The response of a turbulent boundary layer to a step change in surface roughness

Schols, J.L.j.

Award date:
1979

University of Eindhoven
Department of Physics
Heat Transfer Section

The response of a turbulent boundary layer to a step change in surface roughness

Jacq Schols R-386-A

Tutors: Drs. A.M. Koppius
Prof. Dr. K. Krishna Prasad

June, 1979

The response of a turbulent boundary layer to a step change in surface roughness'

Errata

page	position	correction, or addition
6	lines 13,14	',and beyond in' → 'and above'
8	line 22	'mention' → 'mentioned'
9	line 1	'whose slope' → 'the slope of which'
9	line 4	'values' → 'values of δ_i'
11	line 1	'o servations' → 'observations'
11	line 17	'has' → 'have'
12	line 11	'far)' → 'far)†'
12	below	'† note: except for Bradley, p. 11'
47	below	'number j' → 'symbol j'
78	line 27	'and, and' → ',and'
89	line 9	'have' → 'have almost'
94	abscissa of fig 4.25	50.83333 has to be subtracted from the numbers along the X/DELTA0_axis
105	abscissa of fig 4.30	27 has to be subtracted from the numbers along the X/DELTA0-axis
114	line 8	'subscripts, constants' → subscript (i), number (j), von Kármán constant (k), mixing length (l)

Errata (cont'd)

115	line 4	'kinematic' → '(kinematic)'
115	line 9	' , upstream value' (must be added to the text)'
117	above line 10	'the development of surface-shear stress downstream from a roughness-change
118	line 17	'4.14' and '72' must be left out
119	line 2	'4.14' and '72' must be added
B.10	last line in table B.1	33.65 → 33.55

Jacq Schols

TECHNISCHE HOGESCHOOL EINDHOVEN
Afdeling der Technische Natuurkunde
Vakgroep Transportfysica

1. Titel: The response of a turbulent boundary layer to a step change in surface roughness

2. Auteur: Jacq L.J. Schols

3. Datum: May 1979

4. Docent/contactpersoon: prof.dr. D.A. de Vries

5. Begeleiders: prof.dr. K.Krishna Prasad
drs. A.M. Koppius

6. Stageverslag no.:

7. Afstudeerverslag no.: R-386-A

8. Intern Rapport no.:

9. Korte samenvatting:

The calculation of the response of a turbulent boundary layer to a step change in surface roughness in determining transport processes is of vital importance in meteorology, and it will also lead to the development of a more comprehensible theory of turbulent shear flows.
During this investigation a prediction method developed by Bradshaw,Ferriss and Atwell(1967) is used to carry out the calculations. The turbulent shear stress appears to be the most difficult quantity to calculate,because it is much more sensitive to changes in boundary-conditions than the longitudinal component of mean velocity.
The obtained results are compared with data of wind-tunnel experiments.

Acknowledgements.

The author is indebted to prof. dr. K. Krishna Prasad and drs. A.M. Koppius, who lent support to the completion of this report.

Thanks are also due to all, who rendered valuable assistance to the development of the computer program.

Contents.

Abstract

In this report the response of a turbulent boundary layer to a step change in surface roughness has been calculated. Such a study is of vital importance in meteorology, in determining transport processes, and it will also lead to the development of a more comprehensible theory of turbulent shear flows. Therefore, a computer program has been developed, in which a prediction method developed by Bradshaw, Ferriss and Atwell (1967) is used. In this method the turbulent energy equation is converted into a transport equation for the Reynolds shear stress, which quantity seems to be the hardest to predict, since it is much more sensitive to changes in boundary conditions than the mean velocity.

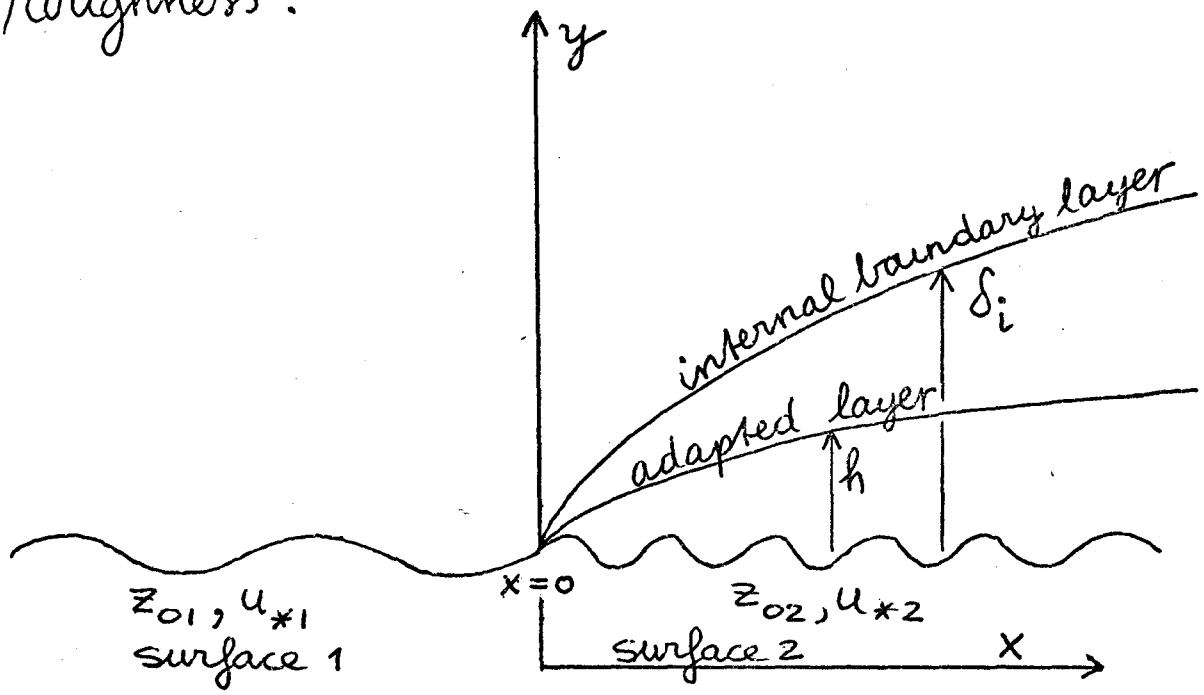
The obtained results are compared with data of wind-tunnel experiments.

1. Introduction

1.1 General description of the problem.

The aim of this work is to model a turbulent boundary layer undergoing a step change in surface roughness. The problem has applications in meteorology in determining transport processes. Moreover, such a study may lead to the development of a more comprehensive theory of turbulent shear flows. The problem has attracted considerable attention in the past twenty years (a brief review of the work done so far is presented in section 1.2). The work described in this report is based on the theory of Bradshaw and Ferriss (1967), which is claimed to predict the behaviour of a turbulent boundary layer for simple types of perturbations such as a sudden change of pressure gradient, wall roughness, or injection or suction of fluid. A computer program has been developed, in which the development of a turbulent boundary layer is calculated. Before carrying out the computations on the response of a turbulent boundary layer, which undergoes

a step-change in surface roughness, the program has been tested on its predicting capabilities.
Fig 1.1 shows the development of a turbulent boundary layer after a step-change in surface roughness.



⊗ Fig. 1.1 Schematic presentation of the changes which occur inside a turbulent boundary layer which negotiates a step change in surface roughness.

Just after the transition there grows what is called an internal boundary layer with a height δ_i. It is presumed that all changes in the character of the boundary layer are limited to this internal layer, at least for a

⊗ Note: surface roughness and surface-friction velocity are denoted by z_0 and u_* respectively.

considerable distance downstream from the step. The flow above the internal boundary layer is simply the normal development of the upstream flow before the step, assuming that displacement effects due to the internal boundary layer are small. Inside the internal layer a second layer is formed in which the flow has fully adapted to the new surface-conditions. Here, the flow is said to be in energy-equilibrium, just as it was in the region near the surface before passing the roughness-change. This means that (Townsend, 1961)

(i) the advection of turbulent energy is small

(ii) the dissipation length L_ϵ, defined as ⊗ $L_\epsilon = \tau^{3/2}/\epsilon$, is directly proportional to y, the relevant length scale in the inner part of the boundary layer.

The latter condition implies that the mixing length ⊗ $l \equiv \sqrt{\tau}/(\partial U/\partial y)$ is also directly proportional to y, in the inner part of the boundary layer, because the production of turbulent energy is very nearly equal to the dissipation ($\tau\, \partial U/\partial y \simeq \epsilon$) over most of the internal layer, making L_ϵ very nearly equal to l.

⊗ Note: τ stands for the kinematic turbulent shear stress, and ϵ denotes the dissipation by viscous forces. U denotes the streamwise component of mean velocity.

The investigation of the streamwise variation of the surface shear stress (τ_w) – the salient parameter involved in turbulent mass and heat transport – is of particular importance. This quantity is strongly related to the growth of the internal boundary layer, which has been subject to many investigations during the last few decades. The next section provides a short review of the work done on the subject until to day. In section 1.3 a summary of the work in this report is presented.

1.2 Survey of literature.

Considering the investigations of the last twenty years, we can divide them into three main groups.

(i) Early theoretical work (e.g. Elliott (1958), Panofsky and Townsend (1964), Townsend (1965), and Blom and Wartena (1969))

(ii) Experimental work. A review on a whole class of relaxing turbulent boundary layers was provided by Tani (1968) Additional experiments since then are those due to Bradley (1968) in the lower part of the atmosphere, and Antonia and Luxton (1971, 1972) in the wind-tunnel.

(iii) Recent theoretical work (e.g. Peterson (1969), Shir (1972), Rao et al. (1974), and Wood (1978)) in which numerical methods and more advanced closure hypotheses are used.

These investigations will now be discussed briefly.

1.2.1 Early theories

These theories assume that the total boundary-layer thickness, δ, of the equilibrium layer which has formed over a uniformly rough surface, before the surface-change, is much larger than any other height considered, like the height, δ_i, of the internal boundary layer. The latter quantity is stated to be again much larger than the height of the roughness elements on both sides of the step.
These conditions make it justifiable to use simple forms for the normal velocity distributions in the equilibrium part of the boundary layer near the surface, and beyond in the internal boundary layer.
A common assumption made, is

$$U = \frac{u_*}{k} \ln \frac{y}{z_0} \qquad (1.1)$$

where, $u_* = \sqrt{\tau_w}$ ⊗, is the surface-friction velocity, and z_0 the roughness height.

In the equilibrium part, the shear stress is said to be independent of height and therefore equal to the surface shear stress.

⊗ Note: τ_w denotes the kinematic wall shear stress.

Elliott (1958) assumes that after the jump in surface roughness the flow adjusts itself to the new surface-condition inside the whole internal layer region and therefore is in an equilibrium state there. A consequence of this theory is that at the interface between the internal flow and the outer flow of the boundary layer there occurs a discontinuity in shear stress, which causes a kink (at $z = \delta_i$) in the calculated velocity profiles. Panofsky and Townsend removed this shortcoming by assuming that there exists a linear distribution of shear stress inside the internal boundary layer. At its upper edge, whose position is now indicated by the top of a transition zone in the semi-logarithmic plot of the mean longitudinal velocity-profile, the shear stress reaches the value of the unperturbed flow.

Townsend (1965) showed that, under the assumption that $\delta \gg \delta_i \gg z_0$, the changes of the flow quantities after a change in surface configuration are self-preserving in form. This means that rates of change of flow properties can be predicted with no more specific assumptions about the nature of turbulent motion than that the large-scale

motion is independent of the fluid viscosity. In his refined theory he assumes self-preserving forms for the shear stress distributions and velocity changes due to flow accelerations.
It was in fact Blom and Wartena (1965) who proved, according to the theory of Townsend, that it is not possible to indicate a layer satisfying the inner boundary condition (1.1) after the roughness-change. Through a modification of the theory, they showed that a domain satisfying the inner boundary-condition does exist. They also treated the case of two subsequent abrupt changes of surface roughness, and found that the adaptation of the boundary layer downwind of the second step is of the same order of magnitude as the adaptation after the first change in roughness height. The velocity profiles calculated according to the theories of the above mention authors agree reasonably well; the calculated values of the surface shear stress differ widely (see fig. 1.3). Also the heights h of the adapted layer seem to differ considerably. Fig 1.2 illustrates this clearly: Elliott's value of h coincides with the edge δ_i of the internal boundary layer,

the so-called interface whose slope is about equal to 1:10. This value was also found by Panofsky and Townsend, although their values are higher than those of Elliott. Blom and Wartena showed that $h \simeq l/10$, where l is of the same order of magnitude as the height of the critical surface, which can be interpreted as a measure of the internal boundary-layer height

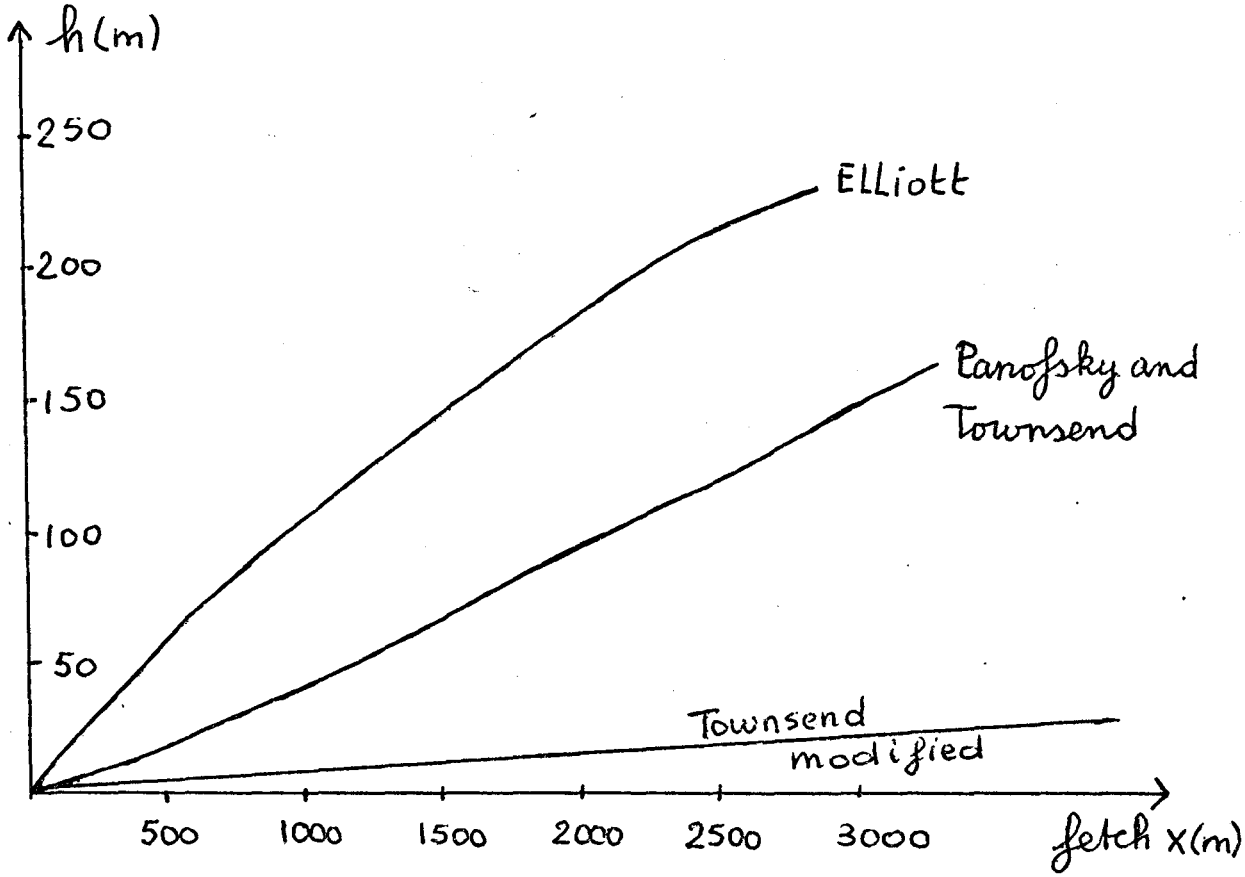


Fig. 1.2 Growth of the adapted layer after a step change in surface roughness (after Blom and Wartena, 1969)

1.2.2 Experimental investigations.

Tani (1968) presents a review of some experimental work concerning investigations on the behaviour of relaxing equilibrium boundary layers. From the results obtained in pipes and wind-tunnels some important deductions were drawn. These are summarized below.

1. The response of a turbulent boundary layer is almost instantaneous in the inner part near the wall, but rather slow in the outer part, as demonstrated by the slow adjustment of the profiles of turbulent intensity and turbulent shear stress, with a relaxation-distance of scores of boundary-layer thicknesses.
2. The mean velocity profile and the non-dimensional parameters such as H and c_F ⊗ are inadequate to indicate whether the boundary layer has attained a new equilibrium state.
3. The streamwise turbulent intensity profile and the shear stress profile appear to be nearly similar, except near the wall, during the relaxation-process towards equilibrium.

⊗ Note: H, stands for the shape-factor, which is equal to the quotient of the displacement thickness and the momentum-deficit thickness.
c_F, denotes the surface-friction coefficient.

Beyond these general observations, however, Tani felt it was difficult to deduce any universal rules governing the boundary-layer flow in the relaxation-process.
Since the publication of Tani's review, two important experimental investigations (Bradley (1968) in the lower part of the atmosphere, and Antonia and Luxton (1971, 1972) in the wind-tunnel) have appeared. The main conclusions drawn from them will be discussed briefly.

1.2.2.1 Bradley's experiments.

Velocity profiles and surface shear stresses were measured for both smooth to rough (here and in the following indicated by s→r) and rough to smooth (denoted by r→s) transitions. No measurements of the normal shear stress distribution has been done because it was very difficult and costly to do it with the equipment that was available at that time. His results were that for a r→s transition the surface shear stress needs a longer distance to adjust than in the reverse case. For both changes the growth of the internal boundary layer obeys the .8 power law of boundary-layer growth along a flat plate under a zero pressure gradient.

1.2.2.2 The investigations of Antonia and Luxton.

The laboratory experiments, in which relatively large roughness heights were employed, show ⊗ that there occurs a difference between the internal boundary-layer growth for a s→r transition ($\delta_i \sim x^{.79}$) and a r→s change ($\delta_i \sim x^{.43}$). In the latter case the surface shear stress needs a longer distance to relax. (Neither a difference in the internal-layer-growth nor a difference in relaxation-distances for both transitions are shown by the investigators mentioned so far) The internal-layer flow near the wall is not in energy equilibrium after the step, so the concept of inner layer similarity breaks down. This has been deduced from mixing length results and an analysis of the turbulent energy equation. Only at a distance well downstream from the roughness change the flow reaches a new self-preserving, equilibrium state.
It is obvious from these experiments that theories based on simple assumptions between the shear stress-distribution and the mean velocity field, such as

⊗ Note: This means that the assumption used by Townsend is no longer valid.

mixing length hypotheses, no longer apply. Modern work using numerical approaches and higher order closure hypotheses may better predict the laboratory findings

1.2.3. Recent theories.

These theories start - as the older theories -, from the mean momentum and the mean continuity equation. One, or maybe more equations, depending on the used model, are added to close the set of governing equations. The closed system of equations for the horizontal and vertical components of mean velocity, and the horizontal shear stress has to be solved numerically. Unlike in previous models there are no a priori assumptions about the distribution of mean velocity or shear stress, and mixing length in the transition region where the flow is not in energy equilibrium.

Peterson used a model based on the hypothesis that the horizontal shear stress is proportional to the turbulent energy. The closure equation is formed by the turbulent energy equation, where a gradient diffusion term and a suitable assumption about the form of the dissipation term are used. This theory

is the first that predicts distributions of turbulent energy. The growth of the internal-layer height, here defined as the point where the shear stress reaches .1 per cent of the upstream-value, agrees with the .8-power law predicted by previous theories and observations. Only in the lower 10 per cent of the internal boundary layer is the flow essentially in equilibrium with the downstream roughness. The predicted transition velocity profiles differ most clearly from those of previous models in that they show an inflexion point.

Shir, who also uses the turbulent energy equation, treats both the mean horizontal and vertical momentum equations through the vorticity equation. His predictions confirm most of Peterson's findings. He points out that the definition of the internal boundary layer by various authors is inconsistent. Some define the boundary according to the velocity profile, while others use the shear stress profile. In his study, two boundary layers are defined, viz., velocity boundary layer δ_i and stress boundary layer δ_τ, where $\delta_\tau \simeq 2\,\delta_i$. This means that the normalized velocity-change profile is not as self-

preserving as the ⊗ normalized stress-change profile (also see Peterson, and Rao et al.)

Rao et al. use a higher-order turbulence closure theory that includes dynamical equations for the Reynolds stresses and viscous dissipation rate. The distributions of the non-dimensional wind shear, defined as $\phi = \frac{ky}{\sqrt{\tau}} \frac{\partial u}{\partial y}$, the dissipation and mixing-length scales, and the ratio of shear stress to turbulent kinetic energy, are shown to differ significantly from their equilibrium distributions in the transition layer, after the surface change.

Rao et al., just a Shir, predict that the flow is in equilibrium with the new downstream roughness only in the lower 10 % of the stress layer, for the $s \to r$ transition, and in the lower 5 % of this layer for the $r \to s$ change. The growth of the

⊗ Note: The normalized velocity-change is defined as the difference between the value of the velocity at some point in the boundary layer and its value at a corresponding position upstream, divided by an appropriate scaling quantity: $\frac{.4\ (u-u')}{u_* \ln(z_o'/z_o)}$. A similar definition holds for the normalized stress-change: $\frac{\tau-\tau'}{\tau_w-\tau'}$. The primed quantities indicate upstream values.

stress layer shows the $\frac{4}{5}$ - power law variation in both cases.
Wood applies the basic Bradshaw - Ferriss - Atwell (1967) method to calculate flow development in a neutral atmosphere after a large step change in surface roughness. He investigated the difference in results obtained using an algebraic expression and a transport equation for the length scale. The latter equation allows for the non - equilibrium form of the dissipation term immediately following the step. It is argued that the approximation $L_\epsilon = ky$, where k is von Kármán's constant, appears reasonable also in the non - equilibrium part of a neutral atmospheric boundary layer, provided only the mean velocity profile and surface - shear stress are required.

1.2.4. A critique on the investigations.

So far, the theoretical results have been compared with few adequate observations. In meteorology, conditions of neutrality and uniform terrain are hard to find. On the contrary, in laboratories these conditions can be achieved easily ; however, scaling problems are severe - even a very large

boundary-layer tunnel may not completely model the properties of atmospheric wind. To justify refinement of the prediction theories there is a need to compare present methods with wind-tunnel experiments. The problem in the initial instance can be restricted to a turbulent boundary layer under zero pressure gradient encountering a step change in surface roughness.

In this section, the comparison between the several theories and the experimental works is considered. Definitive comparisons have only been made with respect to Bradley's work. In general, it can be stated that the earlier theories perform poorly when comparisons are made with wind-tunnel studies.⊗ In a wind-tunnel the internal boundary layer occupies a significant fraction of the total boundary-layer thickness. So the assumption that the internal layer develops inside the 'constant-stress layer' (which extends to about 15% of the lower part of the boundary layer) is no longer valid. The newer theories ought to apply to a large

⊗ Note: See the paper of Antonia and Luxton - (1971, part I, p. 736, fig. 11) - for comparison of the height of the internal boundary layer as measured in the experiment with the theoretical results of Townsend and Elliott.

class of experiments, because they assume less about the physical nature of relaxing flows.

A common problem that occurs in prediction theories is their sensitivity to the values chosen for the roughness heights. In practice, the roughness length, especially the downstream one, is difficult to determine. Therefore it is hard to compare the models directly with observations. Fig. 1.3 presents the development of surface-shear stress downstream from a roughness-change. The differences between several predictions come out strongest here. Both transitions are shown, where $z_{os} = .002$ cm (roughness height on smooth surface) and $z_{or} = .25$ cm (roughness height on rough surface).
The solid curves present the data of Shir (S), Panofsky and Townsend (P&T), and Rao et al (R). For the latter, a second curve is shown in fig 1.3.b, where $z_{os} = .0002$ cm instead of .002 cm.
The other curves present the results of Elliott (- - - -) and Peterson (-.-.-.-.-). Bradley's data are also present (I). τ_r denotes the surface-shear stress on the rough surface, and τ_s that on the smooth surface.

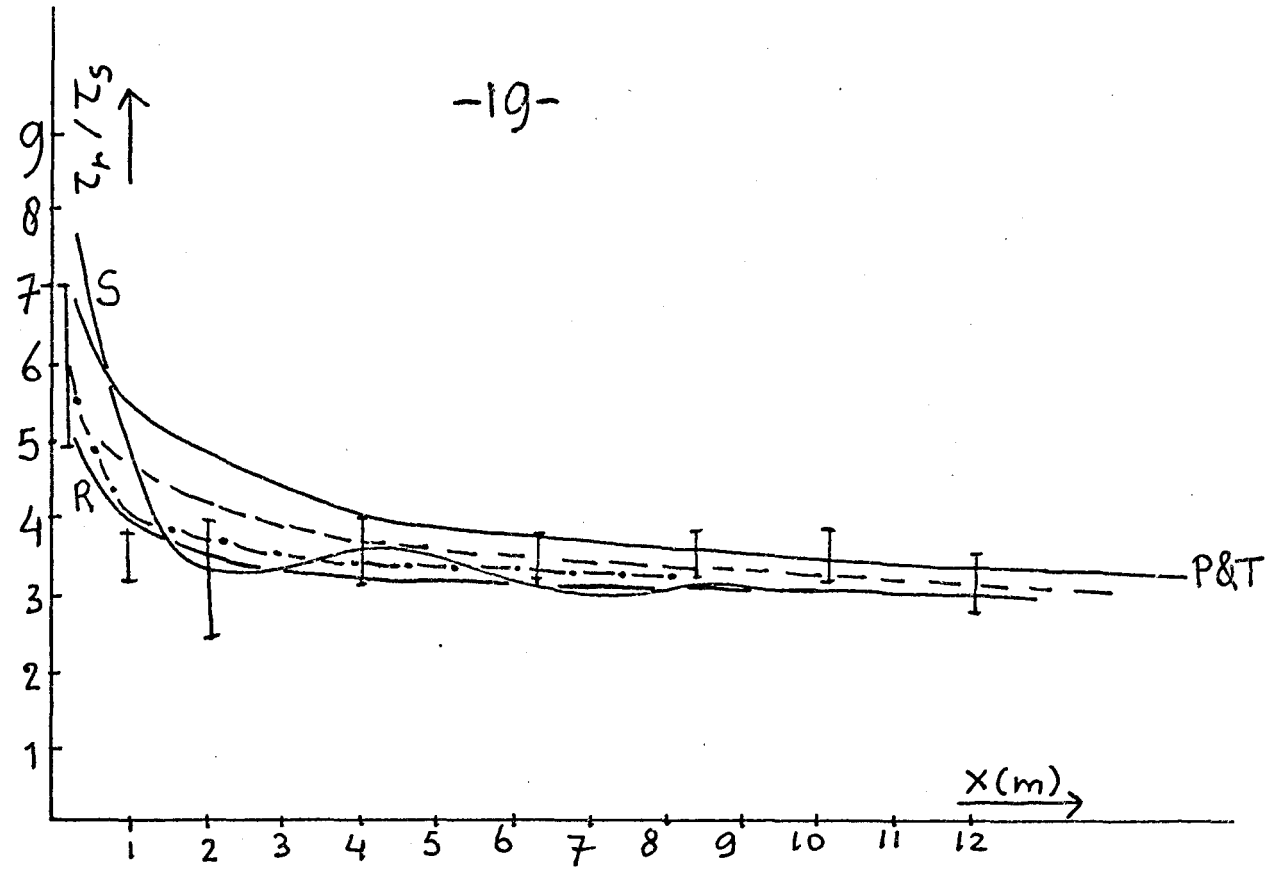


Fig. 1.3.a Smooth to rough transition

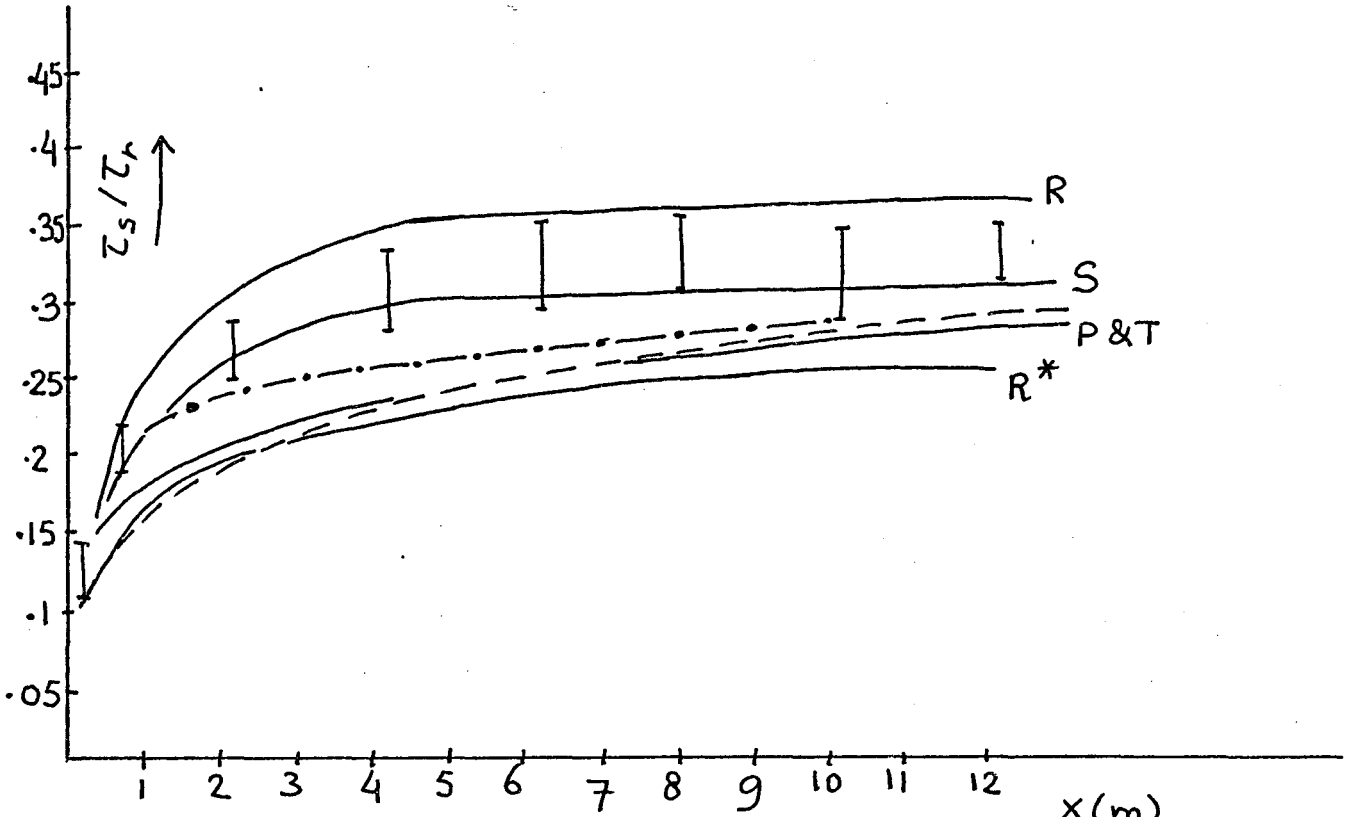


Fig. 1.3.b. Rough to smooth transition

(For a description see p. 18)

The results of Rao et al., in the lower figure on p. 19, for different values of z_{os}, reflect the influence of this parameter on the results. Further it is evident that for the r → s transition there is a considerable scatter among the several methods. Shir's computational results seem to fit best the experimental observations. This may be attributed to the fact that in his method he did not use the boundary layer approximations.

So far, the comparisons between experimental and theoretical investigations have not been enough, because calculations are often carried out for perturbations, or under conditions, which cannot be directly compared with the existing experimental results. The calculations to be described in this report are based on the turbulent energy model of Bradshaw and Ferriss (1967). In their theory they use empirical functions relating the turbulent intensity, diffusion and dissipation to the shear stress profile. The forms of the distributions of these functions are very simple, and based on equilibrium flow conditions (for a description see section 2.2). The question still remains whether these simple choices are valid in a boundary layer undergoing a sudden change in surface

roughness. The first of the assumed relations that the turbulent intensity is directly proportional to the shear stress seems to have been corroborated by the experimental results of e.g. Antonia and Luxton for both transitions in surface roughness. The dissipation-length, that represents the relation between the dissipation-term and the shear stress, stays approximately equal to the mixing length in the transition region after the step, but, they both appear to differ significantly from their equilibrium distributions. The results of the distribution of the third function, $G/(\tau_{max}/(\rho U_\infty^2))^{.5}$, where $G^{\otimes}$ represents the relation between diffusion and shear stress, computed from Antonia and Luxton's data are shown in fig. 1.4. The curves for fully developed smooth (s)-and rough (r) - surface boundary layers lie far beyond Bradshaw's curve$^{\otimes}$ which was derived from Klebanoff's (1955) zero pressure gradient boundary layer along a smooth plate. For values of y/δ between .2 and .7 the s and r curves lie fairly close to each other. In the remainding part part of the boundary layer the rough-wall curve is lower. Also curves for both s→r and r→s transitions are shown.

⊗ Note : see section 2.2 for details.

In both cases the curves differ significantly from their shapes in fully developed smooth- and rough- surface boundary layers. So, one can conclude that in the relaxation region, after a step - change in surface roughness, the distribution of the G- function shows a large variation, which is especially appearant inside the internal part of the boundary layer. ⊗

The question now remains whether the observed differences between the curves of Antonia and Luxton for fully developed smooth and rough wall flow, and Bradshaw's curve are the results of inaccurate measurements. It is obvious that, when detailed measurements of the turbulent energy balance and shear stress are available, which are appropriate for comparison purposes of prediction models, a good insight in the nature of the problem can be achieved.

⊗ Note: In fig 1.4 the fully developed smooth and rough - wall flow data are indicated by S- and R- curves respectively. The curves for the transitions are calculated at positions of about 48 roughness heights and about 130 roughness heights downstream from the respective S→R and R→S - step.

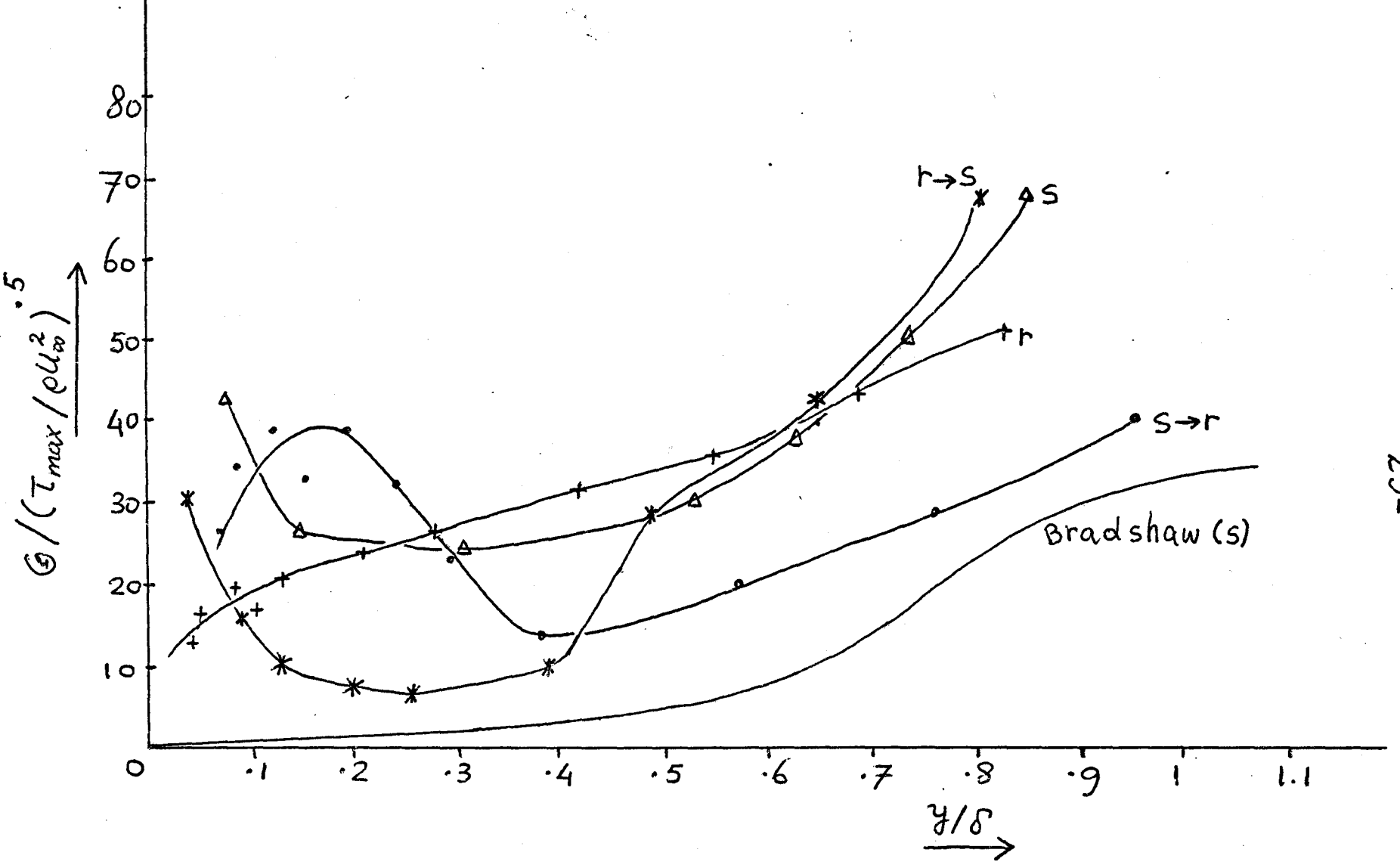


Fig 1.4 Distribution of $\mathcal{G}$-function for Antonia and Luxton's data. Bradshaw's curve for a smooth-wall flow is also shown.

1.3 The present work

The work in this report is based on the theory of Bradshaw and Ferriss (1967). These authors have applied their method of calculation (Bradshaw, Ferriss and Atwell, 1967) to predict the behaviour of a turbulent boundary layer for a simple type of perturbation such as a sudden change of pressure gradient, wall roughness, or injection or suction of fluid. The model is based on the turbulent energy equation, which is converted into a transport equation for the kinematic turbulent shear stress. Very simple assumptions about the relations between the shear stress profile and terms in the turbulent energy equation are made. It has been proved already that such assumptions, made under equilibrium flow conditions, lead to satisfactory prediction results of the behaviour of a turbulent boundary layer perturbed from its equilibrium state. It is interesting, however, to investigate the effects of changing such assumptions and compare the results with observations. It appears most desirable to compare prediction results with wind-tunnel experiments on simple flows, where conditions can easily be specified.

The work in this report is meant to support the efforts to find answers to these questions. The method is outlined in chapter 2. In Chapter 3 the numerical calculations, based on the theory of Bradshaw et al., are presented. The fourth chapter shows the achieved results, which have also been compared to experimental data of wind-tunnel flows. In section 4.2, as a test of the computer program, some equilibrium turbulent boundary layer flows⊗ have been predicted, and the results have been compared with those of Bradshaw+, and experimental data. Section 4.3 shows the calculation results of a step change in surface roughness, and a comparison with the experimental data of Antonia and Luxton (1971, 1972). Finally, in Chapter 5, some conclusions are drawn, and suggestions for further research are made.

⊗ Note: These flows concerned turbulent boundary layers, along a smooth surface, under a pressure gradient.

+ Note: Bradshaw (1967) also presents some calculation results of these boundary layer flows.

2. The calculation procedure.

2.1 Introduction.

We will consider a stationary, incompressible, two-dimensional boundary layer under the influence of a given arbitrary pressure gradient, which negotiates a step change in surface roughness. The distributions of the mean quantities of such a boundary layer (see fig 2.1) are described by the mean continuity equation

$$\partial U/\partial x + \partial V/\partial y = 0, \qquad (2.1)$$

and the mean momentum equation

$$U\,\partial U/\partial x + V\,\partial U/\partial y = U_\infty \frac{dU_\infty}{dx} + \nu \frac{\partial^2 U}{\partial y^2} + \frac{\partial}{\partial y}\left(\overline{-u'v'}\right) \qquad (2.2)$$

where ν denotes the kinematic viscosity and $\overline{-u'v'}$ stands for the kinematic turbulent shear stress, τ/ρ

The primed quantities in Eq.(2.2) denote turbulent quantities. The meaning of the quantities in Eq(2.2) follows from fig. 2.1. The Eqs.(2.1) and (2.2) are derived with the help of the usual averaging procedure and boundary layer approximations from the

Navier – Stokes equations.

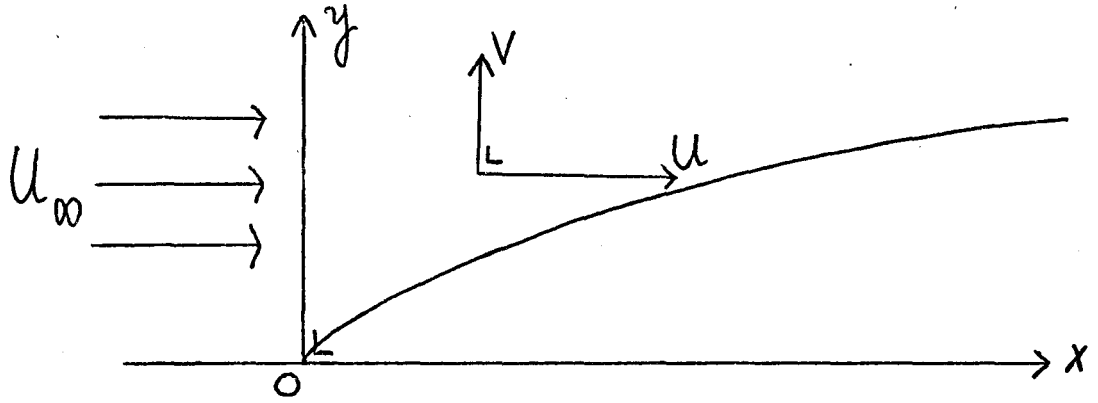


Fig. 2.1 Schematic presentation of a two – dimensional boundary layer.

U, V and $\overline{-u'v'}$ are the unknown quantities, so as it looks now, the above system of equations is indeterminate: the number of equations being one less than the number of unknown quantities. In order to make the system of equations determinate, one has to find an expression for $\overline{-u'v'}$ in terms of the other mean quantities or deduce further relations between the unknown quantities. The solution of this so called closure – problem has been the main aim of many turbulent boundary layer theories. Theories (e.g. Elliott, and Panofsky and Townsend) which have been deduced for micrometeorological purposes are based on mixing – length or eddy – viscosity hypotheses.

In these hypotheses the Reynolds shear stress is related to the local normal gradient of the longitudinal component of mean velocity. A fundamental objection to the use of mixing-length formulas for boundary layers in arbitrary pressure gradients is the fact that $-\overline{u'v'}$ is only related to the local mean quantities, the effect of the past history of the boundary layer being ignored. Townsend (1965) in fact, indicated a new approach to the closure-problem. Starting from the turbulent kinetic energy equation, he showed that the mixing-length hypotheses is valid in the inner, fully turbulent part of the boundary layer. In connection with this new development Bradshaw, Ferriss and Atwell (1967) related the kinematic Reynolds shear stress closely to the turbulent kinetic energy, $\frac{1}{2} \rho q^2$, which quantity, being governed by the turbulent kinetic energy equation, is certainly not determined uniquely by the local mean flow conditions. In this way, the turbulent quantity $-\overline{u'v'}$ is related to other turbulent properties, which seems to be a better hypothesis than relating a turbulent property solely to the properties of the mean velocity-field.

Since their prediction method seems to have been adequately tested for a reasonable number of problems, it was decided to use it for the present investigation. In section 2.2 this prediction method will be treated in more detail.

2.2 The Bradshaw – Ferriss – Atwell prediction method.

With the help of the usual boundary layer approximations the turbulent energy equation can be written as

$$U\frac{\partial \frac{1}{2}\overline{q^2}}{\partial x}+V\frac{\partial \frac{1}{2}\overline{q^2}}{\partial y}=-\overline{u'v'}\frac{\partial U}{\partial y}-\frac{\partial}{\partial y}\left(\frac{1}{2}\overline{q^2v'}+\frac{\overline{p'v'}}{\rho}\right)-\epsilon \qquad (2.3)$$

(Bradshaw, Ferriss, Atwell, 1967)

This equation is valid outside the viscous sublayer and the transition layer, $^{\otimes}y^+=yu_*/\nu>40$. Beyond this region of the boundary layer, the contribution of the viscous term to the total kinematic shear stress, $\tau/\rho=\nu\partial U/\partial y-\overline{u'v'}$, can be ignored.

In Eq. (2.3) the terms on the left represent the rate of change of specific turbulent kinetic energy along a streamline of the mean flow, sometimes called the advection of specific turbulent kinetic energy by the mean flow. The first term on the right stands for the production of specific turbulent kinetic energy from the mean flow, the second term for the diffusion of it in the y-direction, and the last term denotes the dissipation into heat by viscous forces. Near the surface the production and dissipation term balance each

⊗ Note: u_* stands for the surface-friction velocity.

other ; diffusion and advection are negligible there. Near the outer edge of the boundary layer the diffusion is about equal to the advection, while production and dissipation can be neglected.
By introducing the quantities

$$a \equiv \frac{\tau}{\rho \overline{q^2}} ,$$

$$L \equiv \frac{(\tau/\rho)^{3/2}}{\epsilon} ,$$

and

$$G \equiv \left(\overline{\frac{p'v'}{\rho} + \frac{1}{2} q^2 v'} \right) \Big/ \left(\left(\frac{\tau_{max}}{\rho} \right)^{.5} \frac{\tau}{\rho} \right)$$

where $\tau = -\rho \overline{u'v'}$; the Reynolds shear stress;
$\tau_{max} = \tau(y = .25\delta)$, the value of τ at $y/\delta = 1/4$; δ is the boundary-layer thickness;
ρ is the fluid density.
$q^2 = u'^2 + v'^2 + w'^2$, the specific turbulent kinetic energy,

Bradshaw et al. converted Eq. (2.3) into a transport equation for the Reynolds shear stress, which holds outside the viscous sublayer and the transition layer (this

has already been pointed out). The equation reads as

$$U \frac{\partial}{\partial x}\left(\frac{\tau}{2a\rho}\right) + V \frac{\partial}{\partial y}\left(\frac{\tau}{2a\rho}\right) - \frac{\tau}{\rho}\frac{\partial U}{\partial y} + \left(\frac{\tau_{max}}{\rho}\right)^{.5} \frac{\partial}{\partial y}\left(G \frac{\tau}{\rho}\right) + \frac{(\tau/\rho)^{1.5}}{L} = 0 \qquad (2.4)$$

a, L and G are functions of y/δ, which depend on the shape of the shear stress profile. a and G are dimensionless and L has the dimensions of length; L is the most important of the three functions, since over most of the boundary layer the dissipation is much larger than the advection or diffusion.

If adequate assumptions can be made for a, L and G then the Eqs. (2.1), (2.2) and (2.4) form a set of three equations in the unknowns U, V and τ, which is of the hyperbolic type and can be solved numerically. The accuracy of the calculation method depends on the choice of the empirical functions a, L and G. Bradshaw, Ferriss and Atwell obtained a, L and G by using the experimental information of the zero-pressure-gradient turbulent boundary layer of Klebanoff (1955). They showed that the rather simple assumptions of

$a = .15$

$$L = \delta f_1 (y/\delta) ,$$

and

$$\textcircled{\scriptsize G} \quad \Theta = \left(\frac{\tau}{\rho U_\infty^2}\right)^{.5} f_2 (y/\delta) ,$$

(where f_1 and f_2 are numerically specified, or given functions)
are capable of predicting turbulent boundary layer-development in different kinds of pressure gradient flows with satisfactory accuracy. Fig. 2.2 presents the used functions.

The vital assumptions made in the above development are

(a) the boundary-layer approximations;
(b) a, L and Θ change much more slowly than τ and U in the streamwise direction, so that they can be regarded as well-behaved coefficients rather than variables.

⊗ Note: It may be remarked that the empirical variation of Θ with $(\tau_{max}/U_\infty^2)^{.5}$ is the only information needed in the calculations that is not derived solely from the boundary layer in zero pressure gradient.

The second approximation demands that sudden changes at one value of y (for instance, a change in boundary conditions at the surface) shall not produce large changes in a, L and G at other values of y.

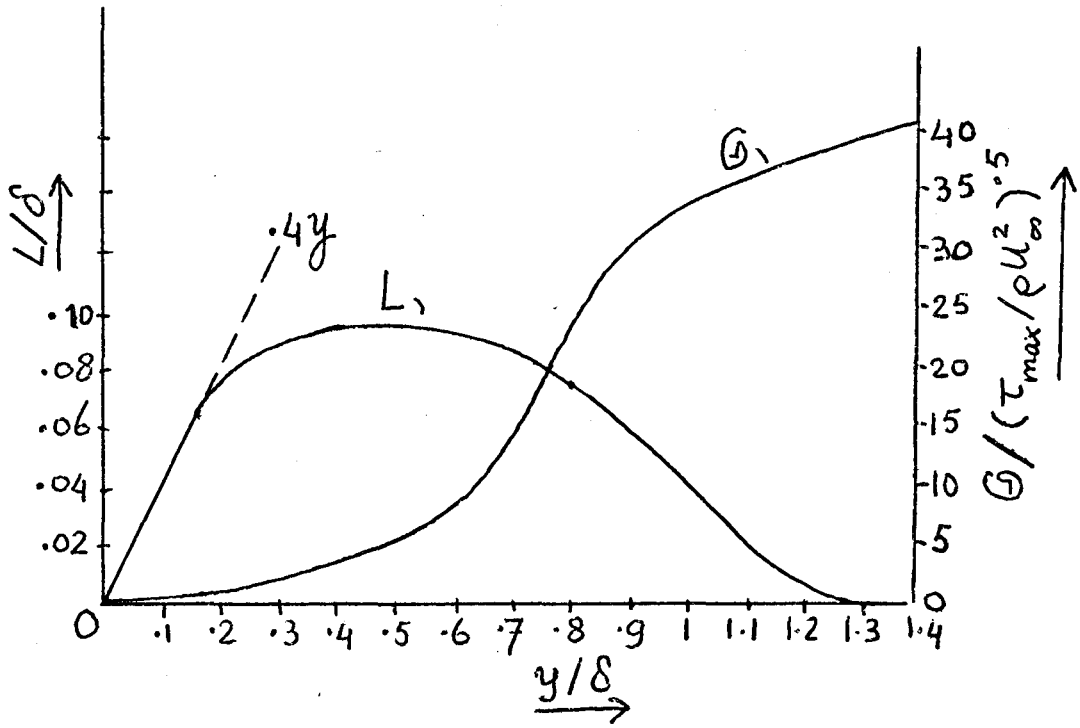


Fig. 2.2 Empirical functions, which are used in the calculations (after Bradshaw et al., 1967)

It can easily be proved that the calculation method of Bradshaw et al. reduces to mixing length theory in those regions - the inner region (thickness about .15 δ) - of the turbulent boundary layer in which that

theory might be expected to be valid. Within the fully turbulent part of the boundary layer, beyond the viscous sublayer and the transition layer, the advection and diffusion of turbulent kinetic energy may be neglected, so that one finds from Eq.(2.4)

$$\frac{\tau}{\rho}\frac{\partial u}{\partial y} = \frac{(\tau/\rho)^{1.5}}{L}, \text{ or } \frac{\partial u}{\partial y} = \sqrt{\tau/\rho}/L \qquad (2.5)$$

Hence, under these circumstances the dissipation length-parameter L is identical with the mixing length. Bradshaw et al. have availed themselves of the mixing-length hypotheses to derive the boundary conditions at $y^+ = u_* y/\nu = 40$ for their numerical solution.

In the next chapter, the numerical calculation scheme developed by Bradshaw, Ferriss and Atwell for solving the system of Eqs.(2.1), (2.2) and (2.4), will be introduced.

3. Scheme of numerical integration.

3.1 Introduction.

The set of governing equations (2.1), (2.2) and (2.4) cannot be solved analytically. To obtain an approximate solution a numerical integration scheme has to be adopted. Therefore, all equations are written in a so-called finite difference form (see section 3.2). A grid, usually rectangular (see fig. 3.1) is placed on the flow field, and the flow quantities are to be calculated only at the nodes (mesh points). Each derivative is approximated by an appropriate difference quotient.

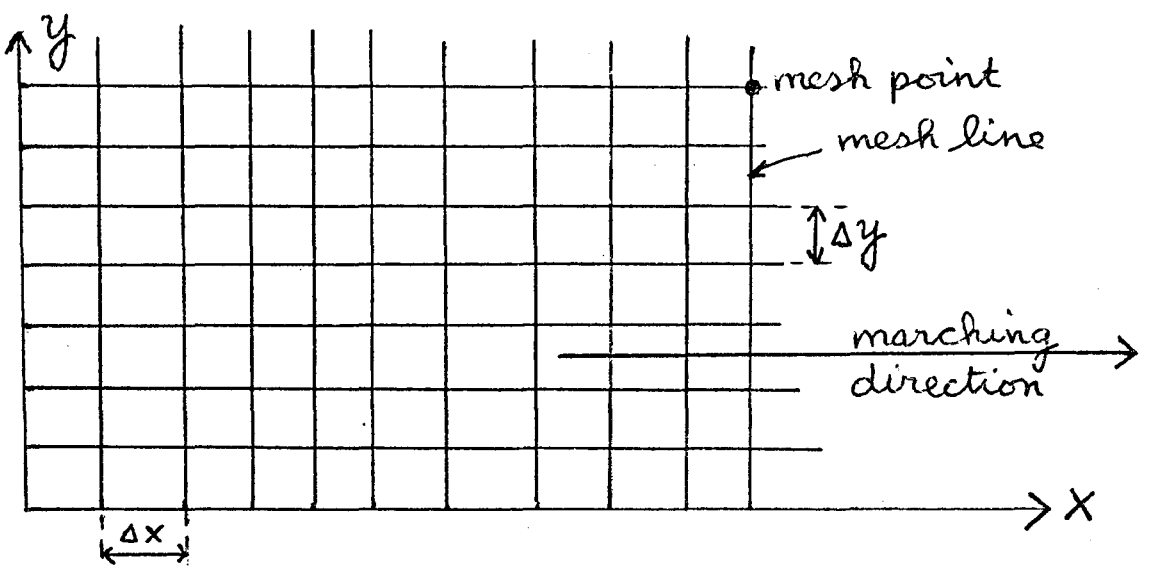


Fig. 3.1 Rectangular grid in the two-dimensional flow field.

3.2 The numerical calculations.

The in section 3.1 described system of partial differential equations is of the hyperbolic type. this means that a difference scheme can be devised which is solved applying the so-called method of characteristics. The method involves a marching procedure, the direction of marching being in the predominant flow direction (see fig. 3.1 of section 3.1). At each X-station (mesh line), the unknowns are calculated in terms of their values at the previous station (see fig. 3.2)
There exist as many real characteristic directions as there are equations for the unknown quantities. Because the boundary layer approximations have been used to derive the equations, at least one characteristic will be vertical. The two others are in the direction of the streamlines, one is inclined at a steeper angle (α) than the other (β). Along these directions the partial differential equations reduce to ordinary differential equations which can be integrated to obtain solutions for the unknowns. In the lower 20 % of the boundary layer the gradients are very large, which implies that the quantities in this region change much

more rapidly than in the remainder of the boundary layer. After the values of the unknowns have been computed at points on a mesh line, a new estimate of these quantities is obtained, by repeating the integration-procedure, in the first 20 %. of the boundary layer.

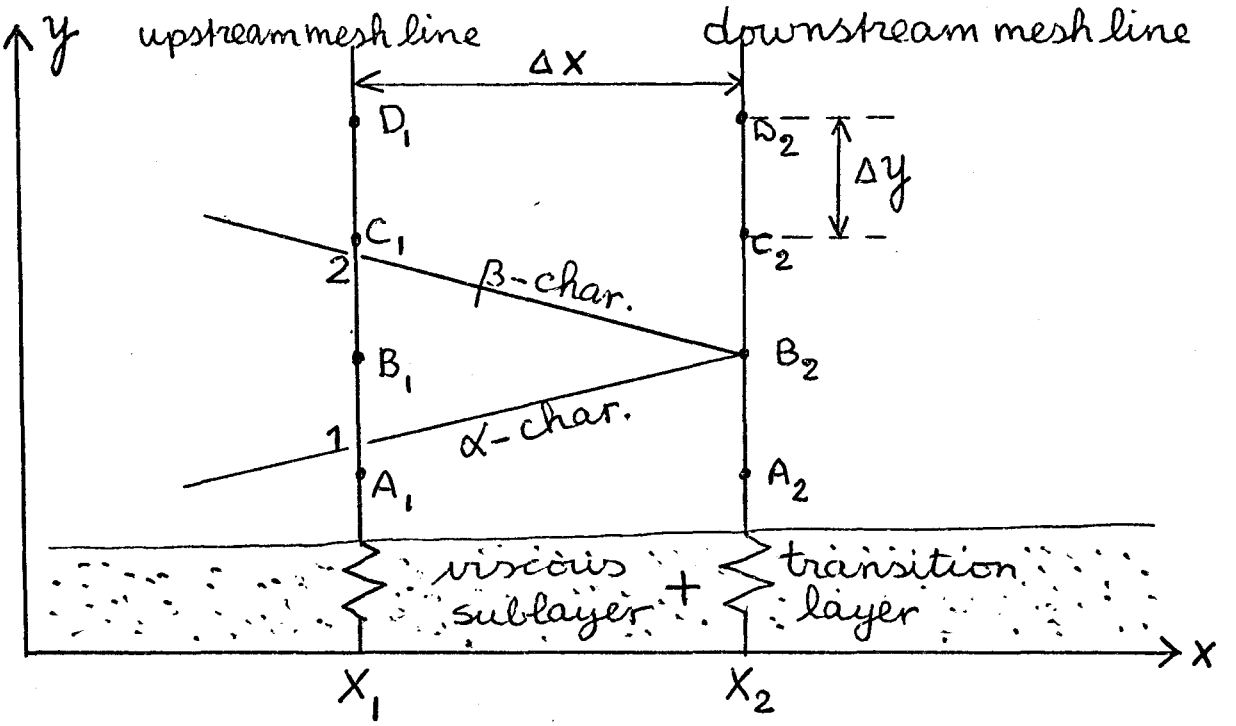


Fig. 3.2 Scheme of integration along the characteristic directions.

The changes in the rest of the boundary layer do not warrant this second iteration.
The most important equations used in the calculation procedure are reproduced here

in finite difference notation. The subscript notation will be used to denote the mesh points A, B, C, etc., and U, V and τ to stand for these values at the mesh points on the downstream line ($X = X_2$), where they have to be computed.
The equations along the α and β characteristics are

$$\tau_\alpha U_i - \tfrac{1}{2}\sigma_\alpha \tau_i = \phi_\alpha \qquad (3.1.a)$$

$$\tau_\beta U_i - \tfrac{1}{2}\sigma_\beta \tau_i = \phi_\beta \qquad (3.1.b)$$

Here,

⊗

$$\phi_\alpha = \tau_\alpha U_1 - \tfrac{1}{2}\sigma_\alpha \tau_1 + \Delta x \frac{\tau_\alpha}{U_\alpha}\left\{\mathcal{P} + a\left(\frac{\sqrt{\tau_\alpha}}{L_\alpha} + \mathcal{G}'_\alpha M\right)\sigma_\alpha\right\} \qquad (3.2.a)$$

$$\phi_\beta = \tau_\beta U_2 - \tfrac{1}{2}\sigma_\beta \tau_2 + \Delta x \frac{\tau_\beta}{U_\beta}\left\{\mathcal{P} + a\left(\frac{\sqrt{\tau_\beta}}{L_\beta} + \mathcal{G}'_\beta M\right)\sigma_\beta\right\} \qquad (3.2.b)$$

$$\sigma_\alpha = \mathcal{G}_\alpha M + \sqrt{\mathcal{G}_\alpha^2 M^2 + 2\tau_\alpha/a} \qquad (3.3.a)$$

$$\sigma_\beta = \mathcal{G}_\beta M - \sqrt{\mathcal{G}_\beta^2 M^2 + 2\tau_\beta/a} \qquad (3.3.b)$$

⊗ Note: the subscripts α and β denote the average values on the characteristics

In these equations, and the following analysis τ is written for τ/ρ, for simplicity. M stands for $\sqrt{\tau_{max}/\rho}$, and P denotes $U_\infty \frac{dU_\infty}{dx}$.

G' stands for the derivative of G with respect to y. The subscripts 1 and 2 indicate the values of quantities at the intersection points 1 and 2 of the characteristic line segments, drawn from points on the downstream mesh line, and the upstream mesh line. The values at the intersections (see fig. 3.2) are calculated using a quadratic interpolation formula (to be described in appendix A) and the known values at three adjacent points (A_1, B_1 and C_1 in fig. 3.2).
Solving the Eqs. (3.1) for U_i and τ_i there result

$$U_i = \frac{\sigma_\alpha \phi_\beta - \sigma_\beta \phi_\alpha}{\sigma_\alpha \tau_\beta - \sigma_\beta \tau_\alpha} \qquad (3.4.a)$$

$$\tau_i = 2\frac{\tau_\alpha \phi_\beta - \tau_\beta \phi_\alpha}{\sigma_\alpha \tau_\beta - \sigma_\beta \tau_\alpha} \qquad (3.4.b)$$

Eqs. (3.4) are only valid outside the viscous sublayer and the transition layer. The mesh point, A, closest to the surface (and therefore called the first mesh point) is

assumed to lie above these regions of the boundary layer. The spacing, Δy, between each pair of mesh points above the first mesh point is taken constant. The quantities in the points B, C, etc. are computed using Eqs.(3.4). The α-characteristic line-segment drawn from A_2 may intersect the upstream mesh line at a point which lies in the transition- or sublayer where the equations are not valid. So, to calculate u_A and τ_A at A_2, the Eq.(3.1.a) along the α-characteristic has to be replaced by another one, which relates u_A and τ_A to each other. The latter equation is called the 'law of the wall'-equation, which has a form that depends on the kind of surface we have to deal with. The shear stress at the wall is calculated from extrapolation of the shear stress profile, and according to the no-slip condition U and V at the wall are set equal to zero. As upper boundary conditions the U-component of mean velocity is set equal to the free-stream velocity, and τ is set equal to zero.

In appendix C a detailed description of the calculations at the first mesh point, and the computation of the wall-shear stress is given.

A computer program (described in appendix B) has been developed to carry out the calculations. In the next chapter the results of the tests on the program are discussed. Thereafter follow the results of the calculations on the behaviour of a turbulent boundary layer that takes a step-change in surface roughness.

In interpreting the results one cannot ignore a few shortcomings of the calculation procedure, which are

- the solution to the problem is not an exact one; the system of non-linear partial differential equations cannot be solved analitically. The adopted numerical approximate solution is characterized by its simplicity and accurateness.

- Since the set of governing equations is only valid beyond the viscous sublayer and the transition layer, some auxiliary equation is needed to calculate U and τ near the wall. The main source of numerical inaccuracy is the need to extrapolate the shear stress inwards from the first mesh point to the wall.

4. Calculation results.

4.1 Introduction.

To carry out the calculations, described in this report, a computer program has been developed. This program has been written in algol, and a listing is presented in appendix B.
Before doing the calculations on the response of a turbulent boundary layer to a step-change in surface roughness, the computer program has been tested for its prediction capabilities. Therefore, the calculations of some equilibrium turbulent boundary layers (see section 4.2), of which detailed experimental data are available (Stanford Conference, 1968), have been carried out. The results of these computations have been compared with those of Bradshaw et al. (1967), who also have tested their method, using the same boundary layers.
Boundary layers of this type are to the first approximation self-preserving⊗, and their behaviour is well understood through the use of the so-called similarity concept.

⊗ Note: The theoretical conditions to be fulfilled to obtain an exact 'self-preserving' or 'equilibrium' boundary-layer flow are that the Clauser (1954) equilibrium-parameter $\delta^*/\tau_w \frac{dp}{dx}$ and u_*/u_∞ must be held constant (Rotta, 1962).

4.2 Tests of the computer program.

In order to check the prediction-qualities of the calculation program, it has been run for some equilibrium turbulent boundary layers along a smooth surface:

a The zero-pressure gradient turbulent boundary layer of Klebanoff (1955), the profiles of the U-component of mean velocity and turbulent shear stress are shown in Bradshaw's (1967) paper.

b Two turbulent boundary layers under the influence of adverse pressure gradients, profiles are also present in Bradshaw's (1967) paper.

c A turbulent boundary layer under a small favourable pressure gradient (Kersels, 1977).

In appendix D all the initial profiles of the U-component of mean velocity and turbulent shear stress, needed to start the calculations, are presented.

4.2.1 Test - results.

The boundary layers, which are discussed in this section, have also been used by Bradshaw et al. to test the prediction method. Their results have been compared with experimental data (Stanford Conference, 1968) of the boundary layers. The present calculation results have been compared with those of Bradshaw et al. and the experimental data.
Unfortunately, it was not possible to compare the calculation results directly with appropriate experimental data of Reynolds shear stress - profiles. In judging the results of the shear stress profiles there has been used the concept of similarity.
The boundary-layer growth and the wall-friction coefficient have also been compared with theoretical predictions, which are valid in equilibrium boundary layers under the influence of a zero, or small pressure gradient. These theoretical prediction-formulas are presented in appendix B, and will be listed here for convenience.
Hinze: $(.5 < Re_x * 10^{-6} < 10)$

$$\delta \sim Re_x^{(H+1)/(3H-1)} \qquad (B.4.a)$$

⊗ Note: Similar formulas hold for δ^* and θ.

$$c_F \sim Re_x^{-2(H-1)/(3H-1)} \qquad (B.4.b)$$

Ludwieg and Tillmann: ($10^3 < Re_\theta < 10^4$)

$$c_F = .246 * 10^{-.678H} Re_\theta^{-.268} \qquad (B.5)$$

Rotta:

$$c_F = \frac{2}{(5.75 \log Re_{\delta^*} + 3.7)^2} \qquad (B.6)$$

The calculation results are presented in graphical form, viz.

- the profiles of the U-component of mean velocity; U/U_∞ vs. y/δ
- the profiles of the turbulent shear stress; τ/τ_w vs. y/δ
- the wall-friction coefficient; c_f/c_{f_0} vs. Re_θ (c_{f_0} is the value of c_f at the starting position of the calculations)
- the boundary-layer growth; δ/δ_0 vs. x/δ_0 (δ_0 is the value of δ at the starting position). In this plot there are also presented the values of H and θ/θ_0 against x/δ_0.

The agreement between the velocity profiles calculated by the present program, and those predicted by Bradshaw, and the experimental data, is very good. The difference between them is very small, and therefore not indicated in the plots of the velocity profiles.

The computational results of Klebanoff's zero pressure gradient turbulent boundary layer are shown in figs. 4.1 to 4.5.
The calculations have been carried out for about 100 x-steps[+] downstream from the starting position. The number of mesh points (imaxe) on the initial profiles was taken equal to 50; [⊗] the corresponding value of ΔX, the increment in X per X-step, is chosen about equal to .6 times the boundary-layer thickness, δ_o, at the starting position (according to the courant-condition (appendix B, Eq.(B.9)), xstep is approximately equal to .6; ΔX is commonly chosen equal to xstep).
In fig. 4.1.b oscillations occur in the shear stress profile near the outer edge of the boundary layer and especially near the wall. It is difficult to handle the calculations near the wall. The large steepness of the longitudinal mean velocity profile, and the

⊗ Note: It may be remarked that Bradshaw et al. used 20 mesh points in their calculations,

+ Note: the number of x-steps is denoted by the number j.

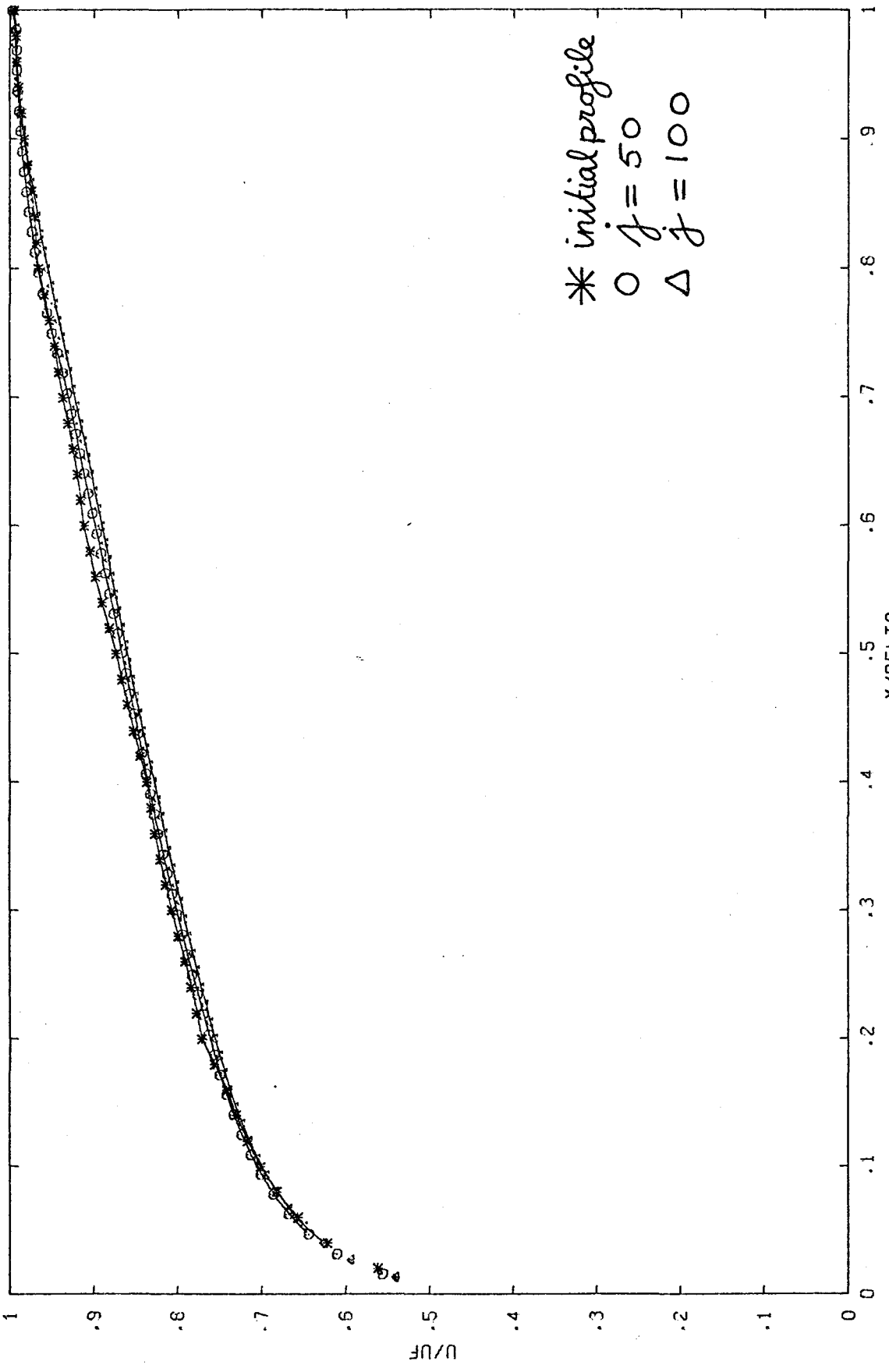


Fig. 4.1.a U-component of mean velocity (Klebanoff, 1955)

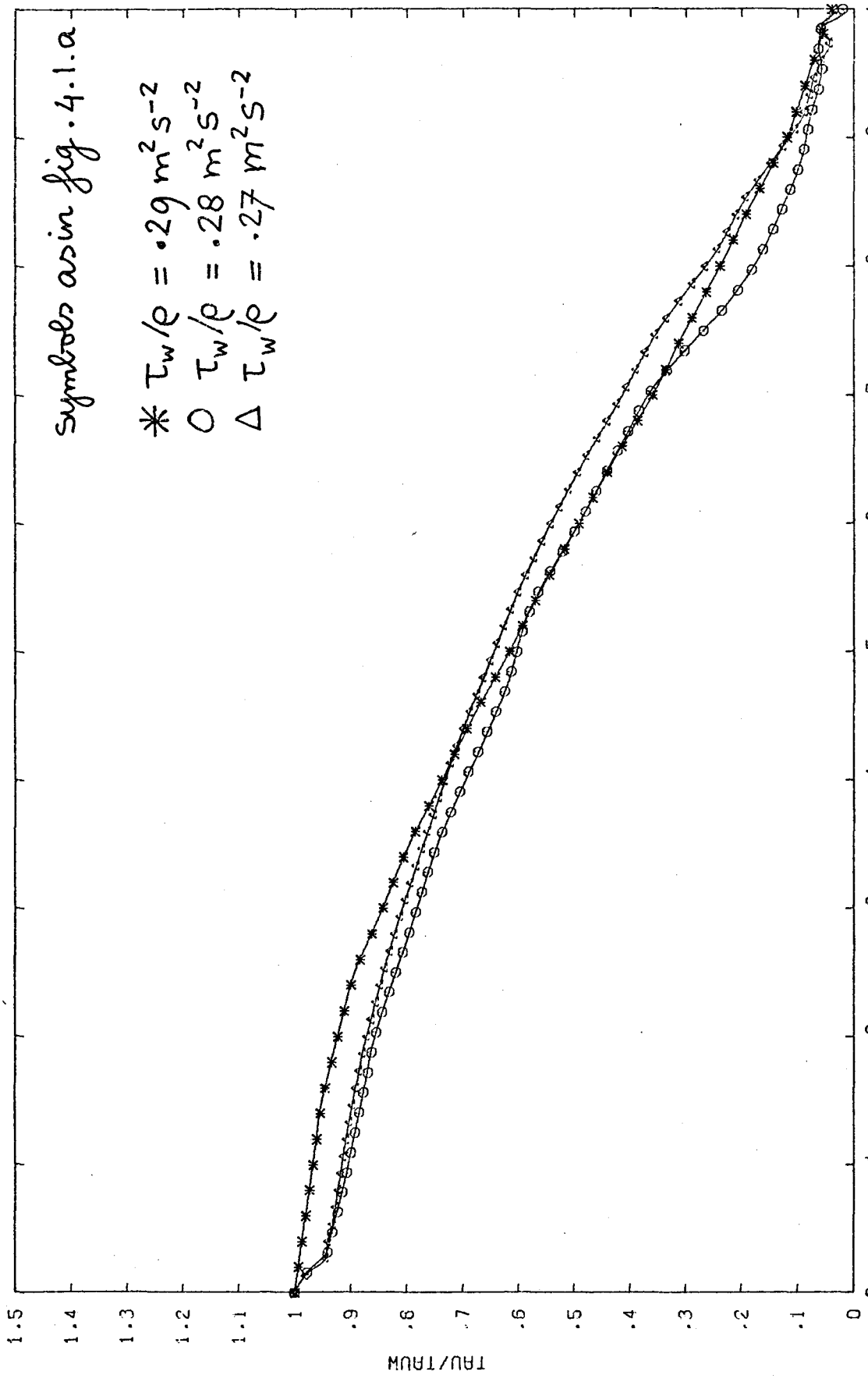


Fig. 4.1.b Turbulent shear stress (Klebanoff, 1955)

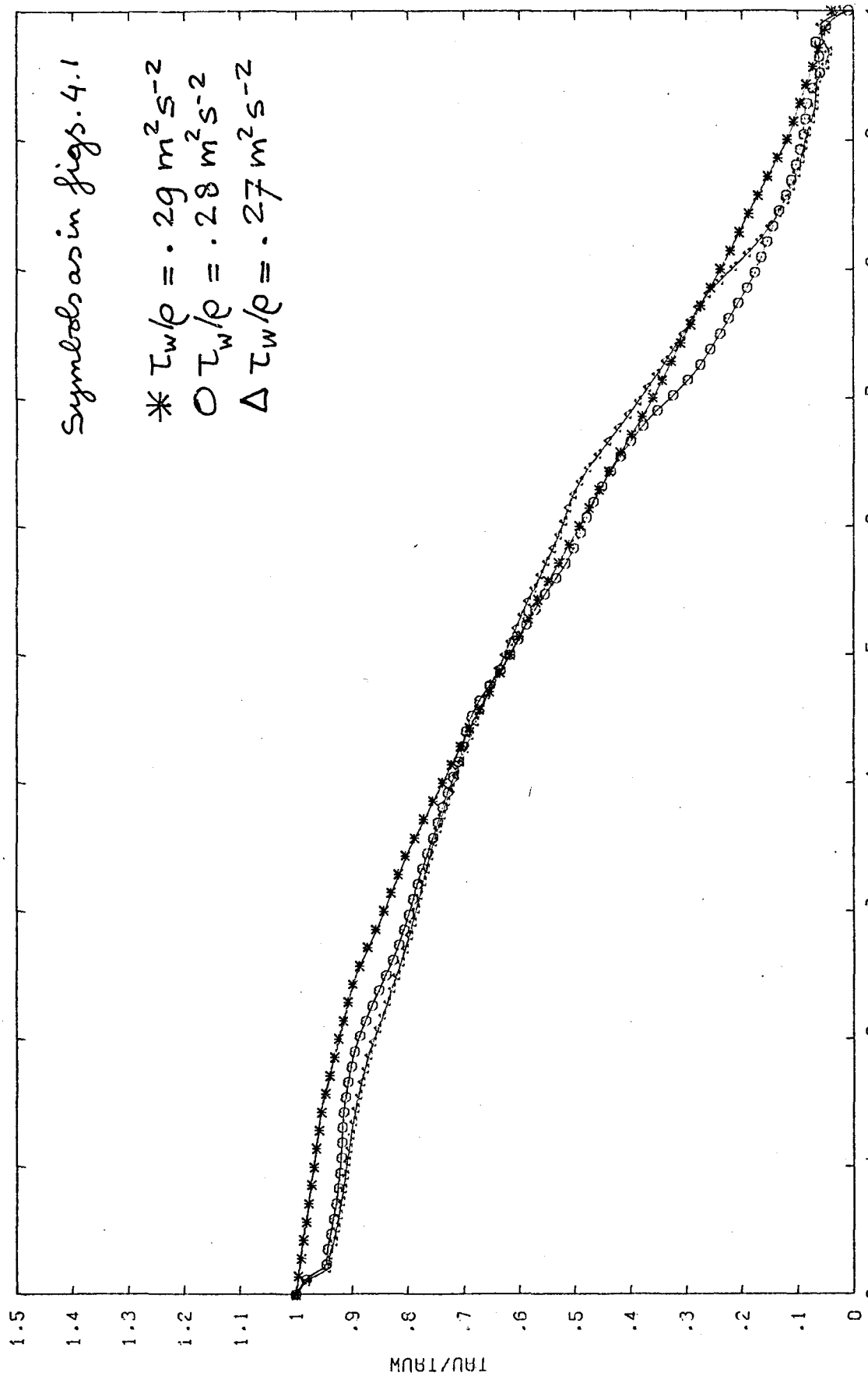


Fig. 4.2 Turbulent shear stress (Klebanoff, imax = 70)

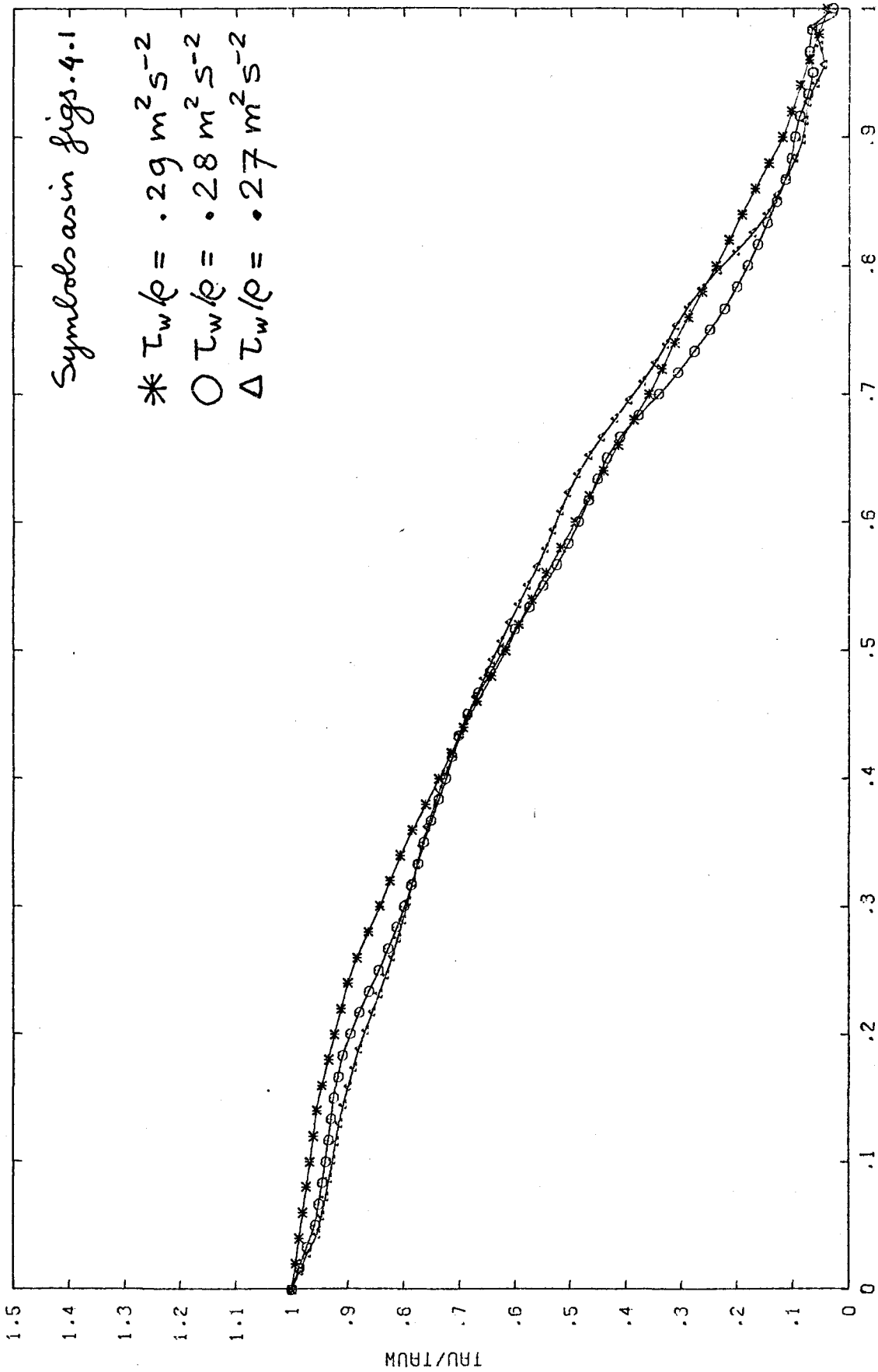


Fig. 4.3 Turbulent shear stress (Klebanoff, $\Delta x = .7 * \times$ step)

rapid changes of its normal derivatives may cause large errors in the finite difference approximations of these derivatives.⊛ Consequently, the computed shear stress may be in error (for, the prediction of shear stress is roughly the same as prediction of the mean velocity gradient (Bradshaw et al., 1967)). The oscillations near the outer edge of the boundary layer are probably caused by the way the outer boundary conditions are handled (appendix B). The oscillations near the wall can be reduced taking more mesh points on the initial profiles, or reducing the value of Δx. Fig. 4.2 presents the results when imax has been taken equal to 70. In fig 4.3 the results for a value of $\Delta x = .7$ * xstep are shown. In the latter figure it can clearly be noticed that the calculations near the wall have improved. The shear stress prediction-results of Bradshaw et al. can be identified with the initial profile. It may be concluded that the present calculation program predicts the shear stress profile, as a whole, reasonably well. According to the similarity-concept, the calculated profiles lie

⊛ Note: The quadratic interpolation in the mean velocity profile between the first and second mesh point is therefore replaced by a logarithmic interpolation, which describes the velocity profile better there.

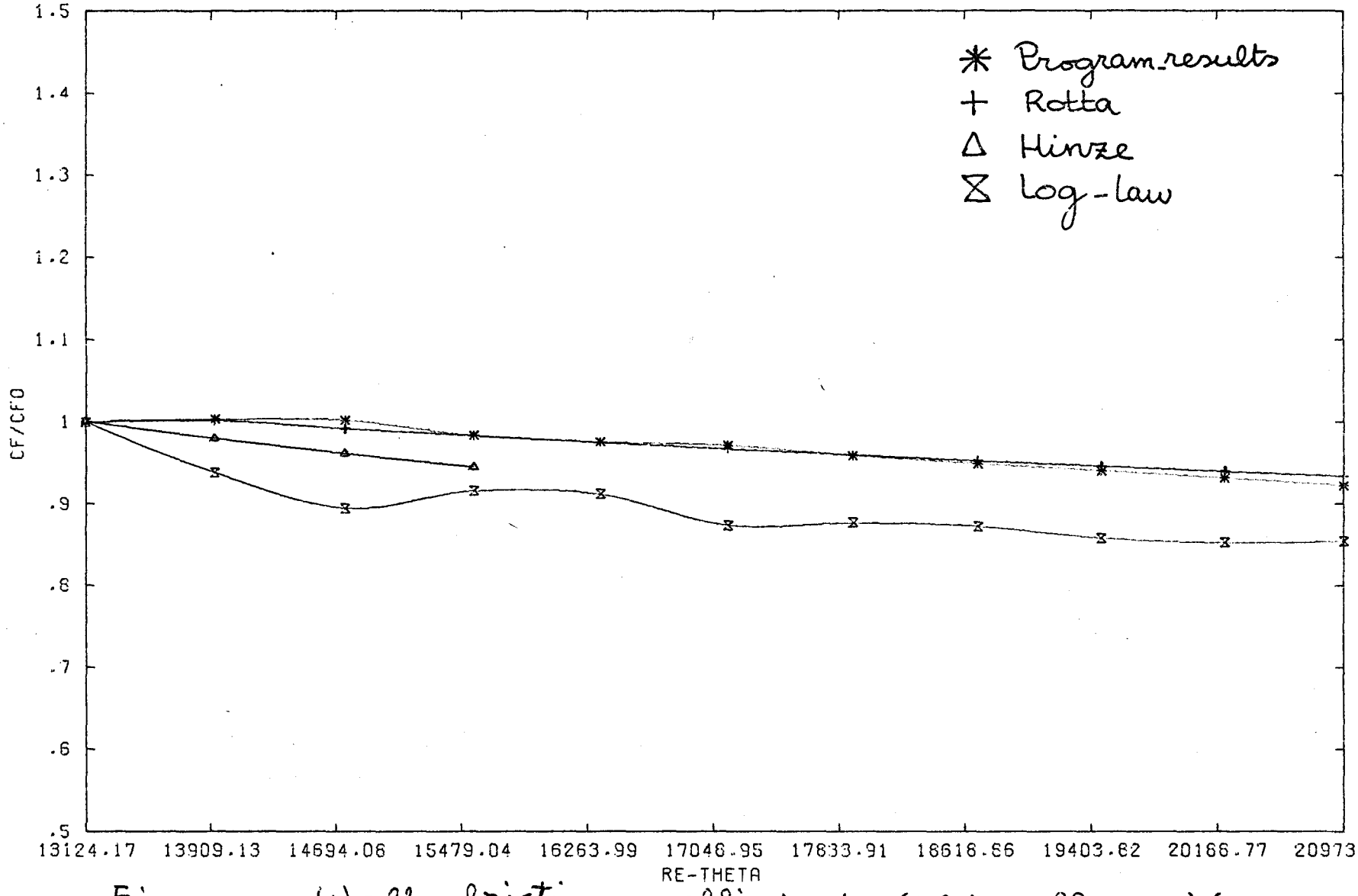


Fig. 4.4 Wall-friction coefficient (Klebanoff, 1955) ($C_{F0}=25*10^{-4}$)

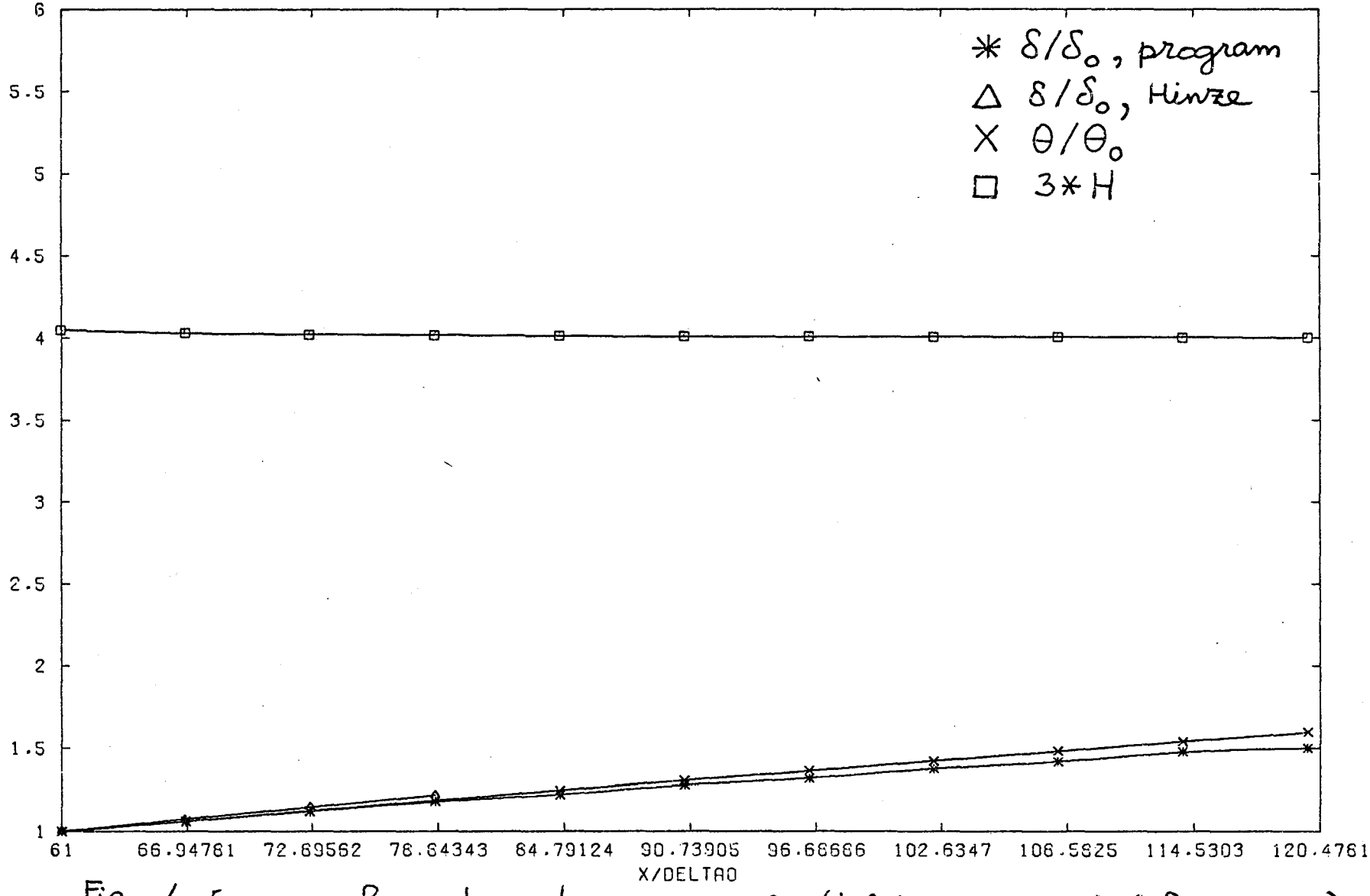


Fig. 4.5 Boundary-layer growth (Klebanoff, 1955) ($\delta_0 = .124m$) ($\theta_0 = .0136m$)

rather close to the initial one.

Fig. 4.4 shows the wall-friction coefficient plotted against the Reynolds number based on the momentum-deficit thickness, θ. Theoretical prediction values are also shown. The lower curve is obtained using the mixing-length assumption that $\partial U/\partial y$ is proportional to $1/y$ inside the fully turbulent part of the boundary layer. ⊗ The oscillations in this curve are due to the differentiation of the longitudinal velocity profile near the surface. Fig. 4.5 presents the boundary-layer growth. Also, here some theoretical predictions have been plotted. Unfortunately, the author of this report, has not been able to use any experimental data, or Bradshaw's calculation results for comparison purposes of boundary-layer growth and wall-friction-coefficient-development of Klebanoff's boundary layer. When compared to the theoretical prediction results (i.e. Hinze), the boundary-layer-growth is predicted too low by the program; the decrement of wall-friction coefficient is calculated too small too. The exponents in Eqs. (B.4.a) and (B.4.b) are equal to .78 and

⊗ Note : This means that there is assumed a semi-logarithmic velocity profile.

$-.22$ respectively ⊗. The boundary-layer thickness and wall-friction coefficient, calculated by the program, yield values of .6 and $-.12$ respectively for the exponent γ, when they are assumed to be proportional to x^{γ}. For θ, the value of γ is equal to .7, so that the momentum-deficit thickness is predicted better than the boundary-layer thickness.

As a severe test, the calculation program has been run for a few boundary layers under adverse pressure gradients. In these types of boundary layers, the simple assumption that the dissipation should be equal to $k\gamma$ in the fully turbulent part becomes questionable. The advection terms are no longer negligible with respect to the dissipation and production terms.

In figs. 4.6 to 4.17 the computational results are shown. They can be divided, according to their origin, into those of:

1. a mild adverse pressure gradient boundary layer, figs. 4.6 to 4.8.
2. a moderate adverse pressure gradient boundary layer, figs. 4.9 to 4.11.
3. a boundary layer, initially at constant

⊗ Note: The exponents are computed, using the value of H, which has been calculated by the computer program.

pressure, developing into equilibrium flow in moderate positive pressure gradient, figs. 4.12 to 4.14.

4. an initial equilibrium boundary layer in moderate positive pressure gradient; pressure gradient abruptly decreases to zero, and flow relaxes toward new equilibrium, figs. 4.15 – 4.17.

In these types of boundary layers, the free-stream velocity varies as $U_\infty \sim x^\alpha$, where α is a constant, so that approximate similarity is obtained (Bradshaw, 1966)

Experimental data of the boundary layers are present in volume II of "Computation of turbulent boundary layers", Stanford Conference 1968. The calculation results of Bradshaw et al. are shown in their report of 1967.

Sub.1: ($\alpha = -.15$)
In fig. 4.6.b, it can clearly be noticed that the oscillations, near the surface, in the turbulent shear stress profile have disappeared. Bradshaw's calculation results, at $x = 2.13$ m, can be identified with the curve of the initial profile.
The calculated shear stress-profiles deviate from

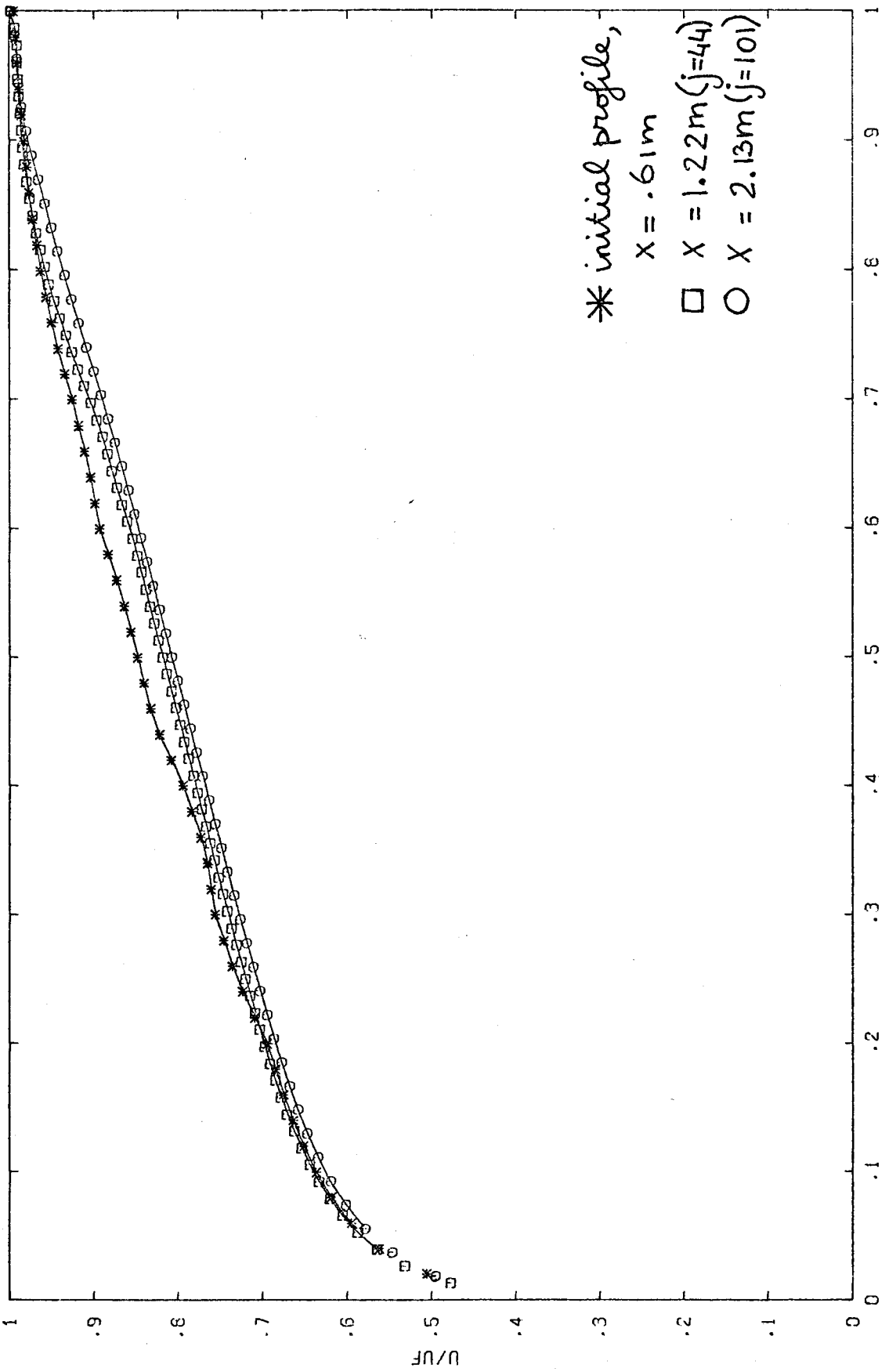


Fig. 4.6.a U-component of mean velocity ($U_\infty \sim x^{-.15}$)

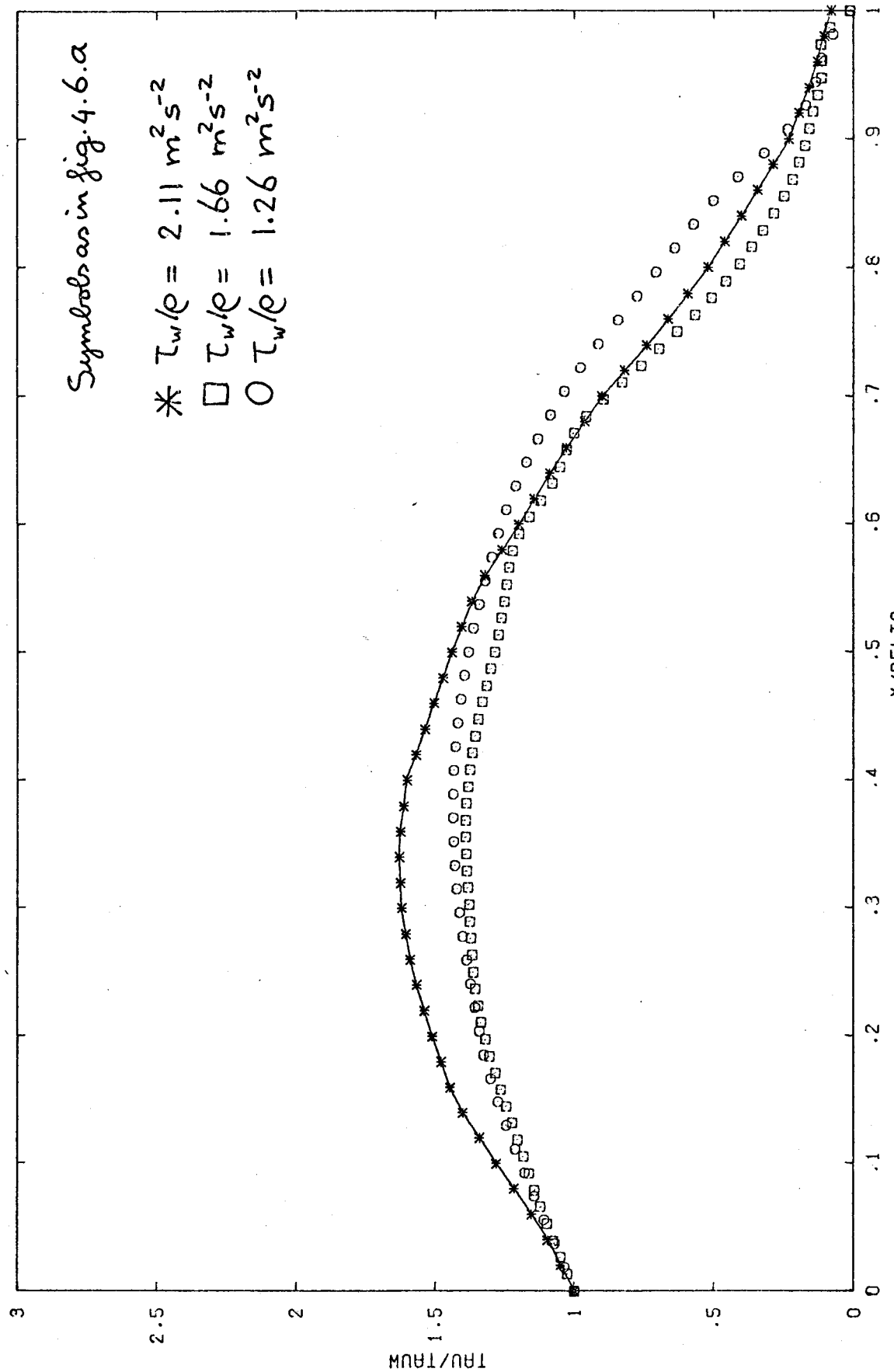


fig. 4.6.b Turbulent shear stress ($U_\infty \sim X^{-.15}$)

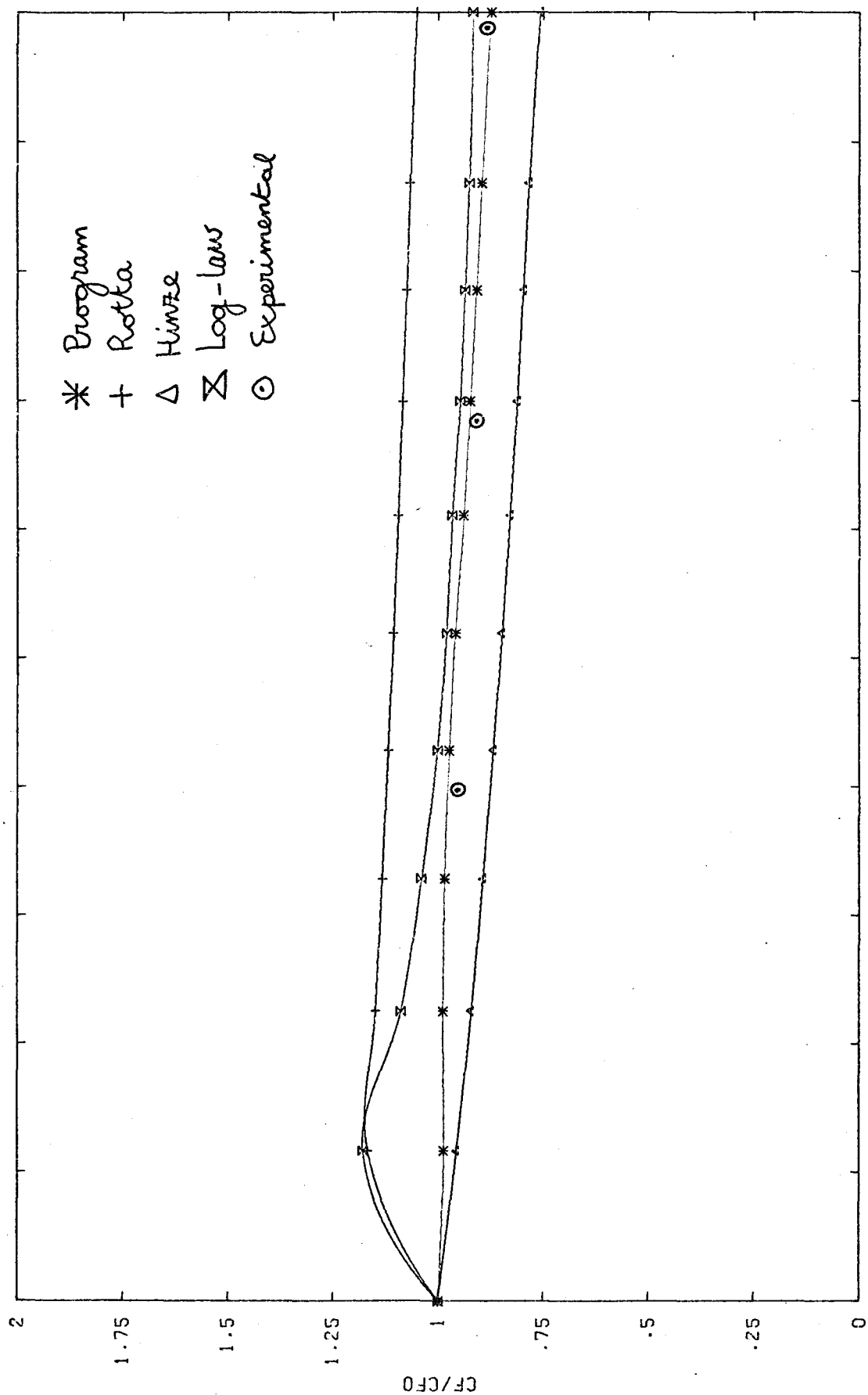


Fig. 4.7 Wall - friction coefficient ($U_\infty \sim x^{-.15}$) ($CF0 = 22*10^{-4}$)

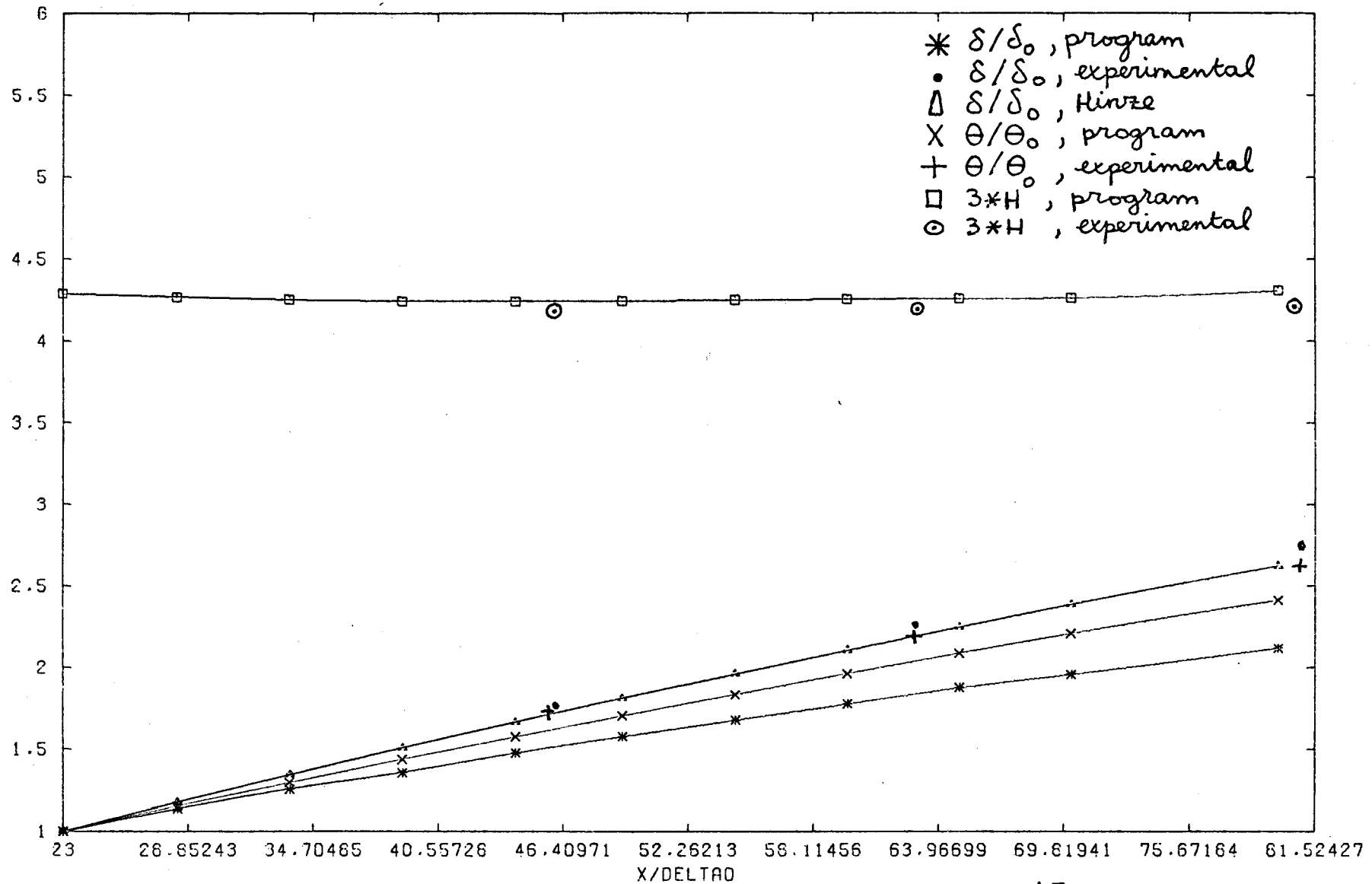


Fig. 4.8. Boundary-layer growth ($U_\infty \sim x^{-.15}$) ($\delta_0 = .0265$ m) ($\theta_0 = .00331$ m)

the initial one. The equilibrium-parameter $\beta = \delta^*/\tau_w \, dp/dx$, calculated by the program, varies from 1.05, at the starting position, to .85 at $x = 2.13$ m; experimental values of β are .99 and .85 respectively.
In fig. 4.7 the wall-friction coefficient results are shown. The experimental values closely agree with those calculated by the program.
Fig. 4.8 presents the boundary-layer growth. There occurs a significant difference between the values predicted by the program, and the theoretical and experimental values. For the momentum-deficit thickness, this difference is smaller.
No results of Bradshaw's calculations were available to put in the figs. 4.7 and 4.8.

Sub 2: ($\alpha = -.255$)
For the boundary layer under a moderate adverse pressure gradient, the differences between the calculation results and the experimental data are more pronounced than for the boundary layer under a mild adverse pressure gradient.
In fig. 4.9.b, the calculated shear stress profiles show a considerable deviation from the initial one. The latter can be identified with the calculated one, by Bradshaw et al., at $x = 2.11$ m.
The equilibrium-parameter, calculated by the program, varies from 5.5 (at the starting position)

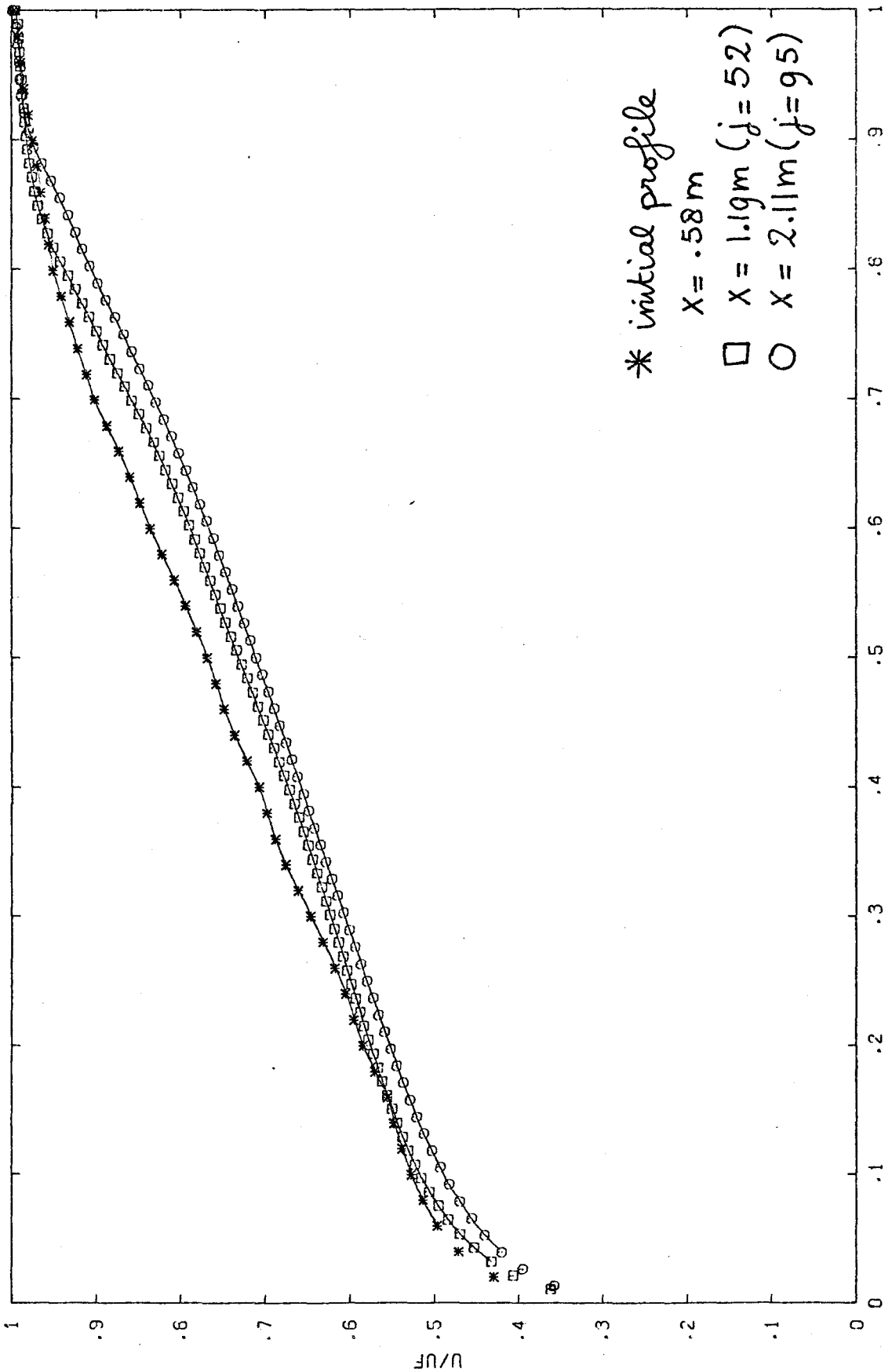


Fig. 4.9.a U-component of mean velocity ($U_\infty \sim x^{-.255}$)

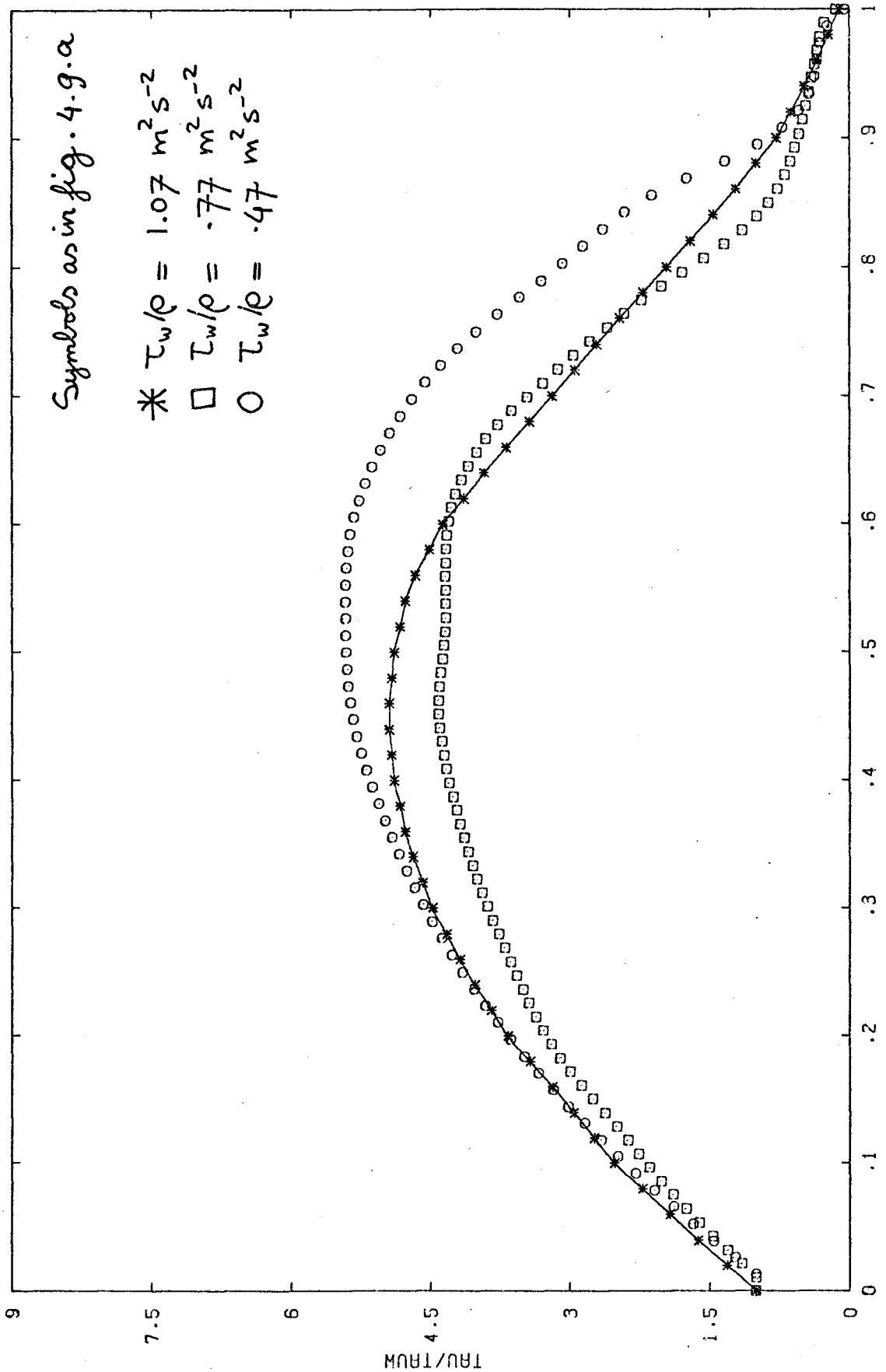


Fig. 4.9.b Turbulent shear stress ($U_\infty \sim x^{-.255}$)

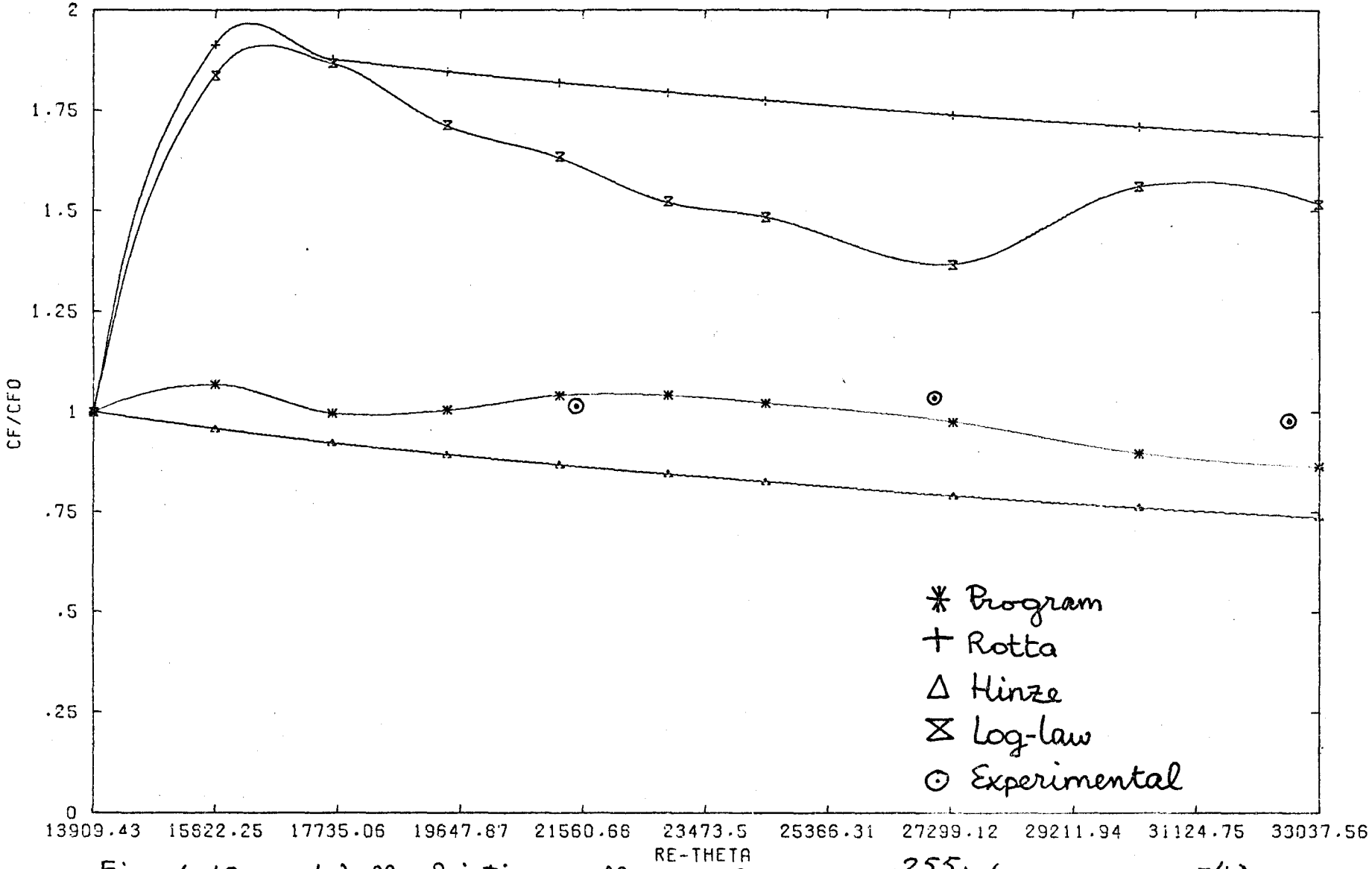


Fig. 4.10 Wall-friction coefficient ($U_\infty \sim x^{-.255}$) ($CF0 = 12 * 10^{-4}$)

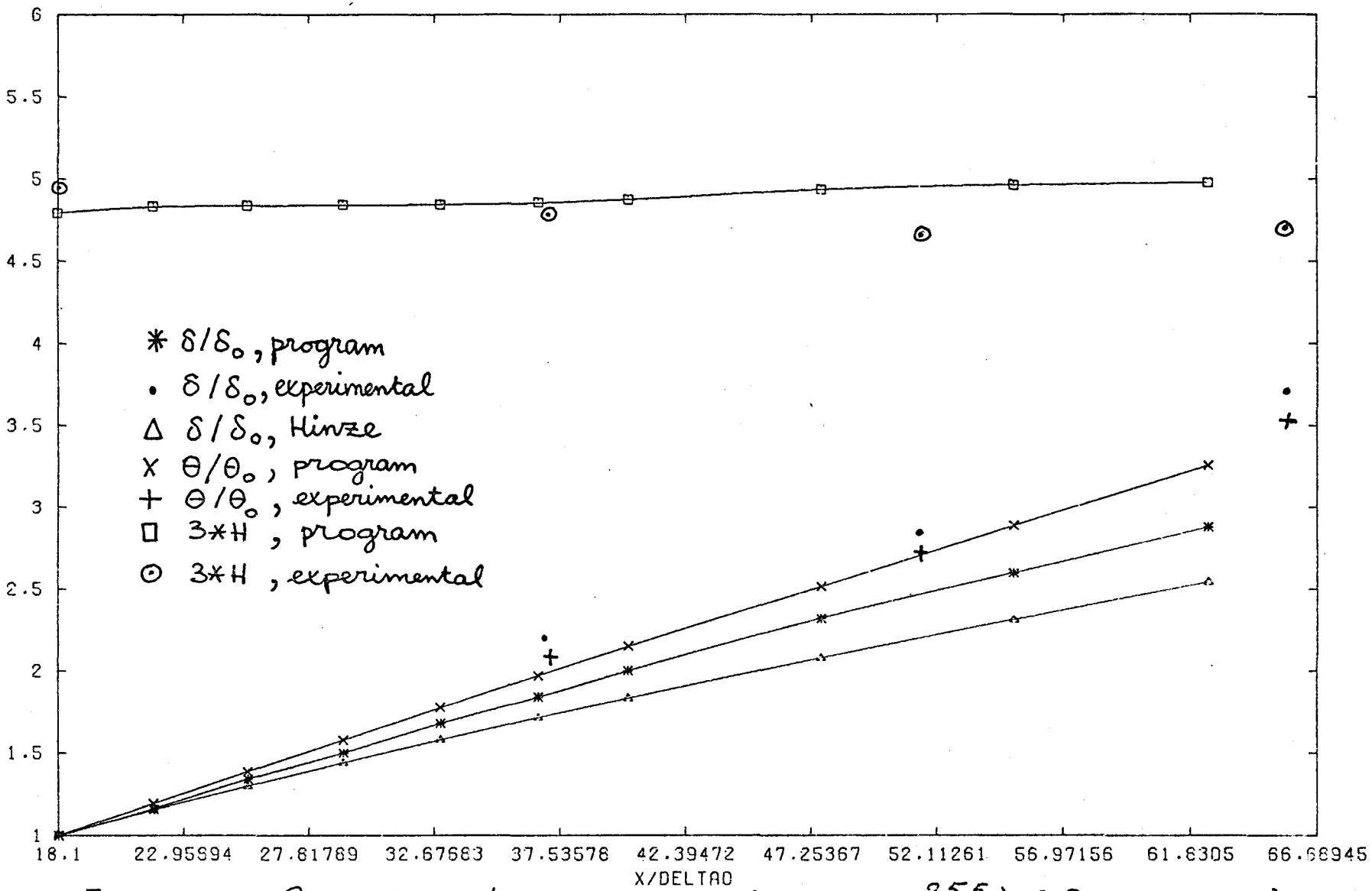


Fig. 4.11 Boundary-layer growth ($U_\infty \sim x^{-.255}$) ($\delta_0 = .032m$) ($\theta_0 = .00485m$)

to 6.3 at $x = 2.11$ m ; the experimental values are 5.1 and 5.4 respectively.
Bradshaw et al. (1967) reported no values of boundary-layer thickness and wall-friction coefficient of this boundary layer.
In the wall-friction coefficient results, in fig. 4.10, there occur oscillations, which may be caused by the large normal gradient of turbulent shear stress, in which case extrapolation to the wall may cause errors.
The log-law values lie far above the program-results; this may be caused by the absence of energy-equilibrium near the wall. However, in fig. 4.9.b, the close agreement, in the inner part of the boundary layer, between the calculated shear stress profiles and the initial one, lends support to the suggestion that the used equilibrium-distribution for the dissipation length L, is a good approximation.
Finally, in fig. 4.11, not only the differences between the calculated and experimental values of δ are present, but also the theoretical values of Hinze differ significantly from the experimental data.

Sub. 3: ($\alpha = 0 \rightarrow -.255$)
In fig. 4.12.a, the development of a velocity profile, from a region of constant pressure into

a region of moderate adverse pressure gradient, has been plotted. The shape of the turbulent shear stress profile changes violently under the influence of the pressure-change. There were no calculated shear stress profiles reported by Bradshaw et al.
For the wall-friction coefficient and the boundary-layer growth, the agreement between Bradshaw's calculation results and the experimental data is very good, except for the shape factor H. Therefore, only the experimental data (for other quantities than H) have been put in the plots of figs. 4.13 and 4.14. Except for δ and H, the program-results compare favourable with those of Bradshaw.

Sub 4: ($\alpha = -.255 \rightarrow 0$)
Finally, the relaxation of an initial equilibrium boundary layer in moderate adverse pressure gradient, which abruptly decreases to zero, has been calculated.
In figs. 4.15.a and 4.15.b the changes in the profiles of longitudinal mean velocity and turbulent shear stress are clearly noticable.
There were no data of calculated shear stress profiles present in the paper of Bradshaw et al.
Now, differences between the experimental results, Bradshaw's calculations and the calculation results of the present program are clearly noticable

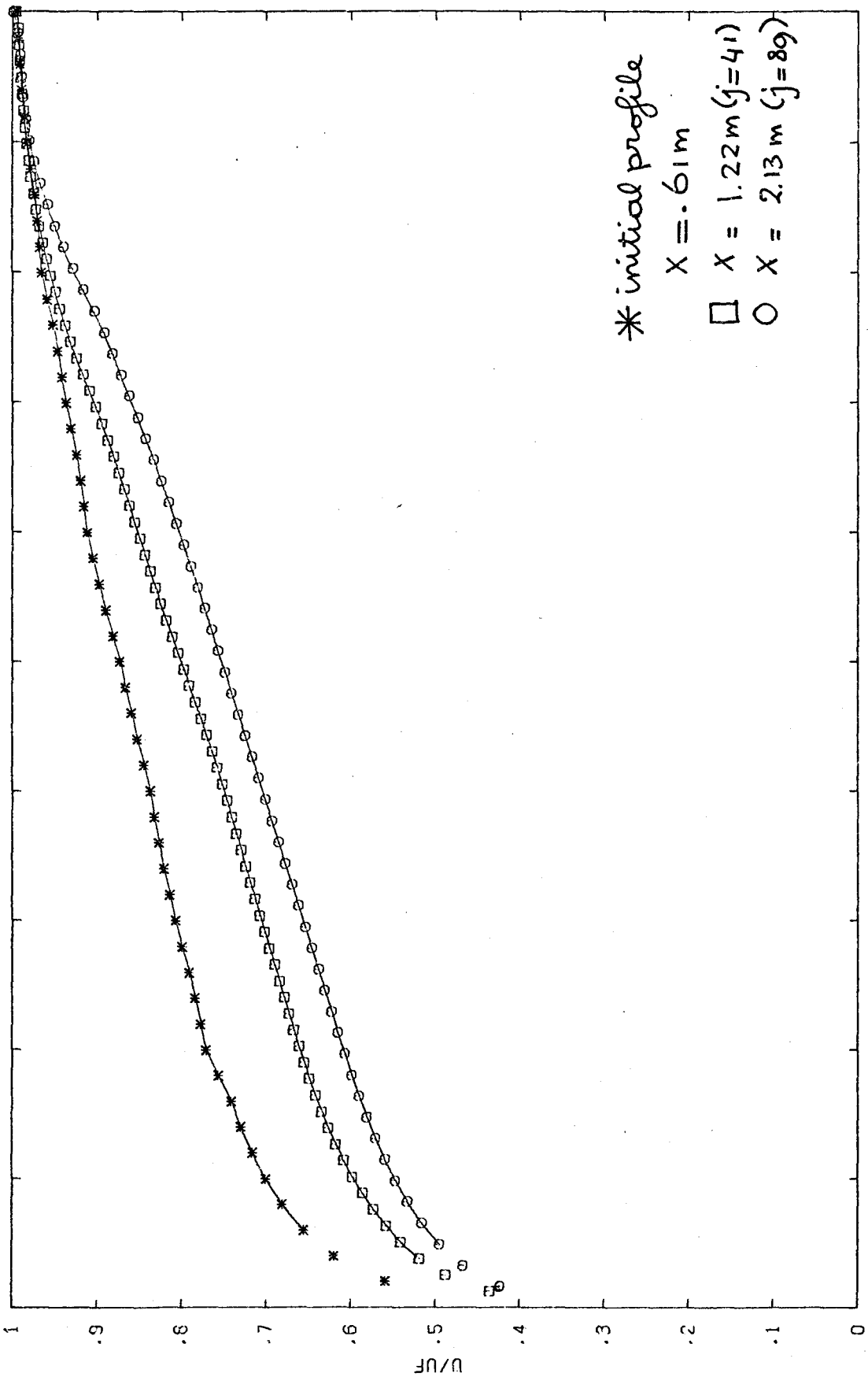


Fig. 4.12.a U-component of mean velocity (α=0 → -.255)

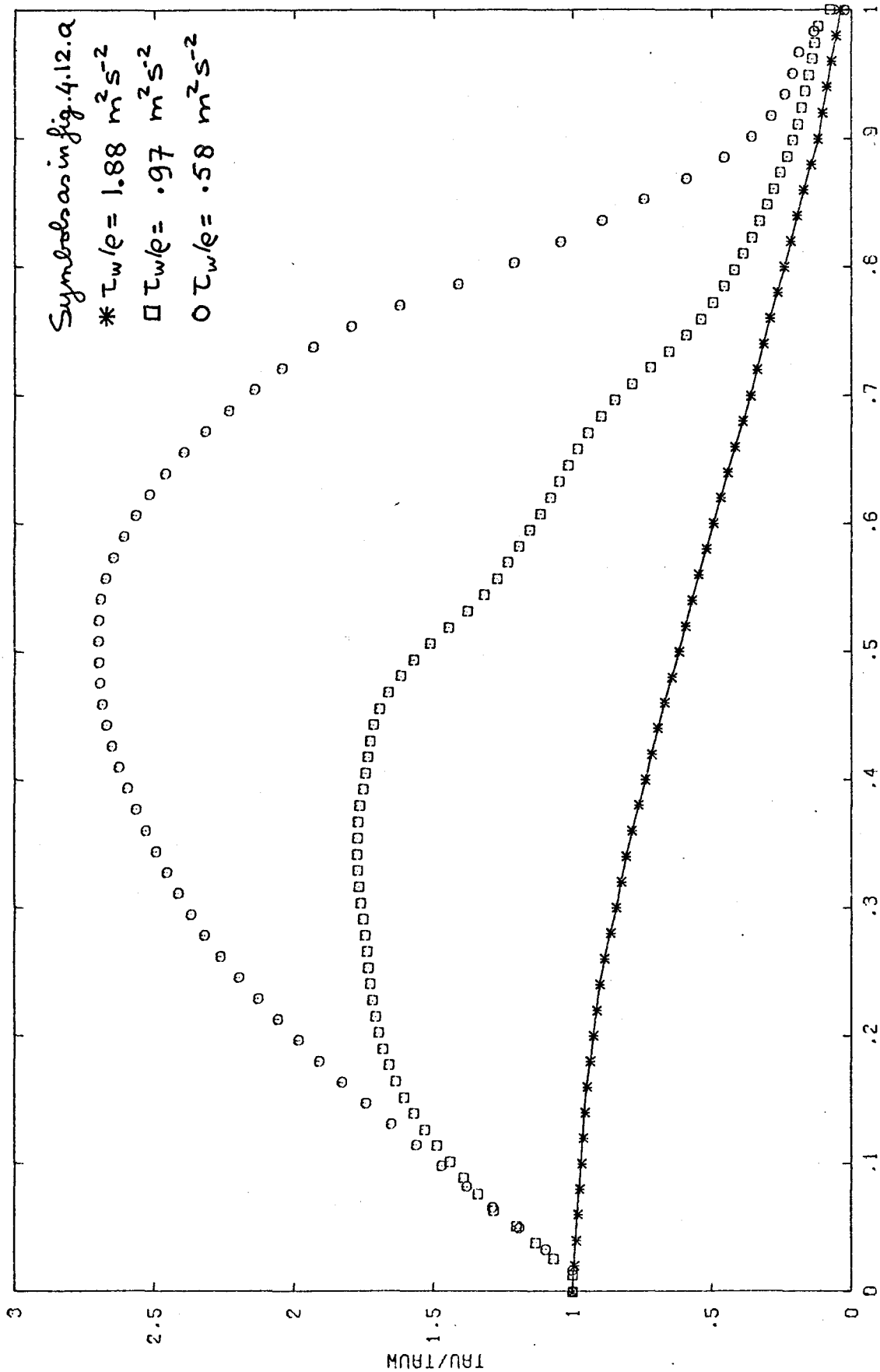


Fig. 4.12.b Turbulent shear stress ($\alpha = 0 \rightarrow -.255$)

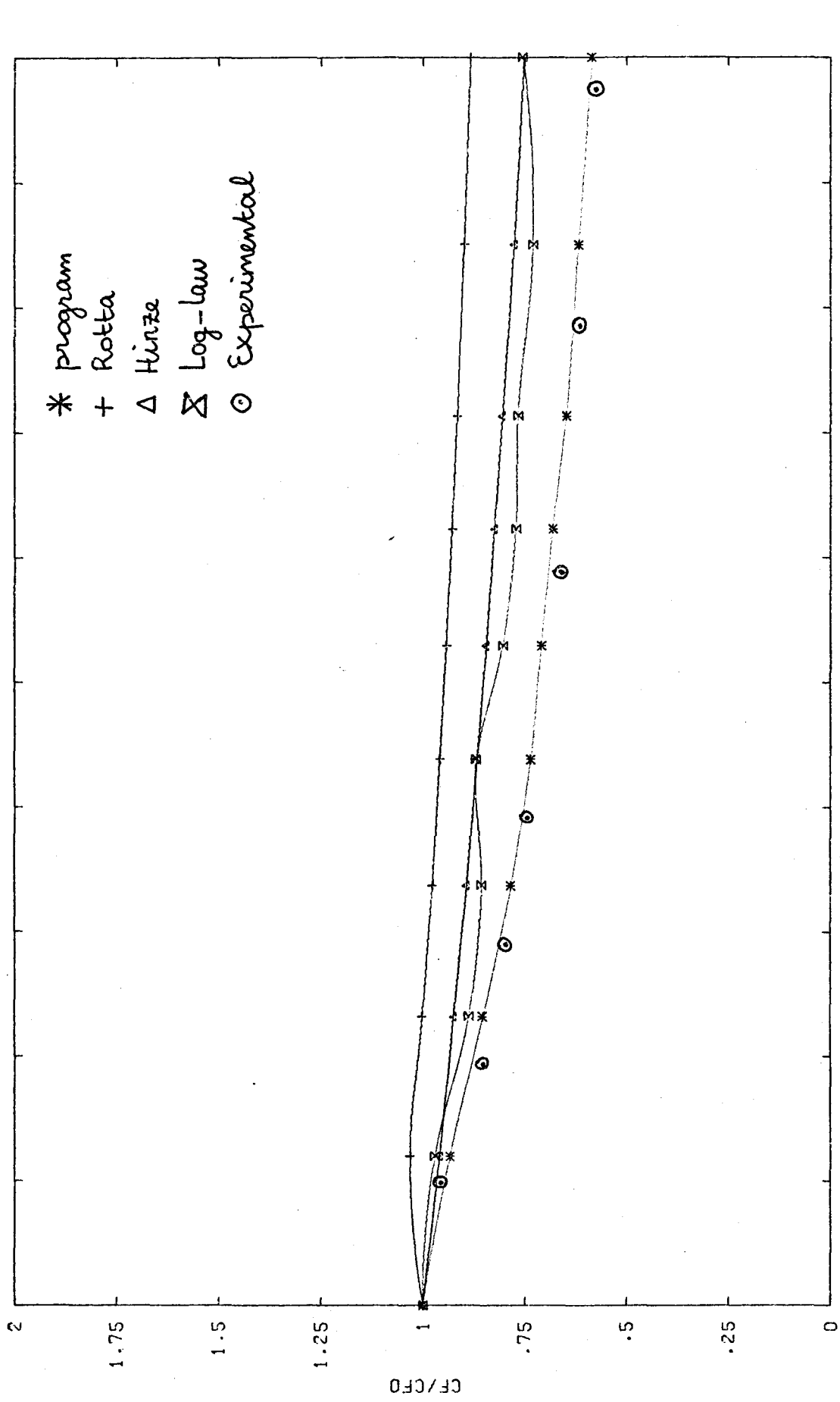


Fig. 4.13. Wall-friction coefficient ($\alpha = 0 \rightarrow -.255$) ($cFe = 26 * 10^{-4}$)

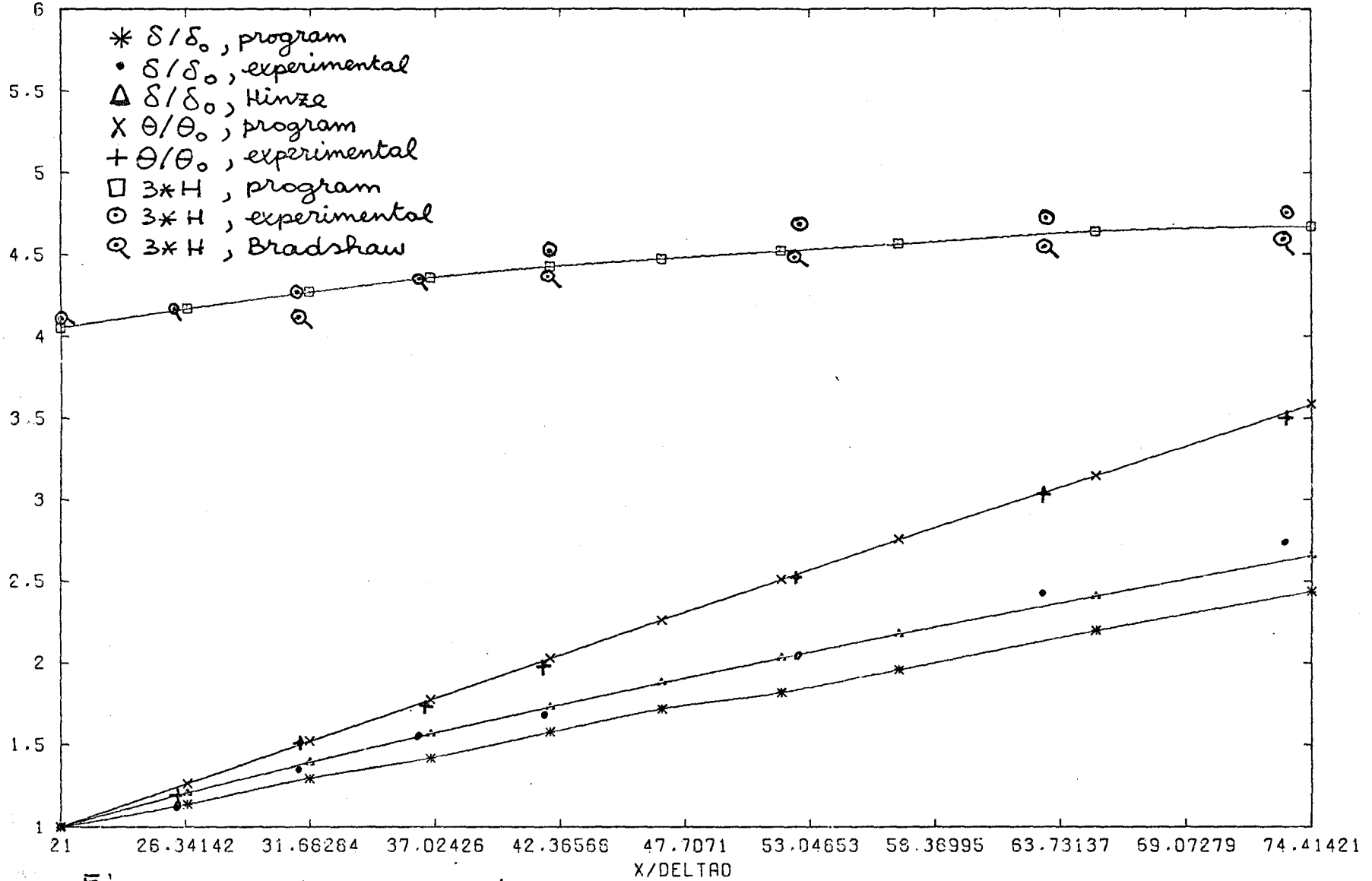


Fig. 4.14 Boundary-layer growth ($\alpha = 0 \rightarrow -.255$) ($\delta_0 = .029m$) ($\theta_0 = .00317m$)

in the plots of the wall-friction coefficient and the boundary-layer growth, figs. 4.16 and 4.17 respectively. The results of the present calculation program agree reasonably well with those of Bradshaw et al., except for the boundary-layer growth, and the wall-friction coefficient in the region just after the removal of the pressure gradient.

Kessels (1977) did some calculations on turbulent boundary layers too. He used a theory developed by Spalding (1972, 1975) which is in fact an application of mixing-length theory. Kessels compared his results with experiments, done in a wind-tunnel, where a turbulent boundary layer under a small favourable pressure gradient had developed. The experiments contain no data of turbulent shear stress profiles. In fact, Kessels's program did not calculate such a profile. So, for the present program, a so-called synthetic shear stress profile had to be generated. Using the measured longitudinal velocity profile, the formula ⊗

$$\tau = \tau_w \left\{ 1 - \frac{y}{\delta} \left(\frac{u}{u_\infty} \right)^2 \right\} \qquad (4.1)$$

provides an appropriate shear stress profile.

⊗ Note: This formula is valid in turbulent boundary layers under zero or small pressure gradients.

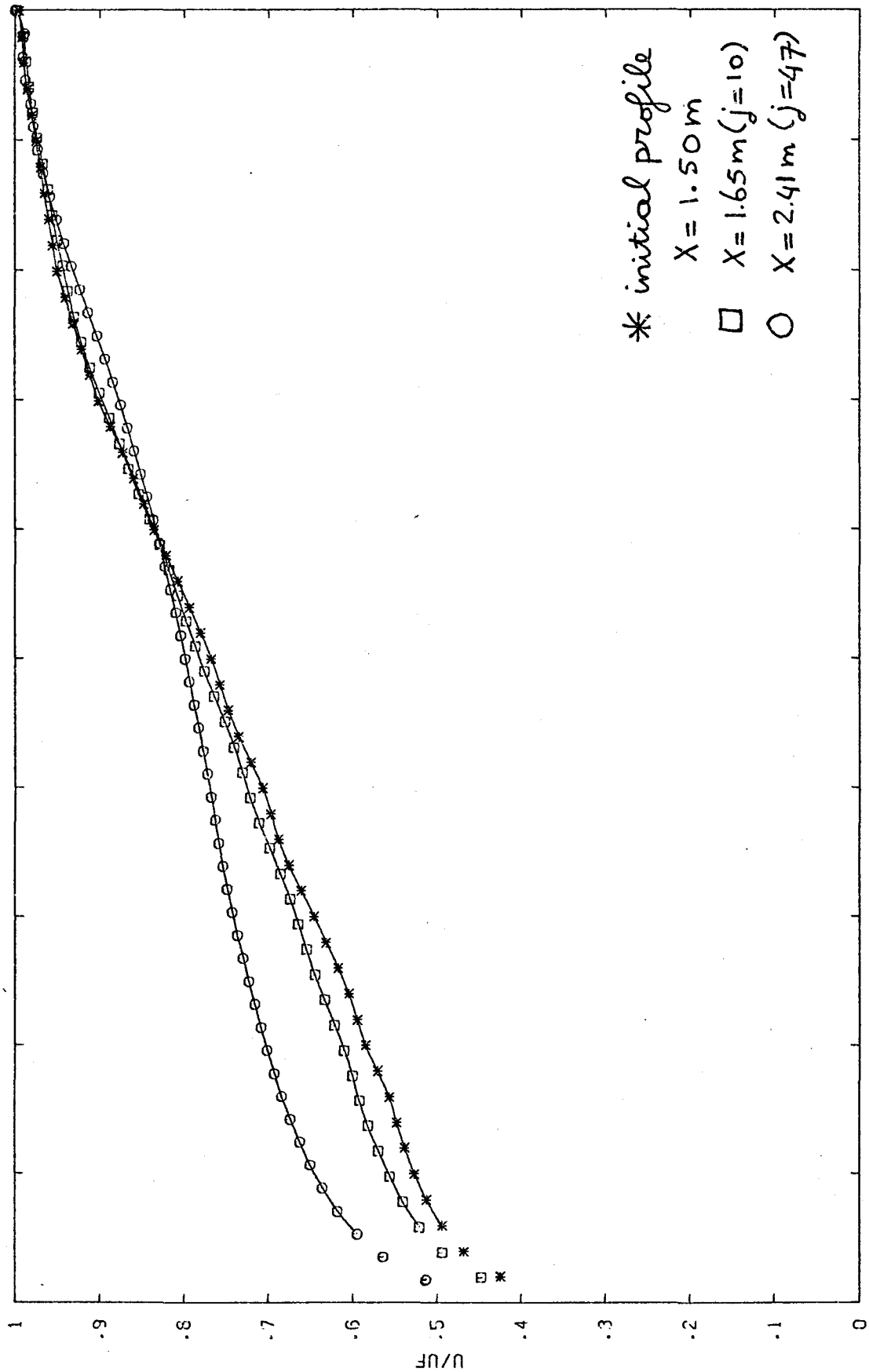


Fig. 4.15.a U-component of mean velocity ($x = -.255 \rightarrow 0$)

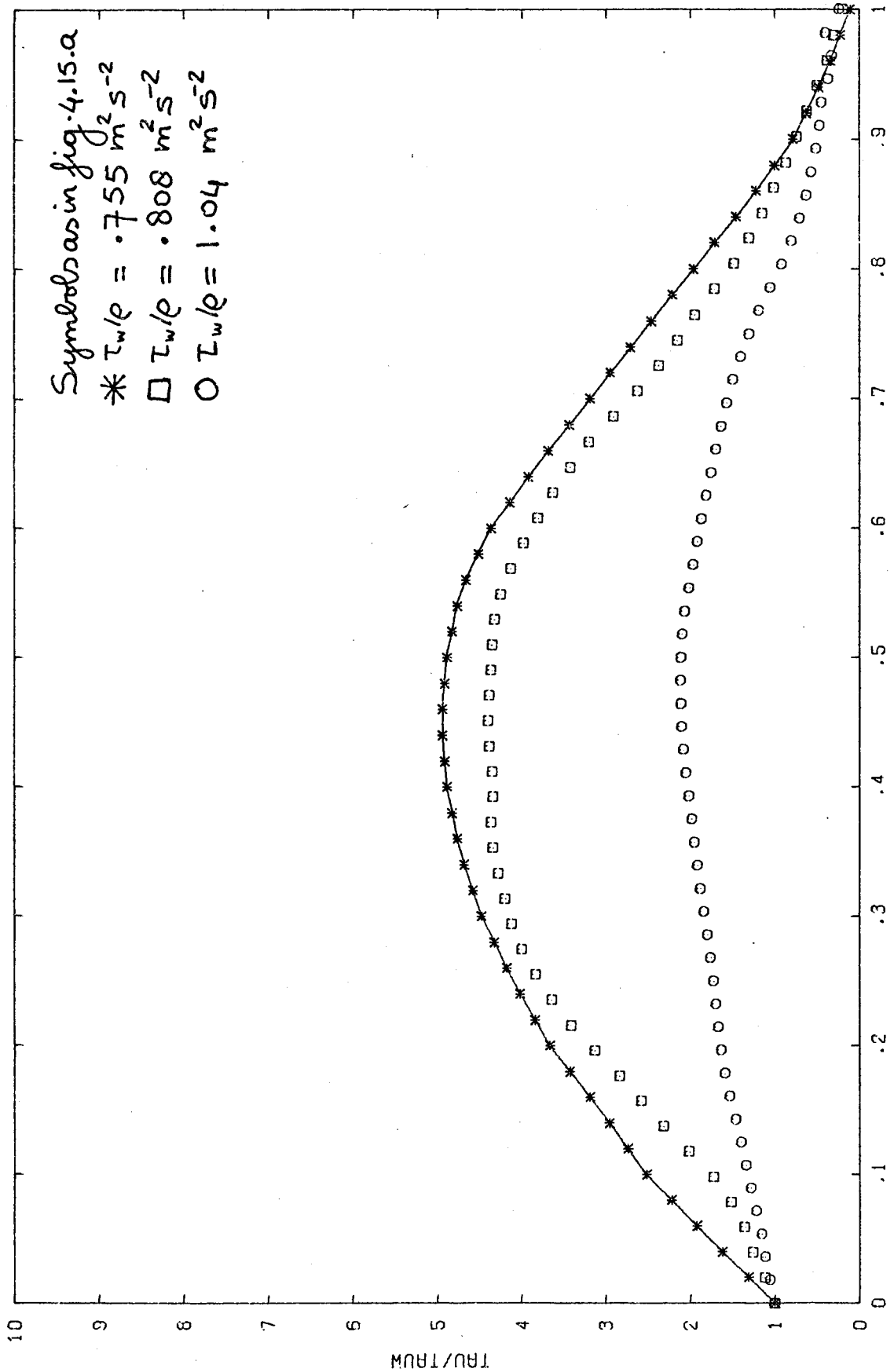


Fig. 4.15.b Turbulent shear stress ($\alpha = -.255 \rightarrow 0$)

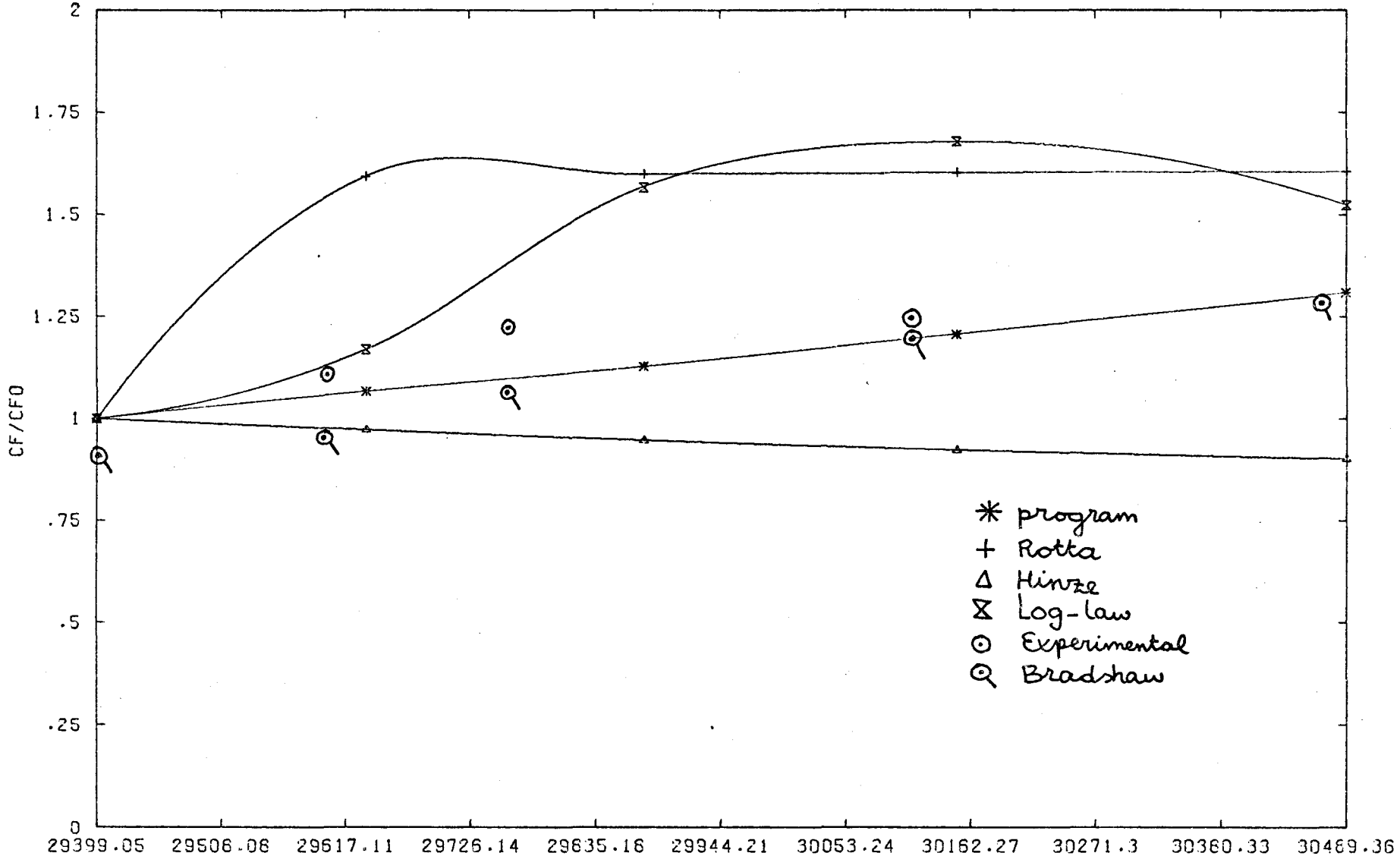


Fig. 4.16 Wall-friction coefficient ($\alpha = -.255 \rightarrow 0$) ($CF0 = 14 * 10^{-4}$)

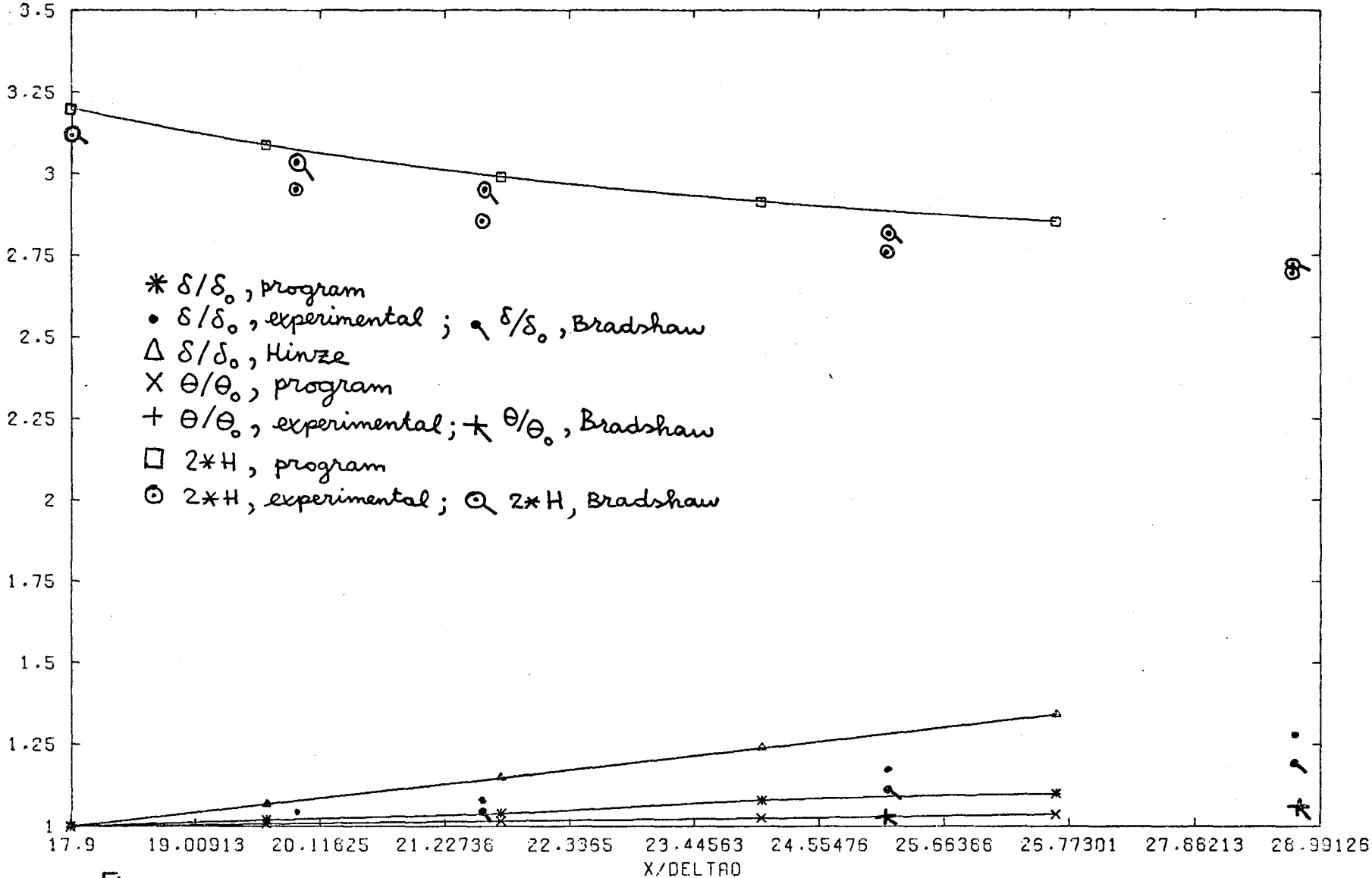


Fig. 4.17 Boundary-layer growth ($\alpha = -.255 \rightarrow 0$) ($\delta_0 = .084$ m) ($\theta_0 = .0127$ m)

The Reynolds number based on the boundary layer thickness, Re_δ, is rather small ($\simeq 2*10^4$). For the present calculation method, the first mesh point had to be taken several y-steps (3, when imax0 = 50) from the surface.
Fig 4.18.b shows that, particularly for a small number ($j \simeq 50$) of x-steps, the computed shear stress profile deviates considerably from the initial one. However, after about 200 x-steps the shear stress profile has become more alike the initial one, fig. 4.19. Beside the fact that Re_δ is small (in which case the procedure may give diverging results (Bradshaw (1967)), the absence of a measured turbulent shear stress profile may be the cause of the deviating behaviour of the shear stress profile in the beginning of the calculations.
Predictions of wall-friction coefficient and boundary-layer growth are shown in figs. 4.20 and 4.21 respectively. The values of the exponents of boundary-layer growth and wall-friction coefficient (Eqs. B.4.a and b) are equal to .75 and -.25 respectively. When assumed to vary as proportional to x^γ, the values of γ for the boundary-layer growth, growth of momentum-deficit thickness and, and wall-friction coefficient calculated by the program are about equal to .60, .65 and -.14 respectively. Kessels's calculations

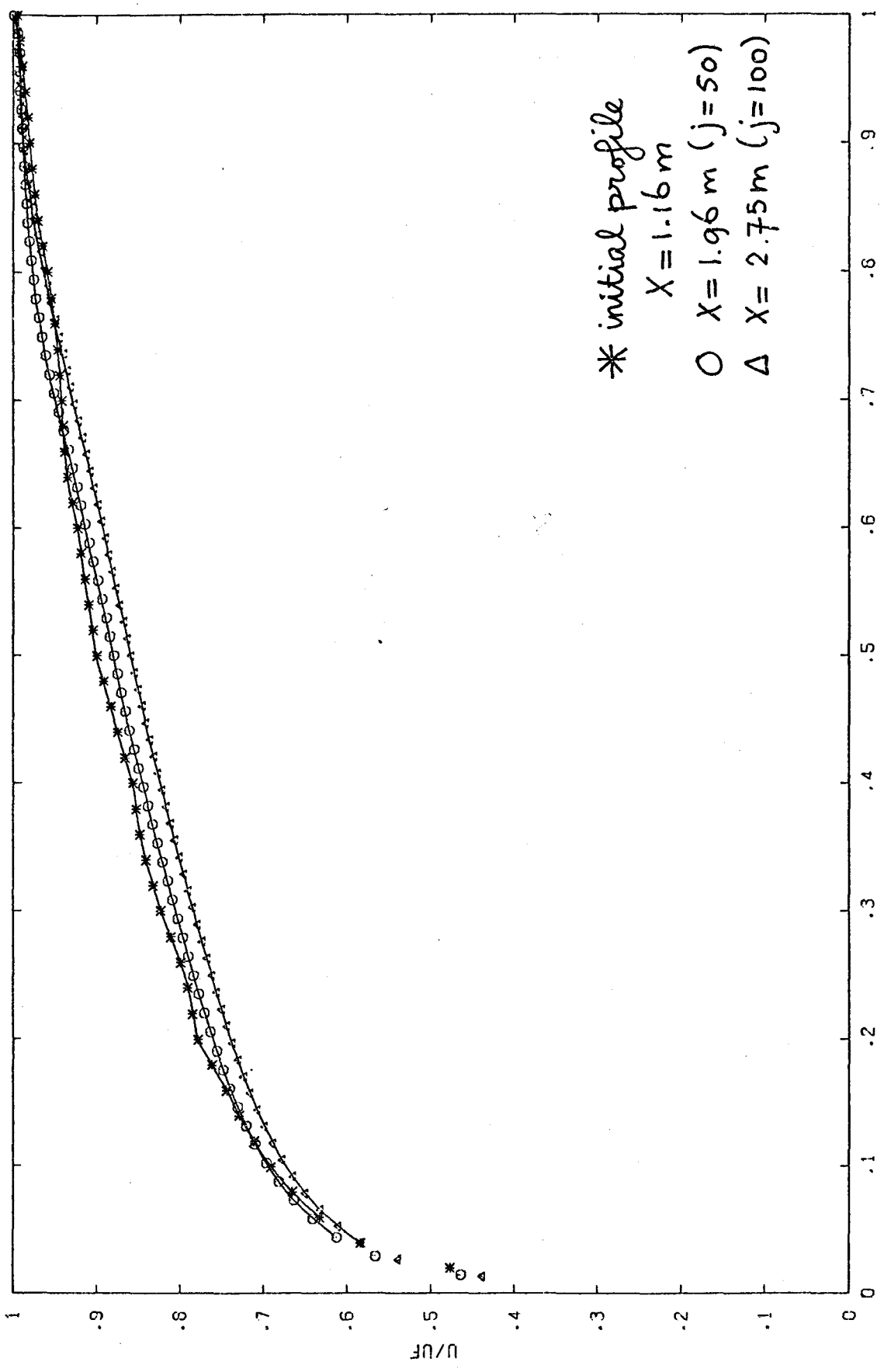


Fig. 4.18.a. U-component of mean velocity (Kamels, 1977)

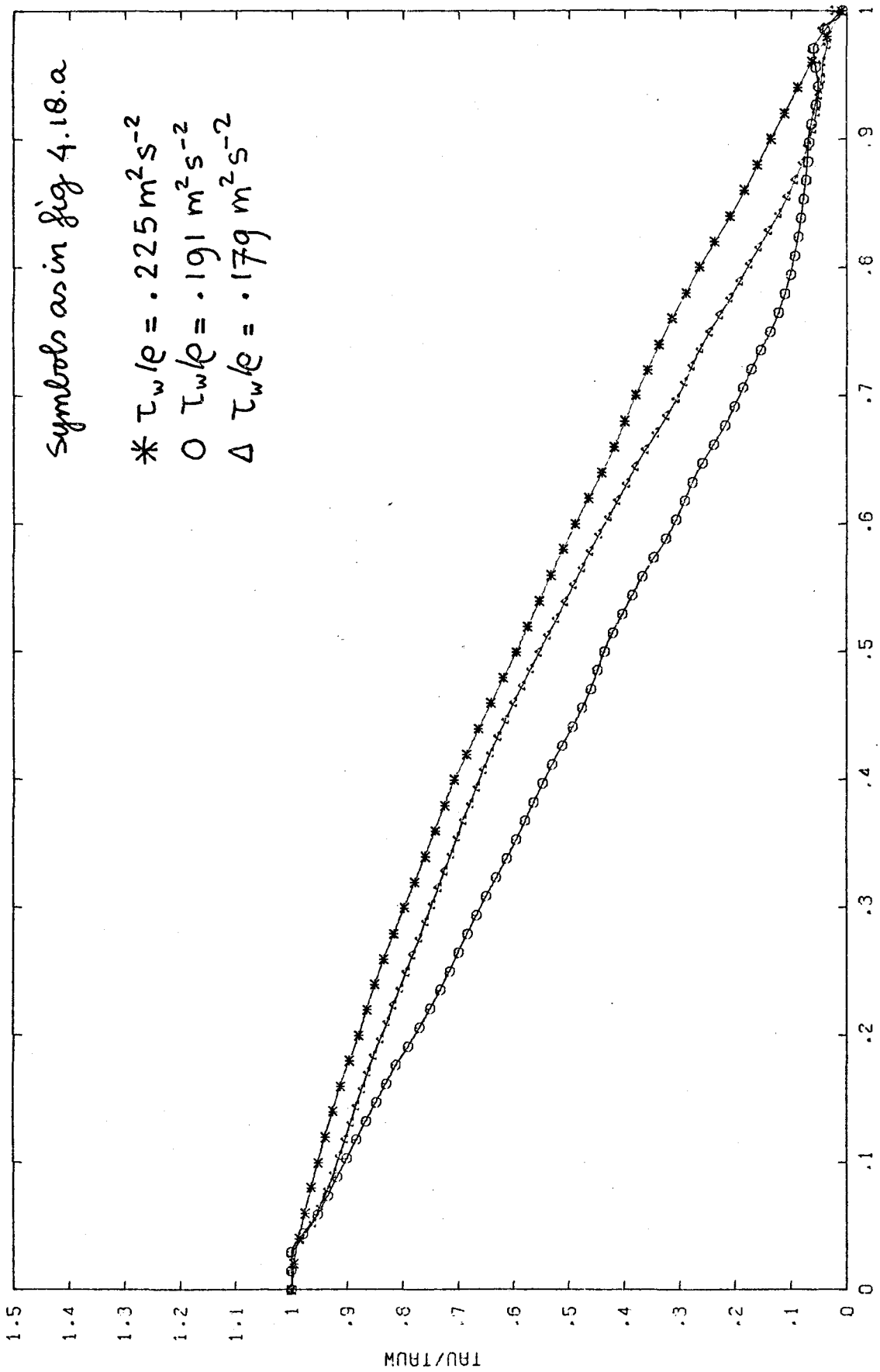


Fig. 4.18.b Turbulent shear stress (Kerrels, 1977)

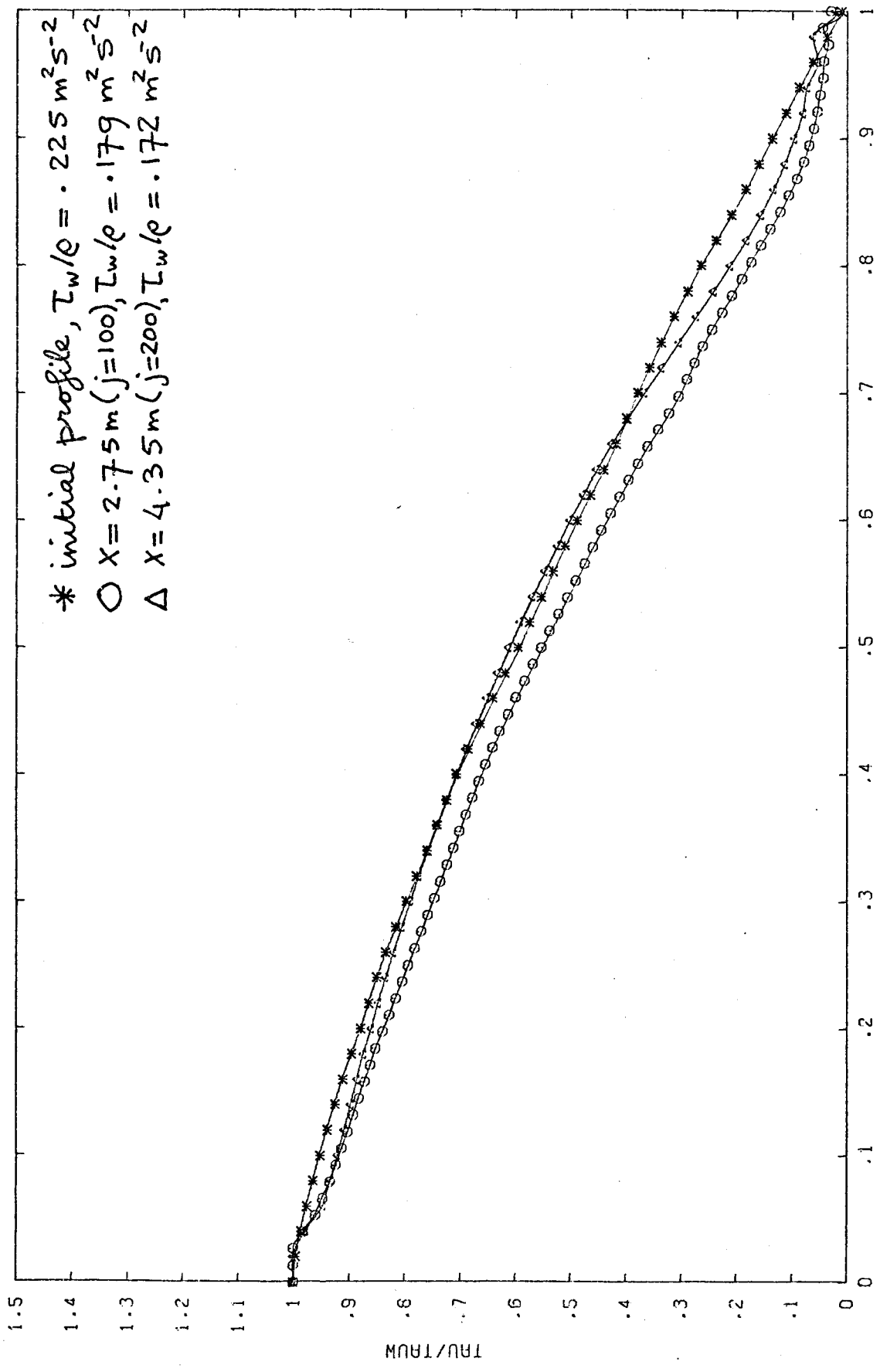


Fig. 4.19 Turbulent shear stress (kemels, 1977)

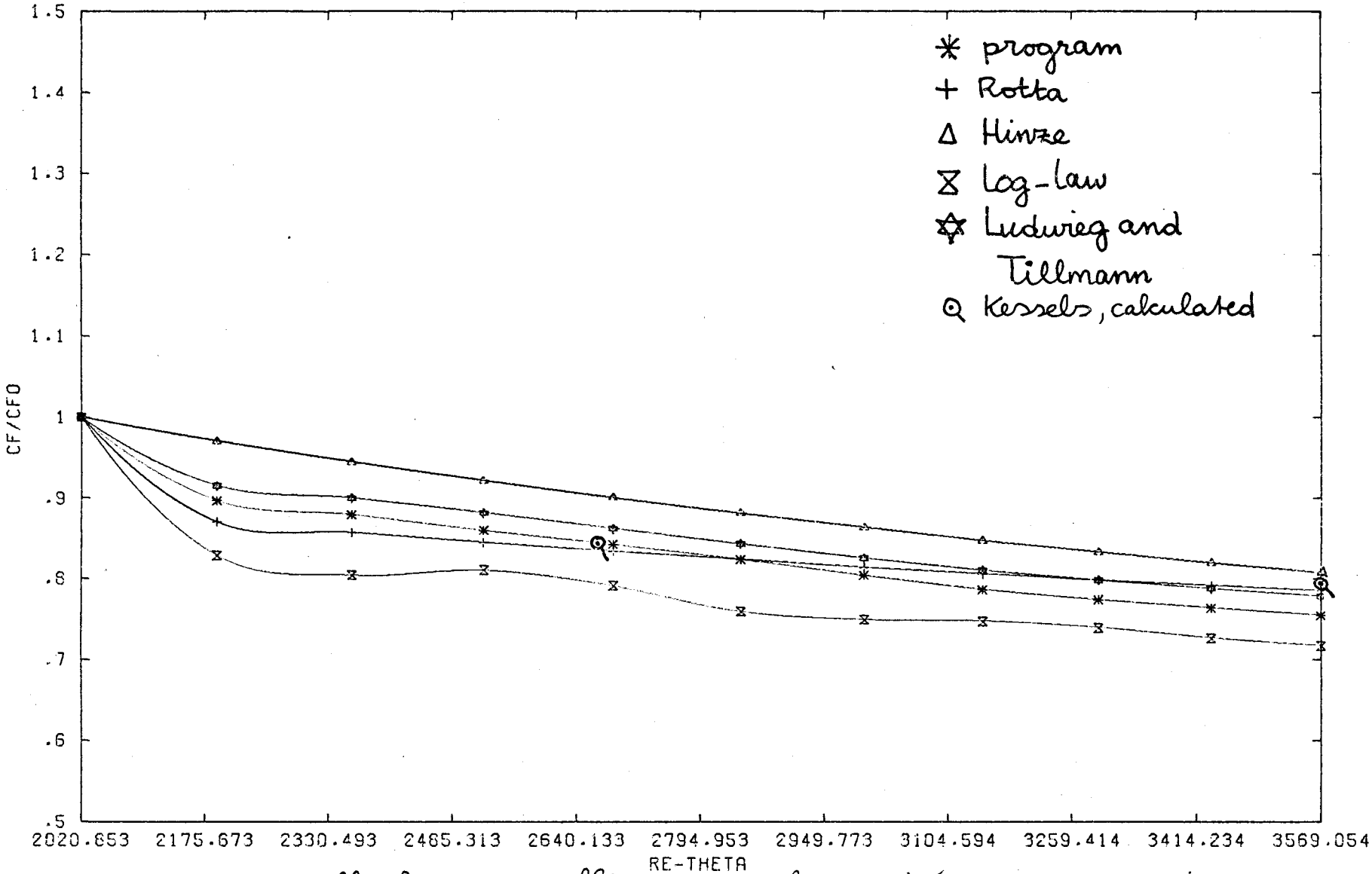


Fig. 4.20 Wall-friction coefficient (Kessels, 1977) ($c_{FO} = 41 * 10^{-4}$)

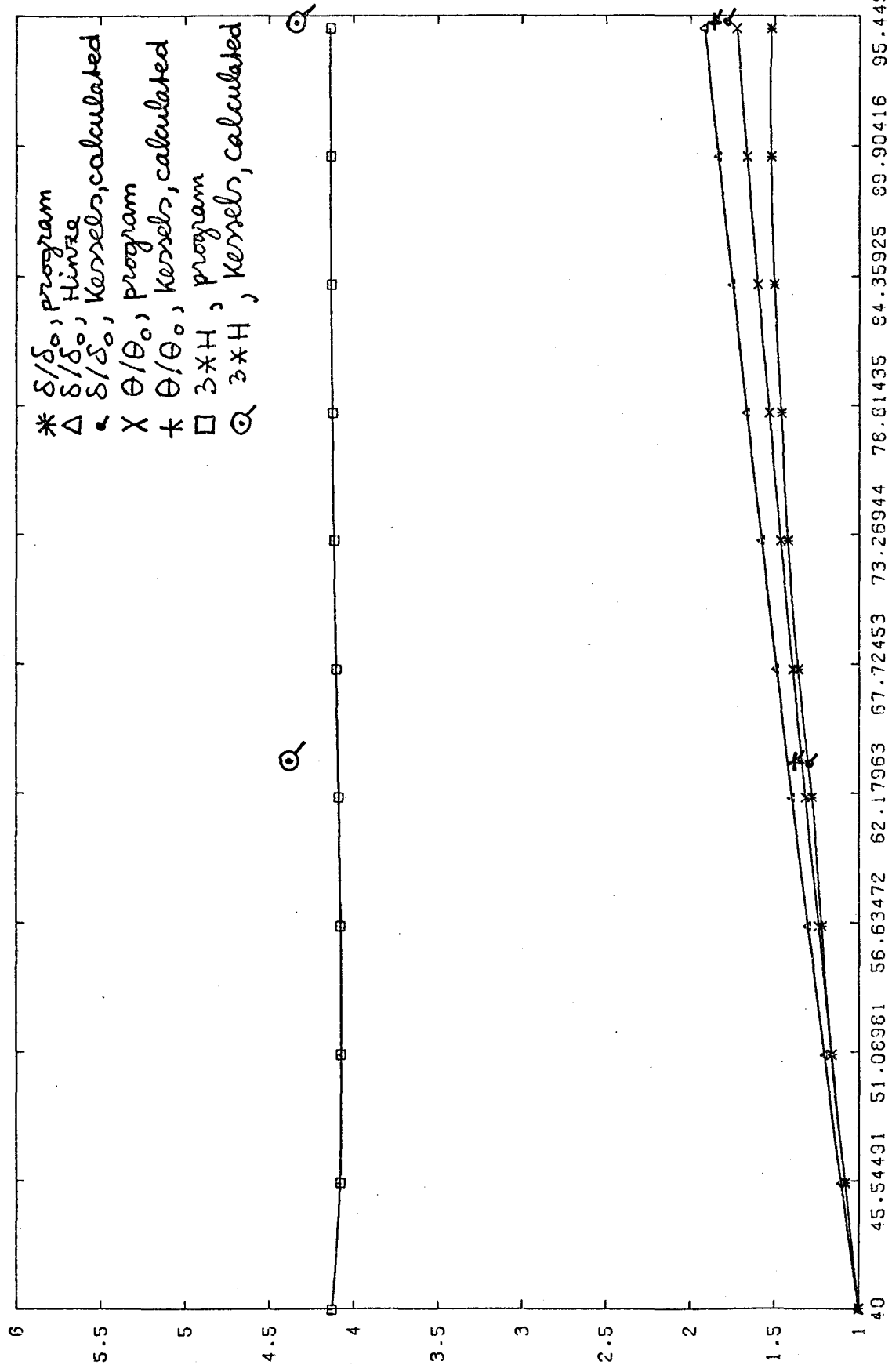


Fig. 4.21 Boundary-layer growth (Kerwels, 1977) ($\delta_0 = .029\,m$, $\Theta_0 = .00302\,m$)

yield a value of γ which is equal to .78 for the boundary layer growth, .70 for the growth of the momentum deficit thickness, and -.13 for the wall-friction coefficient.

Table. 4.1 presents a close comparison of Kessels's calculation results with those of the present program, concerning boundary-layer growth and wall-friction coefficient. Kessels defined δ as the point where U reaches a value of .99 U_∞ instead of .995 U_∞ (as has been used in the present program).

X(m)	$C_{F,K}*10^3$	C_F*10^3	$(\delta/\delta_o)_K$	δ/δ_o	$(\theta/\theta_o)_K$	θ/θ_o	H_K	H
1.84	3.44	3.4	1.35	1.31	1.38	1.34	1.47	1.37
2.84	3.25	3.1	1.80	1.52	1.85	1.77	1.45	1.38
3.84	3.14	2.9	2.37	1.83	2.27	2.15	1.43	1.38
4.84	⊗ 3.05	2.8	2.83	2.17	2.66	2.48	1.42	1.38
5.84	2.99	2.8	3.27	2.48	3.04	2.79	1.42	1.37
6.84	2.93	2.7	3.72	2.76	3.40	3.10	1.41	1.36
7.84	2.89	2.7	4.16	3.03	3.75	3.39	1.41	1.35

Table 4.1 Comparison of the calculation results of Kessels (1977) with those of the present calculation program. Kessels's results are denoted by a subscript K.

⊗ Note: Kessels reports of an experimental value which is equal to $3.29*10^{-3}$ at $x = 4.84$ m.

4.2.2 Conclusions about the test-results.

Summarizing the results, one can conclude that the present calculation program predicts the behaviour of a turbulent boundary layer along a smooth surface satisfactory.
The results agree well with Bradshaw's (1967) and Kessels's (1977) calculation results, except for the prediction of the growth of the boundary-layer thickness, in which case the present calculation program computes too small values.
When the calculated turbulent shear stress profiles can be compared with appropriate experimental data or calculation results, the computer program possibly can be improved.
The so-called momentum-error (ME) (see appendix B, Eq.(B.131)), amounts to a value of about 2 % after a 100 x-steps from the starting-position (Bradshaw (1967) reports a value of 1 %.)

In the next section the prediction of the response of a turbulent boundary layer to a step change in surface roughness will be treated

4.3 The step-change in surface roughness.

4.3.1 Introduction

In this section, the results are presented and discussed for the calculation of the response of a turbulent boundary layer under zero pressure gradient to a step-change in surface roughness. These results are compared with the experimental results of Antonia and Luxton (1971, 1972), who investigated both smooth to rough and rough to smooth transitions in a wind-tunnel.
As pointed out in appendix C, the change in surface roughness will be introduced in the calculation program by altering the so-called 'law of the wall'-equation

4.3.2 The roughness-configuration.

The calculated results were compared with the experimental data of Antonia and Luxton. Fig. 4.22 shows the roughness-configuration used by these authors. At the first step in surface roughness (A) the turbulent boundary layer above the smooth surface was fully developed. The roughness is of the so-called 'k' type (see appendix C) with a pitch to height ratio of 4. From their experimental data, Antonia

and Luxton (1971) concluded that the flow at a distance of about 4′ (≃ 1.22 m) from A, had approached a new self-preserving state. So, the jump from rough to smooth was taken at B. At C, the end of the test-section of their wind-tunnel, the flow was still found to be far from equilibrium (Antonia and Luxton, 1972)

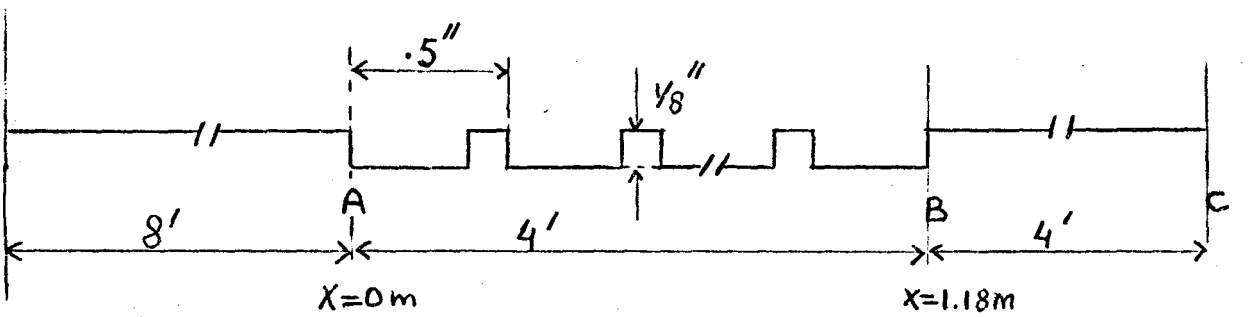


Fig. 4.22 The roughness-configuration used by Antonia and Luxton.

To compare the computational results with the findings of Antonia and Luxton a similar roughness-configuration as described above has been used in the calculation program.

4.3.3 The calculations of the program

First of all, the effect of a single step in surface roughness (at position A in fig. 4.22) has been

investigated.
Special attention is paid to
a the relaxation-proces of the profiles of the U-component of mean velocity and turbulent shear stress;
b the growth of the internal boundary layer;
c the distance needed to achieve a new equilibrium state, which has to be clearly defined.

Thereafter, the position of point B in fig. 4.22 has been fixed, and the response of a turbulent boundary layer to a rough to smooth step in surface roughness has been investigated.

In the 'law of the wall'-equation for the rough surface (appendix C, Eq.(C.10)) the constant was taken equal to 3.1, as can be derived from the experimental data of Antonia and Luxton (1971, part I, p.735).
The zero-pressure gradient profiles of Klebanoff (1955) have been used as input, after scaling. These profiles have a similar shape as measured by Antonia and Luxton (1971).

4.3.4 The calculation-results of the smooth to rough jump.

In figs. 4.23 to 4.27 the computational results for a smooth to rough step-change in surface roughness are presented.
The oscillations occurring in the profiles of mean velocity and Reynolds shear stress, figs. 4.23, clearly indicate how the disturbance, caused by the change in surface-condition, propagates through the boundary layer. At about $X = 2.14$ m, the oscillations have damped out. The oscillations occur at a position, which can be identified with the edge of the internal layer (fig. 4.27); this edge has been determined using the 'knee-point' method of Antonia and Luxton (1971). When plotted versus the square root of the distance, y, above the surface, the mean velocity profile exhibits, in the region near the step, linear trends inside the internal layer and outside the internal layer, which are different. The intersection of the two straight line portions, fig. 4.27, has been shown, by Antonia and Luxton, to be closely related to the edge of the internal layer. Table 4.2 presents the values of δ_i, the edge of the internal layer.

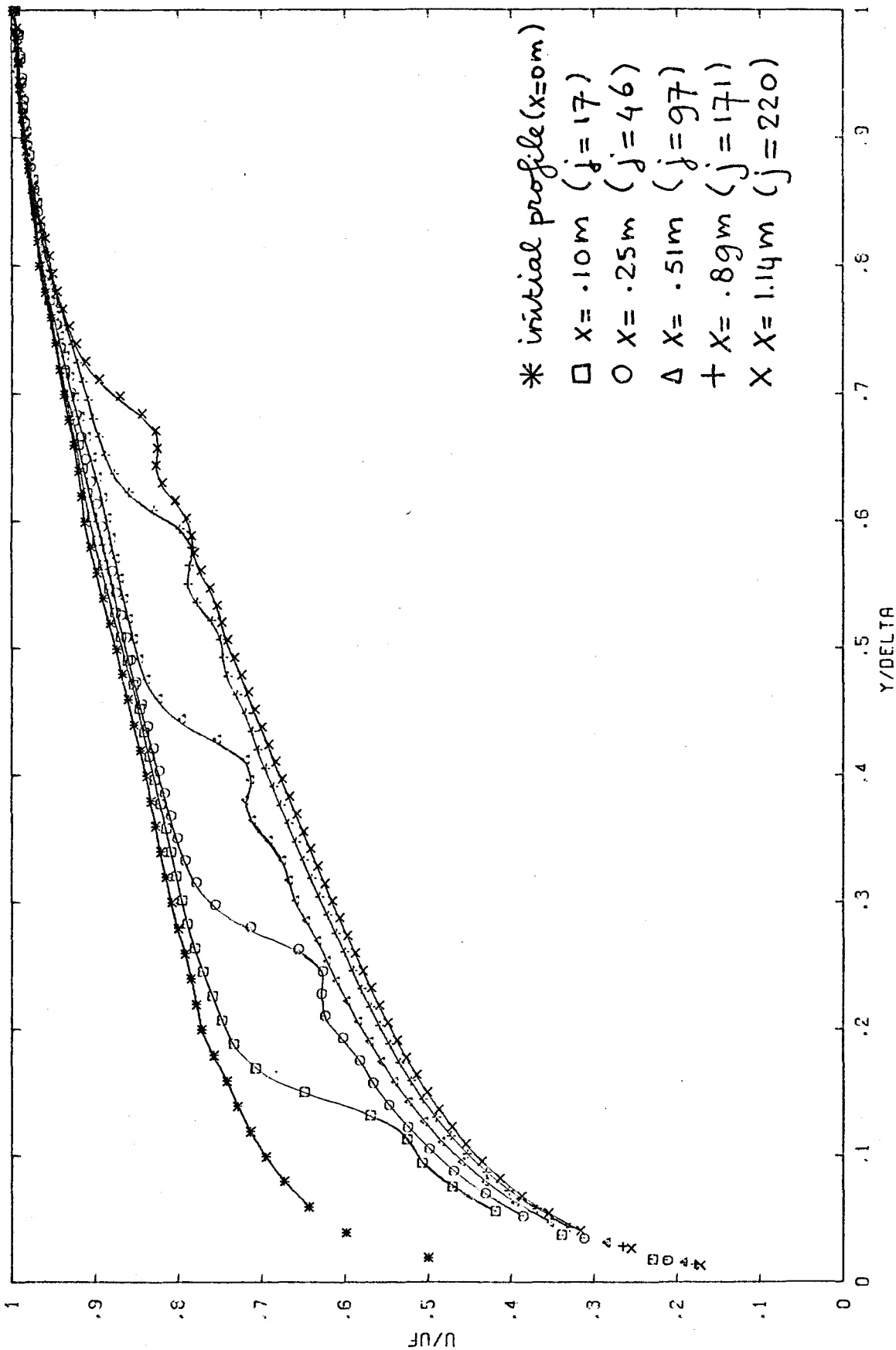


Fig. 4.23.a U-component of mean velocity (s→r)

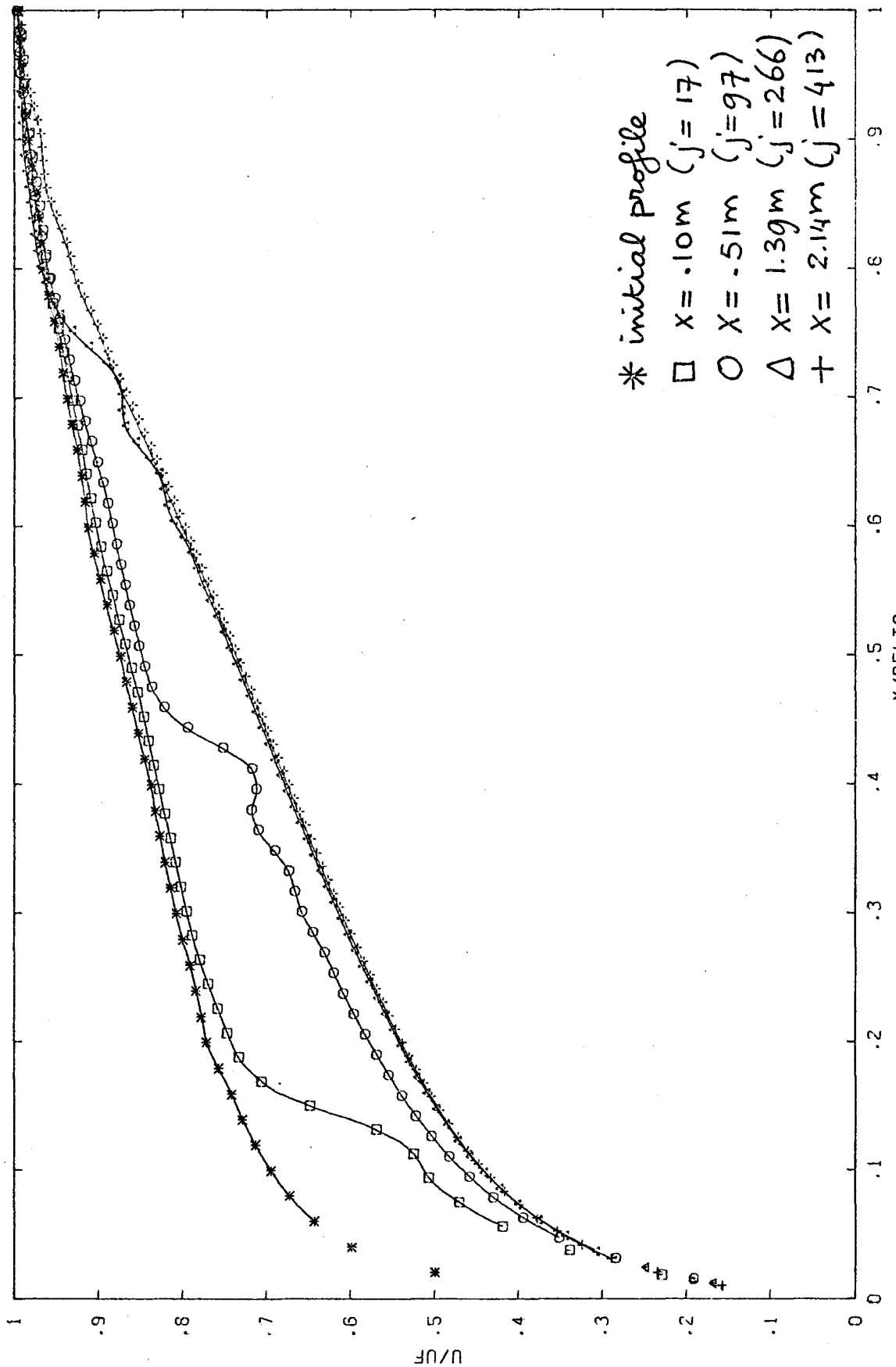


Fig. 4.23.b U-component of mean velocity (s→r)

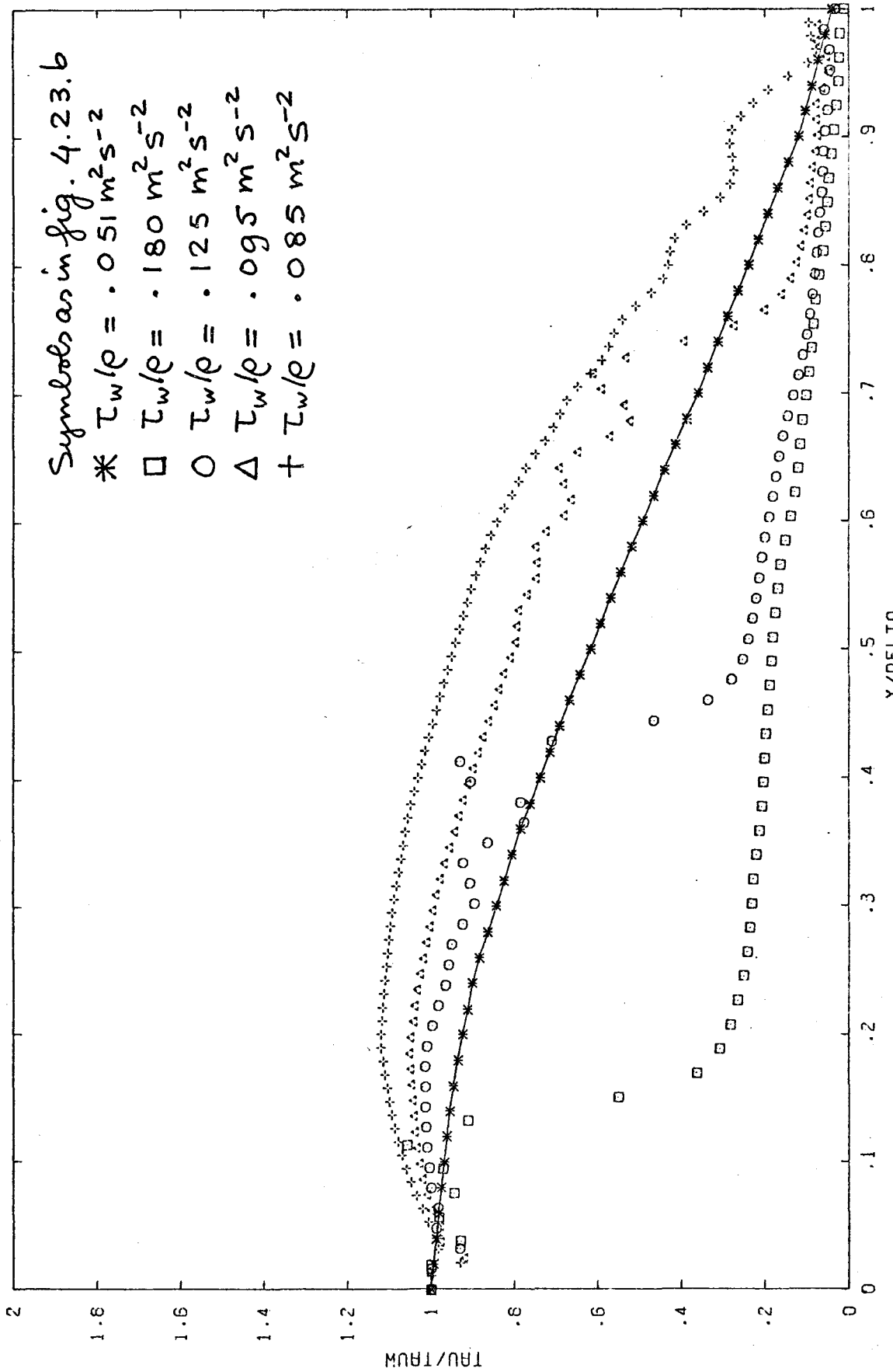


Fig. 4.23.c Turbulent shear stress ($s \rightarrow r$)

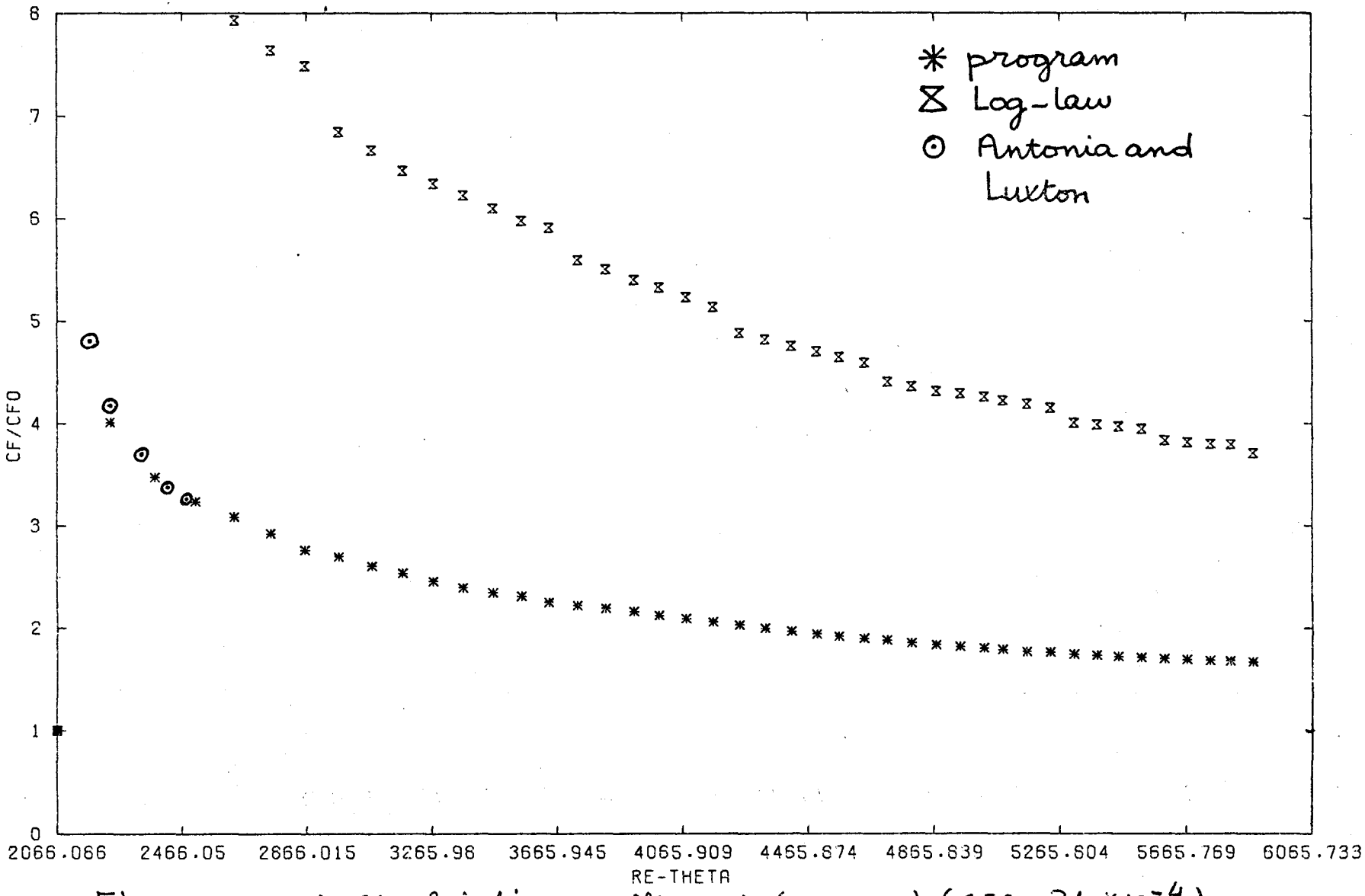


Fig. 4.24 Wall-friction coefficient (s → r) (CF0 = 34 * 10^{-4})

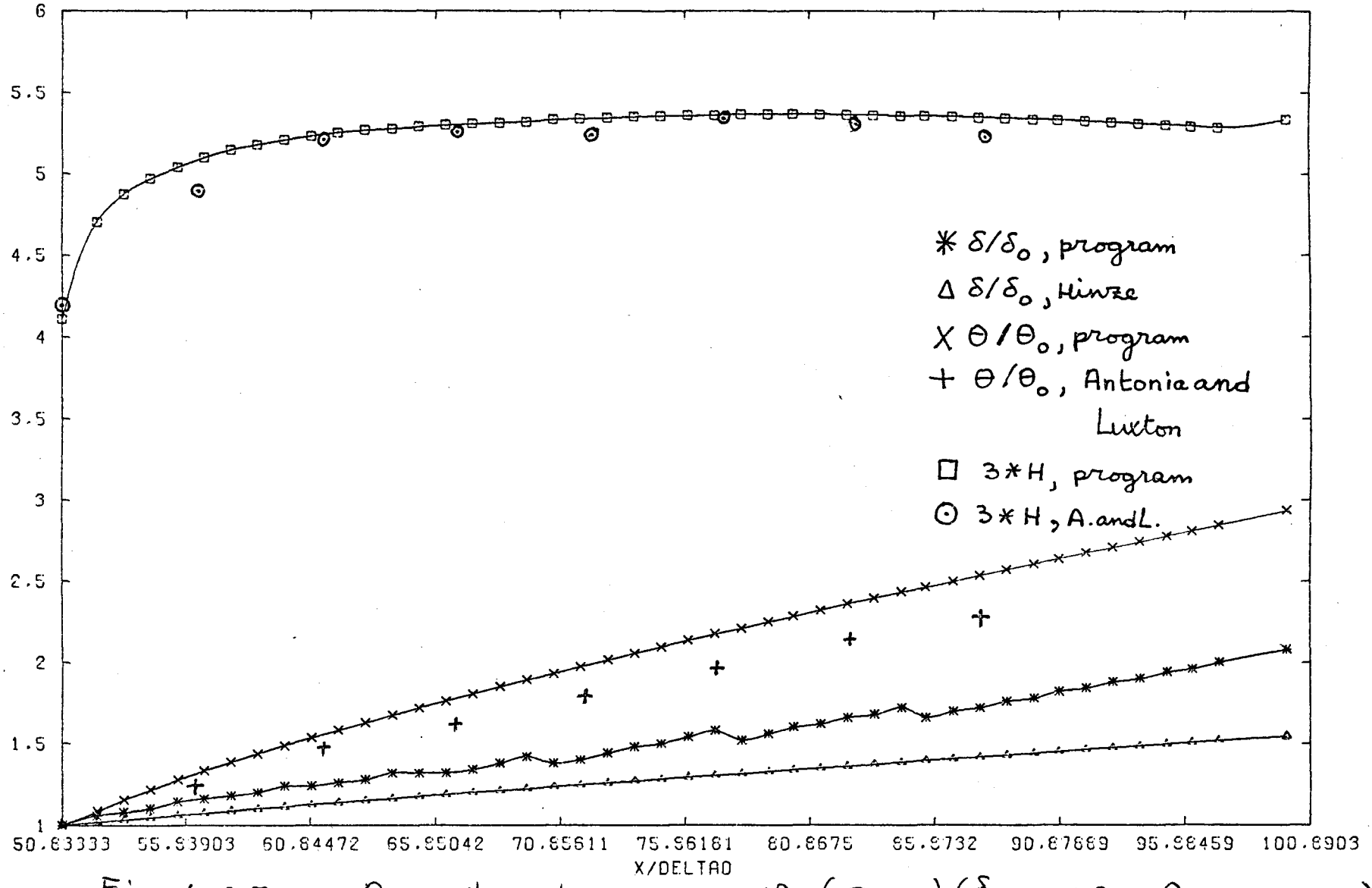


Fig. 4.25. Boundary-layer growth ($s \to r$) ($\delta_0 = .048m$, $\theta_0 = .00527m$)

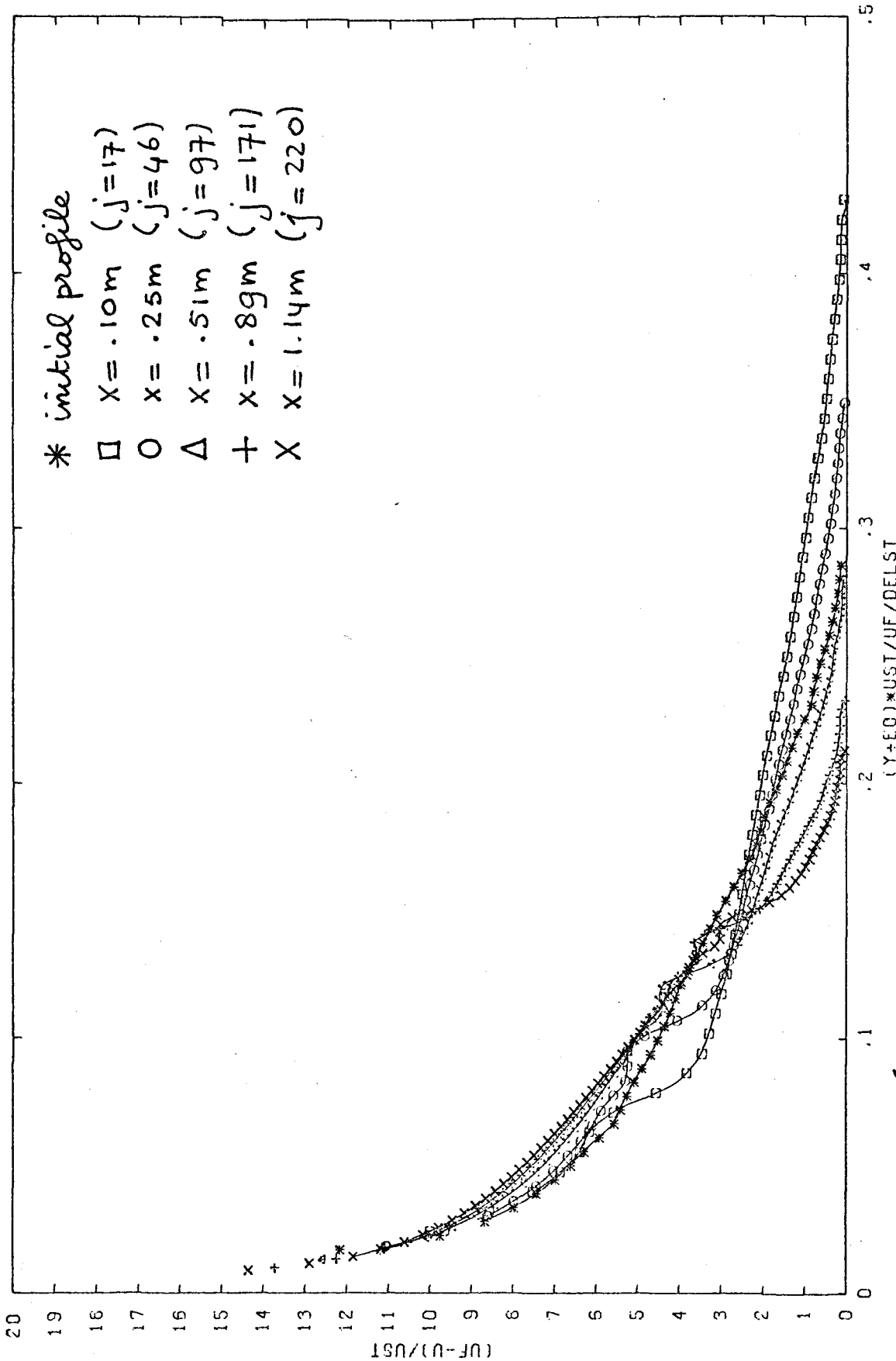


Fig. 4.26 Mean velocity defect profiles (S→R)

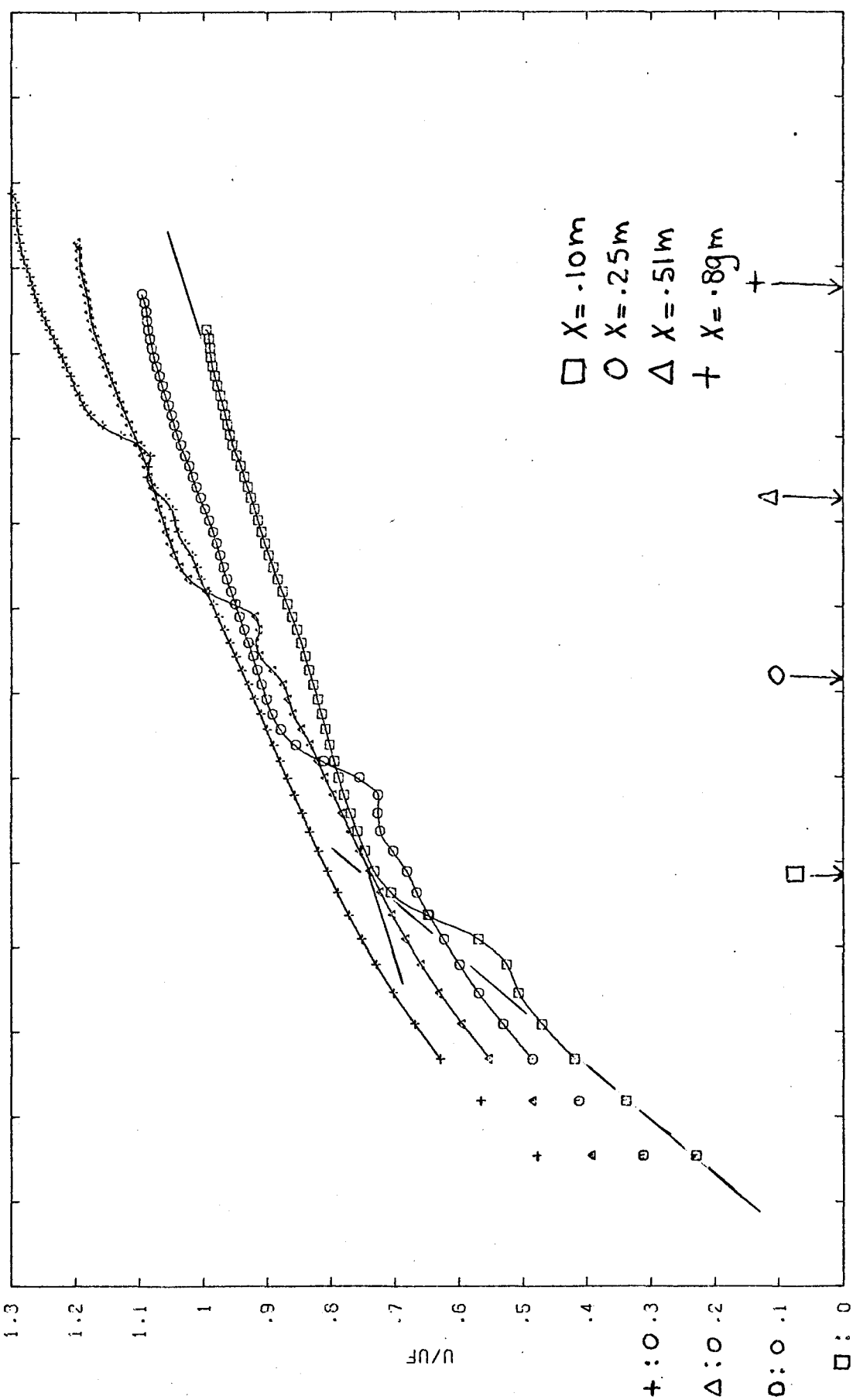


Fig. 4.27 Mean velocity vs. $y^{.5}$ (S→r) (edge of internal boundary layer is indicated by an arrow) (note shift in origin)

x(m)	δ_i(m)	$.046\ x^{.72}$ (m) (A. and L.)
.10	.010	.009
.25	.021	.017
.51	.035	.028
.89	.055	.042

Table 4.2 The edge, δ_i, of the internal boundary layer. Antonia and Luxton's findings ($\delta_i = .046\ x^{.72}$, when x and δ_i in m) are also tabulated.

When regarded proportional to x^{γ}, the value of γ for δ_i, predicted by the program, is equal to .78, which exceeds the value found by Antonia and Luxton by a small amount.

Comparing fig. 4.23.c with figs. 12.b and 14 in part I of Antonia and Luxton's report (1971), it can be clearly noticed that there is a considerable difference in shapes between the measured shear stress profiles of Antonia and Luxton and those calculated by the program, at similar x-positions, in particular near the surface.

In fig. 4.24 the wall-friction results are compared with the experimental data of Antonia and Luxton (1971, part I, fig. 5), who used a so-called 'form drag' method to measure the wall-shear stress. The method has been described by Perry et al. (1969). The agreement between the results is very good.
The log-law values lie far above the calculation results; this may be caused by the absence of energy-equilibrium, inside the internal boundary layer, after the step. In this case a log-law velocity profile may not be present. The sudden rise in wall-friction coefficient at the start of the roughness-change, is followed by a fairly rapid fall, towards a value of about 2 times that above the smooth wall, which occurs within a distance of nearly 25 boundary-layer thicknesses from the step; Antonia and Luxton report a value of 4 boundary-layer thicknesses for the wall-friction coefficient to relaxate. The shape parameter $H = \delta^*/\theta$, plotted in fig. 4.25, increases from a value of 1.37 on the smooth surface to a value of about 1.8, at $X \simeq 1.2$ m, and maintains this value further downstream. In fig. 4.25, the growth of the momentum-deficit thickness, found by Antonia and Luxton is compared with the calculation results of the present program.

The value of $2\,d\theta/dx$ ⊗, for the calculation results, is equal to .0095, which is considerably larger than .0064, the value of the wall-friction coefficient at a distance from the step where this quantity has attained a nearly constant value.
The boundary-layer thickness shows some oscillations, which must be caused by those occurring in the mean velocity profiles.

Mean velocity profiles are plotted in the form $(U_\infty - U)/u_*$, as a function of $(y+\epsilon)\,u_*/\delta^*/U_\infty$ in fig. 4.26. When compared with fig. 7 in part I of Antonia and Luxton, the results show the same trend and lend support to the supposition that at $x \simeq .89$m the mean flow has attained a self-preserving state.

Summarizing the obtained results, one can conclude that it is hard to predict a turbulent shear stress-profile for a boundary layer that takes a step-change in surface roughness.
The distance needed to adapt completely to the new surface-condition seems to be different for the several flow quantities, such as H, c_f (which quantities seem to adjust within a similar distance), the mean velocity profile and the turbulent shear stress profile.

⊗ Note: According to the von Kármán integral momentum-equation (appendix B, Eq. (B.11) $2\,d\theta/dx = c_F$ in a zero-pressure gradient turbulent boundary layer.

In particular, for the profile of turbulent shear stress it is difficult to determine whether this profile has attained a self-preserving shape. To investigate this, a clearly defined criterium of what is meant by 'full adaptation' of the flow to the new surface-configuration, has to be given.

For the present results, this has not been done yet, but regarding the results of the profiles of the longitudinal component of mean velocity and Reynolds shear stress, it seems reasonable to assume that at $x = 2.14$ m the profiles have completely adjusted to the new surface-condition, since at that position the oscillations have dissappeared (by approximation). Antonia and Luxton took a distance of about 1.2 m for the flow to have fully adapted.

To investigate the response of a turbulent boundary layer, which has developed above a rough surface, the profiles of the U-component of mean velocity and turbulent shear stress at $x = 2.14$ m have been taken as input. The computational results are presented in section 4.3.5

4.3.5 The calculation - results of the rough to smooth step.

As input profiles, the profiles at $X = 2.14$ m (See section 4.3.4, figs. 4.23.b and 4.23.c) have been taken.
Figs. 4.28 to 4.32 present the results of the calculations.
The profiles of the U-component of mean velocity and the turbulent shear stress again show oscillations (figs. 4.28), which propagate through the boundary layer. After about 3.50 m from the step, the oscillations have nearly damped out.
In fig. 4.28.b, the oscillating behaviour of the shear stress near the wall, has disappeared above the smooth surface far away from the step. The results of the measured profiles of turbulent shear stress of Antonia and Luxton (part II, fig. 9) now compare better with the present calculation results, than in section 4.3.4 (smooth to rough).
After an undershoot, the wall-friction coefficient (fig. 4.29) relaxes towards an approximately constant value, within a distance of about 25 boundary-layer thicknesses from the step. The results found by Antonia and Luxton (1972) lie considerably lower, and reach an approximately constant value within 10 boundary-layer

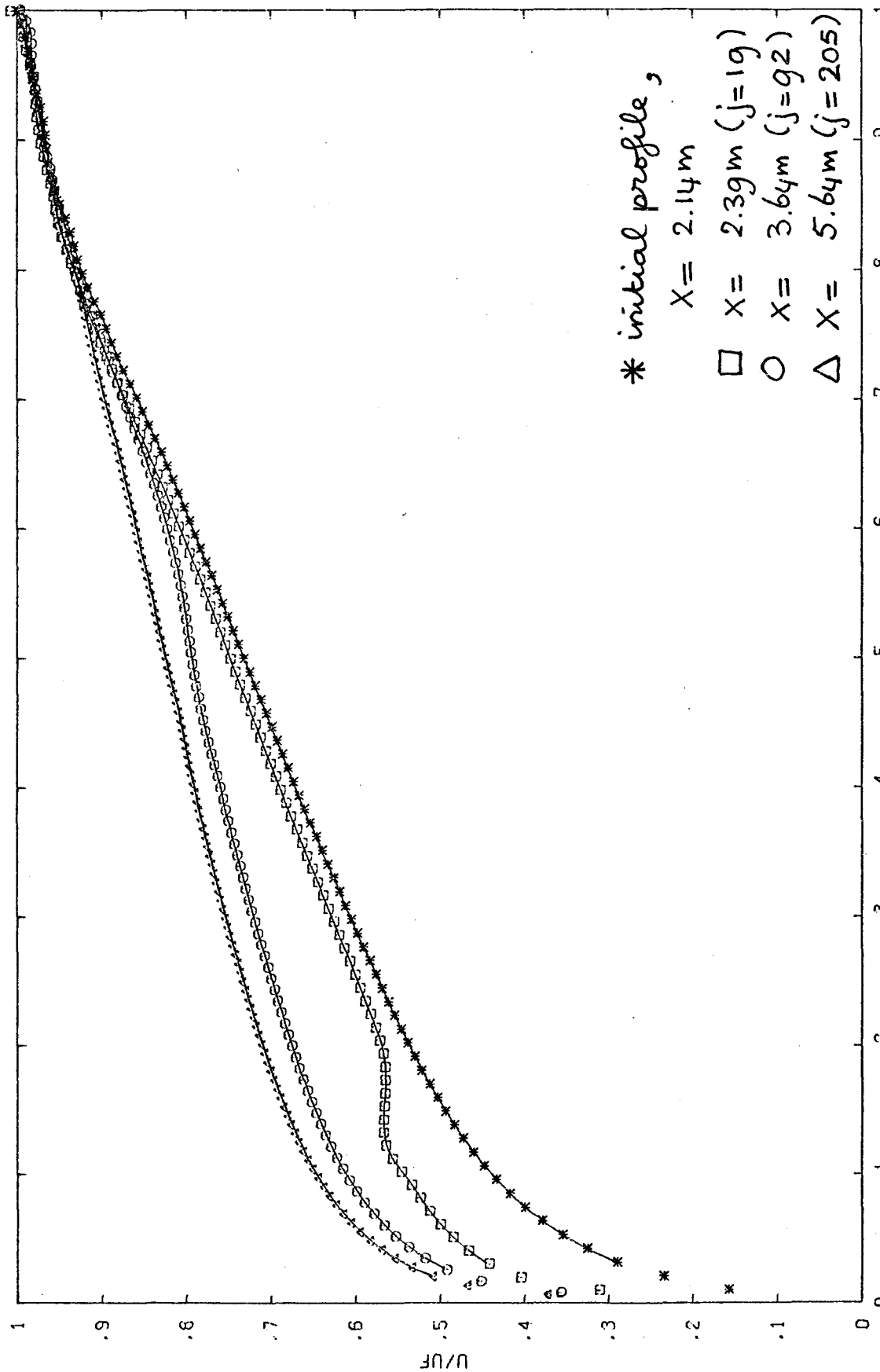


Fig. 4.28.a U-component of mean velocity (r → s)

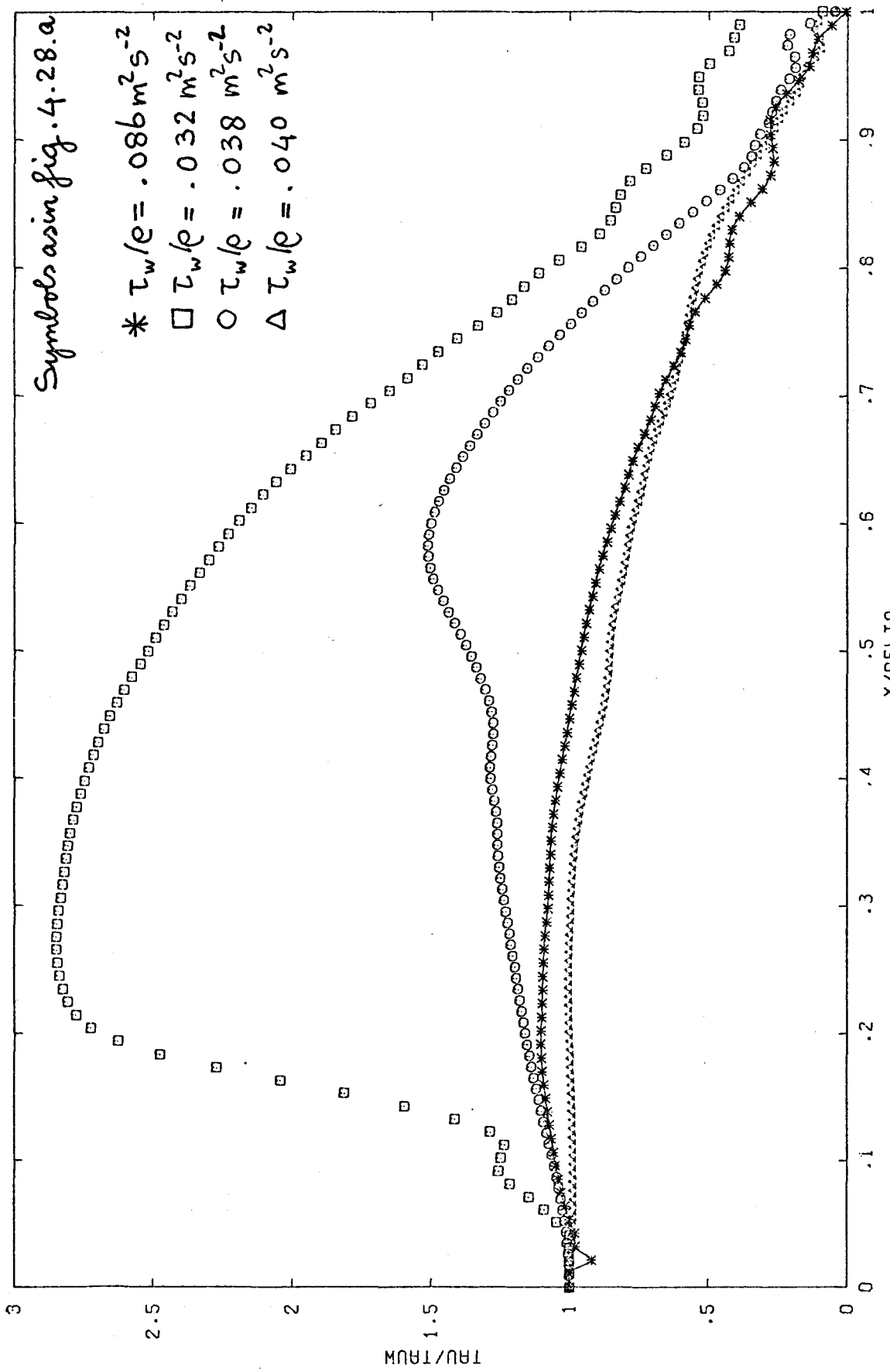


Fig. 4.28.b Turbulent shear stress ($r \to s$)

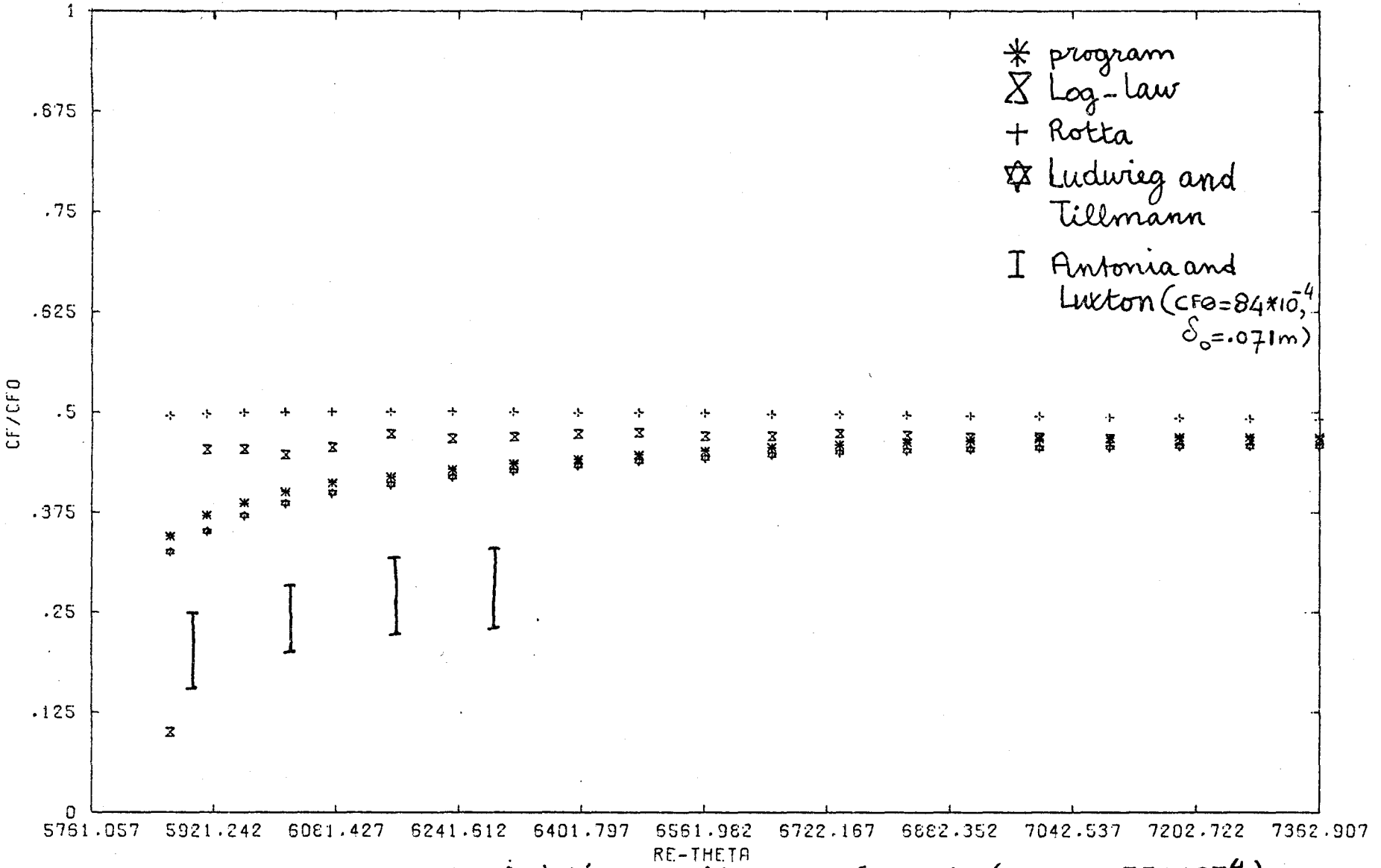


Fig. 4.29 Wall-friction coefficient (r→s) ($CF0 = 57*10^{-4}$)

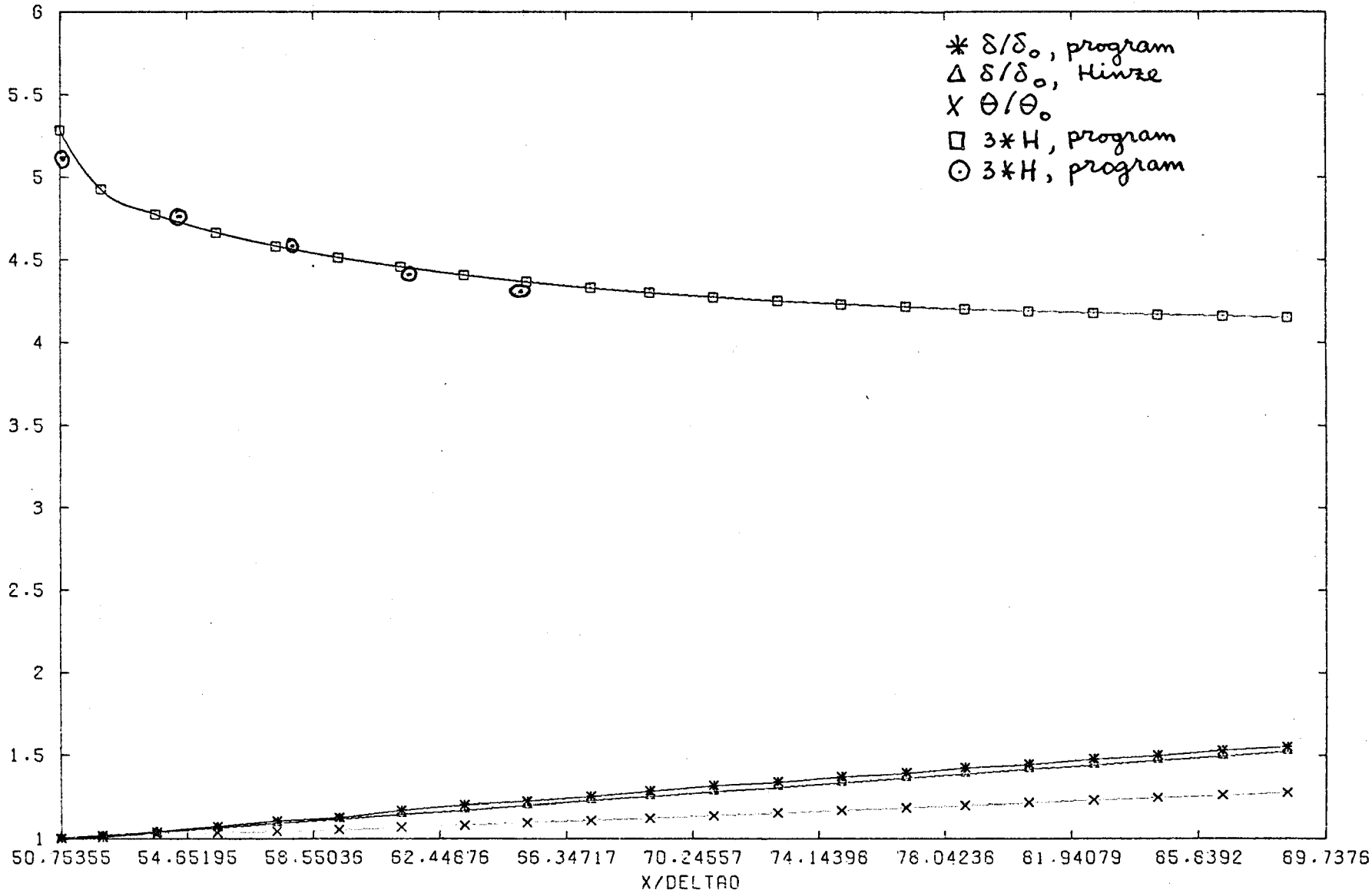


Fig. 4.30 Boundary-layer growth (r → s) (δ_0 = .090m) (θ_0 = .0147m)

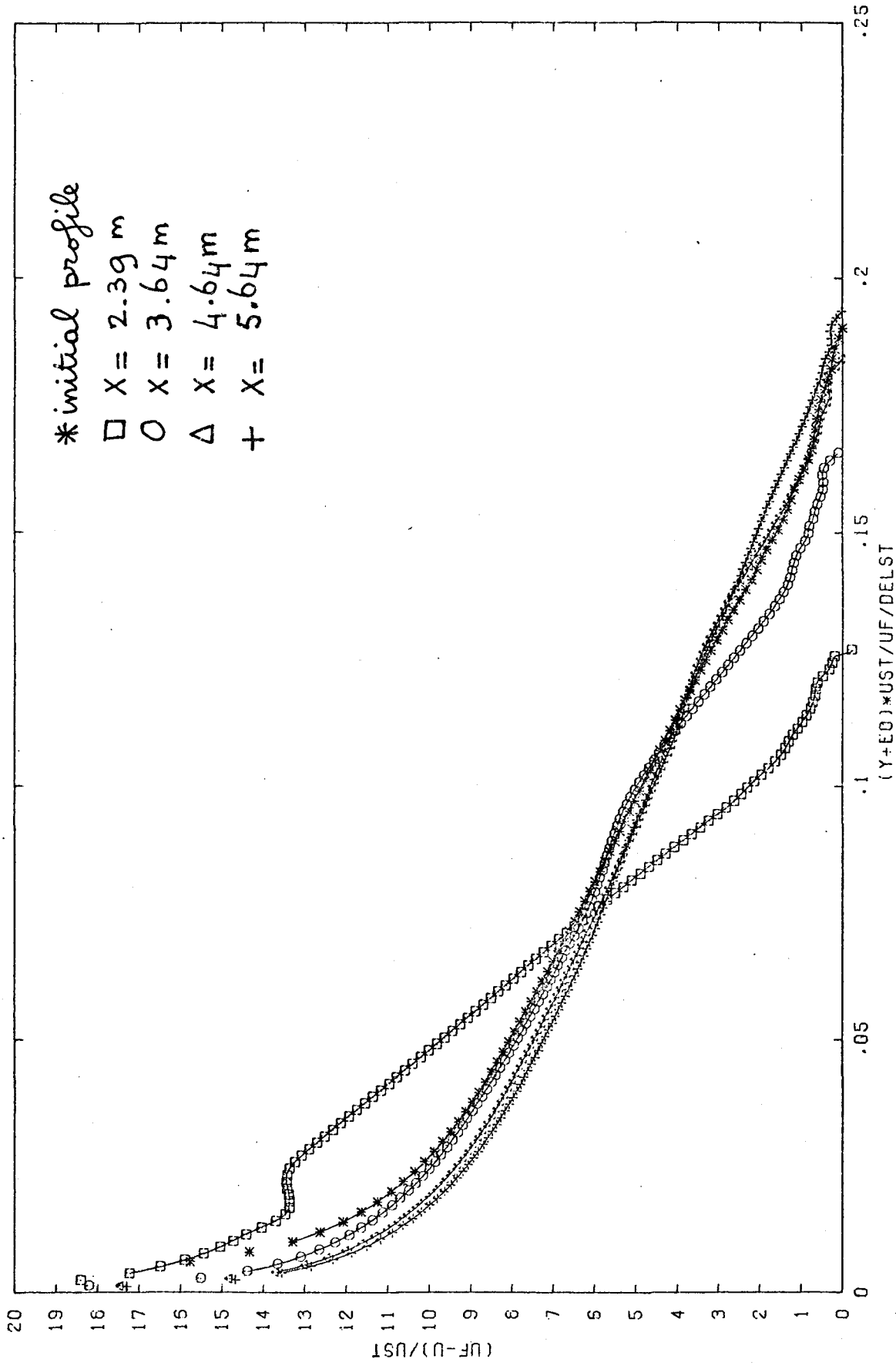


Fig. 4.31 Mean velocity defect profiles ($r \rightarrow s$)

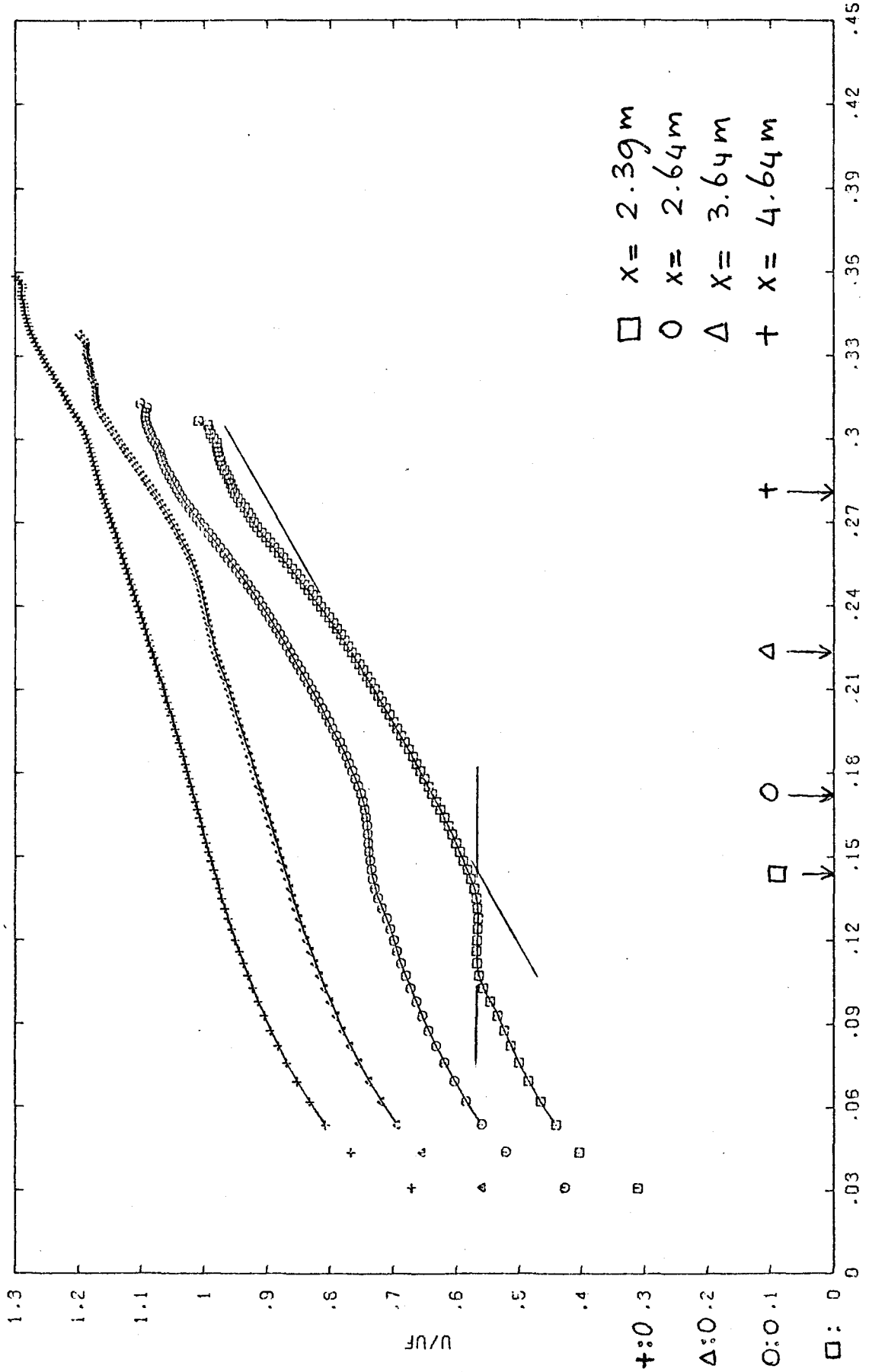


Fig. 4.32 Mean velocity vs. $y^{.5}$ ($r \rightarrow s$) (edge of internal boundary layer is indicated by an arrow) (note shift in origin)

thicknesses from the step. Antonia and Luxton estimated the wall - friction coefficient from the mean velocity distributions on one hand, and measured it with Preston tubes on the other hand. The results of the latter method lie above those of the former one. The difference between the results is indicated by the length of the vertical bars in fig. 4.29, where it must be remarked that the measurements were carried at two different values of the Reynolds number based on the boundary - layer thickness, δ_s, at the step, namely $2.6*10^4$ and $4.8*10^4$. The value of Re_{δ_s} for the present calculations is equal to $3.5*10^4$

The shape parameter H, decreases from a value of 1.8 to 1.4 within a distance of about 25 boundary - layer thicknesses from the step (fig. 4.30). The growth-rate of the boundary layer is very close to the theoretical prediction value of Heinze, while the growth - rate of θ is smaller. The value of $2\,d\theta/dx$ amounts about $24*10^{-4}$, a value which is smaller than C_F ($=27*10^{-4}$) at a distance far from the step.

In fig. 4.31 the mean velocity - defect profiles have been plotted. It can be concluded that at $X = 4.64$, 2.5 m from the step, the mean flow has attained a self - preserving state.

In fig. 4.32 the mean velocity has been plotted versus $y^{.5}$, in order to obtain estimates for the value of δ_i, which are presented in table 4.3

$x-x_0$ (m)	δ_i (m)	$.039(x-x_0)^{.43}$ (A. and L.)
.25	.021	.021
.50	.029	.029
1.50	.050	.046
2.50	.079	.057

$x_0 = 2.14$ m

Table 4.3 The edge, δ_i, of the internal boundary layer. Antonia and Luxton's findings ($\delta_i = .039(x-x_0)^{.43}$, when x and δ_i in m) are also tabulated.

When regarded proportional to x^{γ}, the value of γ for δ_i, predicted by the computer program, is equal to .58, which exceeds the value found Antonia and Luxton by a large amount. One must realize, however, that the values of x and the boundary-layer thickness at the rough to smooth step are different for the present program (namely 2.14 m and .09024 m respectively)

than for Antonia and Luxton's experiments ($\simeq$ 1.18 m and .071 m respectively)

Summarizing, the results for the rough to smooth step-change in surface roughness, one can conclude that it takes a longer distance from the step, for the flow to reach a new self-preserving state, than in the case of a smooth to rough jump. It must again be emphasized that, in order to determine the relaxation-distance, a precise criterium for a fully adapted state must be given.

4.3.6 Conclusions about the results of the calculations on a step-change in surface roughness.

The computer program predicts the behaviour of a turbulent boundary layer, which takes a step change in surface roughness, rather good. This may be concluded, after comparing the calculation results, for a 'k'-type roughness configuration; with experimental data of several turbulent quantities. The calculations of the Reynolds shear stress profile seems to be the most difficult. These calculations may possibly be improved, using empirical functions, which

describe better, the relations between the turbulent shear stress profile and terms in the turbulent energy-equation, in a flow above a rough surface. In particular, in the transition-region, after a step-change in roughness, the shape of the so-called G-function (fig. 1.4, p. 23 of this report) deviates significantly from that in a flow above a smooth or a rough surface. For the L-function a so-called length-scale-equation could be used, which allows for the non-equilibrium form of the dissipation term immediately following the step (Wood, 1978).

5 Conclusions and suggestions for future research.

The computer program, developed during the present investigation on the response of a turbulent boundary layer to a step change in surface roughness, delivers rather satisfactory results. These results have been compared with the experimental results of Antonia and Luxton (1971, 1972) who investigated both smooth to rough and rough to smooth transitions in a turbulent boundary layer under a zero pressure gradient. They used a so-called 'k' type roughness.

It is hard to predict a shear stress profile for a boundary layer which undergoes a step change in surface conditions. Oscillations occur in this profile near the wall and the outer edge of the boundary layer. The oscillations near the wall can be seriously reduced by decreasing the x-step, the increment in x when the calculation procedure marches from one mesh line to another.

The calculations may be improved by making use of a so-called 'length-scale'-equation, which means that the algebraic

expression for $L = (\tau/\rho)^{1.5} / \epsilon$ is replaced by a transport-equation (Wood, 1978). Also, the shape of the G-function (see fig 1.4 of this report) may be modified to obtain better results. Further, a comparison of the calculation results with experimental data of the response of a turbulent boundary layer to a step change in surface-roughness of another type than Antonia and Luxton used, e.g. a 'd' type, may lead to an improvement of the computer program.

Other phenomena, which are worthwile to investigate are:

- the effect on the calculated growth-rate of the internal boundary layer, when different lengths for the rough surface are taken. In this report that distance was taken equal to 2.14 m, while Antonia and Luxton used a rough wall with a length of 1.22 m. The growth-rate, found in this report, of the internal layer, was considerably larger than the value Antonia and Luxton found.
- the application of the outer boundary-conditions at the edge of the internal boundary layer instead of at the edge of the entire boundary layer.

List of symbols

A, B, C, D	mesh points
a	empirical function, $a = .15$
c_F	wall-friction coefficient
f_1, f_2, f_3	functions
G	G-function
H	shape factor
h	roughness-height
i, j, k, l	subscripts, constants
L	L-function
M	$\sqrt{\tau_{max}}$
P	$U_\infty \, dU_\infty / dx$
q^2	turbulent intensity
U, u', U_∞, V, v'	velocity
u_*, u_τ	surface-friction velocity
X, Y, Z	co-ordinates
z_0	roughness-length

Greek symbols

α, β	characteristic directions
γ	exponent
$\delta, \delta_i, \delta_\tau$	boundary-layer thickness
ϵ	dissipation, error in origin
θ	momentum-deficit thickness
ν	kinematic viscosity

ρ	density
σ, ϕ	coefficients, non-dimensional wind shear (ϕ)
τ, τ_w, τ_{max}	kinematic turbulent shear stress.

Superscript

$+$	dimensionless variable with u_* and ν/u_* as scaling parameters
$'$	fluctuating part of the quantity

Special notation

$\equiv$	by definition
$\simeq$	approximately equal to
$\sim$	proportional to

List of tables

List of figures

List of figures (cont'd)

List of figures (cont'd)

Consolidated References.

Antonia, R.A., and Luxton R.E. (1971)
The response of a turbulent boundary layer to a step change in surface roughness, part 1: smooth to rough,
J. Fluid Mech., 48, 721-761

Antonia, R.A. and Luxton R.E. (1972)
the response of a turbulent boundary layer to a step change in surface roughness, part 2: rough to smooth,
J. Fluid Mech., 53, 737-757

Antonia, R.A. and Wood D.H (1975)
Calculation of a turbulent boundary layer downstream of a small step change in surface roughness,
Aero. Quart., 26, 202-210

Blom J., and Wartena L. (1969)
the influence of changes in surface roughness on the development of the turbulent boundary layer in the lower layers of the atmosphere,
J. Atmos. Sci, 26, 255-265

Bradley E.F. (1968)
A micrometeorological study of velocity profiles and surface drag in the region modified by a change in surface roughness, Quart. J. Roy. Meteor. Soc., 94, 361-379

Bradshaw P. (1966)
The turbulence structure of equilibrium boundary layers, J. Fluid Mech., 29, 625-645

Bradshaw P., Ferriss D.H., and Atwell N.P. (1967)
Calculation of boundary-layer development using the turbulent energy equation, J. Fluid Mech., 28, 593-616

Bradshaw P., and Unsworth K. (1974)
An improved Fortran Program for the Bradshaw-Ferriss-Atwell Method of calculating turbulent shear layers, I.C. Aero Report 74-02

Bradshaw P., and Cebeci T. (1977)
Momentum transfer in boundary layers, Mc Graw-Hill, New York

Elliott W.P. (1958)
The growth of the atmospheric internal boundary layer, Trans. Amer. Geophys. Union, 39, 1048-1054

Hinze J.O. (1975)
Turbulence, Mc-Graw Hill, New York

Kessels H.P.J. (1977)
Een numerieke en experimentele bepaling van een luchtstroming door een rechthoekig kanaal, afstudeerverslag, THE

Klebanoff P.S. (1955)
Characteristics of turbulence in a boundary layer with zero pressure gradient, NACA Tech. Rep. 3178

Kline S.J., Reynolds W.C., Schraub F.A., and Runstadler P.W. (1967)
The structure of turbulent boundary layers, J. Fluid Mech., 30, 741-773

Kline S.J., Tani I. (1968)
Report on the AFOSR-IFP-Stanford Conference on computation of turbulent boundary layers, 1, 483-494, 2, 23-25, 222-248, 322-333

Onnen R.J.J. van (1977)
Handleiding voor Plotprocedures, intern rapport, R-317-D, THE

Panofsky H.A., and Townsend A.A. (1964)
Changes of terrain roughness and the wind profile, Quart. J. Roy. Meteor. Soc., 90, 147-155

Perry A.E., Schofield W.H., and Joubert P.N. (1969)
Rough wall turbulent boundary layers, J. Fluid Mech., 37, 383-413

Peterson E.W. (1969)
Modification of mean flow and turbulent energy by a change in surface roughness under conditions of neutral stability, Quart. J. R. Met. Soc., 95, 561-575.

Ralston A., Wilf H.S. (1960)
Mathematical methods for digital computers, John Wiley & Sons, inc. ch. 15, USA

Rao K.S., Wyngaard J.C., and Coté O.R. (1974)
The structure of the two-dimensional internal boundary layer over a sudden change of surface roughness, J. Atmos. Sci., 31, 738-746

Shir C.C. (1972)
A numerical computation of air flow over a sudden change of surface roughness, J. Atmos. Sci., 29, 304-310

Townsend A.A. (1961)
Equilibrium layers and wall turbulence, J. Fluid Mech., 11, 97

Townsend A.A. (1965a)
Self-preserving flow inside a turbulent boundary layer, J. Fluid Mech., 22, 773-797

Townsend A.A (1965b)
The response of a turbulent boundary layer to abrupt changes in surface conditions, J. Fluid Mech., 22, 799-822

Wood D.H (1977)
The growth of the internal layer following a step change in surface roughness, I.C. Aero TN 77-101

Wood D.H. (1978)
Calculation of the neutral wind profile following a large step change in surface roughness,
Dep. of Aeron. Ic., London

The response of a turbulent boundary layer to a step change in surface roughness

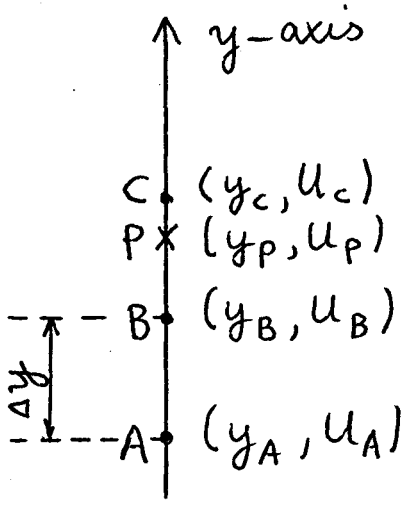
Appendices R-386-A

Contents

Appendix A. The quadratic interpolation.

This section describes the way a quadratic interpolation (section 3.2) is carried out.

Fig. A.1 Equally spaced mesh points A, B and C on the y-axis, and a point P in between them.



The values of a quantity U are only known at the equally spaced mesh points A, B and C. To compute its value at point P it is assumed that U in the region between A and C can be approximated by the quadratic form

$$U = ay^2 + by + c \qquad (A.1)$$

a, b and c are constants, which can be determined from the known values of U

at points A, B and C. This leads to

$$a = (u_A + u_C - 2u_B) / 2 / (\Delta y)^2$$

$$b = \frac{u_C - u_A}{2\,\Delta y} - \frac{2 y_B (u_A + u_C - 2u_B)}{2(\Delta y)^2}$$

$$c = u_B - \frac{(u_C - u_A)\, y_B}{2\Delta y} + \frac{(u_A + u_C - 2u_B)\, y_B^2}{2(\Delta y)^2}$$

Substituting these values for a, b and c in Eq. (A.1) we obtain

$$u_P = u_B + \frac{1}{2\Delta y}(u_C - u_A)(y_P - y_B) + \frac{1}{2(\Delta y)^2}(u_A + u_C - 2u_B)(y_P - y_B)^2 \qquad (A.2)$$

Appendix B. The computer program.

B.1 Organization of the computer program.

The calculations described in ch. 3 can be executed on the Burroughs 7700 computer of the computer centre of Eindhoven. To run a program in which the development of a turbulent boundary layer is calculated, for about two hundred steps downstream from the starting position, it takes approximately one minute computer time. Fig. B.2, p.B.9, presents a flow-chart, in which the organization of the calculation program is depicted. Some comments on the flow-chart are presented here (they follow the numbering system as the chart)

(1) 'Read in initial data'.
The data needed to start the calculations are read in:
- the values of the U-component of mean velocity and the Reynolds shear stress at the mesh points of the initial profiles (appendix D)
- the pressure gradient
- tabulated values of the empirical functions $f_1(y/\delta) = L/\delta$, $f_2(y/\delta) = \frac{\tau}{(\tau/\rho U_\infty^2)^{.5}}$

and a third one, $f_3(y/\delta) = G'\delta/(\tau/\rho U_\infty^2)^{.5}$,

which is used to calculate G', the derivative of G with respect to y. These values were obtained from tables presented by Bradshaw in his Fortran computer program.
The functions are tabulated in table (B.1), p.B.10.
It should be remarked that the value of $f_1(y/\delta)$ in the region $0 \leq y/\delta \leq .15$ corresponds to the mixing-length value, which is equal to $k\,y/\delta$; k is the von Kármán's constant, which is usually taken equal to .4.

(2) The boundary-layer thickness (δ) is defined as the distance from the wall, where U is equal to 99.5 % of the free-stream velocity (U_∞). The displacement thickness (δ^*) and momentum-deficit thickness (θ) are calculated performing numerical integrations using Simpson's formula, given by

$$\int_a^b f(x)\,dx = \frac{h}{3}\big[f(a)+4f(a+h)+2f(a+2h)+ 4f(a+3h)+\cdots\cdots 2f(b-2h)+4f(b-h)+f(b)\big] \qquad (B.1)$$

where $h = \frac{b-a}{N}$, and N is an even number.

Relative deviations of δ and the wall-friction coefficient ($c_f = 2(u_*/u_\infty)^2$), calculated by the program, from their theoretical values are computed. The theoretical formulas for δ and c_f are

*Hinze (1975):

$$\delta = \mathcal{C}\ Re_x^{(n+1)/(n+3)} \qquad (B.2.a)$$

$$c_f = \mathcal{D}\ Re_x^{-2/(n+3)} \qquad (B.2.b)$$

where a power-law velocity profile is assumed, which is valid for zero or small pressure gradients

$$u/u_\infty = (y/\delta)^{1/n} \qquad (B.3)$$

$\mathcal{C}$ and $\mathcal{D}$ are considered to be constants, and Re_x is the Reynolds-number based on x, $Re_x = \frac{u_\infty x}{\nu}$.

With the help of the relation $H = \frac{n+2}{n}$, for the shape parameter, we can convert Eqs. (B.2) into

$$\delta = \mathcal{C}\ Re_x^{(H+1)/(3H-1)} \qquad (B.4.a)$$

$$c_f = \mathcal{D}\ Re_x^{-2(H-1)/(3H-1)} \qquad (B.4.b)$$

* Note: similar formulas hold for δ^* and θ

Hinze's formulas are valid in the region

$$.5 \leq Re_x * 10^6 \leq 10$$

Ludwieg and Tillmann's formula, which is said to be valid in the region

$$10^3 \leq Re_\theta \leq 10^4$$

reads

$$c_f = .246 * 10^{-.678H} Re_\theta^{-.268} \qquad (B.5)$$

where Re_θ is the Reynolds number based on θ, $Re_\theta = U_\infty \theta / \nu$

Rotta's formula

$$c_f = \frac{2}{\{5.75 \log Re_{\delta^*} + 3.7\}^2} \qquad (B.6)$$

where $Re_{\delta^*} = \delta^* U_\infty / \nu$

This formula is, just as Hinze's and Ludwieg and Tillmann's formulas, said to be valid only for zero or small pressure gradients.

(3) j denotes the number of completed x-steps the program runs until a specified maximum number of x-steps has been reached.

(4) The values of V and the characteristic directions $\tan\alpha$ and $\tan\beta$ are calculated at each mesh point. To compute V, the equation along the vertical characteristic, in finite-difference notation of Eq.(2.2), is used

$$V_i = \{ U_i V_{i-1} - (\tau_i - \tau_{i-1}) - \mathcal{P}\,\Delta y \} / U_{i-1} \quad (B.7)$$

where $\mathcal{P} = U_\infty \frac{dU_\infty}{dx}$; and the viscous term in Eq.(2.2) has been neglected, since the region of validity of the calculation method lies beyond the viscous sublayer and the transition layer. V_A, the normal-component of mean velocity in point A (the first mesh point), is taken equal to 0.
To calculate the characteristic directions $\tan\alpha$ and $\tan\beta$, we use

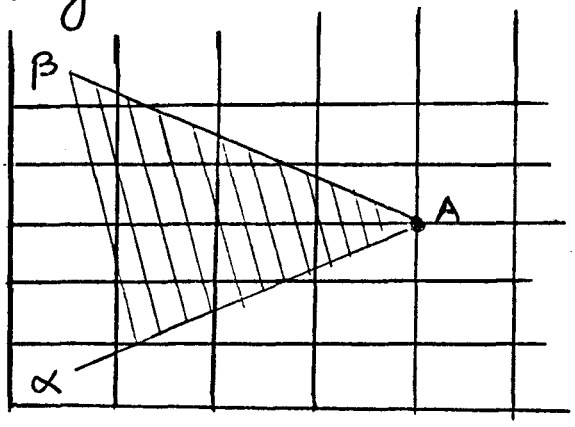
$$\tan\alpha = \frac{1}{U_i} \left\{ V_i + a M G_i + \sqrt{a^2 M^2 G_i^2 + 2a\tau_i} \right\} \quad (B.8.a)$$

$$\tan\beta = \frac{1}{U_i} \left\{ V_i + a M G_i - \sqrt{a^2 M^2 G_i^2 + 2a\tau_i} \right\} \quad (B.8.b)$$

In order to satisfy the boundary-conditions at the upper edge of the boundary layer and to allow the boundary layer to grow as the calculation proceeds, two mesh points are added after every x-step.

U and τ are set there equal to the free-stream velocity and zero respectively. In fact, the boundary layer cannot grow faster than one mesh point per x-step, because the edge of the boundary layer coincides with the 'outgoing' (α)-characteristic.
The key requirement in hyperbolic (and parabolic) problems is the well-known Courant-Friedrichs-Levy-condition, which simply says that the net points determining the values at a given new net point must at least include the correct domain of dependence (fig. B.1) of that point.

Fig. B.1



The domain of dependence (shaded area) enclosed by the α- and β-characteristics drawn from a mesh point A.

Frequently, but not always, the Courant-condition also ensures stability.

The condition is

$$\Delta X \leqslant xstep = \Delta y / (\tan\alpha)_{max} \qquad (B.9)$$

where $(\tan\alpha)_{max}$ is the maximum value, of $\tan\alpha$, along a mesh line. Before proceeding to the downstream mesh line, xstep has to be determined; ΔX is commonly set equal to xstep.

(5) The momentum-deficit thickness is calculated by two methods. On the one hand, the definition-formula is used

$$\theta_{def} \equiv \int_0^\infty \frac{u}{u_\infty}\left(1-\frac{u}{u_\infty}\right) dy \qquad (B.10)$$

On the other hand, the von Kármán integral momentum equation is used

$$\frac{d}{dx}(\theta u_\infty^2) + \mathcal{P}\,\delta^* = \tau_w/\rho \qquad (B.11)$$

Integrating this equation along Δx, between x_1 and x_2, the momentum-deficit thickness at x_2 becomes

$$\theta_k = \frac{\left[\frac{\Delta x}{u_\infty^2}\left\{\bar{\tau}_w - \mathcal{P}\left(\bar{\delta}^* + \theta_k(x_1)\right)\right\} + \theta_k(x_1)\right]}{1 + \frac{\Delta x}{u_\infty^2}\mathcal{P}} \qquad (B.12)$$

At the starting position, x_0, θ_k is set equal to θ_{def}. At all other stations the relative difference between θ_{def} and θ_k,

$$ME = (\theta_k - \theta_{def})/\theta_{def} \qquad (B.13)$$

is calculated, and serves as a measure for the accuracy of the calculation procedure. The bars over the quantities in Eq (B.12), denote average values over the distance $\Delta X = X_2 - X_1$.

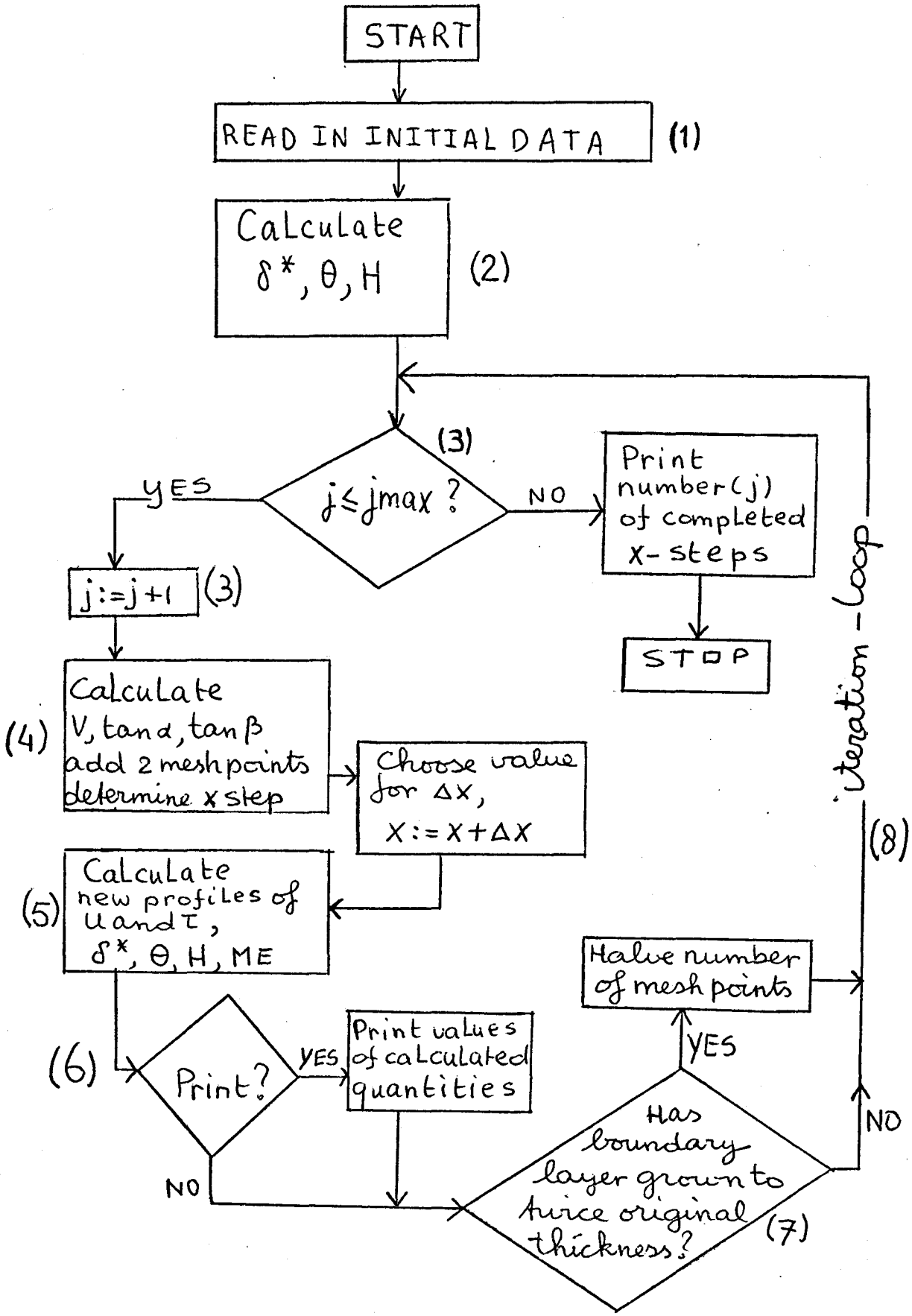
(6) After a preset number of X-steps is completed, all calculated quantities are printed.

(7) When the boundary layer has grown to twice its original thickness, the number of mesh points is halved, and the value of Δy is doubled. This is done to save computer time, and it is not expected to affect the accuracy of the calculation procedure seriously.

(8) Along this iteration-loop the steps from (3) on are repeated.

In next section a listing of the program is shown, and a comment is given on it.

Fig. B.2 Flow chart of the computer program



y/δ	$f_1(y/\delta)$	$f_2(y/\delta)$	$f_3(y/\delta)$
0	0	0	.29
.05	.02	.07	2.37
.1	.04	.24	4.28
.15	.06	.51	6.04
.2	.0752	.88	7.67
.25	.0812	1.33	9.24
.3	.0862	1.86	10.78
.35	.0901	2.48	12.38
.4	.0928	3.18	14.19
.45	.0945	3.96	16.50
.5	.0950	4.82	19.90
.55	.0945	5.76	25.77
.6	.0928	6.77	37.30
.65	.0901	9.33	59.89
.7	.0862	13.88	89.05
.75	.0812	18.42	99.84
.8	.0752	22.97	96.95
.85	.0681	27.51	73.63
.9	.0598	31.68	47.22
.95	.0505	32.62	32.84
1	.04	33.65	26.20

Table B.1 Values of the three empirical functions (taken from Bradshaw, 1967)

The values of $f_3(y/\delta)$ are inferred from those of $f_2(y/\delta)$, by carrying out a differentiation-procedure. Values of the functions at points, in between those given in table 4.1, are computed using a linear interpolation formula. This can be justified, since the functions do not change rapidly and are rather smooth.

B.2 A detailed description of the computer program.

In this section a listing (sec. B.2.2) of the computer program is presented, and a detailed description (sec. B.2.1) of it is given .

B.2.1 A manual of the program.

This section serves as a manual to read the computer program, which can be divided in three main parts.

Part 1 (cards 1001200-1002830, 4002900-4002902).

Declarations of variables.
Integers:
imax0, imax, iimax : number of mesh points.
j, jmax : number of x-steps
Reals:
delta0, delta,
del1, del2, dh : δ, boundary-layer thickness
x0, x : streamwise position on surface
xst, xstep : Δx, resp. xstep, increment of x
ystep : Δy, increment of y
uf0, uf : U_∞, free-stream velocity
delst : δ^*, displacement thickness
theta, CH0, CH1 : θ, momentum-deficit thickness

H, SHAPE : H, G resp., shape parameters
DIF : $(\theta_k - \theta_{def})/\theta_{def}$, momentum-error
UDUDX : $U_\infty \, dU_\infty/dx$
A : $a = .15$
CF, CFT, CH : wall-friction coefficient
REDθ : $1/\nu$, ν is the kinematic viscosity
USTER : $u_* = \sqrt{\tau_w}$, friction velocity
KK : von Kármán's constant

Arrays : (cards 4002901, 4002902)
(C)CU, UU : U-component of mean velocity
(C)CT, TT : kinematic turbulent shear stress
CV, VV : V-component of mean velocity
(T)TAN1 : α-characteristic direction
(T)TAN2 : β-characteristic direction
NF2 : $f_2(y/\delta)$

cards (1002100 – 1002700)
F1 : $f_1(y/\delta)$
F2 : $f_2(y/\delta)$
F3 : $f_3(y/\delta)$
UB, TB : initial profiles of the U-component of velocity and kinematic turbulent shear stress respectively

All variables, which are followed by the suffix 0, are related to their values at the starting-position, x_0.

Part 2 (cards 2001700 - 3008800)

This part contains the declarations of all procedures, which are used to carry out the calculations. It consists of two portions. In part a, several procedures are specified (cards 2001700 - 2042500), which are called in the main calculation procedure, prof (cards 3000100 - 3008800), part b.
Ad.a

Maxvalue (cards 2001700 - 2003000)
The maximum-value, MX, of N elements in an array A, and its index (k) are sought

Thickness (cards 2003300 - 2004400)
The boundary-layer thickness in terms of a number (k) of mesh points is determined,
UD = .995 * UF

Simpson (cards 2004700 - 2006700)
Integrations, using Simpson's rule, are performed. F is the function (array), which has to be integrated between the bounds A and B. H is the integration-step

Linterpol (cards 2007000 - 2008700)
Values of quantities (FF) are calculated at the mesh points, using a linear interpolation formula (card 2008200) between N points, at which these quantities are given as input (F). DT denotes the ratio of the number of mesh points imax and N.

Herlintpol (cards 2009000 - 2011500)
Interpolation - procedure, in which values of the empirical functions f_1, f_2 and f_3 are calculated at the intersection - points of the characteristics and the upstream mesh line.

NF1 : f_1
NF2 : f_2
NF3 : f_3
YI : value of y at intersection-point
F1YI : value of f_1 at intersection point
F2YI : value of f_2 at intersection point
F3YI : value of f_3 at intersection point
TEKEN : Boolean-variable. When teken is 'true' then the α - characteristic-direction is taken, otherwise the β - characteristic direction is chosen.

ME (cards 2011800 – 2014700)
The displacement thickness, momentum-deficit thickness, shape-factor, and momentum-error are computed.

N : number of mesh points
TΘ : wall-shear stress at previous x-station
TN : wall-shear stress at present x-station
H : shape-factor, $H = \delta^* / \theta_{def}$
DIF : $ME = (\theta_k - \theta_{def}) / \theta_{def}$
TDΘ : shear stress at edge of boundary layer, at previous station
TDN : shear stress at edge of boundary layer, at present station
CHΘ : θ_k at previous station
CH1 : θ_k at present station
SG : shape-factor, $SG := \frac{u_\infty}{u_*}(1 - \frac{1}{H})$

VTAN (cards 2015000 – 2017200)
The V-component of mean velocity, and the α- and β- characteristic directions are determined. RV denotes the ratio V/U. To the boundary-layer approximations RV may not exceed .1. $MM = \sqrt{\tau_{max}}$, where τ_{max} is taken equal to the shear stress at $y/\delta = .25$.

Intprof (cards 2017500 - 2019800)
Quadratic interpolations are performed to compute velocity and shear stress at the intersection-points of the characteristics and the upstream mesh line. Between the first and the second mesh point, a logarithmic interpolation-procedure is performed (cards 2019705, 2019710).

UI, TI : U-component of mean velocity and kinematic turbulent shear stress resp., in the intersection-points.

SIGPHI1 (cards 2020100 - 2022200)
Values of σ_α, σ_β, ϕ_α, ϕ_β are computed for calculating the new profiles on the downstream mesh line for the first time. Instead of the averaged values on the characteristics, values at the upstream ends of the characteristics are used, since at the present station no quantity is yet known.

SI : σ_α (teken := true) or σ_β (teken := false)
PI : ϕ_α (teken := true) or ϕ_β (teken := false)

NEWPROF (cards 2022500 - 2036500)

In this procedure, the new profiles of the U-component of mean velocity and the kinematic Reynolds shear stress are calculated. The boundary-condition at the wall is applied to calculate the U-component of mean velocity and the kinematic turbulent shear stress at the first mesh point. In appendix C this will be described in detail. Also this procedure supplies the kinematic wall shear stress at the station concerned.

S1, S2, P1, P2 : σ_α, σ_β, ϕ_α and ϕ_β resp.

T1, T2, U1, U2 : τ_1, τ_2, U_1, U_2 resp.

REP : Boolean variable.
When REP = true, then the second calculation in the first 20 % of the boundary layer is to be performed

RC : Boolean variable.
When RC = true, then we have to deal with a rough surface. Otherwise the surface is smooth.

Y1, Y2 : y_1, y_2 resp.

DEL1, DEL2 : boundary-layer thickness at previous and present station respectively.

MM1, MM2 : $\sqrt{\tau_{max}}$ at previous and present station respectively.

OCU, OCT : profiles at previous station; the arrays CU, CT, tan1, tan2 are overwritten, in this procedure, after every calculation

Now, a detailed description of this important procedure is presented:

Real procedure vel (cards 2024101 - 2024111)
Aspalding velocity - profile is calculated
YP : $y u_* / \nu = y^+$
UT : $U / u_* = U^+$

Real procedure nul (cards 2024200-2026624)
The shear stress (TAUS) at the first mesh point is to be calculated, for the case of a smooth wall (for details, see appendix C)

Real procedure nulr (cards 2026630-2026659)
The shear stress (TAUR) at the first mesh point is to be calculated, for the case of a rough wall (for details, see appendix C)

The calculation of the profiles in a flow past a smooth wall is performed between cards 2026700 and 2035010; those for a rough wall between cards 2035050-2036130

SIGPHI2 (cards 2036800 - 2039400)
Values of σ_α, σ_β, ϕ_α and ϕ_β are computed to be used to recalculate the new profiles in the first 20 % of the boundary layer. Now, the averaged values on the characteristics are used.

Procedure DEV (cards 2039700 - 2040900)
The relative differences between the present value and the previously calculated value of a quantity are computed at each mesh point.

$DU = \frac{CU - LCU}{LCU}$, where LCU is the previously calculated value of CU (velocity)

$DT = \frac{CT - LCT}{LCT}$, where LCT is the previously calculated value of T (stress)

Procedure READDATA (cards 2041200 - 2042100)
In this procedure, the inital profiles are called from the memory of the computer

Procedure VEL (cards 2042201 - 2042211)
A spalding - profile is calculated.

Procedure WRITEDATA (cards 2042400 - 2042500)
In this procedure, an array can be stored in the memory of the computer.

Ad. b

Procedure PROF (cards 3000100 – 3008800)

In this procedure, an entire iteration-loop, in which all quantities at the present station are computed, is carried out.

A description of this procedure now follows

- cards 3001901 – 3001904
  Before calculating the new profiles, the old ones are stored in the arrays OCU and OCT, for the first 20 % of the boundary layer

- cards 3002000 – 3002100
  y, u and τ at the intersections of the characteristic directions and the upstream mesh line are calculated.

- cards 3003000 – 3003600
  The values of f_1, f_2 and f_3 in the intersection points are computed.

- cards 3003700 – 3004000
  σ_α, σ_β, ϕ_α and ϕ_β are calculated, using values of quantities at the upstream ends of the characteristic line segments.

- cards 3004500 – 3004800
  The new profiles are calculated.

– card 3004801
The relative differences between the present values and the previously calculated values of the U- component of mean velocity and kinematic turbulent shear stress at each mesh point are computed. These differences are denoted by the arrays DU and DT respectively.

– cards 3004802 – 3004820
The maximum value of DU or DT, and its index are sought and printed (if demanded)

– cards 3004900 – 3005500
The boundary-layer thickness and the turbulent shear stress at the edge of the boundary layer are determined. Also, the value of τ_{max} is determined.

– cards 3005510 – 3008600
The recalculation in the first 20 % of the boundary layer is performed:
- cards 3005700 – 3005800
  recalculation of the values of V, tan α and tan β
- cards 3005900 – 3006000
  values of y, U and τ at the upstream ends of the characteristics are calculated, using the corrected values of tan α and tan β

- cards 3006100 – 3006700
  values of f_1, f_2, f_3 at the upstream ends of the characteristic line segments are calculated
- cards 3006800 – 3007400
  σ_α, σ_β, ϕ_α and ϕ_β are computed, using averaged values on the characteristics.
- cards 3007500 – 3007800
  the recently calculated profiles are temporarily stored in the arrays LCU and LCT
- cards 3007900 – 3008200
  recalculation of the profiles
- card 3008300
  see card 3004801 for description
- cards 3008310 – 3008513
  see cards 3004802 – 3004820 for description

- card 3008700
  determination of imax

Part 3 (cards 4000001 – 4050950)

In the last part of the calculation program the computations are executed by means of calling the procedures, which are

declared in the preceeding part of the program. Whether we have to deal with a smooth to rough transition or a rough to smooth transition, can be controlled by the choice of the value of the respective boolean variables ROUGH and SMOOTH.

In section B.2.2 a listing of the computer program is presented.

B.2.2. Listing of the program.

On the next pages, a listing of the computer program is shown. The numbering of the lines is indicated in the margin of the pages.
Part I (cards 1001200 - 1002830) is presented on pp. B.25 - B.26.
Part 2 (cards 2001700 - 3008800) is presented on pp. B.27 - B.44.
Part 3 (cards 4000001 - 4050950) and a portion of Part I (cards 4002900 - 4002902) are presented on pp. B.45 - B.66.

```
 $'SET'THELIBRARY                                                          00002000
'BEGIN'                                                                    00003000
$'INCLUDE'"TURB/PICTURE"                                                   00004000
 $ 1000000+100                                                             01000000
 'FILE'OUT,DRAWINU(KIND=PLOTTER11),DRAWINT(KIND=PLOTTER11),                01000100
  DRAWINU1(KIND=PLOTTER11),DRAWINU2(KIND=PLOTTER11),                       01000110
  DRAWINT1(KIND=PLOTTER11),                                                01000120
   DRAWIND(KIND=PLOTTER11),DRAWINC(KIND=PLOTTER11);                        01000130
     'FILE'UBR(KIND=PACK,PACKNAME="USER1.",FILETYPE=7,                     01000200
               TITLE="TURB/UBR.");                                         01000300
     'FILE'TBR(KIND=PACK,PACKNAME="USER1.",FILETYPE=7,                     01000400
               TITLE="TURB/TBR.");                                         01000500
  'FILE'LW   (KIND=PACK,PACKNAME="USER1.",FILETYPE=7,                      01000600
              TITLE="TURB/LW  .");                                         01000700
'FILE'GW   (KIND=PACK,PACKNAME="USER1.",FILETYPE=7,                        01000800
            TITLE="TURB/GW  .");                                           01000900
'FILE'GAFW (KIND=PACK,PACKNAME="USER1.",FILETYPE=7,                        01001000
            TITLE="TURB/GAFW.");                                           01001100
 'FILE'URW(KIND=PACK,PACKNAME="USER1.",FILETYPE=0,UNITS=WORDS,             01001110
           MAXRECSIZE=240,BLOCKSIZE=240,TITLE="TURB/UR.");                 01001120
 'FILE'TRW(KIND=PACK,PACKNAME="USER1.",FILETYPE=0,UNITS=WORDS,             01001130
           MAXRECSIZE=240,BLOCKSIZE=240,TITLE="TURB/TR.");                 01001140
 'FILE'URR(KIND=PACK,PACKNAME="USER1.",FILETYPE=7,                         01001150
           TITLE="TURB/UR.");                                              01001160
 'FILE'TRR(KIND=PACK,PACKNAME="USER1.",FILETYPE=7,                         01001170
           TITLE="TURB/TR.");                                              01001180
  'INTEGER'K,I,Q,IMAX,IIMAX,J,Z,QMAX,RAS,COUNT,JMAX,BE,SUBTRACT;           01001200
   'BOOLEAN'PR,COND1,COND2,COND3,CONDI,WR,SMOOTH,AL,PAL,STORE;             01001300
     'BOOLEAN'PLOT,TEXT,PRINT,PLOTL,PLOTH,TPL;                             01001400
  'BOOLEAN'ADAP,K1,SK,BR,RC,ROUGH,SPL,AS,SYNTH,SC,LOGLAW;                  01001500
   'INTEGER'TEL,II,IMAXO,ZJ,NR,N,KC,KU,KB,BK,STP,J1,J2,LR;                 01001600
```

```
 'REAL'DELTA0,YSTEP,A,MX,XST,UF,MM1,MM2,UDUDX,KK,AA,DEL1,DEL2,MT,          01001700
    ACCU,MI,E0,MM,XMAX,                                                   01001710
 CF,BL,USTER,CFT,RED0,MF,TD0,TDN,X0,UD,MN,RETHETA0,MFA,MFR,XTR,DH,CH,     01001800
 TC,TN,THETA,DELST,X,XSTEP,H,DIF,DELTA,CH0,CH1,SHAPE,RH,XRTR,RY;          01001900
 'REAL'ALFA,VCH0,VCH1,ALFAP,UF0,EKSP,BLS,YP,UT,T,TAU0,CF0;                01002000
  'REAL'DCU;                                                             01002010
'REAL''ARRAY'                                                            01002100
 MINMAXU,MINMAXT,MINMAXD,MINMAXC[1:4],                                   01002300
  MINMAXU1,MINMAXU2,MINMAXT1[1:4],                                       01002350
XP,CFP,CFLOG,CFLUTI,CFHINZE,CFROTTA,DELTAH,DELTAP,DQDP,THETAP,UFP,HP,     01002450
    DELSTP,RP[0:90],                                                     01002451
  Y,LNY,RCU,RCT[0:40],COEF,C0[0:1],                                      01002500
 F1,F2,F3,UB,TB,F[0:21];                                                 01002600
'STRING'COMMENTU[60],COMMENTT[60],TEXTXASU[30],TEXTXAST[30],             01002700
 COMMEND[60],COMMENC[60],TEXTXASC,TEXTXASD[30],TEXTYASD[30],             01002750
          TEXTYASC[30],TEXTYASU[30],TEXTYAST[30],                        01002800
  COMMENTU1[60],COMMENTU2[60],COMMENTT1[60],                             01002810
  TEXTXASU1[30],TEXTXASU2[30],TEXTXAST1[30],                             01002820
  TEXTYASU1[30],TEXTYASU2[30],TEXTYAST1[30];                             01002830
 $ 2000000+100                                                           02000000
'PROCEDURE'DAYTIME(OUT);                                                 02000100
'FILE'OUT;                                                               02000200
'BEGIN'                                                                  02000300
    'INTEGER'HRS,MIN,SEC;                                                02000400
    'REAL'R;                                                             02000500
    SEC:=TIME(1)/60;                                                     02000600
    MIN:=SEC'DIV'60;                                                     02000700
    SEC:=SEC-(MIN*60);                                                   02000800
    HRS:=MIN'DIV'60;                                                     02000900
    MIN:=MIN-(HRS*60);                                                   02001000
    WRITE(OUT,<18("*")/"*",X16,"*"/"*_DATE_:",A2,C4,"___*"/"*",X16,      02001100
          "*"/"*_TIME_:",I2,":",I2,":",I2,"_*"/"*",X16,"*"/              02001200
          18("*")///>,R:=TIME(15),R,HRS,MIN,SEC);                        02001300
'END'PROCEDURE DAYTIME;                                                  02001400
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  02001500
 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  02001600
```

```
'PROCEDURE'MAXVALUE( A,N,K,MX );                                    02001700
'VALUE'N;                                                           02001800
'INTEGER'N,K;                                                       02001900
'REAL'MX;                                                           02002000
'REAL''ARRAY'A[*];                                                  02002100
'BEGIN'                                                             02002200
    'INTEGER'I;                                                     02002300
    MX:=A[0];K:=0;                                                  02002400
    'FOR'I:=1'STEP'1'UNTIL'N                                        02002500
    'DO''IF'A[I]'GTR'MX                                             02002600
        'THEN''BEGIN'                                               02002700
                MX:=A[I];K:=I                                       02002800
            'END'                                                   02002900
'END'PROCEDURE  MAXVALUE;                                           02003000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02003100
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02003200
  'PROCEDURE'THICKNESS(CU,UD,D,K);                                  02003300
  'VALUE'UD;                                                        02003400
'INTEGER'K;                                                         02003500
 'REAL'UD,D;                                                        02003600
  'REAL''ARRAY'CU[*];                                               02003700
'BEGIN'                                                             02003800
    'REAL'I;                                                        02003900
    I:=1;                                                           02004000
      'WHILE'CU[I]'LSS'UD                                           02004100
    'DO'I:=I+1;                                                     02004200
      K:=I;D:=CU[I];                                                02004300
'END'PROCEDURE THICKNESS;                                           02004400
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02004500
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02004600
'REAL''PROCEDURE'SIMPSON(F,A,B,H);                                  02004700
'VALUE'A,B,H;                                                       02004800
'INTEGER'A,B;                                                       02004900
'REAL'H;                                                            02005000
'REAL''ARRAY'F[*];                                                  02005100
'BEGIN'                                                             02005200
```

```
    'INTEGER'I;                                                          02005300
    'REAL'S;                                                             02005400
    'IF'B-A'NEQ'2*((B-A)'DIV'2)                                          02005500
    'THEN''BEGIN'                                                        02005600
              S:=3*(F[B]+F[B-1])/2;                                      02005700
              'FOR'I:=A'STEP'2'UNTIL'B-2                                 02005800
              'DO'S:=S+F[I]+4*F[I+1]+F[I+2]                              02005900
         'END'                                                           02006000
    'ELSE''BEGIN'                                                        02006100
              S:=0;                                                      02006200
              'FOR'I:=A'STEP'2'UNTIL'B-1                                 02006300
              'DO'S:=S+F[I]+4*F[I+1]+F[I+2]                              02006400
         'END';                                                          02006500
    SIMPSON:=S*H/3                                                       02006600
'END' PROCEDURE SIMPSON;                                                 02006700
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02006800
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02006900
'PROCEDURE'LINTERPOL(F,FF,N,DT);                                         02007000
'VALUE'N,DT;                                                             02007100
'INTEGER'N;                                                              02007200
'REAL'DT;                                                                02007300
'REAL''ARRAY'F,FF[*];                                                    02007400
'BEGIN'                                                                  02007500
    'INTEGER'I,J;                                                        02007600
    J:=0;                                                                02007700
    'FOR'I:=1'STEP'1'UNTIL'N                                             02007800
    'DO''BEGIN'                                                          02007900
              'WHILE'J'LEQ'I*DT                                          02008000
         'DO''BEGIN'                                                     02008100
                   FF[J]:= F[I-1]+(J/DT-(I-1))*(F[I]-F[I-1]);            02008200
                   J:=J+1;                                               02008300
              'END';                                                     02008400
        'END';                                                           02008500
     FF[J]:=F[N-1]+(J/DT-(N-1))*(F[N]-F[N-1]);                           02008600
'END'PROCEDURE LINTERPOL;                                                02008700
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02008800
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02008900
```

```
'PROCEDURE'HERLINTPOL(NF1,NF2,NF3,YI,F1YI,F2YI,F3YI,UF,DEL,MM,YSTEP,     02009000
                      TEKEN,N,Z);                                       02009100
'VALUE'UF,DEL,MM,N,YSTEP;                                               02009200
 'INTEGER'N,Z;                                                          02009300
'REAL'UF,DEL,MM,YSTEP;                                                  02009400
'BOOLEAN'TEKEN;                                                         02009500
'REAL''ARRAY'NF1,NF2,NF3,YI,F1YI,F2YI,F3YI[*];                          02009600
'BEGIN'                                                                 02009700
    'INTEGER'I,P,Q,IN;                                                  02009800
    'IF'TEKEN                                                           02009900
    'THEN''BEGIN'                                                       02010000
              P:=1;Q:=0;IN:=2;                                          02010100
         'END'                                                          02010200
    'ELSE''BEGIN'                                                       02010300
              P:=0;Q:=1;IN:=1;                                          02010400
         'END';                                                         02010500
       'FOR'I:=IN'STEP'1'UNTIL'N                                        02010600
    'DO''BEGIN'                                                         02010700
            F2YI[I]:=NF2[I-P]+(YI[I]-YSTEP*(I-P))/YSTEP                 02010800
                    *(NF2[I+Q]-NF2[I-P]);                               02010900
            F1YI[I]:=NF1[I-P]+(YI[I]-YSTEP*(I-P))/YSTEP                 02011000
                    *(NF1[I+Q]-NF1[I-P]);                               02011100
            F3YI[I]:=NF3[I-P]+(YI[I]-YSTEP*(I-P))/YSTEP                 02011200
                     *(NF3[I+Q]-NF3[I-P]);                              02011300
        'END';                                                          02011400
'END'PROCEDURE HERLINTPOL;                                              02011500
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02011600
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02011700
'PROCEDURE'ME(CU,N,J,YSTEP,UF,UDUDX,XST,TO,TN,THETA,DEL2,DELST,H,DIF,    02011800
              TDO,TDN,MF,                                               02011900
                  CH0,CH1,SG,Z);                                        02012000
'VALUE'N,J,YSTEP,UF,UDUDX,XST,TO,TN,DEL2,TDO,TDN,MF;                    02012100
   'INTEGER'N,J,Z;                                                      02012200
'REAL'YSTEP,UF,UDUDX,XST,TO,TN,DEL2,THETA,DELST,H,DIF,CH0,CH1,SG,TDO,   02012300
                TDN,MF;                                                 02012400
'REAL''ARRAY'CU[*];                                                     02012500
```

```
'BEGIN'                                                                  02012600
    'INTEGER'I;                                                          02012700
    'REAL'A,B;                                                           02012800
        A:=SIMPSON(CU,Z-1,N,YSTEP)/UF;                                   02012900
    'FOR'I:=0'STEP'1'UNTIL'N                                             02013000
    'DO'CU[I]:=CU[I]**2;                                                 02013100
        B:=SIMPSON(CU,Z-1,N,YSTEP)/(UF**2);                              02013200
    'FOR'I:=0'STEP'1'UNTIL'N                                             02013300
    'DO'CU[I]:=SQRT(CU[I]);                                              02013400
    'IF'J'GEQ'1                                                          02013500
    'THEN''BEGIN'                                                        02013600
            CH0:=CH1;                                                    02013700
            CH1:=(0.5*XST/(UF**2)*((TN+T0-TD0-TDN)*MF**2-UDUDX*          02013800
            (DELST+DEL2-A+2*CH0))+CH0)/(1+UDUDX*XST/(UF**2));            02013900
               DIF:=(CH1-(A-B))/(A-B);                                   02014000
          'END';THETA:=A-B;DELST:=DEL2-A;H:=DELST/THETA;                  02014100
           SG:=UF/MF/SQRT(TN)*(1-1/H);                                   02014200
              'IF'J=0                                                    02014300
              'THEN''BEGIN'                                              02014400
                       DIF:=0;CH1:=THETA;                                02014500
                  'END';                                                 02014600
'END' PROCEDURE ME;                                                      02014700
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02014800
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02014900
 'PROCEDURE'VTAN(CU,CT,NF2,CV,TAN1,TAN2,RV,A,MM,UF,UDUDX,YSTEP,IMAX,      02015000
            MF,Z);                                                       02015100
  'VALUE'A,MM,IMAX,UF,UDUDX,YSTEP,MF;                                    02015200
 'INTEGER'IMAX,Z;                                                        02015300
'REAL'A,MM,UF,UDUDX,YSTEP,MF;                                            02015400
'REAL''ARRAY'CU,CT,CV,NF2,TAN1,TAN2,RV[*];                               02015500
'BEGIN'                                                                  02015600
    'INTEGER'I;                                                          02015700
    'REAL'T;                                                             02015800
    'FOR'I:=0'STEP'1'UNTIL'Z-1                                           02015900
     'DO'CV[I]:=TAN1[I]:=TAN2[I]:=RV[I]:=0;                              02015901
    CV[Z]:=RV[Z]:=0;                                                     02015902
    T:=SQRT((A*NF2[Z]*MM/UF*MM*MF**2)**2+2*A*CT[Z]*MF**2);               02016000
```

```
     TAN1[Z]:=(CV[Z]+A*NF2[Z]*MM/UF*MM*MF**2+T)/CU[Z];                    02016100
     TAN2[Z]:=(CV[Z]+A*NF2[Z]*MM/UF*MM*MF**2-T)/CU[Z];                    02016200
     'FOR'I:=Z+1'STEP'1'UNTIL'IMAX                                       02016300
    'DO''BEGIN'                                                          02016400
            CV[I]:=(CU[I]*CV[I-1]-(CT[I]-CT[I-1])*MF**2-                 02016500
            UDUDX*YSTEP)/CU[I-1];                                        02016600
        T:=SQRT(    (A*NF2[I]*MM/UF*MM*MF**2)**2+2*A*CT[I]*MF**2);       02016700
            TAN1[I]:=(CV[I]+A*NF2[I]*MM/UF*MM*MF**2+T)/CU[I];            02016800
            TAN2[I]:=(CV[I]+A*NF2[I]*MM/UF*MM*MF**2-T)/CU[I];            02016900
            RV[I]:=CV[I]/CU[I];                                          02017000
        'END'                                                            02017100
'END' PROCEDURE VTAN;                                                    02017200
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02017300
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02017400
'PROCEDURE'INTPROF(CU,CT,UI,TI,YI,N,TEKEN,TANI,YSTEP,XST,Z,REP,OUT);     02017500
'VALUE'N,YSTEP,XST;                                                      02017600
 'INTEGER'N,Z;                                                           02017700
'REAL'YSTEP,XST;                                                         02017800
  'FILE'OUT;                                                             02017850
 'BOOLEAN'TEKEN,REP;                                                     02017900
'REAL''ARRAY'CU,CT,UI,TI,YI,TANI[*];                                     02018000
'BEGIN'                                                                  02018100
     'INTEGER'I,IN,K;                                                    02018200
  'REAL'SLOPE;                                                           02018201
    'IF'TEKEN                                                            02018300
    'THEN'IN:=2                                                          02018400
    'ELSE'IN:=1;                                                         02018500
         'FOR'I:=IN+Z-1'STEP'1'UNTIL'N-1                                 02018600
    'DO''BEGIN'                                                          02018700
            YI[I]:=YSTEP*I-TANI[I]*XST;                                  02018800
            UI[I]:=CU[I]+   1/(2*YSTEP)*(CU[I+1]-CU[I-1])                02018900
                   *(YI[I]-YSTEP*I)+                                     02019000
                   0.5/(YSTEP**2)*(CU[I-1]+CU[I+1]-2*CU[I])              02019100
                   *(YI[I]-YSTEP*I)**2;                                  02019200
            TI[I]:=CT[I]+   1/(2*YSTEP)*(CT[I+1]-CT[I-1])                02019300
```

```
                    *(YI[I]-YSTEP*I)+                                          02019400
                    0.5/(YSTEP**2)*(CT[I-1]+CT[I+1]-2*CT[I])                   02019500
                    *(YI[I]-YSTEP*I)**2;                                       02019600
  'IF'TI[I]'LEQ'0'THEN''BEGIN'                                                 02019610
WRITE(OUT,</,"J=",I3,/,"IMAX=",I3,"I=",I3,"TI=",F12.7>,J,IMAX,I,TI[I]);        02019620
    WRITE(OUT,</,"XST=",F12.4>,XST);WRITE(OUT,</,"TEKEN=">);                   02019630
   WRITE(OUT,/,TEKEN);'END';                                                   02019640
       'END';                                                                  02019700
  'IF''NOT'TEKEN'THEN'                                                         02019705
   UI[Z]:=CU[Z]+(CU[Z+1]-CU[Z])/LN((Z+1)/Z)*LN(YI[Z]/YSTEP/Z);                 02019710
'END'PROCEDURE INTPROF;                                                        02019800
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02019900
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02020000
'PROCEDURE'SIGPHI1(F1YI,F2YI,F3YI,UI,TI,SI,PI,A,XST,UDUDX,MM,TEKEN,N,          02020100
                UF,MF,DEL1,Z);                                                 02020200
'VALUE'A,UDUDX,MM,N,XST,MF,DEL1,UF;                                            02020300
 'INTEGER'N,Z;                                                                 02020400
'REAL'A,UDUDX,MM,XST,MF,DEL1,UF;                                               02020500
'REAL''ARRAY'F1YI,F2YI,F3YI,UI,TI,SI,PI[*];                                    02020600
'BOOLEAN'TEKEN;                                                                02020700
'BEGIN'                                                                        02020800
    'INTEGER'I,IN;                                                             02020900
    'REAL'SIGN;                                                                02021000
    'IF'TEKEN                                                                  02021100
    'THEN''BEGIN'SIGN:=1;IN:=2'END'                                            02021200
    'ELSE''BEGIN'SIGN:=-1;IN:=1'END';                                          02021300
         'FOR'I:=IN+Z-1'STEP'1'UNTIL'N-1                                       02021400
    'DO''BEGIN'                                                                02021500
            SI[I]:=F2YI[I]*MM**2/UF*MF**2+SIGN*SQRT((F2YI[I]*MM**2/UF*         02021600
                MF**2)**2+2*TI[I]/A*MF**2);                                    02021700
            PI[I]:=TI[I]*UI[I]-0.5*SI[I]*TI[I]+XST*                            02021800
                TI[I]/UI[I]*(UDUDX+A*(SQRT(TI[I])*MF/DEL1/F1YI[I]+             02021900
                F3YI[I]*MM**2*MF**2/DEL1/UF)*SI[I]);                           02022000
       'END';                                                                  02022100
'END' PROCEDURE SIGPHI1;                                                       02022200
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02022300
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 02022400
```

```
'PROCEDURE'NEWPROF(S1,S2,P1,P2,T1,T2,CT,CU,KK,AA,YSTEP,IMAX,REP,WR,     02022500
          RC,MM2,UF,F2Y2,NF2,F3Y2,NF3,PR,J,II,Y1,Y2,                    02022600
        REDO,MM1,MF,DEL1,ADAP,RH,OCU,OCT,TAN1,TAN2,                     02022700
 F1Y2,NF1,U1,U2,DEL2,XST,UDUDX,A,Z,OUT,ALFA,RY,AS,TR,BK,EO);            02022800
  'VALUE'KK,AA,IMAX,YSTEP,XST,DEL2,UDUDX,A,MM2,UF,J,                   02022900
       REDO,DEL1,MF,MM1;                                              02023000
 'INTEGER'IMAX,J,II,Z,BK;                                             02023100
 'REAL'KK,AA,YSTEP,XST,DEL2,UDUDX,A,MM2,UF,RH,REDO,MF,DEL1,MM1,ALFA,EO, 02023200
       RY,TR;                                                         02023300
   'BOOLEAN'REP,PR,WR,ADAP,RC,AS;                                     02023400
'REAL''ARRAY'S1,S2,P1,P2,T1,T2,CT,CU,NF1,F1Y2,U1,U2,F2Y2,NF2,F3Y2,NF3,  02023500
       Y1,Y2,OCU,OCT,TAN1,TAN2[*];                                    02023600
 'FILE'OUT;                                                           02023700
'BEGIN'                                                               02023800
    'INTEGER'I,BOUND;                                                 02023900
'REAL'T,TAUS,UST,US,UR,TM1,TM2,UM,LM,TM,GM,GAFM,USTR,TAUR,ALFAR,HPLUS; 02024000
  'REAL'MM,YP,UT,FAC;                                                 02024001
     'REAL''ARRAY'YY,RCT[1:40],CO,COR[0:1];                           02024100
  'REAL''PROCEDURE'VEL(YP,UT);                                        02024101
  'VALUE'YP;                                                          02024102
  'REAL'YP,UT;                                                        02024103
  'BEGIN'                                                             02024104
      VEL:=UT+EXP(-AA)*(EXP(KK*UT)-1-KK*UT-                           02024105
           (KK*UT)**2/2-                                              02024106
           (KK*UT)**3/6-                                              02024107
           (KK*UT)**4/24-                                             02024108
           (KK*UT)**5/120-                                            02024109
           (KK*UT)**6/720)-YP;                                        02024110
  'END' PROCEDURE VEL;                                                02024111
  'REAL''PROCEDURE'NUL(TAUS);                                         02024200
  'REAL'TAUS;                                                         02024300
  'BEGIN'                                                             02024400
  'FILE'OUT;                                                          02024401
```

```
   CT[BK]:=TAUS;                                                                02024416
   'FOR'I:=1'STEP'1'UNTIL'3                                                     02024420
   'DO''BEGIN'                                                                  02024421
           YY[I]:=YSTEP*(BK-1+I);                                               02024422
           RCT[I]:=CT[BK-1+I];                                                  02024423
       'END';                                                                   02024424
   LEAST SQUARES POLYNOMIAL(YY,RCT,3,1,C0);                                     02024425
   ALFA:=C0[1];                                                                 02024426
     CT[0]:=TAUS-ALFA*YSTEP;                                                    02024428
     'IF'AS'THEN'CT[0]:=TAUS;                                                   02024429
    UST:=SQRT(CT[0]);                                                           02025200
     US:=UST/KK*(LN(BK*UST*YSTEP*REDO)+AA-                                      02025300
        2*LN((SQRT(TAUS/CT[0])+1)/2)+                                           02025400
        2*(SQRT(TAUS/CT[0])-1));                                                02025500
'IF'REP                                                                         02025600
'THEN''BEGIN'                                                                   02025700
    TAN2[BK]:=-SQRT(2*A*TAUS)/US;                                               02025701
     Y2[BK]:=YSTEP*BK-XST*TAN2[BK];                                             02025702
    F1Y2[BK]:=NF1[BK]+(Y2[BK]-BK*YSTEP)/YSTEP*(NF1[BK+1]-NF1[BK]);              02025703
    T2[BK]:=OCT[BK]+.5/YSTEP*(OCT[BK+1]-OCT[BK-1])*(Y2[BK]-BK*YSTEP)+           02025704
            .5/(YSTEP**2)*(OCT[BK-1]+OCT[BK+1]-2*OCT[BK])*                      02025705
            (Y2[BK]-BK*YSTEP)**2;                                               02025706
U2[BK]:=OCU[BK]+(OCU[BK+1]-OCU[BK])/LN((BK+1)/BK)*LN(Y2[BK]/BK/YSTEP);          02025707
    LM:=(F1Y2[BK]*DEL1+NF1[BK]*DEL2)/2;                                         02025751
    UM:=(U2[BK]+US)/2;                                                          02025800
    TM:=(T2[BK]+TAUS)/2;                                                        02025900
        'END'                                                                   02025905
   'ELSE''BEGIN'UM:=U2[BK];TM:=T2[BK];LM:=F1Y2[BK]*DEL1;'END';                  02025910
    S2[BK]:=-SQRT(2*TM/A);                                                      02026000
    P2[BK]:=TM *U2[BK]-0.5*S2[BK]*T2[BK]+                                       02026100
           XST*TM/UM*(UDUDX+A*(SQRT(TM)/LM)*                                    02026200
            S2[BK]);                                                            02026300
    NUL:=TAUS-2*(TM*US-P2[BK])/S2[BK];                                          02026600
'END' PROCEDURE NUL;                                                            02026624
```

```
'REAL''PROCEDURE'NULR(TAUR);                                                02026630
'REAL'TAUR;                                                                 02026631
 'BEGIN''FILE'OUT;                                                          02026632
    USTR:=SQRT(TAUR);                                                       02026633
   UR:=USTR/KK*(LN((YSTEP+EO)/RH)+3.1*KK);                                  02026638
'IF'REP                                                                     02026640
'THEN''BEGIN'                                                               02026641
  TAN2[Z]:=-SQRT(2*A*TAUR)/UR;                                              02026642
Y2[Z]:=Z*YSTEP-XST*TAN2[Z];                                                 02026643
F1Y2[Z]:=NF1[Z]+(Y2[Z]-Z*YSTEP)/YSTEP*(NF1[Z+1]-NF1[Z]);                    02026644
T2[Z]:=OCT[Z]+.5/YSTEP*(OCT[Z+1]-OCT[Z-1])*(Y2[Z]-Z*YSTEP)+                 02026645
       .5/(YSTEP**2)*(OCT[Z-1]+OCT[Z+1]-2*OCT[Z])*                          02026646
       (Y2[Z]-Z*YSTEP)**2;                                                  02026647
U2[Z]:=OCU[Z]+(OCU[Z+1]-OCU[Z])/LN((Z+1)/Z)*LN(Y2[Z]/Z/YSTEP);               02026648
           LM:=(F1Y2[Z]*DEL1+NF1[Z]*DEL2)/2;                                02026649
   UM:=(U2[Z]+UR)/2;                                                        02026650
  TM :=(T2[Z]+TAUR)/2;                                                      02026651
     'END'                                                                  02026652
'ELSE''BEGIN'UM:=U2[Z];TM:=T2[Z];LM:=F1Y2[Z]*DEL1;'END';                    02026653
S2[Z]:=-SQRT(2*TM/A);                                                       02026654
                  P2[Z]:=TM *U2[Z]-0.5*S2[Z]*T2[Z]+                         02026655
       XST*TM/UM*(UDUDX+A*(SQRT(TM)/LM)*                                    02026656
                         S2[Z]);                                            02026657
 NULR:=TAUR-2*(TM*UR-P2[Z])/S2[Z];                                          02026658
'END' PROCEDURE NULR;                                                       02026659
'IF''NOT'RC                                                                 02026700
'THEN''BEGIN'                                                               02026701
     'FOR'I:=BK+1'STEP'1'UNTIL'IMAX-1                                       02026710
  'DO''BEGIN'                                                               02026720
         'IF''NOT'REP                                                       02026730
         'THEN''BEGIN'                                                      02026740
                  TM1:=T1[I];TM2:=T2[I];'END'                               02026750
         'ELSE''BEGIN'                                                      02026760
         TM1:=(T1[I]+CT[I])/2;                                              02026770
         TM2:=(T2[I]+CT[I])/2;                                              02026780
               'END';                                                       02026790
```

```
            T:=S1[I]*TM2  -S2[I]*TM1  ;                                          02026800
            CU[I]:=(S1[I]*P2[I]-S2[I]*P1[I])/T;                                  02026810
            CT[I]:=(TM1  *P2[I]-TM2  *P1[I])/T*2;                                02026820
         'END';                                                                  02026830
 'IF'REP'THEN''BEGIN'                                                           02027000
    'IF'AS'THEN'FAC:=@-2'ELSE'FAC:=.7;                                          02027100
     'IF'ZERO IN AB(NUL(TAUS),TAUS,FAC*TR,6*TR,@-6,@-6)                         02031600
'THEN''BEGIN'                                                                   02031700
   CT[BK]:=TAUS;                                                                02031800
'FOR'I:=1'STEP'1'UNTIL'3                                                        02031810
'DO''BEGIN'                                                                     02031820
   YY[I]:=(BK-1+I)*YSTEP;                                                       02031830
   RCT[I]:=CT[BK-1+I];                                                          02031840
    'END';                                                                      02031850
LEAST SQUARES POLYNOMIAL(YY,RCT,3,1,CO);                                        02031860
ALFA:=CO[1];                                                                    02031870
     'FOR'I:=0'STEP'1'UNTIL'BK-1                                                02031920
 'DO''BEGIN'CT[I]:=TAUS-ALFA*YSTEP;'IF'AS'THEN'CT[I]:=TAUS;'END';               02031930
 UST:=SQRT(CT[0]);                                                              02031931
 CU[BK]:=UST/KK*(LN(BK*UST*YSTEP*REDO)+AA-                                      02031932
   2*LN((SQRT(CT[BK]/CT[0])+1)/2)+                                              02031933
   2*(SQRT(CT[BK]/CT[0])-1));                                                   02031934
 'IF'BK>1                                                                       02032006
 'THEN''BEGIN'                                                                  02032007
        'FOR'I:=BK'STEP'-1'UNTIL'1                                              02032008
          'DO''BEGIN'                                                           02032009
       YP:=SQRT(CT[0])*YSTEP*REDO*I;                                            02032010
       'IF'ZERO IN AB(VEL(YP,UT),UT,0,UF/SQRT(CT[0]),@-6,@-6)                   02032011
        'THEN''BEGIN''IF'I=BK'THEN'MM:=UT*SQRT(CT[0]);                          02032012
         'IF'I<BK'THEN'CU[I]:=UT*SQRT(CT[0])+CU[BK]-MM;'END'                    02032013
        'ELSE''BEGIN'WRITE(OUT,</,"NO-SOLUTION-FOR-UT",X4,"I=",I3,              02032014
                  X4,"J=",I3>,I,J);WRITE(OUT,</"UT=",X8,"VEL=",/>);             02032015
                  T:=UF/SQRT(CT[0]);                                            02032016
  'FOR'UT:=0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1                                       02032017
  'DO'WRITE(OUT,</,F8.4,X8,F8.4>,UT*T,VEL(YP,UT*T));                            02032018
        ALARM('('NO-SOLUTION-FOR-UT')');                                        02032019
             'END';                                                             02032020
          'END';                                                                02032021
      'END';                                                                    02032022
   'END'                                                                        02032500
```

```
'ELSE''BEGIN'                                                          02032600
    WRITE(OUT,</,"NO-SOLUTION-FOR-TAUS",X4,"J=",I3>,J);                02032700
 WRITE(OUT,</,"AS=">);WRITE(OUT,/,AS);                                 02032710
 WRITE(OUT,</,"REP=">);WRITE(OUT,/,REP);                               02032750
              WRITE(OUT,</,"TAUS=",X6,"NUL=",/>);                      02032800
  'FOR'TAUS:=2-3,.7,1,2,3,4,5,6,7                                      02032900


  'DO'WRITE(OUT,</,F8.4,X6,F8.4>,TAUS*TR,NUL(TAUS*TR));                02033100
      ALARM('('NO-SOLUTION-FOR-TAUS')');                               02033200
        'END';                                                         02033300
              'END';                                                   02034000
 CU[0]:=0;                                                             02035000
        'END';                                                         02035010
 'IF'RC                                                                02035050
 'THEN''BEGIN'                                                         02035051
 Z:=1;                                                                 02035052
     'FOR'I:=Z+1'STEP'1'UNTIL'IMAX-1                                   02036000
    'DO''BEGIN'                                                        02036001
           'IF''NOT'REP                                                02036002
           'THEN''BEGIN'                                               02036003
                    TM1:=T1[I];TM2:=T2[I];'END'                        02036004
         'ELSE''BEGIN'                                                 02036005
         TM1:=(T1[I]+CT[I])/2;                                         02036006
         TM2:=(T2[I]+CT[I])/2;                                         02036007
              'END';                                                   02036008
         T:=S1[I]*TM2  -S2[I]*TM1  ;                                   02036009
         CU[I]:=(S1[I]*P2[I]-S2[I]*P1[I])/T;                           02036010
         CT[I]:=(TM1  *P2[I]-TM2  *P1[I])/T*2;                         02036011
      'END';                                                           02036012
```

```
  'IF'REP                                                                    02036050
  'THEN''BEGIN'                                                              02036051
     'IF'ZERO IN AB(NULR(TAUR),TAUR,.25*TR,10*TR,a-6,a-6)                    02036052
    'THEN''BEGIN'                                                            02036053
      'FOR'I:=0'STEP'1'UNTIL'Z                                               02036054
      'DO'CT[I]:=TAUR;                                                       02036055
      USTR:=SQRT(CT[0]);                                                     02036056
      'FOR'I:=0'STEP'1'UNTIL'Z-1                                             02036057
      'DO'CU[I]:=0;                                                          02036058
    CU[Z]:=USTR/KK*(LN((YSTEP+E0)/RH)+3.1*KK);                               02036063
           'END'                                                             02036090
      'ELSE''BEGIN'                                                          02036091
      WRITE(OUT,</,"NO-SOLUTION-FOR-TAUR",X4,"J=",I3>,J);                    02036100
  WRITE(OUT,</,"REP=">);WRITE(OUT,/,REP);                                    02036101
              WRITE(OUT,</,"TAUR=",X6,"NULR=",/>);                           02036102
   'FOR'TAUR:=.2,1,2,3,4,5,6,7,8,9,10,11                                     02036103
   'DO'WRITE(OUT,</,F8.4,X6,F8.4>,TAUR*TR,NULR(TR*TAUR));                    02036104
       ALARM('('NO-SOLUTION-FOR-TAUR')');                                    02036105
         'END';                                                              02036110
        'END';                                                               02036120
        'END';                                                               02036130
'END'PROCEDURE NEWPROF;                                                      02036500
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  02036600
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  02036700
'PROCEDURE'SIGPHI2(F1YI,F2YI,F3YI,NF1,NF2,NF3,UI,TI,CU,CT,SI,PI,A,XST,       02036800
            UDUDX,MM2,UF,DEL2,TEKEN,N,MM1,DEL1,MF,Z);                        02036900
'VALUE'A,XST,UDUDX,MM2,UF,DEL2,N,MM1,DEL1,MF;                               02037000
 'INTEGER'N,Z;                                                               02037100
'REAL'A,XST,UDUDX,MM2,UF,DEL2,MM1,DEL1,MF;                                   02037200
'BOOLEAN'TEKEN;                                                              02037300
'REAL''ARRAY'F1YI,F2YI,F3YI,NF1,NF2,NF3,UI,TI,CU,CT,SI,PI[*];                02037400
'BEGIN'                                                                      02037500
    'INTEGER'I,IN;                                                           02037600
    'REAL'SIGN,GM,TM,UM,LM,GAFM;                                             02037700
```

```
    'IF'TEKEN                                                                02037800
    'THEN''BEGIN'SIGN:=1;IN:=2'END'                                          02037900
    'ELSE''BEGIN'SIGN:=-1;IN:=1'END';                                        02038000
         'FOR'I:=IN+Z-1'STEP'1'UNTIL'N-1                                     02038100
    'DO''BEGIN'                                                              02038200
            GM:=(F2YI[I]*MM1/UF+NF2[I]*MM2/UF)/2;                            02038300
            TM:=(TI[I]+CT[I])/2;                                             02038400
            UM:=(UI[I]+CU[I])/2;                                             02038500
            LM:=(F1YI[I]*DEL1+NF1[I]*DEL2)/2;                                02038600
            GAFM:=(F3YI[I]*MM1/DEL1/UF+NF3[I]*MM2/DEL2/UF)/2;                02038700
            SI[I]:=GM*MM2*MF**2+SIGN*SQRT((GM*MM2*MF**2)**2+2*TM/A*          02038800
        MF**2);                                                              02038900
            PI[I]:=TM*UI[I]-0.5*SI[I]*TI[I]                                  02039000
                   +XST*TM/UM*(UOUDX                                         02039100
                   +A*(SQRT(TM)*MF/LM+GAFM*MM2*MF**2)*SI[I]);                02039200
        'END';                                                               02039300
'END'PROCEDURE SIGPHI2;                                                      02039400
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  02039500
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  02039600
   'PROCEDURE'DEV(CU,CT,LCU,LCT,DU,DT,N,Z);                                  02039700
  'VALUE'N;                                                                  02039800
   'INTEGER'N,Z;                                                             02039900
'REAL''ARRAY'CU,CT,DU,DT,LCU,LCT[*];                                         02040000
'BEGIN'                                                                      02040100
      'INTEGER'I;                                                            02040200
           'FOR'I:=0'STEP'1'UNTIL'Z-1'DO'                                    02040310
      DT[0]:=(CT[0]-LCT[0])/LCT[0];                                          02040320
           'FOR'I:=Z'STEP'1'UNTIL'N                                          02040400
    'DO''BEGIN'                                                              02040500
       DU[I]:=(CU[I]-LCU[I])/LCU[I];                                         02040600
       DT[I]:=(CT[I]-LCT[I])/LCT[I];                                         02040700
        'END'                                                                02040800
'END'PROCEDURE DEV;                                                          02040900
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  02041000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  02041100
```

```
'PROCEDURE'READDATA(FILE,N,X,Y);                                    02041200
'VALUE'N;                                                           02041300
'INTEGER'N;                                                         02041400
'ARRAY'X,Y[*];                                                      02041500
'FILE'FILE;                                                         02041600
'BEGIN'                                                             02041700
    'INTEGER'K;                                                     02041800
            READ (FILE,*,'FOR'K:=0'STEP'1'UNTIL'N'DO'X[K]);         02041900
            READ (FILE,*,'FOR'K:=0'STEP'1'UNTIL'N'DO'Y[K]);         02042000
'END'PROCEDURE READDATA;                                            02042100
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  02042200
  'REAL''PROCEDURE'VEL(YP,UT);                                      02042201
  'VALUE'YP;                                                        02042202
  'REAL'YP,UT;                                                      02042203
  'BEGIN'                                                           02042204
      VEL:=UT+EXP(-AA)*(EXP(KK*UT)-1-KK*UT-                         02042205
           (KK*UT)**2/2-                                            02042206
           (KK*UT)**3/6-                                            02042207
           (KK*UT)**4/24-                                           02042208
           (KK*UT)**5/120-                                          02042209
           (KK*UT)**6/720)-YP;                                      02042210
  'END' PROCEDURE VEL;                                              02042211
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  02042300
 'PROCEDURE'WRITEDATA(FILE,N,X,Y);                                  02042400
 'VALUE'N;                                                          02042410
 'INTEGER'N;                                                        02042420
 'ARRAY'X,Y[*];                                                     02042430
 'FILE'FILE;                                                        02042440
 'BEGIN'                                                            02042450
     'INTEGER'K;                                                    02042460
            WRITE(FILE,*,'FOR'K:=0'STEP'1'UNTIL'N'DO'X[K]);         02042470
            WRITE(FILE,*,'FOR'K:=0'STEP'1'UNTIL'N'DO'Y[K]);         02042480
            LOCK(FILE,CRUNCH);                                      02042490
'END'PROCEDURE WRITEDATA;                                           02042500
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  02042510
```

```
$ 3000000+100                                                                  03000000
'PROCEDURE'PROF(CU,CT,CV,RV,IMAX,TAN1,TAN2,F1,F2,F3,YSTEP,                     03000100
                XST,J,II,ADAP,UD,RH,                                           03000200
              MF,REDO,TDO,TDN,                                                 03000300
              'DEL1,DEL2,RC,PR ,UF,MM1,MM2,A,UDUDX,KK,AA,WR,Z,OUT,             03000400
   ALFA,RY,AS,PRINT,ACCU,MI,TR,BK,EO);                                         03000500
   'VALUE'DEL1,UF,MM1,A,UDUDX,KK,AA,XST,YSTEP,J,MF,REDO;                       03000600
'INTEGER'IMAX,J,II,Z,BK;                                                       03000700
'REAL'XST,YSTEP,DEL1,DEL2,RH,UD ,UF,MM1,MM2,A,UDUDX,KK,AA,MF,TDN,TDO,         03000800
   ACCU,MI,TR,EO,                                                              03000810
 REDO,ALFA,RY;                                                                 03000900
'REAL''ARRAY'CU,CT,CV,RV,TAN1,TAN2,F1,F2,F3[*];                                03001000
  'BOOLEAN'PR,WR,ADAP,RC,AS,PRINT;                                             03001100
'FILE'OUT;                                                                     03001200
'BEGIN'                                                                        03001300
   'INTEGER'I,K,IMAXA;                                                         03001400
   'REAL'T;                                                                    03001500
        'REAL'D,MX,MV;                                                         03001600
   'REAL''ARRAY'U1,U2,T1,T2,Y1,Y2, NF1,NF2,NF3,F1Y1,F1Y2,F2Y1,F2Y2,            03001700
     F3Y1,F3Y2,S1,S2,P1,P2[0:2.1*IMAX+1],                                      03001800
   LCU,LCT,OCU,OCT,DU,DT,ADU,ADT[0:(2.1*IMAX+1)'DIV'5];                        03001900
   'FOR'I:=0'STEP'1'UNTIL'IMAX'DIV'5                                           03001901
   'DO''BEGIN'                                                                 03001902
          OCU[I]:=CU[I];OCT[I]:=CT[I]                                          03001903
       'END';                                                                  03001904
 INTPROF(CU,CT,U1,T1,Y1,IMAX,'TRUE',TAN1,YSTEP,XST,Z,'FALSE',OUT);             03002000
 INTPROF(CU,CT,U2,T2,Y2,IMAX,'FALSE',TAN2,YSTEP,XST,Z,'FALSE',OUT);            03002100
   'FOR'I:=IMAX-4'STEP'1'UNTIL'IMAX-1                                          03002110
 'DO''BEGIN'                                                                   03002120
'IF'T2[I]<0'OR'(T1[I]=0'AND'T2[I]=0)                                           03002130
'THEN''BEGIN'T2[I]:=T2[I-1];WRITE(OUT,</,"IMAX=",I3,"I=",I3>,IMAX,I);          03002140
     WRITE(OUT,</,"VALUE-OF-SHEAR-STRESS-CORRECTED">);'END';                   03002150
'IF'T1[I]<0'OR'(T1[I]=0'AND'T2[I]=0)                                           03002153
'THEN''BEGIN'T1[I]:=T1[I-1];WRITE(OUT,</,"IMAX=",I3,"I=",I3>,IMAX,I);          03002154
     WRITE(OUT,</,"VALUE-OF-SHEAR-STRESS-CORRECTED",/>);'END';                 03002155
      'END';                                                                   03002160
```

```
    LINTERPOL(F1,NF1,21,(IMAX)/20);                                          03003000
    LINTERPOL(F2,NF2,21,(IMAX)/20);                                          03003100
    LINTERPOL(F3,NF3,21,(IMAX)/20);                                          03003200
   HERLINTPOL(NF1,NF2,NF3,Y1,F1Y1,F2Y1,F3Y1,UF,DEL1,MM1,YSTEP,'TRUE',       03003300
              IMAX,Z);                                                       03003400
   HERLINTPOL(NF1,NF2,NF3,Y2,F1Y2,F2Y2,F3Y2,UF,DEL1,MM1,YSTEP,'FALSE',      03003500
              IMAX,Z);                                                       03003600
   SIGPHI1(F1Y1,F2Y1,F3Y1,U1,T1,S1,P1,A,XST,UDUDX,MM1,'TRUE',               03003700
         IMAX,UF,MF,DEL1,Z);                                               03003800
   SIGPHI1(F1Y2,F2Y2,F3Y2,U2,T2,S2,P2,A,XST,UDUDX,MM1,'FALSE',              03003900
         IMAX,UF,MF,DEL1,Z);                                               03004000
    NEWPROF(S1,S2,P1,P2,T1,T2,CT,CU,KK,AA,YSTEP,IMAX    ,'FALSE',WR,       03004500
              RC,MM1,UF,F2Y2,NF2,F3Y2,NF3,PR,J,II,Y1,Y2,                     03004600
         REDO,MM1,MF,DEL1,ADAP,RH,OCU,OCT,TAN1,TAN2,                       03004700
 F1Y2,NF1,U1,U2,DEL1,XST,UDUDX,A,Z,OUT,ALFA,RY,AS,TR,BK,EO);               03004800
    DEV(CU,CT,OCU,OCT,DU,DT,IMAX'DIV'5,Z);                                   03004801
 'FOR'I:=0'STEP'1'UNTIL'IMAX'DIV'5                                           03004802
 'DO''BEGIN'                                                                 03004803
         ADU[I]:=ABS(DU[I]);                                                 03004804
         ADT[I]:=ABS(DT[I]);                                                 03004805
     'END';                                                                  03004806
              MAXVALUE(ADU,IMAX'DIV'5,K,MX);T:=MX;MV:=DU[K];                 03004807
              MAXVALUE(ADT,IMAX'DIV'5,I,MX);                                 03004808
        MX:=MAX(T,MX);                                                       03004809



      'IF'MX'EQL'T                                                           03004812
      'THEN''BEGIN'                                                          03004813
               I:=K;                                                         03004814
  'IF'PRINT'THEN'WRITE(OUT,</,"VELOCITY:",/>);                               03004815
           'END''ELSE''BEGIN'MV:=DT[I];                                      03004816
  'IF'PRINT'THEN'WRITE(OUT,</,"SHEAR-STRESS:",/>);'END';                     03004817
'IF'PRINT'THEN'WRITE(OUT,</,"I=",I3,X4,"MAX-REL-DEV=",F8.4,///>,I,MV);       03004818
```

```
  ACCU:=(1+MV)*ACCU;MI:=MI+I;                                                    03004820
       THICKNESS(CU,UD,D,K);                                                     03004900
  'IF'D-CU[K-1]'LSS'0.002*D                                                      03005000
  'THEN'K:=K-1;                                                                 03005100
   DEL2:=K*YSTEP;                                                               03005200
TDN:=CT[K];                                                                     03005300
    IMAXA:=K;                                                                   03005400
 MM2:=SQRT(CT[IMAXA'DIV'4]);                                                    03005500
    D:=0;                                                                       03005510
 'THRU'II'DO''BEGIN'                                                            03005600
                          D:=D+1;                                               03005611
    LINTERPOL(F2,NF2,21,IMAXA/20);                                              03005700
VTAN(CU,CT,NF2,CV,TAN1,TAN2,RV,A,MM2,UF,UDUDX,YSTEP,IMAX'DIV'5,MF,Z);           03005800
INTPROF(OCU,OCT,U1,T1,Y1,IMAX'DIV'5,'TRUE',TAN1,YSTEP,XST,Z,'TRUE',OUT);        03005900
INTPROF(OCU,OCT,U2,T2,Y2,IMAX'DIV'5,'FALSE',TAN2,YSTEP,XST,Z,'TRUE',            03006000
  OUT);                                                                         03006010
    LINTERPOL(F2,NF2,21,(IMAX)/20);                                             03006100
     LINTERPOL(F1,NF1,21,(IMAX)/20);                                            03006200
     LINTERPOL(F3,NF3,21,(IMAX)/20);                                            03006300
   HERLINTPOL(NF1,NF2,NF3,Y1,F1Y1,F2Y1,F3Y1,UF,DEL1,MM1,YSTEP,'TRUE',           03006400
              IMAX'DIV'5,Z);                                                    03006500
   HERLINTPOL(NF1,NF2,NF3,Y2,F1Y2,F2Y2,F3Y2,UF,DEL1,MM1,YSTEP,'FALSE',          03006600
              IMAX'DIV'5,Z);                                                    03006700
   LINTERPOL(F1,NF1,21,IMAXA/20);                                               03006800
   LINTERPOL(F2,NF2,21,IMAXA/20);                                               03006900
   LINTERPOL(F3,NF3,21,IMAXA/20);                                               03007000
   SIGPHI2(F1Y1,F2Y1,F3Y1,NF1,NF2,NF3,U1,T1,CU,CT,S1,P1,A,XST ,UDUDX,           03007100
           MM2,UF,DEL2,'TRUE',IMAX'DIV'5,MM1,DEL1,MF,Z);                        03007200
   SIGPHI2(F1Y2,F2Y2,F3Y2,NF1,NF2,NF3,U2,T2,CU,CT,S2,P2,A,XST ,UDUDX,           03007300
           MM2,UF,DEL2,'FALSE',IMAX'DIV'5,MM1,DEL1,MF,Z);                       03007400
   'FOR'I:=0'STEP'1'UNTIL'IMAX'DIV'5                                            03007500
   'DO''BEGIN'                                                                  03007600
      LCU[I]:=CU[I];LCT[I]:=CT[I]                                               03007700
       'END';                                                                   03007800
```

```
 NEWPROF(S1,S2,P1,P2,T1,T2,CT,CU,KK,AA,YSTEP,IMAX'DIV'5,'TRUE',WR,          03007900
                 RC,MM2,UF,F2Y2,NF2,F3Y2,NF3,PR,J,II,Y1,Y2,                03008000
          REDO,MM1,MF,DEL1,ADAP,RH,OCU,OCT,TAN1,TAN2,                      03008100
    F1Y2,NF1,U1,U2,DEL2,XST,UDUDX,A,Z,OUT,ALFA,RY,AS,TR,BK,EO);            03008200
       DEV(CU,CT,LCU,LCT,DU,DT,IMAX'DIV'5,Z);                            03008300
  'FOR'I:=0'STEP'1'UNTIL'IMAX'DIV'5                                      03008310
  'DO''BEGIN'                                                            03008320
          ADU[I]:=ABS(DU[I]);                                            03008330
          ADT[I]:=ABS(DT[I]);                                            03008340
      'END';                                                             03008350
               MAXVALUE(ADU,IMAX'DIV'5,K,MX);T:=MX;MV:=DU[K];            03008400
               MAXVALUE(ADT,IMAX'DIV'5,I,MX);                            03008410
        MX:=MAX(T,MX);                                                   03008420
       'IF'MX'EQL'T                                                      03008502
       'THEN''BEGIN'                                                     03008503
                I:=K;                                                    03008504
  'IF'PRINT'THEN'WRITE(OUT,</,"VELOCITY:",/>);                           03008505
            'END''ELSE''BEGIN'MV:=DT[I];                                 03008506
  'IF'PRINT'THEN'WRITE(OUT,</,"SHEAR-STRESS:",/>);'END';                 03008507
'IF'PRINT'THEN'WRITE(OUT,</,"I=",I3,X4,"MAX-REL-DEV=",F8.4,///>,I,MV);   03008508
  'IF'PRINT'THEN'WRITE(OUT,</,"NR-OF-INNER-IT.=",I3,/>,D);               03008510
```

```
  ACCU:=(1+MV)*ACCU;MI:=MI+I;                                            03008511
 'IF'PRINT'THEN'WRITE(OUT,</,"CUM.-REL.-DEV.=",F12.4,/,"MEAN-            03008512
   INDEX=",F8.4,/>,ACCU-1,MI/J/(1+D));                                   03008513
            'END';                                                       03008600
         IMAX:=IMAXA;                                                    03008700
'END'PROCEDURE PROF;                                                     03008800
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  03008900
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  03009000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  03009100
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  03009200
```

```
 $ 4000000+100                                                                                                    04000000
    PLOT:='FALSE';                                                                                                04000001
     TEXT:='FALSE';                                                                                               04000002
  PRINT:='FALSE';                                                                                                 04000003
  PAL:='TRUE';                                                                                                    04000004
  STORE:='FALSE';                                                                                                 04000005
DAYTIME(OUT);                                                                                                     04000100
 LOGLAW:='TRUE';                                                                                                  04000110
 PLOTL:='TRUE';                                                                                                   04000120
 PLOTH:='TRUE';                                                                                                   04000130
   AS:='FALSE';                                                                                                   04000200
      SYNTH:='FALSE';                                                                                             04000300
    Z:=1;BK:=1;                                                                                                   04000400
  NR:=20;                                                                                                         04000500
  SC:='FALSE';                                                                                                    04000600
 RC:='FALSE';                                                                                                     04000700
        ROUGH:='TRUE';                                                                                            04000800
     SMOOTH:='FALSE';                                                                                             04000900
WR:='FALSE';                                                                                                      04001100
 PR:='FALSE';                                                                                                     04001200
    ADAP:='FALSE';                                                                                                04001300
        K1:='FALSE';                                                                                              04001600
       SK:='FALSE';                                                                                               04001700
     BR:='TRUE';                                                                                                  04001800
   AL:='TRUE';                                                                                                    04001900
   II:=1;                                                                                                         04002000
    IMAX0:=50;                                                                                                    04002100
    DELTA0:=4.8@-2;UF0:=5.49;MF:=1;TAU0:=0.051;CF0:=34@-4;                                                        04002200
  'IF'ROUGH'THEN''BEGIN' RH:=.0254/8;E0:=.64*RH;'END';                                                            04002210
  EKSP:=7;                                                                                                        04002300
     BLS:=.15;                                                                                                    04002400
    BL:=BLS;                                                                                                      04002500
           A:=0.15;                                                                                               04002600
        QMAX:=20;                                                                                                 04002700
  JMAX:=200;                                                                                                      04002750
```

```
  'BEGIN'                                                                    04002800
     'REAL''ARRAY'STATPR[0:QMAX],FR,                                         04002900
     UU,TT,TTAN1,TTAN2,VV,RV,RRV,CT,CU,CV,TAN1,TAN2,ABSTAN2,NF2,             04002901
      CCU,CCT[0:2.1*IMAX0+2];                                                04002902
  READDATA(UBR,21,F,UB);                                                     04003000
  READDATA(TBR,21,F,TB);                                                     04003100
   READDATA(LW,21,F,F1);                                                     04003200
READDATA(GW,21,F,F2);                                                        04003300
READDATA(GAFW,21,F,F3);                                                      04003400
  'IF'SYNTH                                                                  04003500
  'THEN''BEGIN'                                                              04003600
WRITE(OUT,</,"SYNTHETIC-PROFILES:",/,                                        04003700
"U/UF=(Y/DELTA)**(1/EKSP)",/,                                                04003800
"T/TW=1-(Y/DELTA)*(U/UF)**2",/,                                              04003900
"EKSP=",F6.2,/>,EKSP);                                                       04004000
            'FOR'I:=0'STEP'1'UNTIL'21                                        04004100
            'DO''BEGIN'                                                      04004200
```

- B.46 -

```
                  UB[I]:=0.995*(0.05*I)**(1/EKSP);                           04004300
                  TB[I]:=1-0.05*I*UB[I]**2;                                  04004400
               'END';                                                        04004500
       'END';                                                                04004600
  'FOR'UDUDX:=0                                                              04004700
  'DO''BEGIN'                                                                04004800
      Q:=J:=0;                                                               04004900
  ACCU:=1;MI:=0;                                                             04004901
 IMAX:=IMAX0;                                                                04005000
LINTERPOL(UB,CU,21,IMAX/20);                                                 04005100
LINTERPOL(TB,CT,21,IMAX/20);                                                 04005200
  'FOR'I:=0'STEP'1'UNTIL'IMAX                                                04005210
  'DO''BEGIN'                                                                04005220
         CU[I]:=CU[I]*UF0;                                                   04005230
         CT[I]:=CT[I]*TAU0;                                                  04005240
      'END';                                                                 04005250
```

```
'IF'RC'THEN''BEGIN'                                                    04005260
                READADATA(URR,IMAX,FR,CU);                             04005261
                READDATA(TRR,IMAX,FR,CT);                              04005262
            'END';                                                     04005263
    UD:=0.995*UF0;                                                     04005300
  UF:=UF0;DEL1:=DELTA0;                                                04005400
   YSTEP:=DEL1/IMAX;                                                   04005500
  'IF'ROUGH                                                            04005601
   'THEN'WRITE(OUT,</,"ROUGHNESS-CHANGE",/>);                          04005602
 'IF'BR                                                                04006400
 'THEN''BEGIN'                                                         04006500
          WRITE(OUT,</,"BRADSHAW",/>);                                 04006700
       'END';                                                          04006800
  'IF'K1                                                               04006900
'THEN''BEGIN'                                                          04007000
            WRITE(OUT,</"KESSELS",/>);                                 04007100
            X0:=40*DELTA0;                                             04007200
            KK:=.4;AA:=2.096;RED0:=1/1.57@-5;                          04007300
         'END';                                                        04007400
   TN:=TAU0;CT[0]:=TAU0;                                               04008400
  'IF'BR                                                               04009200
  'THEN''BEGIN'                                                        04009300
   RED0:=1/1.57@-5;                                                    04009400
      KK:=0.41;AA:=2.132;X0:=61*DELTA0;                                04009500
        'END';                                                         04009550
  'IF'AL'THEN''BEGIN'                                                  04009560
                X0:=2.44;KK:=.41;AA:=2.009;RED0:=1/1.4@-5;             04009561
                WRITE(OUT,</,"ANTONIA-AND-LUXTON",/>);                 04009562
   'IF'RC'THEN'X0:=3.62;                                               04009563
              'END';                                                   04009570
  'IF''NOT'RC'THEN''BEGIN'                                             04009580
  K:=ENTIER(BL*IMAX);                                                  04009601
  'FOR'I:=K'STEP'-1'UNTIL'1                                            04009602
   'DO''BEGIN''IF'SQRT(CT[0])*YSTEP*I*RED0'GEQ'40'THEN'                04009603
  CU[I]:=CU[I+1]-SQRT(CT[0])/KK*LN((I+1)/I);                           04009604
     'END';                                                            04009614
                  'END';                                               04009615
```

```
    'IF'SQRT(CT[0])*YSTEP*REDO*MF'LSS'40                                    04009620
     'THEN''BEGIN''IF''NOT'RC'THEN'                                         04009621
WRITE(OUT,</,"MESH-POINT(S)-AT-X0-BELOW-TURBULENT-CORE",//,                 04009622
  "UST*DY/NU=",F8.4,/>,SQRT(CT[0])*MF*REDO*YSTEP);                          04009623
      K:=0;                                                                 04009624
       YP:=SQRT(CT[0])*YSTEP*REDO*MF;                                       04009625
      'WHILE'K*YP'LSS'40                                                    04009626
         'DO'K:=K+1;BK:=K;                                                  04009627
   YP:=K*YP;'IF''NOT'RC'THEN'                                               04009628


 WRITE(OUT,</,"NUMBER-OF-FIRST-MESH-POINT-IN-TURBULENT-CORE:",              04009629
       I3,/,"UST*Y/NU(IN-THIS-MESH-POINT)=",F8.4,/>,K,YP);                  04009630
          'END';                                                            04009631
  'IF''NOT'RC'THEN''BEGIN'                                                  04009639
  'IF'BK>1                                                                  04009640
  'THEN''BEGIN'                                                             04009641
          'FOR'I:=BK'STEP'-1'UNTIL'1                                        04009642
             'DO''BEGIN'                                                    04009643
         YP:=SQRT(CT[0])*YSTEP*REDO*I;                                      04009644
         'IF'ZERO IN AB(VEL(YP,UT),UT,0,UF/SQRT(CT[0]),2-6,2-6)             04009645
          'THEN''BEGIN''IF'I=BK'THEN'MM:=UT*SQRT(CT[0]);                    04009646
           'IF'I<BK'THEN'CU[I]:=UT*SQRT(CT[0])+CU[BK]-MM;'END'              04009647
          'ELSE''BEGIN'WRITE(OUT,</,"NO-SOLUTION-FOR-UT",X4,"I=",I3,        04009648
                    X4,"J=",I3>,I,J);WRITE(OUT,</"UT=",X8,"VEL=",/>);       04009649
                    T:=UF/SQRT(CT[0]);                                      04009650
   'FOR'UT:=0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1                                  04009651
   'DO'WRITE(OUT,</,F8.4,X8,F8.4>,UT*T,VEL(YP,UT*T));                       04009652
           ALARM('('NO-SOLUTION-FOR-UT')');                                 04009653
                'END';                                                      04009654
                 'END';                                                     04009655
          'END';                                                            04009656
                   'END';                                                   04009657
```

```
  'IF''NOT'RC'THEN'BE:=1'ELSE'BE:=Z;                                          04009659
ME(CU,IMAX,J,YSTEP,UF,UDUDX,XST,TO,TN,THETA,DEL1,DELST,H,DIF,TDO,TDN,         04009660
                  MF,CH0,CH1,SHAPE,BE);                                      04009661
DELSTP[0]:=10*DELST/DELTA0;DQDP[0]:=0.5*DEL1/DELST;HP[0]:=H;                  04009662
   THETAP[0]:=THETA;UFP[0]:=UF;                                              04009663
  'IF'K1                                                                     04009670
  'THEN''BEGIN'                                                             04009671
           'FOR'I:=0'STEP'1'UNTIL'IMAX                                      04009672
           'DO'CT[I]:=(1-(CU[I]/UF0)**2*I*YSTEP/DELTA0)*TAU0;               04009673
       'END';                                                               04009674
   'IF'CT[IMAX]'LSS'0                                                       04009700
   'THEN''BEGIN'                                                            04009800
WRITE(OUT,</,"NEGATIVE-VALUES-OF-STRESS-AT-EDGE-CORRECTED",/>);             04009900
            I:=IMAX;                                                        04010000
            'WHILE'CT[I]'LSS'0                                              04010100
            'DO'I:=I-1;                                                     04010200
             K:=I;                                                          04010300
             'FOR'I:=K+1'STEP'1'UNTIL'IMAX                                  04010400
              'DO'CT[I]:=CT[K]*(IMAX+1-I)/(IMAX-K);                         04010500
             CT[K+1]:=(CT[K]+CT[K+2])/2;                                    04010600
         'END';                                                             04010700
         STATPR[0]:=2.54;                                                   04010800
         STATPR[1]:=2.69;                                                   04010900
         STATPR[2]:=2.95;                                                   04011000
         STATPR[3]:=3.33;                                                   04011100
         STATPR[4]:=3.58;                                                   04011200
         STATPR[5]:=4.22;                                                   04011300
        STATPR[6]:=4.88;                                                    04011400
         XTR:=2.44;                                                         04011600
        XRTR:=STATPR[1];                                                    04011700
      XMAX:=3.62;                                                           04011750
     WRITE(OUT,</,"VISCOSITY=",E10.4,/,"MF=",F8.4,/,"KK=",F8.4,/,           04011800
     "AA=",F8.4,/,"X0=",F8.4,/,"REYNOLDS-NO.=",E10.4,/>,                    04011900
     1/ RED0,MF,KK,AA,X0,RED0*X0*UF0);                                      04012000
  X:=X0;                                                                    04012100
  K:=ENTIER(BL*IMAX);                                                       04012200
   'FOR'I:=1'STEP'1'UNTIL'K                                                 04012300
    'DO''BEGIN'                                                             04012400
     F1[I]:=F1[I]*KK/0.4;                                                   04012500
         'END';                                                             04012600
```

```
    WRITE(OUT,</,"IMAX=",I3,X2,"VEL.-AT-EDGE:",F8.4,/>,IMAX,UD);          04012700
   TDD:=CT[IMAX];                                                         04012800
  MM1:=SQRT(CT[IMAX'DIV'4]);                                              04012900
    MINMAXU1[1]:=0;MINMAXU1[2]:=.5;MINMAXU1[3]:=0;MINMAXU1[4]:=20;        04012901
  MINMAXU2[1]:=0;MINMAXU2[2]:=.3;MINMAXU2[3]:=0;MINMAXU2[4]:=1.3;         04012902
  MINMAXT1[1]:=0;MINMAXT1[2]:=.1;MINMAXT1[3]:=0;MINMAXT1[4]:=10;          04012903
    COMMENTU1:='('VELOCITY-DEFECT')';                                     04012904
    COMMENTU2:='('VELOCITY')';                                            04012905
    COMMENTT1:='('SHEAR-STRESS')';                                        04012906
   TEXTXASU1:='('(Y+EO)*UST/UF/DELST')';                                  04012907
   TEXTXASU2:='('SQRT(Y)')';                                              04012908
   TEXTXAST1:='('Y')';                                                    04012909
   TEXTYASU1:='('(UF-U)/UST')';                                           04012910
   TEXTYASU2:='('L/UF')';                                                 04012911
   TEXTYAST1:='('2@3*TAU/UF/UF')';                                        04012912
  COMMENTU:='('VELOCITY')';COMMENTT:='('REYNOLDS SHEAR-STRESS')';         04013000
  COMMENC:='('FRICTION-COEFFICIENT')';                                    04013010
  COMMEND:='('BOUNDARY-LAYER-THICKNESS')';                                04013020
TEXTXASU:='('Y/DELTA')';TEXTXAST:='('Y/DELTA')';                          04013100
     TEXTYASU:='('U/UF')';TEXTYAST:='('TAU/TAUW')';                       04013200
   TEXTXASD:='('X/DELTA0')';TEXTXASC:='('RE-THETA')';                     04013210
    TEXTYASD:='('LENGTH/DELTA0')';                                        04013220
  TEXTYASC:='('CF/CF0')';                                                 04013230
   MINMAXU[1]:=0;MINMAXU[2]:=1;MINMAXU[3]:=0;MINMAXU[4]:=1;               04013300
    MINMAXT[1]:=0;MINMAXT[2]:=1;MINMAXT[3]:=0;                            04013400
        MINMAXT[4]:=2;                                                    04013500
    MINMAXC[1]:=RED0*UFP[0]*THETAP[0];MINMAXD[1]:=X0/DELTA0;              04013610
'IF'PLOT                                                                  04013700
 'THEN''BEGIN'                                                            04013800
  WRITE(OUT,<//,"PLOT-OF-PROFILES",//>);                                  04013900
TEKMARKS(DRAWINU,I,1,IMAX,YSTEP*I/DEL1,CU[I]/UF,                          04014000
                                                                          04014100
        MINMAXU,0,1,11,.15);                                              04014200
TEKCURVE(DRAWINU,I,3,IMAX,YSTEP*I/DEL1,CU[I]/UF,MINMAXU,0,1,1);           04014300
   TEKMARKS(DRAWINT,I,0,IMAX,YSTEP*I/DEL1,CT[I]/CT[0],MINMAXT,0,1,        04014400
           11,.15);                                                       04014500
```

```
   TEKCURVE(DRAWINT,I,0,IMAX,YSTEP*I/DEL1,CT[I]/CT[0],MINMAXT,          04014600
            0,1,1);                                                     04014700
 TEKAXIS2(DRAWINU,MINMAXU,1,10,10,COMMENTU,TEXTXASU,TEXTYASU);          04014800
  TEKAXIS2(DRAWINT,MINMAXT,1,10,10,COMMENTT,TEXTXAST,TEXTYAST);         04014900
  'IF'AL                                                                04014901
  'THEN''BEGIN''IF''NOT'RC'THEN'E0:=0'ELSE'E0:=.64*RH;                  04014902
 TEKMARKS(DRAWINU1,I,1,IMAX,(I*YSTEP+E0)/UF/DELST*SQRT(CT[0]),          04014903
          (UF-CU[I])/SQRT(CT[0]),MINMAXU1,0,1,11,.15);                  04014904
 TEKCURVE(DRAWINU1,I,3,IMAX,(I*YSTEP+E0)/UF/DELST*SQRT(CT[0]),          04014909
 (UF-CU[I])/SQRT(CT[0]),MINMAXU1,0,1,1);                                04014910
   TEKAXIS2(DRAWINU1,MINMAXU1,1,5,20,COMMENTU1,TEXTXASU1,TEXTYASU1);    04014915
  TEKAXIS2(DRAWINU2,MINMAXU2,1,15,13,COMMENTU2,TEXTXASU2,TEXTYASU2);    04014916
        'END';                                                          04014918
      'END';                                                            04015000
  CF:=CF0;                                                              04015100
 LINTERPOL(F2,NF2,21,IMAX/20);                                          04015200
  VTAN(CU,CT,NF2,CV,TAN1,TAN2,RV,A,MM1,UF,UDUDX,YSTEP,IMAX,MF,Z);       04015300
  WRITE(OUT,</,"UDUDX=",E13.6,///>,UDUDX);                              04015400
    'IF'TEXT                                                            04015500
    'THEN''BEGIN'                                                       04015600
    WRITE(OUT,</,"PRINT-OF-PROFILES",/,"U=",X8,"V/U=",X6,"TAU=",X2,     04015700
                 "INDEX=",/>);                                          04015800
    'FOR'I:=0'STEP'1'UNTIL'IMAX'DO'                                     04015900
    WRITE(OUT,</,3(F8.4),I3>,CU[I],RV[I],CT[I],I);                      04016000
          'END';                                                        04016100
'FOR'I:=1'STEP'1'UNTIL'3                                                04016110
```

```
  'DO''BEGIN'Y[I]:=YSTEP*(Z-1+I);RCT[I]:=CT[Z-1+I];'END';               04016120
  LEAST SQUARES POLYNOMIAL(Y,RCT,3,1,C0);                               04016130
  ALFA:=C0[1];                                                          04016140
 WRITE(OUT,</,"X0=",F8.3 ,/,"DELTA0=",F8.5,/,"DISPL_TH0=",F8.5,/,       04016200
               "THETA0=",F8.5,/,"H0=",F8.5,/,"REL_ERR._MI0=",F8.5,/,    04016300
    "G0=",F8.4,/>,X,DEL1,DELST,THETA,H,                                 04016400
 DIF,SHAPE);'IF''NOT'RC'THEN'WRITE(OUT,</,"ALPHA=",F9.4,/>,ALFA);       04016500
```

```
   K:=ENTIER(BL*IMAX);                                                    04016700
          WRITE(OUT,</,"UPPER-INDEX-OF-LOG-REGION:",I3,/>,K);             04016800
   'IF''NOT'RC'THEN'LR:=BK'ELSE'LR:=Z;                                    04016850
   'IF'K-LR+1'GTR'1                                                       04016900
    'THEN''BEGIN'                                                         04017000
   'FOR'I:=1'STEP'1'UNTIL'K-LR+1                                          04017100
'DO''BEGIN'                                                              04017200
   'IF''NOT'RC'THEN'EO:=0;Y[I]:=YSTEP*(LR-1+I)+EO;EO:=.64*RH;             04017300
                     RCU[I]:=CU[LR-1+I];                                 04017700
         LNY[I]:=LN(Y[I]);                                               04018000
     'END';                                                              04018100
    LEAST SQUARES POLYNOMIAL(LNY,RCU,K-LR+1,1,COEF);                     04018200
USTER:=COEF[1]*KK/MF;                                                    04018300
 WRITE(OUT,</,"CF0=",F8.4,X1,"CFLOG0=",F8.4,/,"REL_DEV_CFLOG0_           04018400
 ESTIMATED-WITH-LOG-LAW=",F8.4,/>,CF,(USTER/UF)**2*2*MF**2,              04018500
 ((USTER/UF)**2*2*MF**2-CF)/CF);                                         04018600
          'END'                                                          04018700
 'ELSE'WRITE(OUT,</,"LOG-REGION-TOO-SMALL-TO-ESTIMATE-UST",/>);          04018800
CFT:=0.246*10**(-0.678*H)*(UF*THETA*REDO)**(-0.268);                     04019300
 WRITE(OUT,</,"CF=",F8.4,X1,"CFT=",F8.4,X2,"REL_DEV_CFT_CALC-WITH_       04019400
              L.T.FORM=",F8.4,/>,CF,CFT,(CFT-CF)/CF);                    04019500
  MFR:=1/(5.75*LOG(UF*DELST*REDO)+3.7);                                  04019700
 WRITE(OUT,</,"CF0=",F8.4,X1,"CFR=",F8.4,X2,"REL-DEV-                    04019800
CFR-CALC-WITH-ROTTA-FORM=",F8.4,/>,CF,2*MFR**2,(2*MFR**2-CF)/CF);        04019900
   DH:=((UF/UF0*X/X0)**((H+1)/(3*H-1)))/UF*UF0*DELTA0;                    04020000
  WRITE(OUT,</,"DELTA0=",F8.4,X1,"DELTA-HINZE=",F8.4,/,                  04020100
   "REL-DEV-DH-FROM-DELTA=",F8.4,/>,DEL1,DH,(DH-DEL1)/DEL1);             04020200
  CH:=(UF/UF0*X/X0)**(-2*(H-1)/(3*H-1))*CF0;                             04020300
  WRITE(OUT,</,"CF=",F8.4,X1,"CF-HINZE=",F8.4,/,                         04020400
   "REL-DEV-CH-FROM-CF=",F8.4,/>,CF,CH,(CH-CF)/CF);                      04020500
   TEL:=0;ZJ:=0;'IF'SMOOTH'THEN'RAS:=0'ELSE'RAS:=NR+1;COUNT:=0;           04021800
     STP:=0;J1:=J2:=-2;SUBTRACT:=0;                                      04021810
```

```
     'WHILE'X'LEQ'XMAX'AND'TIME(2)'LEQ'8400                                    04021900
'DO''BEGIN'                                                                    04022000
 J:=J+1;                                                                       04022100
'IF'ADAP                                                                       04022200
'THEN''BEGIN'                                                                  04022300
 'IF'J>100'AND'TEL=0                                                           04022400
'THEN''BEGIN'                                                                  04022500
              'FOR'I:=0'STEP'1'UNTIL'IMAX                                      04022600
              'DO''BEGIN'                                                      04022700
                    CCU[I]:=CU[I];CCT[I]:=CT[I];                               04022800
                'END';                                                         04022900
    TEL:=1;X:=X0;Q:=0;DEL1:=DEL2:=DELTA0;J:=0;UF:=UF0;                         04023000
           LINTERPOL(CCU,CU,IMAX,IMAX0/IMAX);                                  04023100
           LINTERPOL(CCT,CT,IMAX,IMAX0/IMAX);                                  04023200
 IMAX:=IMAX0;Z:=1;BK:=1;YSTEP:=DEL1/IMAX;                                      04023300
  MM1:=SQRT(CT[IMAX'DIV'4]);                                                   04023400
LINTERPOL(F2,NF2,21,IMAX/20);                                                  04023500
 VTAN(CU,CT,NF2,CV,TAN1,TAN2,RV,A,MM1,UF,UDUDX,YSTEP,IMAX,MF,Z);               04023600
WRITE(OUT,</,"PROFILES_ADAPTED-TO-MODEL",/,"U=",X8,"V/U=",X6,"TAU=",X2,        04023700
               "INDEX=",/>);                                                   04023800
  'FOR'I:=0'STEP'1'UNTIL'IMAX'DO'                                              04023900
  WRITE(OUT,</,F8.4,F8.4,F8.4,I3>,CU[I],RV[I],CT[I],I);                        04024000
  'IF''NOT'RC'THEN'BE:=1'ELSE'BE:=Z;                                           04024010


 ME(CU,IMAX,J,YSTEP,UF,UDUDX,XST,TD,TN,THETA,DEL1,DELST,H,DIF,TDU,TDN,          04024100
                MF,CH0,CH1,SHAPE,BE);                                          04024200
   DELSTP[0]:=10*DELST/DELTA0;DQDP[0]:=0.5*DEL1/DELST;                         04024201
WRITE(OUT,</,"X0=",F8.3 ,/,"DELTA0-AD=",F8.5,/,"DISPL_TH0-AD=",F8.5,/,         04024300
       "THETA0-AD=",F8.5,/,"H0-AD=",F8.5,/,"REL_ERR._MI0-AD=",F8.5,/,          04024400
                  "G0-AD=",F8.4,/>,X,DEL1,DELST,THETA,H,                       04024500
      DIF,SHAPE);                                                              04024600
```

```
    'IF'SQRT(CT[0])*YSTEP*REDO*MF'LSS'40                                    04024601
     'THEN''BEGIN'.'IF''NOT'RC'THEN'                                        04024602
WRITE(OUT,</,"MESH-POINT(S)-AT-XO-BELOW-TURBULENT-CORE",/,                  04024603
  "UST*DY/NU=",F8.4,/>,SQRT(CT[0])*MF*REDO*YSTEP);                          04024604
      K:=0;                                                                 04024605
       YP:=SQRT(CT[0])*YSTEP*REDO*MF;                                       04024606
      'WHILE'K*YP'LSS'40                                                    04024607
         'DO'K:=K+1;BK:=K;                                                  04024608
   YP:=K*YP;'IF''NOT'RC'THEN'                                               04024609
 WRITE(OUT,</,"NUMBER-OF-FIRST-MESH-POINT-IN-TURBULENT-CORE:",              04024610
       I3,/,"UST*Y/NU(IN-THIS-MESH-POINT)=",F8.4,/>,K,YP);                  04024611
          'END';                                                            04024612
 CF:=2*(SQRT(CT[0])/UF)**2*MF**2;                                           04024700
   K:=ENTIER(BL*IMAX);                                                      04024900
        WRITE(OUT,</,"UPPER-INDEX-OF-LOG-REGION:",I3,/>,K);                 04025000
  'IF''NOT'RC'THEN'LR:=BK'ELSE'LR:=Z;                                       04025050
  'IF'K-LR+1'GTR'1                                                          04025100
   'THEN''BEGIN'                                                            04025200
   'FOR'I:=1'STEP'1'UNTIL'K-LR+1                                            04025300
'DO''BEGIN'                                                                 04025400
  'IF''NOT'RC'THEN'EO:=0;Y[I]:=YSTEP*(LR-1+I)+EO;EO:=.64*RH;                04025500
                    RCU[I]:=CU[LR-1+I];                                     04025900
        LNY[I]:=LN(Y[I]);                                                   04026200
     'END';                                                                 04026300
    LEAST SQUARES POLYNOMIAL(LNY,RCU,K-LR+1,1,COEF);                        04026400
 USTER:=COEF[1]*KK/MF;                                                      04026500
 WRITE(OUT,</,"CF-AD=",F8.4,X1,"CFLOG-AD=",F8.4,/,"REL_DEV_CFLOG_           04026600
 ESTIMATED-WITH-LOG-LAW=",F8.4,/>,CF,(USTER/UF)**2*2*MF**2,                 04026700
 ((USTER/UF)**2*2*MF**2-CF)/CF);                                            04026800
        'END'                                                               04026900
    'ELSE'WRITE(OUT,</,"LOG-REGION-TOO-SMALL-TO-CALCULATE-UST",/>);         04027000
 LEAST SQUARES POLYNOMIAL(Y,CT,3,1,COEF);                                   04027100
  ALFA:=COEF[1];'IF''NOT'RC'THEN'                                           04027200
WRITE(OUT,</,"ALFA=",F8.4,X6,"TOALFA-AD",F8.4,/>,ALFA,CT[1]-ALFA*YSTEP);    04027300
 CFT:=0.246*10**(-0.678*H)*(UF*THETA*REDO)**(-0.268);                       04027400
  WRITE(OUT,</,"CF-AD=",F8.4,X1,"CFT=",F8.4,X2,"REL_DEV_CFT_CALC-WITH_      04027500
              L.T.FORM=",F8.4,/>,CF,CFT,(CFT-CF)/CF);                       04027600
```

```
   MFR:=1/(5.75*LOG(UF*DELST*REDO)+3.7);                                    04027800
  WRITE(OUT,</,"CFO-AD",F8.4,X1,"CFR=",F8.4,X2,"REL-DEV-                    04027900
CFR-CALC-WITH-ROTTA-FORM=",F8.4,/>,CF,2*MFR**2,(2*MFR**2-CF)/CF);           04028000
    DH:=((UF/UFO*X/XO)**((H+1)/(3*H-1)))/UF*UFO*DELTAO;                     04028100
   WRITE(OUT,</,"DELTAO-AD.=",F8.4,X1,"DELTA-HINZE=",F8.4,/,                04028200
    "REL-DEV-DH-FROM-DELTA-AD.=",F8.4,/>,DEL1,DH,(DH-DEL1)/DEL1);           04028300
  CH:=(UF/UFO*X/XO)**(-2*(H-1)/(3*H-1))*CFO;                                04028400
   WRITE(OUT,</,"CF-AD.=",F8.4,X1,"CF-HINZE=",F8.4,/,                       04028500
    "REL-DEV-CH-FROM-CF-AD.=",F8.4,/>,CF,CH,(CH-CF)/CF);                    04028600
      'END';                                                                04028700
      'END';                                                                04028800
        IMAX:=IMAX+2;                                                       04029000
        CU[IMAX]:=CU[IMAX-1]:=UF;                                           04029100
             CT[IMAX-1]:=0;                                                 04029200
             CT[IMAX]:=0;                                                   04029300
 LINTERPOL(F2,NF2,21,IMAX/20);                                              04029400
  VTAN(CU,CT,NF2,CV,TAN1,TAN2,RV,A,MM1,UF,UDUDX,YSTEP,IMAX,MF,Z);           04029500
   MAXVALUE(TAN1,IMAX,K,MX);MT:=MX;                                         04029600
```

```
   'FOR'I:=0'STEP'1'UNTIL'IMAX'DO'ABSTAN2[I]:=ABS(TAN2[I]);                 04029601
   MAXVALUE(ABSTAN2,IMAX,K,MX);                                             04029602
  MX:=MAX(MT,MX);                                                           04029603
  XSTEP:=YSTEP/MX;                                                          04029604
X:=X+XSTEP;                                                                 04029700
  UF:=SQRT(UFO**2+2*UDUDX*(X-XO));                                          04029710
  UO:=.995*UF;                                                              04029720
'IF'ADAP                                                                    04033300
'THEN''BEGIN'                                                               04033400
   'IF'TEL=1'THEN''BEGIN'TEL:=2;WR:='TRUE';'END';                           04033500
      'END';                                                                04033600
        XST:=XSTEP;TO:=CT[0];                                               04033650
```

```
'IF'X'LEQ'STATPR[STP]'AND'STATPR[STP]-X'LEQ'XSTEP                    04033700
'THEN''BEGIN'                                                       04033710
          J1:=J+1;J2:=J+2;STP:=STP+1;                               04033720
WRITE(OUT,</,"X-PREVIOUS<=X-STAT",//,                               04033730
 "XSTEP=",F12.8,/>,XSTEP);                                          04033740
      'END'                                                         04033790
'ELSE''BEGIN'                                                       04033800
           'IF'X-STATPR[STP]<XSTEP'AND'X-STATPR[STP]>0              04033810
          'THEN''BEGIN'                                             04033820
            J1:=J;J2:=J+1;STP:=STP+1;                               04033830
WRITE(OUT,</,"X-PREVIOUS>X-STAT",//,                                04033840
 "XSTEP=",F12.8,/>,XSTEP);                                          04033850
                  'END';                                            04033900
      'END';                                                        04033910
           'IF'J=J1'THEN''BEGIN'                                    04033920
           XST:=STATPR[STP-1]-X+XSTEP;                              04033930
           X:=STATPR[STP-1];                                        04033940
           UF:=SQRT(UF0**2+2*UDUDX*(X-X0));                         04033950
           UD:=.995*UF;                                             04033960
                          'END';                                    04033970
'IF'ROUGH                                                           04034000
'THEN''BEGIN'                                                       04034010
  'IF'X'GTR'XTR                                                     04034020
'THEN'RC:='TRUE';                                                   04034030
  'IF'X'GTR'XRTR'AND'SMOOTH                                         04034040
'THEN''BEGIN'                                                       04034050
          RC:='FALSE';                                              04034060
          SC:='TRUE';                                               04034070
      'END';                                                        04034080
      'END';                                                        04034090
'IF'RC'THEN'ZJ:=ZJ+1;                                               04034100
  'IF'ROUGH'AND'SC'AND'COUNT'EQL'0                                  04034110
 'THEN''BEGIN'                                                      04034120
      WRITE(OUT,</,"TRANSITION-ROUGH-TO-SMOOTH",//,                 04034130
            "NUMBER-OF-RUNS-ON-ROUGH-SURFACE:",                     04034140
           I3,/>,ZJ);                                               04034150
```

```
      COUNT:=COUNT+1;                                                    04034160
          BL:=BLS;                                                       04034170
           Z:=1; BK:=1;                                                  04034180
          'END';                                                         04034190
 'IF'X'GTR'XRTR'AND'RAS'LEQ'NR                                           04034200
'THEN''BEGIN'                                                            04034210
            AS:='TRUE';                                                  04034220
            RAS:=RAS+1;                                                  04034230
          'END'                                                          04034240
'ELSE'AS:='FALSE';                                                       04034250
'IF'ZJ=1                                                                 04034260
'THEN'PR:='TRUE'                                                         04034270
'ELSE'PR:='FALSE';                                                       04034280
   'IF'PR                                                                04034290
```

-B.57-

```
  'THEN''BEGIN'                                                          04034300
          BL:=BLS;                                                       04034320
 WRITE(OUT,</,"ROUGH-SURFACE",/,"ROUGH-WALL-LAW:",/,                     04034330
  "U/UST=1/K*(LN((Y+EO)/H)+B",/,"ROUGHNESS-HEIGHT=",F12.8,/,             04034340
 "ERROR-IN-ORIGIN=",F12.8,/,                                             04034345
   "J=",I3,/,"Q=",I3,/>,RH,EO,J,Q);                                      04034350
       'END';                                                            04034360
     TPL:=J=J1;                                                          04047152
 'IF'TPL'THEN'                                                           04047153
  PRINT:='TRUE';                                                         04047154
 'IF'TPL                                                                 04047155
 'THEN'WRITE(OUT,</,"PRINT-START",/>);                                   04047156
  'IF'TPL'THEN'WRITE(OUT,</,"J=",I3,/>,J);                               04047157
 'IF'TPL'THEN'WRITE(OUT,</,"UF=",F8.4,/,"UDUDX=",E13.6,/>,               04047158
   UF,UDUDX);                                                            04047159
   'IF'TPL'THEN'WRITE(OUT,</,"XST=",F8.4,/>,XST);                        04047160
```

```
          PROF(CU,CT,CV,RV,IMAX,TAN1,TAN2,F1,F2,F3,YSTEP,                    04047200
                        XST,J,II,ADAP,UD,RH,                                 04047300
    MF,REDO,TDO,TDN,                                                         04047400
DEL1,DEL2,RC,PR,UF,MM1,MM2,A,UDUDX,KK,AA, WR,Z,OUT,ALFA,RY,                  04047500
      AS,PRINT,ACCU,MI,CT[0],BK,EO);                                         04047501
        TN:=CT[0];'IF' 'NOT'RC'THEN'BE:=1'ELSE'BE:=Z;                        04047510
         ME(CU,IMAX,J,YSTEP,UF,UDUDX,XST,TO,TN,THETA,DEL2,DELST,H,DIF,       04047511
          TDO,TDN,MF,                                                        04047512
                    CHO,CH1,SHAPE,BE);                                       04047513
  'IF'TEXT'AND'TPL                                                           04047600
 'THEN''BEGIN'                                                               04047601
 LINTERPOL(F2,NF2,21,IMAX/20);                                               04047602
  VTAN(CU,CT,NF2,CV,TAN1,TAN2,RV,A,MM2,UF,UDUDX,YSTEP,IMAX,MF,Z);            04047603
  WRITE(OUT,</,"UQ=",X7,"VQ/UQ=",X2,"TAUQ=",X2,"INDEX=",/>);                 04047605
  'FOR' I:=0'STEP'1'UNTIL'IMAX                                               04047606
 'DO' WRITE(OUT,</,F8.4,F8.4,F8.4,I3>,CU[I],RV[I],CT[I],I);                  04047607
         'END';                                                              04047608
     SPL:=J=J1;                                                              04047609
 'IF'PLOT'AND'SPL                                                            04047610
'THEN''BEGIN'                                                                04047611
 TEKMARKS(DRAWINU,I,1,IMAX,YSTEP*I/DEL2,CU[I]/UF,                            04047613
        MINMAXU,0,1,Q,.15);                                                  04047614
 TEKCURVE(DRAWINU,I,3,IMAX,YSTEP*I/DEL2,CU[I]/UF,MINMAXU,0,1,1);             04047615
    TEKMARKS(DRAWINT,I,0,IMAX,YSTEP*I/DEL2,CT[I]/CT[0],                      04047616
          MINMAXT,0,1,Q,.15);                                                04047617
   'IF'AL                                                                    04047620
  'THEN''BEGIN''IF''NOT'RC'THEN'EO:=0'ELSE'EO:=.64*RH;                       04047621
  TEKMARKS(DRAWINU1,I,1,IMAX,(I*YSTEP+EO)/UF/DELST*SQRT(CT[0]),              04047622
           (UF-CU[I])/SQRT(CT[0]),MINMAXU1,0,1,Q,.15);                       04047623
  TEKCURVE(DRAWINU1,I,3,IMAX,(I*YSTEP+EO)/UF/DELST*SQRT(CT[0]),              04047628
  (UF-CU[I])/SQRT(CT[0]),MINMAXU1,0,1,1);                                    04047629
  TEKMARKS(DRAWINU2,I,1,IMAX,SQRT(I*YSTEP),                                  04047630
  (Q-SUBTRACT)/10+CU[I]/UF,MINMAXU2,0,1,Q,.15);                              04047631
  TEKCURVE(DRAWINU2,I,3,IMAX,SQRT(YSTEP*I),                                  04047632
  (Q-SUBTRACT)/10+CU[I]/UF,MINMAXU2,0,1,1);                                  04047633
         'END';                                                              04047634
  Q:=Q+1;                                                                    04047640
       'END';                                                                04047698
```

```
   PRINT:='FALSE';                                                                  04047699
     'IF'Q=4'AND'PAL'THEN''BEGIN'PAL:='FALSE';SUBTRACT:=Q;                          04047750
                    'IF'AL'AND'PLOT'THEN''BEGIN'                                    04047751
    SKIPPAPER(DRAWINU2);WRITE(OUT,</,"PLOT(VEL.VS.Y**.5)SKIPPED",/>);               04047752
   TEKAXIS2(DRAWINU2,MINMAXU2,1,15,13,COMMENTU2,TEXTXASU2,TEXTYASU2);               04047753
                                          'END';                                    04047754
                              'END';                                                04047755
CF:=2*(SQRT(CT[0])/UF)**2*MF**2;                                                    04047901
```

```
   'IF'SQRT(CT[0])*YSTEP*REDO*MF'LSS'40                                             04047902
    'THEN''BEGIN''IF'TPL'AND''NOT'RC'THEN'                                          04047903
  WRITE(OUT,</,"MESH-POINT(S)-BELOW-TURBULENT-CORE",/,                              04047904
    "UST*DY/NU=",F8.4,/>,SQRT(CT[0])*MF*REDO*YSTEP);                                04047905
      K:=0;                                                                         04047906
        YP:=SQRT(CT[0])*YSTEP*REDO*MF;                                              04047907
      'WHILE'K*YP'LSS'40                                                            04047908
         'DO'K:=K+1;BK:=K;                                                          04047909
        YP:=K*YP;'IF'TPL'AND''NOT'RC'THEN'                                          04047910
 WRITE(OUT,</,"NUMBER-OF-FIRST-MESH-POINT-IN-TURBULENT-CORE:",                      04047911
      I3,/,"UST*Y/NU(IN-THIS-MESH-POINT)=",F8.4,/>,K,YP);                           04047912
        'END';                                                                      04047913
   K:=ENTIER(BL*IMAX);'IF'TPL'THEN'                                                 04047915
         WRITE(OUT,</,"UPPER-INDEX-OF-LOG-REGION:",I3,/>,K);                        04047916
  'IF''NOT'RC'THEN'LR:=BK'ELSE'LR:=Z;'IF'K-LR+1'GTR'1                               04047917
    'THEN''BEGIN'                                                                   04047918
     'FOR'I:=1'STEP'1'UNTIL'K-LR+1                                                  04047919
'DO''BEGIN'                                                                         04047920
  'IF''NOT'RC'THEN'EO:=0;Y[I]:=YSTEP*(LR-1+I)+EO;EO:=.64*RH;                       04047921
                       RCU[I]:=CU[LR-1+I];                                          04047922
        LNY[I]:=LN(Y[I]);                                                           04047923
    'END';                                                                          04047924
     LEAST SQUARES POLYNOMIAL(LNY,RCU,K-LR+1,1,COEF);                               04047925
```

```
USTER:=COEF[1]*KK/MF;'IF'TPL'THEN'                                          04047926
WRITE(OUT,</,"CFQ=",F8.4,X1,"CFLOGQ=",F8.4,/,"REL_DEV_CFLOGQ-               04047927
ESTIMATED-WITH-LOG-LAW=",F8.4,/>,CF,(USTER/UF)**2*2*MF**2,                  04047928
((USTER/UF)**2*2*MF**2-CF)/CF);                                             04047929
        'END''ELSE''BEGIN''IF'TPL'THEN'                                     04047930
WRITE(OUT,</,"LOG-REGION-TOO-SMALL-TO-CALCULATE-UST",/>);'END';             04047931
'IF'WR'OR''NOT'ADAP                                                         04047932
'THEN''BEGIN''IF'TPL'THEN'                                                  04047933
 WRITE(OUT,<//,"XSTAT=",F8.4, /,"DELTA=",F8.5,/,"DISPL-TH.=",F8.5,/,        04047934
            "THETA=",F8.5,/,"H=",F8.5,/,"REL._ERR._ME=",F8.5,               04047935
                          /,"GQ=",F8.4,                                     04047936
  X6,"J=",I3,X6,"IMAX=",I3,X6,"TO=",F8.4,X6,/,                              04047937
 "TN=",F8.4,X6,"TDO=",F8.4,X6,"TDN=",F8.4,X6,"THETA2=",F8.4,/>,             04047938
  X,DEL2,DELST,THETA,H,DIF,SHAPE,                                           04047939
  J,IMAX,TO,TN,TDO,TDN,CH1);                                                04047940
 'IF''NOT'RC'AND'TPL'THEN'WRITE(OUT,</,"ALPHA=",F9.4,/>,ALFA);              04047941
CFT:=0.246*10**(-0.678*H)*(UF*THETA*REDO)**(-0.268);'IF'TPL'THEN'           04047942
WRITE(OUT,</,"CF=",F8.4,X1,"CFT=",F8.4,X2,"REL_DEV_CFT_CALC-                04047943
      WITH_L.T.FORM=",F8.4,/>,CF,CFT,(CFT-CF)/CF);                          04047944
  MFR:=1/(5.75*LOG(UF*DELST*REDO)+3.7);'IF'TPL'THEN'                        04047945
WRITE(OUT,</,"CFA=",F8.4,X1,"CFR=",F8.4,X2,"REL-DEV-                        04047946
CFR-CALC-WITH-ROTTA-FORM=",F8.4,/>,CF,2*MFR**2,(2*MFR**2-CF)/CF);           04047947
DH:=((UF/UF0*X/X0)**((H+1)/(3*H-1)))/UF*UF0*DELTA0;'IF'TPL'THEN'            04047948
 WRITE(OUT,</,"DELTAQ=",F8.4,X1,"DELTA-HINZE=",F8.4,/,                      04047949
 "REL-DEV-DH-FROM-DELTA=",F8.4,/>,DEL2,DH,(DH-DEL2)/DEL2);                  04047950
  CH:=(UF/UF0*X/X0)**(-2*(H-1)/(3*H-1))*CF0;'IF'TPL'THEN'                   04047951
WRITE(OUT,</,"CF=",F8.4,X1,"CF-HINZE=",F8.4,/,                              04047952
 "REL-DEV-CH-FROM-CF=",F8.4,/>,CF,CH,(CH-CF)/CF);                           04047953
  MX:=CT[IMAX'DIV'4];'IF'TPL'THEN'                                          04047954
WRITE(OUT,</,"TAUOQ=",F8.4,X2,"TMAX=",F8.4,X4,"INDEX=",F8.4,/>,             04047955
            CT[0],MX,IMAX'DIV'4);                                           04047956
        'END';                                                              04047957
      TDO:=TDN;                                                             04048000
```

```
'IF'J'MOD'10'EQL'0                                                    04048001
'THEN''BEGIN'                                                         04048002
            N:=J'OIV'10;XP[N]:=X/DELTA0;HP[N]:=H;                     04048003
    CFP[N]:=2*CT[0]/UF**2/CF0;THETAP[N]:=THETA;UFP[N]:=UF;            04048004
          K:=0;RP[N]:=RED0*THETA*UF;                                  04048005
  K:=ENTIER(BL*IMAX);                                                 04048006


 'IF''NOT'RC'THEN'LR:=BK'ELSE'LR:=Z;'IF'K-LR+1'GTR'1'AND'LOGLAW       04048007
   'THEN''BEGIN'                                                      04048008
  'FOR'I:=1'STEP'1'UNTIL'K-LR+1                                       04048009
'DO''BEGIN'                                                           04048010
 'IF''NOT'RC'THEN'E0:=0;Y[I]:=YSTEP*(LR-1+I)+E0;E0:=.64*RH;           04048014
       RCU[I]:=CU[LR-1+I];                                            04048015
       LNY[I]:=LN(Y[I]);                                              04048018
    'END';                                                            04048019
   LEAST SQUARES POLYNOMIAL(LNY,RCU,K-LR+1,1,COEF);                   04048020
 USTER:=COEF[1]*KK/MF;                                                04048021
CFLOG[N]:=2*(USTER/UF)**2/CF0;                                        04048022
        'END'                                                         04048023
   'ELSE'LOGLAW:='FALSE';                                             04048024
CFLUTI[N]:=(0.246*10**(-0.678*H)*(UF*THETA*RED0)**(-0.268))/CF0;      04048025
  MFR:=1/(5.75*LOG(UF*DELST*RED0)+3.7);CFROTTA[N]:=2*MFR**2/CF0;      04048026
CFHINZE[N]:=(UF/UF0*X/X0)**(-2*(H-1)/(3*H-1));DQDP[N]:=.5*DEL2/DELST;  04048027
  DELTAH[N]:=((UF/UF0*X/X0)**((H+1)/(3*H-1)))/UF*UF0;                 04048028
 DELTAP[N]:=DEL2/DELTA0;DELSTP[N]:=10*DELST/DELTA0;                   04048029
   WRITE(OUT,</,"J=",I3,/,"N=",I3,/>,J,N);                            04048030
  WRITE(OUT,<"PROCESSTIME:",I3,/>,TIME(2)/60);                        04048031
   WRITE(OUT,<"IO-TIME:",I3,/>,TIME(3)/60);                           04048032
       'END';                                                         04048033
  CFP[0]:=CFLUTI[0]:=CFHINZE[0]:=DELTAH[0]:=DELTAP[0]:=1;             04048034
  CFROTTA[0]:=CFLOG[0]:=1;XP[0]:=X0/DELTA0;                           04048035
  RP[0]:=RED0*THETAP[0]*UFP[0];                                       04048036
```

```
      'IF'IMAX'GEQ'2*IMAXC                                                    04048100
        'THEN''BEGIN'                                                         04048200
                    WRITE(OUT,</,"J=",I3,/,"IMAX=",I3,/>,J,IMAX);             04048300
                   IMAX:=IMAX'DIV'2;                                          04048400
                         YSTEP:=2*YSTEP;                                      04048500
                   'FOR'I:=1'STEP'1'UNTIL'IMAX                                04048600
                   'DO''BEGIN'                                                04048700
                         CU[I]:=CU[2*I];CT[I]:=CT[2*I];                       04048800
                     'END';                                                   04048900
                   WRITE(OUT,</,"X=",F8.4,/,"IMAX=",I3,/>,X,IMAX);            04049000
LINTERPOL(F2,NF2,21,IMAX/20);                                                 04049100
 VTAN(CU,CT,NF2,CV,TAN1,TAN2,RV,A,MM2,UF,UDUDX,YSTEP,IMAX,MF,Z);              04049200
     WRITE(OUT,</,"U=",X9,"V/U=",X5,"TAU=",X3,"I=",/>);                       04049300
    'FOR'I:=0'STEP'1'UNTIL'IMAX'DO'                                           04049400
     WRITE(OUT,</,3(F8.4),I3>,CU[I],RV[I],CT[I],I);                           04049500
   'IF'SQRT(CT[0])*YSTEP*REDO*MF'LSS'40                                       04049510
    'THEN''BEGIN''IF'TPL'AND''NOT'RC'THEN'                                   04049511
  WRITE(OUT,</,"MESH-POINT(S)-BELOW-TURBULENT-CORE",/,                        04049512
    "UST*DY/NU=",F8.4,/>,SQRT(CT[0])*MF*REDO*YSTEP);                          04049513
      K:=0;                                                                   04049514
        YP:=SQRT(CT[0])*YSTEP*REDO*MF;                                        04049515
      'WHILE'K*YP'LSS'40                                                      04049516
         'DO'K:=K+1;BK:=K;                                                    04049517
        YP:=K*YP;'IF'TPL'AND''NOT'RC'THEN'                                    04049518
 WRITE(OUT,</,"NUMBER-OF-FIRST-MESH-POINT-IN-TURBULENT-CORE:",                04049519
      I3,/,"UST*Y/NU(IN-THIS-MESH-POINT)=",F8.4,/>,K,YP);                     04049520
        'END';                                                                04049521
             'END';                                                           04049600
        DEL1:=DEL2;MM1:=MM2;                                                  04049700
 'IF'PR                                                                       04049800
 'THEN'WRITE(OUT,</,"NUMBER-OF-FIRST-MESH-POINT-BEYOND-ROUGHNESS=",           04049900
                I2,/>,Z);                                                     04050000
  'IF'TPL                                                                     04050010
  'THEN'WRITE(OUT,</,"END-OF-PRINT",/>);                                      04050020
    'END';                                                                    04050100
```

```
  WRITE(OUT,<"PROCESSTIME-AFTER-END-OF-LOOP:",I3,/>,TIME(2)/60);          04050110
  WRITE(OUT,<"IC-TIME-AFTER-END-OF-LOOP:",I3,/>,TIME(3)/60);              04050115



 WRITE(OUT,</,"X=",F8.4,/,"IMAX=",I3,/>,X,IMAX);                          04050120
      WRITE(OUT,</," J=",I3,X2,"Q=",I3,X2,"N=",I3,/>,J,Q,N);              04050200
     MINMAXC[3]:=0;MINMAXC[4]:=8;MINMAXD[2]:=X/DELTA0;                    04050210
  MINMAXD[3]:=1;MINMAXD[4]:=6; MINMAXC[2]:=RED0*UFP[N]*THETAP[N];         04050220
 KB:=0;KU:=N;K:=0;KC:=N;                                                  04050221
 'IF'RED0*UFP[N]*THETAP[N]<@3                                             04050222
 'OR'RED0*UFP[0]*THETAP[0]>@4                                             04050223
 'THEN'PLOTL:='FALSE';                                                    04050224
  'IF'RED0*UFP[N]*XP[N]*DELTA0<0.5@6                                      04050225
  'OR'RED0*UFP[0]*XP[0]*DELTA0>@7                                         04050226
 'THEN'PLOTH:='FALSE';                                                    04050227
 'IF'RED0*UFP[0]*THETAP[0]'GEQ'@3                                         04050228
 'THEN''BEGIN'                                                            04050229
I:=0;K:=0;                                                                04050230
'WHILE'RED0*UFP[I]*THETAP[I]'LEQ'@4'AND'I'LEQ'N                           04050231
            'DO'I:=I+1;                                                   04050232
            KC:=I-1;                                                      04050233
       'END'                                                              04050234
 'ELSE''BEGIN'                                                            04050235
            I:=0;                                                         04050236
'WHILE'RED0*UFP[I]*THETAP[I]<@3                                           04050237
            'DO'I:=I+1;                                                   04050238
            K:=I;                                                         04050239
'WHILE'RED0*UFP[I]*THETAP[I]'LEQ'@4'AND'I'LEQ'N                           04050240
            'DO'I:=I+1;                                                   04050241
            KC:=I-1;                                                      04050242
      'END';                                                              04050243
```

```
    'IF'REDO*UFP[0]*XP[0]*DELTA0'GEQ'0.536                                  04050244
   'THEN''BEGIN'                                                            04050245
              I:=0;KB:=0;                                                   04050246
   'WHILE'REDO*UFP[I]*XP[I]*DELTA0'LEQ'37'AND'I'LEQ'N                       04050247
              'DO'I:=I+1;                                                   04050248
              KU:=I-1;                                                      04050249
         'END'                                                              04050250
   'ELSE''BEGIN'                                                            04050251
              I:=0;                                                         04050252
   'WHILE'REDO*UFP[I]*XP[I]*DELTA0<0.536                                    04050253
              'DO'I:=I+1;                                                   04050254
              KB:=I;                                                        04050255
   'WHILE'REDO*UFP[I]*XP[I]*DELTA0'LEQ'37'AND'I'LEQ'N                       04050256
              'DO'I:=I+1;                                                   04050257
              KU:=I-1;                                                      04050258
        'END';                                                              04050259
  WRITE(OUT,<"PLOTH=">);WRITE (OUT,/,PLOTH);                                04050260
  WRITE(OUT,</,"PLOTL=">);WRITE (OUT,/,PLOTL);                              04050261
  'IF'PLOTH'THEN'WRITE(OUT,</,"KU-KB=",I3,/>,KU-KB);                        04050262
  'IF'PLOTL'THEN'WRITE(OUT,</,"KC-K=",I3,/>,KC-K);                          04050263
'IF'PLOT                                                                    04050300
'THEN''BEGIN'                                                               04050400
  TEKMARKS(DRAWIND,I,0,N,XP[I],DELTAP[I],MINMAXD,0,1,11,.15);               04050401
  TEKCURVE(DRAWIND,I,0,N,XP[I],DELTAP[I],MINMAXD,0,1,1);                    04050402
    'IF'PLOTH'AND'(KU-KB)>1                                                 04050403
   'THEN''BEGIN'WRITE(OUT,</,"HINZE-PLOT",/>);                              04050404
    TEKMARKS(DRAWIND,I,KB,KU,XP[I],DELTAH[I],MINMAXD,0,1,2,.15);            04050405
    TEKCURVE(DRAWIND,I,KB,KU,XP[I],DELTAH[I],MINMAXD,0,1,1);                04050406
     TEKMARKS(DRAWINC,I,KB,KU,RP[I],CFHINZE[I],MINMAXC,0,1,2,.15);          04050407
          'END';                                                            04050408
    TEKMARKS(DRAWINC,I,0,N,RP[I],CFROTTA[I],MINMAXC,0,1,3,.15);             04050409
     'IF'PLOTL'AND'(KC-K)>1'THEN''BEGIN'WRITE(OUT,</,"L.T.-PLOT",/>);       04050410
     TEKMARKS(DRAWINC,I,K,KC,RP[I],CFLUTI[I],MINMAXC,0,1,14,.15);'END';     04050411
    TEKMARKS(DRAWINC,I,0,N,RP[I],CFLOG[I],MINMAXC,0,1,12,.15);              04050412
    TEKMARKS(DRAWINC,I,0,N,RP[I],CFP[I],MINMAXC,0,1,11,.15);                04050413
```

```
    TEKAXIS2(DRAWIND,MINMAXD,1,10,10,COMMEND,TEXTXASD,TEXTYASD);          04050415
      TEKAXIS2(DRAWINC,MINMAXC,1,10,8,COMMENC,TEXTXASC,TEXTYASC);         04050416
TEKMARKS(DRAWIND,I,0,N,XP[I],THETAP[I]/THETAP[0],MINMAXD,0,1,4,.45);     04050417
   TEKMARKS(DRAWIND,I,0,N,XP[I],3*HP[I],MINMAXD,0,1,0,.15);              04050418
   TEKCURVE(DRAWIND,I,0,N,XP[I],3*HP[I],MINMAXD,0,1,1);                  04050419
TEKCURVE(DRAWIND,I,0,N,XP[I],THETAP[I]/THETAP[0],MINMAXD,0,1,1);         04050420
   'IF''NOT'ROUGH                                                        04050421
   'THEN''BEGIN'                                                         04050422
    'IF'PLOTH'AND'(KU-KB)>1'THEN'                                        04050423
     TEKCURVE(DRAWINC,I,KB,KU,RP[I],CFHINZE[I],MINMAXC,0,1,1);           04050424
   TEKCURVE(DRAWINC,I,0,N,RP[I],CFROTTA[I],MINMAXC,0,1,1);               04050425
     'IF'PLOTL'AND'(KC-K)>1'THEN'                                        04050426
     TEKCURVE(DRAWINC,I,K,KC,RP[I],CFLUTI[I],MINMAXC,0,1,1);             04050427
    TEKCURVE(DRAWINC,I,0,N,RP[I],CFLOG[I],MINMAXC,0,1,1);                04050428
    TEKCURVE(DRAWINC,I,0,N,RP[I],CFP[I],MINMAXC,0,1,1);                  04050429
          'END';                                                         04050430
SKIPPAPER(DRAWINU);                                                      04050500
 SKIPPAPER(DRAWIND);                                                     04050510
 SKIPPAPER(DRAWINC);                                                     04050520
     SKIPPAPER(DRAWINT);                                                 04050600
    SKIPPAPER(DRAWINU1);                                                 04050610
    SKIPPAPER(DRAWINU2);                                                 04050620
    SKIPPAPER(DRAWINT1);                                                 04050630
      'END';                                                             04050700
      'END';                                                             04050800
```

```
    'IF'RC'AND'STORE'THEN''BEGIN'                                          04050810
   WRITEDATA(URW,IMAX,FR,CU);                                              04050920
   WRITEDATA(TRW,IMAX,FR,CT);                                              04050921
    READDATA(URR,IMAX,FR,CU);                                              04050922
    READDATA(TRR,IMAX,FR,CT);                                              04050923
   WRITE(OUT,</,"PRINT-OF-READ-PROFILES",/,"U=",X8,"TAU=",X2,              04050924
                  "INDEX=",/>);                                            04050925
      'FOR'I:=0'STEP'1'UNTIL'IMAX'DO'                                      04050926
      WRITE(OUT,</,2(F8.4),I3>,CU[I],CT[I],I);                             04050927
    WRITE(OUT,</,"IMAX=",I3,/,"DELTA=",F8.5>,IMAX,DEL2);                   04050928
    WRITE(OUT,</,"X=",F8.5,/,"UF=",F8.5>,X,UF);                            04050929
   WRITE(OUT,</,"TAUW=",F8.5,/,"CF=",F8.5,/>,CT[0],CF);                    04050930
               'END';                                                      04050931
 'END';                                                                    04050932
    WRITE(OUT,<"PROCESSTIME-AT-END:",I3,/>,TIME(2)/60);                    04050940
   WRITE(OUT,<"IO-TIME-AT-END:",I3,/>>,TIME(3)/60);                        04050950
'END'.                                                                     04051000
```

Appendix C. The boundary-problem near the wall.

C.1 The solution of the boundary-problem near the surface.

To calculate the U-component of mean velocity, U_A, and the kinematic turbulent shear stress, τ_A, at the first mesh point, A, we have to find⊗ a relation between U_A and τ_A which can be matched to the equation along the β-characteristic (see sec. 3.2)

$$\tau_\beta U_A - \tfrac{1}{2}\sigma_\beta \tau_A = \phi_\beta \qquad (3.1.b)$$

The smooth and rough surfaces, with additional 'law of the wall'-equations will be considered

C.1.1 Smooth surface.

The 'law of the wall' looks like

$$U/u_* = f(yu_*/\nu) \qquad (C.1)$$

where u_* denotes the wall-friction velocity, $u_* = \sqrt{\tau_w}$, and ν represents the kinematic viscosity.

⊗ Note: The α-characteristic through point A may come from the viscous sublayer or transition layer, where the transport equation for the Reynolds shear stress is not valid.

We will use the mixing-length formula, as Bradshaw et al. did

$$\partial u/\partial y = \sqrt{\tau}/ky \qquad (C.2)$$

which is valid outside the viscous sublayer and the transition layer, in the fully turbulent part of the boundary layer. k is the von Kármán-constant.
The turbulent shear stress profile is assumed to be linear near the wall

$$\tau = \tau_w + \alpha y \qquad (C.3)$$

where τ_w denotes the wall-shear stress, and α is a constant with respect to y (the normal distance above the surface)
The value of α is obtained from a least-squares straight line through the first three mesh points on the τ-profile.
Substituting Eq (C.3) into Eq (C.2), we obtain

$$\partial u/\partial y = \frac{u_*}{k}\left\{\frac{\sqrt{1+\alpha y/\tau_w}}{y}\right\} \qquad (C.4)$$

Integrating this equation, and requiring compatibility with the logarithmic law

$$u/u_* = \frac{1}{k}\left\{\ln \frac{yu_*}{\nu} + A\right\} \qquad (C.5)$$

for $\alpha \to 0$, there results

$$\frac{u_A}{u_*} = \frac{1}{k}\left[\ln \frac{y_A u_*}{\nu} + A - 2\ln\left\{\frac{1+\sqrt{\tau_A/\tau_w}}{2}\right\} + 2\left\{\sqrt{\tau_A/\tau_w} - 1\right\}\right] \qquad (C.6)$$

Substituting in this equation

$$u_* = \sqrt{\tau_w} \text{ and } \tau_w = \tau_A - \alpha y_A$$

there results a relation between u_A and τ_A. (α is assumed to be known to a first approximation)
Combining this relation with Eq.(3.1.b) we obtain an equation of the form

$$f(\tau_A) = 0 \qquad (C.7)$$

This equation can be solved using a Newton-Raphson method, or a combination of Regula-Falsi and interval-halving. After τ_A has been computed, a new estimate of α and τ_w can be made. Thereafter u_A is calculated from Eq(C.6). The calculation of τ_A, τ_w, α and u_A

takes place during the re-calculation in the lower 20 % of the boundary layer.
It is required that $u_* y_A / \nu > 40$, and the first mesh point is taken as much y-steps as necessary beyond the surface to satisfy this condition⊗. One y-step is equal to $\Delta y = \delta/imax$, the distance between to consecutive mesh points. When A has to be taken several y-steps above the surface (see fig. C.1), the shear stress in the region between the point A', which lies one y-step below A, and the surface is taken constant and equal to $\tau_w = \tau_A - \alpha \Delta y$.

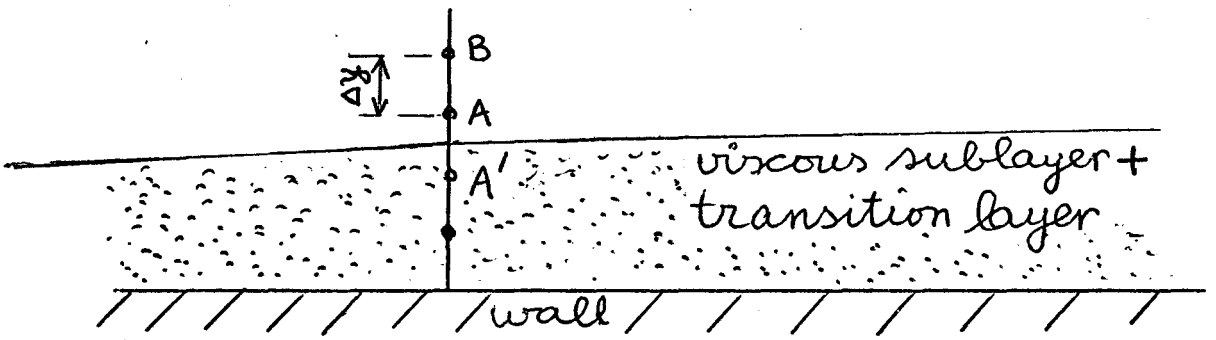


Fig. C.1 Position of the first mesh point (A) beyond the surface.

The U-component of mean velocity in the region between A' and the surface is calculated using spalding's formula

⊗ Note : Bradshaw et al. (1967) took A one y-step beyond the surface. In appendix E, it is argued why in the present calculation procedure a different method has been used.

$$y^+ = U^+ + e^{-A}\left\{e^{\kappa U^+} - \sum_{n=0}^{N} (\kappa U^+)^n/n!\right\} \quad (C.8)$$

where $y^+ = u_* y/\nu$, $U^+ = U/u_*$ and A denotes the additive constant in the log-law (Eq. (C.5)).

C.1.2 Rough surface

The procedure for a calculation of τ_A, τ_W and U_A in the case of a rough surface is similar to that for the smooth surface. The main differences are, that for the rough-wall computations the following assumptions are made

1. the turbulent shear stress between the first mesh point and the wall is taken constant and equal to the shear stress at the first mesh point.
2. it is now assumed that the viscous sublayer and the transition layer are absent. The 'law of the wall'-equation is different from that for a smooth surface.

The latter supposition means that we will only treat fully rough flows.

The effect of the roughness on the velocity profile is confined to a thin wall-layer. The form of the logarithmic velocity distribution for flow over rough walls given by Clauser (1954) reads as

$$U/u_* = \frac{1}{k} \ln \frac{y u_*}{\nu} + A - \frac{\Delta U}{u_*} \qquad (C.9)$$

$\Delta U/u_*$ is the so-called roughness-function, which is dependent on the size, distribution and geometry of the surface-roughness. For a particular roughness, $\Delta U/u_*$ can be written as a function of the flow-Reynolds number and the roughness height; for fully rough flow, the roughness-function only depends on the size of the roughness.

Two major types of roughness, each resulting in a different 'law of the wall'-equation, can be distinghuished. Perry, Schofield and Joubert (1968) report the so-called 'k' and 'd' type roughnesses, characterized by a different roughness-function.
For a 'k' type roughness Eq.(C.9) can be written as follows

$$U/u_* = 1/k \ln y/h + B \qquad (C.10)$$

where h denotes the heights of the

roughness elements, and B is a constant, which is independent of h in the case of a fully rough condition. In general, B is a function of the roughness geometry and density and $h^+ = hu_*/\nu$.
In Eqs. (C.9) and (C.10) y is measured from some distance ϵ below the crests of the roughness elements (see fig. C.2).

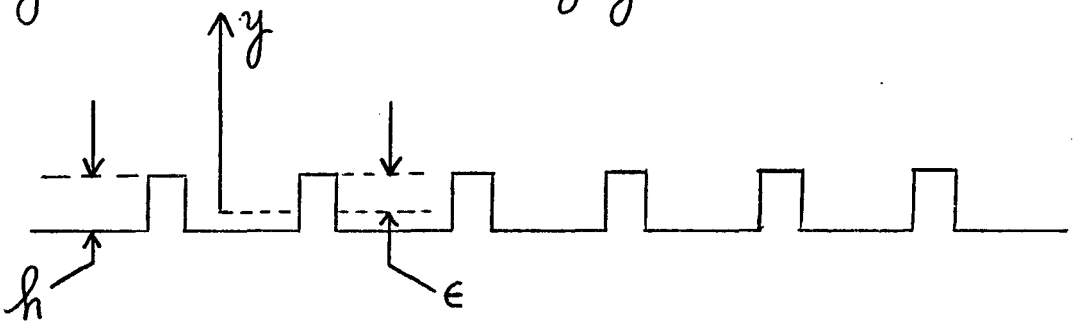


fig. C.2 'k' type roughness

ϵ is called the error in origin, which denotes that distance below the crests of the roughness elements that the effective wall of the boundary layer must be situated in order to give the usual logarithmic distribution of velocity. For flow over a 'k' type roughness, ϵ is proportional to h and relatively large compared to h. For flow over a 'd' type roughness, ϵ is independent of h.
Perry et al. found that for both types of roughness the roughness-function can be

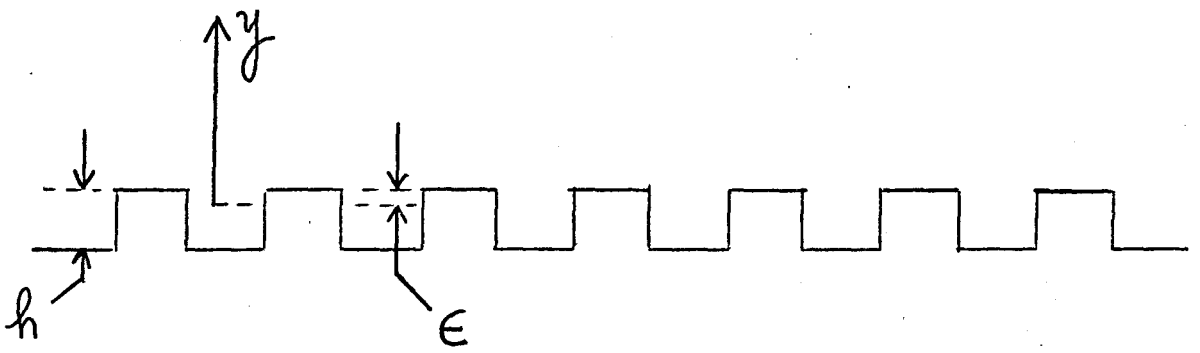


fig C.3 'd' type roughness

written as

$$\Delta U/u_* = \frac{1}{k} \ln \frac{\epsilon u_*}{\nu} + C_{k,d} \qquad (C.11)$$

$C_{k,d}$ is a constant, characteristic of the roughness.

C.2 A detailed description about handling the wall-problem in the computer program.

In the procedure NEWPROF (cards 2022500-2036500) the wall-solution is carried out. Only the smooth-wall case will be described, since the rough case differs only in the form of the 'law of the wall'-equation and setting τ_W equal to τ_A.

Eq. (C.7) is solved by calling the standard-procedure* ZERO IN AB (card 2031600), which is available in the library (memory) of the Burroughs 7700-computer. In this procedure, which is a combination of Regula Falsi and interval-halving, the zero-point of the function $f(\tau_A)$ (NUL(TAUS)) is sought. The region in which the value of TAUS is expected to lie, must also be specified by giving a lower and an upper bound. Only when the sign of NUL at the one boundary is opposite to that at the other one, then a zero-point will be found. In all other cases, the computer program will stop its calculations (cards 2032600-2033300).

* Note: a standard-procedure is a so-called intrinsic procedure, which is available in the library (memory) of the computer.

Finally, the function NUL (cards 2024200 - 2026624), will be briefly described. This function is constructed by means of a real procedure. After an estimate of the value of α (ALFA) has been obtained, by calling a standard - procedure LEAST SQUARES POLYNOMIAL (card 2024425), the value of τ_w (CT[0]) is computed (card 2024428). Eq.(C.6) is constituted (cards 2025300-2025500), to be used in a formula - derived from Eq(3.1-b), (card 2026600) - which specifies NUL

Appendix D. The initial profiles.

In this appendix the initial profiles of the U- component of mean velocity and turbulent shear stress, which have been used to carry out the calculations, are presented. The profiles have been plotted and tabulated, and have been taken from

1. The zero- pressure gradient turbulent boundary layer of Klebanoff (1955), figs. D.1.a and D.1.b, table D.1.

2. The adverse pressure gradient turbulent boundary layers of Bradshaw (1966), figs. D.2.a, D.2.b, D.3.a and D.3.b, and tables D.2 and D.3. In this type of boundary layers, the free- stream velocity is proportional to x^{α}, where α is a negative number, and x denotes the streamwise co- ordinate.

3. The low favourable pressure gradient turbulent boundary layer of Kessels (1977), figs. D.4.a.b, table D.4.

Ad 2.
All the profiles are also present in Bradshaw's paper (1967), (pp. 607, 608, 609).

Ad 3.
The turbulent shear stress profile has been calculated from the velocity profile and

$$\tau = \tau_w \{ 1 - y/\delta \, (u/u_\infty)^2 \} \qquad (4.1)$$

which formula is valid in turbulent boundary layers under zero or small pressure gradients.

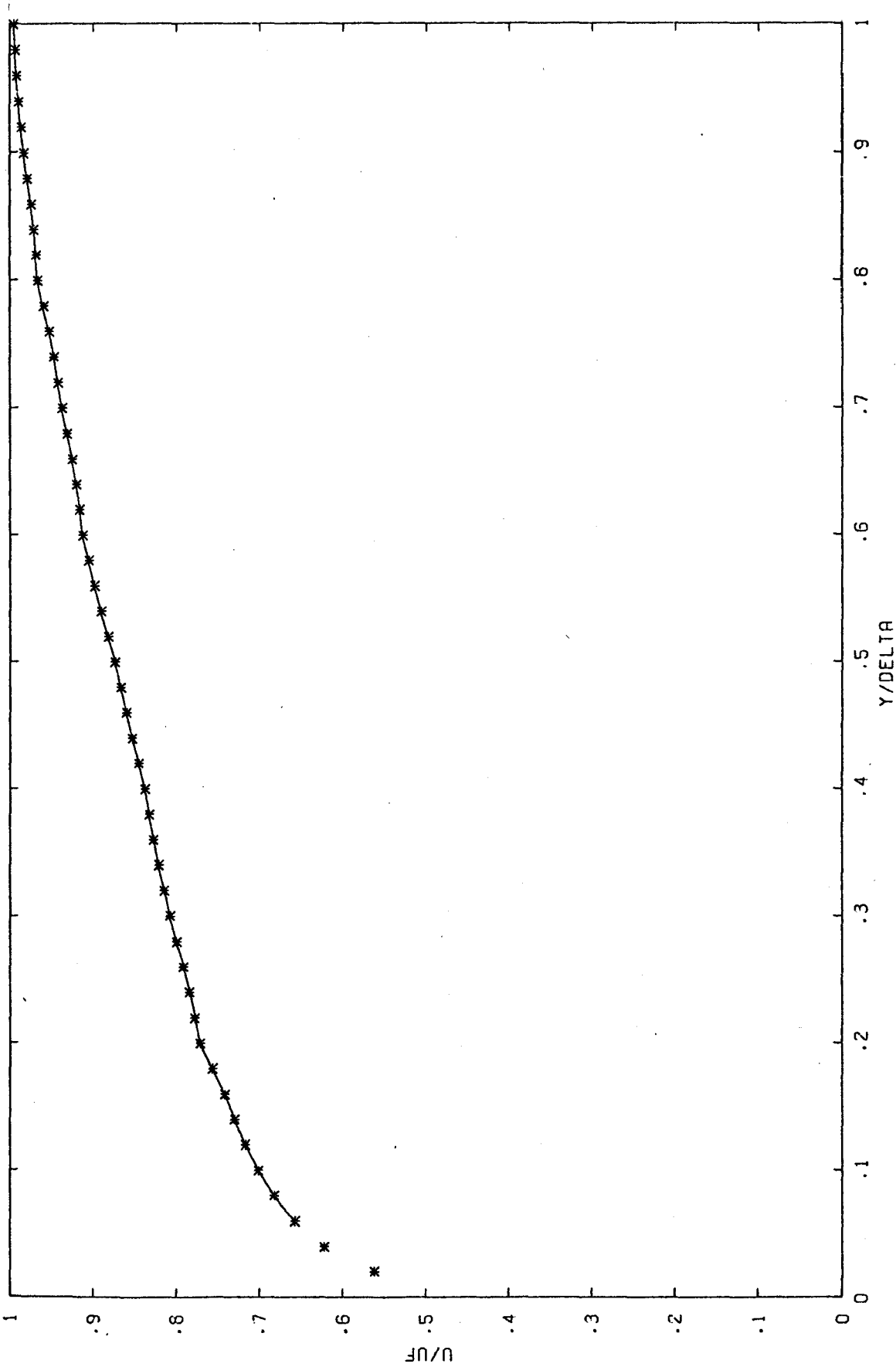


Fig. D.1.a U- component of mean velocity (Klebanoff, 1955)

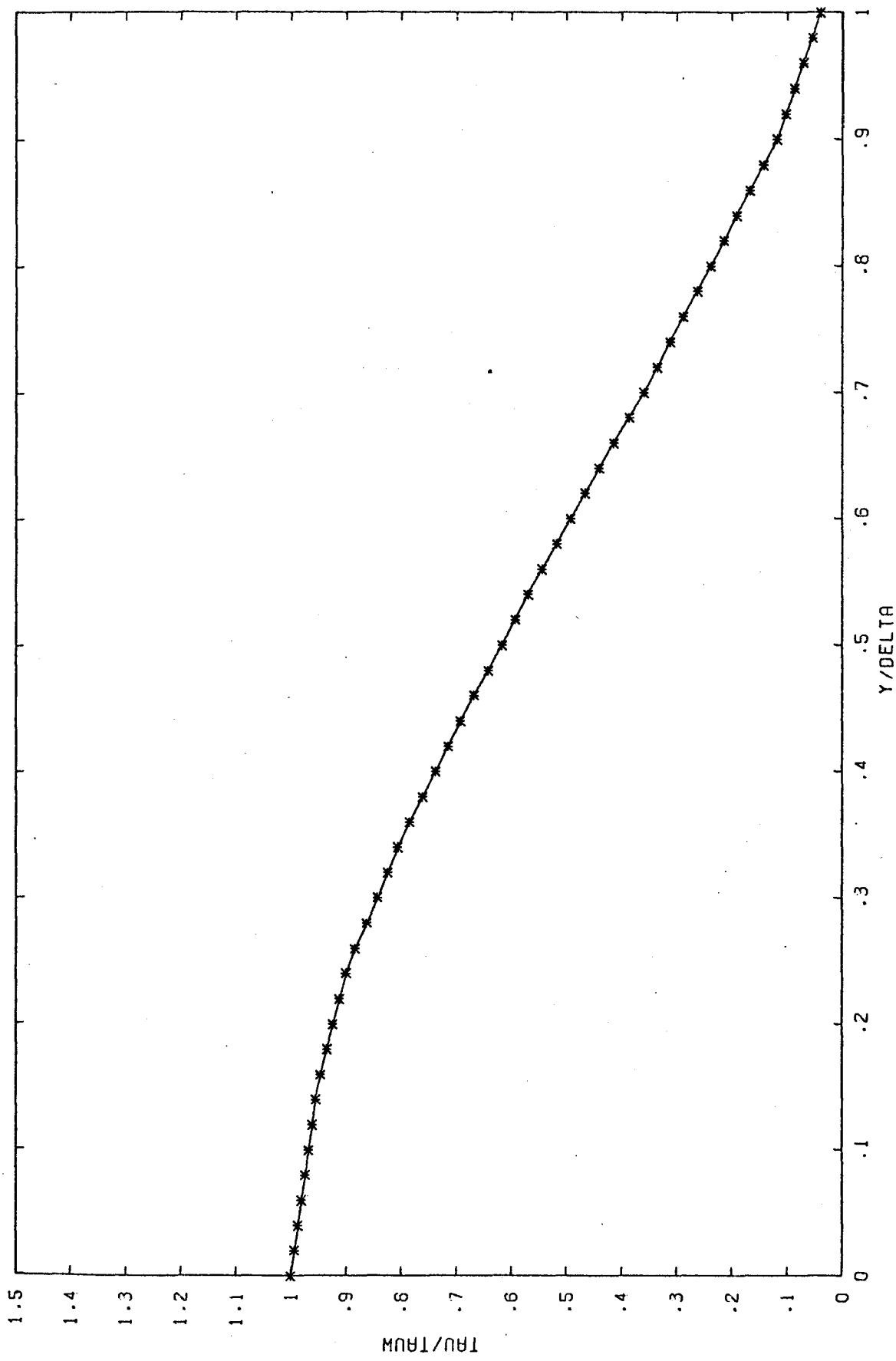


Fig. D.1.b. Turbulent shear stress (Klebanoff, 1955)

y/δ	u/u_∞	τ/τ_w
0	0	1
.05	.624	.984
.1	.698	.968
.15	.734	.952
.2	.771	.924
.25	.787	.894
.3	.807	.843
.35	.824	.797
.4	.837	.738
.45	.856	.681
.5	.873	.617
.55	.894	.558
.6	.912	.492
.65	.992	.428
.7	.937	.359
.75	.949	.301
.8	.966	.238
.85	.972	.180
.9	.983	.118
.95	.991	.078
1	.995	.039

Table D.1 Profiles of U-component of mean velocity and turbulent shear stress (Klebanoff, 1955)

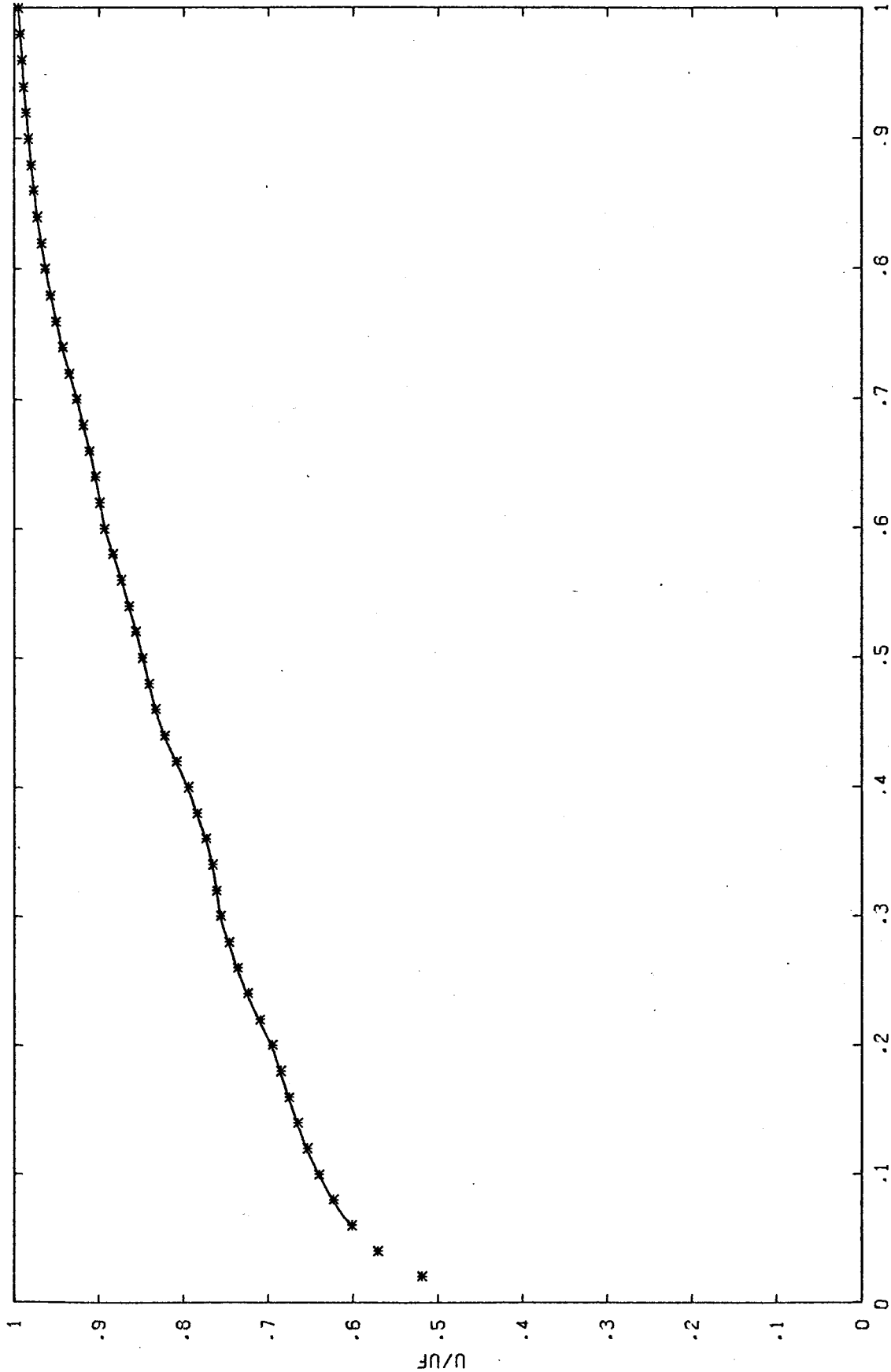


Fig. D.2.a. U-component of mean velocity ($U_\infty \sim x^{-.15}$, Bradshaw, 1966)

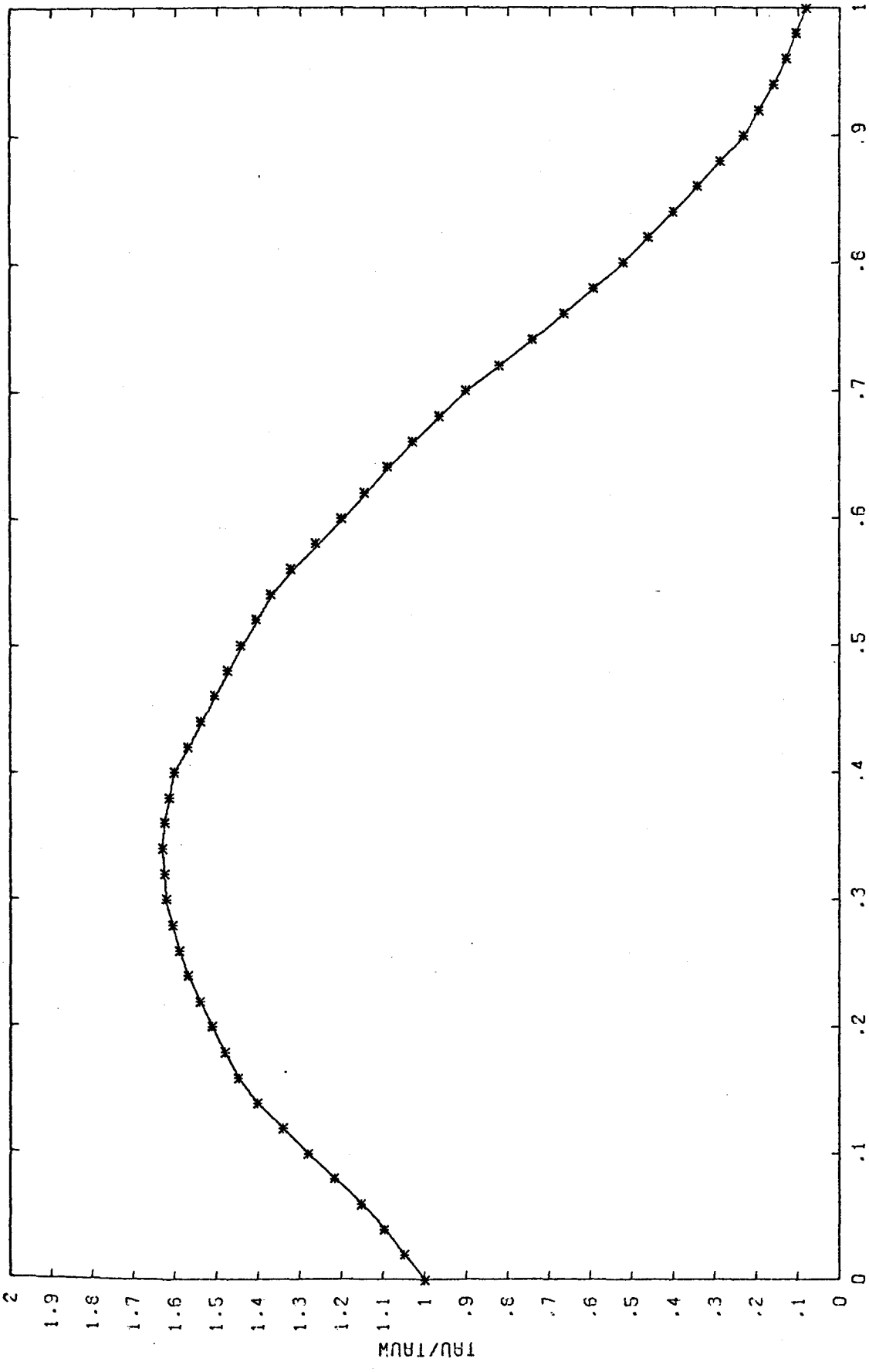


Fig. D.2.b Turbulent shear stress ($U_\infty \sim x^{-.15}$, Bradshaw, 1966)

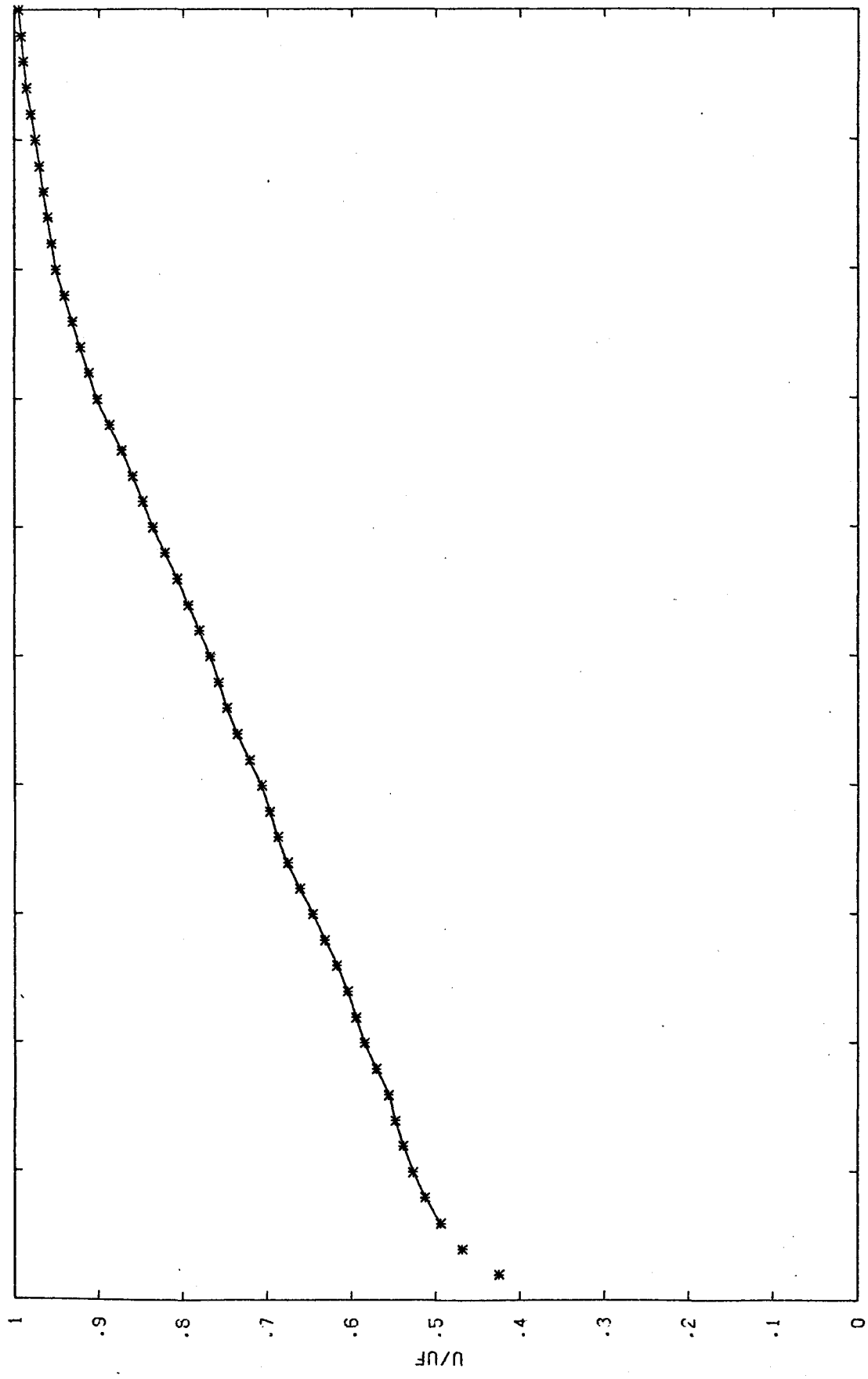


Fig. D.3.a U-component of mean velocity ($U_\infty \sim X^{-.255}$, Bradshaw, 1966)

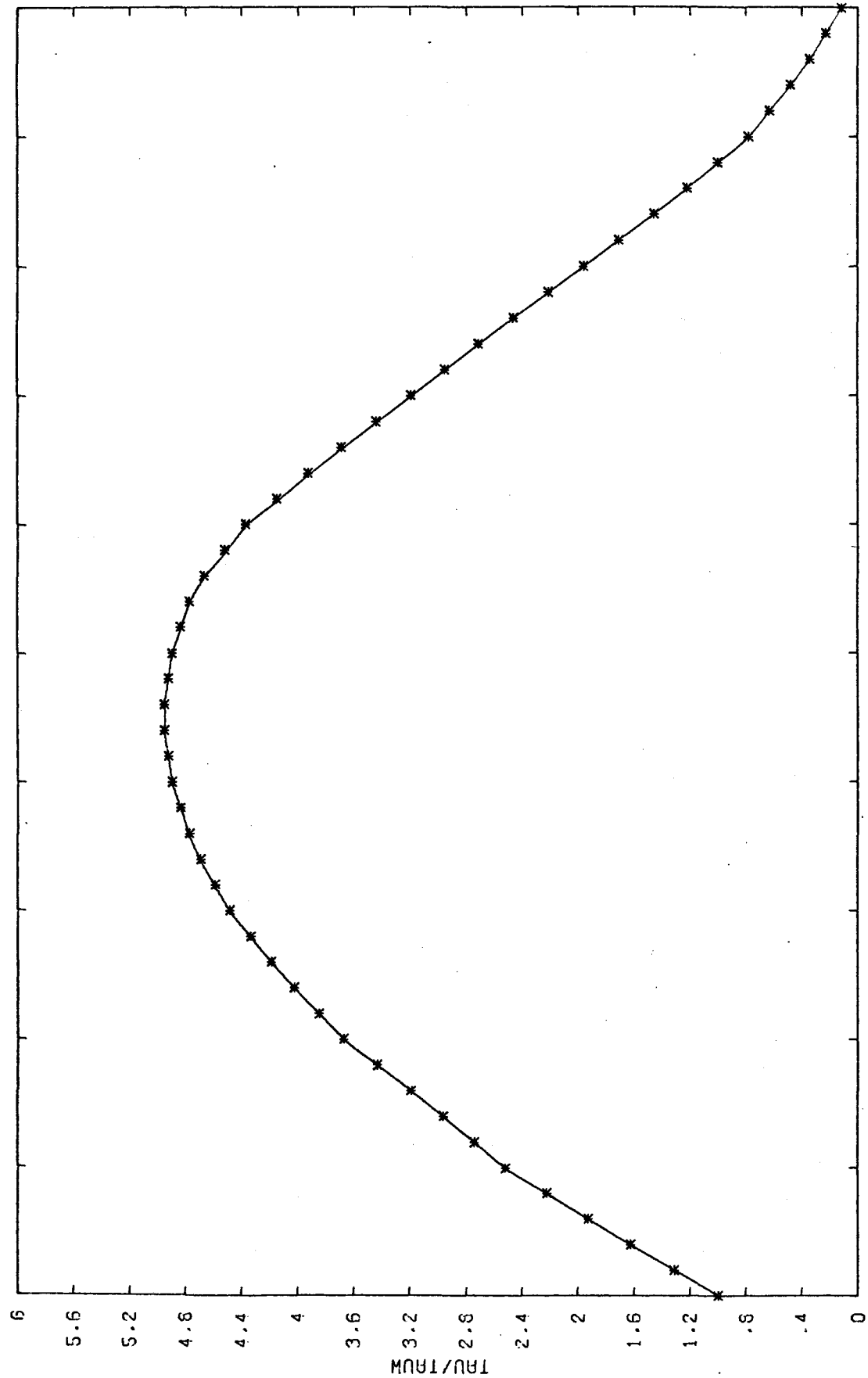


Fig. D.3.b Turbulent shear stress (U_∞ vs $x^{-.255}$, Bradshaw, 1966)

y/δ	u/u_∞	τ/τ_w
0	0	1
.05	.585	1.12
.1	.646	1.28
.15	.670	1.43
.2	.695	1.51
.25	.731	1.58
.3	.756	1.62
.35	.768	1.63
.4	.794	1.60
.45	.829	1.52
.5	.848	1.44
.55	.868	1.35
.6	.893	1.20
.65	.907	1.06
.7	.926	.90
.75	.947	.70
.8	.963	.52
.85	.975	.37
.9	.983	.23
.95	.990	.14
1	.995	.08

Table D.2 Profiles of U-component of mean velocity and turbulent shear stress (Bradshaw, 1966, $U_\infty \sim x^{-.15}$)

y/δ	u/u_∞	τ/τ_w
0	0	1
.05	.476	1.78
.1	.529	2.52
.15	.549	3.07
.2	.585	3.67
.25	.610	4.11
.3	.646	4.48
.35	.683	4.74
.4	.707	4.89
.45	.743	4.96
.5	.768	4.89
.55	.800	4.74
.6	.836	4.37
.65	.866	3.81
.7	.902	3.19
.75	.927	2.59
.8	.951	1.96
.85	.963	1.33
.9	.975	.78
.95	.988	.40
1	.995	.11

Table D.3 Profiles of U-component of mean velocity and turbulent shear stress (Bradshaw, 1966, $U_\infty \sim x^{-.255}$)

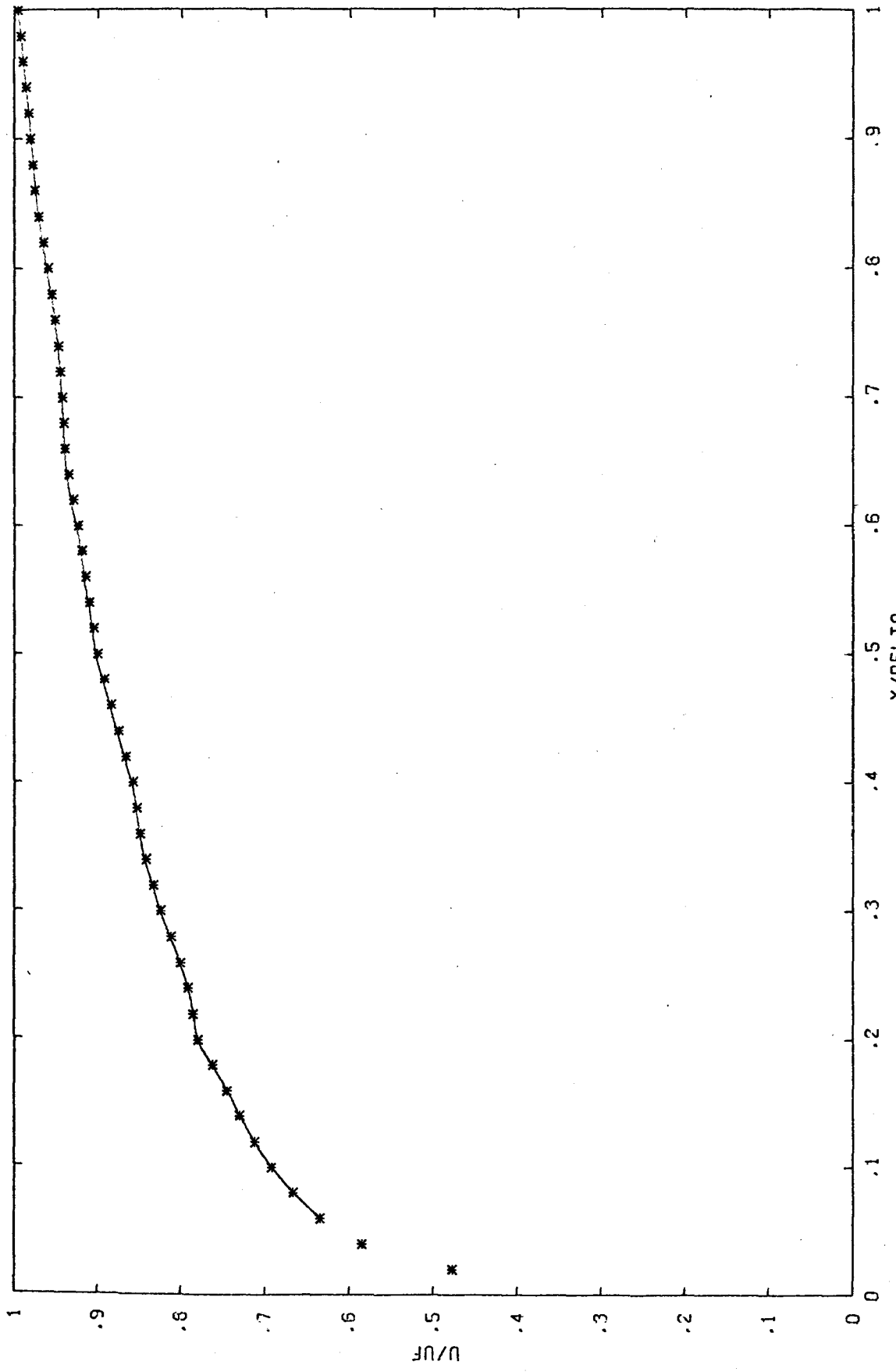


Fig. D.4.a U-component of mean velocity (Kessels, 1977)

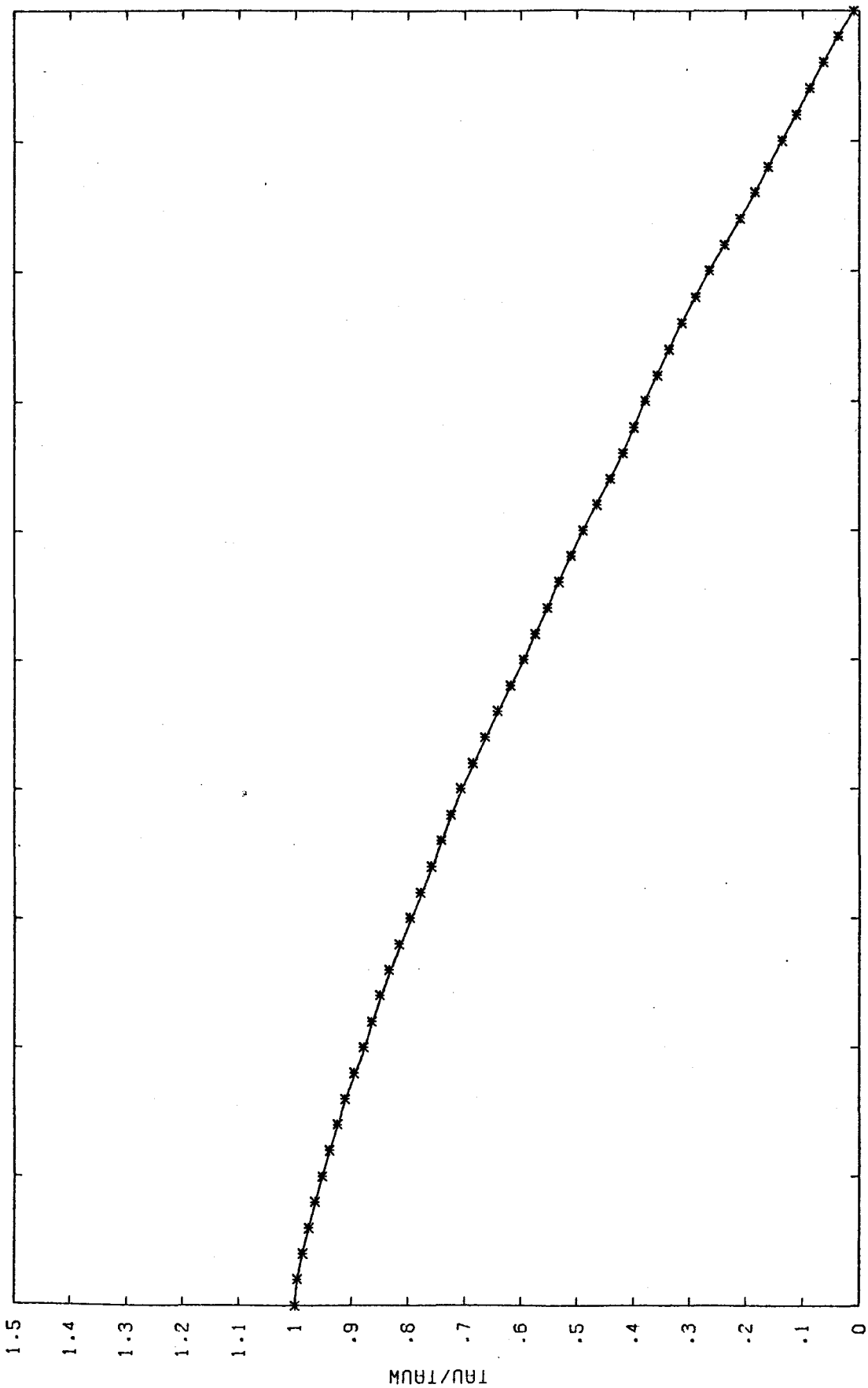


Fig. D.4.b Turbulent shear stress (calculated, Kessels, 1977)

y/δ	u/u_∞	τ/τ_w
0	0	1
.05	.623	.982
.1	.693	.951
.15	.736	.920
.2	.779	.880
.25	.794	.840
.3	.823	.796
.35	.846	.751
.4	.857	.707
.45	.879	.653
.5	.900	.596
.55	.912	.542
.6	.923	.489
.65	.938	.431
.7	.942	.378
.75	.948	.324
.8	.959	.267
.85	.973	.196
.9	.980	.138
.95	.987	.076
1	.995	.009

Table D.4 Profiles of U-component of mean velocity and turbulent shear stress. (Kessels, 1977)

Appendix E. The way Bradshaw et al. (1967) handled the calculations near a smooth surface in the case $u_* \, y_{step} / \nu < 40$.

E.1 Introduction.

In appendix C it has been described how the computations near the surface are done in the case $u_* \, \Delta y / \nu < 40$. Bradshaw et al. (1967) have argued ⊗ that the first mesh point should be taken one y-step beyond the surface. However, in the present calculation program the first mesh point is taken as many y-steps as necessary beyond the surface to lie outside the viscous sublayer and the transition layer. The differences between the two ways of handling the problem are that in Bradshaw's method the velocity in the first mesh point may be in error, and in the present method the wall-shear stress may be in error. In this appendix the data of Kessels (1977) have been used to compare the two manners of solution; the results are shown in sec. E.2.

⊗ Note: Bradshaw et al. (1967) said: 'it leads to inaccuracy and instability in extrapolating the shear stress to the surface at low Reynolds numbers when point A has to be taken several y-steps from the surface' (p. 603)

E.2 Results and discussion.

In figs. E.1 to E.3 the computational results of Bradshaw's method are shown ⊗. When compared to figs. 4.18 to 4.21 (ch.4) the differences between the results are ignorable. The momentum-error (ME) amounts to about 3 % at $x = 2.75$ m ($j=123$); the wall-shear stress value is a little larger than the value found in ch.4. It has been concluded that the difference between Bradshaw's method of handling the calculations near the surface and the present method is of no significance. The present method has been used, with a little preference, because the momentum-error is smaller (and consequently the wall-shear stress is predicted better).

Another way to handle the problem, is to take fewer mesh points on the initial profiles. Figs. E.4 to E.6 show the results in the case the number (imaxo) of mesh points on the initial profiles is equal to 19; this amount seemed to be just small enough to cause $\Delta y\, u_*/\nu$ to be and stay ≥ 40 during the calculations. It can clearly be noticed that the turbulent shear stress profile at $x = 2.75$ m ($j=39$) (fig E.4.b) shows a larger departure from the

⊗ Note: The plots of the profiles of the U-component of mean velocity and turbulent shear stress are generated at similar streamwise positions as in figs. 4.18 (ch.4).

initial profile than in the case imax0 = 50 (figs 4.18.b and E.1.b). Also the momentum-error is larger (about 8%, at x=2.75m). From these results it can be concluded that a decrease of imax0 makes the calculation results worse.

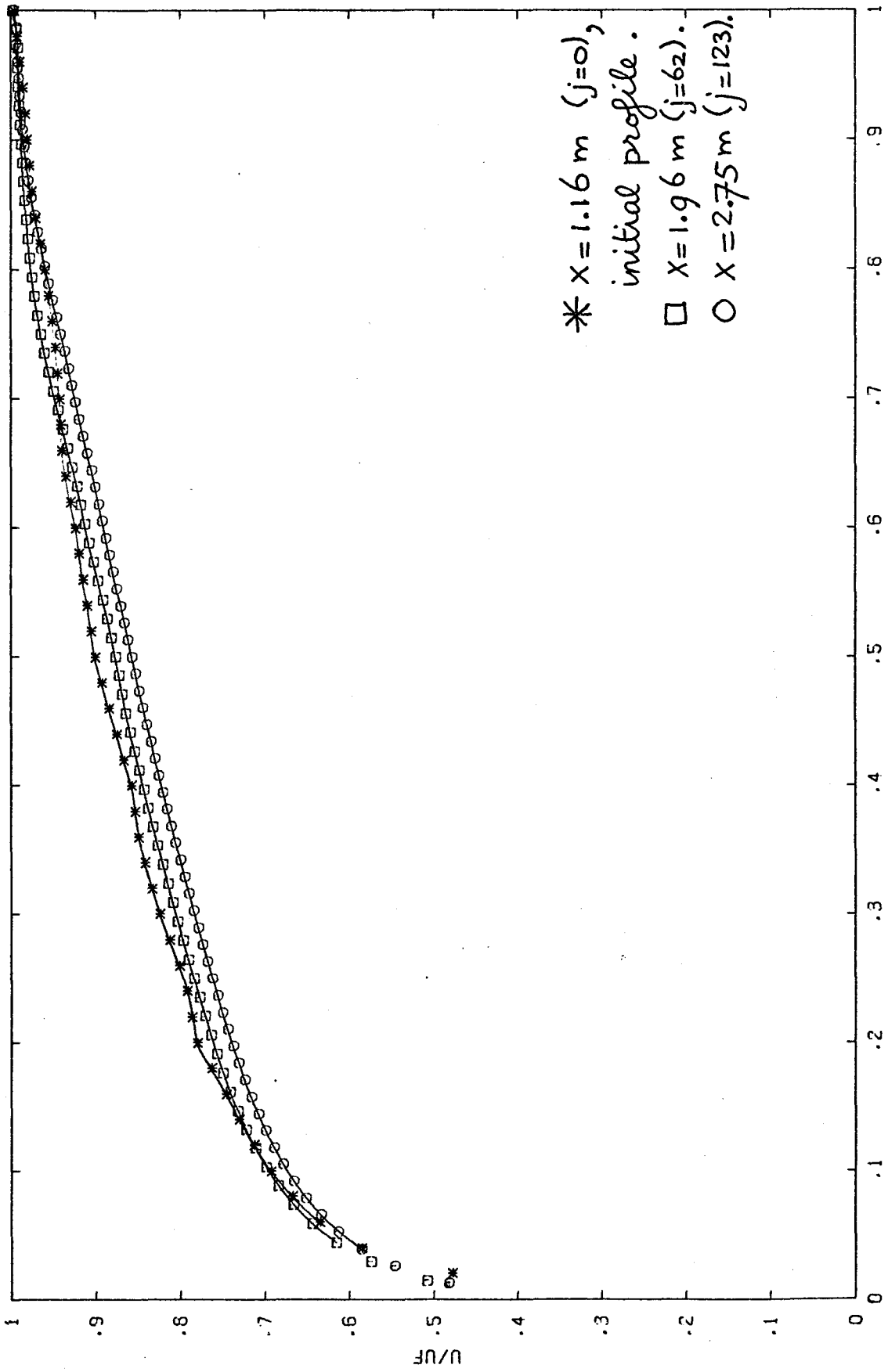


Fig. E.1.a U- component of mean velocity

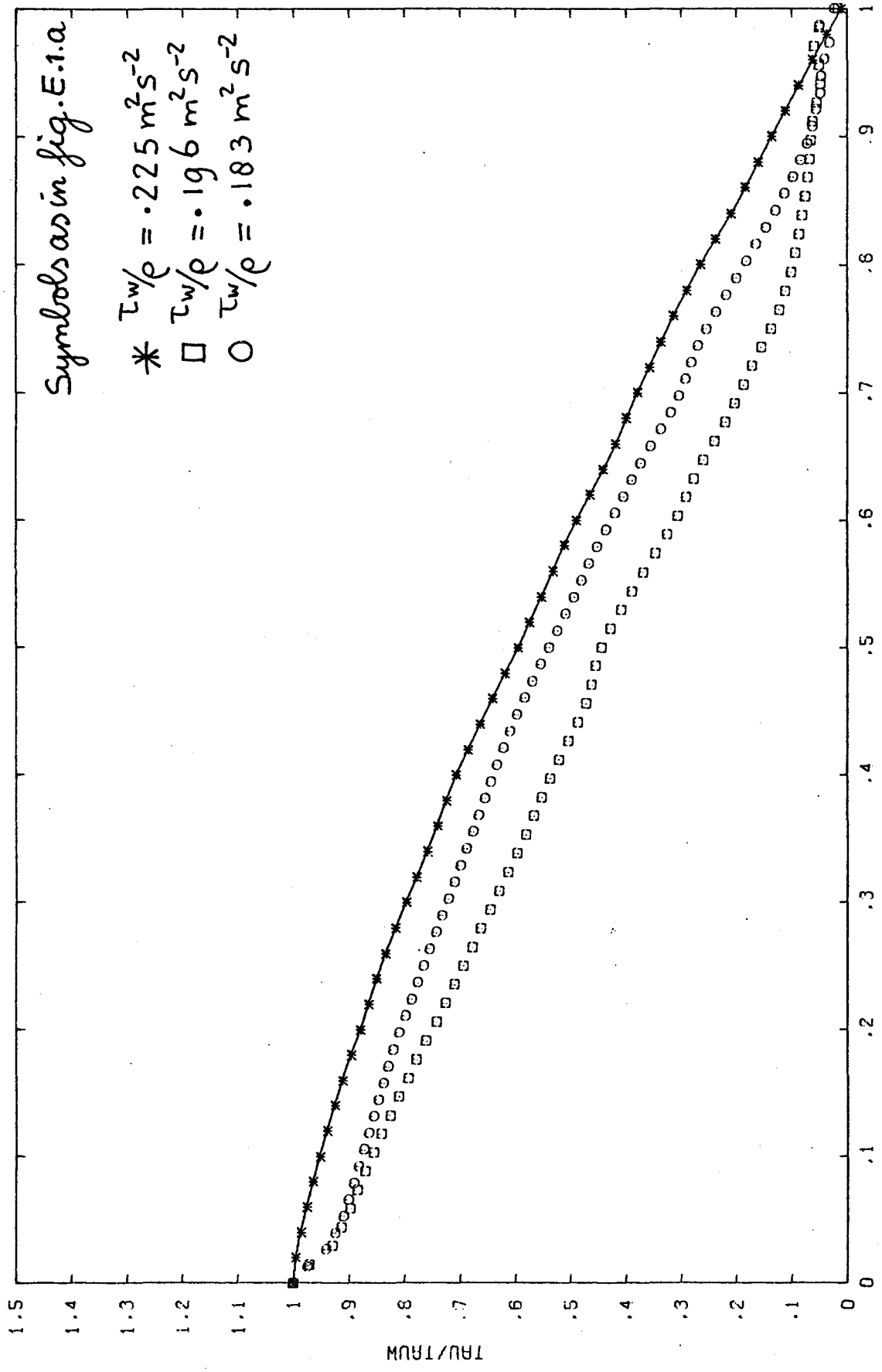


Fig. E.1.b Turbulent shear stress

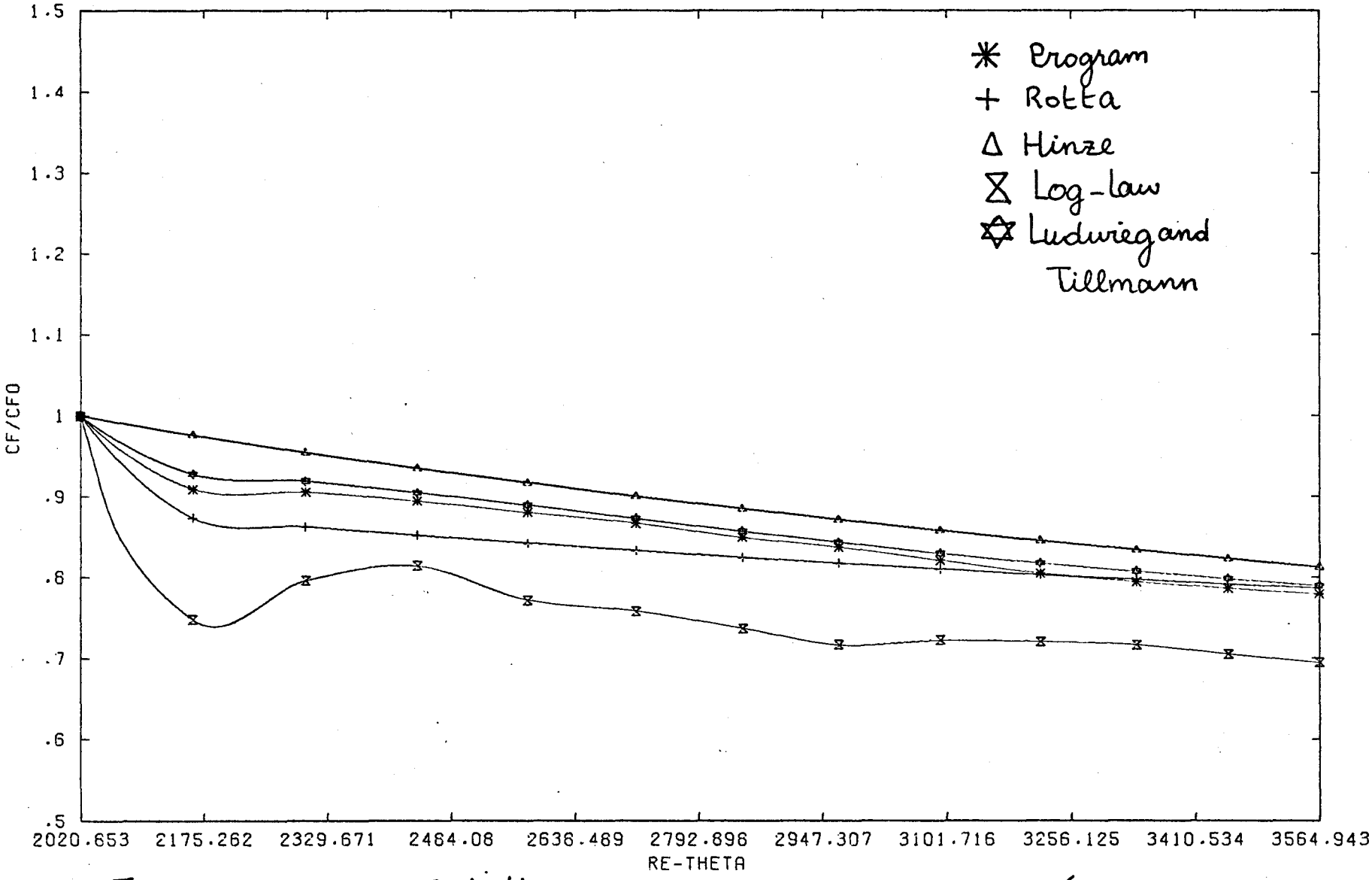


Fig. E.2 Wall-friction coefficient ($CF0 = 41 * 10^{-4}$)

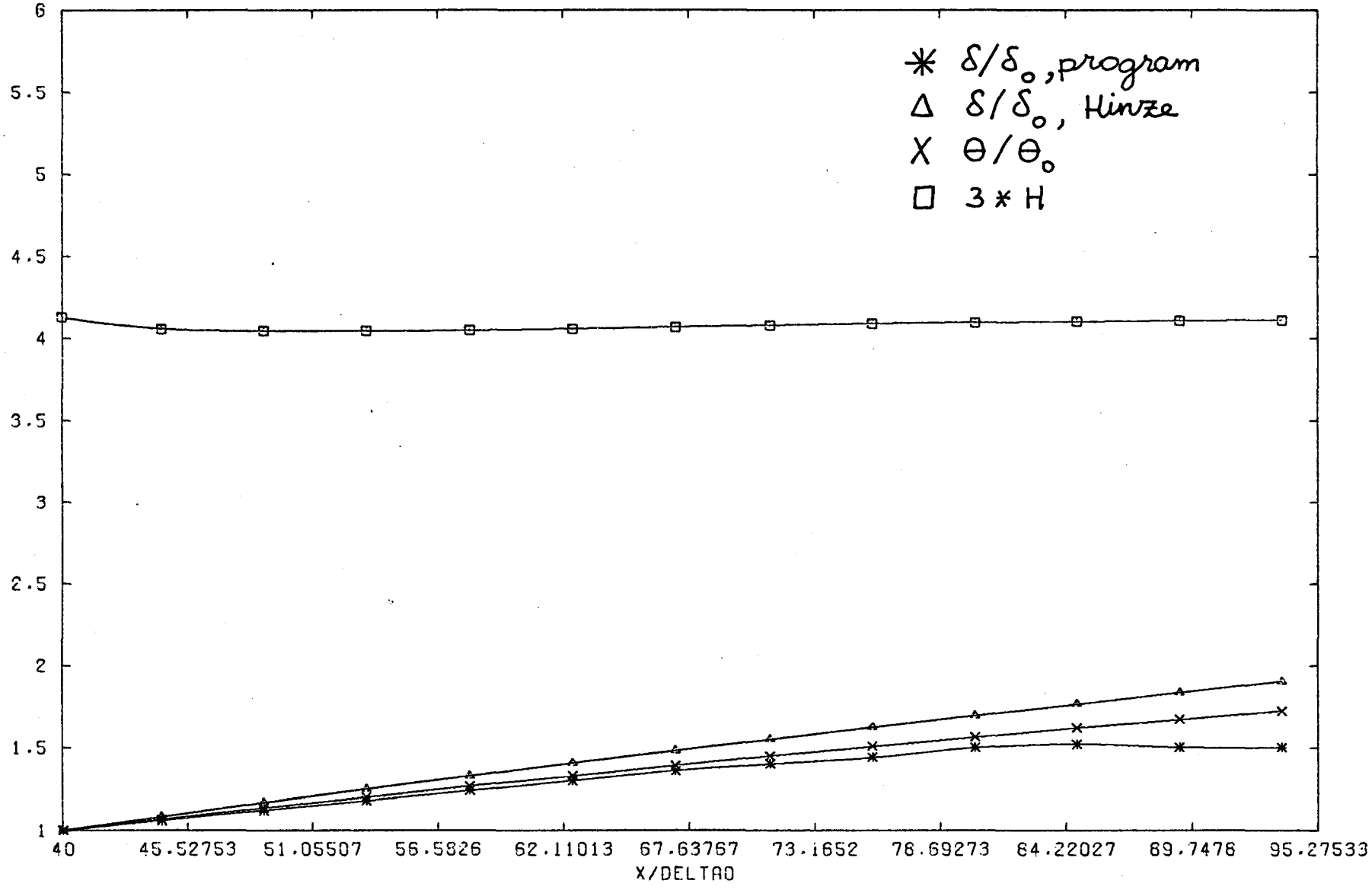


Fig. E.3 Boundary-layer growth ($\delta_0 = .029$ m)

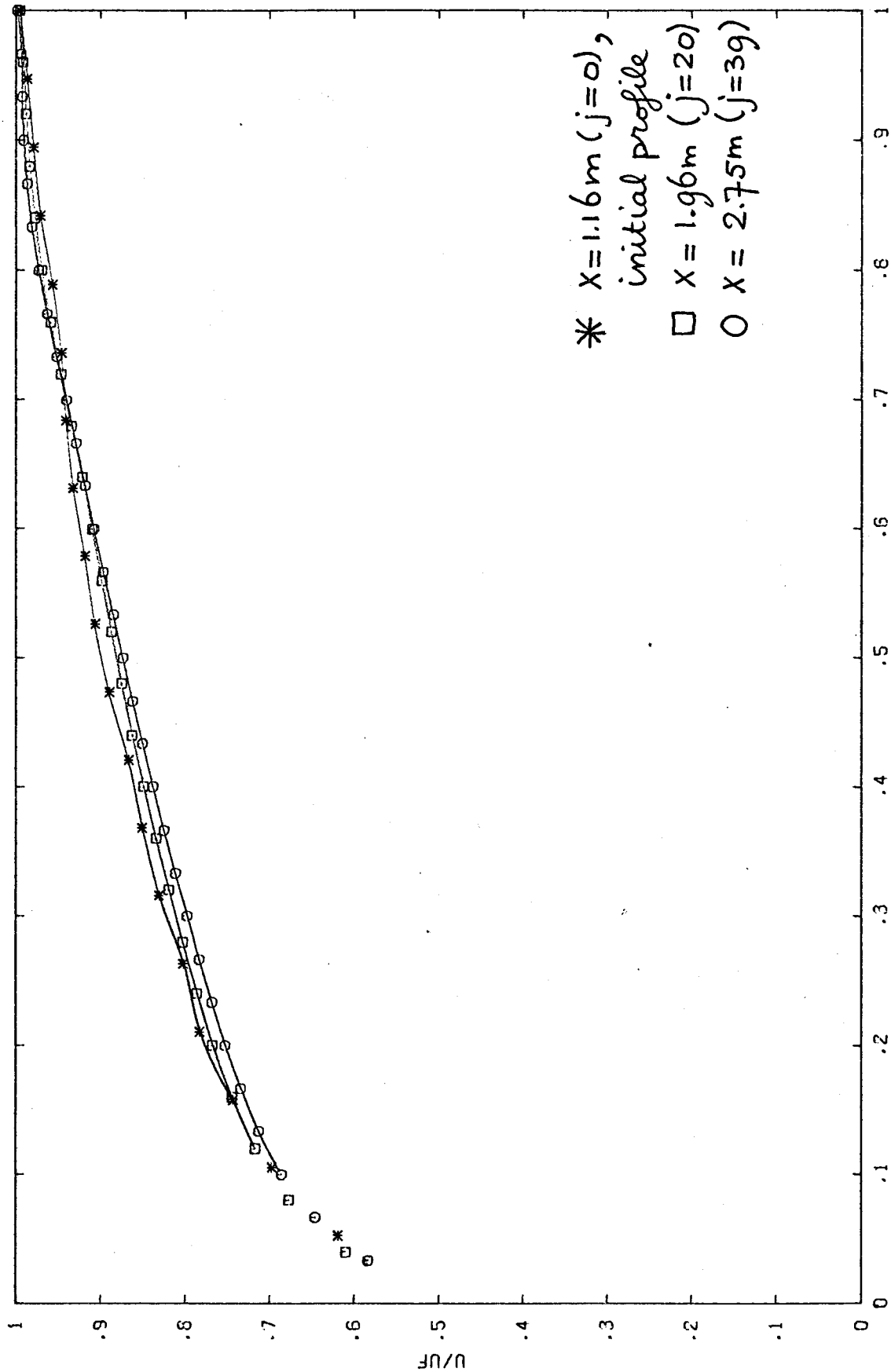


Fig. E.4.a U-component of mean velocity (imax=19)

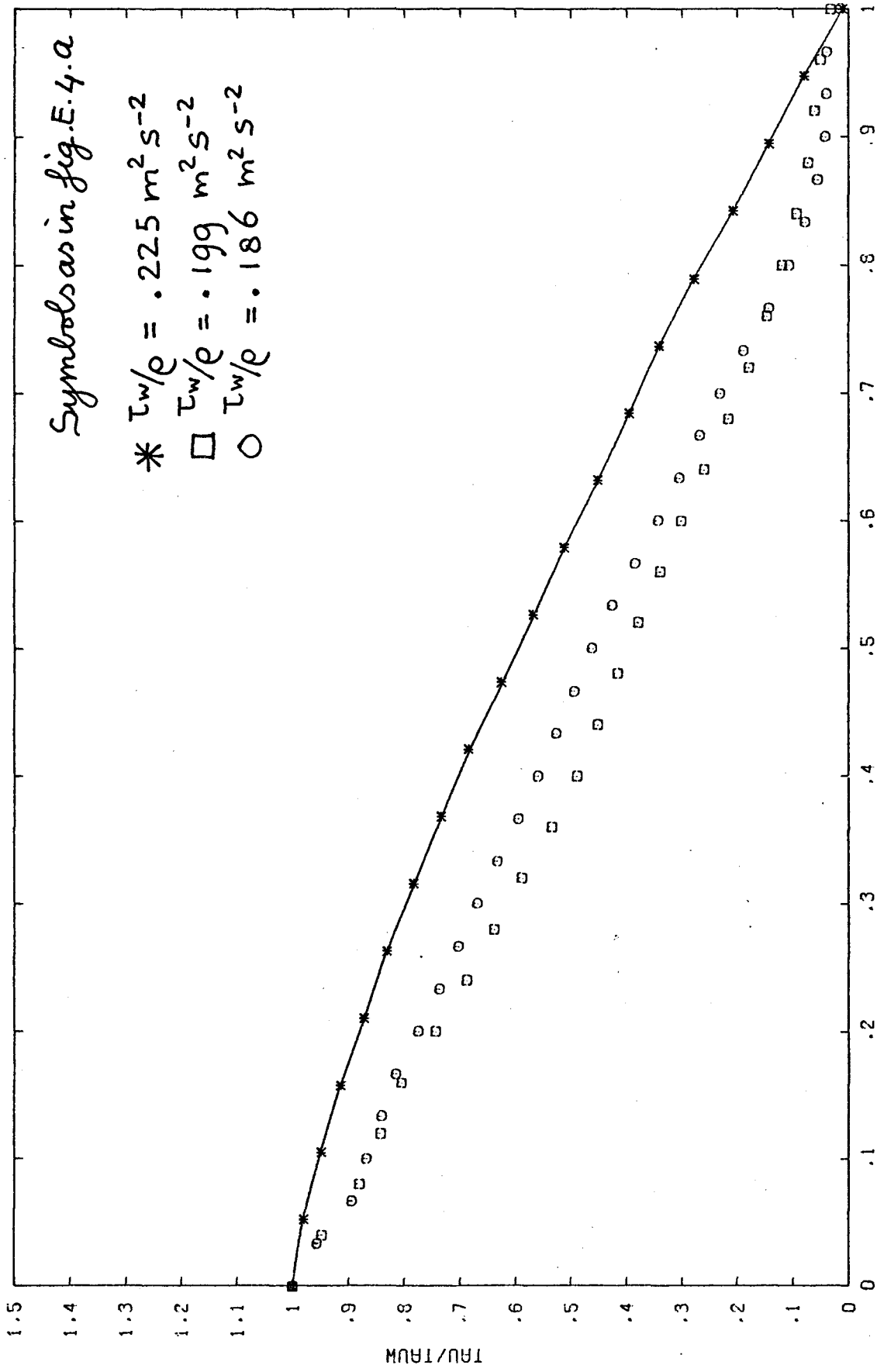


Fig. E.4.b Turbulent shear stress (imacro = 19)

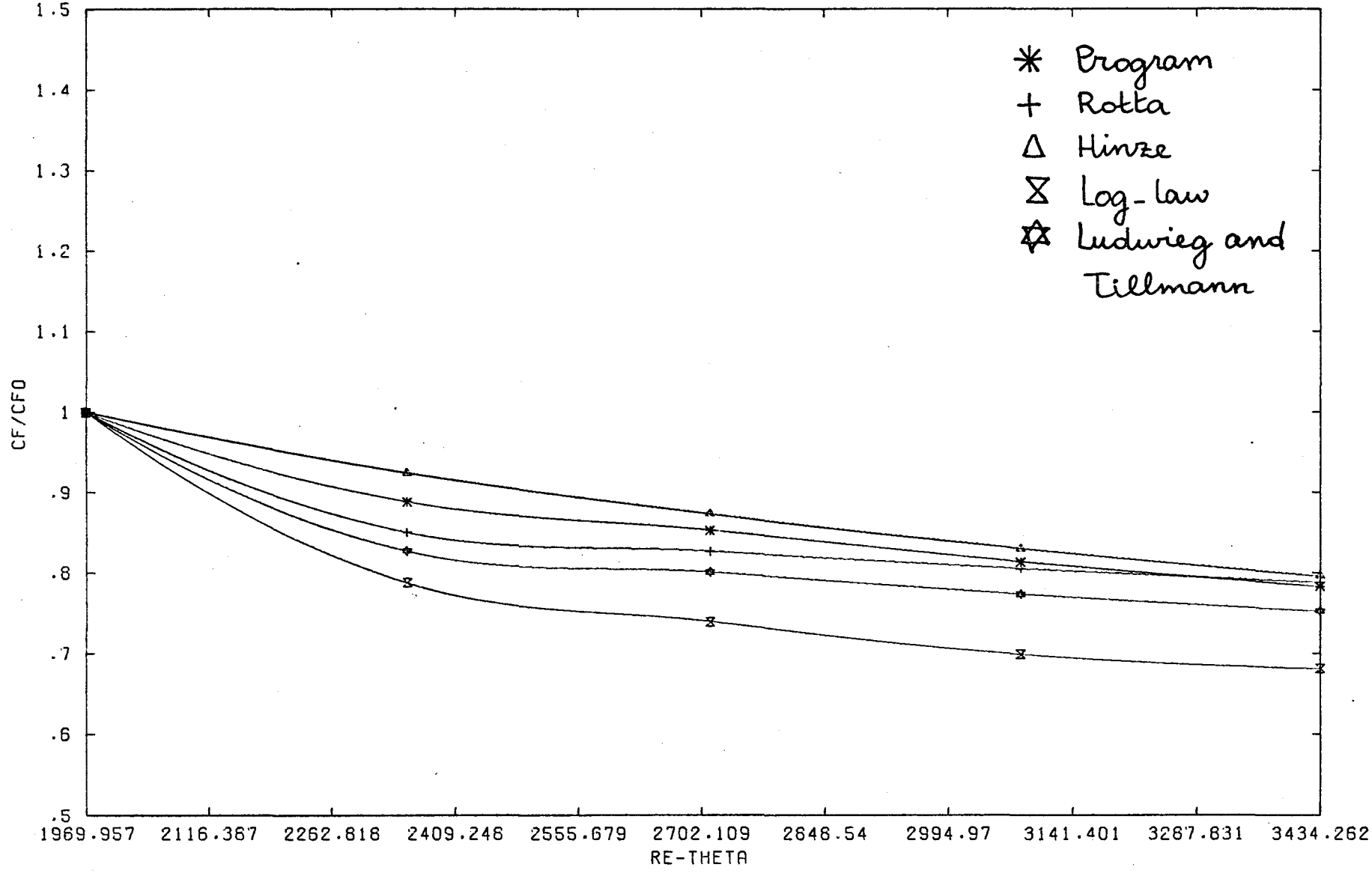


Fig. E.5 Wall-friction coefficient (imax0 = 19)

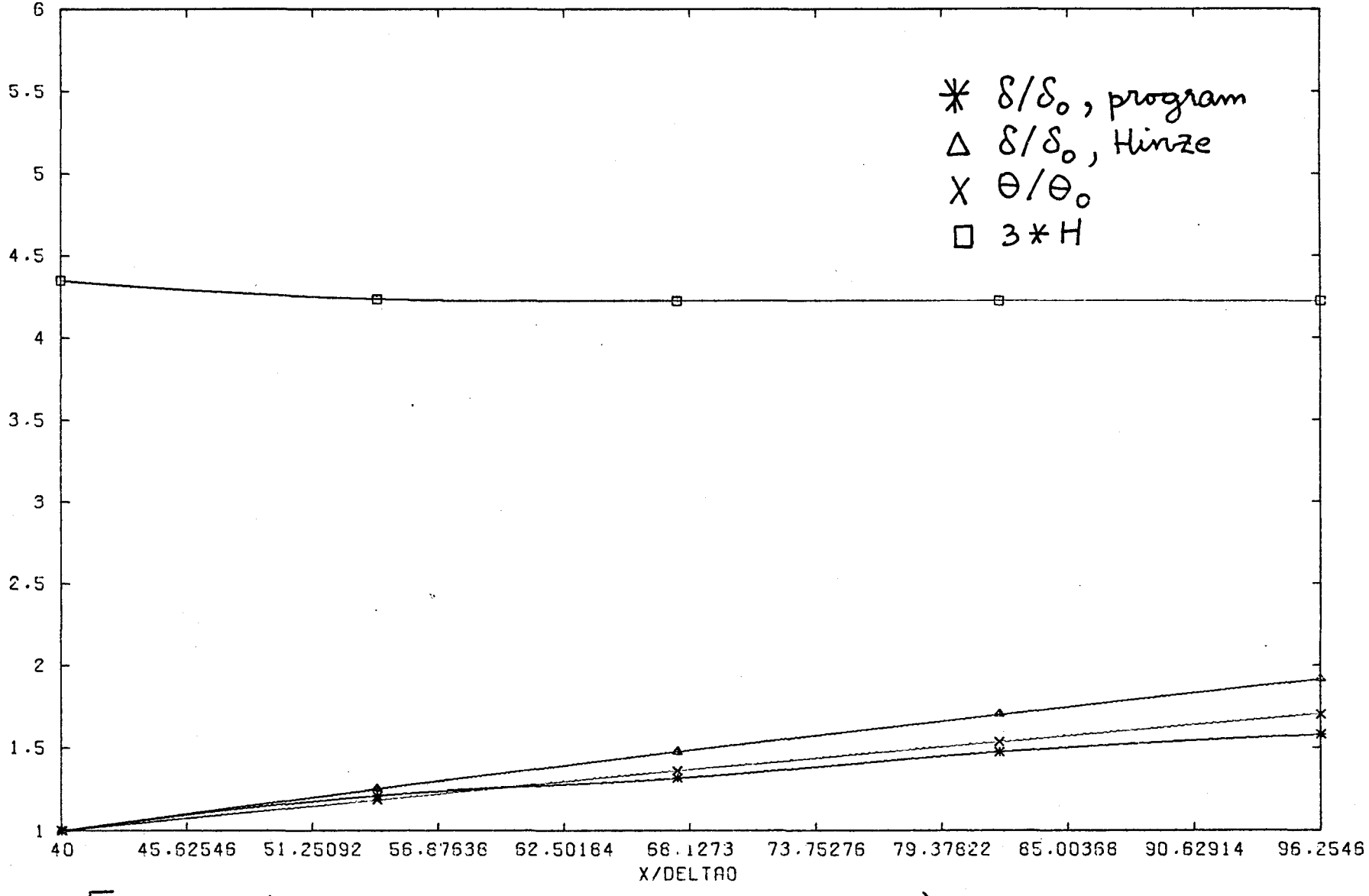


Fig. E.6 Boundary-layer growth (imax = 19)