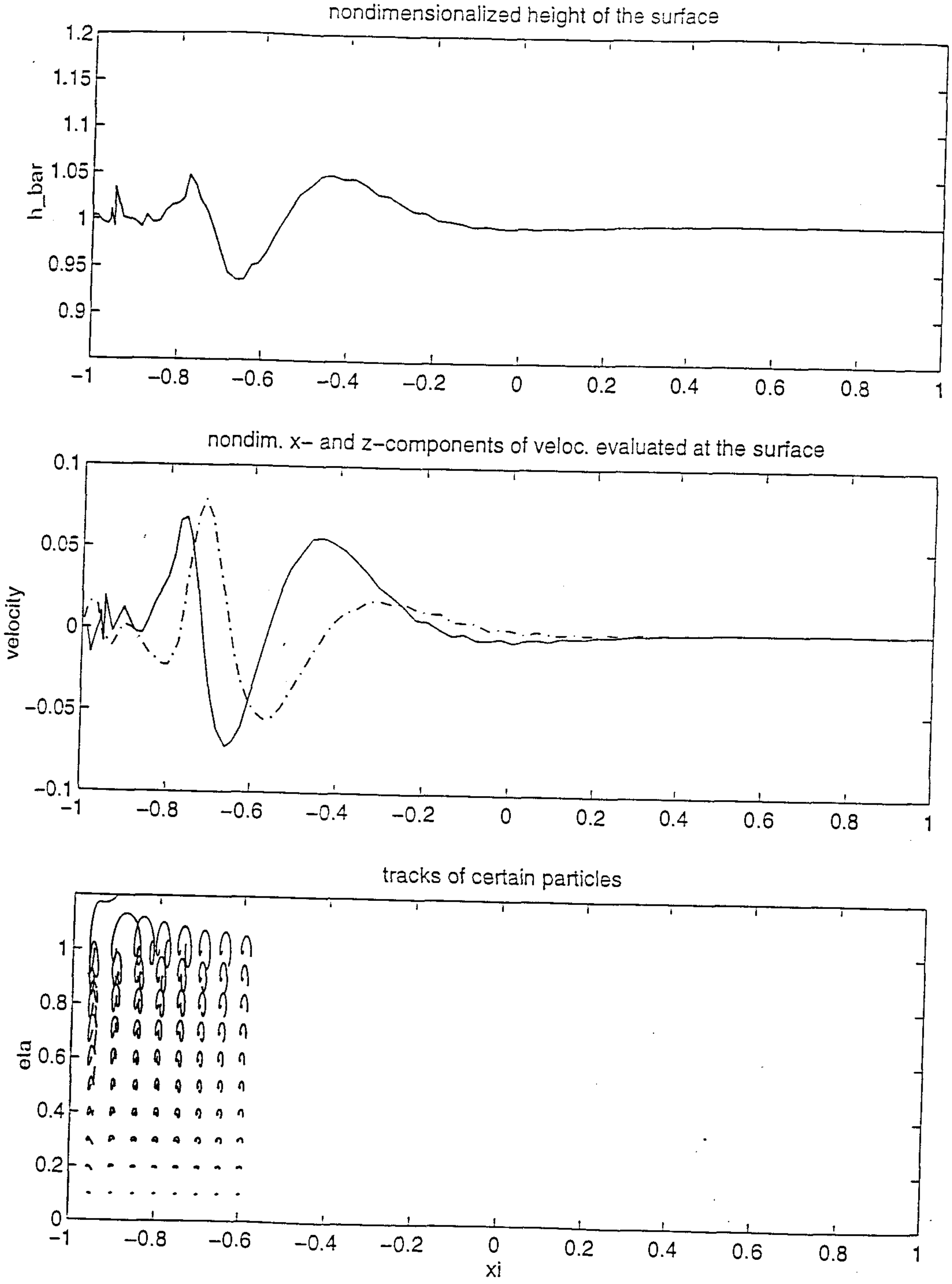


figure 6.4 cont.: Example 4, results after 300 timesteps


nondimensionalized height of the surface

nondim. x– and z–components of veloc. evaluated at the surface

tracks of certain particles

figure 6.4 cont.: Example 4, results after 350 timesteps

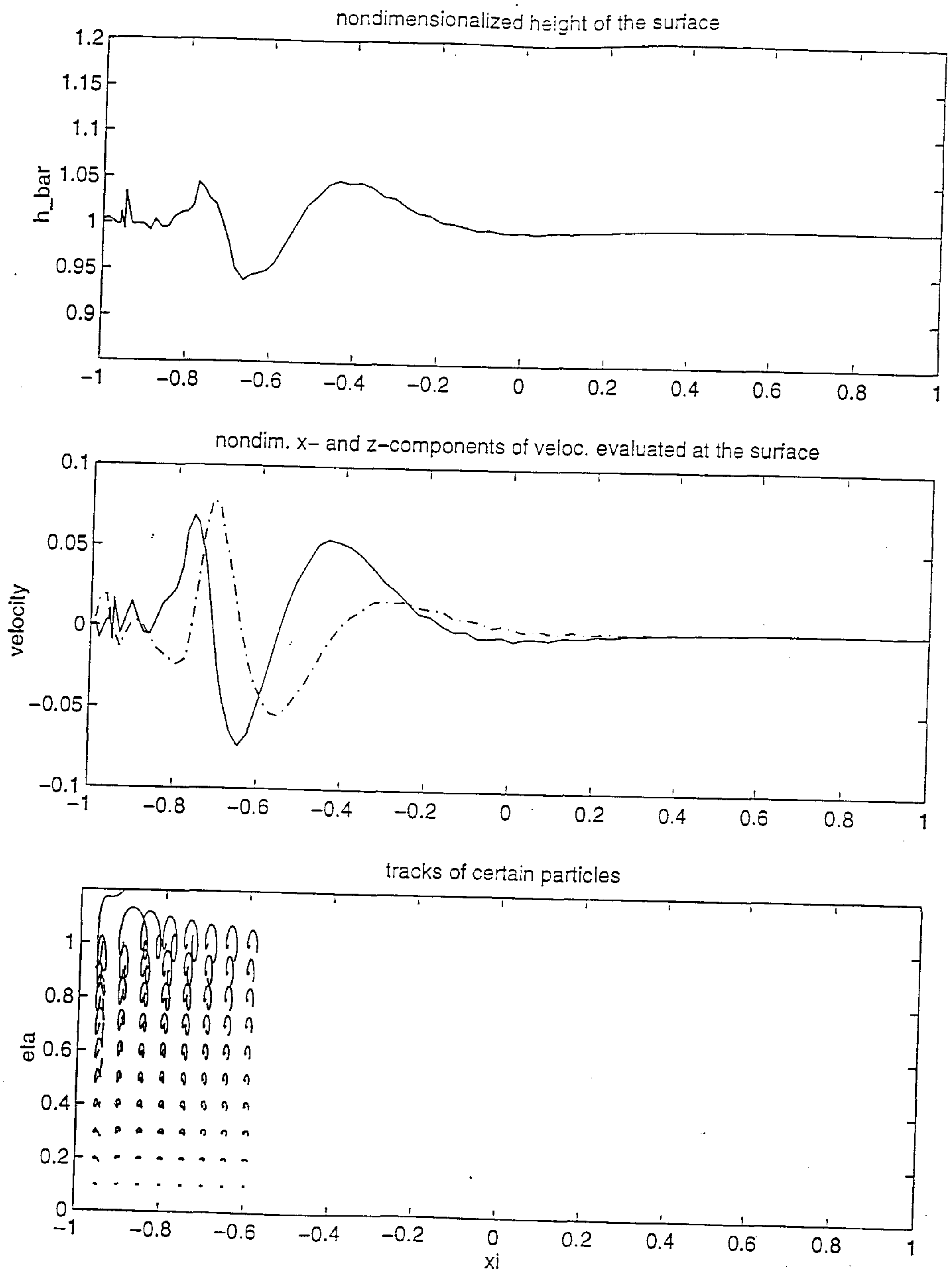


figure 6.4 cont.: Example 4, results after 400 timesteps

6.5 Example 5

For this experiment we employed the finite element model. The initial condition is $\mathbf{v} = 0$. The wave board moves periodically back and forth starting with a very small and successively increasing amplitude.

basis and weight functions used: finite element model
pattern of finite elements used: equation (5.3)
number of computed approximate solutions (according to section 5.3): 2
frequency of the wave board: one cycle in 2 seconds
max. velocity of the wave board at height of $6\,m$: $2\,\frac{m}{s}$
$nb_x = 50$
$nb_z = 5$
$U_0 = 10\,\frac{m}{s}$
timestep $\Delta t = 0.02\,s$
plotted results every 50 timesteps (one plot after each second), beginning with timestep 50
We employed the idea of rising and falling surface at the boundary.

Remark :

Now, we may compare this experiment with example 1. We notice that the only difference between these two experiments is the kind of basis functions employed. The differences in the results are considerable: one immediately recognizes that employing the finite element-model yields far more realistic results than employing the polynomial model does.

The right end of the wave tank remains undisturbed by the motion of the wave board.

Moreover, we see that the z-component of the velocity looks much more realistic in the finite element experiment than in the polynomial experiment.

Finally we see that numerical instabilities occur when the amplitude of the moving wave board becomes too large. For this case, the model is still too weak because the "little corners" due to basis functions become too large. To prevent this occurance, the basis functions should be more dense, which is equivalent to having more basis functions.

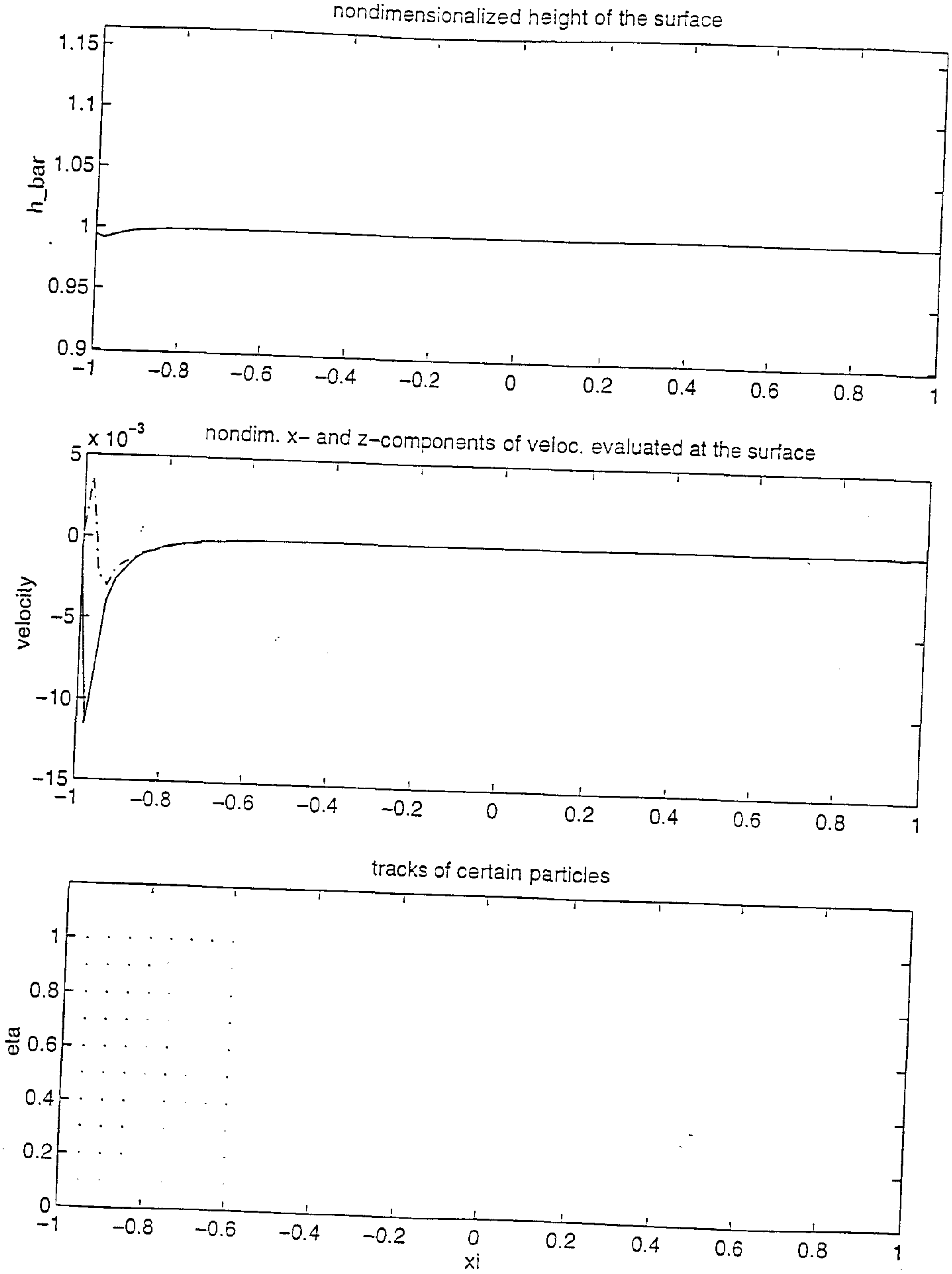


figure 6.5: Example 5, results after 50 timesteps

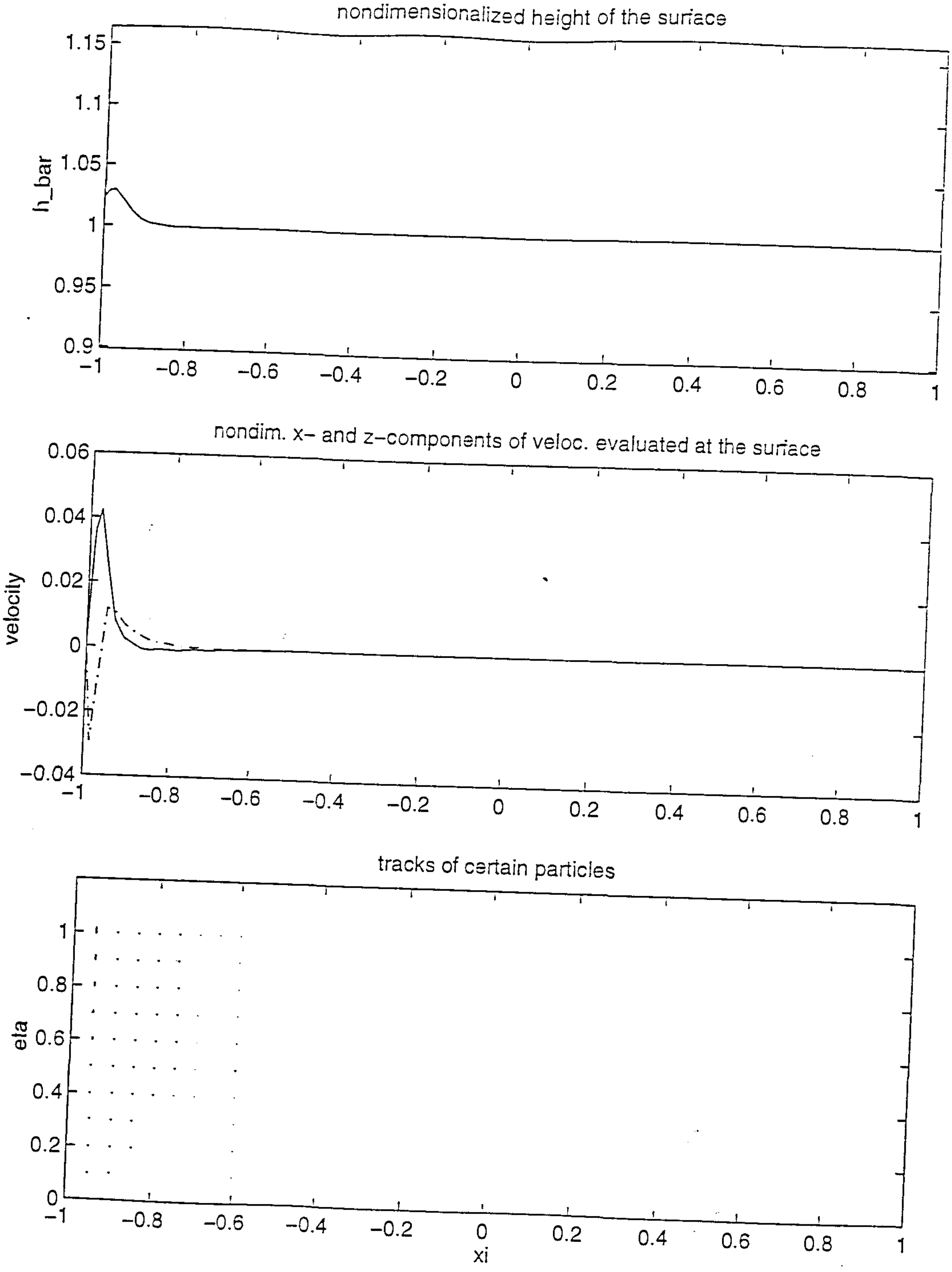
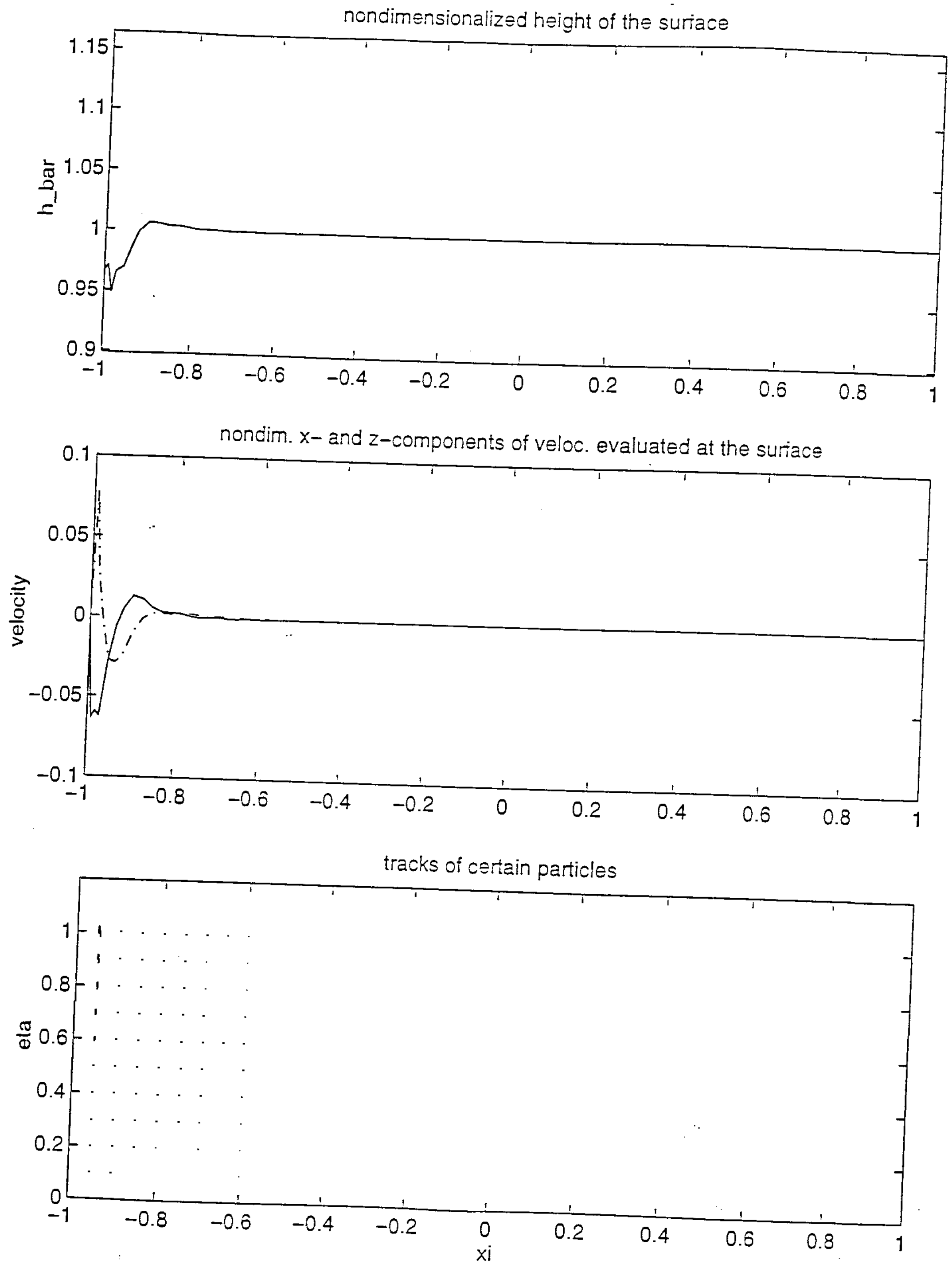


figure 6.5 cont.: Example 5, results after 100 timesteps


nondimensionalized height of the surface

nondim. x- and z-components of veloc. evaluated at the surface

tracks of certain particles

figure 6.5 cont.: Example 5, results after 150 timesteps

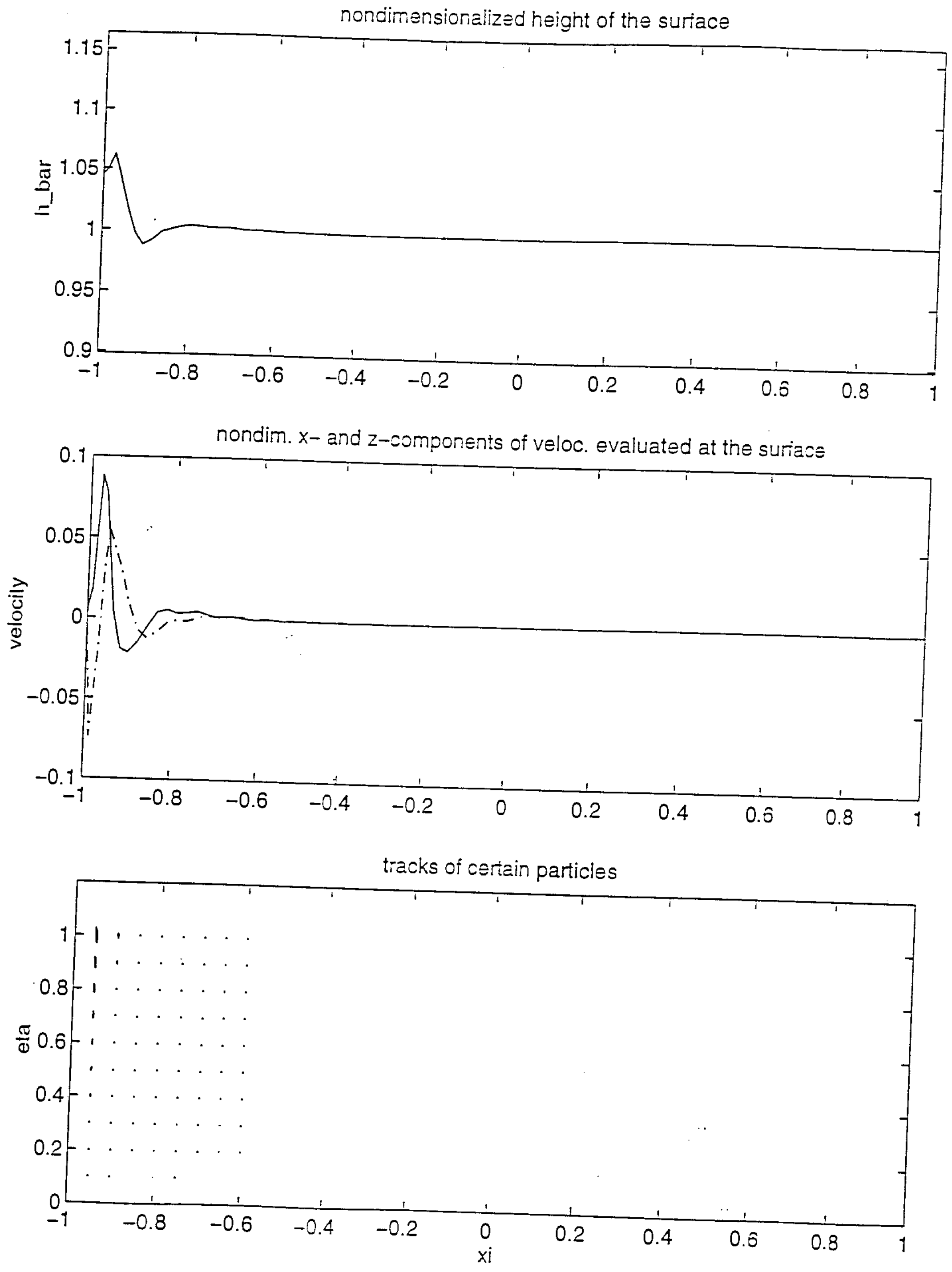


figure 6.5 cont.: Example 5, results after 200 timesteps

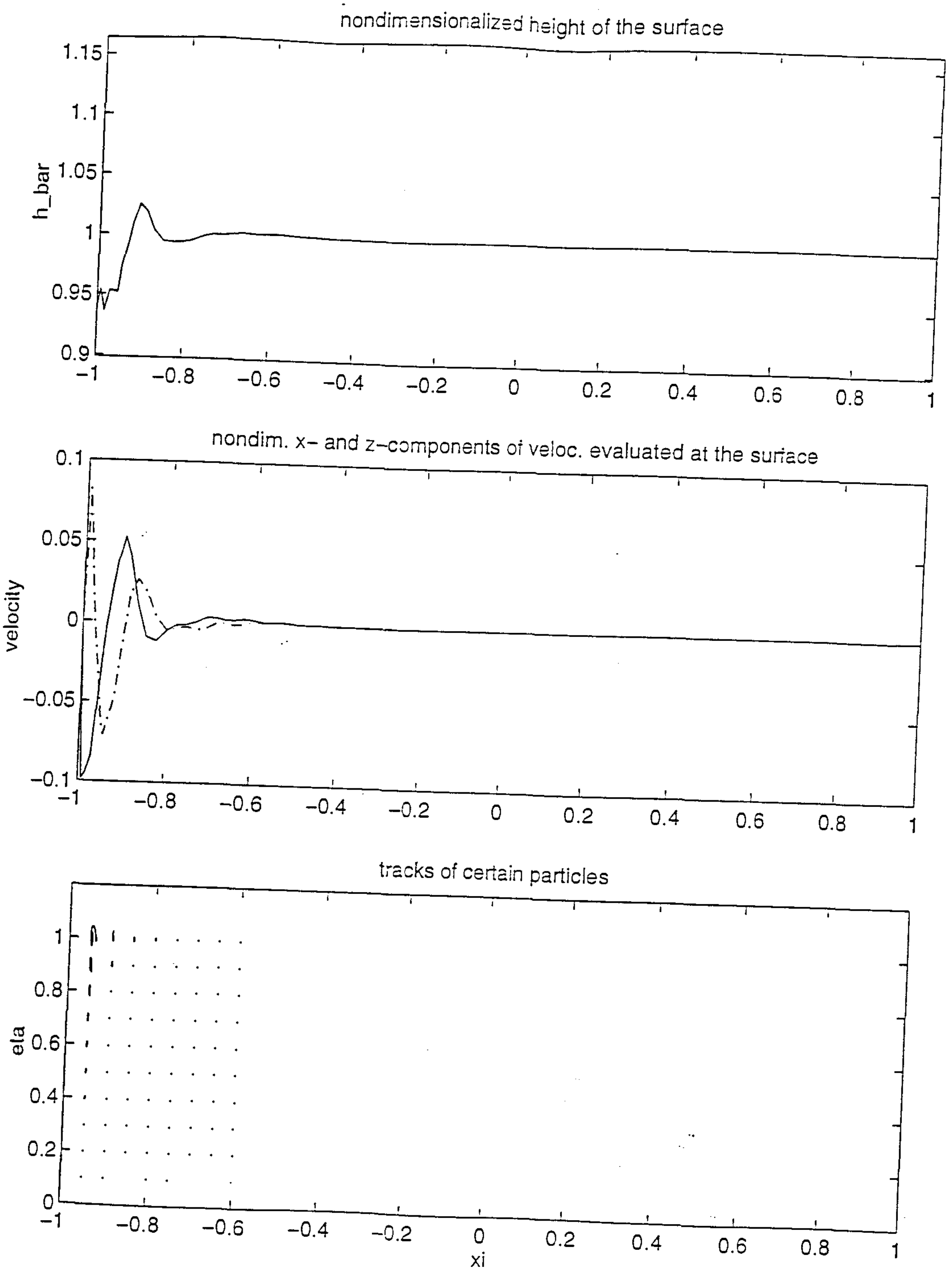


figure 6.5 cont.: Example 5, results after 250 timesteps

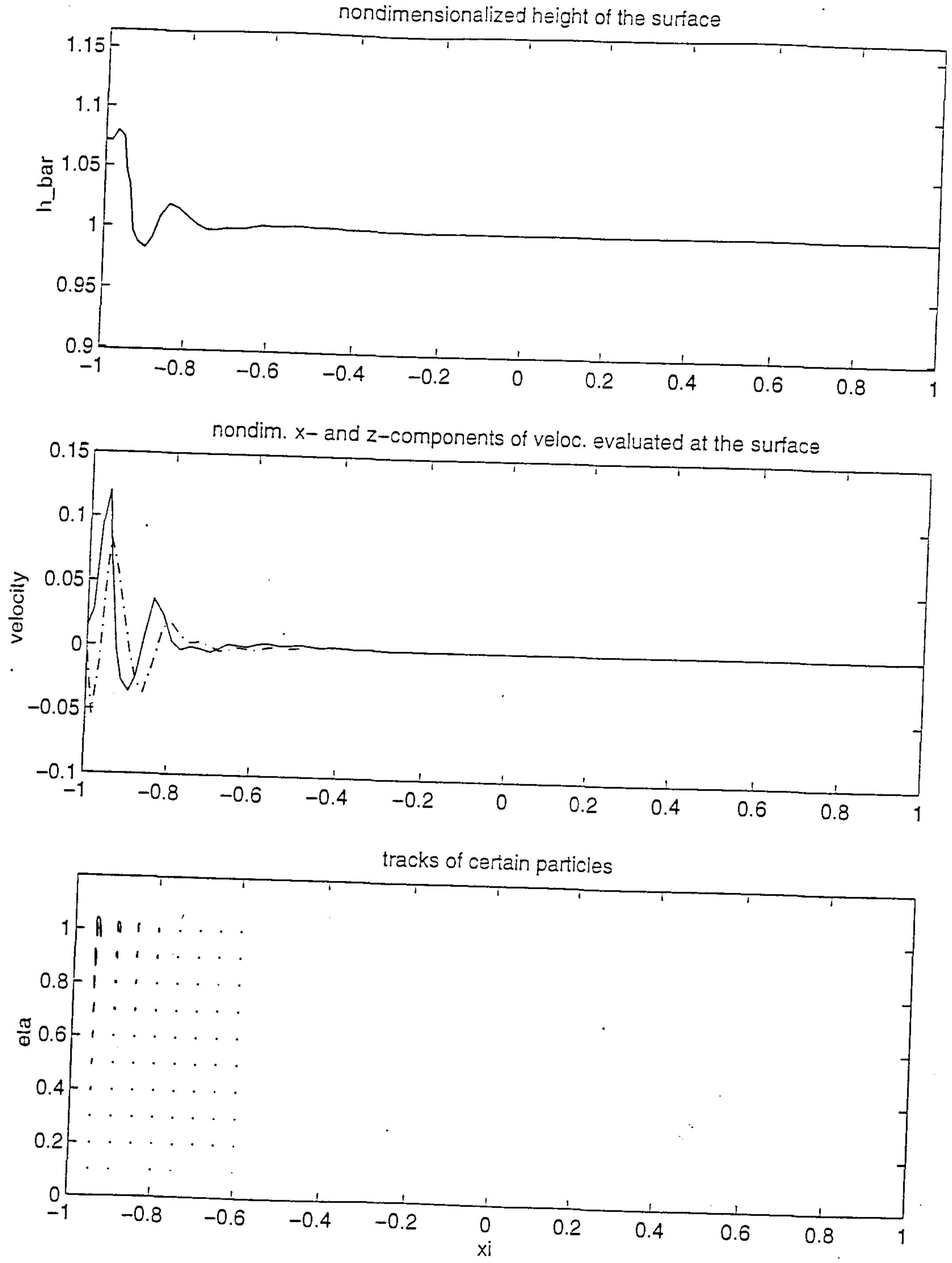


figure 6.5 cont.: Example 5, results after 300 timesteps

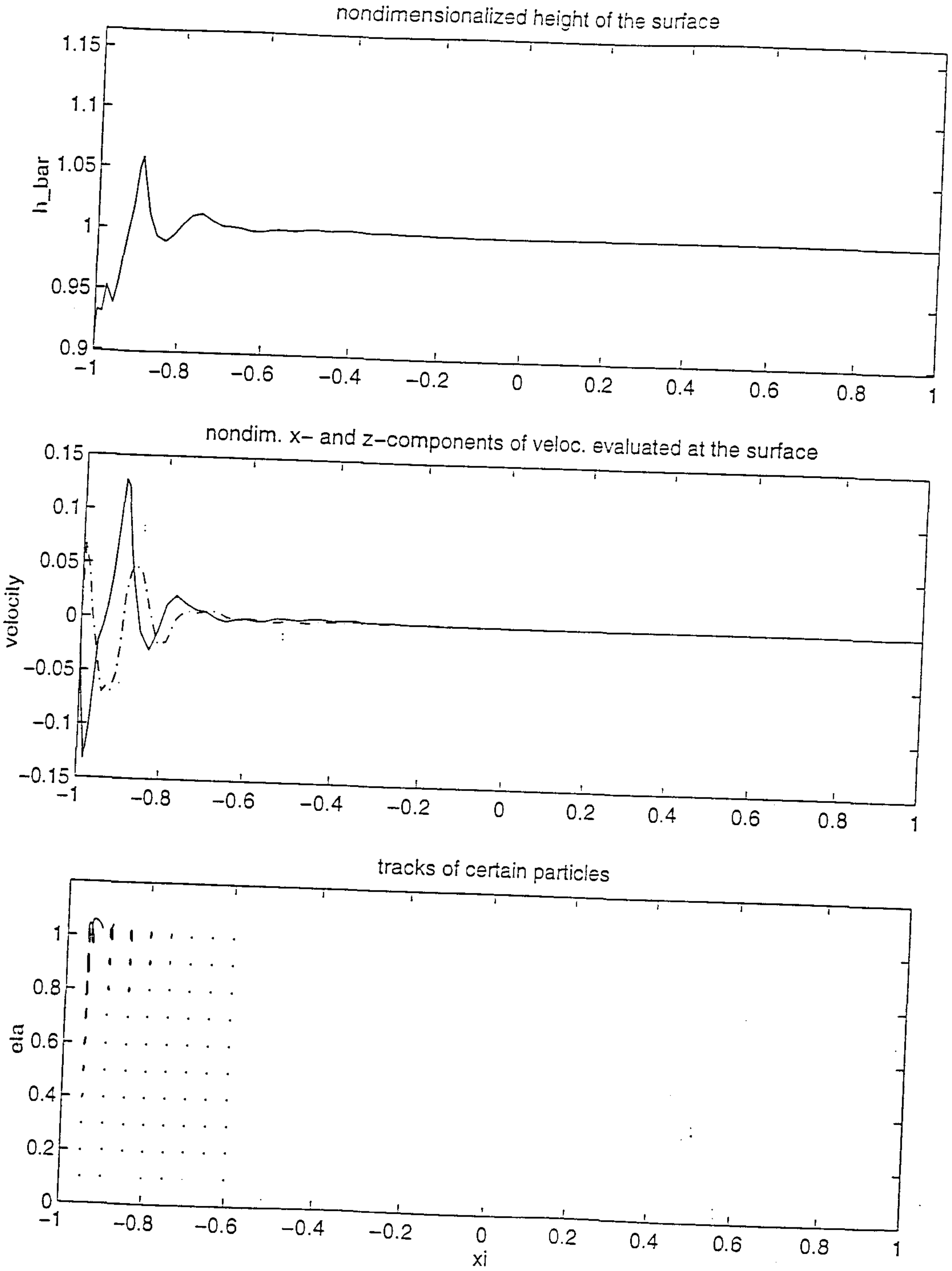


figure 6.5 cont.: Example 5, results after 350 timesteps

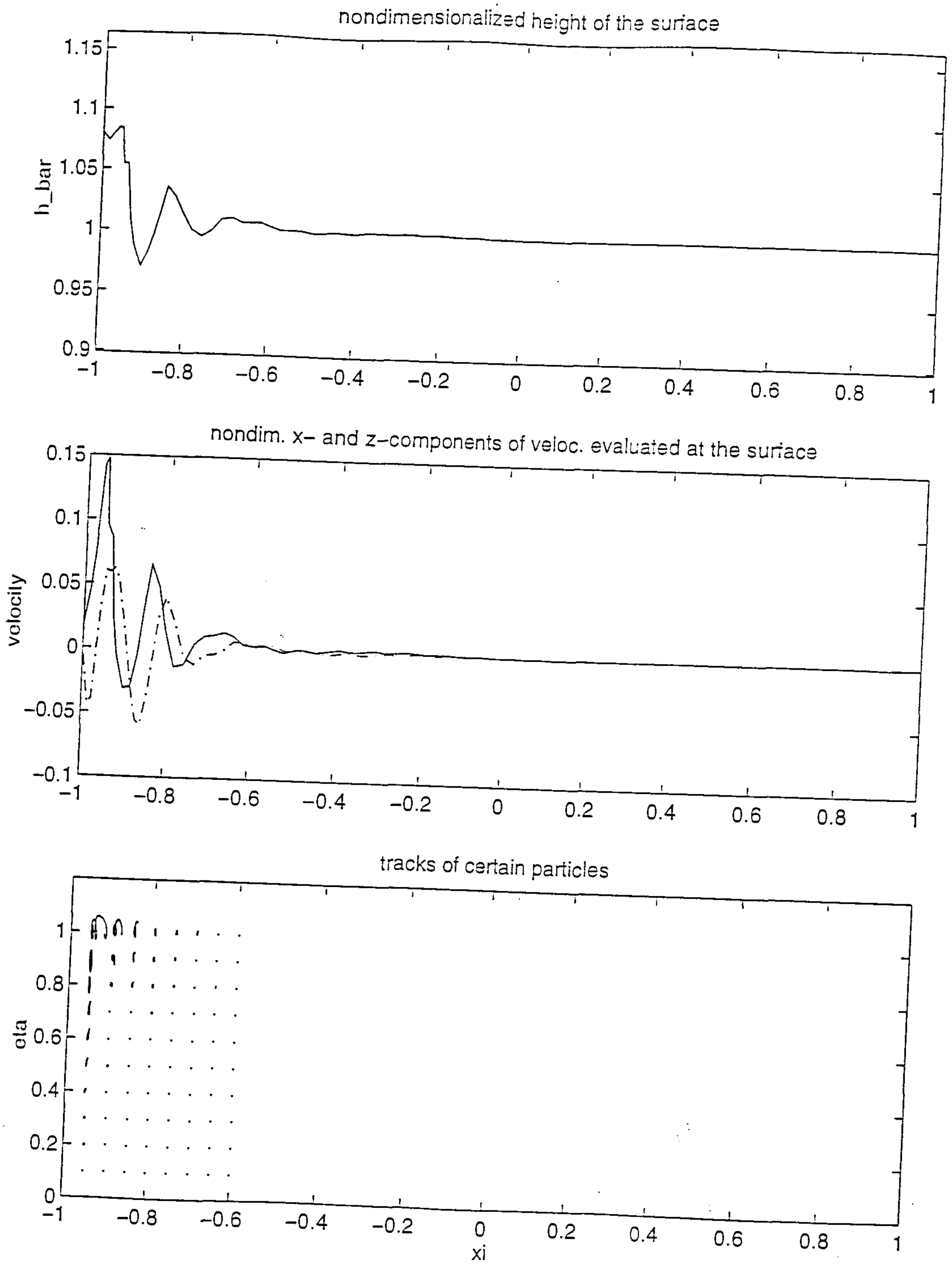
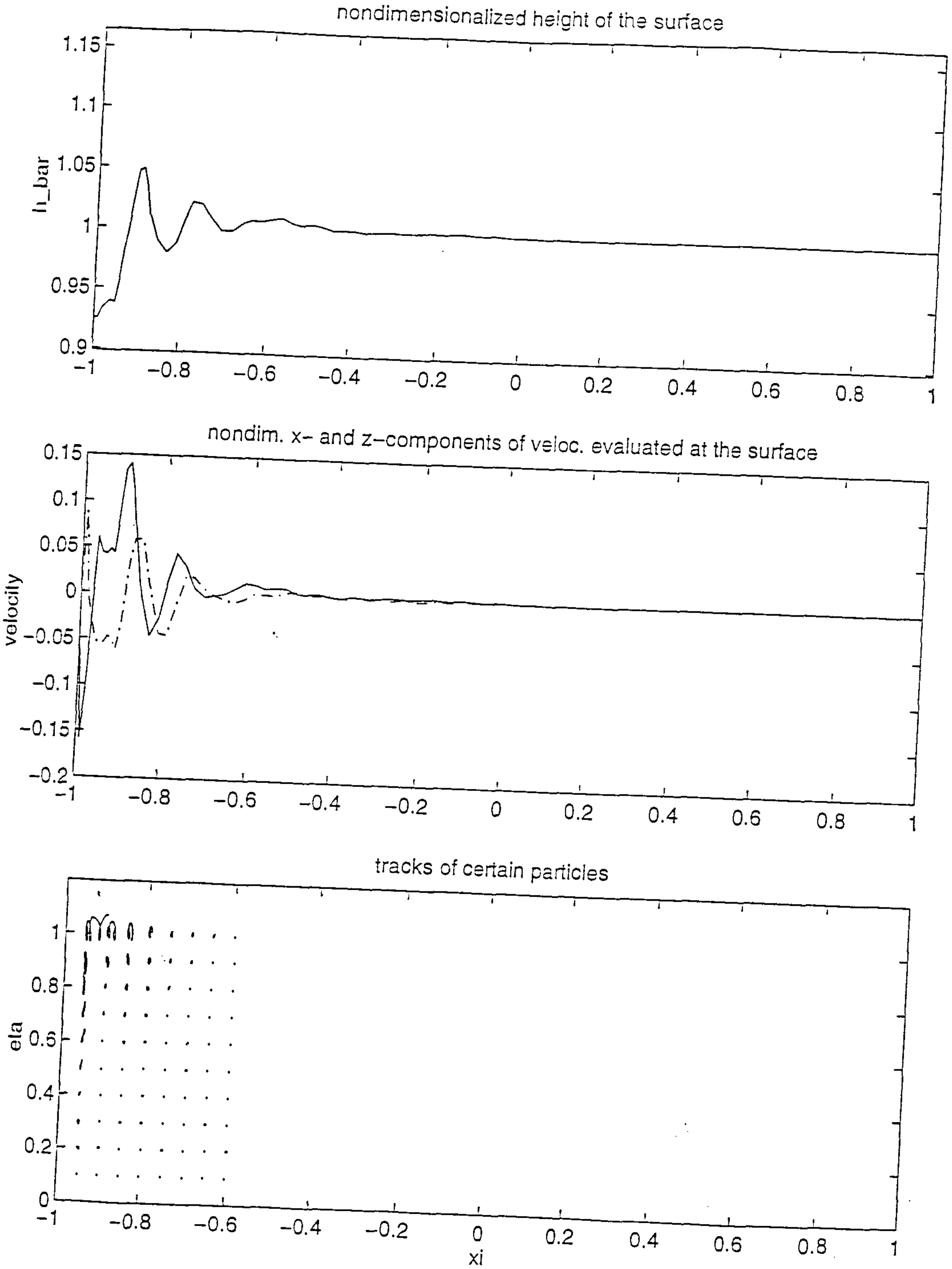


figure 6.5 cont.: Example 5, results after 400 timesteps

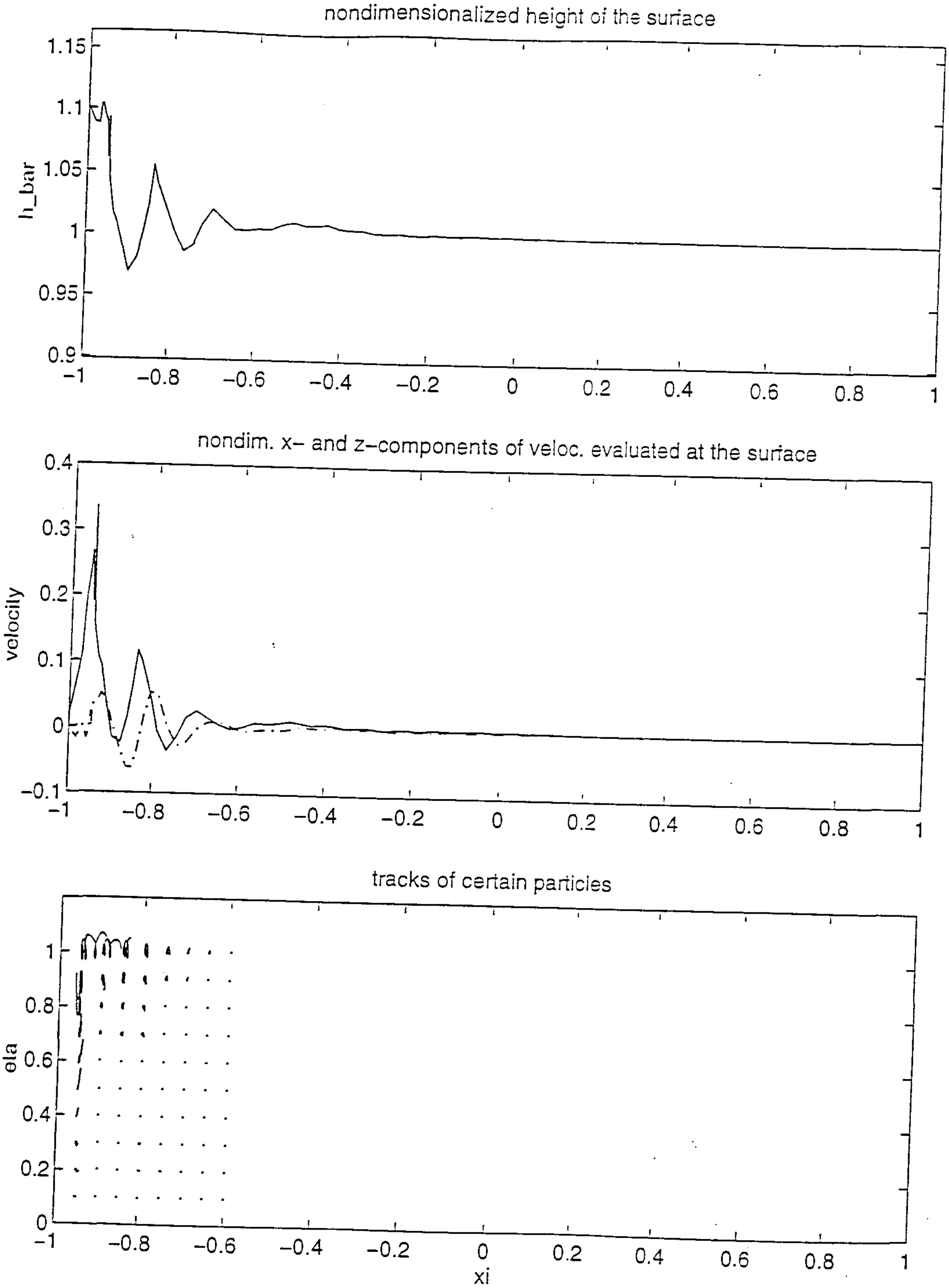


nondimensionalized height of the surface

nondim. x- and z-components of veloc. evaluated at the surface

tracks of certain particles

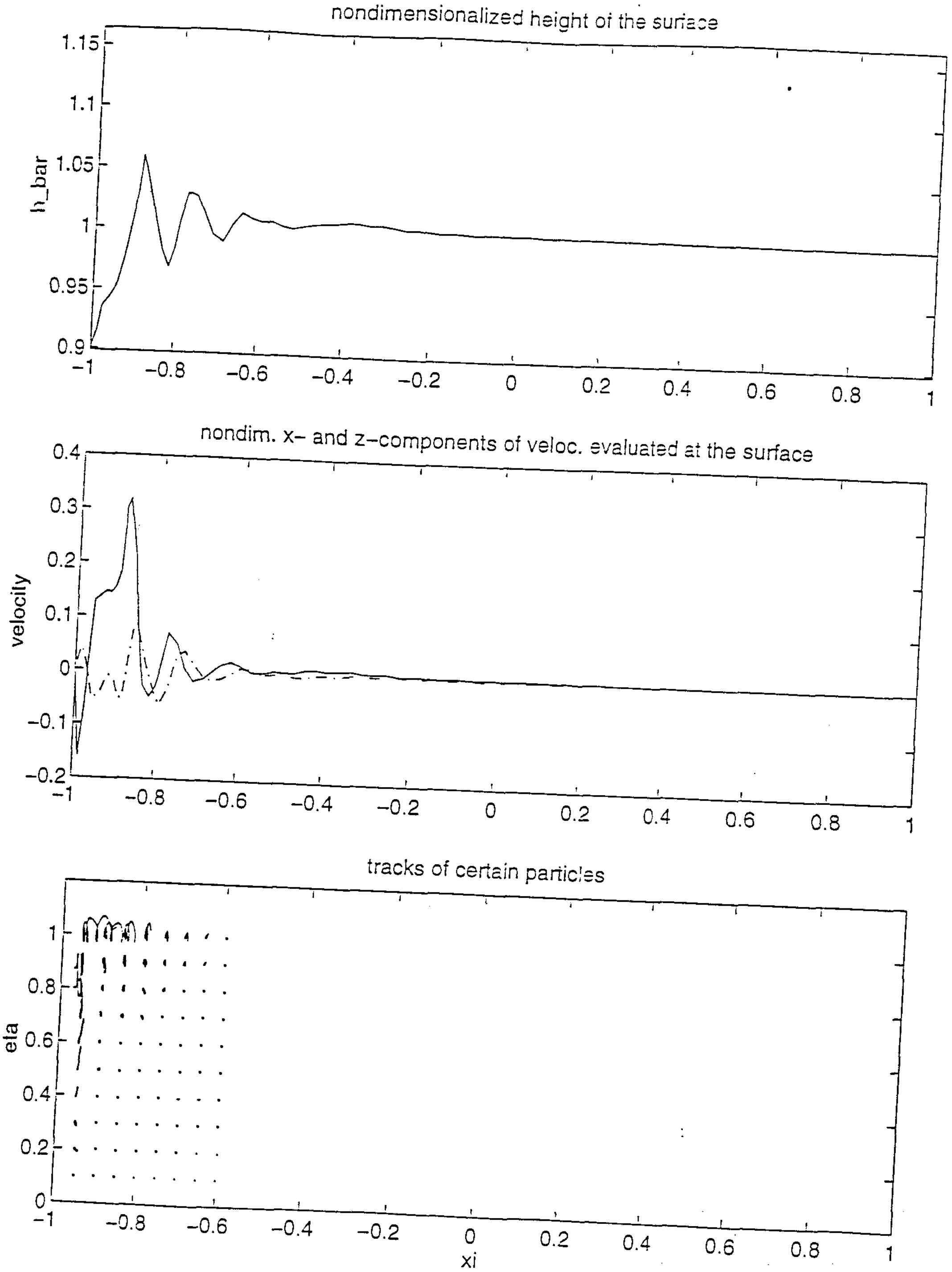
figure 6.5 cont.: Example 5, results after 450 timesteps


nondimensionalized height of the surface

nondim. x– and z–components of veloc. evaluated at the surface

tracks of certain particles

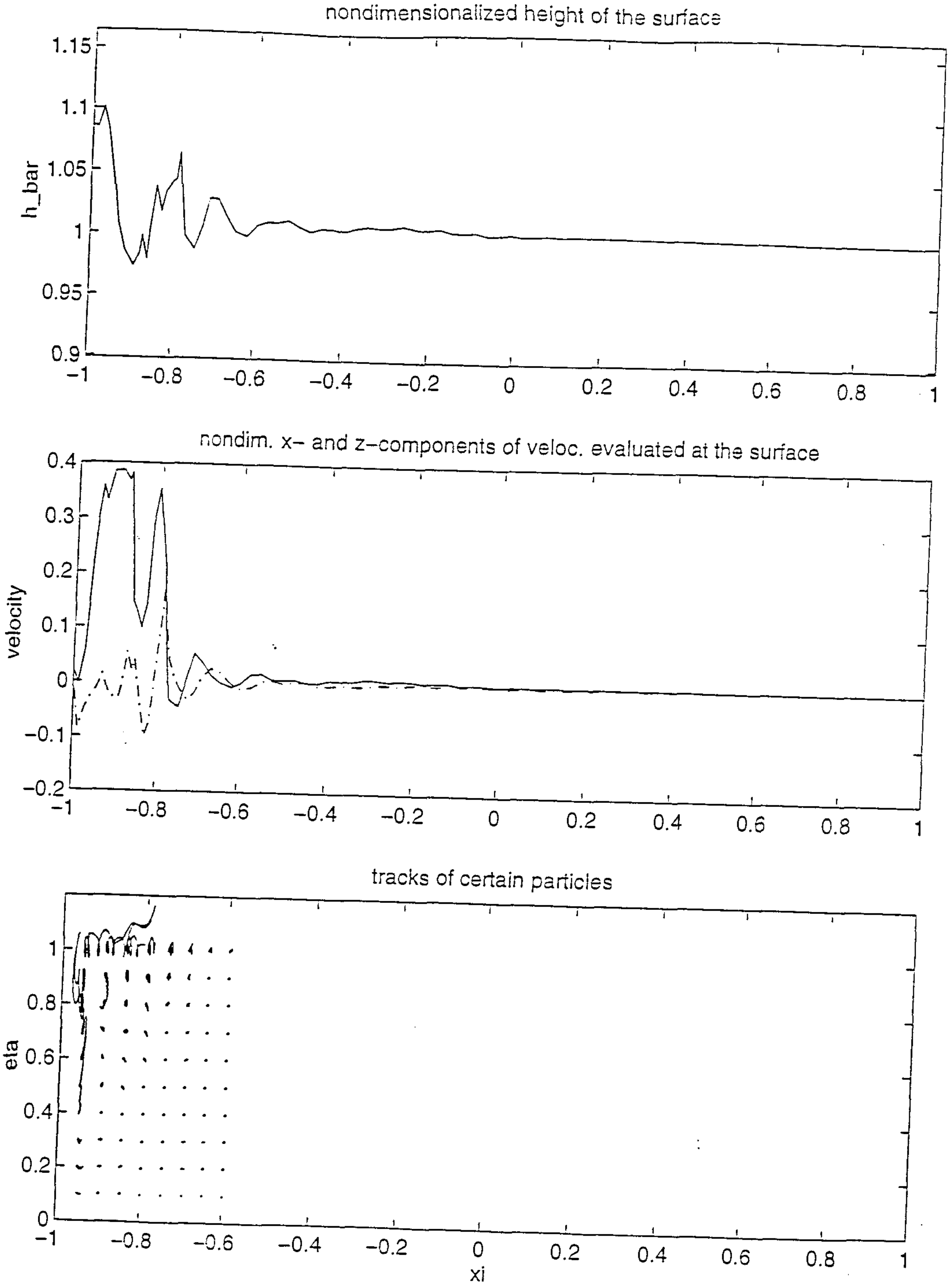
figure 6.5 cont.: Example 5, results after 500 timesteps


nondimensionalized height of the surface

nondim. x- and z-components of veloc. evaluated at the surface

tracks of certain particles

figure 6.5 cont.: Example 5, results after 550 timesteps



nondimensionalized height of the surface

nondim. x– and z–components of veloc. evaluated at the surface

tracks of certain particles

figure 6.5 cont.: Example 5, results after 600 timesteps

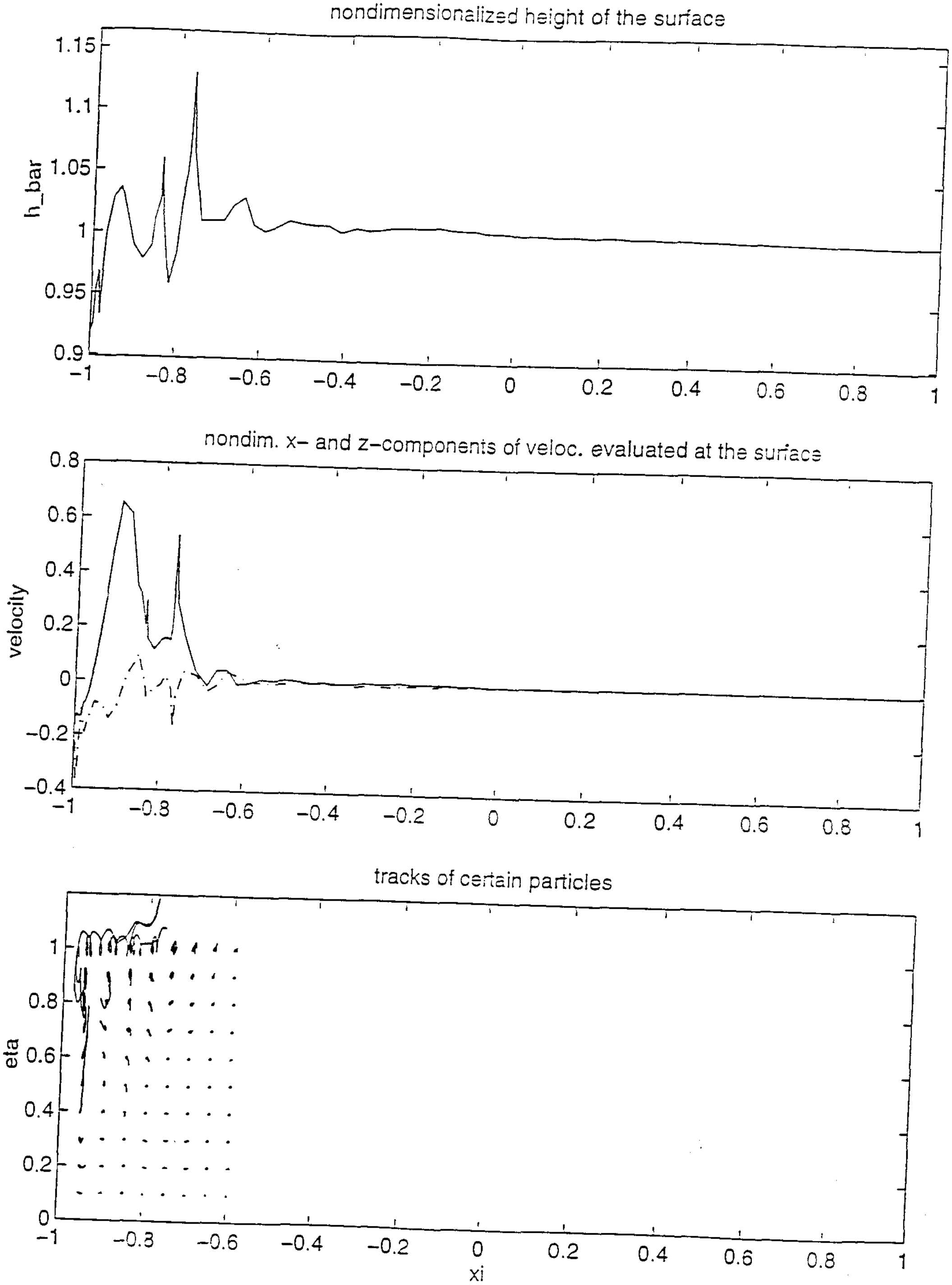


figure 6.5 cont.: Example 5, results after 650 timesteps

Chapter 7: Conclusions and Proposals for Future Research

We recognize that this thesis is the first step of a sequence of steps to be done in order to approach the final goal of the present research project.

We developed a mathematical model based on the well known physical model for incompressible free surface flow by solving the Navier-Stokes equations. We suggested a method to model the phenomenon of rising and falling water surface at a vertical boundary. By implementing the mathematical model as a computer program in MATLAB and by introducing different types of basis functions we showed experimentally that the model delivers reasonable, realistic results. Our computer program is very flexible. For further research, new kinds of basisfunctions can easily be added to the already existing program.

However, we did not make any attempt to show that the set of functions

$$B = \{\sum_{i=1}^{N} b_i \bar{e}^{(i)} \ , \ b_i \in \mathbf{R}, \ N = 1, 2, 3...\}$$

is dense in the set of functions

$$F = \{ \ f(x,y,z) \in C^3([-1,1] \times [-1,1] \times [0,H]), \mathbf{R}^3 \ , \ \nabla \cdot f = 0 \ \}.$$

Furthermore, we did not prove the consistency and stability of our time discrete schemes for the velocity and the surface. At this point we can already say that the stability of all iteration methods performed will depend on the time step size Δt.

In our experiments, we chose a conveniently small step size such that all iterations converged. It would be highly informative to give estimates for choosing Δt such that the iterations will converge.

Another point is to search for new types of basis functions, especially, basis functions that lead to small matrix condition numbers for the matrix $\mathbf{L}^{(i,k)}$ and that reduce the computational cost. One idea would be the use of an orthonormal basis. This would simplify certain integrals which would result in increasing computational speed.

Another very important step consists in the extension of the numerical algorithms exhibited in this thesis to the third dimension. This would allow us to study very complex phenomena, for example the occurence of crosswaves close to the wave board.

Bibliography

1. A. Baker, Finite element computational fluid mechanics. Washington: Hemisphere Pub. Corp. (1983)

2. M. Breuer, D Hänel, A dual time-stepping method for 3-D, viscous, incompressible vortex flows, Computers and Fluids Vol.22 No. 4/5 1993. Oxford: Pergamon Press Ltd.

3. A. Dold, B. Eckman, Approximation Methods for Navier-Stokes Problems. Berlin: Springer-Verlag (1980)

4. M.J. Fritts, E.W. Miner, O.M. Griffin, Numerical Calculation of Wave-Stucture Interactions, Computer methods in fluids. Plymouth: Pentech Press (1980)

5. W. Heinrichs, Efficient Iterative Solution of Spectral Systems for the Navier-Stokes Equations, 1.Aufl., Berlin: Verlag Dr. Koester (1992)

6. G. Jensen, H. Suedling, Ship wave-resistance computations, Notes on numerical fluid mechanics, vol.25, ed. E.H. Hirschel, Finite approximations in fluid mechanics II, Vieweg Verlag, 1989

7. N. Kanenko, T. Kodama, M. Kawahara, Estimation of eddy viscosity coefficients in a periodic shallow water equation, Computer methods in

applied mechanics and engineering 112 (1994).
Amsterdam: North-Holland Pub. Co.

8. P. Martinez, J. Harbaugh, Simulating Nearshore Environments. in: Computer Methods in the Geosciences, New York: Pergamon Press (1986)

9. J. Proux, T. Huges, A boundary integral modification of the Galerkin least squares formulation for the Stokes problem, Computer methods in applied mechanics and engineering 113 (1994). Amsterdam: North-Holland Pub. Co.

10. H.R. Schwarz, Methode der Finiten Elemente, 2. Überarbeitete und erweiterte Auflage, Teubner Studienbuecher, Stuttgart: B.G. Teubner (1984)

11. J. Serrin, Mathematical Principles of Classical Fluid Mechanics. Handbuch der Physik, hrsg. von S. Flügge, Berlin: Springer-Verlag (1959)

12. SWAMP Group, Ocean Wave Modelling, New York: Plenum Press (1985)

13. E. Truckenbrodt, Strömungsmechanik. Grundlagen und technische Anwendungen. Berlin: Springer-Verlag (1968)

14. F. Ursell, Ship Hydrodynamics, Water Waves and asymtotics: collected papers. Advanced series on fluid mechanics, Singapore: World scientific (1994)