THE UNIVERSITY OF ASTON IN BIRMINGHAM

Department of Mathematics

A NUMERICAL SOLUTION OF THE THREE-DIMENSIONAL TURBULENT BOUNDARY LAYER EQUATIONS

M.J.DRUMM

Supervisor:- Professor R.Hetherington

Thesis submitted for the degree of Ph.D.

January 1971.

SUMMARY.

The present investigation is concerned with the computation of three-dimensional turbulent boundary layers. A numerical method has been developed to solve the three-dimensional boundary layer equations using an iterative scheme based essentially on the Crank-Nicholson finite difference approximation. The scheme also employs a streamline-type transformation which enables the individual velocity profiles to be iterated for independently of each other so improving the efficiency of the calculation. The effective viscosity is computed from the mixing length concept and an empirical correlation for the outer layer. The logarithmic law of the wall is used as the effective wall condition. A listing of a computer program written in Fortran IV to calculate boundary layer development using this method is also included.

Extensive comparisons of the present theory with both experiment and alternative theories have been included. Two-dimensional flows have been calculated with reasonable success, predictions for which compare favourably with calculations based on Head's entrainment approach, and two severe cases were treated competently. In the first the pressure gradient was suddenly removed from an equilibrium layer, and in the second the flow was maintained in a near-separating condition. The pseudo-three-dimensional flows considered show that crossflow angles can be treated quite successfully while in three-dimensional comparisons, even though the crossflow is predicted well, the crossflow angle tends to be significantly underestimated. The two three-dimensional turbulent boundary layers calculated provide good overall agreement with experiment.

The present work provides a firm basis on which to further investigate the three-dimensional turbulent boundary layer and the enclosed program will provide a useful tool for predicting such flows. It is felt however that the effective viscosity model used in the outer layer should be more broadly based by considering more experimental configurations for the purpose of the empirical correlation. A great benefit will be obtained overall by considering this problem even on a two-dimensional basis. Nevertheless the present scheme is capable of coping adequately with varying types of boundary layer development in both two and three dimensions.

C O N T E N T S.

Chapter Four

A FINITE DIFFERENCE METHOD OF SOLUTION OF THE BOUNDARY LAYER EQUATIONS

Chapter Five

TWO-DIMENSIONAL CALCULATIONS

Chapter Six

THREE-DIMENSIONAL CALCULATIONS

INTRODUCTION.

Since the beginning of this century an over increasing amount of attention has been given to the investigation of boundary layers with a view to being able to completely understand the behaviour of the boundary layer and its effects as they occur in various branches of engineering and technology, notably in the aeronautical and compressor and turbine fields. It must be expected that the purpose of the current intensive research is to provide the designer with a tool for determining performance characteristics without having to resort to the laborious process of rig simulation, and from this point of view it must be admitted that the capabilities of the science at the present time are poor.

Methods currently available for calculating turbulent two-dimensional boundary layers are numerous and generally (with a few exceptions) of poor performance and restricted application since they are greatly dependent on empirical information extracted from a small number of experiments. There, moreover, is an increasing awareness at the present time that methods of calculation generated for the two-dimensional boundary layer are of limited practical application unless they can be modified to take account of three-dimensional effects. The purpose of the present investigation is in fact to generate a method for calculating three-dimensional turbulent boundary layers.

There is a tendency when referring to a boundary layer as being three-dimensional to mean simply that a crossflow component of velocity exists (i.e. there is flow within the boundary layer normal to the flow at the outer edge) so that the problem might in the mathematical sense still be two-dimensional. The expression 'three-dimensional' will here be reserved for flows three-dimensional in the mathematical sense i.e. in the more restricted sense, and

boundary layers whose parameters are dependent on two space variables only but also contain crossflows will be referred to as 'pseudo-three-dimensional'.

A method is presented for calculating laminar or turbulent boundary layers over two- (with or without crossflows) or three-dimensional solution spaces. The solution scheme to be discussed - restricted to the problem of steady, incompressible flow over a smooth, flat or developable, impermeable surface - is based on the boundary layer approximation to the time-averaged turbulent Navier-Stokes equations, complemented by an effective viscosity function which makes use of the mixing length concept. The scheme to be discussed uses the logarithmic law of the wall, which is well substantiated for the two-dimensional turbulent boundary layer, as the boundary condition at the wall and a frequently postulated extension of this law to three dimensions. In this manner skin friction at the wall is provided implicitly. The only other assumption required to extend the two-dimensional calculation to three dimensions is the assumption that in the turbulent boundary layer the shear stress vector is parallel to the maximum rate of strain vector of the mean flow. It must be stressed however that the computer program which has been written is structured in such a way as to make it reasonably simple to test effective viscosity hypotheses different to those used in the present calculations. The numerical method is essentially an iterative scheme based on the Crank-Nicholson finite difference approximation. The general approach to the problem is similar to that employed by Spalding in two-dimensions although the mathematical techniques used here have necessarily been chosen so as to facilitate the three-dimensional calculation.

Toward the end of this investigation it came to the attention of the author that Nash had attempted a similar approach to the same

problem but had based his calculations on the turbulent kinetic energy equation as initiated by Bradshaw in two-dimensions. The only experiment for which predictions of Nash and the present method have been compared show that both methods have more or less the same capabilities for calculating crossflows.

Owing to the introduction of a transformation closely resembling streamline coordinates the time taken to generate solutions using the present method on the IBM S360/65 computer was found to be reasonably efficient.

One problem that was thought would present some difficulty in the present calculation method was that of prescribing the side boundary conditions to the problem, although if sufficient care is taken it was found that this difficulty can quite easily be overcome.

We now proceed to give a short account of the contents of the chapters which constitute this present work. Chapter One gives a brief description of the concept of the boundary layer and states the equations governing the motion of the three-dimensional turbulent boundary layer while in Chapter Three are collected a number of experimental and theoretical results which will either be useful in the derivation of the scheme for the solution of the boundary layer equations considered in Chapter Four or be used as a check on the results obtained. Chapter Two gives a brief summary of the methods presently in use for calculating two-dimensional boundary layers and the attempts that have been made to include crossflow effects or to calculate three-dimensional boundary layers. Particular attention has been paid to the amount of empirical information necessary for each of these methods in two dimensions and the feasibility of obtaining the additional empiricism needed to extend the individual methods scope of application beyond the two-

dimensional case. Also included in Chapter Two is a more detailed account of the reasons for the choice of the approach to the problem used in the present investigation.

A description of the proposed solution scheme is to be found in Chapter Four and a listing of the computer program written in Fortran IV is supplied as an appendix. In Chapter Five this program has been used to simulate a number of two-dimensional experiments with reasonable success. The empirical constants inherent in the assumptions underlying the calculations were adjusted to ensure agreement with a two-dimensional experiment. The two constants required for the law of the wall were found to be adequately represented by their accepted experimental values i.e. in the usual notation

$$\kappa = 0.41 \qquad A = 4.9$$

and as might be expected it was found necessary to take the same value for κ in Prandtl's mixing length hypothesis as that used in the logarithmic law of the wall. The only other empirical information - that involved in determining the position of the outer region of the boundary layer i.e. the region in which the mixing length ceases to provide an adequate explanation of the flow - was obtained empirically to provide agreement with one retarding two-dimensional experiment.

Having determined the empirical constants and hypotheses to be used and validifying the choice by simulating a number of other two-dimensional experiments (Chapter Five) a number of pseudo-three-dimensional and three-dimensional boundary layers were simulated in Chapter Six with some success. The triangular model for the polar plot was largely confirmed by these calculations, which also gave a very convincing account of the process of crossflow reversal. The finite difference scheme was verified by means of a simulation of a laminar boundary layer for which an analytic solution existed.

Throughout what follows references (p.106) will be referred to by numbers in square brackets and ℓn will refer to the natural logarithm.

C H A P T E R O N E

THE TURBULENT BOUNDARY LAYER

1.0) Introduction.

The concept of the boundary layer was introduced by Prandtl at the beginning of this century to explain why in the flow of a real fluid over a streamlined body the streamline pattern at high Reynolds numbers very closely resembles that of a perfect (i.e. frictionless) fluid. In such flows, Prandtl suggested, the effects of the viscosity of the fluid are confined to a narrow region enveloping the body and the fluid in the remaining region is for all practical purposes inviscid.

In non-viscous fluids contacting layers of the fluid moving relative to each other experience no tangential forces (i.e. frictional or shearing forces) but only normal (pressure) forces and fluid layers close to an immersed body move over the body without experiencing any retarding effects. In real fluids however frictional forces come into play and effect shearing stresses between adjacent layers of fluid possessing relative motion, and in particular prevent layers of fluid adjacent to an immersed body from sliding over the body i.e. a 'no-slip' condition is introduced. This means that even in a fluid whose internal relative motion is not of such a magnitude as to produce frictional forces, the containing vessel or any immersed body moving relative to the fluid may produce significant frictional forces throughout the region termed its boundary layer. This layer is that region over which the velocity of the fluid varies between the zero velocity of the fluid relative to the wetted surface and the velocity in the body of the fluid at a point where the flow can be considered frictionless. Characteristically this region is very narrow so that velocity gradients through the layer and particularly close to the wetted surface are very large hence

1.0) contd.

giving rise to the shearing stresses previously described.

Also at high Reynolds numbers the phenomenon known as turbulence i.e. the amplification of small oscillations within the flow, plays a significant part in determining the behaviour of the flow close to the immersed body and is effectively the cause of a further increase in the shearing stress within the boundary layer.

It is the purpose of the present chapter to consider the equations of motion governing the flow within the boundary layer and in these to take account of the effects of turbulence.

1.1) The three-dimensional turbulent boundary layer equations.

The equations governing the flow of a ~~inviscid~~ viscous fluid, the equations which in fact form the basis of the whole science of fluid mechanics, are the Navier-Stokes equations which can be written for the steady flow of an incompressible fluid with zero body forces as

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) \qquad (1.1.1)$$

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) \qquad (1.1.2)$$

$$u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial z} + \nu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) \qquad (1.1.3)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \qquad (1.1.4)$$

where u,v,w are the localised velocity components associated with the rectangular co-ordinate directions x,y,z respectively, p is the pressure and ρ,ν are the fluid properties density and kinematic

1.1) contd.

viscosity. These four equations contain four unknowns u,v,w,p and it is thus possible, at least in theory, to solve for the four unknowns from equations (1.1.1-4) once the proper boundary conditions have been prescribed. In practice these equations have been solved in their entirity only in a selected number of simple cases. In particular the equations governing the flow of a perfect fluid are as above but with the second order terms deleted. This simplification is significant in that although the solution of the equations is considerably simplified at the same time the conditions needed to be specified at boundaries in a real fluid cannot all be satisfied.

The most promising approach to the solution of equations (1.1.1-4) in many cases of practical interest is to use the boundary layer concept to divide the solution space into two regions. In the first region, the main body of the fluid, viscous forces are to be ignored and the flow is to be treated as inviscid. The second region is the boundary layer in which although viscous forces need be considered it is possible to obtain simplified equations of motion which hold throughout this region. Before doing this however consideration must be given to the problem of turbulence.

In laminar fluid flow, characterised by low Reynolds numbers, the fluid particles during the course of their motion move along regular smooth paths. As the Reynolds number is increased however the regular motion of the fluid particles breaks down and superimposed on the overall tendency of the flow are random fluctuations of the individual particles giving rise to turbulent motion and high vorticity. The process of change from laminar motion to turbulent motion is termed transition and is of particular relevance to boundary layers.

1.1) contd.

The usual approach to the problem of turbulence is to substitute for the velocity components u,v,w, which are varying randomly with time about some mean value at any particular point, composite velocities of the form

$$u = \bar{u} + u' \tag{1.1.5}$$

where $\bar{u}$ is the mean value at a point of the component of motion in the direction of the co-ordinate axis x i.e.

$$\bar{u} = \frac{1}{T} \int_{t_o}^{t_o+T} u \, dt$$

where the integration is taken over a sufficiently long period of time T to ensure that $\bar{u}$ is independent of time. The component of velocity u' is thus the fluctuation about this mean such that

$$\overline{u'} = 0$$

Defining expressions of the form typified by equation (1.1.5) for u,v,w,p, substituting these into the Navier-Stokes equations and averaging the equations over a time interval T leads after manipulation to the equations

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} = - \frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right)$$

$$- \left(\frac{\partial}{\partial x} \overline{u'^2} + \frac{\partial}{\partial y} \overline{u'v'} + \frac{\partial}{\partial z} \overline{u'w'} \right) \tag{1.1.6}$$

$$u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} = - \frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2} \right)$$

$$- \left(\frac{\partial}{\partial x} \overline{u'v'} + \frac{\partial}{\partial y} \overline{v'^2} + \frac{\partial}{\partial z} \overline{v'w'} \right) \tag{1.1.7}$$

1.1) contd.

$$u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial z} + \nu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right)$$

$$-\left(\frac{\partial}{\partial x}\overline{u'w'} + \frac{\partial}{\partial y}\overline{v'w'} + \frac{\partial}{\partial z}\overline{w'^2}\right) \qquad (1.1.8)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \qquad (1.1.9)$$

where the bars have been dropped from the time-averaged velocity components and pressure for convenience since equations (1.1.6-9) hold for laminar flow if the fluctuating components are taken as zero. The additional terms on the right hand sides of equations (1.1.6-8) compared with the original Navier-Stokes equations can be seen to play a role similar to the viscous terms on the right hand sides of equations (1.1.1-3). For this reason the main effect of the turbulence on the mean motion is to imbue the motion with additional stresses which are called apparent or Reynolds stresses. It should be noted that the equations for the time-averaged quantities u,v,w,p cannot now be solved unless additional information is provided for the evaluation of the terms which contain time-averaged fluctuating quantities.

As has already been stated it is possible to introduce into the full Navier-Stokes equations some simplifying assumptions relevant to the flow in the boundary layer. This boundary layer approximation holds when the boundary layer thickness is small in comparison with a 'characteristic length', and is tantamount to assuming that the rate at which quantities change as the boundary layer is traversed is much greater than the rate of change of quantities in directions parallel to the plane of the wall on which the boundary layer is developing. Thus, if z is the perpendicular

1.1) contd.

distance measured from the wall into the body of the fluid and x,y are co-ordinate axes in the plane of the wall, second order derivatives with respect to x and y can be disregarded in preference to other terms in the Navier-Stokes equations. Although the reasoning given here lacks the more rigorous treatment often associated with the boundary layer approximations the consequence is the same and because a detailed derivation of the boundary layer equations for two-dimensional flow is provided by H.Schlichting [1] and the extensions to three-dimensional and turbulent flows are very similar and throw little light onto the mechanism of the flow no more will be said here beyond quoting the boundary layer equations for three-dimensional turbulent motion:

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \frac{\partial}{\partial z}\left(\nu_{ex}\frac{\partial u}{\partial z}\right) \qquad (1.1.10)$$

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial y} + \frac{\partial}{\partial z}\left(\nu_{ey}\frac{\partial v}{\partial z}\right) \qquad (1.1.11)$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \qquad (1.1.12)$$

The effective kinematic viscosities, defined such that

$$\nu_{ex}\frac{\partial u}{\partial z} = \nu\frac{\partial u}{\partial z} - \overline{u'w'} \qquad (1.1.13)$$

$$\nu_{ey}\frac{\partial v}{\partial z} = \nu\frac{\partial v}{\partial z} - \overline{v'w'} \qquad (1.1.14)$$

were originally introduced by Bousinesq who drew the analogy between the effective kinematic viscosities and the coefficient of viscosity ν in Stokes' law. The laminar shear stress terms in equations (1.1.13-14) are significant only in a narrow region very close to the wall and are usually ignored. The momentum equation associated with the z direction reduces as a consequence

1.1) contd.

of the boundary layer approximations to

$$\frac{\partial p}{\partial z} = 0$$

Thus, p is a function of x and y only and can be determined from the freestream pressure distribution i.e. the velocity distribution at the outer edge of the boundary layer must be known in order that the pressure distribution can be derived from Bernoulli's equation

$$p + \tfrac{1}{2}\rho(U^2 + V^2) = \text{constant} \qquad (1.1.15)$$

where U,V are the freestream velocity components associated with the x,y directions respectively. It should be noted that the consequence of equation (1.1.15) in the boundary layer equations is to impose a condition of irrotationality on the freestream velocity distribution.

The three equations (1.1.10-12) involve the unknowns u,v,w and it is thus possible to solve these equations for u,v,w, once the correct boundary conditions have been prescribed, if ν_{ex}, ν_{ey} can be correlated with the mean velocity field. An alternative approach is to make use of the turbulent energy equation

$$u\frac{\partial \bar{t}}{\partial x} + v\frac{\partial \bar{t}}{\partial y} + w\frac{\partial \bar{t}}{\partial z} + \overline{u'w'}\frac{\partial u}{\partial z} + \overline{v'w'}\frac{\partial v}{\partial z}$$

$$+ \frac{\partial}{\partial z}\left(\frac{1}{\rho}\overline{w'p'} + \overline{w't}\right) + \varepsilon = 0 \qquad (1.1.16)$$

(obtained by manipulating the Navier-Stokes equations, time-averaging and making use of the notation introduced in equation (1.1.5)) to provide an equation from which the turbulent shear stress can be determined if the turbulent kinetic energy

$$t = \tfrac{1}{2}(u'^2 + v'^2 + w'^2)$$

1.1) contd.

ε (dissipation of turbulent energy by viscous forces) and the time averaged quantity

$$\frac{1}{\rho}\ \overline{w'p'} + \overline{w't}$$

can be provided by some empirical source.

1.2) The three-dimensional momentum integral equations.

A simplification to the equations governing the behaviour of the boundary layer is introduced by integrating the boundary layer equations (1.1.10-12) through the thickness of the boundary layer to provide the momentum integral equations. When such an approach is employed it has become customary to write the boundary layer equations in 'streamline co-ordinates' i.e. to replace the x,y co-ordinates by the co-ordinate system formed by the projection perpendicular to the wall of the external streamlines and their orthogonal trajectories. Velocity components within the boundary layer parallel to the wall are resolved similarly. We consider a co-ordinate system in which ξ is the metric measured along a streamline η = constant and denote the velocity components within the boundary layer in the direction of ξ, η increasing by u_1, u_2 respectively so that at the edge of the boundary layer

$$u_1 = U_1 \quad , \quad u_2 = 0.$$

Integrating the boundary layer momentum equations as described and using the continuity equation to eliminate w the following equations are obtained ~~by~~:

$$U_1 \frac{\partial \theta_{11}}{\partial \xi} + \frac{1}{h_2} \frac{\partial \theta_{12}}{\partial \eta} + \frac{\partial U_1}{\partial \xi} (2\theta_{11} + \delta_1^*)$$

$$+ \frac{U_1}{h_2} \frac{\partial h_2}{\partial \xi} (\theta_{11} - \theta_{22}) = \frac{c_{f_1}}{2} \qquad (1.2.1)$$

1.2) contd.

$$U_1 \frac{\partial \theta_{21}}{\partial \xi} + \frac{1}{h_2} \frac{\partial \theta_{22}}{\partial \eta} + 2 \frac{\partial U_1}{\partial \xi} \theta_{21} + \frac{1}{h_2 U_1} \frac{\partial U_1}{\partial \eta} (\theta_{11}+\theta_{22}+\delta_1^*) + 2 \frac{U_1}{h_2} \frac{\partial h_2}{\partial \xi} \theta_{21} = \frac{c_{f_2}}{2} \qquad (1.2.2)$$

The momentum thicknesses are defined:

$$\theta_{11} = \int_0^\infty \left(1 - \frac{u_1}{U_1}\right) \frac{u_1}{U_1} \, dz, \quad \theta_{21} = -\int_0^\infty \frac{u_1 u_2}{U_1^2} \, dz$$

$$\theta_{12} = \int_0^\infty \left(1 - \frac{u_1}{U_1}\right) \frac{u_2}{U_1} \, dz \, , \quad \theta_{22} = -\int_0^\infty \frac{u_2^2}{U_1^2} \, dz \qquad (1.2.3)$$

the displacement thicknesses:

$$\delta_1^* = \int_0^\infty \left(1 - \frac{u_1}{U_1}\right) dz \, , \quad \delta_2^* = -\int_0^\infty \frac{u_2}{U_1} \, dz \qquad (1.2.4)$$

and the coefficients of friction:

$$c_{f_1} = \frac{\tau_{01}}{\frac{1}{2}\rho U_1^2} \, , \quad c_{f_2} = \frac{\tau_{02}}{\frac{1}{2}\rho U_1^2} \qquad (1.2.5)$$

where τ_{01}, τ_{02} are the components of the turbulent shear stress at the wall in the ξ, η directions i.e.

$$\frac{\tau_{01}}{\rho} = \nu \frac{\partial u_1}{\partial z} - \overline{u_1' w'} \qquad (1.2.6)$$

$$\frac{\tau_{02}}{\rho} = \nu \frac{\partial u_2}{\partial z} - \overline{u_2' w'} \qquad (1.2.7)$$

the left hand sides being evaluated at $z = 0$. The metric factor h_2 is that associated with η and is assumed to be a function of ξ, η such that

1.2) contd.

$$ds^2 = \frac{1}{U_1^2}\, d\xi^2 + h_2^2\, d\eta^2 + dz^2.$$

In a similar way it is possible to obtain energy integral equations but these are rarely used as a means of calculation and will not be discussed here. Equations (1.2.1-2) will need obviously be supplemented by other relationships since these two equations contain seven independent unknowns (δ_2^* was eliminated using $\delta_2^* = \theta_{21} - \theta_{12}$). In the two-dimensional problem the one momentum integral equation contains the three unknowns

$$\theta_{11}\ ,\quad \delta_1^*\ ,\quad c_{f_1}.$$

It should be pointed out that all the equations summarised in this chapter are generally accepted as being applicable to boundary layers developing over boundaries of ~~large~~ small curvature (in comparison with the boundary layer thickness) and not only on flat surfaces.

C H A P T E R T W O.

METHODS OF COMPUTING BOUNDARY LAYERS.

2.0) Introduction.

The numerous and varied methods that are currently available for the calculation of the two-dimensional turbulent boundary layer testify both to the large amount of attention that has been given to the problem over the last ten to twenty years and also to the lack of reliance the individual contributors placed on contemporary methods of solution.

These calculation methods can be broadly divided into two groups the first of which, by far the largest and most profuse, are those termed integral methods in which the boundary layer equations are abandoned in favour of the momentum integral equations so reducing the problem space by one dimension. In the two-dimensional boundary layer, integral methods are generally based on the assumption that the shape and scale of the velocity profile are adequately represented by two parameters. As a means of calculating these parameters the integral equation (which contains three unknowns any two of which on the basis of this assumption specify the third) is solved in conjunction with some ancillary relation, which will have to be determined from empirical correlations in combination possibly with some hypothesis. It is the different approaches employed in fulfilling this last requirement that have given rise to the diversity of current calculation methods of this type. The second group of calculation methods includes those methods which are based on the boundary layer equations with the necessary additional assumptions made concerning the fluctuating components of velocity.

Although at first sight the prospect of generating empirical relations to explain the small scale behaviour of the turbulent terms of the boundary layer equations, the process of turbulence not being understood, seems quite formidable when

2.0) contd.

compared with that of correlating gross boundary layer parameters whose significance is more easily appreciated, this in fact transpires not to be so. The prominence of integral methods is more easily understood when one realises that they were calculation methods developed with the intention of being applied to the slide rule and desk machine. These same methods of computation would obviously make the solution of the boundary layer equations too lengthy a task. Today, with the development of the high speed computer, one would expect to see some movement away from this one-sided situation but regretfully this is not so to any marked extent. The present author is of the view that since methods based on the boundary layer equations are more readily adapted to three-dimensions, once a satisfactory two-dimensional method has been developed, more time could be profitably spent in improving the methods of calculation based on these equations.

2.1) Integral methods of calculation.

Restricting the present discussion to two-dimensional boundary layers (i.e. where the problem is dependent only on two space variables and the velocity component v associated with the third direction is identically zero) we will adopt the notation generally used in this context:

$$\theta = \theta_{11} \ , \ \delta^* = \delta_1^* \ , \ U = U_1 \ , \ c_f = c_{f_1}$$

and introduce the shape factor H, and the Reynolds number R_θ based on the momentum thickness

2.1) contd.

$$H = \frac{\delta^*}{\theta} \quad , \quad R_\theta = \frac{\theta U}{\nu} \tag{2.1.1}$$

The momentum integral equation can now be written

$$\frac{dR_\theta}{dx} = \frac{c_f}{2} \frac{U}{\nu} - (H+1) \frac{R_\theta}{U} \frac{dU}{dx} \tag{2.1.2}$$

Equation (2.1.2) is solved for R_θ, the shape factor H and coefficient of friction c_f being provided respectively by an empirical auxiliary equation which is usually of the form

$$\theta \frac{dH}{dx} = M - L \frac{\theta}{U} \frac{dU}{dx} \tag{2.1.3}$$

where L, M are in the most general case functions of H, R_θ, and a skin friction equation which can reliably be assumed to be of the form

$$c_f = f(R_\theta, H) \tag{2.1.4}$$

It is also generally considered that specifying the parameters R_θ, H is sufficient to define u/U as a function of z/θ.

Thompson [2] has given a thorough assessment of the dependability of the various two-dimensional auxiliary equations, as distinguished by different L,M in equation (2.1.3), available for calculating two-dimensional incompressible turbulent boundary layers and compared theoretical predictions with a wide range of published experimental results. A point of fundamental importance revealed by Thompson (by the discrepancy between measured θ development and that as predicted by the momentum integral equation evaluated using experimental shape factor distributions) is the presence of what are almost certainly significant three-dimensional effects in the majority of what were intended to be two-dimensional boundary layers. The effectiveness of the various auxiliary

2.1) contd.

equations was compared using the measured R_θ distributions and Thompson concluded as a result of his calculations that

> 'with the exception of the entrainment equation of Head, no shape factor equation provides satisfactory agreement with more than one half of the measured developments that have been used'

He also points out that established methods have been generally accepted on the basis of only a few comparisons with experiment that have produced comparatively good results. Thompson thus concludes that, with the exception of Head's entrainment method [3] which gave reasonable agreement with experiment, all methods for calculating two-dimensional turbulent boundary layers are generally indifferent to very poor. Two additional points made by Thompson cannot be stressed too often. The first is that two-dimensional experimental results must be accompanied by some indications as to the effects of convergence or divergence of the flow and, secondly, computation methods must be compared with a wide range of experiments before their validity can be established. Thompson, rather surprisingly in view of the poor performance he attributes to integral methods in general, appears to dismiss calculation methods based on the boundary layer equations with the remark:

> 'In the case of turbulent flows, no universal expression is known relating the Reynolds' stresses to the mean velocity distribution, and no exact solution of the boundary layer equations are possible.'

Other integral methods have been developed which make use of the so-called energy integral equation and the moment of momentum integral equation which determine the growths of the energy thickness

2.1) contd.

$$\delta^{**} = \int_0^\infty \left(1 - \left(\frac{u}{U}\right)^2\right) \frac{u}{U} \, dz \qquad (2.1.5)$$

and the moment of momentum thickness

$$\theta_z = \int_0^\infty z\left(1 - \frac{u}{U}\right) \frac{u}{U} \, dz \qquad (2.1.6)$$

respectively in a manner similar to the way equation (2.1.2) determined the growth of the momentum thickness θ. The shape factor equation is carried over similarly to provide an equation for the developments of the shape factors based on δ^{**}, θ_z respectively. Comments concerning the performance of these methods will be deferred until the next section.

It has only been in recent years that integral methods have with much success been applied to three-dimensional boundary layer calculations - all such attempts have though to the knowledge of the present author been restricted to pseudo-three-dimensional boundary layers and no attempt has yet been made to solve the three-dimensional integral equations over a two-dimensional (x,y) space. A review of the state of the knowledge (1963) of three-dimensional turbulent boundary layers, particularly with reference to calculation methods, has been made by Cooke [4] who noted that all contemporary calculation methods assumed small or zero cross flows and used established two-dimensional velocity profiles and skin friction equations for the three-dimensional streamwise counterparts. Cooke also reviewed the various proposals put forward for the representation of the crossflow velocity profiles and more recently Cumpsty has made comparisons of experimental three-dimensional velocity

2.1) contd.

profiles with those obtained from the various proposed prediction methods. Cumpsty [5] has shown that the streamwise velocity profiles and skin fricton can quite adequately be approximated to by the two-dimensional velocity profile families and skin friction laws. The triangular representation of the crossflow profiles, generally ascribed to Johnston, is also considered by Cumpsty [6] to be applicable in a wide range of situations and to be easily extended to cases where crossover profiles exist. Some means of effecting the calculation of the parameters on which Johnston's triangle depends are still yet to be formulated however. The crossflow profile proposed by Mager

$$\frac{u_2}{U_1} = \alpha \left(1 - \frac{z}{\delta}\right)^2 \frac{u_1}{U_1} \qquad (2.1.7)$$

where δ is the boundary layer thickness and α is a parameter representing the extent of the crossflow, Cumpsty considered to be applicable only in the case of modest crossflows and in a selected number of other situations.

Calculations of pseudo-three-dimensional boundary layers have been made by P.D.Smith [7] who considered the flow over an infinite swept wing and compared his calculations with some of his own experimental data. Smith tested in all six different variants of integral methods and found in all his calculations considerable discrepancies between theory and experiment which Smith attributed to either the inapplicability of the two-dimensional skin friction law to three-dimensional flow or to the neglect of certain terms in the derivation of the streamwise momentum integral equation. Smith considered the former to be the more likely cause. Mager's crossflow

2.1) contd.

representation and a power law approximation to the streamwise profiles were used throughout Smith's calculations.

The same problem was considered both theoretically and experimentally by Cumpsty and Head [8] who extended the entrainment method of Head to account for crossflow (a possibility also investigated by P.D.Smith). A family of two-dimensional velocity profiles constructed by Thompson was used in conjunction with equation (2.1.7) to represent the velocity distributions. Predictions of θ, H and crossflow profiles were found to be considerably underestimated the situation being improved somewhat by a small adjustment to the spanwise velocity which produced 'tolerable agreement' with experiment. There seems to be some doubt as to the feasibility of attempting the experimental simulation of the infinite swept wing, a point which Cumpsty and Head alude to but Smith dismisses. The type of flow studied by Cham and Head [9] would seem to be more reliably two-dimensional (in the mathematical sense), the experiment being concerned with a rotating circular disc. In this case the velocity representation of Thompson was said to be of considerable accuracy and that of Mager reasonable although to produce overall agreement of the theory (similar to that of Cumpsty and Head) with experiment a 30% reduction in entrainment as compared with the two-dimensional theory was required.

2.2) Methods based on the boundary layer equations.

The obvious apprcach to the solution of the boundary layer equations is to assume that the local turbulent shear stress can be empirically related to the mean velocity. Such an approach has been made by Spalding and Patankar [10] who

2.2) contd.

solved the heat-, mass- and momentum transfer equations for the two-dimensional turbulent boundary layer. The equations were written in terms of a non-dimensional stream function to account for boundary layer growth and the logarithmic law of the wall was employed as the effective wall condition. The calculations performed by Spalding and Patankar made use of Prandtl's mixing length although the point is made that any other hypothesis for ν_{ex} could conveniently be incorporated into their solution scheme. Spalding and Patankar, being primarily concerned with the problem of heat transfer, gloss over the capabilities of their method for computing turbulent boundary layers but the few predictions that are available appear to give plausible agreement with experiment.

Bradshaw, Ferris and Atwell [11] chose to base their calculation method on the turbulent energy equation on the assumption that the turbulent shear stress was likely to be more closely related to other properties of the turbulence than to the mean velocity field. Bradshaw in the solution to the two-dimensional problem defined the length parameter L

$$L\varepsilon = \tau^{3/2}$$

where $\tau = -\overline{u'w'}$ is the kinematic shear stress outside the laminar sublayer, and introduced functions

$$\frac{L}{\delta} = \frac{\tau^{3/2}}{\varepsilon\delta}$$

$$G = \frac{(\overline{p'w'} + \overline{tw'})}{\tau\ \tau_{max}^{1/2}}$$

$$a = \frac{\tau}{\overline{t}}$$

where L/δ and G were taken to be functions of z/δ and a was

2.2) contd.

taken as constant. These assumptions allowed the boundary layer equations (1.1.10,12,16) to be solved for u,w, τ. As with the method of Spalding the boundary conditions for u at the wall was the logarithmic law of the wall. The additional assumption of a linear shear stress relationship at the wall and in the freestream $\tau \doteq 0$ completed the boundary conditions for the problem.

The situation with respect to the dependability of the methods of Spalding and Bradshaw is very much as described by Thompson to be the case for integral methods; the published literature on both these methods shows only a few comparisons with experiment which have all the appearances of showing reasonable agreement.

Recently Nash [12] has extended the method of Bradshaw and calculated a three-dimensional boundary layer (a simulation of the experiment of Hounung and Joubert [13]) with promising results. The only additional assumption made, over those introduced by Bradshaw, was that the shear stress and the maximum rate of strain of the mean flow have a common line of action at any point i.e.

$$\overline{u'w'} : \overline{v'w'} :: \frac{\partial u}{\partial z} : \frac{\partial v}{\partial z}$$

A recent investigation [14] made to determine how prediction methods of all types would compare in calculating two-dimensional turbulent boundary layer developments came to the conclusion that 'most prediction methods do rather well'. Some attempt was made as part of this study to rank the different methods in order of performance by placing each method into one of three

2.2) contd.

groups. The first group comprised of two methods based on each of the energy integral equation and moment of momentum integral equation, one based on the turbulent energy equation (in association with the boundary layer equations) and two based on the boundary layer equations. Calculations based on the entrainment approach fell mostly within the second group.

2.3) Proposed solution scheme.

Having developed a method of calculating two-dimensional turbulent boundary layers the amount of effort necessary to extend the calculation method to the three-dimensional problem is primarily influenced by whether the original method is based on an integral equation or on the boundary layer equations. The additional assumptions required to extend an integral method are considerable, witness to this being provided by the additional information necessary for the extension of the two-dimensional method to the pseudo-three-dimensional problem, while it would appear to be a relatively simple matter to extend either of the two main methods of solving the boundary layer equations.

Having formulated the problem the relative merits of the methods are reversed when the prospect of solving the equations is considered - it is undoubtedly simpler to solve the momentum integral equations together with any ancillary relations over a two-dimensional space than it is to solve the complete boundary layer equations over a three-dimensional space. In deciding on the approach to the problem of calculating three-dimensional boundary layers therefore we must weigh the mathematical

2.3) contd.

considerations against the problem inherent in attempting to supply all the necessary empirical information for the definition of the problem. It would almost seem that the choice has been made for us when we see that while it has been found possible to solve the boundary layer equations over a three-dimensional space the momentum integral equations have never been solved in more than one dimension.

Since it would appear that it will not be possible for some time to satisfactorily correlate crossflow velocity profiles and skin friction values we will restrict our attention in the present work to the boundary layer equations. The work of Nash only became known to the present author towards the end of the present investigation so that it was fortuitous that it was decided to omit the turbulent energy equation and concentrate on the effective viscosity approach to the problem. This decision was partly made on the basis that the mixing length analogy of Prandtl has found application in such a wide variety of situations besides boundary layers that the reasons often given for its rejection seem not altogether acceptable. In addition it was felt that theturbulent energy equation was too dependent on empirical information.

CHAPTER THREE

PROPERTIES OF THE TURBULENT BOUNDARY LAYER.

3.0) Introduction.

Having in Chapters One and Two derived the equations of motion for the three-dimensional turbulent boundary layer and decided what approach to take in solving these equations, we now examine a number of physical properties of the turbulent boundary layer which will be required to facilitate the solution scheme to be presented in Chapter Four.

The nature of the turbulent boundary layer equations in their two-dimensional form (i.e. the equations independent of y, with v identically zero) presents two main difficulties in any proposed numerical method of solution. The first of these is the pertinent fact that the two equations available for the determination of u and w even when the correct boundary conditions have been prescribed are still not fully defined. The effective viscosity is as yet undetermined so that some empirical information is required to enable it to be calculated from the velocity field. The second problem concerns the difficulties inherent in trying to apply as the boundary condition at the wall the obvious fact that all velocity components must vanish there. Extending any solution method to facilitate the computation of three-dimensional boundary layers will obviously increase the difficulties originating from these two sources. In this chapter certain experimental and theoretical observations will be presented with the prime purpose of overcoming the difficulties associated with the solution of the two-dimensional turbulent boundary layer equations and to hypothesise, with the aid of the limited three-dimensional data available, relationships that will enable the two-dimensional method of solution to be extended to three-dimensions.

3.0) contd.

It is also anticipated that the boundary layer properties to be discussed here will provide a means of establishing the calculations to be presented in Chapters Five and Six.

The discussion contained in the remainder of this chapter will be concerned with two-dimensional turbulent boundary layers except where it is explicitly stated otherewise.

3.1) The effective viscosity concept.

The form of the boundary layer equations (1.1.10-12) makes use of the effective viscosity function as introduced by Bousinesq and although this device enables the equations to be expressed in a familiar form (the equations are now in line with the laminar equations except that the kinematic viscosity ν is replaced by a turbulence exchange coefficient) the problem of how to account ν_e $(= \nu_{ex})$ with the velocity field is still present.

The earliest attempt to allow for the effect of turbulence in the boundary layer equations was Prandtl's now well-known mixing length hypothesis which from physical considerations of the mechanism of turbulence deduced that

$$\nu_e = \ell^2 \left|\frac{\partial u}{\partial z}\right| \qquad (3.1.1)$$

where the so-called mixing length ℓ is still an unknown function but indications are that it is not influenced by the magnitude of the velocity and it is a purely local function. The concept of the mixing length has been proved to be very useful and, with simple postulations made concerning ℓ, has been applied to turbulent wall flows (including pipe and channel

3.1) contd.

flows in addition to the more usual boundary layer problem) and also to free turbulent flows (where fluid mixing takes place in the absence of a solid wall). With reference to boundary layers a number of arguments have been proposed for the determination of the mixing length ℓ and von Karman by means of a similarity hypothesis suggested

$$\ell = \kappa' \left| \frac{\partial u}{\partial z} \Big/ \frac{\partial^2 u}{\partial z^2} \right| \qquad (3.1.2)$$

where κ' is an empirical constant. The alternative presentation however

$$\ell = \kappa z \qquad (3.1.3)$$

where κ is another empirical constant, being simpler than equation (3.1.2) has been widely used in the calculation of turbulent boundary layers and has been credited with giving satisfactory results when applied to the region near the wall. Beyond this region the mixing length is generally assumed to tend to some constant value.

3.2) <u>The law of the wall.</u>

It has been appreciated for a long time that points taken near the wall from a mean velocity boundary layer profile can be rescaled into what is known as the law of the wall which states

$$\frac{u}{u_\tau} = f\left(\frac{z u_\tau}{\nu}\right) \qquad (3.2.1)$$

where u_τ is the so called friction velocity and f is a univeral function. If τ is the total stress (i.e. the sum of viscous and turbulent stresses) and τ_0 is the value τ attains at the wall

3.2) contd.

then the friction velocity is defined

$$u_\tau = \sqrt{\frac{\tau_0}{\rho}} \qquad (3.2.2)$$

The coefficient of friction c_f can now be written:

$$c_f = 2\left(\frac{u_\tau}{\bar{U}}\right)^2 \qquad (3.2.3)$$

Equation (3.2.1) readily follows from a dimensional argument applied in the region of the wall. Prior to the development of the mixing length analogy the law of the wall was sometimes taken to be a power law in the absence of any better representation. In the laminar sublayer adjacent to the wall, where viscous stresses can be assumed to suppress any turbulence effects, the law of the wall can be plausibly expounded as a linear relationship, viz:

$$\frac{u}{u_\tau} = \frac{zu_\tau}{\nu} \qquad (3.2.4)$$

A particularly relevant conclusion concerning the form of the function f can be obtained by assuming, as experiment has shown to be the case, that there is a fully turbulent region outside the laminar sublayer in which the local shearing stress τ is approximately constant and equal to that at the wall. Making this assumption in conjunction with Prandtl's mixing length analogy, equation (3.1.1) using either von Karman's or the simplified model (equations (3.1.2,3) respectively) for the mixing length results in the equation

3.2) contd.

$$\frac{\partial u}{\partial z} = \frac{u_\tau}{\kappa z} \qquad (3.2.5)$$

which integrates to give

$$u = \frac{u_\tau}{\kappa} \ell n\ z + c$$

where c is a constant of integration (a function of x). This last equation is usually rewritten to bring it into line with the law of the wall as

$$\frac{u}{u_\tau} = \frac{1}{\kappa} \ell n \frac{z u_\tau}{\nu} + A \qquad (3.2.6)$$

where A is a constant - presumably the same constant for all turbulent boundary layer mean velocity profiles, in which form it is known as the logarithmic law of the wall.

The logarithmic law of the wall has been well established experimentally. It was first formulated from observations of turbulent flow in pipes and was later extended to include the mean velocity in a turbulent boundary layer. Ludwieg and Tillman (1949) from experimental data concluded that f for flows in boundary layers was independent of pressure gradient and established the logarithmic law experimentally. The logarithmic law is now believed to be applicable generally independently of the prevailing boundary conditions - boundary layers, pipes, channels (although different constants are required) - and typical constants quoted for boundary layer flows are

$$\kappa = 0.41 \quad , \quad A = 4.9$$

where the law can generally be assumed to hold for

3.2) contd.

$$\frac{zu_\tau}{\nu} > 30$$

away from separation. The form of the velocity profile in the inner part of the boundary layer is plotted in figure (3.2.1).

A second derivation of the logarithmic law of the wall of especial interest is that due to Millikan. Defect laws of the form

$$\frac{U - u}{u_\tau} = F \qquad (3.2.7)$$

where F is a function of z/z' (where various length scales z' have been proposed), parameters such as u_τ/U and terms dependent on the pressure gradient, have frequently been proposed for the outer part of the mean velocity profile. Millikan (1938) assumed that F was dependent on the scaled distance z/δ only

$$\frac{U - u}{u_\tau} = F\left(\frac{z}{\delta}\right) \qquad (3.2.8)$$

(the same argument holds for a more generalised form however) and that this defect law extended far enough into the boundary layer for there to be a region, generally referred to as the overlap region, in which the velocity profile is equally well represented by the law of the wall and the defect law equation (3.2.8). Obtaining from each of these equations an expression

3.2) contd.

$\frac{z}{u_\tau}\ \frac{\partial u}{\partial z}$ and equating them the equation

$$\frac{zu_\tau}{\nu}\ f'\left(\frac{zu_\tau}{\nu}\right) = -\ \frac{z}{\delta}\ F'\left(\frac{z}{\delta}\right) \qquad (3.2.9)$$

results where the dash denotes differentiation with respect to the argument of the function. The two sides of equation (3.2.9) can be independently functions of $zu_\tau/\nu, z/\delta$ respectively only if both sides are equal to a constant and if this constant is taken to be $1/\kappa$ the logarithmic law of the wall immediately follows.

A detailed discussion of the law of the wall has been given by Coles [15] who also analysed a wide range of experimental data to give a very convincing argument as to the validity of the logarithmic law of the wall in the turbulent boundary layer.

3.3) <u>The effective viscosity function of Mellor and Gibson.</u>

Mellor and Gibson [16] in response to the work of Clauser [17], who investigated the effect of pressure gardient on equilibrium turbulent boundary layers i.e. boundary layers in which the velocity defect equation (3.2.7) assumes the simple form of equation (3.2.8), generated an effective viscosity function to span the boundary layer outside the laminar sublayer. Clauser [18] had shown that it was possible to analyse the outer region (80%) of an equilibrium boundary layer by assuming the effective viscosity to be of the form

3.3) contd.

$$\nu_e = K\ U\ \delta^* \qquad (3.3.1)$$

where K, an absolute constant, was taken to be 0.016. The equilibrium flow profiles investigated by Clauser were those for which the parameter

$$\beta' = \frac{\delta^*}{\tau_0}\ \frac{dp}{dx} \qquad (3.3.2)$$

was held constant so that the defect law can be written

$$\frac{U - u}{u_\tau} = F\left(\frac{z}{\delta}\ ,\ \beta'\right) \qquad (3.3.3)$$

Clauser generated two equilibrium flows experimentally - those characterised by $\beta' = 1.8$ and 8.0 respectively. Mellor and Gibson concluded as a result of their analysis that the effective viscosity model defined by equation (3.3.1) in the outer region and the simpler mixing length model (equations (3.1.1,3))

$$\nu_e = \kappa^2 z^2 \left|\frac{\partial u}{\partial z}\right| \qquad (3.3.4)$$

in the overlap region suffice to predict defect profiles in equilibrium turbulent flows in the range

$$-0.5 < \beta' < \infty$$

with 'considerable precision'. For $\beta' < -0.5$ no solution was found to exist to satisfy the boundary conditions and the flow was considered separated ($\beta' > 0$ were decelerating flows, $-0.5 < \beta' < 0$ accelerating).

3.3) contd.

In a second paper Mellor [19] extended the effective viscosity model formerly proposed to include the laminar sub-layer where ν_e must tend to ν as z tends to zero. The restriction to equilibrium profile flows was also removed and the choice for the effective viscosity function parameter is reinforced by Mellor by a dimensional argument. The effective viscosity function expresses ν_e/ν as a function of ζ where

$$\zeta = \frac{\kappa^2 z^2}{\nu} \left|\frac{\partial u}{\partial z}\right|$$

as follows

$$\left.\begin{aligned} \frac{\nu_e}{\nu} &= \phi(\zeta) && \zeta < 11 \\ \frac{\nu_e}{\nu} &= \zeta && 11 < \zeta < \frac{KU\delta^*}{\nu} \\ \frac{\nu_e}{\nu} &= \frac{KU\delta^*}{\nu} && \frac{KU\delta^*}{\nu} < \zeta \end{aligned}\right\} \qquad (3.3.5)$$

where $\phi(\zeta)$ is a prescribed function. Figure (3.3.1) shows the composite effective viscosity model as proposed by Mellow. Since ζ increases and then decreases to zero again as the boundary layer is traversed from the wall the above formulation (equation (3.3.5)) for ν_e is not quite correct as it is intended that the third expression should hold exclusively in the outer part of the boundary layer.

3.3) contd.

It has already been mentioned that an alternative approach in the outer part of the boundary layer might be to assume that the mixing length ℓ tends to some constant value. An examination of experimental data presented by Maise and McDonald [20] would seem to support the latter of these alternatives i.e. that the mixing length rather than the effective viscosity should be taken as constant in this region.

We might also note here that because of the lack of turbulence measurements in three-dimensional boundary layer flows we are in a position to do no more than make the obvious extensions to three dimensions of the viscosity models discussed above. That is we will assume that the shear stress at any point acts in the same direction as the maximum rate of strain i.e.

$$\nu_{ex} = \nu_{ey} = \nu_e$$

and that the above formulations hold in three dimensions so long as u is replaced by the resultant velocity q parallel to the wall

$$q = \sqrt{u^2+v^2}$$

3.4) <u>The work of Coles.</u>

Coles [15] has suggested that it is possible to represent the mean velocity profiles of two-dimensional incompressible boundary layers as a linear combination of two functions viz

$$\frac{u}{u_\tau} = f\left(\frac{zu_\tau}{\nu}\right) + h(x,z) \qquad (3.4.1)$$

where f is the usual law of the wall and h is an arbitrary function of x, z except that it is negligibly small in some

3.4) contd.

narrow region near the wall. Coles points out that in certain special cases, notably for uniform pipe and channel flows and the boundary layer on a flat plate in a uniform stream, equation (3.4.1) is found from experiment to have the special form

$$\frac{u}{u_\tau} = f\left(\frac{zu_\tau}{\nu}\right) + g\left(\pi, \frac{z}{\delta}\right) \qquad (3.4.2)$$

where π is a flow parameter independent of x,z.

Coles made an extensive survey of mean velocity profile measurements in various two-dimensional boundary layer flows examining the form of the function h(x,z) and concluded that h(x,z) reduced to a second universal similarity law by which equation (3.4.1) can be amended to

$$\frac{u}{u_\tau} = f\left(\frac{zu_\tau}{\nu}\right) + \frac{1}{\kappa}\,\pi\, w\left(\frac{z}{\delta}\right) \qquad (3.4.3)$$

where π is now a profile parameter. The function w, which is tabulated by Coles and shown in figure (3.4.1), is called the law of the wake and is claimed to be common to all two-dimensional turbulent boundary layer flows and to be characteristic of the mean-velocity profile at separation or re-attachment. If the wake w is normalised so that

$$w(0) = 0 \quad , \quad w(1) = 2$$

$$\int_0^2 \frac{z}{\delta}\, dw = 1$$

Coles has shown the profile parameter π to be related to c_f and δ^* respectively by

3.4) contd.

$$\frac{U}{u_\tau} = \frac{1}{\kappa} \ell n \left(\frac{\delta u_\tau}{\nu}\right) + A + \frac{2\pi}{\kappa} \qquad (3.4.4)$$

$$\frac{\kappa \delta^* U}{\delta u_\tau} = 1 + \pi \qquad (3.4.5)$$

by which δ and π are uniquely defined. Equations (3.4.4,5) in effect provide a skin friction law.

Letting u_τ approach zero equations (3.4.3,5) reduce to

$$\frac{u}{U} = \frac{1}{2} w\left(\frac{z}{\delta}\right) \qquad (3.4.6)$$

which shows that at points of separation or reattachment the velocity profile is the pure wake function.

It is of interest to note that a defect law can be obtained from Coles' wake model equation (3.4.3) and written explicitly as

$$\frac{U-u}{u_\tau} = -\frac{1}{\kappa} \ell n \left(\frac{z}{\delta}\right) + \frac{\pi}{\kappa}\left(2 - w\left(\frac{z}{\delta}\right)\right)$$

$$= F\left(\frac{z}{\delta}, \pi\right) \qquad (3.4.7)$$

Equation (3.4.7) is not only valid within the logarithmic region but, according to Coles' formulation, will also apply now to all two-dimensional boundary layers and not only to equilibrium flows (i.e. it applies to the general boundary layer where π is a function of x and not just to equilibrium boundary layers where π is constant). The determination of ν and three of U, u_τ, δ, π will completely specify the velocity profile.

Having formulated a general theory for two-dimensional velocity profiles by means of combinations of the law of the wall

3.4) contd.

and the law of the wake Coles postulated as to how these might be applied to the yawed boundary layer. The general profile Coles tentatively wrote as

$$\underline{q} = \underline{q}_f + \underline{q}_w \qquad (3.4.8)$$

where $\underline{q}$ is the velocity vector parallel to the wall on which the boundary layer is developing while $\underline{q}_f$ corresponds to the law of the wall and $\underline{q}_w$ to the law of the wake. The law of the wall asserts that close to the wall the flow remains basically unidirectional as the boundary layer is traversed in the z direction

$$\underline{q}_f = \underline{q}_\tau \; f\left(\frac{z q_\tau}{\nu}\right) \qquad (3.4.9)$$

$\underline{q}_\tau$ being the vector having the same direction as the limiting surface shear stress and q_τ is the usual friction velocity

$$\underline{\tau}_0 = \rho \; q_\tau \; \underline{q}_\tau \qquad (3.4.10)$$

The contribution to the resultant velocity from the wake component of the flow $\underline{q}_w$, which again presumably will be negligibly small close to the wall, Coles postulates will be of the form

$$\underline{q}_w = \frac{1}{\kappa} \; \pi \; \underline{q}_\tau \; w\left(\frac{z}{\delta}\right) \qquad (3.4.11)$$

where π, a function of two space co-ordinates (x and y), was defined as a tensor in the general three-dimensional case. It readily follows that the generalised friction law is

$$\underline{Q} = \underline{q}_\tau \; f\left(\frac{\delta q_\tau}{\nu}\right) + \frac{2}{\kappa} \; \pi \; \underline{q}_\tau \qquad (3.4.12)$$

The existence of a region close to the wall in the three-dimensional boundary layer in which the velocity

3.4) contd.

profile is approximately collateral is quite well substantiated and it can also be reliably assumed that the outer part of this region (and the inner part of the skewed profile) is logarithmic in character [13,21,22]. The logarithmic law of the wall holds as in two dimensions except that the region over which it is operative is more restricted.

The evidence concerning the law of the wake in three dimensions is not quite so definitive. In an investigation of the velocity profiles in plane of symmetry flows Pierce [23] shows that the law of the wake is applicable except near separation. In analysing velocity profiles in the skewed boundary layer however there is a tendency to examine the wake function by considering

$$\frac{2q\ \sin\beta}{Q\ \sin\beta_o} = w\left(\frac{z}{\delta}\right) \qquad (3.4.13)$$

(see figure (3.5.1) for notation used) which is immediately deducible from equation (3.4.8). Such an approach is surely misleading since while Coles presumably intended the law of the wake to take account of the streamwise velocity profile the left hand side of equation (3.4.13) can be associated more with the crossflow velocity profile for small or moderate angles of yaw (i.e. β_o). This might be appreciated more when it is pointed out that the left hand side of equation (3.4.13)) becomes singular for the special case of collateral flows. It is interesting to note that the curves obtained by plotting this expression as a function of z/δ (see figure (3.4.1)) although decidedly different from the wake function do have a typical shape [13,22]. Coles has outlined a method of analysing skewed velocity profiles [15]

3.4) contd.

to obtain a more realistic test of the wake function and the data analysed in this way ([15] and discussion in [21]) although not providing any definite confirmation of the applicability of the wake function to three-dimensional flows does realise profiles which are more wake-like in form than those obtained from equation (3.4.13).

3.5) Johnston's triangular model for yawed flows.

A scheme to describe the yawed velocity profile which has met with considerable success is the so-called triangular model. Johnston [21] has established that if data from a yawed velocity profile are plotted in polar co-ordinates (i.e. if u_2 is plotted as a function of u_1 to obtain in effect the locus of the tip of the velocity vector projected on to the wall) then the points fall along two straight lines (see figure (3.5.1)). Thus to specify u_2 as a function of u_1 we need only know the values assumed by β_0, γ, where γ (the outer angle of the triangle) is the parameter denoting the shearless nature of the flow ~~and~~ which is related by Johnston to the main flow turning angle α (radians) by

$$\tan \gamma = -2\alpha$$

for circular-arc-shaped streamlines. The second angle β_0 is related to the frictional character of the flow.

We will adopt the notation introduced by Johnston and denote the two separate regions of the triangle by I and II and refer to quantities at the apex of the triangle by appending a suffix p as in figure (3.5.1). The outer part of region I was considered by Johnston to be in the logarithmic region (although the maximum value of $z_p\, q_\tau/\nu$ is only 16 - whereas Hornung and Joubert

3.5) contd.

encountered $z_p\, q_T/\nu$ values as high as 150). The relative sizes of regions I and II is considerably misrepresented by the polar plot and because u_1 increases very rapidly close to the wall this region is very narrow indeed, so narrow in fact that it is difficult to obtain extensive measurements corresponding to the inner side of the triangle.

3.6) Velocity defect law for yawed flows.

Hornung and Joubert [13] analysed the data from their own experiment to examine the plausibility of a three-dimensional defect law of the form

$$\frac{|\underline{Q} - \underline{q}|}{|\underline{Q} - \underline{q}_p|} = F\left(\frac{z}{\delta}\right) \qquad (3.6.1)$$

where $\underline{q}_p$ is $\underline{q}$ at the point at which the ~~defect~~ component of $\underline{Q} - \underline{q}$ tangential to $\underline{Q}$ attains a maximum value. The form of the left hand side of equation (3.6.1) in fact treats the velocity relative to the moving external stream which according to the outer edge of Johnston's triangle is collateral (at 'p' as can be seen from figure (3.5.1) the magnitude of $\underline{Q} - \underline{q}$ attains its maximum). The data of Hornung and Joubert showed little scatter when plotted according to equation (3.6.1) but Johnston (in a discussion in [24]) subsequently showed by analysing data from various sources that the scatter was considerable.

CHAPTER FOUR

A FINITE DIFFERENCE METHOD OF SOLUTION OF THE BOUNDARY LAYER EQUATIONS.

4.0) Introduction.

It is the purpose of the present chapter to describe a method of solution of the boundary layer equations which will be suitable for the computation of two- or three-dimensional, laminar or turbulent boundary layers.

Sections 4.1-2 of this chapter are concerned with developing a grid upon which finite difference approximations to the boundary layer equations can be conveniently based. The grid moreover must be devised in such a way that it can expand or contract to keep pace with boundary layer thickness development. Having transformed the boundary layer equations in accordance with this grid system in section 4.3 a finite difference scheme is proposed whose principle features are the use, for the boundary condition at the wall in the turbulent boundary layer, of the logarithmic law of the wall and the introduction of a transformation which considerably simplifies the calculation of velocity profiles in the three-dimensional boundary layer. Having described in section 4.4 how the law of the wall is to be used as a boundary condition the next four sections proceed to discuss in some detail the proposed finite difference scheme. The present method is similar to that of Spalding and Patankar [10] in that it solves the boundary layer equations in conjunction with an effective viscosity function, but it differs in the way it treats the logarithmic law at the wall and also in that it abandons the stream function in favour of a geometric transformation to account for boundary layer growth since the former is inapplicable to three dimensions.

Section 4.9 discusses a computer program which has been written for the IBM S360/65 computer to calculate boundary layer development using the method described. A copy of the program is included in the appendices together with a detailed description

4.0) contd.

of its structure.

4.1) Solution mesh.

It is proposed that the solution scheme to be presented will solve the boundary layer equations over a rectangular area of the wall in question, the velocity components being calculated at the nodes of a mesh positioned over this rectangle. The plane of the wall is taken to be the plane $z = 0$, where z is assumed to be measured positive into the body of the fluid, and the directions of the x and y axes are parallel to the wall and such that the primary direction of flow is taken as the direction of the x-axis and the co-ordinate axes x,y,z form a right hand set.

The parabolic nature of the boundary layer equations necessitates a marching type solution procedure so it is proposed to march in the x direction and to confine the calculation between y = constant planes. With such an arrangement it will be possible to set up a three-dimensional rectangular mesh, aligned with the rectangular axes, over the solution space and base the finite difference approximations to the boundary layer equations on velocity components at the nodes of the mesh. At each marching step u,v,w will be found at the nodes of the mesh in an x = constant plane before advancing to the next plane a distance f downstream to repeat the procedure. In what follows the grid at $x = x_\ell$ (i.e. the ℓ th step) will be referred to as solution face ℓ. At any solution face u,v will be calculated at all node points on this face, while w will be calculated at points on the plane midway between adjacent solution faces where the mesh lines intersect this plane.

4.1) contd.

Reference to any particular node can be made by enumerating its grid reference (ℓ,m,n) where ℓ denotes the solution face on which the node falls, m denotes the vertical line on this face on which the node falls (this line will be referred to as section m on face ℓ or simply section (ℓ,m)) and n denotes the number of the node as enumerated from the wall (point n on section (ℓ,m) or point (ℓ,m,n)). Thus the classification: solution face, section, point describes the mesh in a manner suitable for the proposed solution scheme (see figure 4.1.1).

The grid spacing in the x,y,z directions will be denoted by f,g,h respectively and the number of sections on a solution face by M and the number of points on a section by N.

The above solution mesh must obviously contain the boundary layer i.e. the region over which significant changes in u,v occur. This will mean that since the boundary layer growth downstream will be unknown at the commencement of the solution some means of adjusting the grid as the solution progresses must be devised. This will be discussed further in section 4.2. It might also be noted here that even across a solution face appreciable differences in boundary layer thickness may occur. To compensate for this and also to allow for more points over the region where large changes in velocity occur i.e. near the wall, the following scheme is proposed: at each ~~section~~ solution face the grid illustrated in figure 4.1.2 in which a specified number of the mesh intervals near the wall have been subdivided will be used.

Although the same grid will be used at each ~~section~~ solution face

4.1) contd.

its vertical scale, as dictated by h, will be subject to variation in a manner to be discussed in the next section i.e. h should more correctly be replaced by h_ℓ. This will compensate for boundary layer growth as the calculation proceeds.

The grid in figure 4.2 has been obtained by dividing the λ increments of width h near the wall each into ω smaller increments of width h/ω. It should be noted that N is now taken to refer to the number of points at each section and not to the number of h increments.

4.2) Derivation of an adjustable mesh to accommodate boundary layer growth.

As a simple means of adjusting the grid spacing so as to compensate for the effect of boundary layer growth the mesh illustrated in figure 4.2.1 will be used. The figure shows a cross section through a y = constant plane - the cross sections through all such planes being the same. Planes radiating from the line x = X, z = 0 will constitute the grid planes which will be positioned so that they approximately keep pace with the boundary layer growth between solution faces at $x = x_\ell$ where velocity profiles will be known and $x = x_{\ell+1}$ where they are to be calculated. This can readily be arranged by varying the position of the line x = X, z = 0 and the inclinations of the planes. The grid can also be chosen so as to ensure that the z increments will be constant over the solution face at $x = x_\ell$ as well as at $x = x_{\ell+1}$ although the z increment will obviously not be the same at both faces. Using such a scheme it will be possible to adjust the grid at each step to progressively allow for changes in the rate of growth of the

4.2) contd.

boundary layer.

Figure 4.2.2 shows the possibilities of the proposed mesh when inclined mesh planes and subdivided mesh intervals near the wall are incorporated.

Transforming from z to ξ using

$$z = a\xi(x - X) \qquad (4.2.1)$$

will produce in x,y,ξ co-ordinate system the grid discussed above since in a plane x = constant ξ is simply proportional to z and surfaces ξ = constant are planes passing through the line x = X, z = 0. Knowing the grid spacing required at solution faces ℓ and $\ell+1$ to be h_ℓ and $h_{\ell+1}$ respectively then X is determined by noting that at the first grid plane from the wall we have $\xi = \xi_1$ say where ξ_1 is a constant so that at $x = x_\ell$ equation (4.2.1) becomes

$$h_\ell = a\xi_1(x_\ell - X)$$

and at $x = x_{\ell+1}$

$$h_{\ell+1} = a\xi_1(x_{\ell+1} - X).$$

Dividing these last two expressions

$$\frac{h_{\ell+1}}{h_\ell} = \frac{x_{\ell+1} - X}{x_\ell - X}$$

so that

$$X = x_\ell - \frac{h_\ell f}{h_{\ell+1} - h_\ell}$$

since $x_{\ell+1} = x_\ell + f$. The arbitrary scaling factor a is now chosen so that the increment in ξ between adjacent ξ = constant mesh planes is the same as the z increment at $x = x_\ell$ thus equation (4.2.1) becomes

$$h_\ell = ah_\ell(x_\ell - X)$$

4.2) contd.

so that

$$\frac{1}{a} = \frac{h_\ell f}{h_{\ell+1} - h_\ell}$$

and the required transformation is

$$z = \xi \left(\frac{h_{\ell+1} - h_\ell}{h_\ell f} (x - x_\ell) + 1 \right) \qquad (4.2.2)$$

Since the boundary layer growth between adjacent solution faces will not be known before the solution method proceeds to calculate velocity components at solution face $\ell+1$ it will be necessary to make an initial guess at the transformation equation (4.2.2). How the mesh is adjusted more precisely to accommodate the thickest part of the boundary layer at successive solution faces will be dealt with in a later section.

4.3) Transformation of the boundary layer equations.

To implement the grid described in section 4.2 the boundary layer equations will be transformed so that the perpendicular distance measured from the wall will be substituted for by ξ using the relation

$$z = \xi(ax + b) \qquad (4.3.1)$$

where a and b are constants (given by equation (4.2.2)) chosen to regulate the grid and scale ξ conveniently. The boundary layer equations (1.1.10-12) transformed into x, y, ξ co-ordinates are

4.3) contd.

$$u\frac{\partial u}{\partial x} - \frac{a\xi}{ax+b}\,u\frac{\partial u}{\partial \xi} + v\frac{\partial u}{\partial y} + \frac{1}{ax+b}\,w\frac{\partial u}{\partial \xi} =$$

$$U\frac{\partial U}{\partial x} + V\frac{\partial V}{\partial x} + \frac{1}{(ax+b)^2}\frac{\partial}{\partial \xi}\left(\nu_e \frac{\partial u}{\partial \xi}\right) \qquad (4.3.2)$$

$$u\frac{\partial v}{\partial x} - \frac{a\xi}{ax+b}\,u\frac{\partial v}{\partial \xi} + v\frac{\partial v}{\partial y} + \frac{1}{ax+b}\,w\frac{\partial v}{\partial \xi} =$$

$$U\frac{\partial U}{\partial y} + V\frac{\partial V}{\partial y} + \frac{1}{(ax+b)^2}\frac{\partial}{\partial \xi}\left(\nu_e \frac{\partial v}{\partial \xi}\right) \qquad (4.3.3)$$

$$\frac{\partial u}{\partial x} - \frac{a\xi}{ax+b}\frac{\partial u}{\partial \xi} + \frac{\partial v}{\partial y} + \frac{1}{ax+b}\frac{\partial w}{\partial \xi} = 0 \qquad (4.3.4)$$

where the assumption that $\nu_{ex} = \nu_{ey} = \nu'_e$ has been introduced.

At this stage it is proposed to introduce a contraction into the transformed equations which will considerably simplify the finite difference scheme to be considered later. We write

$$q\frac{\partial}{\partial s} \equiv u\frac{\partial}{\partial x} + v\frac{\partial}{\partial y} \qquad (4.3.5)$$

where q is the magnitude of the vector sum of velocity components u and v

$$q = (u^2+v^2)^{\frac{1}{2}} \qquad (4.3.6)$$

The equation (4.3.5) is suggestive of streamline co-ordinates but since no account is taken of the w component of velocity this is not quite so. The line along which the derivate $\frac{\partial}{\partial s}$ is to be taken is the locus of points in a ξ = constant plane at which the vector sum of the velocity components u and v is tangential to the line, and the increment in s along this line is given by

$$\delta s = (\delta x^2 + \delta y^2)^{\frac{1}{2}} \qquad (4.3.7)$$

Without any apology, in what follows, we refer to such lines

4.3) contd.

as streamlines despite the fact that these so-called streamlines can only be related to actual streamlines in the limiting case at the wall and in the freestream. In addition we will replace ν_e in equations (4.3.2-3) by ν_e' such that

$$\nu_e = (ax+b)\nu_e' \tag{4.3.8}$$

so that the three-dimensional turbulent boundary layer equations now become

$$q\frac{\partial u}{\partial s} + \frac{-a\xi u + w}{ax+b}\frac{\partial u}{\partial \xi} = U\frac{\partial U}{\partial x} + V\frac{\partial V}{\partial x} + \frac{1}{ax+b}\frac{\partial}{\partial \xi}\left(\nu_e'\frac{\partial u}{\partial \xi}\right) \tag{4.3.9}$$

$$q\frac{\partial v}{\partial s} + \frac{-a\xi u + w}{ax+b}\frac{\partial v}{\partial \xi} = U\frac{\partial U}{\partial y} + V\frac{\partial V}{\partial y} + \frac{1}{ax+b}\frac{\partial}{\partial \xi}\left(\nu_e'\frac{\partial v}{\partial \xi}\right) \tag{4.3.10}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} - \frac{a\xi}{ax+b}\frac{\partial u}{\partial \xi} + \frac{1}{ax+b}\frac{\partial w}{\partial \xi} = 0 \tag{4.3.11}$$

It should be noted that the two momentum equations now contain only derivatives with respect to s and ξ (except for the pressure terms which are prescribed functions). This will be seen to be an advantage when approximating to the momentum equations in the three-dimensional case since derivatives with respect to x and y are not present explicity. To preserve this state of affairs it should be appreciated that only a transformation of the z co-ordinate is applicable.

Equations (4.3.9-11) are the forms of the boundary layer equations that will be solved using finite difference techniques.

4.4) Wall boundary condition.

Before proceeding to approximate to the three-dimensional boundary layer equations (4.3.9-11) a number of complications which arise in the section of the boundary layer adjacent to the wall must be considered. Firstly, because close to the wall in the turbulent boundary layer the changes in velocity (and also in velocity gradient normal to the wall) are so great over small distances normal to the wall, it is not possible to obtain an estimate of derivatives normal to the wall at a point simply by subtracting functional values at points equidistant on either side of the point of interest. In particular the velocity u at a small distance h from the wall divided by h will not provide a realistic estimate of the gradient $\frac{\partial u}{\partial z}$ at a point a distance $\frac{1}{2}h$ from the wall (remembering u at the wall is zero). The same problem does not however arise in laminar boundary layers since although changes in velocity near the wall are great the velocity gradient normal to the wall is approximately constant over a small distance close to the wall.

Secondly, owing to the presence in the turbulent boundary layer of a laminar sublayer, it is not feasible to provide an empirical relationship for the effective viscosity within a narrow region close to the wall.

The consequence of the above complications is that it would be extremely difficult in a finite difference scheme to impose as the inner boundary the fact that all velocity components must vanish at the wall. To overcome this problem the present method proposes that the logarithmic law of the wall should be used as the wall boundary condition. This will mean

4.4) contd.

that the region between the wall and the inner limit of the logarithmic region, which includes the laminar sublayer, need not enter into the finite difference scheme and the difficulties mentioned above it is to be expected will not arise. The present section is thus concerned with interpreting the logarithmic law of the wall in such a way as to make it accessible as a boundary condition and to generate a number of relations that will be required in the development of the finite difference scheme.

It is well established (section 3.2) that in the two-dimensional turbulent boundary layer within a region close to the wall but not adjacent to it points from the velocity profile fall along the logarithmic curve:

$$\frac{q}{q_\tau} = \frac{1}{\kappa} \ell n \frac{z q_\tau}{\nu} + A \qquad (4.4.1)$$

where q is the component of the velocity parallel to the wall (and hence in the direction of the mainstream) at a distance z from it, q_τ is the so called friction velocity, ν is the kinematic viscosity and κ, A are empirical constants. It has been noticed, and it is particularly well illustrated by Johnson's polar plots (section 3.5), that even in three-dimensional boundary layers there is a region close to the wall in which the flow is essentially coplanar i.e. the direction of $\underline{q}$ remains fixed as this region is traversed perpendicularly to the wall and Coles has suggested that the logarithmic law is valid within this region where it assumes the form of equation (4.4.1) where q is now given by equation (4.3.6). It will be assumed here for the purpose of the present computational scheme that, in the three-dimensional

4.4) contd.

boundary layer, there is a region close to the wall in which the flow is both coplanar and where points from the velocity profile fall upon the logarithmic law of the wall.

As has been stated previously the finite difference approximation to the three-dimensional boundary layer equations is not attempted at all grid points up to the wall but only at each section up to some grid point which is known to be the point closest to the wall which could, on the basis of the accepted limits for the logarithmic region, be regarded as being within the logarithmic region. This point will be referred to as the log-point and its grid reference will be denoted by $n = n^*$. It is within the region bounded by the grid lines above and below this log-point that we must assume that the flow is both coplanar and the logarithmic law is operative.

Transforming the logarithmic law of the wall viz

$$\frac{q}{q_\tau} = \frac{1}{\kappa} \ell n \frac{(ax+b)\xi q_\tau}{\nu} + A \qquad (4.4.2)$$

it is in the same co-ordinate system adopted in the boundary layer equations (4.3.9-11). Rewriting equation (4.4.2) as

$$u = \frac{u}{q} \frac{q_\tau}{\kappa} \left(\ell n \frac{(ax+b)\xi q_\tau}{\nu} + \kappa A \right) \qquad (4.4.3)$$

or in a similar form in which u is replaced by v, we have an expression for u or v which when applied in the vicinity of the log-point only involves ξ explicitly on the right hand side since when q is coplanar

$$\frac{u}{q} \quad , \quad \frac{v}{q}$$

are functions of x,y only as is q_τ. Thus

4.4) contd.

differentiating equation (4.4.3) with respect to ξ it is possible to write

$$\frac{\partial u}{\partial \xi} = \frac{u}{q} \frac{q_T}{\kappa \xi} \qquad (4.4.4)$$

which with

$$\epsilon = \frac{\kappa q_n^*}{q_T} \qquad (4.4.5)$$

can be written

$$\frac{\partial u}{\partial \xi} = \frac{u_n^*}{\epsilon \xi} \qquad (4.4.6)$$

where suffix n in q_n etc refers to the point at which q is evaluated. When applying this last expression at any section u_n^* and ϵ should be those values obtained at that section. A point of particular interest in equation (4.4.6) is the resemblance of the right hand side to the finite difference approximation to the derivative.

Another expression which will be required at the log-point is obtained from the definition of the operator $\frac{\partial}{\partial s}$ (equation (4.3.5)) and equation (4.4.3) and is

$$\frac{\partial u}{\partial s} = f_1 + \ell n \xi \; f_2 \qquad (4.4.7)$$

where f_1, f_2 are functions of x,y only, and from which it readily follows that

$$\ell n \frac{\xi_{n^*}}{\xi_{n^*+1}} \left.\frac{\partial u}{\partial s}\right|_{n^*-1} + \ell n \frac{\xi_{n^*+1}}{\xi_{n^*-1}} \left.\frac{\partial u}{\partial s}\right|_{n^*} + \ell n \frac{\xi_{n^*-1}}{\xi_{n^*}} \left.\frac{\partial u}{\partial s}\right|_{n^*+1} = 0 \qquad (4.4.8)$$

4.4) contd.

with a similar expression in which u is replaced by v. From equation (4.4.2) we will also require the following relation

$$q_{n^*-1} = q_{n^*}\left(1 + \frac{1}{\epsilon}\,\ell n\,\frac{\xi_{n^*-1}}{\xi_{n^*}}\right) \qquad (4.4.9)$$

We are now faced with the following situation: the boundary layer momentum equations can be approximated to at the log-point using only velocity components at and above this point and because points below the log-point will not be available use will have to be made of the relationships contained in equations (4.4.6) and (4.4.8-9) to make up this deficiency. Equations (4.4.6) and (4.4.9) do however demand that ϵ i.e. q_τ be known so that the logarithmic law (4.4.2) must be solved for q_τ at the log-point i.e.

$$\frac{q_{n^*}}{q_\tau} = \frac{1}{\kappa}\,\ell n\,\frac{(ax+b)\xi_{n^*}\,q_\tau}{\nu} + A$$

must be solved for q_τ. This last equation can alternatively be rewritten as

$$\left.\begin{aligned} \epsilon &= -\,\ell n\,\epsilon + B \\ \text{where} \qquad B &= \ell n\,\frac{\kappa(ax+b)\xi_{n^*}\,q_{n^*}}{\nu} + \kappa A \end{aligned}\right] \qquad (4.4.10)$$

and can be solved for ϵ using the iterative scheme

$$\epsilon^{(r+1)} = \epsilon^{(r)}\left(1 - \frac{\epsilon^{(r)} + \ell n\,\epsilon^{(r)} - B}{1 + \epsilon^{(r)}}\right) \qquad (4.4.11)$$

once B is prescribed. The iterative scheme represented by equation (4.4.11) has been obtained from an application of the Newton-Raphson iterative method and can be shown to have a quadratic rate of convergence. We might also note here that the use of the

4.4) contd.

above scheme for prescribing the inner boundary condition in the momentum equations has the advantage of implicitly supplying a value for the coefficient of friction at the wall.

The inner boundary condition in the continuity equation is slightly different from that in the momentum equation in that we will be integrating the former with respect to ξ through the thickness of the boundary layer which will demand that we integrate from the wall and thus impose as the inner boundary condition the fact that the velocity components vanish at the wall. It would not be expected that this approach would be at all impracticable but in order to be consistent with the method in which the momentum equations were treated we choose not to integrate from the wall i.e. to integrate through all the nodes from and including the wall, but to impose the inner boundary condition effectively at $n = n^*$. This is readily accomplished if it is assumed that to a reasonable degree of approximation the q component of velocity between the wall and the point $n = n^*$ can be represented by a power law of the form

$$\frac{q}{q_{n^*}} = \left(\frac{\xi}{\xi_{n^*}}\right)^{\frac{1}{\epsilon}} \qquad (4.4.12)$$

which ensures agreement with the logarithmic law of the wall at the log-point in q and $\frac{\partial q}{\partial \xi}$. The form of equation (4.4.12) anticipates ϵ to be of the order of 6 or 7 and this is of particular relevance when considered in the context of equation (4.4.6). The assumption of the power law equation (4.4.12) together with that of coplanar flow close to the wall enables us to integrate the continuity equation (4.3.11) with respect to ξ and deduce that

(ignoring the dependence of ϵ on x, y)

4.4) contd.

$$w = -\frac{ax + b}{1 + \frac{1}{\epsilon}} \xi\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} - \frac{a\xi}{ax+b}\frac{\partial u}{\partial \xi}\right) \qquad (4.4.13)$$

which can be used to obtain w at the log-point, together with

$$\frac{w}{w_{n^*}} = \left(\frac{\xi}{\xi_{n^*}}\right)^{1 + \frac{1}{\epsilon}} \qquad (4.4.14)$$

for w between the log-point and the wall.

The interpretation of the logarithmic law of the wall as the inner boundary condition obviously only applies to turbulent flow. Laminar flow could be treated in a similar fashion simply by substituting a linear relationship in place of the logarithmic one (since the velocity gradient close to the wall in a laminar boundary layer is essentially constant) and generally putting ϵ equal to unity. It should be noted that this is essentially no different to applying a zero velocity condition at the wall.

4.5) General discussion of the solution scheme.

Having, in sections 4.1-3 of this chapter, derived a mesh upon which a finite difference scheme can be based and having transformed the boundary layer equations to facilitate the use of this mesh it is now possible to generate a solution scheme.

The parabolic nature of the boundary layer equations makes it necessary when solving these equations using finite difference techniques to employ a marching type solution procedure and to do this we generalise the more well-known implicit schemes to the three variable non-linear problem. Thus knowing velocity prfiles at all sections on a solution face profiles at the next solution face can be calculated and so on downstream.

4.5) contd.

While solving for the velocity profiles at each solution it will be desirable, primarily because of the non-linear features of the boundary layer equations, to iterate to the correct solution from an initial guess of it. An iterative scheme will also enable the grid transformation, equation (4.3.1), to be adjusted to accommodate the boundary layer growth precisely and in addition allow the inner boundary condition as described in section 4.4 to be applied correctly. In addition with an implicit scheme the iteration can be repeated until the accuracy of the solution is within a required tolerance.

At each solution face individual sections will be considered in turn and corresponding to every point on each section finite difference approximations will be made to the momentum equations. This will produce a system of linear algebraic equations involving the unknown u,v components of velocity at all points on each section; the solution of these linear equations will provide it is anticipated better estimates of these same velocity components. Having iterated at all sections for u,v finite difference approximations will then be made to the continuity equation supplying in a similar way better estimates of the w component of velocity. It is expected that successive repetitions of the above procedure will provide an iterative scheme which will converge to the correct velocity profiles.

Details of the solution scheme are provided in the following two sections.

4.6) Finite difference approximations to the three-dimensional momentum equations.

The present section is concerned with a means of improving approximations to the u and v components of velocity at points on a solution face when the u and v profiles are known at all sections on the adjacent upstream solution face (denoted as solution face ℓ). This will be done as has already been mentioned by setting up finite difference approximations to the momentum equations corresponding to the points on each profile. A means of improving the approximations to the w velocity components, which will be stored on the plane midway between faces ℓ and $\ell+1$, is to be discussed in the next section.

There are numerous ways of setting up a finite difference approximation to differential equations especially when the equations are non-linear, and depending on how it is done will determine the rate of convergence of the iteration process. To enable the most attractive scheme to be determined or at least to provide some room to manoeuvre it is proposed to introduce into the scheme to be described a number of weighting factors the variation of which will it is anticipated lead to the development of a satisfactory solution scheme.

We now proceed to set up the finite difference approximation to the momentum equations corresponding to the n th point at section m (figure (4.6.1)). Since the momentum equations are written in streamline co-ordinates the finite difference approximation need necessarily be based on the streamline through the point of interest. It is thus necessary to fit from the n th point on section $(\ell+1,m)$ a streamline back to face ℓ and calculate:

4.6) contd.

1) the x,y co-ordinates of the point where the momentum equations are to be approximated on this streamline

2) the point where the streamline intercepts the n th grid line on face ℓ and

3) the length of the streamline

i.e. the values of α,β,γ,s from figure (4.6.1) need be known. The method used to fit the stremaline will be discussed in Appendix A1 and for the moment it will be assumed that α,β,γ,s can be calculated.

It can be seen from the diagram that the point at which the momentum equations are to be approximated can be varied by changing the value of the weight $\psi_1(\phi_1 = 1 - \psi_1)$, the well known Crank-Nicholson scheme being based on $\psi_1 = 0.5$. It is also apparent that the streamline through point n on section $(\ell+1,m)$ is not necessarily the same as those at points n - 1 or n + 1 on the same section. However, for the purpose of the approximations to the momentum equations at point n, they will be assumed to be the same.

In order to approximate to the momentum equations at the point of current interest the quantities indicated in figure (4.6.2) are required where the notation is self-explanatory. The quantities necessary on section $(\ell,m+\alpha)$ can be obtained by interpolating between the known profiles at face ℓ and those on section $(\ell+1,m)$ are provided from the last iterated solutions at face $\ell+1$. Values assumed by the transformed effective viscosity function ν_e' (where in figure (4.6.2) the dash and the subscript have been omitted for convenience) will have to be evaluated from currently available velocity components on the basis of some

4.6) contd.

hypothetical function in the case of turbulent flow.

The actual finite difference approximation to the momentum equation corresponding to the x direction is given in Appendix A2 together with the resulting linear equation relating u_{n-1}^{r+1}, u_{n}^{r+1}, u_{n+1}^{r+1} (where the superscripts denote iteration, the section being understood as $(\ell+1,m)$). The equations generated to relate the iterated v components of velocity are very similar. Hence it has been found necessary only to make brief mention of these at the end of Appendix A2. Table 4.6.1 lists the relaxation factors used and provides some measure of explanation of their relevance. The manner in which weights were used was made partly in reference to 'Difference Methods for Initial-Value Problems' by Richtmyer [25] and the allusions to particular cases is based on the simpler finite difference schemes presented by Richtmyer.

Remembering what has been stated in section (4.4) with regard to the points where the momentum equation will be approximated to it can be seen that equation (A2.2) can be applied for

$$n = n^*+1,\ n^*+2,\ \ldots\ N-1$$

slight modifications being necessary when $n = \lambda\omega$. It might be pointed out now that introducing the transformation

$$q \frac{\partial}{\partial s} \equiv u \frac{\partial}{\partial x} + v \frac{\partial}{\partial y}$$

enables equations to be set up relating the iterated u,v velocity components at each section on face $\ell+1$ independently of each other and also of the same unknowns at adjacent sections, coupling between the sections being provided via the continuity equation. It is considered that this is a significant simplification

4.6) contd.

in the present method.

To provide an approximation of the momentum equations at the log-point ($n = n^*$) it is necessary to make use of some of the relations developed in section 4.4 where the logarithmic law of the wall was used to generate relationships between different quantities in the neighbourhood of the log-point. The approximations will be made to the momentum equations on the assumption that u and v velocity components will not be available for incorporation into the finite difference scheme at $n = n^* - 1$. The resulting approximations to the terms in the momentum equation are given in Appendix A3, and Appendix A4 then explains how they are to be calculated for laminar flows.

Applying equation (A3.6) at $n = n^*$ and equation (A2.2) at the points stated above (remembering that u_N is prescribed as the freestream boundary condition) there results a system of $N - n^*$ linear equations in $N - n^*$ unknowns:

$$u_n^r \qquad n = n^*, n^* + 1, \ldots N-1$$

These $N - n^*$ equations form a tri-diagonal system except that when $\omega > 1$ the equation corresponding to the point $(\ell, m, \lambda\omega)$ has one term displaced off the triple-diagonal since it relates

$$u_n^r \qquad n = (\lambda-1)\omega,\ \lambda\omega,\ \lambda\omega+1$$

however this is easily remedied as explained in Appendix A2 so that the system of equations can now be assumed to be tri-diagonal and solved accordingly (see Appendix A5). The velocity profile at section $(\ell+1, m)$ can then be completed by using the law of the wall (see section 3.2) to generate

4.6) contd.

u_n^r, v_n^r for n = 1,2, ... n*-1.

4.7) Finite difference approximation to the equation of continuity.

Having obtained iterates for the u,v velocity components in section 4.6 the continuity equation must now be approximated to in order that improved estimates of the w component of velocity can be determined. The continuity equation being linear in w means that knowing u and v components across any two adjacent solution faces w profiles can be obtained midway between these faces by directly integrating the continuity equation with respect to ξ.

The equation of continuity is left in its rectangular cartesian co-ordinate form (equation (4.3.11)) and approximated to at the point $(\ell+\frac{1}{2}, m, n-\frac{1}{2})$ as shown in figure (4.7.1). The values of the components shown can be found by suitably averaging known values and can be used to provide an approximation to the continuity equation thus

$$\frac{u_3-u_1}{f} + \frac{v_3-v_1}{2g} - \frac{a\,\xi_{n-\frac{1}{2}}}{a\,x_{\ell+\frac{1}{2}}+b}\,\frac{u'_{\ell_3}-u'_{\ell_1}}{h} + \frac{1}{a x_{\ell+\frac{1}{2}}+b}\,\frac{w_n^{r+1}-w_{n-1}^{r+1}}{h} = 0 \qquad (4.7.1)$$

From this equation values of w at points successively further from the wall can be calculated until the profile at section $(\ell+\frac{1}{2}, m)$ is complete. Repetition at different sections will enable iterated profiles across the whole mid-face to be determined. Equation (4.7.1) might be applied from the wall (n=1) using the condition that w=0 at the wall but in preference to this we will interpret the wall boundary condition used for the momentum equation in a way which will make it applicable here and apply

4.7) contd.

equation (4.7.1) only for

$$n = n^*+1,\ n^*+2,\ \ldots\ N$$

We have already shown that at the log-point the wall condition gives equation (4.4.13) which we approximate to by

$$w_n = -\frac{ax_{\ell+\frac{1}{2}}+b}{1+\frac{1}{\epsilon}}\ \xi_n\left(\frac{u_3-u_1}{f}+\frac{v_3-v_1}{2g}-\frac{a\,u_2}{(ax_{\ell+\frac{1}{2}}+b)\epsilon}\right) \qquad (4.7.2)$$

to be applied at $n = n^*$, where the notation will be made clear by figure (4.7.2). Values of w between the log-point and the wall can be obtained by referring to equation (4.4.14)

$$w_n = \left(\frac{\xi_n}{\xi_{n^*}}\right)^{1+\frac{1}{\epsilon}} w_{n^*} \qquad (4.7.3)$$

which is applicable for $n = 1,2,\ \ldots\ n^*-1$.

It will be noticed that when equations (4.7.1,2) are applied at the end sections i.e. the sections denoted by m=1, m=M, values of v are required at points lying outside the solution space. Thus v profiles (or alternatively $\frac{\partial v}{\partial y}$ profiles) will need be specified as a boundary condition at the bounding y = constant planes to establish the flow of fluid into the solution space.

4.8) Recapitulation of the initial and boundary conditions.

We will now collect together for future reference the initial and boundary conditions that have arisen in the discussion of the present solution scheme. The conditions listed below are those relevant to the solution of the general three-dimensional boundary layer and considerable simplifications

4.8) contd.

can be expected when the scheme is used for the solution of the two-dimensional or pseudo-three-dimensional problems.

The only initial condition required is the specification of u and v velocity profiles at all sections across the initial solution face - w profiles are not required. The boundary condition to be provided in the freestream is the definition of the velocity components U,V at all points over the area of the wall of interest while at the bounding y = constant planes either v or $\frac{\partial v}{\partial y}$ need be specified. The latter condition, which is only applicable to the general three-dimensional boundary layer or to plane of symmetry flows, is more readily satisfied than might first appear to be the case and some of the methods that have successfully been used to provide this boundary condition can be found in the discussion of the three-dimensional boundary layers treated in Chapters Five and Six.

In addition of course we must yet specify some hypothesis for the effective viscosity.

4.9) The computer program.

The listing of a computer program that has been written in Fortran IV for the IBM S360/65 computer to calculate boundary layer development using the method outlined in this chapter is included as Appendix A6 while Appendix A7 contains a description of the program structure (with flow diagrams) and discusses the requirements necessary for the implementation of the program.

The program was written in such a way that it

4.9) contd.

would be able to cater for three-dimensional, pseudo-three-dimensional,plane of symmetry or two-dimensional boundary layers, whether laminar or turbulent simply by varying a few input parameters. All initial conditions need be specified by card input while the boundary conditions and the effective viscosity function are provided via subroutines which are referred to within the structure of the program. Empirical and physical constants, the mesh specification and solution weights and tolerances are all to be provided as card input. A facility has also been included to allow changes in forward steps, frequency of output, etc as the solution progresses. A thorough description of such matters is, as has been mentioned, contained in Appendix A7.

CHAPTER FIVE

TWO-DIMENSIONAL CALCULATIONS

5.0) Introduction.

Having programmed the solution scheme described in Chapter Four in Fortran for the IBM S360/65, it was first necessary to investigate the effects of step sizes and solution weights on the stability and rate of convergence of the iterative process before proceeding to determine those values for the empirical constants which would ensure the best possible agreement between calculation and experiment.

Since no particular difficulties have been found concerning the stability of the present calculations, no further consideration of this aspect of the solution scheme will be given in this chapter; some detailed observations relating to stability can be found in Appendix A8 however.

The choice of weights used in all the calculations to be discussed is shown in the last column of Table 4.6.1. Firstly it will be noticed that all the weights introduced arise in the two-dimensional scheme and so no further consideration need be given to this problem when three-dimensional calculations are being considered in Chapter Six. Although the values assigned to the weights $\psi_1, \psi_2, \ldots \psi_7$ were decisive in determining the most satisfactory scheme (especially ψ_1, ψ_5) the overall method was not particularly sensitive to any of these. The relaxation factor ψ_8 however did prove to have a critical effect on the rate of convergence of the calculation; $\psi_8 = 0.75$ was found to produce the best overall results.

We must next consider what values to assign to the empirical constants and determine the most efficacious effective viscosity model; an experiment ('E') of Schubauer and Spangenberg [26] was used as the original basis for this choice (Section 5.1). Because the constant κ appears in both the logarithmic law of the wall and

5.0) contd.

in Prandtl's mixing length concept, initial tests were made to determine if this constant need necessarily be the same in its two applications. In order to obtain a smooth curve for the logarithmic law of the wall (equation (3.2.6)) from the computed velocity profiles, it was found that in both instances κ should be the same and that it could be taken to assume its usually accepted value of 0.41. Similarly it was found that the constant A could satisfactorily assume its accepted value of 4.9. For the purpose of determining the log-point, the minimum value of $\frac{zq_T}{\nu}$ for which the logarithmic law of the wall could be assumed to be valid was taken to be 30.

Some difficulty was encountered however while attempting to find a representation for ν_e in the outer layer. Originally Clauser's representation for this region (equation (3.3.1)) was applied to the calculation of the above experiment. However, with the value of K quoted by Clauser (0.016) the shape factor H did not increase quickly enough, better results being given by K = 0.011. Since the empirical constants κ,A are quite well determined, it was not considered unreasonable to adjust K, which has little experimental verification, so as to ensure that the calculation agreed with experiment in this particular case. However when used to calculate anything other than retarding boundary layers this simple form for ν_e in the outer layer was not found to give satisfactory results. In addition, neither of the simple alternatives considered, where the outer layer was variously defined

1) $\ell = \kappa z \qquad z \leqslant d\delta^*$

$\ell = \kappa d\delta^* \qquad z \geqslant d\delta^*$

(where d is an empirical constant)

5.0) contd.

2) $\frac{\nu_e}{\nu} = \zeta \qquad \zeta \leqslant \frac{KU\theta}{\nu}$

$\nu_e = KU\theta \qquad \zeta \geqslant \frac{KU\theta}{\nu}$

(cf equation (3.3.5) where K is now a different empirical constant)

were found to offer any improvement. To effect a remedy to this situation it was decided to adopt equation (3.3.1) but to make K dependent on the pressure gradient, consequently K was made a function of

$$\Gamma = 10^4 \frac{\theta}{U} \frac{dU}{dx} \qquad (5.0.1)$$

viz

$$K = 0.016 + 0.00015\ \Gamma \qquad (5.0.2)$$

Equation (5.0.2) was formulated to ensure detailed agreement with experiment 'E' of Schubauer and Spangenberg (Section 5.1). Equation (5.0.2) is highly tentative, and may quite easily be replaced in the computer program by any other model that may be preferable, but has been found to give reasonable results in most of the calculations considered in this chapter.

The following sections of this chapter are then concerned with comparing the predictions of this calculation scheme with a number of experiments of varying boundary layer development. Also comparisons are made with predictions based on Head's method. Head's method has been used partly because a computer program was readily available for doing this and partly because the method has been shown to give good results when considered in the light of boundary layer calculation methods at large. The actual program used to compute boundary layers using the entrainment approach was one developed by Rolls-Royce

5.0) contd.

Limited and inevitably the constants and empirical functions used are not identical to those proposed in Head's original formulation of the method. In addition it must be pointed out that this latter program firstly assumes as an initial condition that H = 1.32, so that the method could not be applied to all the experiments considered and secondly, will not predict separation by predicting zero skin friction, this being avoided by the imposition of an upper limit on H(2.7). Separation is generally assumed to have occurred just before this maximum is achieved.

5.1) Schubauer and Spangenberg.

Schubauer and Spangenberg [26] investigated the effects of forced mixing (i.e. that induced by the introduction of fixed obstacles on the wall) on a boundary layer developing under an adverse pressure gradient leading to eventual separation. Three experiments (those denoted 'C', 'D', 'E'), made in the absence of forced mixing, will be considered here. These experiments were concerned with incompressible flows over a smooth flat wall and were, in the opinions of the authors, accurately two-dimensional. Thompson [2] in his review claims that only experiment 'D' is closely two-dimensional, although no more than slight discrepancies are exhibited by the other two. The experimental results for these three runs are plotted in figures (5.1.1.-9) together with the values recalculated by Thompson. No values for the skin friction are quoted in reference [26] the only indication being the comment that 'values of the local skin friction coefficients c_f ... were found to decrease monotonically from around 0.0032 at x = 0 to around 0.0003 at the indicated separation point. Failure to reach zero

5.1) contd.

is attributed to the fact that the dust method indicates the upstream extreme of a fluctuating separation point'.

Experiment 'E' was, as has already been mentioned, used to determine values for the empirical quantities contained in the present calculation method and this should be borne in mind in the following comparisons between the present calculation method and experiment.

Included in figures (5.1.1-9) are predictions both of the present method and also calculations based on the method of Head. Both sets of calculations predict separation with reasonable accuracy in experiments 'C' and 'D' whereas, while predictions for experiment 'E' tend toward separation at the required point, the present method recovers just before separation is achieved. Neither calculation is able to correctly predict the sudden increase in shape factor immediately prior to separation in 'C' or 'E' although it might be expected that the discrepancy here is caused by the three-dimensional effects indicated by Thompson, since in the experiment which was shown to be precisely two-dimensional ('D') H is predicted accurately.

Overall both methods agree quite closely and give reasonable agreement with experiment. The calculation based on the entrainment approach consistently predicts an R_θ growth slightly greater than the present method, and similarly with H development to a less marked extent. On the basis of the three present comparisons it is difficult to say which is giving the better results.

Plotted in figure (5.1.8) is the H development predicted using K = 0.011 (the constant value chosen to ensure the best overall agreement with experiment and obtain separation

5.1) contd.

at the required point). It can be seen that in this case H increases too quickly over the first few feet and maintains this discrepancy throughout, and it was partly as an attempt to remedy this that K was made dependent on the pressure gradient parameter Γ (equations (5.0.1,2)). In choosing the present model for $K(\Gamma)$ a compromise had to be reached between letting (in experiment 'E') the flow tend toward separation at $x = 16'$ and allowing H to maintain the low values indicated by experiment for $x > 12'$. The discrepancy still apparent may be attributable to a three-dimensional effect which occurred for $x > 12'$. Although the agreement between predicted H for experiment 'E' and experiment is not entirely satisfactory the present model for $K(\Gamma)$ was retained on the basis of experiment 'D' which Thompson indicates is precisely two-dimensional and which the present method predicts very well even close to separation.

To conclude the discussion on the comparisons of the present theory with the experiments of Schubauer and Spangenberg we make the following points

1) the predicted skin friction values exhibit, at the beginning of the calculation, a certain amount of scatter which, as can be seen from figure (5.1.6) (where the points represent the calculated values) are soon smoothed out and the c_f curves given have been drawn through the mean values

2) all the experimental information shown in figures (5.1.1-9) has been taken from reference [2]

3) the fact that the R_θ curves predicted by the present method agree more closely with the theoretical values calculated by Thompson (see Section 2.1) than with the experimental points would reinforce the conclusions made by Thompson concerning

5.1) contd.

3) contd.

the three-dimensional natures of the present flows (see figures (5.1.1,7)).

5.2) Bradshaw and Ferriss.

Bradshaw and Ferriss [27] investigated the effect of the sudden removal of pressure gradient on an equilibrium boundary layer. The experiment was devised both as a severe test for boundary layer calculation methods and also to obtain detailed turbulence measurements on which to base future methods.

The equilibrium boundary layer investigated by Bradshaw and Ferriss was one which maintained a pressure distribution corresponding to $U \propto x^{-0.255}$ (the experiment was denoted by 'a = -0.255') and another experiment (denoted 'a = -0.255 →0') investigated the effect of the transformation of the boundary layer from this equilibrium flow in an adverse pressure gradient to eventual equilibrium in zero pressure gradient. This latter boundary layer was considered to be a particularly severe test of any calculation method since the flow was dominated by the advection of turbulent kinetic energy from upstream, so that the turbulent energy is unlikely to be dependent upon local conditions only.

The predictions of the present method and comparisons with experiment are shown in figures (5.2.1-7). Head's method was not used to calculate this experiment because of the large initial H values involved. The δ^*, θ predictions are quite good in the equilibrium boundary layer, both increasing linearly from their initial values, although despite this the shape factor H is in error and is predicted to have an equilibrium value of 1.71 compared with the experimental value of about 1.54. The experimental

5.2) contd.

skin friction values in figure (5.2.4) were measured directly. In the equilibrium boundary layer c_f is considerably underestimated by the present calculation. The predictions shown in figures (5.2.1-4) for the experiment 'a = -0.255 → 0' are good particularly when it is remembered that it is this case which Bradshaw and Ferriss considered to be the severe test for calculation methods.

The experimental and predicted velocity profiles for both experiments are plotted in figures (5.2.5,6). In both cases a slight discrepancy introduced in the input velocity profile, as compared with the experimental profile, at the outer edge of the boundary layer is progressively removed as the calculation proceeds, whereas error is being introduced near the wall. The point of inflexion in the velocity profiles is predicted quite well, although Bradshaw and Ferriss point out that it is not reproduced by Thompson's velocity profile family.

All the experimental velocity profiles measured by Bradshaw and Ferriss, correspond to the logarithmic law (equation (3.2.6)) with A = 5.85 whereas the present calculation was performed with A = 4.9. It was anticipated that this difference between theory and experiment might help account for the observed discrepancies in the calculated velocity profiles but a number of computer runs made with this amended value for A failed to produce any significant differences from the original calculation.

Bradshaw and Ferriss, as a means of emphasising the poor performance of a number of calculation methods, compared $\theta\frac{dH}{dx}$ as predicted by the various methods with experimental values. The methods of Head, von Doenhoff and Tetervin, Spence and Maskell were considered. Figure (5.2.7) shows the predictions of Head's

5.2) contd.

method as reported in reference [27] and that of the present calculation. All the other methods listed above gave predictions for $-\theta \frac{dH}{dx}$ less than that given by Head and on this comparison the present method fares very well.

In addition we may add that, in reference to the two-dimensional nature of the flow, Bradshaw and Ferriss noted that for the equilibrium flow the 'tunnel provides as good an approximation to the two-dimensional flow as one can expect in a tunnel of reasonable width' whereas with the flow 'a = -0.255 → 0' they noted that 'after the removal of the pressure gradient the boundary layer started to diverge'.

5.3) Schubauer and Klebanoff.

Schubauer and Klebanoff [28] investigated the turbulent boundary layer developing over a simulated aerofoil with curved (convex) surfaces between x = 0 and x = 7' (radius of curvature 23') and between x = 18' and x = 28' (31' radius). Detailed measurements of velocity profiles together with turbulent shearing stress profiles were made up to separation. Reported values for the surface shearing stress, obtained by extrapolating the turbulent shearing stress profiles to the wall, need in the opinion of Coles [15] be reduced by 31% because of the excessively large values obtained. In view of this experimental skin friction values will not be used here for the purpose of comparison.

Thompson [2] considered that the flow was closely two-dimensional only in the initial region of favourable pressure gradient, and in the region of rising pressure (x > 20') the flow was said to be 'less accurately two-dimensional as separation is approached'. The discrepancies between the two-dimensional theory

5.3) contd.

and experiment encountered by Thompson may be due in part to either the convergence of the flow or the downstream curved surface.

The predictions of the present method for this experiment are shown in figures (5.3.1-4). The experimental c_f values shown in figure (5.3.3) have been computed from the logarithmic law of the wall using the experimental velocity profiles. No noticeable discrepancies occur in the predicted momentum thickness until x = 24' while H is evidentialy in error at x = 20'. There is considerable scatter in the experimental skin friction values but it appears that the calculated value deviates from the experimental as early as x = 14'.

Also shown in figures (5.3.1-4) are the curves obtained by treating the flow downstream of x = 14' as a plane of symmetry flow in which flow convergence has been introduced into the motion. The degree of convergence that has been imposed is simply that necessary to account for the observed discrepancies and in particular to induce separation at the required point. The convergence, as effected by a cross-flow velocity gradient on the plane of symmetry given by

$$\frac{\partial V}{\partial y} = - \frac{1}{3}\left(\frac{x-14}{5}\right)^5 \quad (/\,\text{sec}) \qquad (5.3.1)$$

was found to be sufficient to ensure reasonable overall agreement with experiment although θ is now somewhat greater than that indicated by experiment. The computer program was used so that it assumed on the axis of symmetry (y = 0)

$$\lim_{y \to 0} \frac{v}{V} = f\left(\frac{z}{\delta}\right) \qquad (5.3.2)$$

where the function f (which was taken from one of the three-

5.3) contd.

dimensional calculations to be considered in Chapter Six, and which was found to be of the same general shape for boundary layers developing in adverse pressure gradients) is plotted and tabulated in Appendix A7.

Velocity profile comparisons are shown in figure (5.3.4). Predictions from the two-dimensional calculations are good even up to x = 20' but are in considerable error by x = 24'. The velocity profiles as given by the axially symmetric calculation are in reasonable agreement with experiment overall.

The predictions for this experiment based on the entrainment method are in very close agreement with the predictions of the present method as given by the two-dimensional calculation.

5.4) Spangenberg, Rowland and Mease.

In an investigation into near separating flows Spangenberg, Rowland and Mease [29] made detailed mean velocity and turbulence measurements in two boundary layers (denoted 'A' and 'B') in both of which the skin friction was maintained at small values over prolonged distances of a smooth flat wall. Experiment 'B' was closer to separation i.e. smaller skin friction values (as given by the logarithmic law of the wall from the mean velocity profiles) were maintained, than experiment 'A'. The maximum pressure gradient that could be produced in the duct was introduced from x = 0 and then reduced so as to just prevent separation, although small areas of transitory stall still intermittent occurred along a large portion of the test surface length.

Of the two experiments only experiment 'A' is simulated here. The experimental velocity distribution of experiment 'A' was

5.4) contd.

empirically fitted by Spangenberg, Rowland and Mease to

$$U = 94.92\ (x+0.83)^{-0.33}\ (\text{ft/sec}) \qquad (5.4.1)$$

No attempt has been made to simulate experiment 'B' because as noted by the experimenters 'the differences between the two pressure distributions were of the same order as the reading errors and no consistent change was indicated'. Because of this the experimental data from both experiments has been plotted for comparison with ~~experiment~~ calculation.

These predictions for experiment 'A' from both the present calculation and Head's method are shown in figures (5.4.1-4). The pertinent fact concerning the present simulation is the large difference between the predictions for the two calculations. The prediction from the entrainment approach remains close to the experimental points for $x < 80"$ but then fails to recover and separation is predicted at $x = 110"$. The present method tends overall to remain closer to the points corresponding to experiment 'A' than experiment 'B' (although the scatter of points in figure (5.4.3) does leave this matter in some doubt as far as H is concerned) and the flow is not predicted to separate until 190".

The present method then performs considerably better than Head's in this comparison, although as one might expect and as indeed was intimated by Spangenberg, Rowland and Mease the classical boundary layer approximations are not entirely valid in near separating flows such as that considered here. In addition it must be pointed out that a characteristic of this type of flow is the presence of random cross-stream currents within the boundary

5.4) contd.

layer although 'checks on either side of the duct centre line showed essentially the same mean flow conditions across the duct'.

5.5) Conclusions.

The present chapter contains a number of comparisons of the present theory with experiment and enables the adequacy of these predictions to be judged in the light of predictions as provided by a well-tried and, in the context of calculation methods at large, accurate calculation method, namely that due to Head. The predictions of the present theory are moderately good and are generally at least as good as those given by Head. The present theory has also provided realistic predictions of two experiments which both provide quite severe tests for any calculation method viz Bradshaw and Ferriss's experiment in which the pressure gradient was suddenly removed from an equilibrium boundary layer and the near separating flow of Spangenberg, Rowland and Mease [27,29].

The main drawback inherent in attempting to determine the adequacy of the calculations presented in this chapter is the lack of any experimental quantitative information concerning the two-dimensional character of the flow. It would be extremely useful if in two-dimensional experimental investigations adequate consideration were given to this point.

C H A P T E R SIX

THREE-DIMENSIONAL CALCULATIONS

6.0) Introduction.

We are now in a position to be able to extend the predictions considered in Chapter Five to include both pseudo-three-dimensional boundary layers and three-dimensional boundary layers proper. Sections 6.1-4 are concerned with the pseudo-three-dimensional cases i.e. cross-flows exist within the boundary layers although the flows are dependent on only two space variables. Sections 6.2-4 provide comparisons of the present theory with experiment while sections 6.1-3 also consider alternative methods of calculation. Section 6.5 is concerned with an axially symmetric laminar stagnation flow which is included as a check on the three-dimensional calculation method, while the next two sections consider the predictions of two three-dimensional boundary layers both concerning the secondary flow induced upstream of a circular cylinder mounted perpendicularly on a flat surface. The latter of these two experiments (section 6.7) was intensively investigated and so enables some detailed comparisons between theory and experiment to be made.

It will be recalled from Chapter Four that the extension of the two-dimensional calculation method to three dimensions entails two further assumptions concerning the flow. These are firstly, the existence of a planar velocity profile in the inner part of logarithmic region at the wall, as exemplified by Johnston's triangular model for the flow (section 3.5), and secondly that the shear stress acts in the same direction as the maximum rate of strain i.e.

$$\nu_{ex} = \nu_{ey}$$

(section 3.3). Further the definition of the effective viscosity parameter K as defined as a function of Γ in equation (5.0.2) will be retained, but Γ will necessarily be amended as follows

6.0) contd.

$$\Gamma = 10^4 \frac{\theta_{11}}{Q} \frac{\partial Q}{\partial s} \qquad (6.0.1)$$

Equation (6.0.1) is thus a generalisation of equation (5.0.1) and reduces to it for two-dimensional flow. It was also found necessary however to impose a lower limit on K (0.007) in equation (5.0.2) since values of Γ encountered in the three-dimensional calculations were low enough to provide a negative K from equation (5.0.2). The complete function $K(\Gamma)$ is plotted in Appendix A7.

As in Chapter Five the section headings in this chapter will refer to the names of the original experimenters or, in the case of sections 6.1,5, to the original investigator of the particular theory considered.

6.1) Cumpsty and Head (1967).

Cumpsty and Head [30] in an application of their theory for calculating pseudo-three-dimensional boundary layers (section 2.1) considered the hypothetical case of an infinite swept wing for which they predicted boundary layer developments for a number of wing and flow configurations.

The swept wing was assumed to have, over the forward part of the chord, a region of constant freestream velocity (equal to that in the undisturbed flow) followed by a region in which the chordwise velocity decreased linearly while the spanwise velocity remained constant. Measuring x, normal to the leading edge, from the beginning of the region of adverse pressure gradient, the velocity components are given for $x > 0$ by

$$\left.\begin{aligned} U &= Q_o \cos\alpha_o (1-\kappa x) \\ V &= Q_o \sin\alpha_o \end{aligned}\right\} \qquad (6.1.1)$$

where Q_o is the undisturbed freestream velocity, α_o is the angle of sweep and κ the velocity gradient. The wing being assumed infinite

6.1) contd.

the flow is independent of spanwise position (y).

The cases treated by Cumpsty and Head were those listed in Table 6.1.1 all of which were calculated with the following initial conditions at x = 0

$$\theta_{11} = 0.00234'$$
$$H = 1.41 \qquad (6.1.2)$$
$$R_{\theta_{11}} = 2690$$

It was decided to simulate the same cases with the present theory to provide a check on the feasibility of the predictions provided by the solution scheme for a pseudo-three-dimensional flow.

As with Cumpsty and Head, the case

$$\alpha_0 = 35^\circ , \quad \kappa = 0.267$$

was used as an initial test. The predictions from both calculation methods for this flow are shown in figures (6.1.1-4), where θ_{11}, H, c_{fx} and β_0 developments have been plotted (c_{fx} is the component of the resultant skin friction in the chordwise (i.e. x) direction and β_0 is the angle between the freestream velocity and the limiting flow direction at the wall.) The two predictions agree reasonably well although the present calculation proceeds a little more slowly towards separation and consequently predicts a slower growth for H and β_0. A typical velocity profile from the present calculation (x = 1.2') is shown in figure (6.1.5) where it has been plotted in terms of streamwise (u_1) and crossflow (v_1) velocity components. The crossflow component of velocity exhibits the typical shape for such a profile and also the polar velocity plot (v_1 plotted as a function of u_1) has its expected triangular shape. The dashed portion of this latter profile

6.1) contd.

indicates the region within which the calculation method assumes a coplanar flow i.e. corresponds to the region between the log-point and the wall.

Those cases listed in Table 6.1.1 where the velocity gradient (parameter κ) is varied while the sweep α_o remains constant are treated in figures (6.1.6-9). Again the present calculation predicts a later separation than the predictions of Cumpsty and Head, the discrepancy between the two calculations increasing as the velocity gradient parameter is decreased. The momentum thickness predictions are in close agreement as before, ~~and~~ while the shape factor development as given by the present calculation is considerably lower than that calculated by Cumpsty and Head even allowing for the later separation in the present method. Separation however occurs at approximately the same value of H owing to the very rapid increase in $\frac{dH}{dx}$ in the present method as separation is approached.

It must be appreciated that the process of separation encountered in all the cases treated so far in this section is radically different from that observed in two-dimensional separation. Separation is caused essentially by the curvature of the streamlines. ~~because~~ The paths followed by particles of the fluid near the wall are deflected towards the spanwise direction so that all such particles at different spanwise positions are being deflected towards a common streamline, Separation must thus occur along this line. Shape factors encountered at separation in such an instance, as is borne out by the present calculations, are thus less than those met with in two-dimensional separation since the streamwise component of skin friction c_{f_1} does not in the former case necessarily tend to zero.

Considering now the effects of varying the angle of sweep of the wing while the velocity gradient remains constant, we

6.1) contd.

see that a marked difference in predicted behaviour exists between the two calculations as is shown in figures (6.1.10-13). With increasing sweep the entrainment calculation predicts that the separation point will first move downstream and then upstream again (as α_o is increased). The present theory on the other hand predicts a slight upstream movement of the point of separation as α_o increases for smaller angles of sweep, and then a more marked downstream movement for larger angles. The present theory moreover shows a more complicated behaviour pattern as can be seen from θ_{11}, c_{fx} predictions given in figures (6.1.10,12). Both predictions show momentum thicknesses to be largely independent of sweep angle, very little variation occurring for $x < 1.0'$, while the development of shape factor is markedly different for the four sweep angles considered, the present theory indicating a slower H growth as before. The crossflow angles predicted by the two calculations agree only in the magnitude of the angles to be expected while the pattern of behaviour encountered as α_o is increased differs considerably (see figure (6.1.13)).

As can be seen from equation (6.1.2) the boundary layers as calculated by Cumpsty and Head were dependent on $R_{\theta_{11}}$, θ_{11}, H only for initial conditions. It was thus thought necessary in the present investigation to determine if the same were true here and consequently the case $\alpha_o = 0°$, $\kappa = 0.25$ was re-run with ν, U, V all doubled. At $x = 1.3'$ H was found to have varied by 0.0005, c_{fx} by 10^{-6} and θ by $^1/_4\%$.

While considering the problem of the infinite swept wing it was thought worthwhile to further test the capabilities of the present method of calculation by considering an additional hypothetical flow, namely that in which a 'cross over' profile exists.

6.1) contd.

Such a profile may occur when the curvature of the external streamline changes sign so that the external flow induces in the boundary layer a crossflow contrary to that initially present. The result is, when the change in curvature is rapid enough, that within the boundary layer two separate crossflows, acting in opposite directions, exist. The newly imposed crossflow is introduced at the wall and gradually extends its influence outward until the original crossflow is entirely removed. It was found that with the present calculation scheme such a situation arose when in equations (6.1.1) with $\alpha_o = 35°$ the velocity gradient parameter κ was made dependent on x as follows

$$\kappa = 1 - x.$$

The external streamline for this flow then possesses a point of inflexion at $x = 0.5'$. With the same initial conditions as used previously (i.e. as given by equation (6.1.2)) this flow was calculated as far as $x = 1.2'$; the predictions are plotted in figure (6.1.14). The usual parameters θ_{11}, H, c_{fx}, β_o have been plotted for completeness although for the present purpose only the crossflow angle β_o is relevant. The curvature of the streamline upstream of the point of inflexion decreases progressively so that the crossflow angle β_o begins to decrease well before it is actually reached, and would be expected to tend assymptotically to zero if no further curvature were introduced. However since the curvature of the external flow changes sign β_o is to be expected to also change sign and, as can be seen from figure (6.1.14), β_o passes through zero at $x = 0.83'$ and begins to increase in magnitude again as the flow continues downstream.

If such a flow were calculated using the usual assumptions implicit in the entrainment calculations for three-

6.1) contd.

dimensional boundary layers (section 2.1) it would be necessary to assume that the flow in fact was coplanar at the point where $\beta_o = 0$ and that the reversed crossflow is introduced only after the original crossflow is removed. An examination of the velocity profiles obtained in the present calculation would indicate that this is far from the truth.

In figure (6.1.15) streamwise and crossflow velocity profiles are plotted for x = 0.6', 0.9', 1.2' and it can be seen that at x = 0.9' a definite 'crossover' crossflow profile exists. It has often been postulated that the polar plot of such a 'crossover' profile would be expected to exhibit a double triangle (i.e. approximate to three straight lines) but the prediction of the present calculation indicates that this is in fact not so (see figure (6.1.15)). It can also be seen that even after the crossover profile has been removed there is some delay before the outside edge of the Johnston's triangle is reinstated as a straight line. A more detailed selection of polar plots is given in figure (6.1.16) where the profiles at x = 0.75'(0.05')1 have been plotted. One minor point to be noticed from this figure is that the assumption at the wall of a coplanar velocity profile might now be called into doubt for the purpose of the present calculation (see the profile at x = 0.75'). However this assumption seems generally to have coped with the situation quite well and even so it would be possible to remove this assumption and replace it by a more general one for such a situation as this.

Figure (6.1.17) shows the limiting streamline at the wall and in the freestream for this latter flow. The point of inflexion of the external streamline is shown and also the apparent is point of zero crossflow angle β_o (where the two streamlines are parallel).

The present theory has thus given feasible results

6.1) contd.

for the boundary layer flow over an infinite swept wing (for which predictions with experiment are given in the next two sections) and also provided a very convincing explanation of the behaviour of the flow within the boundary layer in which there is a severe point of inflexion in the external streamline.

6.2) Cumpsty and Head (1970).

As a means of estimating the effectiveness of the boundary layer calculations performed by Cumpsty and Head [30] and discussed in the last section, an attempt was made to simulate experimentally an infinite swept wing. Cumpsty and Head [8] consequently measured the flow over a wing of 18" chord swept at 61°, in a wind tunnel of 48" working section width. Mean velocity profiles were measured downstream of the line of minimum pressure up to the separation line at two spanwise positions.

The predictions for this experiment based on the present theory are shown in figures (6.2.1-3) where some predictions of Cumpsty and Head have also been included (the experimental results included in these figures (circles) are those obtained with the 'slender traverse gear'). Figure (6.2.1) shows the predictions of both methods when infinite swept wing theory is used, not only to infer spanwise independence of the flow, but also to determine the direction of the mainstream flow since only the magnitude of the velocity of the flow has been recorded experimentally. (The 'downstream' pressure distribution has been used in both calculations). For both calculations the assumption that the V component of velocity is constant over the chord and equal to that at the leading edge ($V_{\ell e}$), leads to an inadequate explanation of the flow since the

6.2) contd.

rates of change of all parameters plotted θ_{11}, H, c_{fx}, β_o are underestimated, the present calculation giving the poorer of the two predictions.

In an attempt to reduce the discrepancy between the experimental results and their calculations Cumpsty and Head imposed an additional spanwise velocity (a 5% increase in V was introduced, the pressure distribution remaining the same) on the flow in order to reconcile the observed and predicted momentum thickness (θ_{11}) development. This assumption produced a slight improvement in shape factor development and considerably improved that of β_o as is shown in figure (6.2.2). The present theory, in conjunction with the same assumption $V = 1.05V_{\ell e}$, also predicts an improved θ_{11} development, although the changes in H, β_o are here only slight. There is however some movement towards the separation of the flow.

Figure (6.2.3) shows the various displacement and momentum thicknesses which are dependent on the crossflow as determined by experiment and also as calculated by the present theory. The various thicknesses shown have been predicted better than the β_o comparisons shown previously would indicate, as is also the improvement in prediction achieved by increasing the crossflow from $V = V_{\ell e}$ (curve 1) to $V = 1.05\ V_{\ell e}$ (curve 2).

The assumption that the flow over the experimental arrangement is equivalent to the theoretical infinite swept wing is obviously suspect. Apart from the problem of how to determine the direction of the flow outside the boundary layer when only the resultant velocity there is known, we need obviously consider the possibility of a spanwise dependence of the flow. Cumpsty and Head measured the pressure distributions across the chord at two spanwise

6.2) contd.

positions (denoted 'upstream' and 'downstream') and the difference in the static pressure coefficient c_p was found to be as much as 0.04 at the point of minimum pressure and 0.08 at the trailing edge. It was thought more realistic within the present calculation scheme to introduce this observed spanwise pressure gradient into the calculation rather than proceeding to investigate further the effects of different assumptions for V in the pseudo-three-dimensional context.

Two attempts have been made to reconcile theory and experiment by introducing the difference in static pressure at the two spanwise positions. Both calculations were computed using a three-dimensional mesh and imposed on the flows considered above spanwise gradients in U,V respectively sufficient to account for the encountered pressure discrepancies. A spanwise velocity gradient in U had a slight effect, largely at the trailing edge, but in the reverse direction to that required. A similar gradient in V produced no significant change in predicted developments.

We conclude therefore that theory and experiment can largely be reconciled by an overall increase in spanwise velocity, while the experimentally observed spanwise pressure gradient could not readily be used to explain the still apparent discrepancy in shape factor development. This latter discrepancy might still be attributable to some variation in the direction of the flow in the mainstream not already considered although it would appear that the fault more likely lies with the calculation schemes considered.

6.3) P.D.Smith.

The experiments of P.D.Smith [7] were, as was that considered in the last section, concerned with a simulated

6.3) contd.

infinite swept wing. The measurements (mean velocity profiles) were made on the lower surface of a flat plate below which there was fixed a porous circular cylinder fitted with a Thwaites flap. Boundary layer suction was applied to the circular cylinder to prevent separation on the cylinder. The different experimental configurations were obtained by varying the inclination of the Thwaites flap and the distance between the plate and cylinder; all experiments were performed with both the plate and cylinder inclined at $26\frac{1}{2}^{\circ}$ to the mainstream flow.

Only three of the nine runs investigated by Smith are considered here, these are 'runs 1,5,6', the experimental results for which are shown in figures (6.3.1-3). Also included in these figures is one set of predictions calculated by Smith. This prediction, shown by the dashed line ('method 3' as denoted by Smith) is based essentially on the entrainment approach and assumes a power profile for the streamwise velocity and Mager's relation (equation (2.1.7)) for the crossflow. 'Method 3' gave the best results overall of the six calculation methods (all integral methods) considered by Smith. The solid lines in figures (6.3.1-3) are the predictions for the flow based on the present theory.

Both predictions indicate slightly exaggerated momentum thickness (θ_{11}) growths, the two calculations giving very similar results. Shape factor predictions based on the present method are poor and only in the severest flow ('run 6') is H predicted at all well and even then the calculation tends to separation at the trailing edge, a feature not exhibited by the experiment. Crossflow angle predictions from both theories are reasonable.

6.3) contd.

The tendency for the present calculation to over-estimate shape factor development is the reverse of that noted in the swept wing considered in section 6.2, although as there the effectiveness of the calculation scheme is obviously confounded with any extraneous three-dimensional effects within the experiment. Since in section 6.2 no such effect could be shown to explain the discrepancy in H predictions we are inevitably led to expect the same here. Indeed the sweep of the wing in the present case is not as severe as that considered previously and so presumably the flow is more reliably pseudo-three-dimensional.

The mainstream velocity distribution used in the present calculation was, as in section 6.2, based on the use of infinite swept wing theory to determine the direction of the flow. No data has been published in the present case to establish the pseudo-three-dimensional nature of the flow.

6.4) Hoadley.

The experiment of Hoadley [31] was concerned with the flow in a diffuser in which swirl had been introduced into the motion. The dimensions of the experimental arrangement are shown in figure (6.4.1); the mean velocity profiles were measured at the axial positions shown.

The present theory predicts this flow quite well as is shown in figures (6.4.2-5). The curves in these graphs were computed using the velocity distribution measured by Hoadley. Both the magnitude Q and the deviation from the axial direction α_o of the mainstream velocity field were measured; the values are shown tabulated in Table 6.4.1. It can be seen that the α_o values possess

6.4) contd.

a great deal of scatter and since it is the rate of change of this quantity that will determine the development of the crossflow within the boundary layer the possibility of being able to predict this aspect of the flow does not seem very encouraging. Resolving Q into its components U,V we see these values appear more reasonably distributed and it was from a linear interpolation of these values that the prediction was calculated.

The main discrepancy between theory and experiment is the marked increase in rates of change of the boundary layer parameters as separation is approached which results in separation being predicted before it is achieved experimentally. The scatter in the data could obviously have been the reason for the excessive β_o predictions which in previous calculations has been predicted somewhat more accurately.

The results obtained here are obviously very encouraging since the likelihood of extraneous three-dimensional effects in the present experimental arrangement are much less than those encountered in the infinite swept wing simulation.

6.5). Froessling.

As a check on the finite difference approximations to the three-dimensional boundary layer equations (1.1.10-12) it was decided to simulate a laminar boundary layer for which a known solution existed. That chosen was the axisymmetric stagnation flow against a flat surface [1] which, since the calculation was performed over a rectangular grid, was three-dimensional as far as the present calculation scheme was concerned.

6.5) contd.

N.Froessling [32] solved the complete Navier-Stokes equations for this flow but, because the terms which are deleted in the course of the boundary layer approximations cancel of their own accord from the Navier-Stokes equations, the problem can equally be treated as a boundary layer flow. In the latter case if the mainstream potential distribution is assumed to be given by

$$U = ax \qquad V = ay$$

it is possible to write the velocity components within the boundary layer as

$$u = ax\phi'$$

$$v = ay\phi'$$

$$w = -2\sqrt{a\nu}\ \phi$$

where ϕ, a function of ζ

$$\zeta = \sqrt{\frac{a}{\nu}}\ z,$$

must satisfy

$$\phi'^2 - 2\phi\phi'' = 1 + \phi'''$$

with the boundary conditions

$$\zeta = 0 \quad : \quad \phi = \phi' = 0$$

$$\zeta = \infty \quad : \quad \phi' = 1$$

Froessling has tabulated the functions ϕ, ϕ'. Although, as mentioned above, the flow is three-dimensional as far as the present calculation is concerned it is obvious from the above considerations that boundary layer is both coplanar and of constant thickness.

The boundary layer was solved by starting the calculation at a point slightly away from the stagnation point and by inputing a sine profile as a first approximation to the streamwise velocity profile. The calculation was then continued

6.5) contd.

downstream until the ϕ, ϕ' profiles had settled down to the fifth signficant figure and the solutions at individual faces were convergent to a tolerance of 10^{-12}. The ϕ, ϕ' profiles obtained are plotted in figure (6.5.1) and tabulated in table 6.5.1.

The difference between the ϕ computed here and that calculated by Froessling is as little as $^1/_{20}\%$ at $\zeta = 3.6$ where it must be remembered in the present calculation ϕ' is calculated directly and then integrated to obtain ϕ, so that any error in ϕ at the edge of the boundary layer is the accumulation of those arising in ϕ' in the boundary layer. The present comparison thus verifies the adequacy of the finite difference scheme used in the present calculations.

6.6) Hornung and Joubert.

Hornung and Joubert [13] investiaged experimentally the secondary flow within the boundary layer upstream of a cylinder mounted perpendicularly on a flat plate. This experiment, as is that to be considered in the next section, is one of the few instances in which detailed mean velocity profile measurements have been made in a three-dimensional turbulent boundary layer.

The flat plate on which the boundary layer was developing, was 20' long and approximately 5' wide at the position of the model. The model was of semi-circular section (22" diameter) on the upstream side and faired at the rear. It was mounted 17' from the leading edge of the plate.

The effect of the model was to induce a region of recirculation upstream and it was over the region prior to separation that the mean velocity and yaw profiles were measured. The precise positions were the experimental measurements were made ~~is~~ are shown in figure (6.6.1).

6.6) contd.

The present calculation of this flow has been performed over the mesh shown in figure (6.6.1) over which the mainstream flow was assumed to be approximated to by the potential flow about a near circular cylinder between parallel walls as given by Kennard [33]. The velocity components for this flow are given in Appendix A9. As can be seen in figure (6.6.1) the solution was started at x = -4' where planar velocity profiles were input, the same profile being used at all sections across this face. The solution was matched on the axis of symmetry at x = -2.125' to the experimental data; both the initial conditions and potential distribution were amended to ensure agreement at this matching point. In addition, to effecting a solution using the present scheme, it was necessary to impose as boundary conditions v distributions outside the boundary planes y = 0', -2'. The symmetry condition was used at y = 0' and at y = -2' two alternative boundary conditions were considered, viz.

$$\frac{v}{V} = f(z) \qquad (6.6.1)$$

$$\frac{v}{V} = (1+\alpha y)\, f(z) \qquad (6.6.2)$$

These conditions allowed the v prfiles outside the solution space to be calculated iteratively from the solution within the space at the same x, i.e. α and the function f were obtained by applying either of the two above boundary conditions as a condition of continuity on the $\frac{v}{V}$ profiles. While these alternative conditions gave virtually the same results when used in the calculation scheme the latter was found to give a more favourable explanation of the behaviour of v across the boundary y = -2', and was retained for the purpose of the present calculation.

6.6) contd.

The calculation was performed over nine sections at each face, while the configuration of points at each section was the same as that detailed in Appendix A8; the program computed 18 solution faces in 8.3 mins on the IBM S360/65 before reaching separation. Once separation is reached at any section on a solution face the calculation method breaks down for all sections on that face.

Detailed comparisons between theory and experiment are given in figures (6.6.2-9) where boundary layer parameter (θ_{11},H,c_f,β_o) comparisons are made at sectional planes y = 0', -0.5', -1' and x = -2.125', -1.75' and velocity profile comparisons at all positions where the above two sets of planes intersect. The disposition of points and planes where comparisons have been made are also shown in figure (6.6.1).

Before proceeding to discuss these predictions we will repeat that the solution was started at x = -4' with uniform initial conditons with respect to y and zero cross flow. The mainstream distribution used was the potential distribution about a near circular cylinder between parallel walls (Appendix A9) and the boundary conditions used were the symmetry condition at y = 0' and equation (6.6.2) at y = -2'. The solution was matched so as to agree with experiment at x = -2.125', y = 0'.

Figure (6.6.2) would suggest that the present theory has indicated separation at approximately the correct point although θ_{11}, H are developing a little too quickly along the axis of symmetry. The correct pattern of behaviour has been predicted as y increases in magnitude (figures (6.6.2-4)). Predictions for θ_{11}, H show development being retarded as we move away from the axis of symmetry and also the commencement of a region of increase in skin

6.6) contd.

friction with increasing x at y = -1', both of which are in agreement with experiment. Crossflow angles are also being predicted competently although at y = -1' β_o is underestimated by some 3%.

Considering comparisons at x = constant planes figures (6.6.5,6) we can see that the shape of all the profiles has been predicted quite well. The main shortcoming again is in the underestimation of the crossflow angle β_o. It is relevant to note for the purpose of these comparisons that the increment g for the y direction was 0.25'. Both calculation and experiment indicate with y increasing both θ_{11}, H tending to a constant value at each plane. The slight difference between theory and experiment here might obviously be accounted for by the presence of some additional effects caused by the presence of the wind tunnel walls. In this respect it might also be noted that the effect of the region of recirculation upstream of the cylinder appears to have had little effect upon the mainstream velocity distribution in this region.

Figures (6.6.7,8) show the experimental and computed mainstream and crossflow velocity distributions for y =-0',-0.5',-1' at both x = -2.125' and x = -1.75' respectively. The run numbers included are those assigned by Hornung and Joubert. Streamwise velocities are predicted quite well while the crossflow profiles are underestimated at y = -1' as would be expected from the observation made above concerning β_o. A slight assymmetry of the flow is also apparent from the experimental data.

Figure (6.6.9) shows comparisons between calculated and experimental polar plots. The plots at y = -0.5' are in error

6.6) contd.

only at the wall while the plots for y = -1' show an overall discrepancy. The two sections of the curves corresponding to the present calculation in figure (6.6.9) relate to the calculated velocity profile and the assumed planar profile between the log-point and the wall respectively.

It must be pointed out that more correctly the solution scheme demands that the initial condition be specified across the initial solution face and not just at one section on it as was done here. This point might very largely explain the discrepancies encountered in the β_0 predictions (see figures (6.6.4,5)). In view of this the present calculation has provided excellent agreement with experiment.

This experiment has also been simulated by Nash [12] who solved the turbulent energy equations (section 2.2). Nash started his calculations at x = -7' in a manner similar to that employed here and matched calculation to experiment at the same point. The crossflows as predicted by Nash were underestimated by the same order of magnitude as those in the present calculation.

6.7) East and Hoxey.

The experimental arrangement investigated by East and Hoxey [22] was very similar to that considered by Hornung and Joubert (section 6.6). The model used had a semicircular leading edge of 24" diameter and was mounted on a flat plate of 9'9" width. The experimental measurements were considerably more intensive than those of Hornung and Joubert as can be seen from figure (6.7.1). This enabled the experimental mainstream velocity distribution to be used as a basis for the calculation to be considered below and also meant that the calculation could realistically

6.7) contd.

be matched to the observed conditions across a x = constant plane.

Figure (6.7.1) shows the mesh over which the calculation was developed. Uniform mainstream velocity profiles were input at x = -40" and the solution was then continued to x = -30", the mainstream velocity distribution being basically that obtained by extrapolating from the experimental distribution for x > -30" but was modified slightly (together with the initial conditions) to produce the correct H, θ_{11} distribution at x = -30". Beyond x = -30" the solution was continued up to separation using the experimental velocity distribution.

Equation (6.6.2) and the symmetry condition were used as the respective boundary conditions as outlined in section 6.6. Ten sections were used at each face while the array of points used at each section was again the same as that used in section 6.6. The program computed 28 solution faces up to separation in 14.35 minutes on the IBM. S360/65.

The comparisons between theory and experiment shown in figures (6.7.2-14) are at the sectional planes indicated in figure (6.7.1) where the circles represent the experimental points. The usual boundary layer parameters θ_{11}, H, c_f, β_o are shown plotted in figures (6.7.2-8) at y = 0,3,6,9", x = -30, -26, -23", while figures (6.7.9-14) show predictions for the crossflow boundary layer thicknesses δ_2^*, θ_{12}, θ_{22}. For convenience when plotting β_o, δ_2^*, θ_{12} changes in sign across the plane of symmetry have been ignored and instead where the sign of the experimental quantity is the opposite of that shown a vertical line has been drawn through the circle at that point. Figures (6.7.6,12) show

6.7) contd.

both these sets of parameters plotted at the plane where theory was matched to experiment. With respect to θ_{11} a marked assymmetry of the flow is apparent and the calculation was necessarily matched to the average for both sides of the plane of symmetry. As is readily apparent the crossflow at this initialising plane is less than that indicated by experiment; an attempt was made to account for this by imposing a crossflow at x = -40" but this was found to produce a marked change in θ_{11} at x = -30" and it was not possible to readjust the pressure distribution for x < -30" to remedy this in the limited time available. However, as was the case in previous comparisons, the crossflow is being predicted somewhat more accurately than the crossflow angle β_o predictions would indicate.

The comparisons of θ_{11}, H, c_f, β_o at the y=constant planes show that the correct type of behaviour is being predicted. The movement towards the separation of the flow along y = 0 is not being predicted to occur as quickly as the experiment although at the other y = constant planes c_f predictions are closer to experiment. Crossflow angles β_o on the other hand are being quite seriously underestimated. The overall behaviour here is not much different from that found in the experiment of Hornung and Joubert.

Considering now the same parameters at sectional planes x = constant. Firstly we notice here a slightly different type of behaviour to that encountered in the last section, although in the present problem it must be

6.7) contd.

remembered that the area of the wall of interest is somewhat more restricted than that considered previously. The most apparent difference is that whereas before the momentum thickness θ_{11} attained a maximum on the plane of symmetry and then tended to a constant value as we moved away, here we observe that θ_{11} on the plane of symmetry attains a minimum and then a maximum at some short distance away. This would appear to be due to the relative remoteness of the wind tunnel walls in the present flow so presenting less resistence to the divergence of the flow on the plane of symmetry. The assymmetry of the flow at x = -23" is more marked than that further upstream as is the error in the crossflow angle β_o.

With respect to the boundary layer thicknesses δ_2^*, θ_{12}, θ_{22} plotted in figures (6.7.9-14) it would appear that the greater part of the discrepancies here can be traced to the incorrect initial condition at x = -30", and it can be seen that the correct type of behaviour is being predicted overall.

With respect to the data shown plotted in figures (6.7.2-8) the skin friction values at y = 3,9" and x = -23" have been obtained by averaging the experimental values at the planes one inch on either side of the plane concerned and all the experimental c_f values plotted are those obtained by Preston tube measurements.

There is also some need to point out here that

6.7) contd.

no attempt was made in the present calculation to ensure the irrotationality of the mainstream flow (section 1.1) for $x \leq -30"$. The result of this was that although the velocity profiles settled down assymptotically at the edge of the boundary layer, at the last one or two points a slight twist in the velocity vector $\underline{q}$ was apparent. This effect increased as x increased within the range $-40" < x < -30"$ to a maximum of 1° at $x = -30"$, $y = 9"$ and then decreased for $x > -30"$. Although the effect of this pecularity on the calculation scheme was probably very slight it was necessary to take it into account when analysing the crossflow velocity profiles and consequently for this purpose it was necessary to ignore a few outer points.

6.8) Conclusions.

Chapter Six contains comparisons between the present theory and both experiment and alternative theories for three-dimensional and pseudo-three-dimensional boundary layer flows.

With regard to the pseudo-three-dimensional flows the present theory although apparently able to predict momentum thickness θ_{11} and crossflow development with reasonable competence, was in error in calculating shape factor development. This deficiency could not be attributed to extraneous spanwise velocity gradients although it may still have been caused by a deviation of the flows from infinite swept wing theory (in the pseudo-three-dimensional context).

6.8) contd.

However since P.D.Smith's extensions to Head's entrainment calculation predict H quite well for this flow this is thought unlikely to be so. The other possibility is that the model employed for the effective viscosity in the present calculations is at fault. Improvements in this model, in the two-dimensional context, should be further investigated although the lack of proven reliable two-dimensional experiments makes this not altogether an easy matter. Crossflows have been predicted with reasonable accuracy thus confirming the plausibility of the three-dimensional effective viscosity model i.e.

$$\nu_{ex} = \nu_{ey} = \nu_e$$

The two three-dimensional turbulent boundary layer flows considered have provided good agreement between theory and experiment. It would appear that although there is a tendency in these calculations to underestimate the crossflow angle β_o this same tendency is not exhibited in the crossflow thickness δ_2^* predictions which are calculated reasonably well.

C O N C L U S I O N S.

The present work has been concerned with the extension of existing two-dimensional turbulent boundary layer calculation methods to three dimensions. It was decided to employ the effective viscosity approach within the present calculations basically because it provided the method which required the least empirical information both to establish the two-dimensional calculation and then to extend this to three dimensions. Although the mixing length concept is generally recognised as a plausible model for the flow away from both the wall and freestream it was necessary to assess thoroughly the capability of the model in the outer edge of the boundary layer. As a result a simple model of the flow wasdeveloped for the outer layer (see Appendix A7) from a two-dimensional retarding boundary layer and was consequently used as the basis of all the calculations considered here (in both two and three dimensions).

The two-dimensional calculations presented here have provided reasonable agreement with experiment and compared favourably with predictions for the same experiments as provided by Head's entrainment method. The pseudo-three-dimensional calculations have proved to provide the same measure of agreement with experiment although here the predictions for the infinite swept wing, presumably because of the inapplicability of the effective viscosity model to this type of flow, gave disappointing shape factor predictions. The two experimental three-dimensional turbulent boundary layers considered (both of the retarded flow type essentially) gave good agreement with experiment. The crossflow thickness δ_2^* was calculated quite well though the crossflow angle β_o was seriously underestimated. The present investigation nevertheless shows the feasibility of computing three-dimensional

flows with the aid of only a few simple assumptions for the extension of the two-dimensional computation scheme, and the present finite difference scheme provides a good framework on which to calculate the three-dimensional turbulent boundary layer.

The only other attempt made to calculate the turbulent three-dimensional problem to the knowledge of the present author, was by Nash who solved the turbulent energy equation and the few indications that there are in the literature point to the fact that both methods are predicting crossflows with the same accuracy.

The present calculation scheme employs a streamline type of transformation which allows the iterative scheme which has been developed to calculate the iterated u,v velocity profiles at each section (a line through the boundary layer perpendicular to the wall) independently of each other and independently of the same profiles at adjacent sections along the same marching plane. This has the advantage that the resulting set of linear algebraic equations that have to be solved, as well as being tri-diagonal, will have the same number of unknowns at each solution as the number of points at each section. The equations relating the unknown u,v velocity component profiles at any section moreover are the same (i.e. the equations relating the u's are the same as those relating the v's) the differences appearing only on the 'right hand sides'. Both profiles can thus be calculated simultaneously. The finite difference scheme used has been substantiated by comparison to a laminar boundary layer for which an analytic solution exists. The boundary condition at the wall in the turbulent case has been verified in that it provides accurate predictions for the coefficient of skin friction.

The included computer program provides an efficient computation scheme; three section iterations were computed per second on average on the IBM S360/65, seven iterations per step were required

on average once the calculation had settled down and while the calculation was not close to separation. A realistic three-dimensional boundary layer calculation takes of the order of ten minutes. In addition if a two-dimensional version of the present program were produced both the computer storage and running time would be considerably reduced for that problem (the established values for the solution weights might also be written implicitly into the program to the same effect). The program has been structured in such a way as to allow alternative effective viscosity models to be readily incorporated should this be desirable.

It would also facilitate easier use of the enclosed program, especially for three-dimensional calculations, if some means were incorporated into the program for setting up the velocity profiles at the commencement of a calculation. There is also an obvious need for having the effective viscosity model used here more broadly based and taking into account more varied boundary layer flows; it appears that it will be sufficient to do this on a purely two dimensional basis.

With regard to further developments to the present calculation scheme: it would be useful and comparatively simple if the introduction of body forces were facilitated to enable computations to be made for rotating systems and also if the surface curvature of the wall could be allowed for. The present investigation moreover, together with that of Nash, lend considerable support to the future development of the differential approach to three-dimensional turbulent boundary layer problems as opposed to the integral approach.

ACKNOWLEDGEMENTS

I would like to acknowledge the guidance of Professor R.Hetherington who supervised me during the course of the present work and the help received from the Mathematics Department of the University of Aston in Birmingham. I would also like to thank Rolls-Royce Limited, Derby for kindly allowing me to make use of their computing facility.

NOTATION.

Owing to the profusion of symbols used only those of general interest are listed below. Where symbols have been used for more than one application the notation below has been restricted to one particular chapter. Those symbols not listed have application to one section or appendix only and where this is so explanations concerning their use will be found in that section or appendix.

a	constant in grid transformation equation (4.3.1) (Chapter 4)
A	empirical constant in law of the wall equation (3.2.6)
b	constant in grid transformation equation (4.3.1)
c_f	magnitude of coefficient of skin friction $(c_{f_1}^2 + c_{f_2}^2)^{\frac{1}{2}}$
c_{f_1}, c_{f_2}	components of c_f in streamwise and crossflow directions respectively, equation (1.2.5)
c_{fx}	component of c_f in x direction
f,g,h	increments of grid associated with x,y,ξ directions (Chapter 4)
H	shape factor $\frac{\delta_1^*}{\theta_{11}}$
K	constant in effective viscosity function equation (3.3.5)
ℓ	Prandtl's mixing length, equation (3.1.1) (Chapter 3)
ℓ,m,n	counters on grid associated with x,y,ξ directions
n^*	counter on grid associated with log-point
M,N	number of sections on solution face, and points on a section
p	pressure
q	velocity component in the boundary layer parallel to the wall $(u^2+v^2)^{\frac{1}{2}}$
q_T	three-dimensional form of friction velocity u_T
Q	velocity component in the freestream parallel to the wall $(U^2+V^2)^{\frac{1}{2}}$ $(\equiv U_1)$

r	iteration counter
$R_{\theta_{11}}$	Reynolds number based on θ_{11}
s	streamwise distance, equation (4.3.5)
u,v,w	velocity components in the boundary layer associated with x,y,z directions
u_1,v_1 u_1,u_2	velocity components in the boundary layer associated with the streamwise and crossflow directions
u_τ	friction velocity, equation (3.2.2)
U,V	velocity components in the freestream associated with x,y directions
w	wake function, equation (3.4.3) (Chapter 3)
α,β,γ	streamline section coordinates, fig.(4.6.1) (Chapter 4)
β_o	angle between limiting streamline at the wall and external streamline
Γ	velocity gradient parameter, equation (6.0.1)
δ	'boundary layer thickness'
δ_i	z at which q = iQ
δ_1^*,δ_2^*	displacement thicknesses, equation (1.2.4)
ϵ	factor in wall condition, equation (4.4.5)
ζ	effective viscosity function parameter, equation (3.3.5) (also stagnation flow variable, section 6.5)
θ_{11},θ_{12} θ_{21},θ_{22}	momentum thicknesses, equation (1.2.3)
κ	constant in Prandtl's mixing length, equation (3.1.3), and logarithmic law of the wall equation (3.2.6)
λ	number of large increments subdivided at the wall
ν	kinematic laminar viscosity
ν_e	effective viscosity
ν_e'	transformed effective viscosity, equation (4.3.8)
ν_{ex},ν_{ey}	effective viscosity appearing in x,y momentum equations respectively
ξ	transformed z co-ordinate, equation (4.3.1)
ρ	density
τ	shear stress

τ_o	shear stress at the wall
τ_{o1}, τ_{o2}	components of shear stress at the wall in streamwise, crossflow directions respectively, equations (1.2.6-7)
ϕ	function in effective viscosity model, equation (3.3.5) (also function in stagnation flow, section 6.5)
ϕ_i	$= 1-\psi_i \quad i = 1,8$
ψ_i	finite difference solution weights, table 4.6.1
ω	number of subdivisions per large increment at the wall.

The fluctuating components of turbulent quantities have been denoted by dash and the time-averaged quantities by a bar (which generally for convenience has been omitted in connection with u,v,w).

REFERENCES.

1) Schlichting, H., Boundary Layer Theory. McGraw-Hill, New York, Fourth Edition, 1960.

2) Thompson, B.G.J., A Critical Review of Existing Methods of Calculating The Turbulent Boundary Layer, A.R.C. R. & M. 3447, 1967.

3) Head, M.R., Entrainment in the Turbulent Boundary Layer, A.R.C. R.& M. 3152, 1960.

4) Cooke, J.C., Three-Dimensional Turbulent Boundary Layers, A.R.C. C.P.635, 1963.

5) Cumpsty, N.A., A Critical Examination of the Use of a Two-Dimensional Turbulent Profile Family to Represent Three-Dimensional Boundary Layers, A.R.C. C.P.1068, 1968.

6) Cumpsty,N.A., Crossflow in Turbulent Boundary Layers, A.R.C. C.P.1067, 1968.

7) Smith, P.D., Calculation Methods for Three-Dimensional Turbulent Boundary Layers, A.R.C. R.& M. 3523, 1968.

8) Cumpsty, N.A., and Head, M.R., The Calculation of Three-Dimensional Turbulent Boundary Layers, Part IV.. Comparison with measurements on the rear of a swept wing, Aero Quart. 21, 2, 121-32, 1970.

9) Cham, T.S., and Head,M.R., Turbulent boundary layer flow on a rotating disc, J.Fluid Mech.37, 1, 129-147, 1969.

10) Spalding,D.B., and Patankar, S.V., Heat and Mass Transfer in Turbulent Boundary Layers, Morgan-Grampian, London. 1967.

11) Bradshaw, P., Ferriss, D.H., and Atwell,N.P., — Calculation of boundary layer development using turbulent energy equations, J.Fluid Mech. 28, 3, 593-616, 1967.

12) Nash, J.F., — The calculation of three-dimensional turbulent boundary layers in incompressible flow, J.Fluid Mech. 37, 4, 625-642, 1969.

13) Hornung, H.G., and Joubert,P.N., — The mean velocity profile in three-dimensiona turbulent boundary layers, J.Fluid Mech. 15, 3, 368-385, 1963.

14) — Proceedings, Computation of Turbulent Boundary layers, Volume 1, AFOSR-IFP-STANFORD CONFERENCE, 1968.

15) Coles, D., — The law of the wake in the turbulent boundary layer, J.Fluid Mech. 1, 2,191-226, 1956.

16) Mellor, G.L., and Gibson,D.M., — Equilibrium turbulent boundary layers, J.Fluid Mech. 24, 2, 225-253, 1966.

17) Clauser,F.H., — Turbulent boundary layers in adverse pressure gradients, J.Aero.Sci. 21, 91-108, 1954.

18) Clauser, F.H., — The Turbulent Boundary Layer, Advances in Applied Mechanics, Volume IV, 1-51, 1956.

19) Mellor, G.L., — The effects of pressure gradients on turbulent flow near a smooth wall, J.Fluid Mech. 24, 2, 255-274, 1966.

20) Maise, G., and McDonald, H., — Mixing Length and Kinematic Eddy Viscosity in a Compressible Boundary Layer, AIAA Journal 6, 1, 73-80, 1968.

21) Johnston,J.P., — On the Three-Dimensional Turbulent Boundary Layer Generated by Secondary Flow, Trans ASME, Series D, 82, 233-248, 1960.

22) East,L.F., and Hoxey,R.P., Low-speed three-dimensional turbulent boundary layer data, Part 1, R.A.E. Tech Report 69041, 1969.

23) Pierce,F.J., The Law of the Wake and Plane of Symmetry Flows in Three-Dimensional Boundary Layers, Trans ASME, Series D, 88, 101-108, 1966.

24) Joubert, P.N., Perry, A.E., and Brown, K.C., Critical Review and Current Developments in Three-Dimensional Turbulent Boundary Layers, Fluid Mechanics of Internal Flow, 210-237, Elsevier, Amsterdam, 1967.

25) Richtmyer, R.D., Difference Methods for Initial-Value Problems, Interscience, New York, 1957.

26) Schubauer,G.B., and Spangenberg,W.G., Forced mixing in boundary layers, J.Fluid Mech. 8, 1, 10-32, 1960.

27) Bradshaw, P., and Ferriss, D.H., The Response of a Retarded Equilibrium Turbulent Boundary Layer to the Sudden Removal of Pressure Gradient, N.P.L. Aero.Report, 1145, 1965.

28) Schubauer,G.B., and Klebanoff, P.S., Investigation of Separation of the Turbulent Boundary Layer, NACA Report 1030, 1951.

29) Spangenberg, W.G., Rowland, W.R., and Mease, N.E. Measurements in a Turbulent Boundary Layer Maintained in a Nearly Separating Condition, Fluid Mechanics of Internal Flow, 110-151, Elsevier, Amsterdam, 1967.

30) Cumpsty,N.A., and Head, M.R., Calculation of Three-Dimensional Turbulent Boundary Layers, Part I. Flow over the rear of an infinite swept wing, Aero.Quart. 18, 1, 55-84, 1967.

31) Hoadley, D., Ph.D. Thesis, Cambridge, 1969.

32) Froessling, N., 1940 see [1], p.92

33) Kennard, E.H., Irrotational flow of Frictionless Fluids, Mostly of Invariable Density, David Taylor Model Basin, Washington D.C., Report 2299, 1967.

APPENDIX A1

STREAMLINE CALCULATION

When making finite difference approximations to the three-dimensional boundary layer equations written in streamline co-ordinates (section 4.6), it is necessary to be able to determine the position of the streamline (as defined in section 4.3) through any grid point. It is with this problem that the present appendix is concerned.

Figure (A1.1) below, which takes a section through a ζ = constant plane, summarises the situation. The three grid points A,B,C are on the upstream solution face where velocity profiles have previously been calculated so that at these points the directions of the velocity vectors are known,

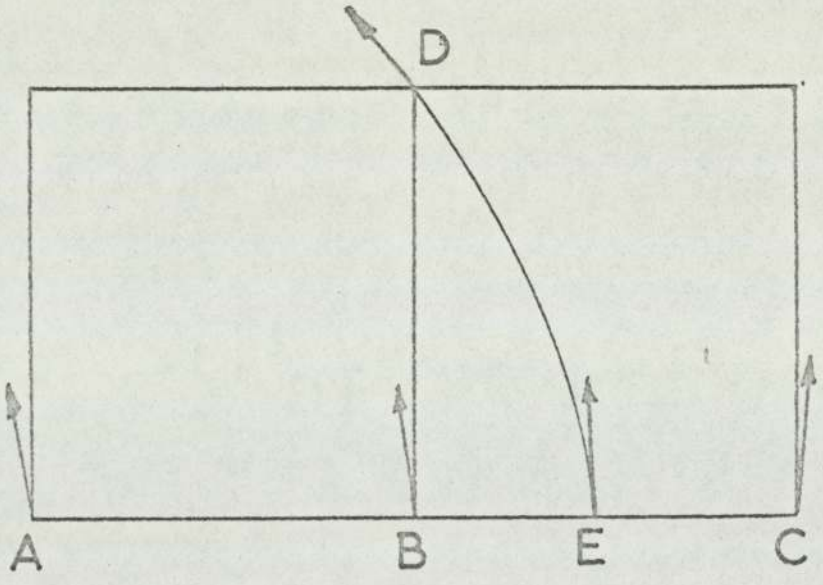


Figure (A1.1)

while the velocity profiles at the section through the point D on the downstream solution face are currently being calculated so that only estimates of the velocity components at D are available. The streamline DE is fitted between D and the line AC so that at the end point D the direction of the streamline is parallel to the current estimate of the velocity direction while at E the direction is as given by interpolating velocity components between the points A,B,C.

The method used to determine the position of the streamline will now be described. A parabolic curve requiring four parameters to define (scale, orientation and lateral and transverse displacements) is fitted between D and the line AC. Obviously specifying any point E on AC where the direction of the velocity is known a quadratic curve DE could be fitted since there are four conditions to be satisfied (position and directional conditions at both D and E). The position of E will however be determined so that if the intercept of the tangents to the quadratic at D and E is F then the two intercept lengths DF, EF are the same. If more than one such point E exists along AC (or AC extended) then that providing the shortest intercept length will be chosen.

For convenience the notation used in this appendix will not be related except in a superficial way to that used elsewhere. The information needed to determine the arc DE is summarised in figure (A1.2) where it is hoped the notation is self-explanatory. It should be noted that y as shown in this diagram increases with decreasing y and is thus measured in the opposite sense to y as defined in section 4.6 (the same will apply to β which is to be introduced below).

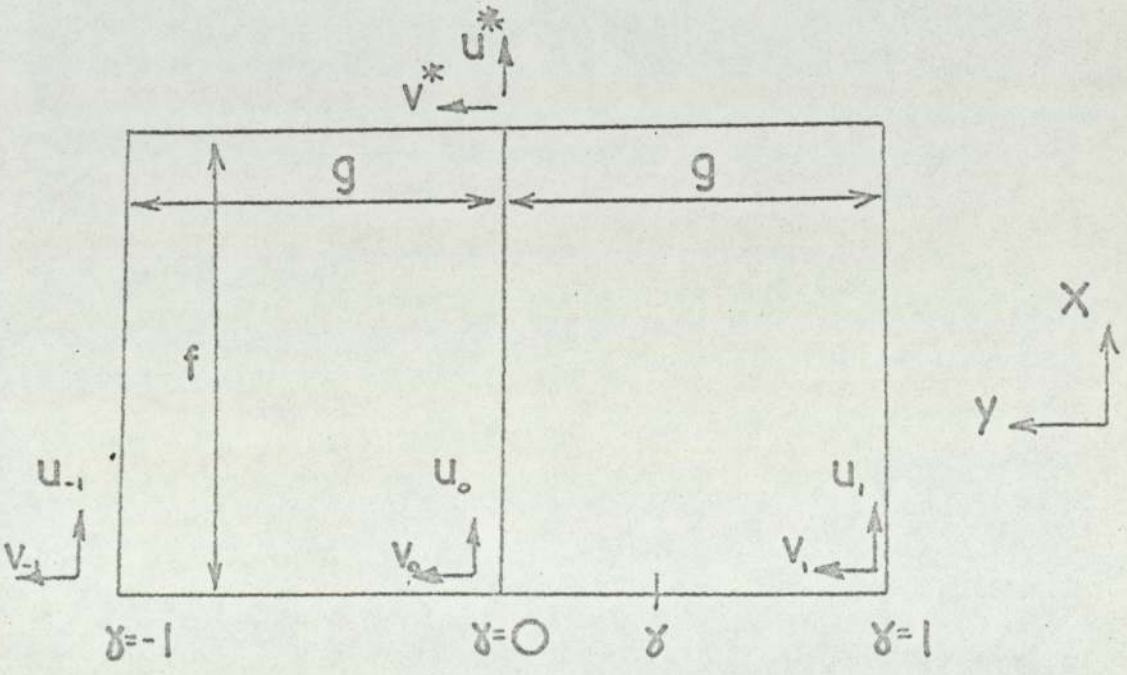


Figure (A1.2)

At any point specified by y on the line denoted by AC in figure (A1.1) the velocity components in the x,y directions can be determined by quadratic interpolation between the three points $y = -1, 0, 1$, i.e.

$$u_y = \frac{u_1 - 2u_0 + u_{-1}}{2} y^2 + \frac{u_1 - u_{-1}}{2} y + u_0 \qquad (A1.1)$$

with a similar expression for v_y. It is anticipated that u will nowhere be zero so that the direction of the streamline at any point can be represented by

$$t = \frac{v}{u}$$

Letting y now denote the point where the required streamline cuts the line AC the quadratic needs then satisfy the two directional conditions

$$t^* = \frac{v^*}{u^*} \qquad t_y = \frac{v_y}{u_y} \qquad (A1.2)$$

where y is yet to be determined and is chosen such that the angles EDF, DEF in figure (A1.3) below are the same. This latter condition leads to the expression

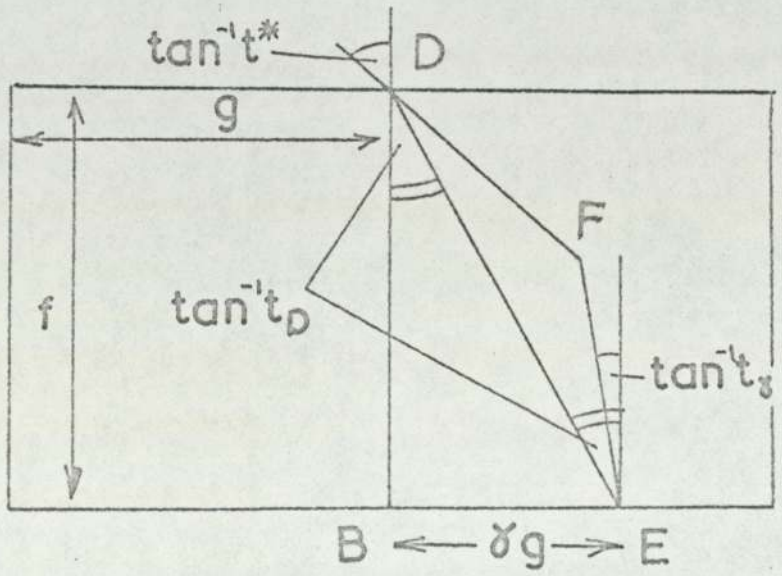


Figure (A1.3).

$$t_D^2 + 2Tt_D - 1 = 0 \qquad \text{(A1.3)}$$

after some manipulation, where

$$\left.\begin{aligned} t_D &= \frac{\gamma g}{f} \\ T &= \frac{1 - t^* t_\gamma}{t^* + t_\gamma} \end{aligned}\right] \qquad \text{(A1.4)}$$

which has to be solved for γ. Remembering that in the most general case t_γ is the ratio of two quadratics in γ we see that the above condition is equivalent to a quartic equation in γ providing, it is to be expected, up to four solutions!

Since no solution can in the most general case be obtained explicitly from the system of equations (A1.1-4) it becomes necessary to establish an iterative scheme and this we do now. Assuming we have an estimate $\gamma^{(r)}$ to a solution γ we first evaluate $t_\gamma^{(r)}$ from equations (A1.1,2) and then $T_\gamma^{(r)}$ using this estimate of t_γ from (A1.4), equation (A1.3) it is suggested can then be solved to obtain an improved estimate of γ

$$t_D^{(r+1)} = \frac{\gamma^{(r+1)} g}{f} = -T^{(r)} \pm \sqrt{1 + T^{(r)2}} \qquad \text{(A1.5)}$$

where the sign associated with the square root is chosen consistently throughout. It is immediately apparent that if the above iterative scheme is convergent it will yield real roots and the two solutions obtained by considering alternative signs in equation (A1.5) will be of opposite sign. In the pseudo-three-dimensional problem where u_γ, v_γ are constant for all γ the solution can be obtained without iteration.

Although only two roots are provided by equation (A1.5) it is not thought necessary to investigate the other two since in the

particular cases when this has been done the remaining roots have been found to be either imaginary or to lack any plausibility as meaningful solutions. No further consideration will then be given to this point since the present scheme has provided reasonable solutions in all the cases treated.

We must however discuss where the above solution scheme breaks down. The only apparent cause of trouble in equation(A1.5) is when T becomes singular which is so when

$$t_y^{(r)} = - t^*$$

and for this to be consistent with a solution we require $y = 0$ i.e.

$$t_o = - t^* \qquad \text{implies } y = 0 \qquad \text{(A1.6)}$$

where we have the situation in figure (A1.4) (which incidentally includes the simple two-dimensional problem). Excepting this case it has been found that the choice of sign in equation (A1.5) is given by considering the sign of $t_o + t^*$ as follows

$$\left.\begin{aligned} t_o + t^* &> 0 \quad \text{implies } y > 0 \quad (+ \text{ sign}) \\ t_o + t^* &< 0 \quad \text{implies } y < 0 \quad (- \text{ sign}) \end{aligned}\right] \qquad \text{(A1.7)}$$

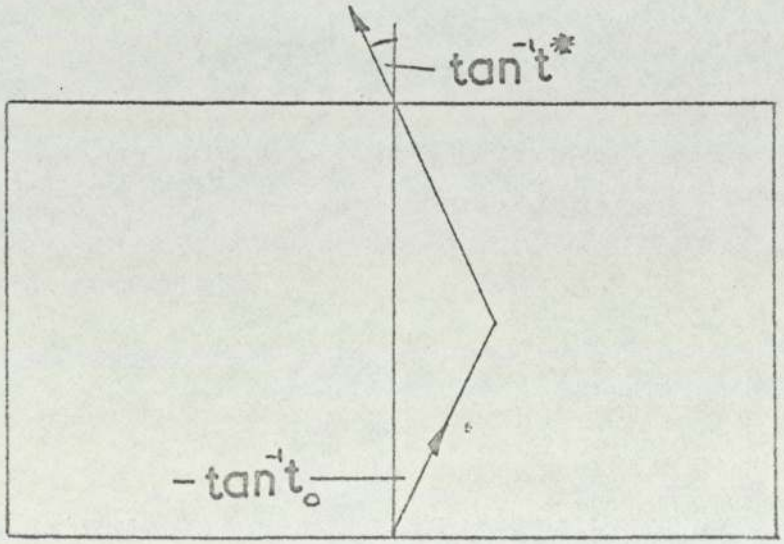


Figure (A1.4)

Being in a position to calculate y we can now proceed to fit a curve between the end points of the streamline which are now known. It has been found adequate for the purpose of the present finite difference scheme to restrict the point at which the finite difference approximation is to be taken to the mid-point of the streamline which in the present calculations will considerably simplify the algebra. Even in this simplified problem the algebra necessary to fit the quadratic and determine its mid-point and length is tedious so we will here only quote the results of the manipulations to determine α,β,s shown in figure (A1.5)

$$\alpha = \frac{1}{4}\ \frac{2+3t_D t^* - t_D^2}{1+t_D t^*} \tag{A1.8}$$

$$\beta = \frac{f}{4g}\ \frac{t^*+t_D+2t_D^2\ t^*}{1+t_D t^*} \tag{A1.9}$$

$$\left.\begin{aligned} s &= a\left(\frac{1}{b}\ \ell n(b+c) + c\right) \\ \text{where} \\ a &= \frac{f}{2}\ \sqrt{1+t_D^2} \\ b &= \frac{t_D - t^*}{1+ t_D t^*} \\ c &= \sqrt{1+b^2} \end{aligned}\right\} \tag{A1.10}$$

The above equations for α,β,s may become singular when

$$t_D = -\ \frac{1}{t^*}$$

which it can readily be shown is inconsistent

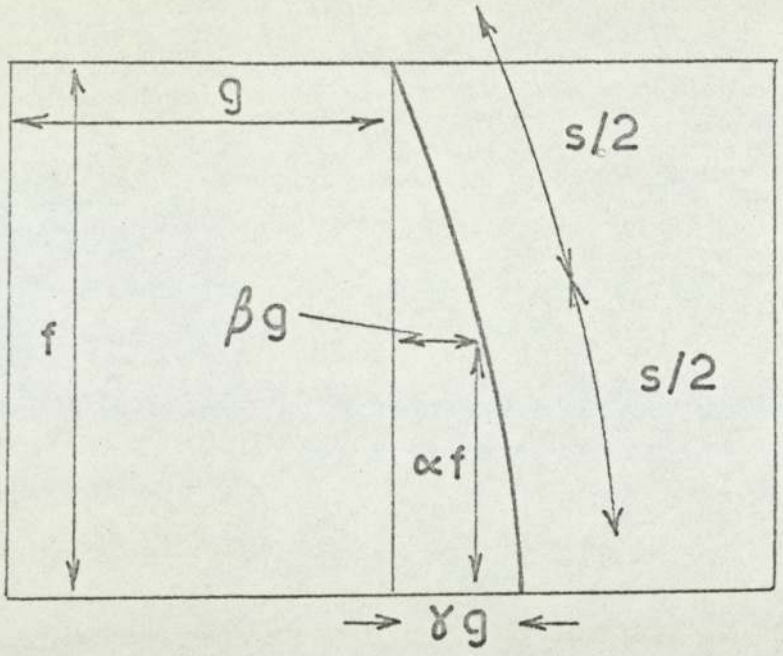


Figure (A1.5)

with a solution, and in addition the expression for s does not hold when

$$t_D = t^*$$

in which case the streamline is linear and the length s should then be calculated from

$$s = 2a \quad \text{when } b = 0 \qquad \text{(A1.11)}$$

while expressions for α, β remain unchanged.

The equations (A1.1-11) in this appendix thus form the basis of the streamline calculation subroutine which is included in the computer program listing (Appendix A6). Again it will be emphasised that the signs of β, γ used in this appendix are opposite to those used in Chapter Four.

APPENDIX A2.

FINITE DIFFERENCE APPROXIMATIONS TO THE MOMENTUM EQUATIONS AT THE POINT $(\ell+1,m,n)$.

This appendix provides a detailed description of the finite difference approximations to the momentum equations referred to in section 4.6. The notation used is that indicated in figure (4.6.2) for the computable quantities, and the unknown quantities (i.e. iterates) will be referred to by subscripting u,v by their point number and superscripting them by r (iteration number) e.g. u^r_{n-1}. The weights to be introduced will be denoted by $\phi(\psi=1-\phi)$ subscripted by an integer which by reference to table 4.6.1 will distinguish between their different uses.

We begin by stating the finite difference approximations to the individual terms of the x momentum equation (equation (4.3.9)) which, using the contractions

$$C = -a\xi(\phi_1 u_{\ell_2} + \psi_1 u_2) + w_2$$

$$D = a(x_\ell + \alpha f) + b$$

can be written:

$$q\frac{\partial u}{\partial s} = \frac{\psi_2}{2} q_1 \left(\frac{\psi_3 u^r_{n-1} + \phi_3 u_1 - u_{\ell_1}}{s}\right) + \phi_2 q_2 \left(\frac{\psi_3 u^r_n + \phi_3 u_2 - u_{\ell_2}}{s}\right) + \frac{\psi_2}{2} q_3 \left(\frac{\psi_3 u^r_{n+1} + \phi_3 u_3 - u_{\ell_3}}{s}\right)$$

$$\frac{-a\xi u+w}{ax+b}\frac{\partial u}{\partial \xi} = \frac{C}{D}\left(\phi_1 \frac{u_{\ell_3} - u_{\ell_1}}{2\xi_{inc}} + \psi_1\left(\psi_4 \frac{u^r_{n+1} - u^r_{n-1}}{2\xi_{inc}} + \phi_4 \frac{u_3-u_1}{2\xi_{inc}}\right)\right)$$

$$\frac{1}{ax+b}\frac{\partial}{\partial \xi}\left(\nu'_e \frac{\partial u}{\partial \xi}\right) = \frac{\phi_1}{D\xi_{inc}}\left(\nu_{\ell_2}\frac{u_{\ell_3}-u_{\ell_2}}{\xi_{inc}} - \nu_{\ell_1}\frac{u_{\ell_2}-u_{\ell_1}}{\xi_{inc}}\right) + \frac{\psi_1}{D\xi_{inc}}\left(\phi_5\left(\nu_2\frac{u_3-u_2}{\xi_{inc}} - \nu_1\frac{u_2-u_1}{\xi_{inc}}\right)\right.$$

$$+ \psi_5 \left(\nu_2 \frac{u^r_{n+1} - u^r_n}{\xi_{inc}} - \nu_1 \frac{u^r_n - u^r_{n-1}}{\xi_{inc}} \right) \Big) \qquad \text{(A2.1)}$$

Substituting these approximations into the relevant equation and collecting together terms involving the iterated u components on to the left hand side we obtain

$$u^r_{n-1} \left(\frac{\psi_2\psi_3}{2s} q_1 - \frac{C}{D} \frac{\psi_1\psi_4}{2\xi_{inc}} - \frac{\psi_1\psi_5}{D\xi^2_{inc}} \nu_1 \right)$$

$$+ u^r_n \left(\frac{\phi_2\psi_3}{s} q_2 + \frac{\psi_1\psi_5}{D\xi^2_{inc}} (\nu_1+\nu_2) \right)$$

$$+ u^r_{n+1} \left(\frac{\psi_2\psi_3}{2s} q_3 + \frac{C}{D} \frac{\psi_1\psi_4}{2\xi_{inc}} - \frac{\psi_1\psi_5}{D\xi^2_{inc}} \nu_2 \right)$$

$$= - \frac{\psi_2}{2s} q_1 (\phi_3 u_1 - u_{\ell_1}) - \frac{\phi_2}{s} q_2 (\phi_3 u_2 - u_{\ell_2})$$

$$- \frac{\psi_2}{2s} q_3 (\phi_3 u_3 - u_{\ell_3})$$

$$- \frac{C}{2D\xi_{inc}} (\phi_1 (u_{\ell_3} - u_{\ell_1}) + \psi_1\phi_4 (u_3 - u_1))$$

$$+ P_u + \frac{1}{D\xi^2_{inc}} \Big[\phi_1 (\nu_{\ell_2} (u_{\ell_3} - u_{\ell_2}) - \nu_{\ell_1} (u_{\ell_2} - u_{\ell_1}))$$

$$+ \psi_1\phi_5 (\nu_2 (u_3 - u_2) - \nu_1 (u_2 - u_1)) \Big] \qquad \text{(A2.2)}$$

where

$$P_u = \left(U \frac{\partial U}{\partial x} + V \frac{\partial V}{\partial x} \right)_{\ell+\alpha, m+\beta} \qquad \text{(A2.3)}$$

Here as in figure (4.6.2) ν refers to the transformed effective viscosity ν'_e (equation (4.3.8)) which will have to be evaluated from some hypothetical relationship at the points indicated in figure (4.6.2).

As mentioned in section 4.6, when the above equations are evaluated at $n = \lambda\omega$ special care should be taken and in particular

subscripts n-1 in the above equations (and n-1 in figure (4.6.2)) should be replaced by $n-\omega$. In additon to make the resulting set of equations tri-diagonal this equation, which we suppose can be written

$$\gamma\, u^{r}_{n-\omega} + \alpha\, u^{r}_{n} + \beta\, u^{r}_{n+1} = \delta' \qquad (A2.4)$$

where $n = \lambda\omega$ and $\alpha, \beta, \gamma, \delta'$ can all be calculated, is transformed into

$$\gamma\, u^{r}_{n-1} + \alpha\, u^{r}_{n} + \beta\, u^{r}_{n+1} = \delta \qquad (A2.5)$$

by introducing

$$\delta = \delta' + \gamma(u^{r-1}_{n-1} - u^{r-1}_{n-\omega}) \qquad (A2.6)$$

This equation (A2.5) now conforms to the general format.

The finite difference approximation to the momentum equation corresponding to the y direction (equation (4.3.10)) is very similar to that given above for the x momentum equation and we will only list the differences. Following through the same process that led to equation(A2.2) we find that the coefficients of v^{r}_{n-1}, v^{r}_{n}, v^{r}_{n+1} are identical to those corresponding to the u's on the left hand side of equation (A2.2) while on the right hand side u where it occurs explicitly is replaced by v and P_u is replaced by P_v:

$$P_v = \left(U\frac{\partial U}{\partial y} + V\frac{\partial V}{\partial y}\right)_{\ell+\alpha, m+\beta} \qquad (A2.7)$$

(the definition of C remains unchanged). Other points mentioned above with respect to the x momentum equation obviously apply equally to the y momentum equation.

Thus we are able to relate the three iterated velocity components in both x,y directions by an equation which has the form of equation (A2.5) where moreover at any section for the same n,r the coefficients α, β, γ are identical for the two equations relating u's

and v's respectively. This means ~~that~~ in addition that all the unknown u (or v) iterated components at any section are related by a tri-diagonal set of linear algebraic equations and the matrix of coefficients of the unknowns is the same for both u and v, the differences being on the right hand sides. Such a situation obviously simplifies the problems associated with the storage of coefficients and the solution of the linear equations.

APPENDIX A3

FINITE DIFFERENCE APPROXIMATIONS TO THE MOMENTUM EQUATIONS AT THE LOG-POINT AT SECTION (ℓ', m).

Not being able to apply the finite difference approximations developed in Appendix A2 when $n=n^*$ we must consider this particular case separately. To approximate to the momentum equation associated with the x-direction at the log-point we proceed as was outlined in section 4.6 making use of equations obtained in section 4.4.

As in Appendix A2 we will first state the finite difference approximations to the individual terms of equation (4.3.9) using the notation of figure (4.6.2). These are as follows:-

$$q\frac{\partial u}{\partial s} = \frac{\psi_6}{2} q_2 \left(1 - \frac{1}{\epsilon} \ell n \ \frac{\xi_2}{\xi_1}\right)$$

$$\left(\frac{\ell n\ \xi_3/\xi_1}{\ell n\ \xi_3/\xi_2}\left(\frac{\psi_3\ u_n^r + \phi_3 u_2 - u_{\ell_2}}{s}\right)\right.$$

$$\left. - \frac{\ell n\ \xi_2/\xi_1}{\ell n\ \xi_3/\xi_2}\left(\frac{\psi_3 u_{n+1}^r + \phi_3 u_3 - u_{\ell_3}}{s}\right)\right)$$

$$+ \phi_6 q_2 \left(\frac{\psi_3 u_n^r + \phi_3 u_2 - u_{\ell_2}}{s}\right)$$

$$+ \frac{\psi_6}{2} q_3 \left(\frac{\psi_3\ u_{n+1}^r + \phi_3 u_3 - u_{\ell_3}}{s}\right)$$

(using equations (4.4.8,9)) which we rewrite

$$q\frac{\partial u}{\partial s} = \left(\phi_6 + \frac{\psi_6}{2} E_1 E_2\right) q_2$$

$$\left(\frac{\psi_3 u_n^r + \phi_3 u_2 - u_{\ell_2}}{s}\right)$$

$$+ \frac{\psi_6}{2}\left(q_3 + q_2 E_1(1-E_2)\right)$$

$$\left(\frac{\psi_3\ u_{n+1}^r + \phi_3 u_3 - u_{\ell_3}}{s}\right) \qquad (A3.1)$$

defining E_1, E_2 as follows (where as above we will make use of the following contractions $\xi_1 = \xi_{n-1}$, $\xi_2 = \xi_n$, $\xi_3 = \xi_{n+1}$)

$$E_1 = 1 - \frac{1}{\epsilon} \ell n \ \frac{\xi_3}{\xi_1}$$

$$E_2 = \frac{\ell n \ \xi_3/\xi_1}{\ell n \ \xi_3/\xi_2}$$

$$= 1 + \frac{\ell n \ \xi_2/\xi_1}{\ell n \ \xi_3/\xi_2} \qquad \text{(A3.2)}$$

To continue:

$$\frac{-a\ \xi u+w}{ax+b} \ \frac{\partial u}{\partial \xi} = \psi_7 \ \frac{C}{2D} \frac{\psi_1(\psi_4 \ u_n^r + \phi_1 u_2)+u_{\ell 2}\phi_1}{\epsilon \ \xi_n}$$

$$+ \ \phi_7 \ \frac{C}{2D} \left(1 + \frac{\xi_{n+\frac{1}{2}}}{\xi_{n-\frac{1}{2}}}\right)$$

$$\left(\psi_1\phi_4 \ \frac{u_3-u_2}{\xi_{inc}} + \psi_1\psi_4 \ \frac{u_{n+1}^r - u_n^r}{\xi_{inc}} + \phi_1 \ \frac{u_{\ell 3}-u_{\ell 2}}{\xi_{inc}}\right) \qquad \text{(A3.3)}$$

(using equation (4.4.6)) where C,D are defined in Appendix A2, and

$$\frac{1}{ax+b} \ \frac{\partial}{\partial \xi}\left(\nu_e' \ \frac{\partial u}{\partial \xi}\right) =$$

$$\frac{\phi_1}{D\xi_{inc}^2}\left(\nu_{\ell 2}\ (u_{\ell 3} - u_{\ell 2}) - \nu_{\ell 1} E_3 u_{\ell 2}\right)$$

$$+ \ \frac{\psi_1\phi_5}{D\xi_{inc}^2}\left(\nu_2\ (u_3-u_2) - \nu_1 E_3 u_2\right)$$

$$+ \ \frac{\psi_1\psi_5}{D\xi_{inc}^2}\left(\nu_2\ (u_{n+1}^r - u_n^r) - \nu_1 E_3 u_n^r\right) \qquad \text{(A3.4)}$$

(using equation (4.4.6)) where we have introduced another contraction

$$E_3 = \frac{1}{\epsilon(n-\frac{1}{2})} \qquad \text{(A3.5)}$$

Equations (A3.1-5) are all to be applied at $n=n^*$. Substituting the finite difference approximations to the individual terms into equation (4.3.9) and collecting together terms in u_n^r, u_{n+1}^r on to the left hand side we have

$$u_n^r \left(\psi_3 \left(\phi_6 + \frac{\psi_6}{2} E_1 E_2 \right) \frac{q_2}{s} \right.$$

$$+ \frac{C}{2D} \frac{\psi_4 \psi_7}{\epsilon \xi_n} - \frac{C}{2D} \phi_7 \frac{2n}{n-\frac{1}{2}} \frac{\psi_1 \psi_4}{\xi_{inc}}$$

$$\left. + \frac{\psi_1 \psi_5}{D \xi_{inc}^2} (\nu_2 + \nu_1 E_3) \right)$$

$$+ u_{n+1}^r \left(\frac{\psi_3 \psi_6}{2s} (q_3 + q_2 E_1 (1-E_2)) \right.$$

$$\left. + \frac{C}{2D} \phi_7 \frac{2n}{n-\frac{1}{2}} \frac{\psi_1 \psi_4}{\xi_{inc}} - \frac{\psi_1 \psi_5}{D \xi_{inc}^2} \nu_2 \right)$$

$$= - \left(\phi_6 + \frac{\psi_6}{2} E_1 E_2 \right) q_2 \frac{\phi_3 u_2 - u_{\ell 2}}{s}$$

$$- \frac{\psi_6}{2} (q_3 + q_2 E_1 (1-E_2)) \frac{\phi_3 u_3 - u_{\ell 3}}{s}$$

$$- \frac{C}{2D} \psi_7 \frac{\psi_1 \phi_4 u_2 + u_{\ell 2} \phi_1}{\epsilon \xi_n}$$

$$- \phi_7 \frac{C}{2D} \frac{2n}{n-\frac{1}{2}} \frac{1}{\xi_{inc}} (\psi_1 \phi_4 (u_3 - u_2) + \phi_1 (u_{\ell 3} - u_{\ell 2}))$$

$$+ P_u + \frac{1}{D} \frac{\phi_1}{\xi_{inc}} (\nu_{\ell 2} (u_{\ell 3} - u_{\ell 2}) - \nu_{\ell 1} u_{\ell 2} E_3)$$

$$+ \frac{1}{D} \frac{\phi_5 \psi_1}{\xi_{inc}^2} (\nu_2 (u_3 - u_2) - \nu_1 u_2 E_3) \qquad \text{(A3.6)}$$

which again is to be applied at $n=n^*$. The finite difference equation corresponding to the y momentum equation can be obtained by precisely the same means as were described in Appendix A2 (P_u, P_v are also as defined in this last appendix). The only outstanding consideration is the problem of obtaining ϵ which occurs in equation (A3.6) as well as in the definitions of E_1, E_3. Only approximations to ϵ will be available initially since ϵ will be obtained of necessity from an estimate of q at the log-point using equations (4.4.10,11). More accurate values of ϵ are

obtained as the iteration progresses until finally when the procedure has converged the true value of ϵ can be used to obtain the coefficient of friction at the wall from equation (4.4.5).

APPENDIX A4

MODIFICATIONS NECESSARY TO THE FINITE DIFFERENCE SCHEME FOR IT TO BE APPLICABLE TO THE LAMINAR PROBLEM.

The finite difference equations obtained in sections 4.6 and 4.7 and Appendices A2, A3 were concerned exclusively with turbulent flows and as yet little mention has been made of the laminar boundary layer. Although our prime concern here is the turbulent problem the program as written will cater for laminar flows.

Within the boundary layer equations the only alteration necessary for the equations to hold for laminar flows is that the transformed effective viscosity ν'_e should be replaced by the transformed kinematic viscosity ν'

$$\nu' = (ax+b)\nu'$$

This alteration is readily carried into the finite difference approximation to the momentum equations. In addition in these approximations however we need insert a different boundary condition at the wall. Obviously in laminar flows the mesh at the wall may be somewhat coarser than that necessary for turbulent flows but however we will still require that the grid points at the wall will be close enough to ensure that the gradients of the u,v profiles over the first two grid intervals are constant. We can now make use of equation (A3.6) if we apply it at $n=n^*=1$ and put

$$\psi_7 = 1$$

$$\epsilon = 1$$

$$E_1 = 0$$

$$E_3 = 1$$

(E_2 is now redundant). Putting $\epsilon = 1$ with $n=n^*=1$ also allows the finite difference approximation to the continuity equation at the wall obtained in section 4.7 to be used.

APPENDIX A5

SOLUTION OF TRI-DIAGONAL LINEAR ALGEBRAIC EQUATIONS.

A system of I tri-diagonal linear algebraic equations in the I unknowns $u_i (i = 1,2, \ldots I)$

$$\begin{aligned} \alpha_1 u_1 + \beta_1 u_2 &= \delta_1 \\ \gamma_{i-1} u_{i-1} + \alpha_i u_i + \beta_i u_{i+1} &= \delta_i \qquad i = 2,3, \ldots I-1 \\ \gamma_{I-1} u_{I-1} + \alpha_I u_I &= \delta_I \end{aligned}$$

where the coefficients $\alpha, \beta, \gamma, \delta$ are known, can be solved using the following algorithm

$$\left.\begin{aligned} \alpha_i &= \alpha_i - \beta_{i-1} \frac{\gamma_{i-1}}{\alpha_{i-1}} \\ \delta_i &= \delta_i - \delta_{i-1} \frac{\gamma_{i-1}}{\alpha_{i-1}} \end{aligned}\right\} \quad i = 2,3, \ldots I$$

$$\delta_I = \frac{\delta_I}{\alpha_I}$$

$$\delta_i = \frac{\delta_i - \beta_i \delta_{i+1}}{\alpha_i} \qquad i = I-1, I-2, \ldots 1$$

where the equals sign has its usual programming significance and operations are to be performed in precisely the order indicated, the solution being finally given by

$$u_i = \delta_i \qquad i = 1,2, \ldots I$$

It might be noted here, and it is of particular relevance to the present solution scheme, that when we have two sets of I linear equations in respectively u_i, v_i say and the coefficients of the v_i are the same as those of the corresponding u_i, the differences between the two sets of equations being confined solely to the right hand sides, the second set can be solved simultaneously with the first with only a slight increase in storage requirements.

APPENDIX A6

PROGRAM LISTING

```
      INTEGER OMEGA                                                     0001AC99
      REAL  NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                         0002AC99
      EXTERNAL AC99Y                                                    0002AC99
      COMMON    MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,           0003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),           0004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),V0(50),V00(50),                 0005AC99
     3XL,Y0,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                        0006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                          0007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                           0008AC99
      COMMON    KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,          0009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,               0010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,         0011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                               0012AC99
     4D,D1,D2,T11,T12,T21,T22,HAPE,RT11,QT,CF,CFX,CF1,CF2,PHI           0013AC99
      DIMENSION HEAD1(20),HEAD2(20),TAPHI(10)                           0020AC99
      MIN=1                                                             0030AC99
      MOUT=3                                                            0040AC99
      READ(MIN,60) (HEAD1(I),I=1,20)                                    0050AC99
      READ(MIN,60) (HEAD2(I),I=1,20)                                    0060AC99
      READ(MIN,62) KAPPA,KAPPAL,K,A,CMIN                                0070AC99
      READ(MIN,61) MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NSMAX,              0080AC99
     1                                        ITMAX,LFREQ,MFREQ         0085AC99
      READ(MIN,62) XL,Y0,F,G,THETA,NU                                   0090AC99
      READ(MIN,62) PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8              0100AC99
      READ(MIN,62) (A1(N),N=1,NN)                                       0110AC99
      READ(MIN,62) (A2(N),N=1,NN)                                       0113AC99
      READ(MIN,66) (TAPHI(M),M=1,MM)                                    0116AC99
      READ(MIN,63) TOL                                                  0120AC99
      PHI1=1.0-PSI1                                                     0130AC99
      PHI2=1.0-PSI2                                                     0140AC99
      PHI3=1.0-PSI3                                                     0150AC99
      PHI4=1.0-PSI4                                                     0160AC99
      PHI5=1.0-PSI5                                                     0170AC99
      PHI6=1.0-PSI6                                                     0180AC99
      PHI7=1.0-PSI7                                                     0190AC99
      PHI8=1.0-PSI8                                                     0200AC99
```

```
      DO 1 M=1,MM                                                        0205AC99
      UU=AC99U(XL,YO+(M-1)*G)                                            0210AC99
      VV=AC99V(XL,YO+(M-1)*G)                                            0220AC99
      PI=TAPHI(M)*(A1(1)/A2(1))                                          0230AC99
      DO 1 N=1,NN                                                        0240AC99
      UL(M,N)=UU*A1(N)-PI*VV*A2(N)                                       0250AC99
      VL(M,N)=VV*A1(N)+PI*UU*A2(N)                                       0270AC99
      U(M,N)=0.0                                                         0290AC99
      V(M,N)=0.0                                                         0300AC99
      W(M,N)=0.0                                                         0310AC99
    1 CONTINUE                                                           0320AC99
      U1=UL(1,NN)                                                        0330AC99
      V1=VL(1,NN)                                                        0340AC99
      Q1=SQRT(U1**2+V1**2)                                               0350AC99
      H=THETA/AC99T(AC99Y,1,NMINC,NN,OMEGA,1.0,1,1,0,0.0)                0360AC99
      DSTER=AC99T(AC99Y,1,NMINC,NN,OMEGA,H,1,0,0,0.0)                    0370AC99
      HAPE=DSTER/THETA                                                   0380AC99
      RTHET=THETA*Q1/NU                                                  0390AC99
      WRITE(MOUT,64) (HEAD1(I),I=1,20),                                  0400AC99
     1                (HEAD2(I),I=1,20),XL,NSMAX,F,TOL                  0410AC99
      WRITE(MOUT,65) THETA,HAPE,RTHET,NU,                                0420AC99
     1          PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8                  0430AC99
      AT=0.0                                                             0440AC99
      BT=1.0                                                             0450AC99
      NSTEP=1                                                            0460AC99
    2 CONTINUE                                                           0470AC99
      CALL AC99H                                                         0475AC99
      CALL AC991                                                         0480AC99
      CALL AC990                                                         0490AC99
      IF(NSTEP-NSMAX) 3,5,5                                              0495AC99
    3 XL=XL+F                                                            0500AC99
      H=H*(AT*XL+BT)                                                     0510AC99
      AT=AT/(AT*F+1.0)                                                   0520AC99
      BT=1.0-AT*XL                                                       0530AC99
      DO 4 M=1,MM                                                        0560AC99
      DO 4 N=1,NN                                                        0570AC99
```

```
      UL(M,N)=U(M,N)                                                    0580AC99
      VL(M,N)=V(M,N)                                                    0590AC99
    4 CONTINUE                                                          0600AC99
      NSTEP=NSTEP+1                                                     0610AC99
      GO TO 2                                                           0620AC99
    5 CALL EXIT                                                         0630AC99
   60 FORMAT(20A4)                                                      0640AC99
   61 FORMAT(11I3)                                                      0650AC99
   62 FORMAT(8F10.6)                                                    0660AC99
   63 FORMAT(E10.3)                                                     0670AC99
   64 FORMAT(1H1///2(1H ,25X,20A4//)//1H ,5X,30HTHE CALCULATION STARTS F0680AC99
     1ROM X=,F10.6,17H AND PROCEEDS FOR,I3,9H STEPS OF,F10.6,15H,A TOLER0690AC99
     2ANCE OF,E10.3,16H BEING SATISFIED)                                0700AC99
   65 FORMAT(1HO,16X,27HINITIAL CONDITIONS ARE O11=,F10.6,4H  H=,F10.6, 0710AC99
     17H  R   =,F10.3,20H (LAMINAR VISCOSITY=,F11.8,1H)/1H+,39X,1H-/1H ,0720AC99
     270X,3HO11/1H+,70X,1H-/1HO,50X,31HSOLUTION PROCEDURE WEIGHTS WERE/ 0730AC99
     32(45X,4F10.4/)//1H1)                                              0740AC99
   66 FORMAT(10F8.4)                                                    0750AC99
      END                                                               0760AC99
```

```
      SUBROUTINE AC991                                                  1000AC99
C          THIS SUBROUTINE CALCULATES VELOCITY PROFILES                 1   AC99
C          AT THE DOWNSTREAM SOLUTION FACE                              1   AC99
      INTEGER OMEGA                                                     1001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                          1002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,            1003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),           1004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),VO(50),VOO(50),                 1005AC99
     3XL,YO,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                        1006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                          1007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                           1008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,           1009AC99
```

```
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,                1010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,         1011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                               1012AC99
     4CSTAR(10),C,C,E1,E2,E3                                            1013AC99
      IF(NSTEP-1) 1,1,2                                                 1020AC99
    1 ITN=-1                                                            1030AC99
      GO TO 3                                                           1040AC99
    2 ITN=0                                                             1050AC99
    3 CONTINUE                                                          1060AC99
      CALL AC990                                                        1070AC99
      CALL AC99X                                                        1080AC99
    4 CONTINUE                                                          1090AC99
      IF(ITN) 5,5,8                                                     1100AC99
    5 DO 7 M=1,MM                                                       1110AC99
      USC=U(M,NN)/UL(M,NN)                                              1120AC99
      IF(-ABS(VL(M,NN)/UL(M,NN))-0.001) 55,56,56                        1125AC99
   55 VSC=0.0                                                           1130AC99
      VSCU=V(M,NN)/UL(M,NN)                                             1132AC99
      GO TO 57                                                          1135AC99
   56 VSC=V(M,NN)/VL(M,NN)                                              1140AC99
      VSCU=0.0                                                          1142AC99
   57 N1=NCPTS(M)                                                       1145AC99
      DO 6 N=1,N1                                                       1150AC99
      U(M,N)=USC*UL(M,N)                                                1160AC99
    6 V(M,N)=VSC*VL(M,N)+VSCU*UL(M,N)                                   1170AC99
      DO 7 N=N1,NN                                                      1180AC99
      U(M,N)=U(M,NN)                                                    1190AC99
    7 V(M,N)=V(M,NN)                                                    1200AC99
    8 CONTINUE                                                          1210AC99
      CALL AC99A                                                        1215AC99
      CALL AC992                                                        1220AC99
      ITN=ITN+1                                                         1230AC99
      IF(ITN) 12,12,9                                                   1240AC99
    9 IF(SOS/NOSOS-TOL) 16,16,10                                        1250AC99
   10 IF(ITN-ITMAX) 12,12,11                                            1260AC99
   11 WRITE(MOUT,17) NSTEP,SOS,NOSOS                                    1270AC99
```

```
      GO TO 16                                                          1280AC99
   12 CALL AC99B(2,1.0)                                                 1290AC99
      BLMAX=BLT(1)                                                      1300AC99
      DO 14 M=1,MM                                                      1310AC99
      IF(BLMAX-BLT(M)) 13,14,14                                         1320AC99
   13 BLMAX=BLT(M)                                                      1330AC99
   14 CONTINUE                                                          1340AC99
      IF(ABS(BLMAX+NSLAP-NN)-0.01) 8,8,15                               1350AC99
   15 HLP1=(BLMAX-NMINC+LAMDA)*H*(AT*(XL+F)+BT)                         1360AC99
     1                    /(NN-NMINC+LAMDA-NSLAP)                      1365AC99
      AT=(HLP1-H)/(F*H)                                                 1370AC99
      BT=1.0-AT*XL                                                      1380AC99
      GO TO 4                                                           1390AC99
   16 RETURN                                                            1400AC99
   17 FORMAT(1HO,5X,3HE11,5X,I6,E12.5,I6)                               1410AC99
      END                                                               1420AC99
```

```
      SUBROUTINE AC992                                                  2000AC99
C          THIS SUBROUTINE ITERATES ONCE FOR THE VELOCITY               2    AC99
C          PROFILES AT THE DOWNSTREAM SOLUTION FACE                     2    AC99
      INTEGER OMEGA                                                     2001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                          2002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,            2003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),           2004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),V0(50),V00(50),                 2005AC99
     3XL,Y0,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                        2006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                          2007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                           2008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,           2009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,               2010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,         2011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                               2012AC99
```

```
     4DSTAR(10),C,D,E1,E2,E3                                          2013AC99
      SOS=0.0                                                         2020AC99
      NOSOS=0                                                         2030AC99
      DO 1 M=1,MM                                                     2040AC99
      CALL AC99R(M)                                                   2050AC99
    1 CONTINUE                                                        2060AC99
      M=1                                                             2070AC99
    2 N=LOGPT(M)                                                      2080AC99
      NEQ=1                                                           2090AC99
      IL=1                                                            2093AC99
      ILP1=1                                                          2096AC99
      CALL AC99S(M,N)                                                 2100AC99
      CALL AC99M(M,N)                                                 2110AC99
      CALL AC99D(M,N,IL,ILP1)                                         2115AC99
      CALL AC99F(M,N)                                                 2120AC99
    3 N=N+1                                                           2130AC99
      CALL AC99S(M,N)                                                 2150AC99
      NEQ=NEQ+1                                                       2140AC99
      CALL AC99M(M,N)                                                 2160AC99
      CALL AC99D(M,N,IL,ILP1)                                         2165AC99
      CALL AC99G(NEQ,M,N)                                             2170AC99
      IF(N-NOPTS(M)+1) 3,4,4                                          2180AC99
    4 CALL AC99Z(NEQ)                                                 2190AC99
      SOS1=0.0                                                        2200AC99
      I1=LOGPT(M)-1                                                   2210AC99
      DO 5 I=1,NEQ                                                    2220AC99
      J=I+I1                                                          2230AC99
      SOS1=SOS1+(SQRT(U(M,J)**2+V(M,J)**2)                            2240AC99
     1          -SQRT(A4(I)**2+A5(I)**2))**2                          2250AC99
    5 CONTINUE                                                        2260AC99
      NOSOS=NOSOS+NEQ                                                 2270AC99
      SOS=SOS+SOS1/(U(M,NN)**2+V(M,NN)**2)                            2280AC99
      DO 6 I=1,NEQ                                                    2290AC99
      J=I+I1                                                          2300AC99
      U(M,J)=PSI8*A4(I)+PHI8*U(M,J)                                   2310AC99
      V(M,J)=PSI8*A5(I)+PHI8*V(M,J)                                   2320AC99
```

```
 6 CONTINUE                                                          2330AC99
   GO TO (75,64),LT                                                  2335AC99
64 N=LOGPT(M)                                                        2340AC99
   Q2=SQRT(U(M,N)**2+V(M,N)**2)                                      2350AC99
   ALPHA=1.0                                                         2360AC99
   CALL AC99L(N)                                                     2370AC99
   QT=KAPPAL*Q2/EPS                                                  2373AC99
   DZ=(AT*(XL+F)+BT)*H/OMEGA                                         2376AC99
   N=N-1                                                             2380AC99
   DO 7 I=1,N                                                        2390AC99
   ZP=I*DZ*QT/NU                                                     2394AC99
   IF(ZP-11.0) 65,65,66                                              2398AC99
65 QI=QT*ZP                                                          2402AC99
   GO TO 67                                                          2406AC99
66 QI=QT*(ALOG(ZP)/KAPPAL+A)                                         2410AC99
67 U(M,I)=U(M,N+1)*QI/Q2                                             2414AC99
   V(M,I)=V(M,N+1)*QI/Q2                                             2418AC99
 7 CONTINUE                                                          2420AC99
75 IF(M-MM) 8,9,9                                                    2430AC99
 8 M=M+1                                                             2440AC99
   GO TO 2                                                           2450AC99
 9 D=AT*(XL+0.5*F)+BT                                                2460AC99
   M=1                                                               2470AC99
10 N=LOGPT(M)                                                        2480AC99
11 CALL AC99C(M,N)                                                   2490AC99
   CALL AC99W(M,N)                                                   2500AC99
   IF(N-NN) 12,13,13                                                 2510AC99
12 N=N+1                                                             2520AC99
   GO TO 11                                                          2530AC99
13 CONTINUE                                                          2540AC99
   IF(M-MM) 14,15,15                                                 2550AC99
14 M=M+1                                                             2560AC99
   GO TO 10                                                          2570AC99
15 RETURN                                                            2580AC99
   END                                                               2590AC99
```

```
      SUBROUTINE AC99A                                                  A000AC99
      INTEGER OMEGA                                                     A001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                          A002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,            A003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),           A004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),VO(50),VOO(50),                 A005AC99
     3XL,YO,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                        A006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                          A007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                           A008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,           A009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,               A010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,         A011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                               A012AC99
     4DSTAR(10),C,D,E1,E2,E3                                            A013AC99
      DO 1 N=1,NN                                                       A020AC99
      VO(N)=0.0                                                         A030AC99
    1 VOO(N)=0.0                                                        A040AC99
      RETURN                                                            A050AC99
      END                                                               A060AC99
```

```
      SUBROUTINE AC99B(IT,AS)                                           B000AC99
C         THIS SUBROUTINE CALCULATES THE BOUNDARY LAYER                 B   AC99
C         THICKNESS AT EACH SECTION .                                   B   AC99
      INTEGER OMEGA                                                     B001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                          B002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,            B003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),           B004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),VO(50),VOO(50),                 B005AC99
     3XL,YO,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                        B006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                          B007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                           B008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,           B009AC99
```

```
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,                B010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,         B011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                               B012AC99
     4DSTAR(10),C,D,E1,E2,E3                                            B013AC99
      GO TO (1,2),IT                                                    B020AC99
    1 TOL1=0.99                                                         B030AC99
      GO TO 3                                                           B040AC99
    2 TOL1=0.999                                                        B050AC99
    3 CONTINUE                                                          B060AC99
      DO 8 M=1,MM                                                       B070AC99
      QNP1=(1.0-AS)*SQRT(UL(M,NN)**2+VL(M,NN)**2)                       B080AC99
     1          +AS*SQRT(U(M,NN)**2+V(M,NN)**2)                        B090AC99
      QB=TOL1*QNP1                                                      B100AC99
      N=NN-1                                                            B110AC99
    4 QN=(1.0-AS)*SQRT(UL(M,N)**2+VL(M,N)**2)                           B120AC99
     1        +AS*SQRT(U(M,N)**2+V(M,N)**2)                            B130AC99
      IF(QN-QB) 6,6,5                                                   B140AC99
    5 N=N-1                                                             B150AC99
      QNP1=QN                                                           B160AC99
      GO TO 4                                                           B170AC99
    6 BLT(M)=N+(QB-QN)/(QNP1-QN)                                        B180AC99
      GO TO (7,8),IT                                                    B190AC99
    7 BLT(M)=BLT(M)-NMINC+LAMDA                                         B200AC99
    8 CONTINUE                                                          B210AC99
      RETURN                                                            B220AC99
      END                                                               B230AC99
```

```
      SUBROUTINE AC99C(M,N)                                             C000AC99
C         THIS SUBROUTINE SETS UP ALL THE NECESSARY                     C   AC99
C         QUANTITIES FOR THE FINITE DIFFERANCE APPROXIMATION            C   AC99
C         TO THE CONTINUITY EQUATION                                    C   AC99
      INTEGER OMEGA                                                     C001AC99
```

```
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                          C002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,           C003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),           C004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),V0(50),V00(50),                 C005AC99
     3XL,Y0,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                        C006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                          C007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                           C008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,           C009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,               C010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,         C011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                               C012AC99
     4DSTAR(10),C,D,E1,E2,E3                                            C013AC99
      IF(N-NMINC) 1,1,2                                                 C020AC99
    1 XI=(N*H)/OMEGA                                                    C030AC99
      XINC=H/OMEGA                                                      C040AC99
      GO TO 3                                                           C050AC99
    2 XI=(N-NMINC+LAMDA)*H                                              C060AC99
      XINC=H                                                            C070AC99
    3 IF(N-LOGPT(M)) 4,4,11                                             C080AC99
    4 U1=UL(M,N)                                                        C090AC99
      U3=U(M,N)                                                         C100AC99
      U2=0.5*(U1+U3)                                                    C110AC99
      IF(M-1) 5,5,6                                                     C120AC99
    5 V1=V0(N)                                                          C130AC99
      GO TO 7                                                           C140AC99
    6 V1=0.5*(VL(M-1,N)+V(M-1,N))                                       C150AC99
    7 V2=0.5*(VL(M,N)+V(M,N))                                           C160AC99
      IF(M-MM) 8,9,9                                                    C170AC99
    8 V3=0.5*(VL(M+1,N)+V(M+1,N))                                       C180AC99
      GO TO 10                                                          C190AC99
    9 V3=V00(N)                                                         C200AC99
   10 Q2=SQRT(U2**2+V2**2)                                              C210AC99
      ALPHA=0.5                                                         C220AC99
      CALL  AC99L(N)                                                    C230AC99
      GO TO 18                                                          C240AC99
   11 XI=XI-0.5*XINC                                                    C250AC99
```

```
      U1=0.5*(UL(M,N)+UL(M,N-1))                                        C260AC99
      U3=0.5*(U(M,N)+U(M,N-1))                                          C270AC99
      IF(M-1) 12,12,13                                                  C280AC99
   12 V1=0.5*(V0(N)+V0(N-1))                                            C290AC99
      GO TO 14                                                          C300AC99
   13 V1=0.25*(VL(M-1,N)+VL(M-1,N-1)+V(M-1,N)+V(M-1,N-1))               C310AC99
   14 IF(M-MM) 15,16,16                                                 C320AC99
   15 V3=0.25*(VL(M+1,N)+VL(M+1,N-1)+V(M+1,N)+V(M+1,N-1))               C330AC99
      GO TO 17                                                          C340AC99
   16 V3=0.5*(V00(N)+V00(N-1))                                          C350AC99
   17 NU1=0.5*(UL(M,N-1)+U(M,N-1))                                      C360AC99
      NU2=0.5*(UL(M,N)+U(M,N))                                          C370AC99
   18 RETURN                                                            C380AC99
      END                                                               C390AC99
```

```
      SUBROUTINE AC99D(M,N,IL,ILP1)                                     D000AC99
C          THIS SUBROUTINE CALCULATES THE LAMINAR OR                    D   AC99
C          TURBULENT VISCOSITY TERMS                                    D   AC99
C                - MELLOR AND GIBSON VISCOSITY MODEL                    D   AC99
      INTEGER OMEGA                                                     D001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                          D002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,            D003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),           D004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),V0(50),V00(50),                 D005AC99
     3XL,Y0,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                        D006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                          D007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                           D008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,           D009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,               D010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,         D011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                               D012AC99
     4DSTAR(10),C,D,E1,E2,E3                                            D013AC99
```

```
      DSTB=DSTAR(M)                                                    D020AC99
      IF(N-LOGPT(M)) 1,1,4                                             D030AC99
    1 GO TO (2,3),LT                                                   D040AC99
    2 NU1=AC99E(0.0,DSTB,LT,ILP1)                                      D050AC99
      NUL1=AC99E(0.0,DSTB,LT,IL)                                       D060AC99
      GO TO 7                                                          D070AC99
    3 Q=SQRT(U2**2+V2**2)                                              D080AC99
      ZETA=KAPPA**2*(XI-0.5*XINC)*Q/EPS                                D090AC99
      NU1=AC99E(ZETA,DSTB,LT,ILP1)                                     D100AC99
      IF(ZETA.GT.DSTB) ILP1=2                                          D110AC99
      Q=SQRT(UL2**2+VL2**2)                                            D120AC99
      ZETA=KAPPA**2*(XI-0.5*XINC)*Q/EPS                                D130AC99
      NUL1=AC99E(ZETA,DSTB,LT,IL)                                      D140AC99
      IF(ZETA.GT.DSTB) IL=2                                            D150AC99
      GO TO 7                                                          D160AC99
    4 GO TO (5,6),LT                                                   D170AC99
    5 NU1=AC99E(0.0,DSTB,LT,ILP1)                                      D180AC99
      NUL1=AC99E(0.0,DSTB,LT,IL)                                       D190AC99
      GO TO 7                                                          D200AC99
    6 DQ=SQRT(U2**2+V2**2)-SQRT(U1**2+V1**2)                           D210AC99
      DQ=ABS(DQ)                                                       D220AC99
      ZETA=(KAPPA*(XI-0.5*XINC))**2*DQ/XINC                            D230AC99
      NU1=AC99E(ZETA,DSTB,LT,ILP1)                                     D240AC99
      IF(ZETA.GT.DSTB) ILP1=2                                          D250AC99
      DQ=SQRT(UL2**2+VL2**2)-SQRT(UL1**2+VL1**2)                       D260AC99
      DQ=ABS(DQ)                                                       D270AC99
      ZETA=(KAPPA*(XI-0.5*XINC))**2*DQ/XINC                            D280AC99
      NUL1=AC99E(ZETA,DSTB,LT,IL)                                      D290AC99
      IF(ZETA.GT.DSTB) IL=2                                            D300AC99
    7 GO TO (8,9),LT                                                   D310AC99
    8 NU2=AC99E(0.0,DSTB,LT,ILP1)                                      D320AC99
      NUL2=AC99E(0.0,DSTB,LT,IL)                                       D330AC99
      GO TO 10                                                         D340AC99
    9 DQ=SQRT(U3**2+V3**2)-SQRT(U2**2+V2**2)                           D350AC99
      DQ=ABS(DQ)                                                       D360AC99
      ZETA=(KAPPA*(XI+0.5*XINC))**2*DQ/XINC                            D370AC99
```

```
      NU2=AC99E(ZETA,DSTB,LT,ILP1)                                      D380AC99
      IF(ZETA.GT.DSTB) ILP1=2                                           D390AC99
      DQ=SQRT(UL3**2+VL3**2)-SQRT(UL2**2+VL2**2)                        D400AC99
      DQ=ABS(DQ)                                                        D410AC99
      ZETA=(KAPPA*(XI+0.5*XINC))**2*DQ/XINC                             D420AC99
      NUL2=AC99E(ZETA,DSTB,LT,IL)                                       D430AC99
      IF(ZETA.GT.DSTB) IL=2                                             D440AC99
   10 RETURN                                                            D450AC99
      END                                                               D460AC99
```

```
      FUNCTION AC99E(ZETA,DSTAR,LT,II)                                  E000AC99
C         THIS FUNCTION PROVIDES THE LAMINAR OR TURBULENT                E   AC99
C         VISCOSITY FUNCTION                                             E   AC99
C                        - MELLOR AND GIBSON VISCOSITY MODEL             E   AC99
      GO TO (1,2),LT                                                    E020AC99
    1 AC99E=DSTAR                                                       E030AC99
      RETURN                                                            E040AC99
    2 GO TO (3,5),II                                                    E050AC99
    3 IF(ZETA-DSTAR) 4,4,5                                              E060AC99
    4 AC99E=ZETA                                                        E070AC99
      RETURN                                                            E080AC99
    5 AC99E=DSTAR                                                       E090AC99
      RETURN                                                            E100AC99
      END                                                               E110AC99
```

```
      SUBROUTINE AC99F(M,N)                                                  F000AC99
C          THIS SUBROUTINE SETS UP THE COEFFICIENTS TO THE                  F    AC99
C          'Z-EQUATIONS' AT THE FIRST POINT IN THE                          F    AC99
C          'LOG-REGION'                                                     F    AC99
      INTEGER OMEGA                                                         F001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                              F002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,                F003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),               F004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),V0(50),V00(50),                     F005AC99
     3XL,Y0,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                            F006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                              F007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                               F008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,               F009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,                   F010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,             F011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                                   F012AC99
     4DSTAR(10),C,D,E1,E2,E3                                                F013AC99
      GO TO (1,2),LT                                                        F015AC99
    1 E3=1.0/N                                                              F017AC99
      GO TO 3                                                               F019AC99
    2 E3=EPS*(N-0.5)                                                        F020AC99
    3 A1(1)=Q2/S+C/(2.0*D*EPS*XI)                                           F030AC99
     1     +(NU2+NU1/E3)/(2.0*D*XINC**2)                                    F040AC99
      A2(1)=-NU2/(2.0*D*XINC**2)                                            F050AC99
      A4(1)=UL2*Q2/S-C*UL2/(2.0*D*EPS*XI)                                   F060AC99
     1     +P1B+(NUL2*(UL3-UL2)-NUL1*UL2/E3)/(2.0*D*XINC**2)                F070AC99
      A5(1)=VL2*Q2/S-C*VL2/(2.0*D*EPS*XI)                                   F080AC99
     1     +P2B+(NUL2*(VL3-VL2)-NUL1*VL2/E3)/(2.0*D*XINC**2)                F090AC99
      RETURN                                                                F100AC99
      END                                                                   F110AC99
```

```
      SUBROUTINE AC99G(NEQ,M,N)                                          G000AC99
C          THIS SUBROUTINE SETS UP THE COEFFICIENTS TO THE               G   AC99
C          LINEAR ALGEBRAIC EQUATIONS AT THE GENERAL POINT               G   AC99
      INTEGER OMEGA                                                      G001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                           G002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,             G003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),            G004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),VO(50),VOO(50),                  G005AC99
     3XL,YO,F,G,H,AT,PT,NSTEP,NSMAX,LFREQ,MFREQ,                         G006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                           G007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                            G008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,            G009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,                G010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,          G011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                                G012AC99
     4DSTAR(10),C,D,E1,E2,E3                                             G013AC99
      A1(NEQ)=PHI2*PSI3*Q2/S+PSI1*PSI5*(NU1+NU2)/(D*XINC**2)             G020AC99
      A2(NEQ)= PSI2*PSI3*Q3/(2.0*S)+C*PSI1*PSI4/(2.0*D*XINC)             G030AC99
     1        -PSI1*PSI5*NU2/(D*XINC**2)                                  G040AC99
      A3(NEQ-1)= PSI2*PSI3*Q1/(2.0*S)-C*PSI1*PSI4/(2.0*D*XINC)           G050AC99
     1         -PSI1*PSI5*NU1/(D*XINC**2)                                 G060AC99
      A4(NEQ)=-PSI2*Q3*(PHI3*U3-UL3)/(2.0*S)-PHI2*Q2*(PHI3*U2-UL2)/S     G070AC99
     1        -PSI2*Q1*(PHI3*U1-UL1)/(2.0*S)                              G080AC99
     2        -C*(PHI1*(UL3-UL1)+PSI1*PHI4*(U3-U1))/(2.0*D*XINC)          G090AC99
     3        +P1B*(PHI1*(NUL2*(UL3-UL2)-NUL1*(UL2-UL1))                  G100AC99
     4        +PSI1*PHI5*(NU2*(U3-U2)-NU1*(U2-U1)))/(D*XINC**2)           G110AC99
      A5(NEQ)=-PSI2*Q3*(PHI3*V3-VL3)/(2.0*S)-PHI2*Q2*(PHI3*V2-VL2)/S     G120AC99
     1        -PSI2*Q1*(PHI3*V1-VL1)/(2.0*S)                              G130AC99
     2        -C*(PHI1*(VL3-VL1)+PSI1*PHI4*(V3-V1))/(2.0*D*XINC)          G140AC99
     3        +P2B*(PHI1*(NUL2*(VL3-VL2)-NUL1*(VL2-VL1))                  G150AC99
     4        +PSI1*PHI5*(NU2*(V3-V2)-NU1*(V2-V1)))/(D*XINC**2)           G160AC99
      IF(N-NMINC) 2,1,2                                                  G180AC99
    1 N1=NMINC-OMEGA                                                     G190AC99
      A4(NEQ)=A4(NEQ)+A3(NEQ-1)*(U(M,N-1)-U(M,N1))                       G200AC99
      A5(NEQ)=A5(NEQ)+A3(NEQ-1)*(V(M,N-1)-V(M,N1))                       G210AC99
    2 IF(N-NOPTS(M)+1) 4,3,3                                             G220AC99
```

```
    3 A4(NEQ)=A4(NEQ)-A2(NEQ)*U(M,N+1)                                  G230AC99
      A5(NEQ)=A5(NEQ)-A2(NEQ)*V(M,N+1)                                  G240AC99
    4 RETURN                                                            G250AC99
      END                                                               G260AC99


      SUBROUTINE AC99H                                                  H000AC99
      RETURN                                                            H010AC99
      END                                                               H020AC99


      FUNCTION AC99I(M,N,GAMMA,VEL,MM)                                  I000AC99
C         THIS FUNCTION INTERPOLATES VELOCITY COMPONENTS                I   AC99
      DIMENSION VEL(10,50)                                              I020AC99
      IF(MM-1) 1,1,2                                                    I030AC99
    1 AC99I=VEL(M,N)                                                    I040AC99
      RETURN                                                            I050AC99
    2 IF(M-1) 3,3,4                                                     I060AC99
    3 I=2                                                               I070AC99
      AG=GAMMA-1.0                                                      I080AC99
      GO TO 7                                                           I090AC99
    4 IF(M-MM) 6,5,5                                                    I100AC99
    5 I=MM-1                                                            I110AC99
      AG=GAMMA+1.0                                                      I120AC99
      GO TO 7                                                           I130AC99
    6 I=M                                                               I140AC99
      AG=GAMMA                                                          I150AC99
    7 CONTINUE                                                          I160AC99
      AA=(VEL(I+1,N)-2.0*VEL(I,N)+VEL(I-1,N))/2.0                       I170AC99
      BB=(VEL(I+1,N)-VEL(I-1,N))/2.0                                    I180AC99
```

```
      CC=VEL(I,N)                                                    I190AC99
      AC99I=AA*AG**2+BB*AG+CC                                        I200AC99
      RETURN                                                         I210AC99
      END                                                            I220AC99


      FUNCTION AC99J(M,BETA,PRM,MM)                                  J000AC99
C         THIS FUNCTION INTEPOLATES PARAMETERS                       J   AC99
      DIMENSION PRM(10)                                              J020AC99
      IF(MM-1) 1,1,2                                                 J030AC99
    1 AC99J=PRM(M)                                                   J040AC99
      RETURN                                                         J050AC99
    2 IF(M-1) 3,3,4                                                  J060AC99
    3 I=2                                                            J070AC99
      AG=BETA-1.0                                                    J080AC99
      GO TO 7                                                        J090AC99
    4 IF(M-MM) 6,5,5                                                 J100AC99
    5 I=MM-1                                                         J110AC99
      AG=BETA+1.0                                                    J120AC99
      GO TO 7                                                        J130AC99
    6 I=M                                                            J140AC99
      AG=BETA                                                        J150AC99
    7 CONTINUE                                                       J160AC99
      AA=(PRM(I+1)-2.0*PRM(I)+PRM(I-1))/2.0                          J170AC99
      BB=(PRM(I+1)-PRM(I-1))/2.0                                     J180AC99
      CC=PRM(I)                                                      J190AC99
      AC99J=AA*AG**2+BB*AG+CC                                        J200AC99
      RETURN                                                         J210AC99
      END                                                            J220AC99
```

```
      SUBROUTINE AC99L(N)                                                L000AC99
C         THIS SUBROUTINE FITS THE 'LOG LAW OF THE WALL'                  L   AC99
      INTEGER OMEGA                                                      L001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                           L002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,             L003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),            L004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),V0(50),V00(50),                  L005AC99
     3XL,Y0,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                         L006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                           L007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                            L008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,            L009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,                L010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,          L011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                                L012AC99
     4DSTAR(10),C,D,E1,E2,E3                                             L013AC99
      GO TO (1,2),LT                                                     L020AC99
    1 EPS=1.0                                                            L030AC99
      RETURN                                                             L040AC99
    2 MAXIT=20                                                           L050AC99
      TOL1=0.00001                                                       L060AC99
      B=(AT*(XL+ALPHA*F)+BT)*H*N*KAPPAL*Q2                               L070AC99
      B=ALOG(B/(OMEGA*NU))+KAPPAL*A                                      L080AC99
      EPS1=6.0                                                           L090AC99
      DO 4 I=1,MAXIT                                                     L100AC99
      EPS=EPS1*(1.0-(EPS1+ALOG(EPS1)-B)/(1.0+EPS1))                      L110AC99
      IF((EPS-EPS1)**2-TOL1) 5,5,3                                       L120AC99
    3 EPS1=EPS                                                           L130AC99
    4 CONTINUE                                                           L140AC99
      WRITE(MOUT,6) Q2,N,ALPHA,EPS,EPS1                                  L150AC99
    5 CONTINUE                                                           L160AC99
      RETURN                                                             L165AC99
    6 FORMAT(1HO,5X,3HEL1,5X,F12.6,I6,3F12.6)                            L170AC99
      END                                                                L180AC99
```

```
      SUBROUTINE AC99M(M,N)                                                  M000AC99
C          THIS SUBROUTINE SETS UP ALL THE NECESSARY                          M   AC99
C          QUANTITIES FOR THE FINITE DIFFERANCE APPROXIMATION                 M   AC99
C          TO THE MOMENTUM EQUATION                                           M   AC99
      INTEGER OMEGA                                                          M001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                               M002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,                M003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),               M004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),V0(50),V00(50),                     M005AC99
     3XL,YO,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                            M006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                              M007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                               M008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,                M009AC99
     1UL1.UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,                    M010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,              M011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                                   M012AC99
     4DSTAR(10),C,D,E1,E2,E3                                                M013AC99
      IF(N-NMINC) 1,2,2                                                      M020AC99
    1 XI=(N*H)/OMEGA                                                         M030AC99
      XINC=H/OMEGA                                                           M040AC99
      GO TO 3                                                                M050AC99
    2 XI=(N-NMINC+LAMDA)*H                                                   M060AC99
      XINC=H                                                                 M070AC99
    3 IF(N-LOGPT(M)) 4,4,5                                                   M080AC99
    4 U1=0.0                                                                 M090AC99
      V1=0.0                                                                 M100AC99
      UL1=0.0                                                                M110AC99
      VL1=0.0                                                                M120AC99
      Q1=0.0                                                                 M130AC99
      GO TO 9                                                                M140AC99
    5 IF(N-NMINC) 7,6,7                                                      M150AC99
    6 NM1=N-OMEGA                                                            M160AC99
      GO TO 8                                                                M170AC99
    7 NM1=N-1                                                                M180AC99
    8 UL1=AC99I(M,NM1,GAMMA,UL,MM)                                           M190AC99
      VL1=AC99I(M,NM1,GAMMA,VL,MM)                                           M200AC99
```

```
      U1=U(M,NM1)                                                        M210AC99
      V1=V(M,NM1)                                                        M220AC99
      Q1=SQRT((PHI1*UL1+PSI1*U1)**2+(PHI1*VL1+PSI1*V1)**2)               M230AC99
    9 UL2=AC99I(M,N,GAMMA,UL,MM)                                         M240AC99
      VL2=AC99I(M,N,GAMMA,VL,MM)                                         M250AC99
      U2=U(M,N)                                                          M260AC99
      V2=V(M,N)                                                          M270AC99
      Q2=SQRT((PHI1*UL2+PSI1*U2)**2+(PHI1*VL2+PSI1*V2)**2)               M280AC99
      UL3=AC99I(M,N+1,GAMMA,UL,MM)                                       M290AC99
      VL3=AC99I(M,N+1,GAMMA,VL,MM)                                       M300AC99
      U3=U(M,N+1)                                                        M310AC99
      V3=V(M,N+1)                                                        M320AC99
      Q3=SQRT((PHI1*UL3+PSI1*U3)**2+(PHI1*VL3+PSI1*V3)**2)               M330AC99
      IF(N-LOGPT(M)) 10,10,11                                            M340AC99
   10 CALL AC99L(N)                                                      M350AC99
   11 C=-AT*XI*(PHI1*UL2+PSI1*U2)+AC99I(M,N,BETA,W,MM)                   M390AC99
      D=AT*(XL+ALPHA*F)+BT                                               M400AC99
      P1B=AC99J(M,BETA,P1,MM)                                            M410AC99
      P2B=AC99J(M,BETA,P2,MM)                                            M420AC99
      RETURN                                                             M430AC99
      END                                                                M440AC99
```

```
      SUBROUTINE AC990                                                  0000AC99
C          THIS SUBROUTINE SETS UP THE OUTPUT QUANTIES                  0   AC99
      INTEGER OMEGA                                                     0001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                          0002AC99
      EXTERNAL AC99Y                                                    0002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,  -         0003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),           0004AC99
     2R1(10),R2(10),LOGPT(10),NOPTS(10),VC(50),V00(50),                 0005AC99
     3XL,YO,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                        0006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                          0007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                           0008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,           0009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,               0010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,         0011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                               0012AC99
     4D,D1,D2,T11,T12,T21,T22,HAPE,RT11,QT,CF,CFX,CF1,CF2,PHI           0013AC99
      H1=H*(AT*(XL+F)+BT)                                               0020AC99
      CALL AC99D(1,1.0)                                                 0023AC99
      DO 7 M=1,MM                                                       0026AC99
      ALPHA=1.0                                                         0030AC99
      N=LOGPT(M)                                                        0040AC99
      U2=U(M,N)                                                         0050AC99
      V2=V(M,N)                                                         0060AC99
      Q2=SQRT(U2**2+V2**2)                                              0070AC99
      CALL AC99L(N)                                                     0080AC99
      C=H1*BLT(M)                                                       0100AC99
      U1=U(M,NN-4)                                                      0110AC99
      V1=V(M,NN-4)                                                      0120AC99
      Q1=SQRT(U1**2+V1**2)                                              0130AC99
      D1 = AC99T(AC99Y,M,NMINC,NOPTS(M),OMEGA,H1,1,0,0,1.0)             0140AC99
      D2 =-AC99T(AC99Y,M,NMINC,NOPTS(M),OMEGA,H1,0,0,1,1.0)             0150AC99
      T11= AC99T(AC99Y,M,NMINC,NOPTS(M),OMEGA,H1,1,1,0,1.0)             0160AC99
      T12= AC99T(AC99Y,M,NMINC,NOPTS(M),OMEGA,H1,1,0,1,1.0)             0170AC99
      T21=-AC99T(AC99Y,M,NMINC,NOPTS(M),OMEGA,H1,0,1,1,1.0)             0180AC99
      T22=-AC99T(AC99Y,M,NMINC,NOPTS(M),OMEGA,H1,0,0,2,1.0)             0190AC99
      HAPE=D1/T11                                                       0200AC99
```

```
      RT11=Q1*T11/NU                                                    0210AC99
      QT=KAPPAL*Q2/EPS                                                  0220AC99
      IF(IT.EQ.1) QT=0.0                                                0225AC99
      CF=2.0*(QT/Q1)**2                                                 0230AC99
      CFX=CF*U2/Q2                                                      0240AC99
      CFY=CF*V2/Q2                                                      0250AC99
      CF1=CFX*U1/Q1+CFY*V1/Q1                                           0260AC99
      CF2=CFX*V1/Q1-CFY*U1/Q1                                           0270AC99
      PH1=(V1*U2-V2*U1)/(U1*U2+V1*V2)                                   0280AC99
      II=1                                                              0285AC99
      IF(NSTEP-(NSTEP/LFREQ)*LFREQ) 6,1,6                               0290AC99
    1 MM1=M-1                                                           0292AC99
      IF(MM1-(MM1/MFREQ)*MFREQ) 6,2,6                                   0294AC99
    2 II=2                                                              0296AC99
      DO 3 N=1,NMINC                                                    0300AC99
    3 A1(N)=N*H1/OMEGA                                                  0310AC99
      DO 4 N=NMINC,NN                                                   0320AC99
    4 A1(N)=(N-NMINC+LAMDA)*H1                                          0330AC99
      DO 5 N=1,NN                                                       0340AC99
      U2=U(M,N)                                                         0350AC99
      V2=V(M,N)                                                         0360AC99
      A2(N)=V2/U2                                                       0370AC99
      A3(N)=(U1*U2+V1*V2)/Q1**2                                         0380AC99
      A4(N)=(V1*U2-U1*V2)/Q1**2                                         0390AC99
      A5(N)=(V1*U2-V2*U1)/(U1*U2+V1*V2)                                 0400AC99
    5 CONTINUE                                                          0410AC99
    6 CALL AC99P(M,II)                                                  0420AC99
    7 CONTINUE                                                          0430AC99
      RETURN                                                            0440AC99
      END                                                               0450AC99
```

```
      SUBROUTINE AC99P(M,II)                                            P000AC99
      INTEGER OMEGA                                                     P001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                          P002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,            P003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),           P004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),V0(50),V00(50),                 P005AC99
     3XL,Y0,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                        P006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                          P007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                           P008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,           P009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,               P010AC99
     2NUL1,NUL2,NU1,NU2,X1,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,         P011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                               P012AC99
     4D,D1,D2,T11,T12,T21,T22,HAPE,RT11,QT,CF,CFX,CF1,CF2,PHI           P013AC99
      IF((II-1)*(LFREQ+MFREQ-2)) 2,2,1                                  P020AC99
    1 WRITE(MOUT,70)                                                    P030AC99
    2 XLP1=XL+F                                                         P032AC99
      H1=H*(AT*XLP1+BT)                                                 P034AC99
      YM=Y0+(M-1)*G                                                     P036AC99
      WRITE(MOUT,71) NSTEP,M,XLP1,YM,H1,ITN,LOGPT(M),EPS                P040AC99
      WRITE(MOUT,72) D,HAPE,RT11,D1,D2,T11,T12,T21,T22                  P050AC99
      WRITE(MOUT,73) QT,CF,CF1,CF2,CFX,PHI                              P060AC99
      N=LOGPT(M)                                                        P070AC99
      CALL AC99S(M,N)                                                   P080AC99
      WRITE(MOUT,76) ALPHA,BETA,GAMMA,S                                 P090AC99
      CALL AC99S(M,NN)                                                  P100AC99
      WRITE(MOUT,76) ALPHA,BETA,GAMMA,S                                 P110AC99
      GO TO (5,3),II                                                    P120AC99
    3 WRITE(MOUT,74)                                                    P130AC99
      DO 4 N=1,NN                                                       P140AC99
      WRITE(MOUT,75)  A1(N),U(M,N),V(M,N),W(M,N),A2(N),                 P150AC99
     1                A3(N),A4(N),A5(N)                                P160AC99
    4 CONTINUE                                                          P170AC99
      WRITE(MOUT,70)                                                    P190AC99
    5 RETURN                                                            P200AC99
   70 FORMAT(1H1)                                                       P210AC99
```

```
 71 FORMAT(1HO,2X,4HFACE,I4,2X,8H,SECTION,I4,6X,2HX=,F11.6,2X,3H,Y=,   P220AC99
   1F11.6/1H ,100X,1H*/1H ,6X,31HTHE Z INCREMENT WAS ADJUSTED TO,     P230AC99
   2F12.9,30H AND THE SOLUTION CONVERGED IN,I3,16H ITERATIONS WITH,   P240AC99
   34H N =,I3,7H AND C=,F10.6/1H+,110X,1H-)                           P245AC99
 72 FORMAT(1HO,6X,40H THE PROFILE PARAMETERS ARE AS FOLLOWS -,12X,2HS=P250AC99
   1,F10.6,5X,2HH=,F9.5,3X,5HR    =,F10.3/1H+,58X,1HC/1H ,90X,3HO11/1H+P260AC99
   2,90X,1H-/1H ,8X,1H*,15X,1H*/1H ,7X,3HS1=,F10.6,3X,3HS2=,F10.6,3X, P270AC99
   34HO11=,F10.6,3X,4HO12=,F10.6,3X,4HO21=,F10.6,3X,4HO22=,F10.6/1H+, P280AC99
   47X,1HC,15X,1HC,15X,1H-,16X,1H-,16X,1H-,16X,1H-)                   P290AC99
 73 FORMAT(1HO,7X,3HQ =,F10.7,3X,3HC =,F10.7,3X,4HC  =,F10.7,3X,4HC  =P300AC99
   1,F10.7,3X,4HC  =,F10.7,5X,2HO=,F10.6/1H+,92X,1H//1H ,8X,1HT,15X,  P310AC99
   21HF.15X,2HF1,15X,2HF2,15X,2HFX/1H+,8X,1H()                        P320AC99
 74 FORMAT(1HO,3X,8HDISTANCE,8X,31HRECTANGULAR VELOCITY COMPONENTS,7X,P330AC99
   17HTANGENT,11X,12H'STREAMWISE',10X,7HTANGENT/1H ,3X,8HFROM THE,47X,P340AC99
   25HANGLE,10X,17HVELOCITY PROFILES,8X,5HANGLE/1H ,5X,4HWALL,12X,1HU,P350AC99
   312X.1HV,12X,1HW,11X,3HQ-U,12X,2HU1,11X,2HV1,10X,4HQ-QS/)           P360AC99
 75 FORMAT(3X,F10.6,2X,3(3X,F10.4),3X,F9.6,2X,2(3X,F10.6),3X,F9.6)    P370AC99
 76 FORMAT(1H ,4F12.6)                                                P380AC99
    END                                                               P390AC99


    SUBROUTINE AC99Q                                                  Q000AC99
C       THIS SUBROUTINE SETS UP THE FREESTREAM                        Q   AC99
C       BOUNDARY CONDITION FOR THE THREE-                             Q   AC99
C       DIMENSIONAL BOUNDARY LAYER                                    Q   AC99
    INTEGER OMEGA                                                     Q001AC99
    REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                          Q002AC99
    COMMON  MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,             Q003AC99
   1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),           Q004AC99
   2P1(10),P2(10),LOGPT(10),NOPTS(10),V0(50),V00(50),                 Q005AC99
   3XL,YO,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                        Q006AC99
   4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                          Q007AC99
   5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                           Q008AC99
```

```
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,          Q009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,              Q010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,        Q011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                              Q012AC99
     4DSTAR(10),C,D,E1,E2,E3                                           Q013AC99
      DO 1 M=1,MM                                                      Q020AC99
      U(M,NN)=AC99U(XL+F,YO+(M-1)*G)                                   Q030AC99
      V(M,NN)=AC99V(XL+F,YO+(M-1)*G)                                   Q040AC99
      P1(M)= (U(M,NN)+UL(M,NN))*(U(M,NN)-UL(M,NN))/(2.0*F)             Q050AC99
     1       +(V(M,NN)+VL(M,NN))*(V(M,NN)-VL(M,NN))/(2.0*F)            Q060AC99
      P2(M)= (U(M,NN)+UL(M,NN))*(AC99U(XL+0.5*F,YO+M*G)                Q070AC99
     1                          -AC99U(XL+0.5*F,YO+(M-2)*G))/(4.0*G)   Q080AC99
     2       +(V(M,NN)+VL(M,NN))*(AC99V(XL+0.5*F,YO+M*G)               Q090AC99
     3                          -AC99V(XL+0.5*F,YO+(M-2)*G))/(4.0*G)   Q100AC99
    1 CONTINUE                                                         Q110AC99
      VO(NN)=AC99V(XL+0.5*F,YO-G)                                      Q120AC99
      VOO(NN)=AC99V(XL+0.5*F,YO+MM*G)                                  Q130AC99
      RETURN                                                           Q140AC99
      END                                                              Q150AC99
```

```
      SUBROUTINE AC99R(M)                                              R000AC99
C          THIS SUBROUTINE SETS UP LAMINAR OR TURBULENT                 R   AC99
C          VISCOSITY FUNCTION PARAMETERS                                R   AC99
C                    - MELLOR AND GIBSON VISCOSITY MODEL                R   AC99
      INTEGER OMEGA                                                    R001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                         R002AC99
      EXTERNAL AC99Y                                                   R002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,           R003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),          R004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),VO(50),VOO(50),                R005AC99
     3XL,YO,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                       R006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                         R007AC99
```

```
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                          R008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,          R009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,              R010AC99
     2NUL1,NUL2,NU1,NU2,X1,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,        R011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                              R012AC99
     4DSTAR(10),C,D,E1,E2,E3                                           R013AC99
      AL=0.5                                                           R017AC99
      GO TO (1,2),LT                                                   R020AC99
    1 DSTAR(M)=NU/(AT*(XL+AL*F)+BT)                                    R030AC99
      RETURN                                                           R040AC99
    2 CALL  AC99S(M,NN)                                                R050AC99
      U1=AC99I(M,NN,GAMMA,UL,MM)                                       R060AC99
      V1=AC99I(M,NN,GAMMA,VL,MM)                                       R070AC99
      U3=U(M,NN)                                                       R080AC99
      V3=V(M,NN)                                                       R090AC99
      Q1=SQRT(U1**2+V1**2)                                             R100AC99
      Q3=SQRT(U3**2+V3**2)                                             R110AC99
      DQ=Q3-Q1                                                         R120AC99
      Q2=0.5*(Q3+Q1)                                                   R130AC99
      H1=H*(AT*(XL+ALPHA*F)+BT)                                        R140AC99
      Q1=Q2                                                            R150AC99
      U1=0.5*(U3+U1)                                                   R160AC99
      V1=0.5*(V3+V1)                                                   R170AC99
      TH11=AC99T(AC99Y,M,NMINC,NOPTS(M),OMEGA,H1,1,1,0,ALPHA)          R180AC99
      AA=1.0E 4*TH11*(DQ/S)/Q2                                         R190AC99
      K=0.016+0.00015*AA                                               R200AC99
      IF(K-0.007) 7,7,8                                                R210AC99
    7 K=0.007                                                          R220AC99
    8 CONTINUE                                                         R230AC99
      DSTAR(M)=K*Q2*                                                   R240AC99
     1      AC99T(AC99Y,M,NMINC,NOPTS(M),OMEGA,H,1,0,0,ALPHA)          R250AC99
      RETURN                                                           R260AC99
      END                                                              R270AC99
```

```
      SUBROUTINE AC99S(M,N)                                              S000AC99
C          THIS SUBROUTINE FITS 'STREAMLINES' BETWEEN POINTS             S   AC99
C          ON THE DOWNSTREAM SOLUTION FACE AND THE UPSTREAM              S   AC99
C          SOLUTION FACE                                                 S   AC99
      INTEGER OMEGA                                                      S001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                           S002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,             S003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),            S004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),V0(50),V00(50),                  S005AC99
     3XL,Y0,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                         S006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                           S007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                            S008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,            S009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,                S010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,          S011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                                S012AC99
     4DSTAR(10),C,D,E1,E2,E3                                             S013AC99
      TOL1=0.001                                                         S020AC99
      TOL2=0.001                                                         S030AC99
      TOL3=0.001                                                         S040AC99
      MAXIT=10                                                           S050AC99
      TSTAR=V(M,N)/U(M,N)                                                S060AC99
      TGAMA=VL(M,N)/UL(M,N)                                              S070AC99
      TG=0.0                                                             S080AC99
      AA=TSTAR+TGAMA                                                     S090AC99
      IF(ABS(AA)-TOL1) 1,1,2                                             S100AC99
    1 GAMMA=0.0                                                          S110AC99
      GO TO 7                                                            S120AC99
```

```
 2 IF(AA) 3,3,4                                                          S130AC99
 3 SGN=-1.0                                                              S140AC99
   GO TO 5                                                               S150AC99
 4 SGN= 1.0                                                              S160AC99
 5 GAMMA=-TSTAR*F/G                                                      S170AC99
   DO 6 I=1,MAXIT                                                        S180AC99
   TGAMA=AC99I(M,N,GAMMA,VL,MM)/AC99I(M,N,GAMMA,UL,MM)                   S190AC99
   T=(1.0-TSTAR*TGAMA)/(TSTAR+TGAMA)                                     S200AC99
   TG=-T+SGN*SQRT(1.0+T**2)                                              S210AC99
   ERROR=GAMMA+TG*F/G                                                    S220AC99
   GAMMA=-TG*F/G                                                         S230AC99
   IF(ABS(ERROR)-TOL2) 7,7,6                                             S240AC99
 6 CONTINUE                                                              S250AC99
   WRITE(MOUT,11) M,N                                                    S260AC99
   CALL EXIT                                                             S270AC99
 7 ALPHA=(2.0+3.0*TG*TSTAR-TG**2)/(4.0*(1.0+TG*TSTAR))                   S280AC99
*  BETA=-F*(TSTAR+TG+2.0*TG**2*TSTAR)/(4.0*G*(1.0+TG*TSTAR))             S290AC99
   AS=F*SQRT(1.0+TG**2)/2.0                                              S300AC99
   BS=(TG-TSTAR)/(1.0+TG*TSTAR)                                          S310AC99
   CS=SQRT(1.0+BS**2)                                                    S320AC99
   IF(ABS(BS)-TOL3) 8,8,9                                                S330AC99
 8 S=2.0*AS                                                              S340AC99
   GO TO 10                                                              S350AC99
 9 S=AS*(ALOG(BS+CS)/BS+CS)                                              S360AC99
10 RETURN                                                                S370AC99
11 FORMAT(1HO,5X,3HES1,5X,216)                                           S380AC99
   END                                                                   S390AC99
```

```
      FUNCTION AC99T(R,M,NMINC,NOPTS,OMEGA,H,I1,I2,I3,A)                T000AC99
C          THIS FUNCTION CALCULATES THE INTEGRAL OF THE                 T   AC99
C          FUNCTION R THROUGH THE BOUNDARY LAYER                        T   AC99
      INTEGER OMEGA                                                     T010AC99
      T1=0.5*(R(M,0,I1,I2,I3,A)+R(M,NMINC,I1,I2,I3,A))                  T020AC99
      IJ=NMINC-1                                                        T030AC99
      DO 1 I=1,IJ                                                       T040AC99
    1 T1=T1+R(M,I,I1,I2,I3,A)                                           T050AC99
      T2=0.5*(R(M,NMINC,I1,I2,I3,A)+R(M,NOPTS,I1,I2,I3,A))              T060AC99
      II=NMINC+1                                                        T070AC99
      IJ=NOPTS-1                                                        T080AC99
      DO 2 I=II,IJ                                                      T090AC99
    2 T2=T2+R(M,I,I1,I2,I3,A)                                           T100AC99
      AC99T=H*(T1/OMEGA+T2)                                             T110AC99
      RETURN                                                            T120AC99
      END                                                               T130AC99


      FUNCTION AC99V(X,Y)                                               V000AC99
      AC99V=0.0                                                         V010AC99
      RETURN                                                            V020AC99
      END                                                               V030AC99
```

```
      SUBROUTINE AC99W(M,N)                                              W000AC99
C          THIS SUBROUTIME CALCULATES W AT ANY POINT                     W   AC99
      INTEGER OMEGA                                                      W001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                           W002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,             W003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),            W004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),V0(50),V00(50),                  W005AC99
     3XL,Y0,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                         W006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                           W007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                            W008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,            W009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,                W010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,          W011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                                W012AC99
     4DSTAR(10),C,D,E1,E2,E3                                             W013AC99
      IF(N-LOGPT(M)) 1,1,2                                               W020AC99
    1 C=AT*U2/(D*EPS)                                                    W030AC99
      GO TO 3                                                            W040AC99
    2 C=(AT*XI/D)*(NU2-NU1)/XINC                                         W050AC99
    3 E=(U3-U1)/F+(V3-V1)/(2.0*G)-C                                      W060AC99
      IF(N-LOGPT(M)) 4,4,6                                               W070AC99
    4 W(M,N)=-(D*XI/(1.0+1.0/EPS))*E                                     W080AC99
      DO 5 I=1,N                                                         W090AC99
    5 W(M,I)=(FLOAT(I)/N)**(1.0+1.0/EPS)*W(M,N)                          W100AC99
      RETURN                                                             W110AC99
    6 W(M,N)=W(M,N-1)-D*E*XINC                                           W120AC99
      RETURN                                                             W130AC99
      END                                                                W140AC99
```

```
      SUBROUTINE AC99X                                                   X000AC99
C          THIS SUBROUTINE DETERMINES THE NUMBER OF                      X   AC99
C          POINTS AND THE LOG POINT AT EACH SECTION                      X   AC99
      INTEGER OMEGA                                                      X001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                           X002AC99
```

```
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,                X003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),              X004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),V0(50),V00(50),                    X005AC99
     3XL,Y0,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                           X006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                             X007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                              X008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,              X009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,                  X010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,            X011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                                  X012AC99
     4DSTAR(10),C,D,E1,E2,E3                                               X013AC99
      CALL AC99B(2,0.0)                                                    X020AC99
      DO 2 M=1,MM                                                          X030AC99
      NOPTS(M)=BLT(M)+NSLAP+1                                              X040AC99
      IF(NOPTS(M)-NN) 2,2,1                                                X050AC99
    1 NOPTS(M)=NN                                                          X060AC99
    2 CONTINUE                                                             X070AC99
      GO TO (3,5),LT                                                       X080AC99
    3 DO 4 M=1,MM                                                          X090AC99
    4 LOGPT(M)=1                                                           X100AC99
      RETURN                                                               X110AC99
    5 N1=NMINC-OMEGA                                                       X120AC99
      DO 6 M=1,MM                                                          X130AC99
      DO 6 N=2,N1                                                          X140AC99
      Q2=SQRT(UL(M,N)**2+VL(M,N)**2)                                       X150AC99
      ALPHA=0.0                                                            X160AC99
      CALL AC99L(N)                                                        X170AC99
      C=N*KAPPAL*H*Q2/(OMEGA*NU*EPS)                                       X180AC99
      IF(C-CMIN) 6,7,7                                                     X190AC99
    6 CONTINUE                                                             X200AC99
      WRITE(MOUT,9) NSTEP,M,C                                              X210AC99
      N=N1                                                                 X220AC99
    7 LOGPT(M)=N                                                           X230AC99
    8 CONTINUE                                                             X240AC99
      RETURN                                                               X250AC99
    9 FORMAT(1H0,5X,3HEX1,5X,2I6,F12.6)                                    X260AC99
      END                                                                  X270AC99
```

```
      FUNCTION AC99Y(M,N,I1,I2,I3,AL)                                    Y000AC99
C          THIS FUNCTION PROVIDES THE ARGUMENTS NECESSARY                 Y   AC99
C          TO CALCULATE THE DISPLACEMENT AND MOMENTUM                     Y   AC99
C          THICKNESSES                                                    Y   AC99
       INTEGER OMEGA                                                     Y001AC99
       REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                          Y002AC99
       COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,            Y003AC99
      1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),           Y004AC99
      2P1(10),P2(10),LOGPT(10),NOPTS(10),V0(50),V00(50),                 Y005AC99
      3XL,Y0,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                        Y006AC99
      4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                          Y007AC99
      5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                           Y008AC99
       COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,           Y009AC99
      1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,               Y010AC99
      2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,         Y011AC99
      3A1(50),A2(50),A3(50),A4(50),A5(50),                               Y012AC99
      4DSTAR(10),C,D,E1,E2,E3                                            Y013AC99
       IF(N) 1,1,2                                                       Y020AC99
    1 UU=0.0                                                             Y030AC99
       VV=0.0                                                            Y040AC99
       GO TO 3                                                           Y050AC99
    2 CONTINUE                                                           Y060AC99
       UAL=(1.0-AL)*UL(M,N)+AL*U(M,N)                                    Y070AC99
       VAL=(1.0-AL)*VL(M,N)+AL*V(M,N)                                    Y075AC99
       UU=(UAL*U1+VAL*V1)/Q1**2                                          Y080AC99
       VV=(UAL*V1-VAL*U1)/Q1**2                                          Y085AC99
    3 CONTINUE                                                           Y090AC99
       AA=1.0                                                            Y100AC99
       IF(I1) 4,5,4                                                      Y110AC99
    4 AA=AA*(1.0-UU)**I1                                                 Y120AC99
    5 IF(I2) 6,7,6                                                       Y130AC99
    6 AA=AA*UU**I2                                                       Y140AC99
    7 IF(I3) 8,9,8                                                       Y150AC99
    8 AA=AA*VV**I3                                                       Y160AC99
    9 AC99Y=AA                                                           Y170AC99
       RETURN                                                            Y180AC99
       END                                                               Y190AC99
```

```
      SUBROUTINE AC99Z(NEQ)                                              Z000AC99
C         THIS SUBROUTINE SOLVES THE LINEAR ALGEBRAIC EQUATIONS           Z   AC99
      INTEGER OMEGA                                                      Z001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                           Z002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,             Z003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),            Z004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),V0(50),V00(50),                  Z005AC99
     3XL,Y0,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                         Z006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                           Z007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                            Z008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,            Z009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,                Z010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,          Z011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                                Z012AC99
     4DSTAR(10),C,D,E1,E2,E3                                             Z013AC99
      DO 1 I=2,NEQ                                                       Z020AC99
      A3D1=A3(I-1)/A1(I-1)                                               Z030AC99
      A1(I)=A1(I)-A2(I-1)*A3D1                                           Z040AC99
      A4(I)=A4(I)-A4(I-1)*A3D1                                           Z050AC99
      A5(I)=A5(I)-A5(I-1)*A3D1                                           Z060AC99
    1 CONTINUE                                                           Z070AC99
      A4(NEQ)=A4(NEQ)/A1(NEQ)                                            Z080AC99
      A5(NEQ)=A5(NEQ)/A1(NEQ)                                            Z090AC99
      DO 2 J=2,NEQ                                                       Z100AC99
      I=NEQ-J+1                                                          Z110AC99
      A4(I)=(A4(I)-A2(I)*A4(I+1))/A1(I)                                  Z120AC99
      A5(I)=(A5(I)-A2(I)*A5(I+1))/A1(I)                                  Z130AC99
    2 CONTINUE                                                           Z140AC99
      RETURN                                                             Z150AC99
      END                                                                Z160AC99
```

APPENDIX A7

PROGRAM DESCRIPTION

Introduction.

The present appendix contains a description of the structure of the computer program listed in Appendix A6 as well as instructions for using the program.

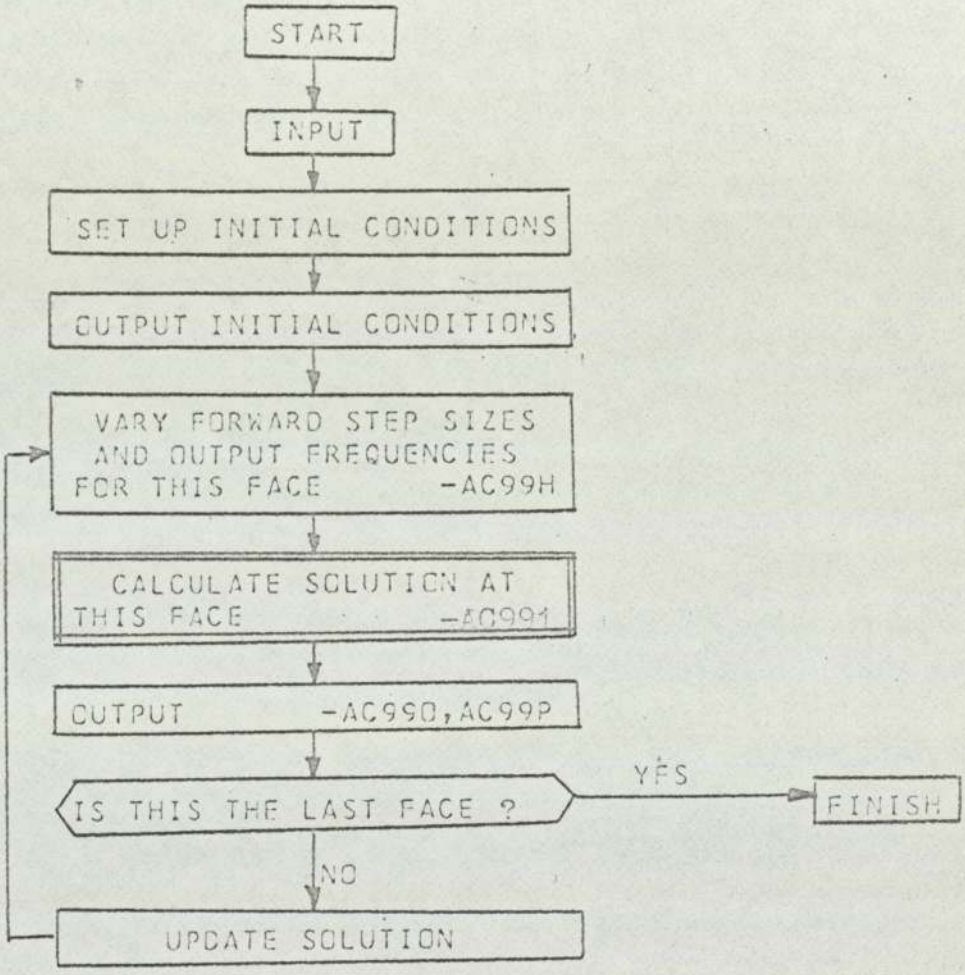
The calculation scheme is based on that presented in detail in Chapter Four and Appendices A1-5 where finite difference approximations have been made to the transformed boundary layer equations (4.3.9-11). We note here that the boundary layer equations were transformed using equation (4.3.1) and that the effective viscosity ν_e has been replaced by ν_e' as given by equation (4.3.8).

The program itself is effectively built up of three major subroutines which determine the overall progress of the calculation. The first subroutine, the 'main program', provides the control in that it reads in the data and calls on another subroutine AC991 to compute velocity profiles at each solution face before proceeding to output the results of the calculation. Subroutine AC991 calculates each solution face by repeatedly calling AC992 which at each pass performs one complete iteration of the boundary layer equations. The main program and subroutine AC991 are thus primarily concerned with the organisation of the calculation while subroutine AC992 contains the basic calculation scheme, although some calculation, such as ammending the side boundary conditions and the grid development between iterations, is contained in subroutine AC991. These three major subroutines call upon numerous others which will be detailed later.

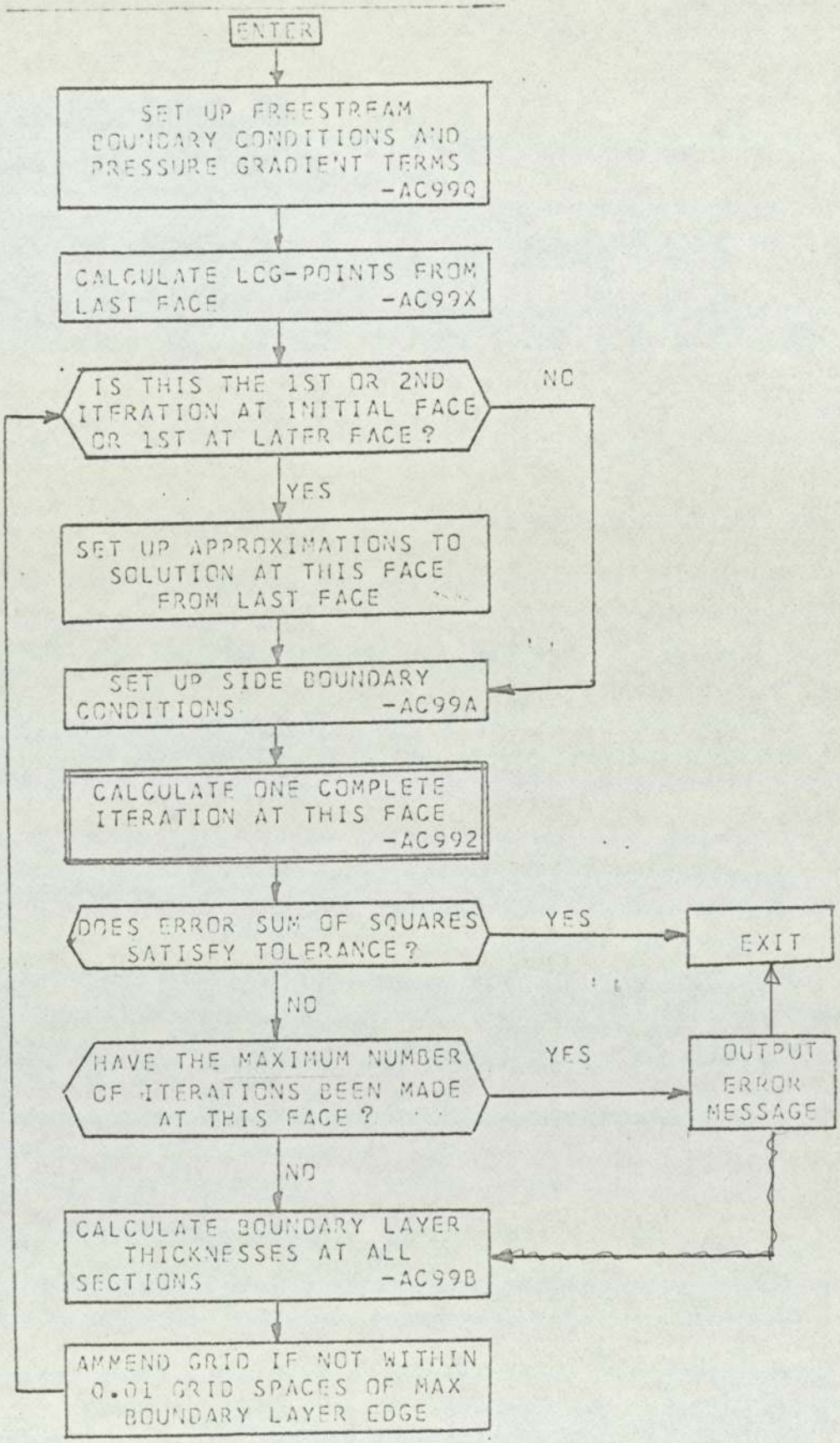
Flow diagrams are included below to give a descriptive account of the progress within these three major subroutines, while following these we will give a more detailed breakdown of the common storage and the subroutines used. Finally an account of how the

program should be used is included together with subroutine and data input for a sample application (that of Hornung and Joubert considered in section 6.6).

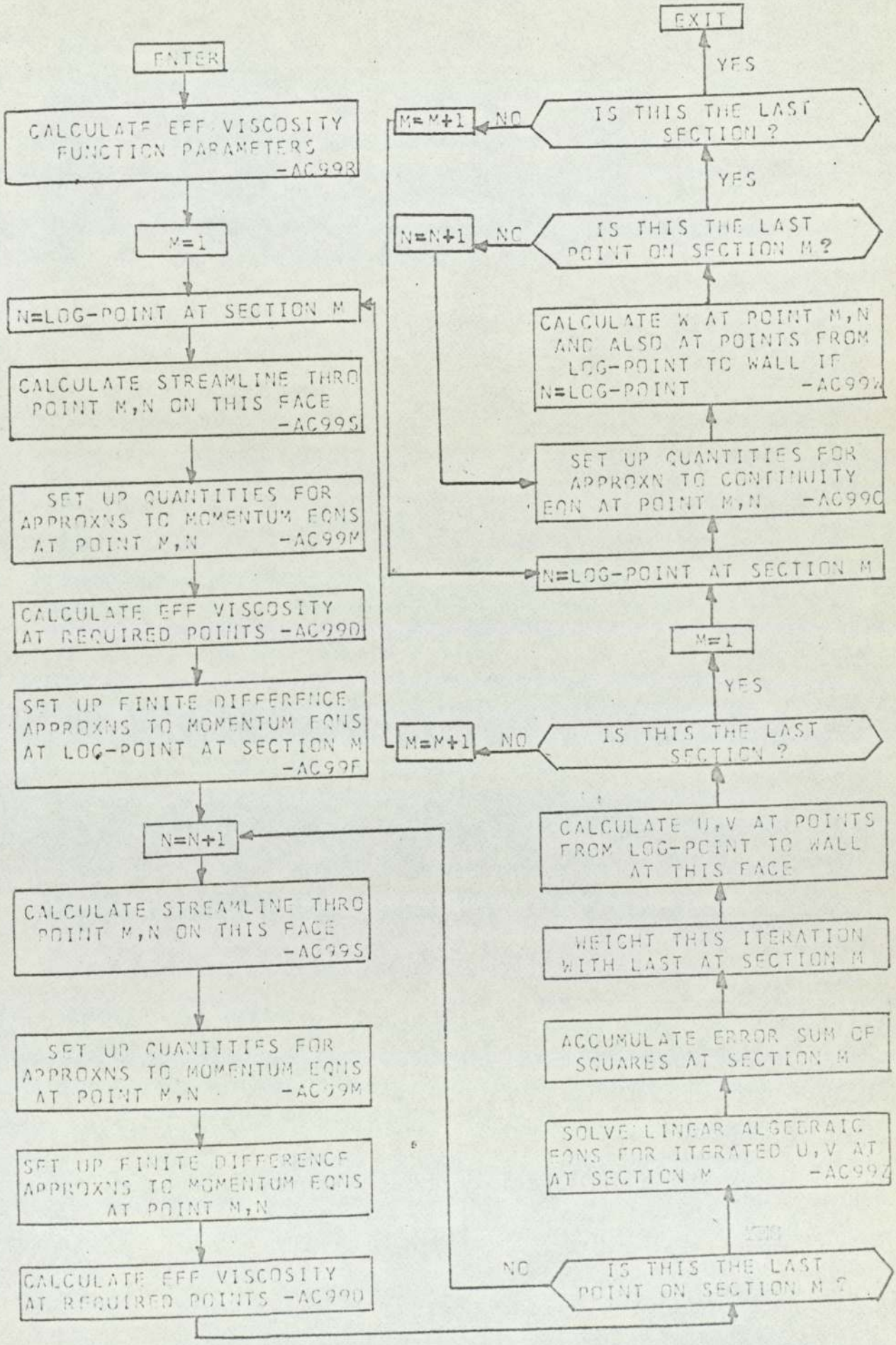
FLOW DIAGRAM - MAIN PROGRAM



FLOW DIAGRAM - SUBROUTINE AC991



FLOW DIAGRAM - SUBROUTINE AC992



Common storage.

The items of data stored in common are listed below with array sizes where these are relevant. The program included will cater with a 10 × 50 grid although obviously it would be very easy to alter the program in this respect to suit any particular need. Reference below to 'this face' is to the solution face currently being calculated, 'last face' to that immediately preceeding. Individual items will be referred to the standard notation used in the remainder of this work.

MIN	Card input unit number
MOUT	Printer output unit number
MM	M
NN	N
LAMDA	λ
OMEGA	ω
NMINC	$\lambda\omega$
NSLAP	minimum number of slack points to be accommodated above boundary layer edge
LT	=1, laminar flow
	=2, turbulent flow
NU	ν
UL,VL	u,v components of velocity at points on last face (each 10 × 50 array)
U,V	u,v components of velocity at points on this face (10 × 50)
W	w component of velocity at points on mid-face (10 × 50)
BLT	δ at each section (10)
P1,P2	pressure terms in x,y momentum equations at each section (10)
LOGPT	the log-point at each section (10)
NOPTS	the number of points at each section (10)
VO,VOO	crossflow velocity profiles at side boundary planes (50)

XL	x_ℓ
YO	y at first section
F,G,H	f,g,h
AT,BT	a,b
NSTEP	number of this solution face
NSMAX	maximum number of solution faces to be calculated
L-,MFREQ	frequency of full velocity profile outputs in x,y directions
PSI1,-8	ψ_i i = 1,8
PHI1,-8	ϕ_i i = 1,8
KAPPA,-L	κ as it appears in effective viscosity function, logarithmic law of the wall
K,A	K,A
CMIN	minimum value of zq_τ/ν for which law of the wall is assumed valid
ITN	iteration counter at this face
ITMAX	maximum number of iterations at each solution face
TOL	tolerance to which solution is to be iterated
SOS	accumulated error sum of squares for current iteration
NOSOS	number of points at which same has been accumulated
UL1,... P2B	quantities appearing in finite difference approximations to momentum equations (see figure (4.6.2)) and continuity equation (see figure(4.7.1,2)).
ALPHA,...S	α,β,γ,s streamline coordinates (see figure (4.6.1))
EPS	ϵ
A1,-5	coefficients of linear algebraic equations (50)
DSTAR	effective viscosity parameter (10)
C,...E3	contractions used in finite difference approximations to momentum equations at wall
D,...PHI	boundary layer parameters for output

The last two sets of variables partitioned by dashed lines are alternative storages.

Although the overall calculation is referred to a 10 × 50

grid, at section I the number of points at which the solution is calculated (NOPTS(I)) may be reduced to avoid calculating excessive points at sections where the boundary layer is thinnest. At least NSLAP 'slack points' are accommodated above the boundary layer edge ($\delta_{0.999}$) at all sections however.

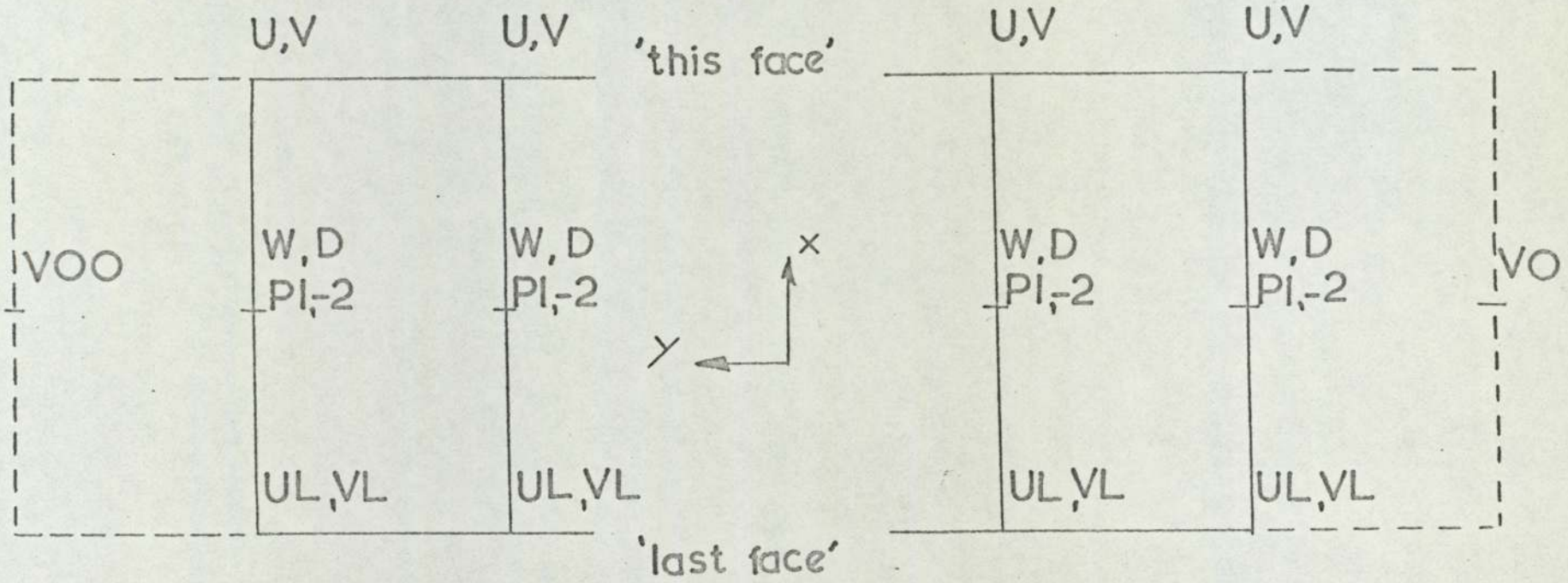
The arrays A1,-5 are used for storing the coefficients of the linear algebraic equations described in section 4.6. Remembering that the first point to be used in the finite difference scheme is the log-point the general equation is then

$$a_{3,i-1}\, u_{n^*+i-2} + a_{1,i}\, u_{n^*+i-1} + a_{2,i}\, u_{n^*+i} = a_{4,i}$$

$$a_{3,i-1}\, v_{n^*+i-2} + a_{1,i}\, v_{n^*+i-1} + a_{2,i}\, v_{n^*+i} = a_{5,i}$$

The output facility caters for full velocity profile outputs only at sections where these are specified and elsewhere only boundary layer parameters are output. For example, if LFREQ=2 and MFREQ=3 velocity profiles would be output at sections 1,4,7,10 (assuming there were ten sections at each face) at faces 2,4,6,...

The diagram below shows a plan of two adjacent solution faces and indicates where the various velocity profiles are stored.



ARRAY STORAGE

Subroutines.

Included below are brief notes on the subroutines that make up the present program. More detailed information can be obtained by referring to the program listing in Appendix A6 or to the flow diagrams included above where the sequence of some of the subroutines has been included. Subroutines AC99A,-H,-V as included in the program listing have only restricted application, some possibilities for their extended use however will be given later.

Main program

Reads data input, sets up initial conditions. Updates solution before each face is calculated.

Subroutine AC991

Controls velocity profile calculation at current solution face. Adjusts grid as and when necessary. Outputs error message E11 when solution at this face will not converge to required tolerance, calculation then continues.

Subroutine AC992

Controls calculation of one complete iteration

Subroutine AC99A

Sets up side flow boundary condition. Zero crossflow subroutine included will cater for both two-dimensional and pseudo-three-dimensional calculations. Called before each iteration.

Subroutine AC99B (IT,AS)

Calculates boundary layer thicknesses across a section and stores in BLT(10). To calculate δ at last face AS = 0.0, at this face AS = 1.0 (and pro rata). In addition if

Subroutine AC99B (IT,AS)(contd.)

IT = 1 $\delta_{0.99}$ calculated for output and stored as number of large increments from wall

IT = 2 $\delta_{0.999}$ calculated for grid control and stored in terms of grid point numbers.

Subroutine AC99C(M,N)

Sets up quantities for approximation to continuity equation at point (M,N).

Subroutine AC99D(...)

Calculates effective viscosity terms.

Function AC99E(...)

Laminar or turbulent effective viscosity model.

Subroutine AC99F(M,N)

Sets up linear algebraic equations corresponding to approximations to momentum equations at log-point at section M. Weights listed in the last column of Table 4.6.1 are implied.

Subroutine AC99G(NEQ,M,N)

Sets up linear algebraic equations (equation NEQ at section M) corresponding to approximations to momentum equations at point M,N (not log-point)

Subroutine AC99H

Called before calculation at each face to allow forward step sizes, output frequencies, etc. to be altered as required.

Function AC99I (M,N,GAMMA, VEL,MM)

Interpolates velocity components stored in VEL(10,50). At point N, array VEL is interpolated to provide VEL at M+GAMMA. Will not cater for MM=2.

Function AC99J(M,BETA,PRM,MM)

Interpolates parameters stored in PRM(10).

Array PRM is interpolated to provide PRM at M+BETA(MM/2).

Subroutine AC99L(N)

Calculates ϵ by solving lagarithmic law of wall (equations (4.4.10,11)). Puts ϵ=1 for laminar flow. For turbulent flow requires α, q_2 set up in common. Error message EL1 output when solution will not converge.

Subroutine AC99M(M,N)

Sets up quantities for approximations to momentum equations at point (M,N) (see figures (4.6.1,2)).

Subroutines AC990,-P

Respectively calculate and print output quantities.

Subroutine AC99Q

Sets up freestream condition by calling AC99U,-V.

Subroutine AC99R(M)

Sets up effective viscosity function parameters in DSTAR(10) at each section.

Subroutine AC99S(M,N)

Calculates α,β,γ,s corresponding to streamline through point (M,N) on this solution face (see Appendix A1). Outputs error message ES1 when solution will not converge, computation discontinued.

Function AC99T(R,...)

Integrates the function R through the boundary layer using the trapezium rule.

Functions AC99U,-V(X,Y)

Calculate the U,V components of velocity respectively in the mainstream at the point X,Y.

Subroutine AC99W(M,N)

Calculates W at point (M,N) from approximation to continuity equation.

Subroutine AC99X

Calculates NOPTS, LOGPT at each section based on last solution face. Puts LOGPT = 1 for laminar flow and for turbulent ensures

$$2 \leqslant \text{LOGPT} \leqslant \omega(\lambda-1)$$

If $zq_1/\nu >$ CMIN at point $\omega(\lambda-1)$ error message EX1 output, and proceeds with upper bound for LOGPT.

Function AC99Y(M,N,I,J,K,AL)

Supplies integrand for AC99T to evaluate

$$\int_0^\infty \left(1 - \frac{u_1}{U_1}\right)^I \left(\frac{u_1}{U_1}\right)^J \left(\frac{v_1}{U_1}\right)^K dz$$

at section (M,N) (AL as in AC99B).

U1,V1,Q1 must be set up as for freestream prior to entry in common.

Subroutine AC99Z (NEQ)

Solves the set of NEQ tri-diagonal linear algebraic equations (see Appendix A5).

To use the program.

The input requirements for the computer program are shown below. The majority of the symbols used will be found in the list of symbols (p.103) and a few comments will now be made concerning the remainder.

Item 3. With the program as included the following limitations need be imposed

$3 \leqslant M \leqslant 10$ or $M = 1$

$N \leqslant 50$

$\lambda \geqslant 2$

$\omega \geqslant 1$

Explanations of NSLAP, LT, NSMAX, ITMAX, LFREQ, MFREQ can be found in the common storage list included previously.

Item 4. x_o, y_o are the coordinates of the first section on the initial solution face and θ_{11} is the momentum thickness at this face.

Items 6 and 7. Surplus blank cards should be removed. The velocity profiles should be specified on the mesh defined previously and the streamwise velocity profile should be scaled to unity at the boundary layer edge.

Item 8. Specifies the crossflow to be included at the commencement of the calculation. $\mathrm{Tan}\beta_o$ is specified at each section.

All quantities listed are retained throughout the calculation unless changed in AC99H.

Sample subroutines and input data are included following the data listing (these are those used to simulate the experiment of Hornung and Joubert described in section 6.6). In the subroutine AC99A listed V0 is calculated from the symmetry condition and V00 from equation (6.6.2). Subroutine AC99H changes forward step and output

frequencies during the course of the calculation (N.B. halving the forward step when NSTEP = 4 causes the shorter step to be applied before face 4 is calculated). The U,V velocity distribution used is that given in Appendix A9.

There then follows the profile used to account for crossflow at the commencement of the calculation and that used to account for the convergency of the flow in section 5.3. Tables and graphs are included for both cases.

During the course of the calculation error messages may be output to signify that some fault has occurred with the calculation. The course followed as each of these errors is encountered and the cause is listed below:

E11 Error sum of squares has failed to reach required tolerance in specified maximum number of iterations.
Output NSTEP, SOS, NOSOS.
Calculation continues to next face.

EL1 Iteration process for calculating ϵ has failed to converge.
Output Q2, N, ALPHA and last two iterates.
Calculation continues using last iterate.

ES1 Iteration process for calculating streamline has failed to converge.
Point M,N at which error occurred output.
Program discontinued.

EX1 zq_T/ν within range in which log-point may fall, is always less than CMIN.
Output section concerned (L,M) and value of zq_T/ν at outermost point.
Continue calculation with log-point set at this outermost point.

It should also be pointed out here that a transformed version of the effective viscosity function equation (3.3.5) has been used within the program. The model, employed outside the laminar sublayer only, can be written such that the transformed effective viscosity ν_e' is a function of ζ' where

$$\zeta' = \kappa^2 \xi^2 \left|\frac{\partial q}{\partial \xi}\right|$$

as follows

$$\nu_e' = \zeta' \qquad \zeta' < K Q d^*$$

$$\nu_e' = K Q d^* \qquad \zeta' > K Q d^*$$

where

$$d^* = \int_0^\infty \left(1 - \frac{u_1}{U_1}\right) d\xi$$

The empirical function incorporated into the program

$$K = K(\Gamma)$$

is defined as follows

$$K = 0.016 + 0.00015\ \Gamma \qquad \Gamma > -60$$

$$= 0.007 \qquad \Gamma < -60$$

where

$$\Gamma = 10^4\ \frac{\theta_{11}}{Q}\ \frac{\partial Q}{\partial s}$$

The function is shown plotted toward the end of this appendix.

As mentioned previously alternative effective viscosity models could easily be incorporated into the program.

Finally we refer back to the discontinuities ~~noted~~ exhibited in shape factor predictions at the commencement of the calculations considered in Chapter Five. We note here that the same feature was seen in β_o predictions in Chapter Six and as a result it is considered preferable where possible to vary β_o at the beginning of the calculation to ensure agreement between predicted crossflows (δ_2^* say) and the crossflow required as an initial condition.

CARD INPUT REQUIREMENTS - ~~DIMENSIONS~~ UNITS SHOULD BE CONSISTENT THROUGHOUT

item 1 Heading Cards

item 2 Empirical Constants

κ κ(log law) X A CMIN

item 3

M N λ ω $\lambda\omega$ VSLAP LT VSMAX ITMAX LEVEL MERES

item 4

x_0 y_0 f g θ_{11} ν

item 5 Solution Weights

Ψ_1 Ψ_2 Ψ_3 Ψ_4 Ψ_5 Ψ_6 Ψ_7 Ψ_8

item 6 Input Velocity Profile

item 7 Crossflow Velocity Profile

item 8 Tan β_o at input, at each section

item 9 Solution Tolerance

ERR

HORNUNG AND JOUBERT
SECONDARY FLOW INDUCED BY CYLINDER BETWEEN PARALLEL WALLS

```
0.410     0.410     0.011     4.9       30.0

9 48  2 10 20  6  2 12 20  3  4

 -4.0      0.0       0.25      0.25      0.03      0.00017

0.5       0.0       1.0       1.0       1.0       0.0       1.0       0.75

.4087     .4641     .4974     .5214     .5404     .5561     .5696     .5813
.5918     .6011     .6096     .6174     .6246     .6311     .6373     .6431
.6485     .6536     .6584     .6629     .7090     .7485     .7843     .8160
.8443     .8696     .8922     .9120     .9291     .9437     .9560     .9661
.9743     .9808     .9859     .9898     .9928     .9949     .9965     .9977
.9985     .9990     .9994     .9996     .9998     .9999     1.0000    1.0000

 .411      .466      .488      .497      .500      .499      .497      .493
 .439      .433      .478      .472      .466      .461      .455      .449
 .443      .438      .432      .427      .371      .313      .260      .212
 .172      .138      .109      .085      .065      .049      .036      .025
 .017      .011      .006      .003      .000      .000      .000      .000
 .000      .000      .000      .000      .000      .000      .000      .000
 .000      .000      .000      .000      .000      .000      .000      .000

1.000E- 9
```

SAMPLE INPUT - THAT USED TO SIMULATE THE EXPERIMENT OF HORNUNG AND JOUBERT, SECTION 6.6

SAMPLE SUBROUTINES - AC99A,-H,-U,-V

THOSE USED TO SIMULATE THE EXPERIMENT OF

HORNUNG AND JOUBERT, SECTION 6.6

```
      SUBROUTINE AC99A                                                    A000AC99
      INTEGER OMEGA                                                       A001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                            A002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,              A003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),             A004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),VO(50),VOO(50),                   A005AC99
     3XL,YO,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                          A006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                            A007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                             A008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,             A009AC99
     1UL1,UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,                 A010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,           A011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                                 A012AC99
     4DSTAR(10),C,D,E1,E2,E3                                              A013AC99
      DO 1 N=1,NN                                                         A020AC99
      VO(N)=-0.5*(V(2,N)+VL(2,N))                                         A030AC99
      VOO(N)=VOO(NN)*(2.0*(V(MM,N)+VL(MM,N))/(V(MM,NN)+VL(MM,NN))         A040AC99
     1            -(V(MM-1,N)+VL(MM-1,N))/(V(MM-1,NN)+VL(MM-1,NN)))       A050AC99
    1 CONTINUE                                                            A060AC99
      RETURN                                                              A070AC99
      END                                                                 A080AC99
```

```
      SUBROUTINE AC99H                                                  H000AC99
      INTEGER OMEGA                                                     H001AC99
      REAL NU,NUL1,NUL2,NU1,NU2,K,KAPPA,KAPPAL                          H002AC99
      COMMON   MIN,MOUT,MM,NN,LAMDA,OMEGA,NMINC,NSLAP,LT,NU,           H003AC99
     1UL(10,50),VL(10,50),U(10,50),V(10,50),W(10,50),BLT(10),           H004AC99
     2P1(10),P2(10),LOGPT(10),NOPTS(10),V0(50),V00(50),                 H005AC99
     3XL,YO,F,G,H,AT,BT,NSTEP,NSMAX,LFREQ,MFREQ,                        H006AC99
     4PSI1,PSI2,PSI3,PSI4,PSI5,PSI6,PSI7,PSI8,                          H007AC99
     5PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8                           H008AC99
      COMMON   KAPPA,KAPPAL,K,A,CMIN,ITN,ITMAX,TOL,SOS,NOSOS,           H009AC99
     1UL1.UL2,UL3,VL1,VL2,VL3,U1,U2,U3,V1,V2,V3,Q1,Q2,Q3,               H010AC99
     2NUL1,NUL2,NU1,NU2,XI,XINC,P1B,P2B,ALPHA,BETA,GAMMA,S,EPS,         H011AC99
     3A1(50),A2(50),A3(50),A4(50),A5(50),                               H012AC99
     4DSTAR(10),C,D,E1,E2,E3                                            H013AC99
      IF(NSTEP-4) 2,1,2                                                 H020AC99
    1 NSMAX=21                                                          H030AC99
      F=F/2.0                                                           H040AC99
      GO TO 6                                                           H050AC99
    2 IF(NSTEP-12) 4,3,4                                                H060AC99
    3 MFREQ=1                                                           H070AC99
      GO TO 6                                                           H080AC99
    4 IF(NSTEP-16) 6,5,6                                                H090AC99
    5 NSMAX=39                                                          H100AC99
      F=F/4.0                                                           H110AC99
      LFREQ=1                                                           H120AC99
    6 RETURN                                                            H130AC99
      END                                                               H140AC99
```

```
      FUNCTION AC99U(X,Y)                                                U000AC99
      PI=3.142                                                           U001AC99
      A=5.0                                                              U002AC99
      B=37.18                                                            U003AC99
      UDB=1.699                                                          U004AC99
      SX=2.0*PI*X/A                                                      U005AC99
      SY=2.0*PI*Y/A                                                      U006AC99
      CSX=(EXP(SX)+EXP(-SX))/2.0                                         U007AC99
      CSY=COS(SY)                                                        U008AC99
      H=CSX-CSY                                                          U009AC99
      AC99U=B*(UDB-2.0*PI*(CSX*CSY-1.0)/(A*H**2))                        U010AC99
      RETURN                                                             U011AC99
      END                                                                U012AC99
```

```
      FUNCTION AC99V(X,Y)                                                V000AC99
      PI=3.142                                                           V001AC99
      A=5.0                                                              V002AC99
      B=37.18                                                            V003AC99
      SX=2.0*PI*X/A                                                      V004AC99
      SY=2.0*PI*Y/A                                                      V005AC99
      CSX=(EXP(SX)+EXP(-SX))/2.0                                         V006AC99
      CSY=COS(SY)                                                        V007AC99
      SSX=(EXP(SX)-EXP(-SX))/2.0                                         V008AC99
      SSY=SIN(SY)                                                        V009AC99
      H=CSX-CSY                                                          V010AC99
      AC99V=-2.0*PI*B*SSX*SSY/(A*H**2)                                   V011AC99
      RETURN                                                             V012AC99
      END                                                                V013AC99
```

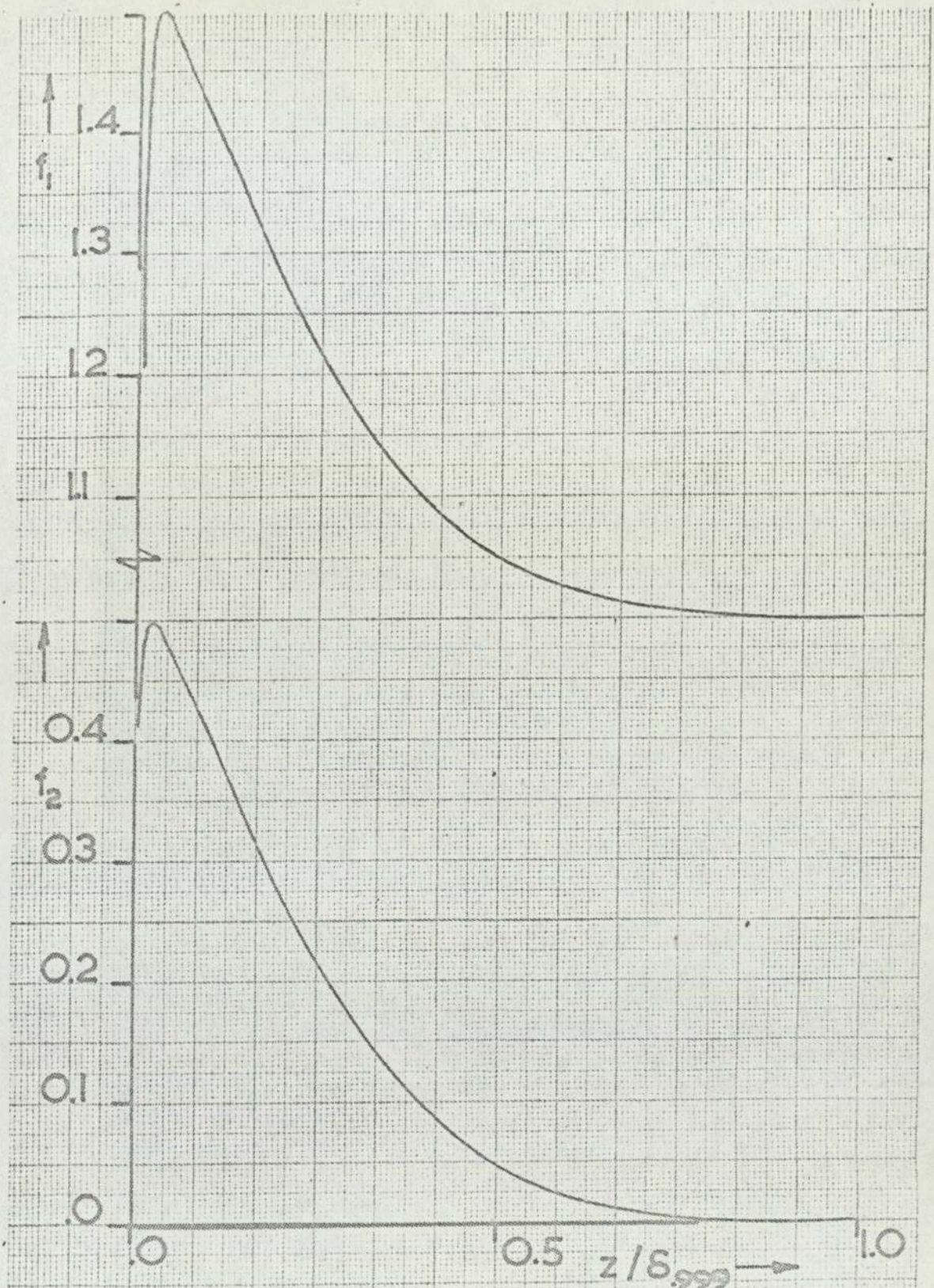
THE PROFILE USED TO ALLOW FOR CROSSFLOW AT START OF CACULATION - THE FUNCTION f_2, WHICH IS SCALED TO GIVE REQUIRED CROSSFLOW, IS TABULATED AGAINST

$$\frac{z}{\delta_{0.999}} = \frac{\eta}{24}$$

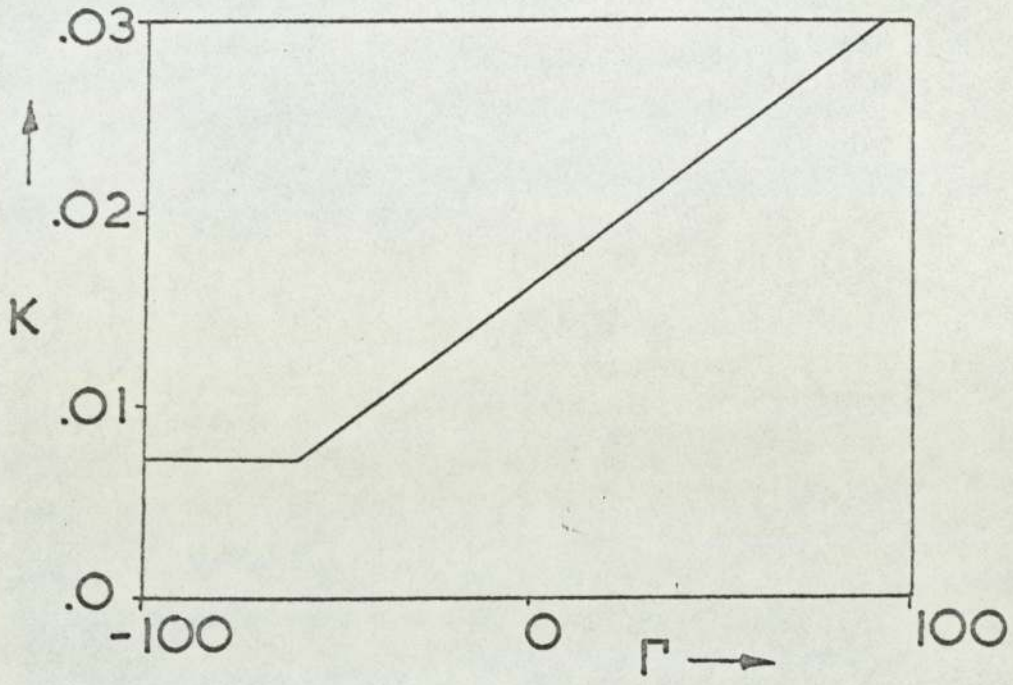
η	f_2		η	f_2
0.1	0.411		6.0	0.212
0.2	0.466		7.0	0.172
0.3	0.488		8.0	0.138
0.4	0.497		9.0	0.109
0.5	0.500		10.0	0.085
0.6	0.499		11.0	0.065
0.7	0.497		12.0	0.049
0.8	0.493		13.0	0.036
0.9	0.489		14.0	0.025
1.0	0.483		15.0	0.017
1.1	0.478		16.0	0.011
1.2	0.472		17.0	0.006
1.3	0.466		18.0	0.003
1.4	0.461		19.0	0.000
1.5	0.455		20.0	0.000
1.6	0.449		21.0	0.000
1.7	0.443		22.0	0.000
1.8	0.438		23.0	0.000
1.9	0.432		24.0	0.000
2.0	0.427		25.0	0.000
			26.0	0.000
3.0	0.371		27.0	0.000
4.0	0.313		28.0	0.000
5.0	0.260		29.0	0.000
			30.0	0.000

THE PROFILE USED TO ALLOW FOR CONVERGENCY ON PLANE OF SYMMETRY - THE FUNCTION $f_1 = \underset{y \to o}{\text{limit}} \frac{v}{V}$ IS TABULATED AGAINST $\frac{z}{\delta_{0.999}} = \frac{\eta}{24}$

η	f_1		η	f_1
0.1	1.2061		6.0	1.2141
0.2	1.3815		7.0	1.1722
0.3	1.4472		8.0	1.1374
0.4	1.4775		9.0	1.1087
0.5	1.4918		10.0	1.0852
0.6	1.4978		11.0	1.0662
0.7	1.4990		12.0	1.0508
0.8	1.4975		13.0	1.0385
0.9	1.4942		14.0	1.0287
1.0	1.4898		15.0	1.0211
1.1	1.4847		16.0	1.0152
1.2	1.4792		17.0	1.0107
1.3	1.4735		18.0	1.0074
1.4	1.4675		19.0	1.0049
1.5	1.4616		20.0	1.0032
1.6	1.4558		21.0	1.0019
1.7	1.4498		22.0	1.0010
1.8	1.4441		23.0	1.0005
1.9	1.4384		24.0	1.0001
2.0	1.4328		25.0	1.0000
			26.0	1.0000
3.0	1.3778		27.0	1.0000
4.0	1.3189		28.0	1.0000
5.0	1.2632		29.0	1.0000
			30.0	1.0000



Profiles used to allow (i) for convergency on plane of symmetry: $f_1 = \lim_{y \to 0} \frac{v}{V}$, and (ii) for crossflow at input: $f_2 \propto v_1$



The function K(Γ)

APPENDIX A8.

STABILITY CONSIDERATIONS

A number of additional simulations of experiment 'E' of Schubauer and Spangenberg (see section 5.1) were made with the intention of obtaining an indication of the effects of step sizes and initial conditions on the solution scheme; we will discuss these now.

The figures at the end of this appendix show the effects of varying initial conditions. The solution scheme seems to be insensitive to changes in the initial value of the shape factor H (leaving θ unchanged) as is shown in figure (A8.1). It seems strange that by increasing H at the start of the calculation the value of H as separation is approached should be reduced, even though in the three separate calculations made H tends to the same value (approximately) within a short distance of the start of the calculation. R_θ, c_f developments are surprisingly only slightly affected by initial H despite the apparently large discrepancies in H at x = 16 ½'. The solution scheme is however more sensitive to changes in the value assumed by R_θ at the beginning of the calculation as is shown by figures (A8.2,3). All three runs shown start with H = 1.3 at x = 0, curve (2) being the run plotted in figures (5.1.8-9) while curves (1),(3) have R_θ increased, reduced respectively by 3%. This imposed difference in R_θ is maintained throughout the calculations and the flow corresponding to curve (1) is predicted to separate just short of x = 16'.

It is difficult to state what the precise physical effects of the above considerations would be, except to say that the effect of varying R_θ at x = 0 is very much as might be expected, but it is obviously of some consequence that the disturbed initial conditions do not produce any instability in the solution.

We now proceed to discuss the effects of step sizes on the solution scheme. All the computer runs mentioned above were made using the following grid specification:-

$$N = 48 \qquad \lambda = 2 \qquad \omega = 10$$

and the grid was continuously adjusted so that the large z increment at any section was

$$h = \frac{1}{24} \delta_{0.999}$$

where $\delta_{0.999}$ is the boundary layer thickness corresponding to $u = 0.999U$. The marching step f for experiment 'E' was (from $x = 0$) 24 steps of $\frac{1}{2}$ followed by 48 steps of $\frac{1}{8}$ (all units in feet). Over the first twelve feet the forward step varied from 4 to 1.33 boundary layer thicknesses and for $x > 12$ (where changes were occurring much more rapidly) the forward step was from 0.33 to 0.15 boundary layer thicknesses. Such a run took 3 minutes on the IBM S360/65 computer, 7-8 iterations being required on average at each step to obtain velocity components correct to 4 significant figures. The calculation made using the grid specified above will be used as the basis of comparisons with the calculations to be discussed below.

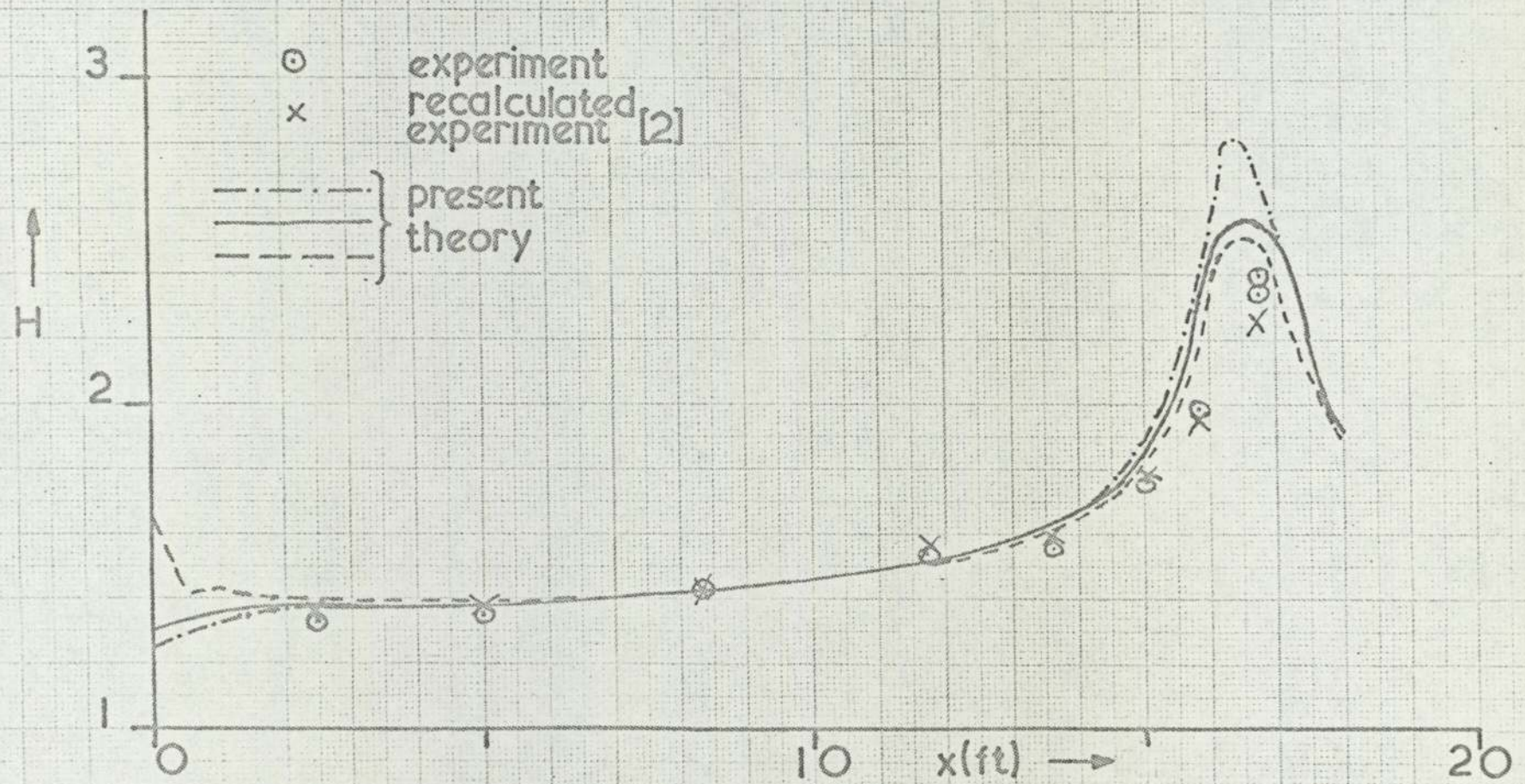
If for $x < 12'$ we take $f = 1$ (everywhere at least 2.66 boundary layer thicknesses) H changes by only $\frac{1}{4}\%$ at $x = 12'$ and if $f = \frac{1}{4}$ for $x < 12'$ H changes by only $^1/_{10}\%$ at $x = 12'$. For $x > 12'$ if we take $f = \frac{1}{4}$ the change produced in H (between $x = 12'$ and $x = 18'$) is less than $\frac{1}{2}\%$, c_f being increased by 1.4% which however for the small values encountered at $x = 18'$ the absolute change was less than 10^{-5}. Such changes confirm the marching step chosen as being adequate.

Next a run was made with the number of points at each section reduced by a factor of two as follows:-

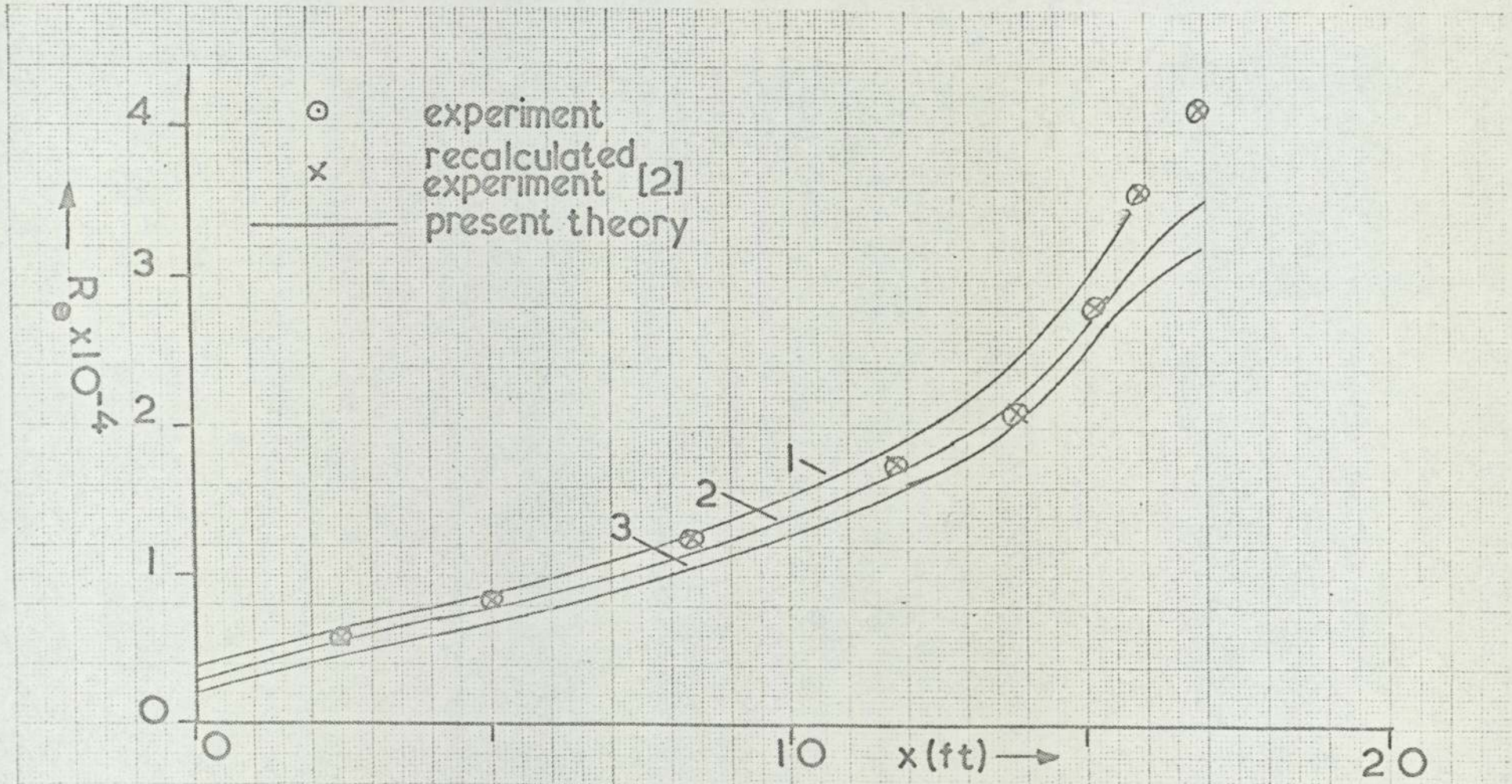
$$N = 23 \qquad \lambda = 2 \qquad w = 5$$

$$h(\text{large } z \text{ increment}) = \frac{1}{12}\,\delta_{0.999}$$

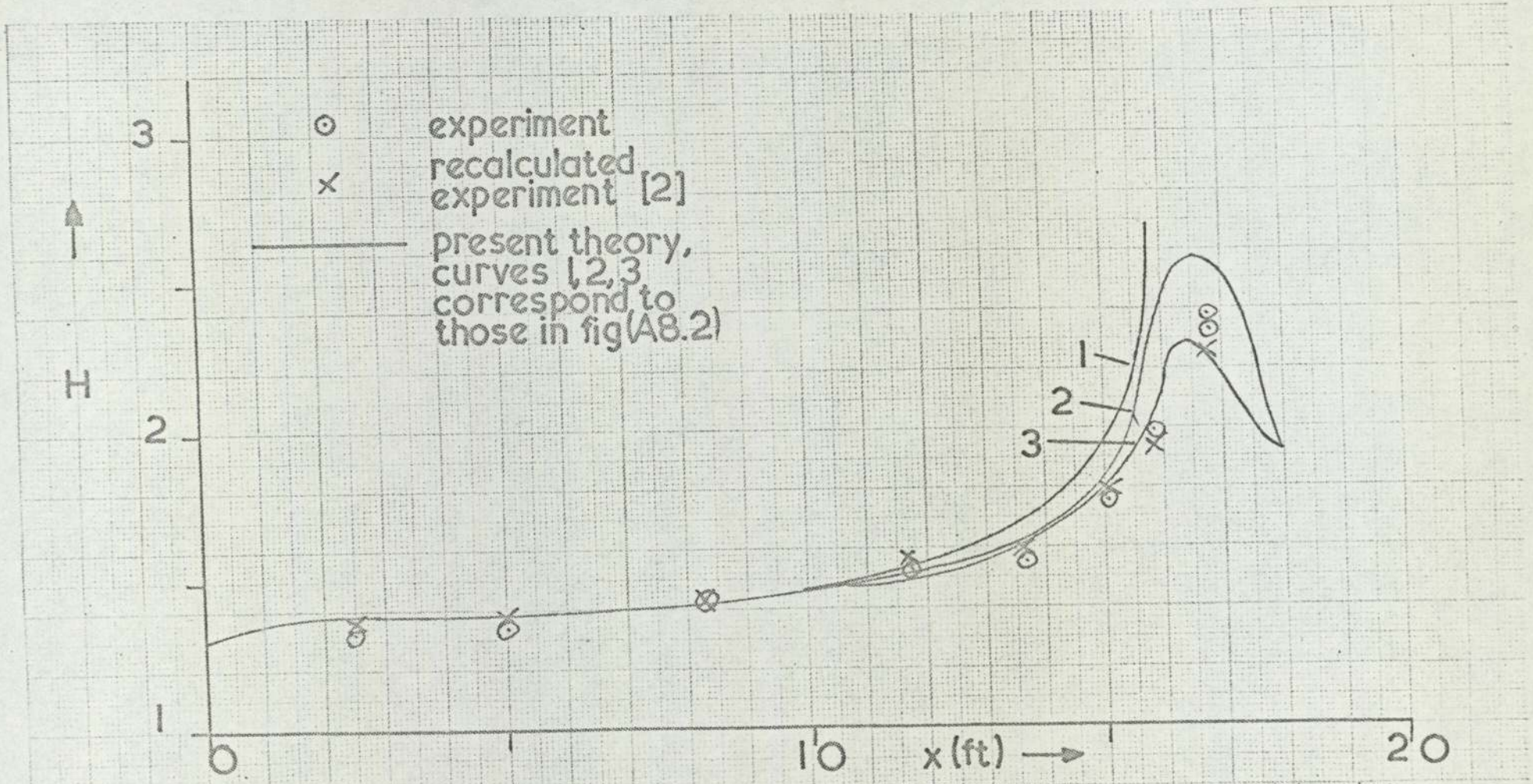
which results in the smaller mesh intervals at the wall being increased by a factor of 4 (the run took 1.2 minutes, the number of iterations/step remaining on average unchanged). At x = 18' the differences encountered (starting the calculation from x = 0) were greater than those obtained above, H being reduced by 1½%, but this was not considered excessive since the integral thicknesses had to be obtained (using the trapezium rule) from a much coarser mesh and H is very sensitive to changes in δ^*, θ. It is anticipated that the mesh used throughout all the present calculations is finer, with respect to the z increment (being based on N = 48), than need necessarily be the case and it is expected that N may be reduced, without significant loss of accuracy, to economise on computer storage and time. Even so the scheme based on N = 48 is still economical in terms of computer time although it must be admitted that since the three-dimensional program is being used for a two-dimensional calculation computer storage could be reduced considerably.



FIG(A8.1) Schubauer and Spangenberg [26], experiment 'E'
Effect of varying initial H in present calculation



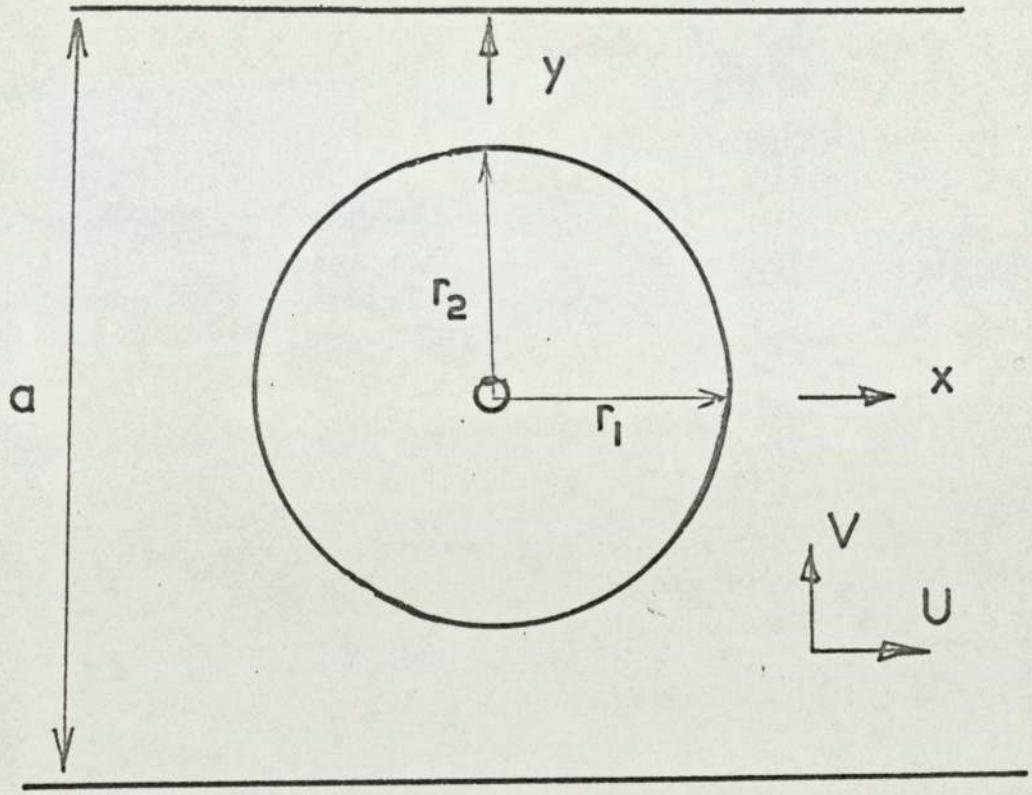
FIG(A8.2) Schubauer and Spangenberg [26], experiment 'E'
Effect of varying initial R_θ in present calculation



FIG(A8.3) Schubauer and Spangenberg [26], experiment 'E'
Effect of varying initial R_θ in present calculation

APPENDIX A9

POTENTIAL FLOW ABOUT A NEAR-CIRCULAR CYLINDER BETWEEN PARALLEL WALLS



Kennard [33] gives the potential distribution for a stream flowing between parallel walls a distance a apart and about a near-circular cylinder mounted midway between the walls as

$$U = U_o - \frac{2\pi B}{aH^2}\left(\cosh\frac{2\pi x}{a}\cos\frac{2\pi y}{a} - 1\right)$$

$$V = -\frac{2\pi B}{aH^2}\sinh\frac{2\pi x}{a}\sin\frac{2\pi y}{a}$$

where U,V are the velocities associated with the x,y directions respectively, U_o and B are constants to be determined and H is given by

$$H = \cosh\frac{2\pi x}{a} - \cos\frac{2\pi y}{a}$$

For the purpose of the simulation discussed in section 6.6 we choose

$$a = 5, \quad U_o = 63.17, \quad B = 37.18$$

so that the radii of the cylinder are approximately

$$r_1 \doteq r_2 \doteq 0.9167$$

TABLES

TABLE 4.6.1

WEIGHTING FACTORS USED IN APPENDICES A2, A3

	Purpose	Plausible Range	Particular Cases [25]	Values Used.
ψ_1	Specifies point at which finite difference approximation is to be made	$0 \leq \psi \leq 1$	$\psi_1 = 0$ explicit scheme $\psi_1 = \frac{1}{2}$ Crank-Nicholson $\psi_1 = 1$ Laasonen	$\frac{1}{2}$
ψ_2	specifies weight between points n-1, n, n+1 in approximation to $\partial u/\partial x$	$0 \leq \psi < 1$	$\psi_2 = \frac{1}{6}$ allows longer h than $\psi_2 = 0$ with same convergence	0
ψ_3	Specifies to what extent approximation to terms is dependent on r-1 th or r th iteration. $\frac{\partial u}{\partial s}$	$0 < \psi \leq 1$	$\psi_3 = 0$ iteration r-1 used $\psi_3 = 1$ iteration r used	1
ψ_4	$\frac{\partial u}{\partial \xi}$	$0 \leq \psi \leq 1$	$\psi_4 = 0$ iteration r-1 used $\psi_4 = 1$ iteration r used	1
ψ_5	$\frac{\partial u}{\partial \xi}$ (viscous)	$0 \leq \psi \leq 1$	$\psi_5 = 0$ iteration r-1 used $\psi_5 = 1$ iteration r used	1
ψ_6	as ψ_2 but at $n = n^*$	$0 \leq \psi < 1$		0
ψ_7	specifies one of two approximations to $\partial u/\partial \xi$ at $n=n^*$	$\psi = 0,1$		1
ψ_8	overall solution weight (relaxation factor)	$\psi > 0$	$\psi_8 = 1$ iteration r-1 discounted	0.75

If $\psi_1 = 0$ ψ_4, ψ_5 have no effect.

TABLE 6.1.1

CASES TREATED IN THE SIMULATION OF THE INFINITE SWEPT WING

α_o (DEGREES)	0.0	17.5	35.0	52.5
κ (/FOOT)	.25	.25	.2 .25 .267 .3	.25

TABLE 6.4.1

EXPERIMENTAL DATA USED IN THE SIMULATION OF HOADLEY'S DIFFUSER

X INS	Q FT/S	α_o DEGREES	U FT/S	V FT/S
-13	87.28	30.5	75.2	44.3
3	85.38	30.2	73.8	42.9
11	76.36	32.8	64.2	41.4
19	69.52	34.0	57.6	38.9
27	59.02	32.2	49.9	31.5

TABLE 6.5.1

THE FUNCTIONS ϕ, ϕ' DERIVED IN THE SOLUTION OF THE AXIALLY SYMMETRIC LAMINAR STAGNATION FLOW

ζ	ϕ'	ϕ
0.00	0.0000	0.0000
0.15	0.1857	0.0139
0.30	0.3489	0.0540
0.45	0.4898	0.1169
0.60	0.6091	0.1993
0.75	0.7078	0.2981
0.90	0.7872	0.4102
1.05	0.8494	0.5330
1.20	0.8966	0.6639
1.35	0.9312	0.8010
1.50	0.9557	0.9425
1.65	0.9725	1.0872
1.80	0.9835	1.2339
1.95	0.9905	1.3819
2.10	0.9947	1.5308
2.25	0.9972	1.6802
2.40	0.9985	1.8298
2.55	0.9993	1.9796
2.70	0.9996	2.1295
2.85	0.9998	2.2795
3.00	0.9999	2.4295
3.15	1.0000	2.5795
3.30	1.0000	2.7294
3.45	1.0000	2.8794
3.60	1.0000	3.0294

F I G U R E S

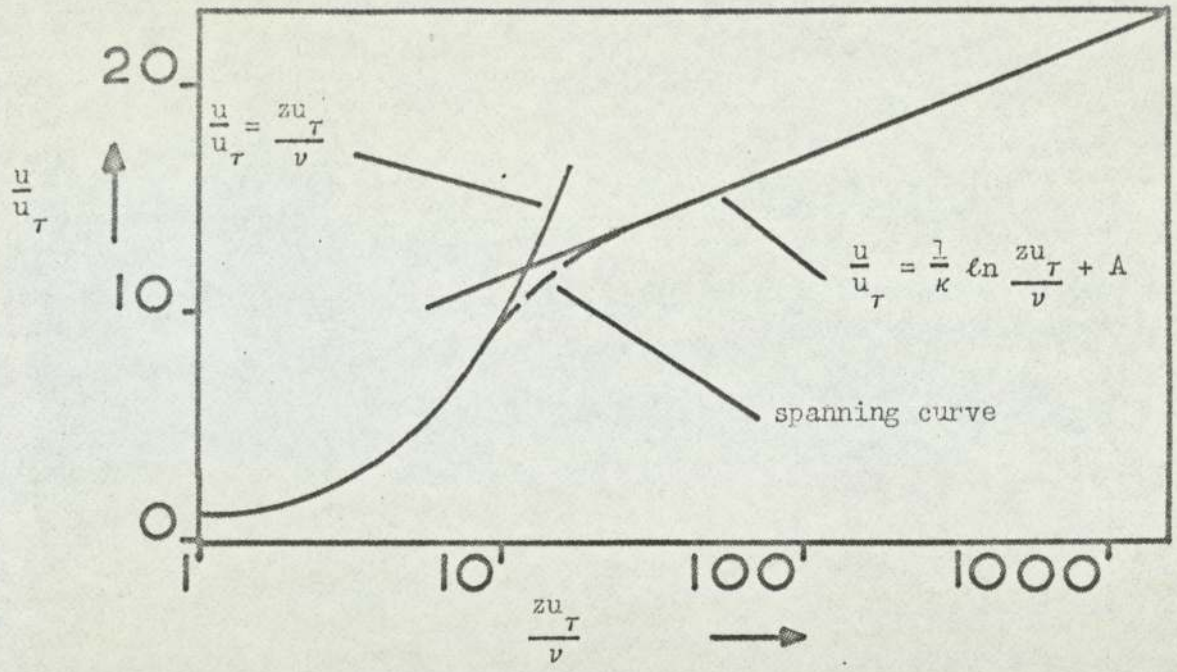


FIG (3.2.1) Semi-logarithmic plot of the law of the wall.

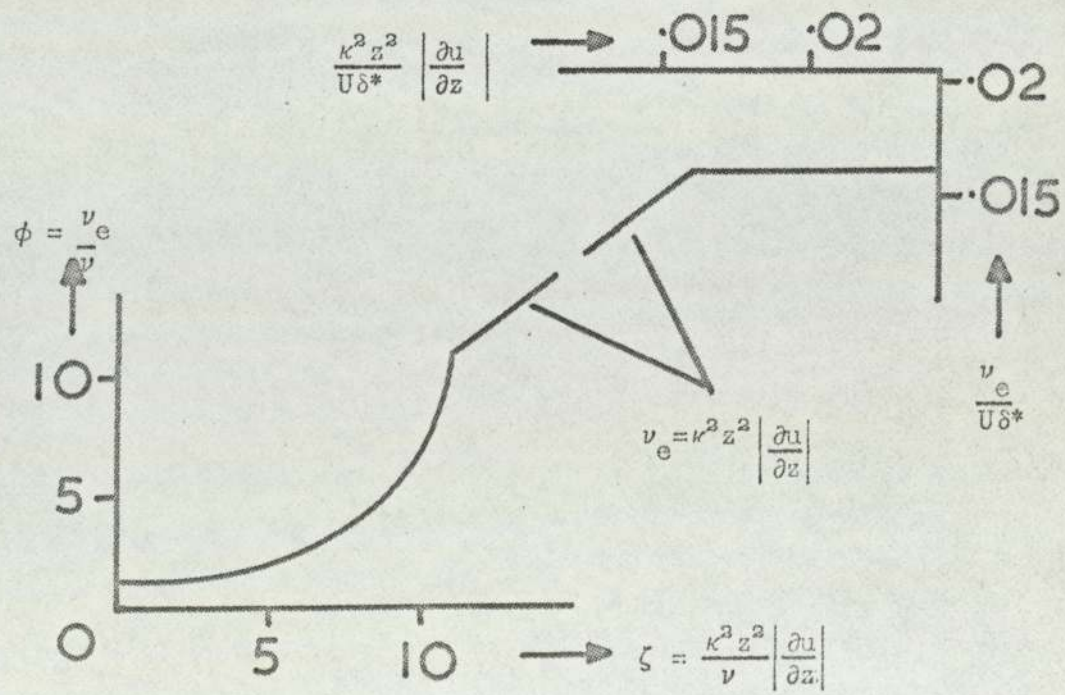


FIG (3.3.1) Composite effective viscosity function proposed by Mellor [19].

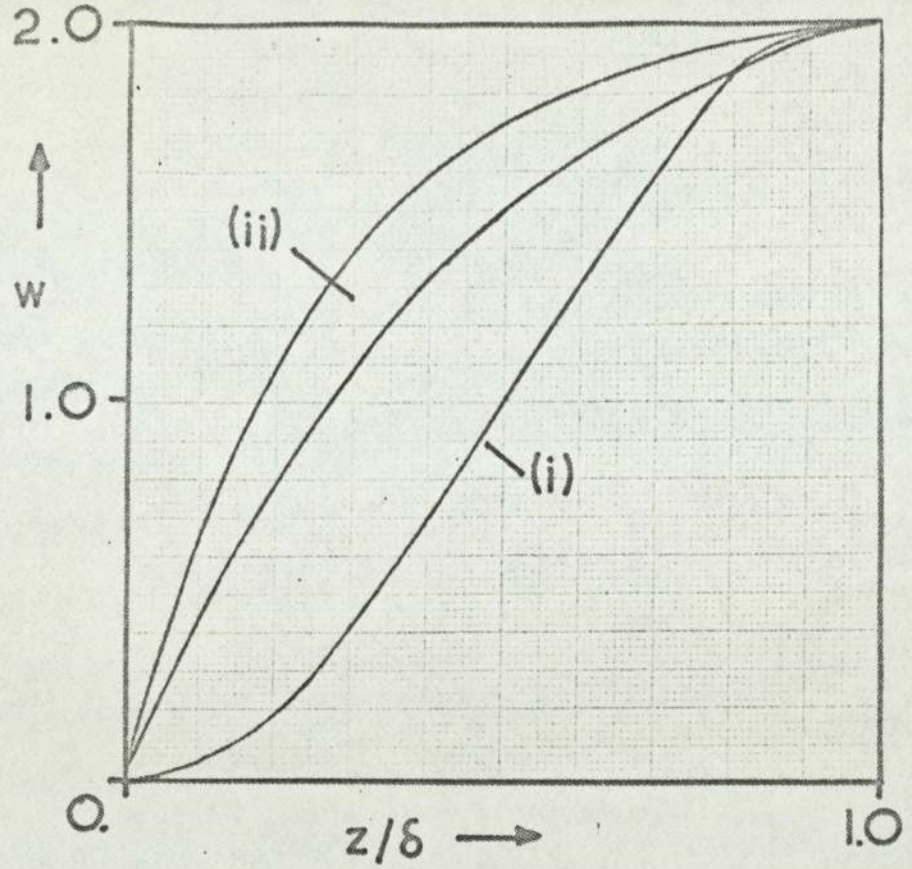


FIG (3.4.1) (i) Coles' law of the wake [15] and (ii) the scatter of data points for $\frac{2q \sin\beta}{Q \sin\beta_0}$ [13,22].

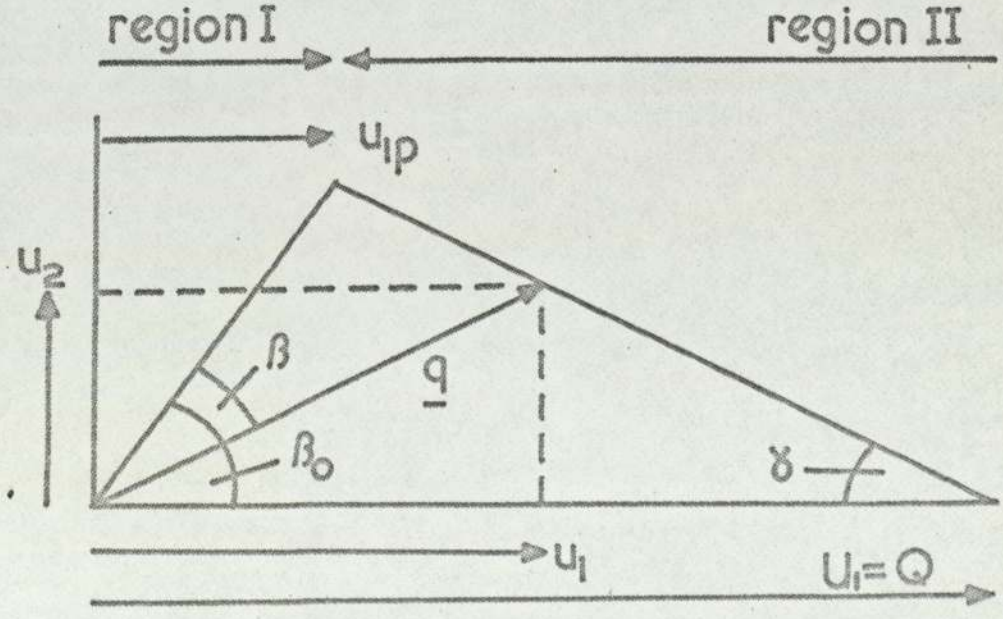
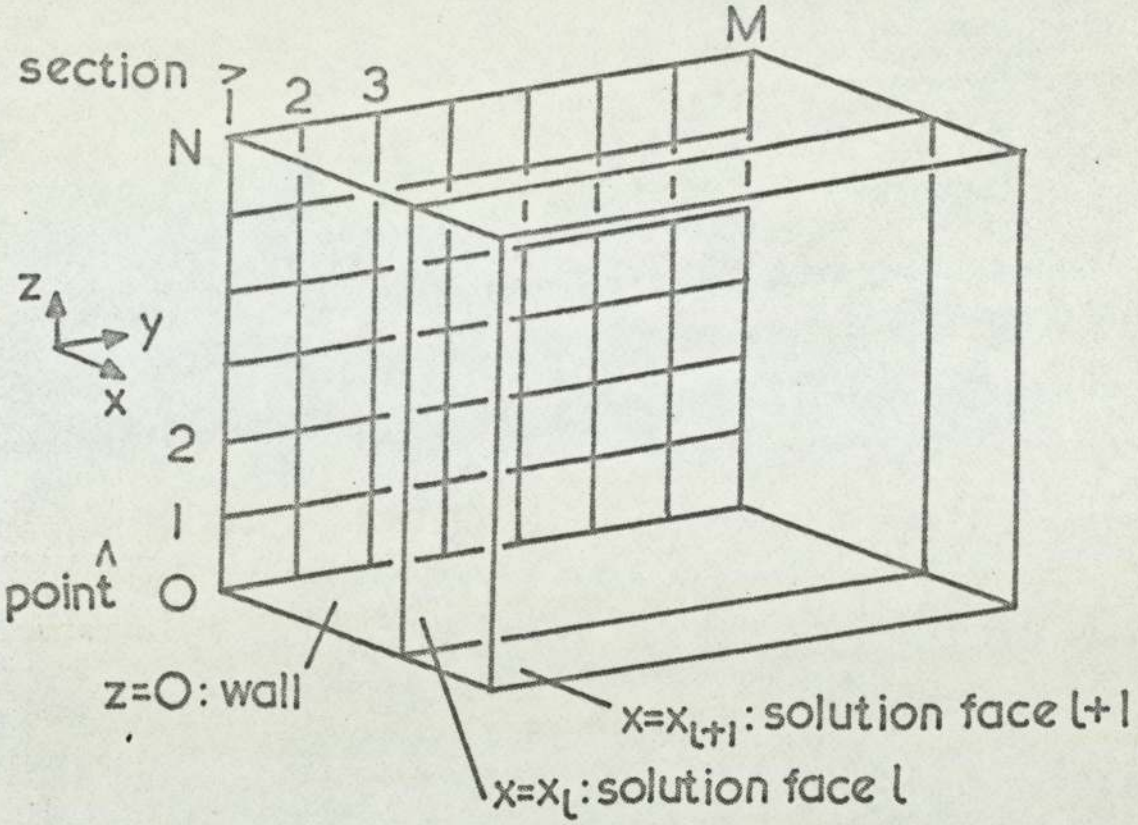
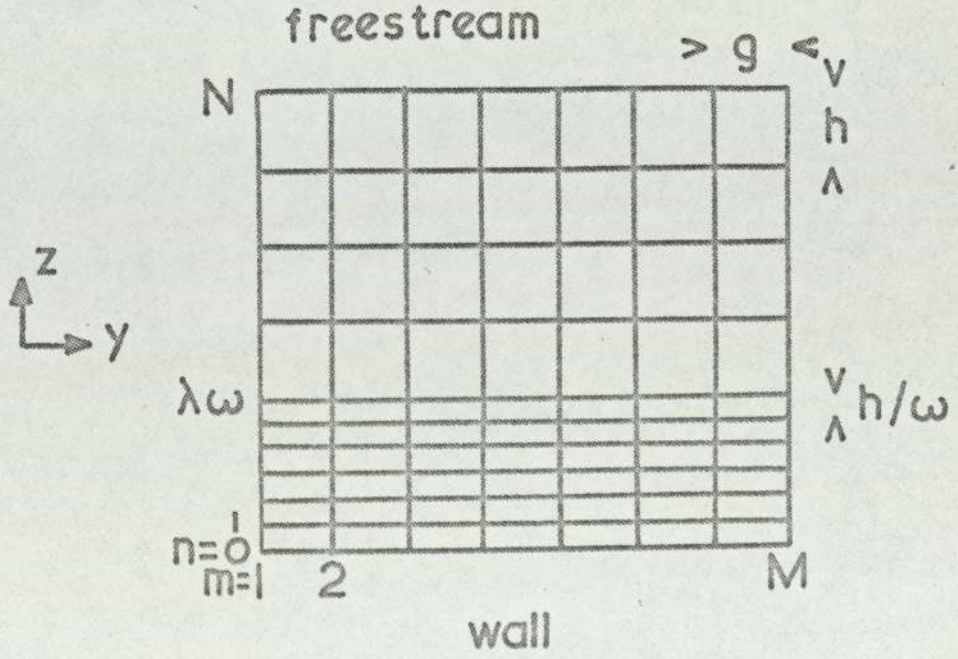


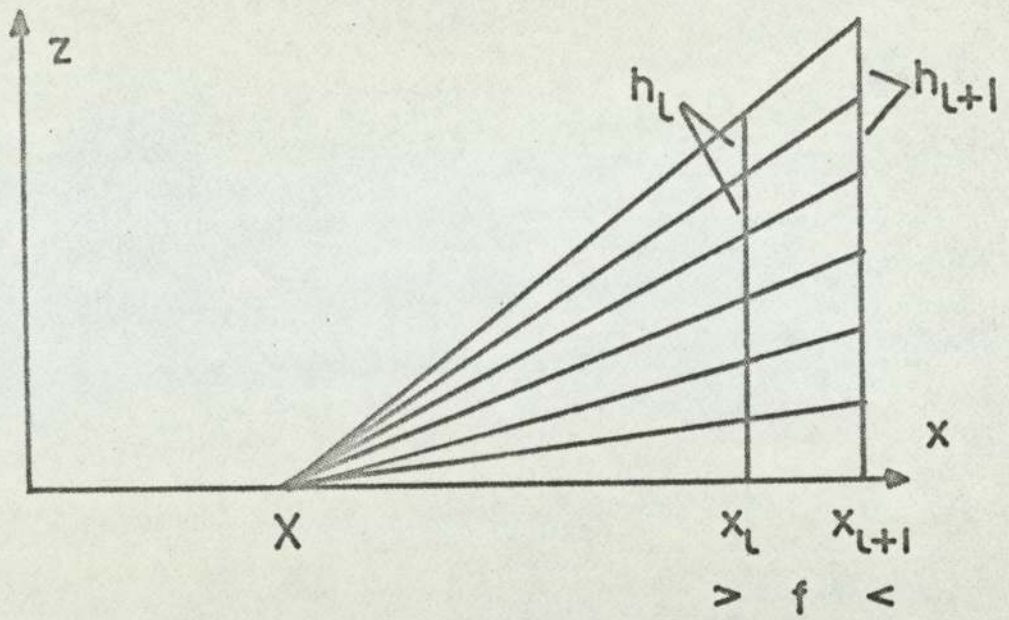
FIG (3.5.1) Johnston's triangular model [21].



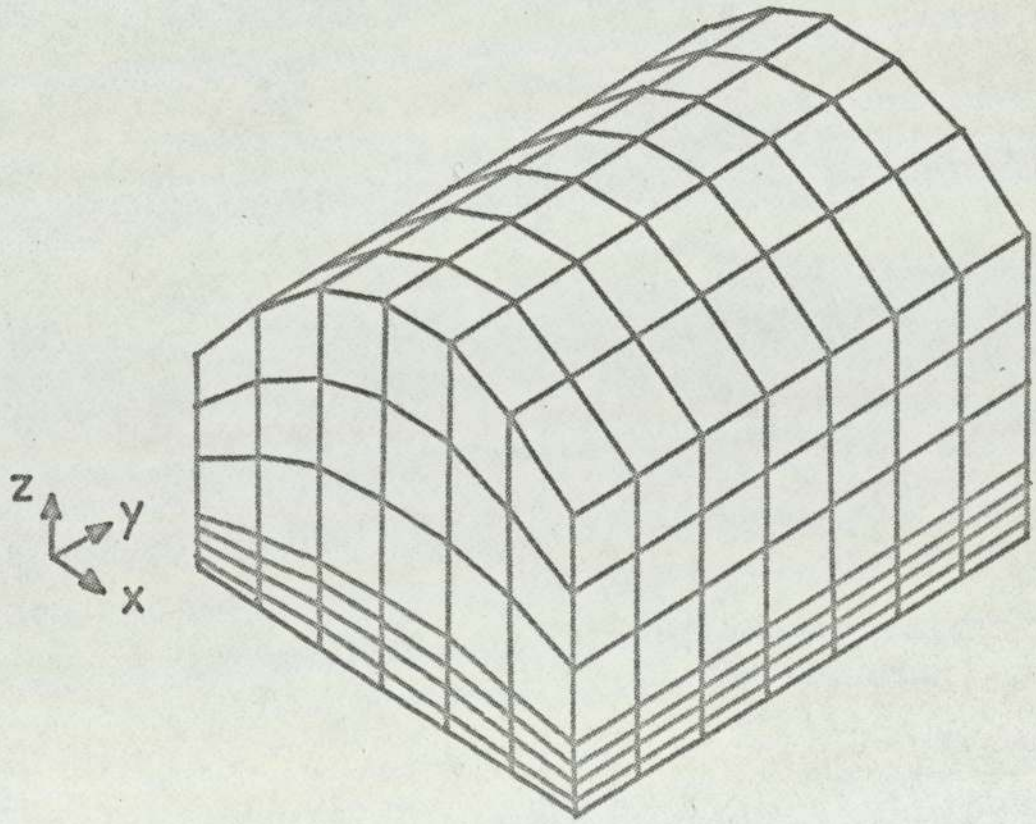
FIG(4.1.1) Solution face, section, point notation on the solution mesh.



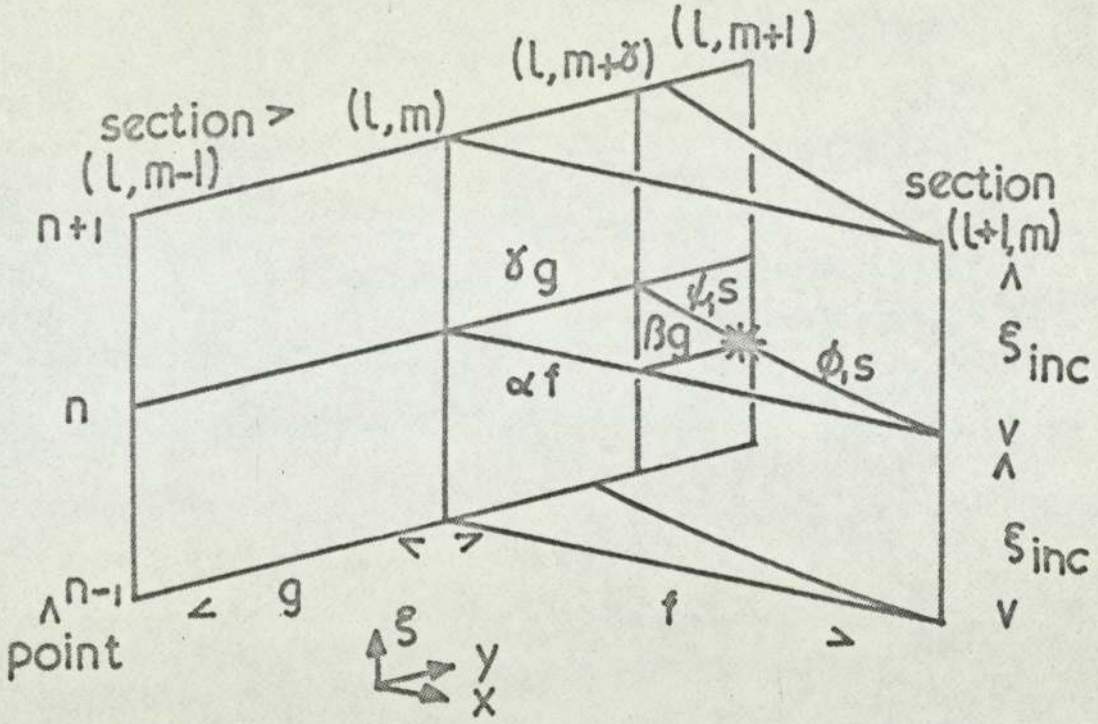
FIG(4.1.2) Subdivision of increments near the wall ($\lambda=2, \omega=3, N=10, M=8$).



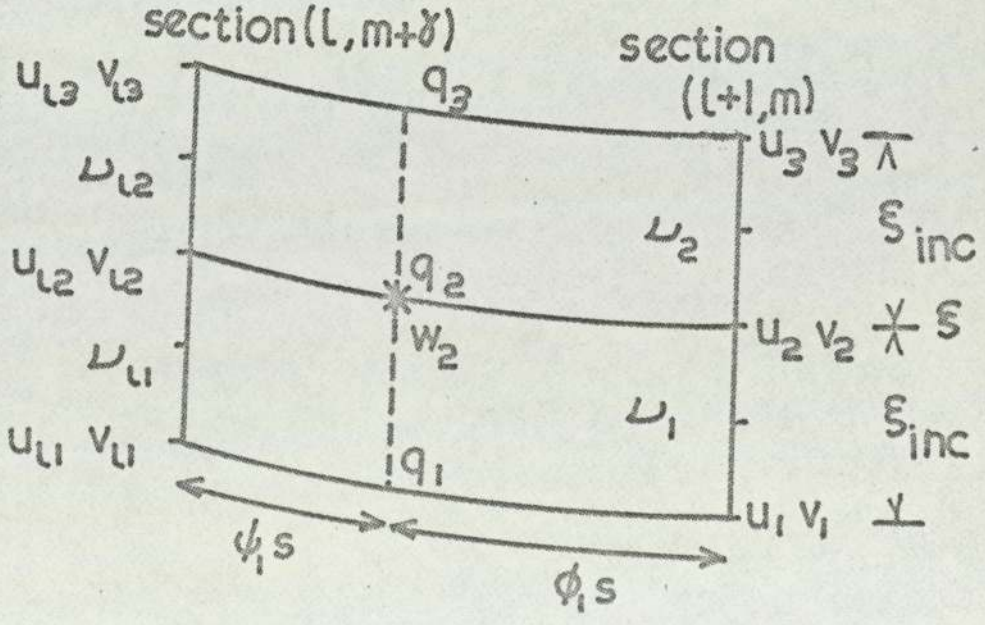
FIG(4.2.1) An adjustable grid.



FIG(4.2.2) An illustration of the proposed solution mesh.



FIG(4.6.1) The streamlines through points on section (l+1,m) and the point (*) at which the momentum equations are to be approximated.



FIG(4.6.2) Values necessary for the finite difference approximations to the momentum equations.

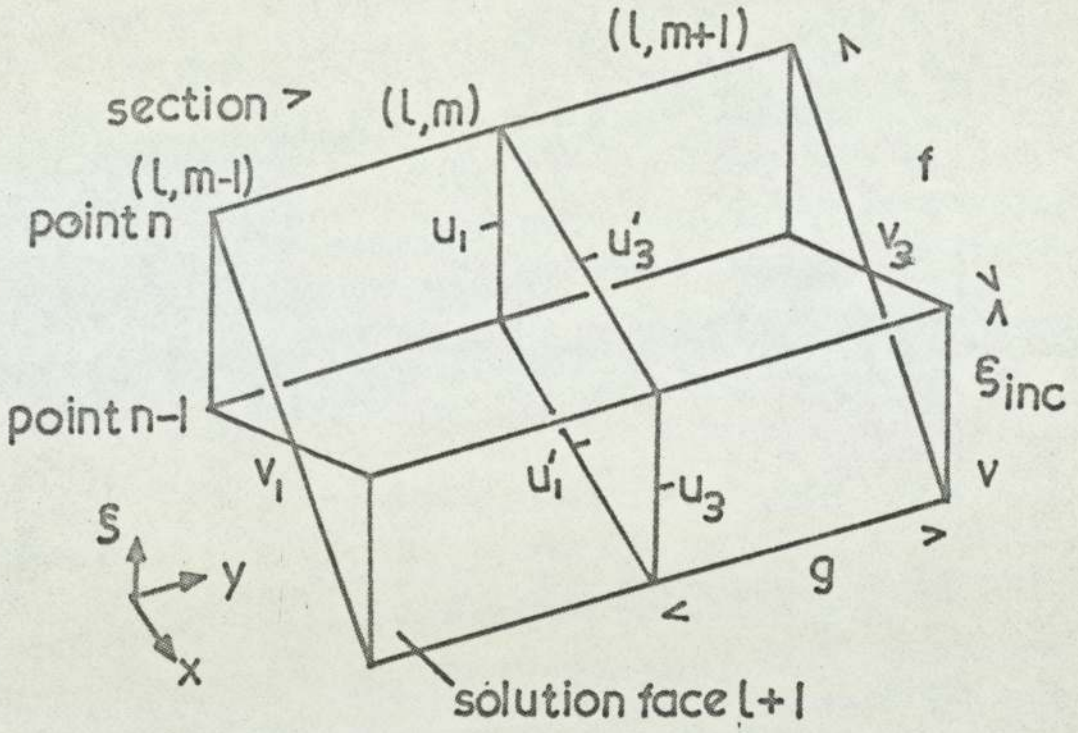


FIG (4.7.1) Values necessary for the finite difference approximation to the continuity equation.

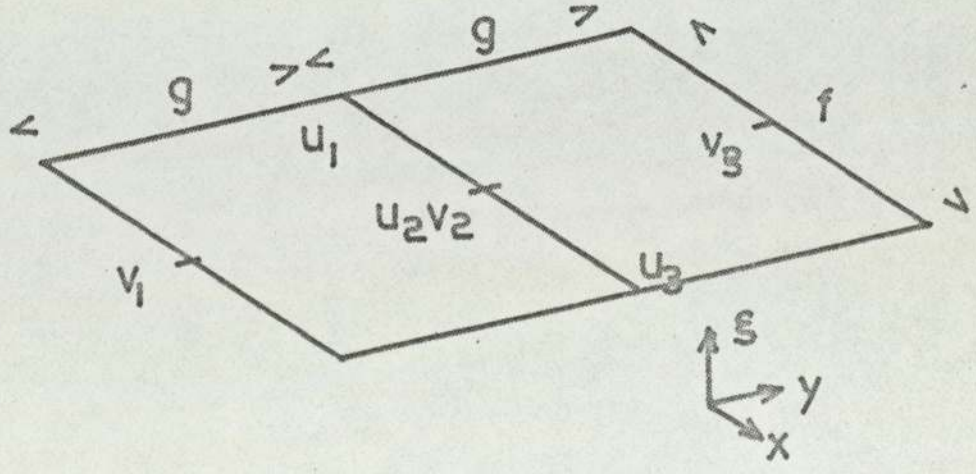


FIG (4.7.2) Values necessary for the finite difference approximation to the continuity equation at the log-point.

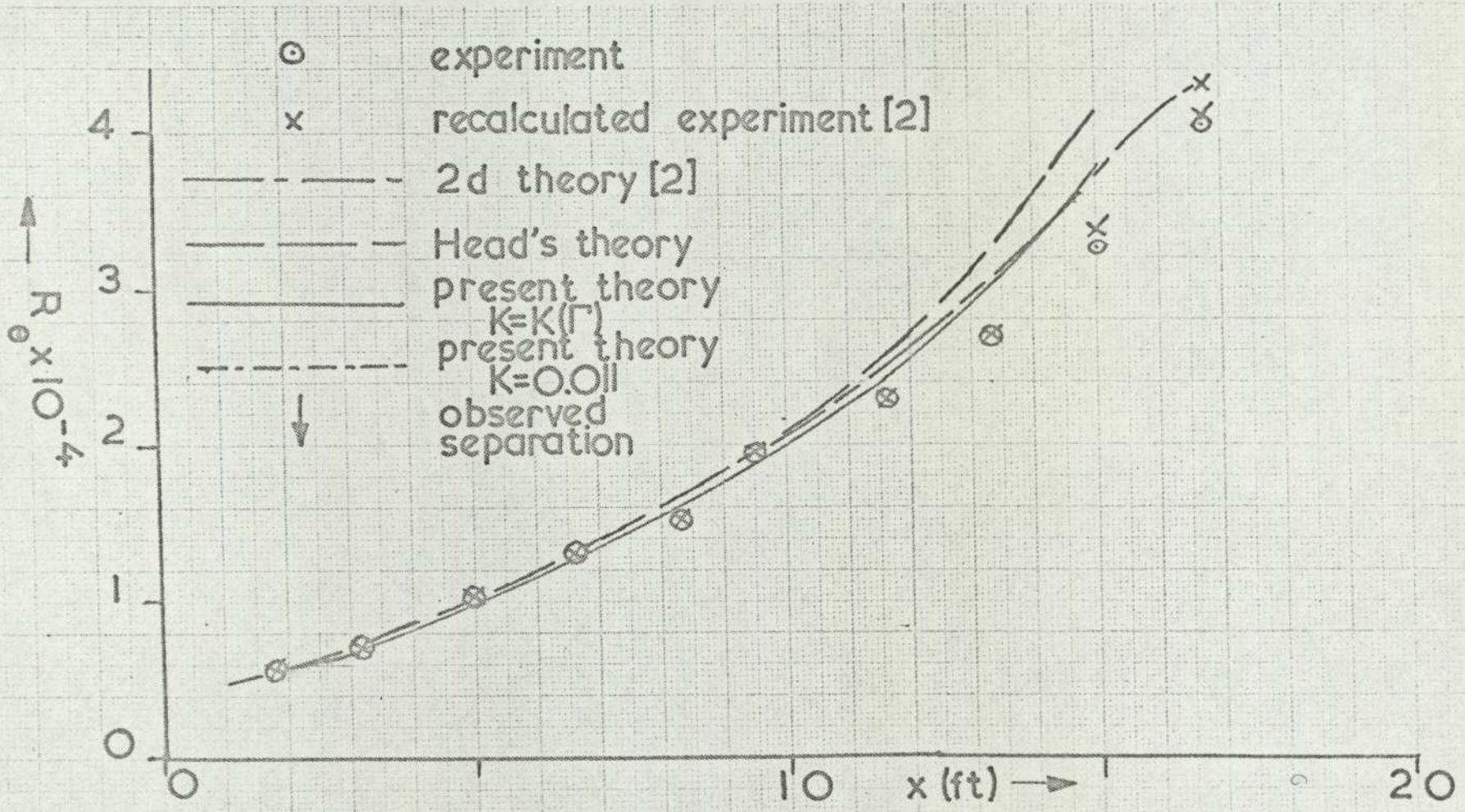


FIG (5.1.1) Schubauer and Spangenberg [26], experiment 'C'

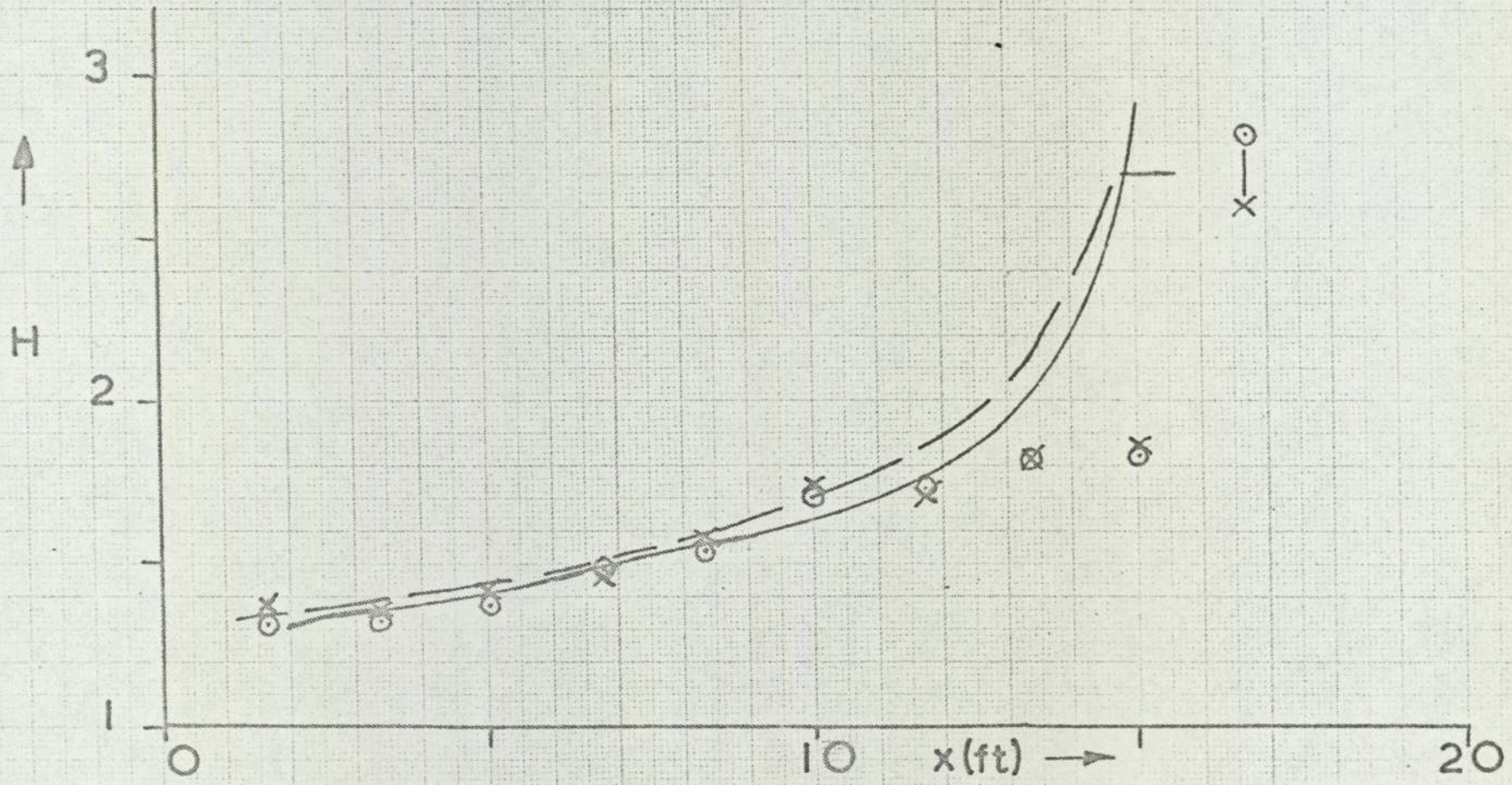


FIG (5.1.2) Schubauer and Spangenberg [26], experiment 'C'
Notation as in figure (5.1.1)

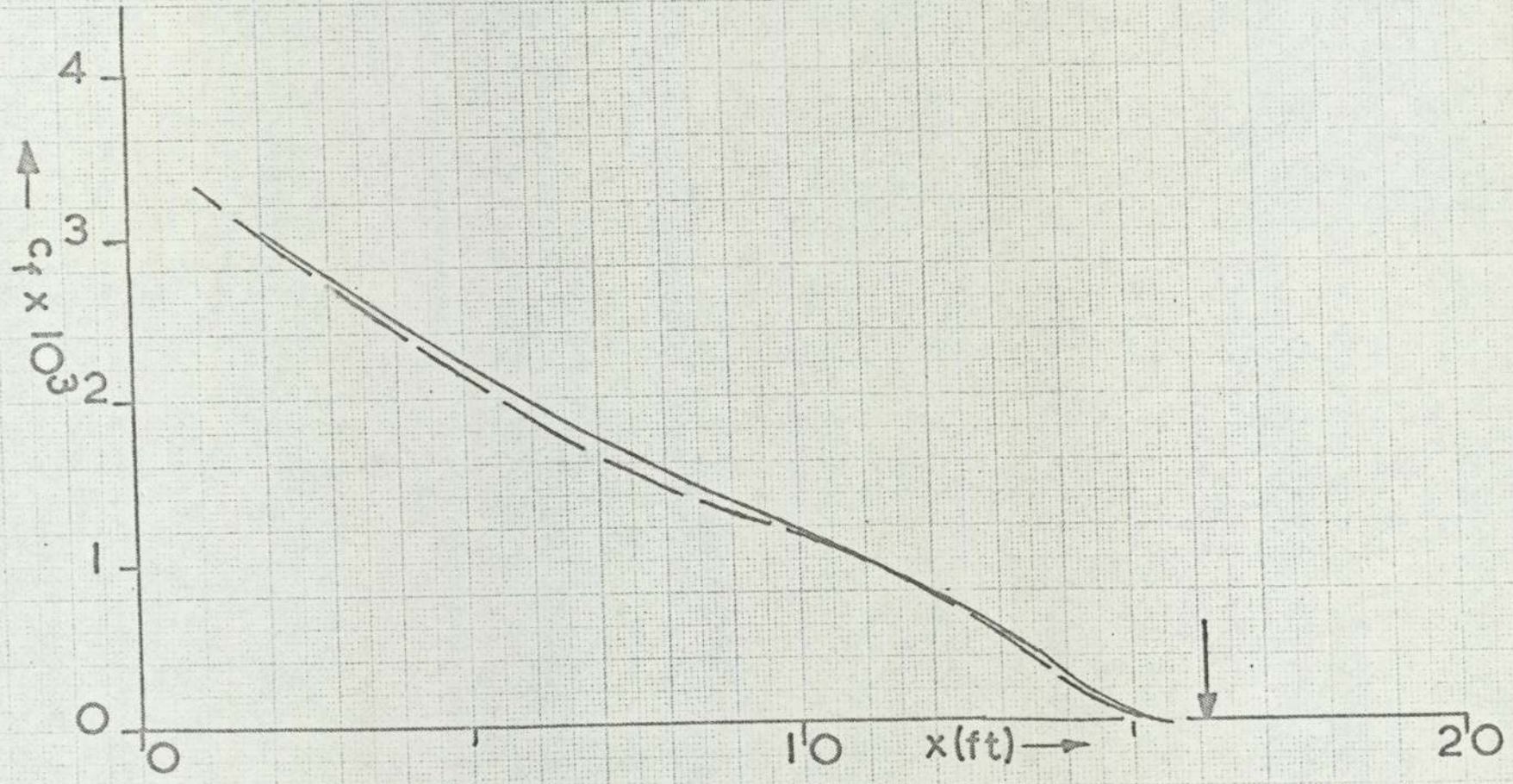


FIG (5.1.3) Schubauer and Spangenberg [26], experiment 'C'
Notation as in figure (5.1.1)

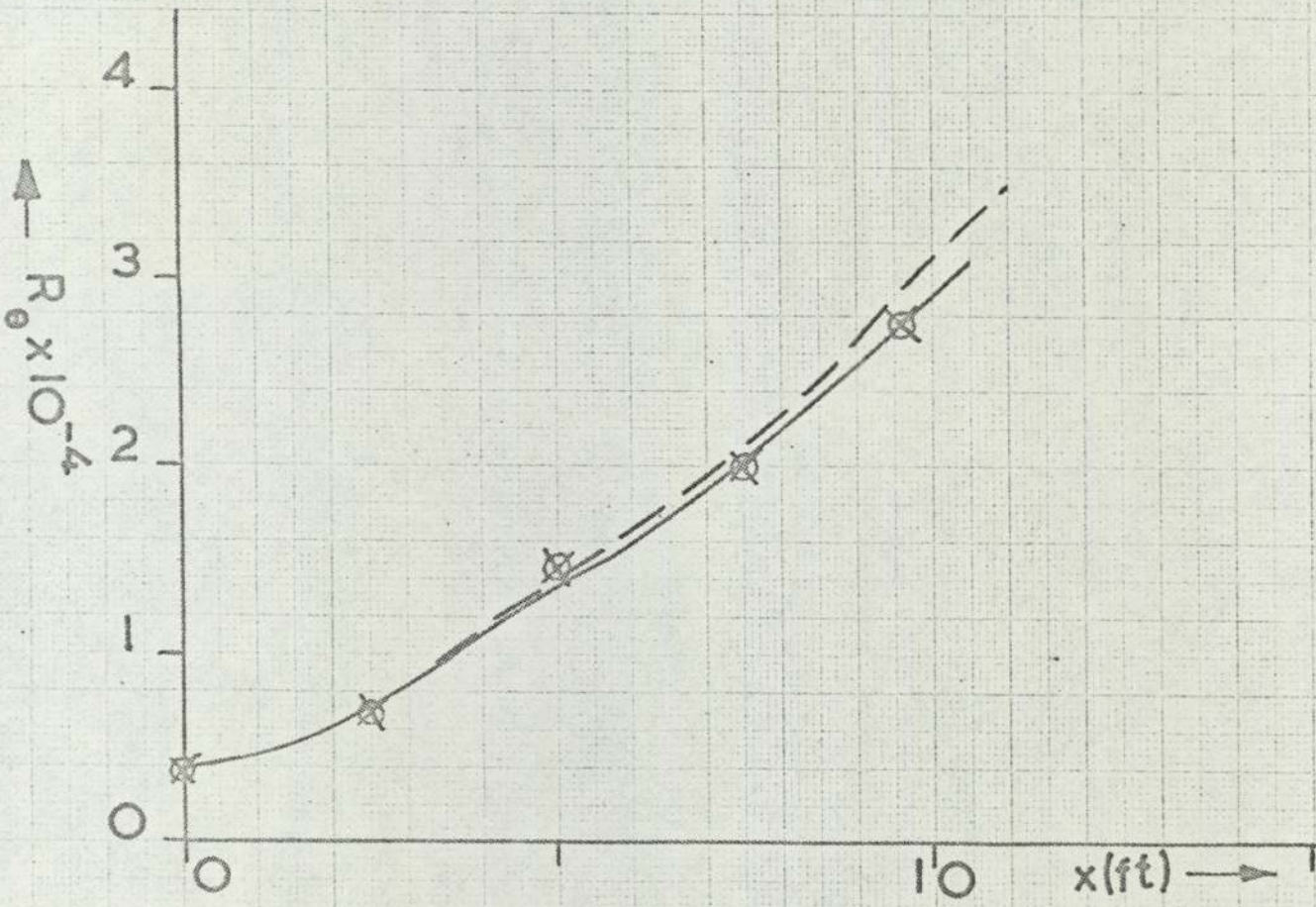


FIG (5.14) Schubauer and Spangenberg [26], experiment 'D'
Notation as in figure (5.1.1)

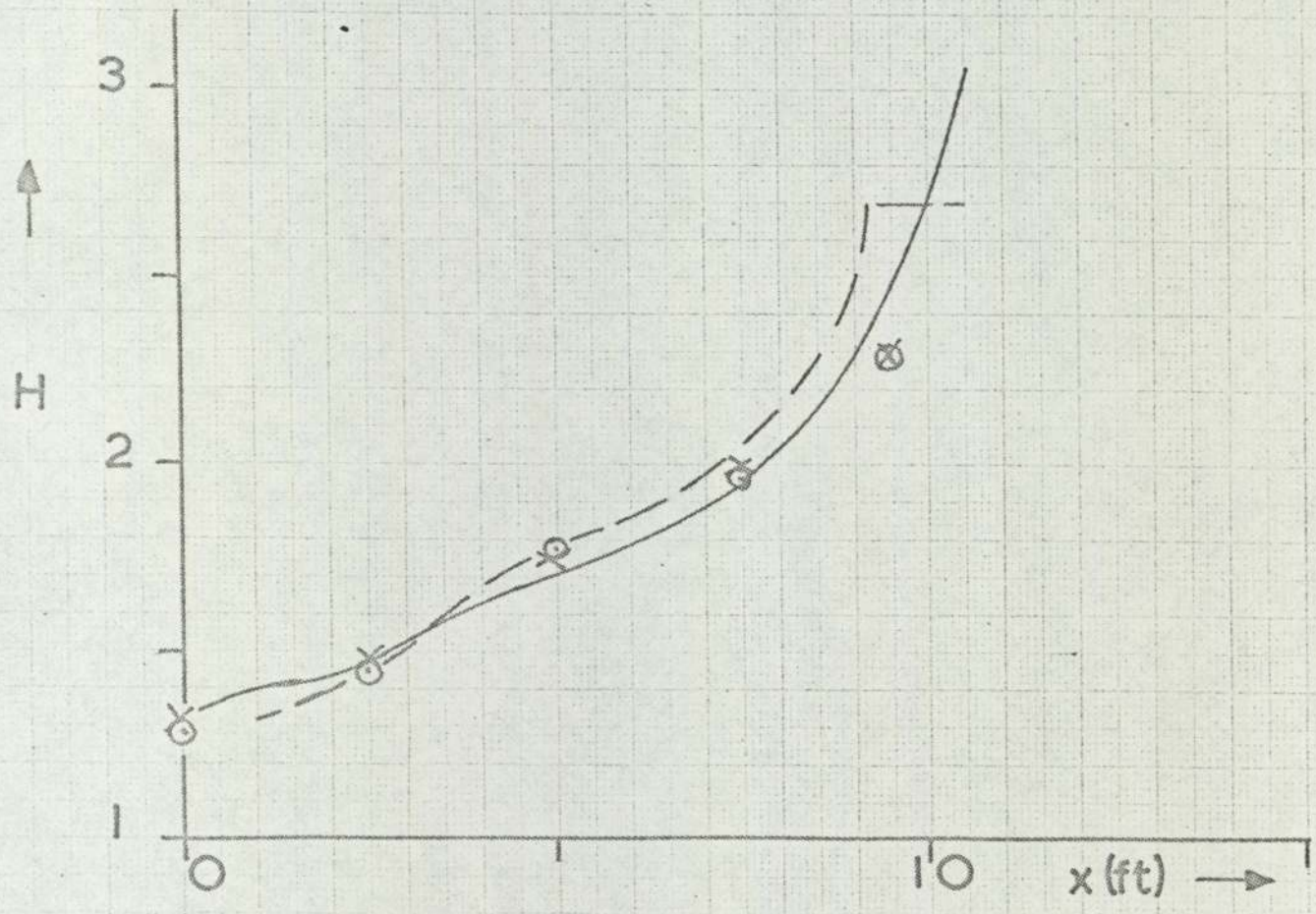


FIG (5.1.5) Schubauer and Spangenberg [26], experiment 'D'
Notation as in figure (5.1.1)

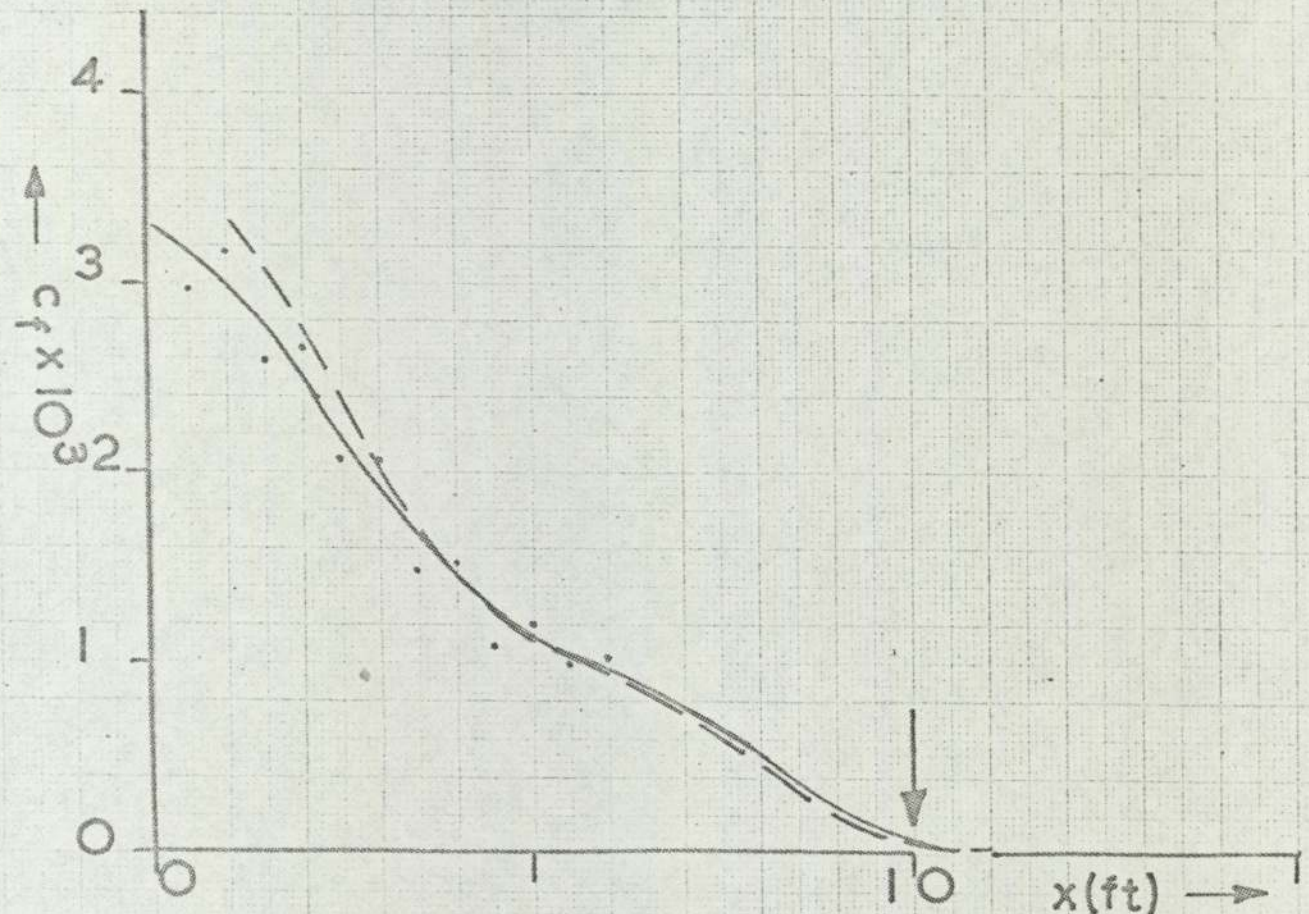


FIG (5.1.6) Schubauer and Spangenberg [26], experiment 'D'
Notation as in figure (5.1.1)

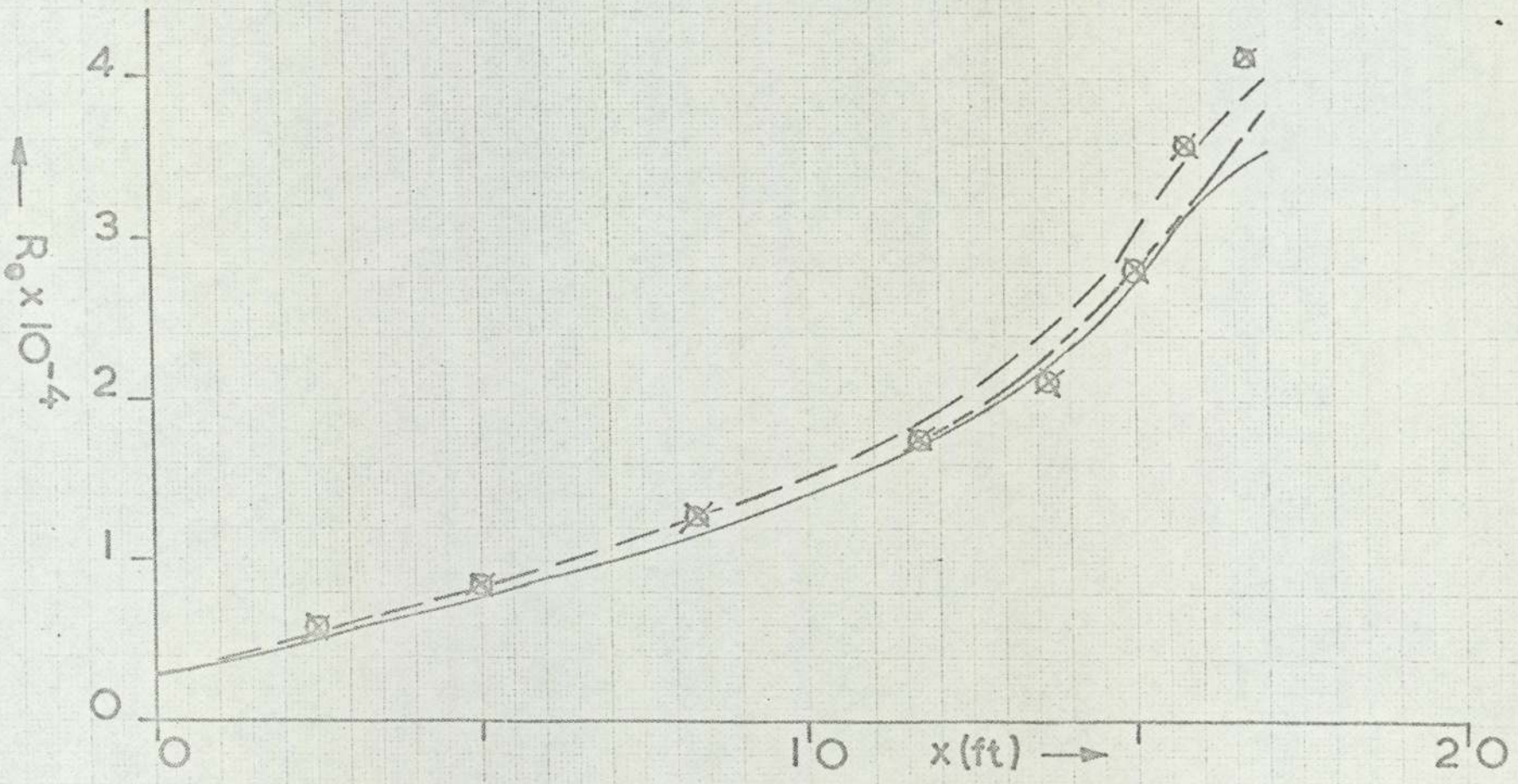


FIG (5.1.7) Schubauer and Spangenberg [26], experiment 'E'
Notation as in figure (5.1.1)

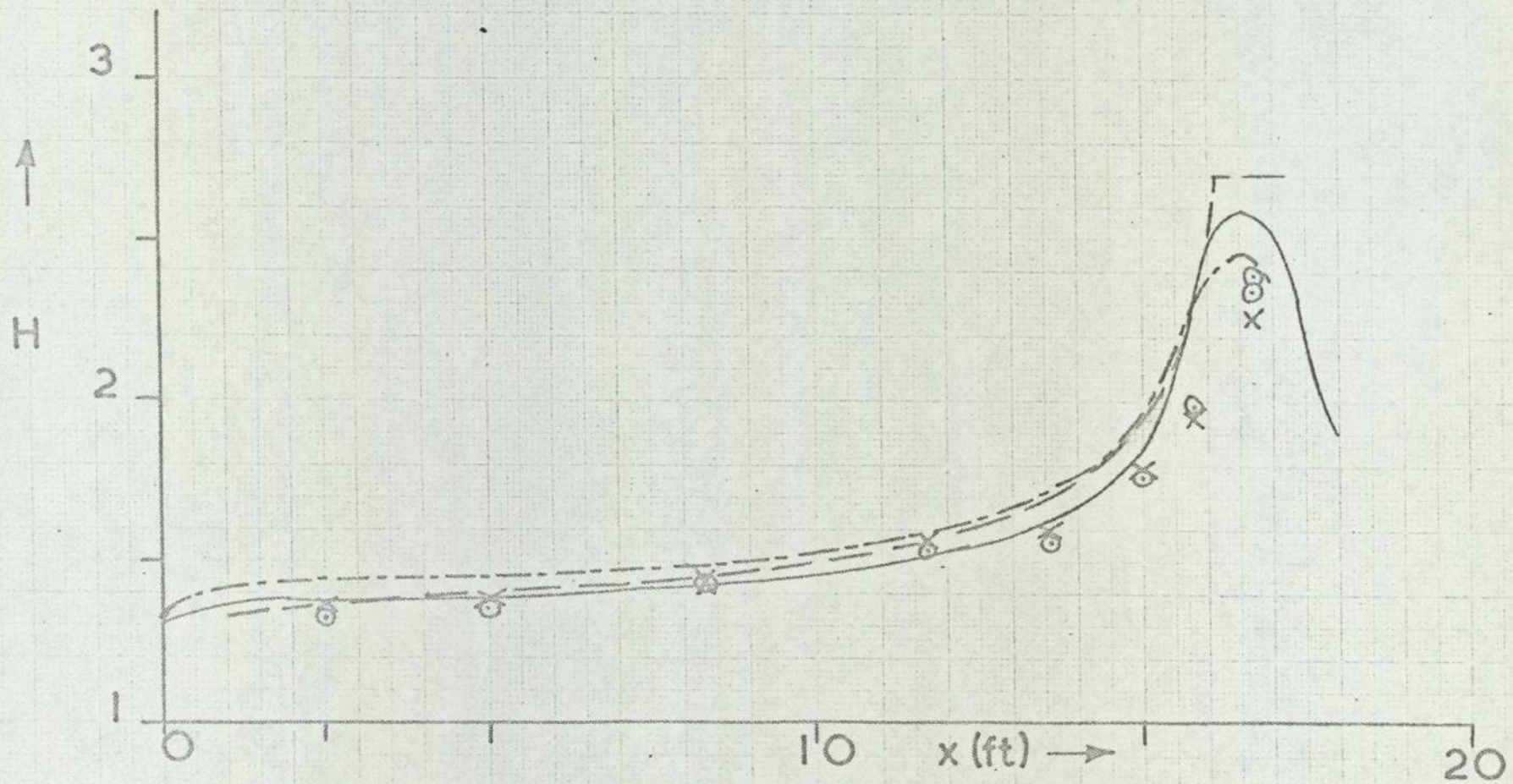


FIG (5.1.8) Schubauer and Spangenberg [26], experiment 'E'
Notation as in figure (5.1.1)

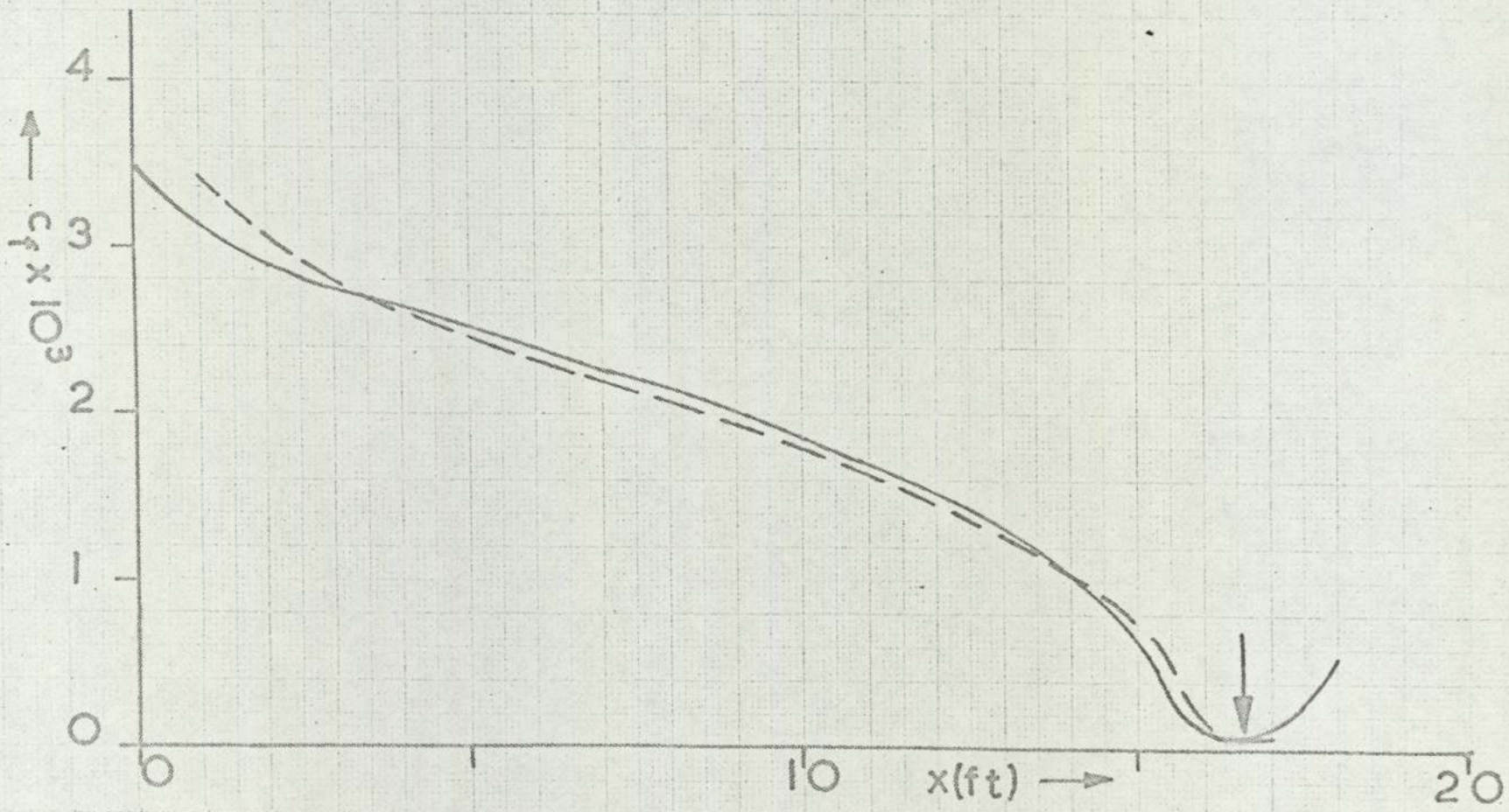
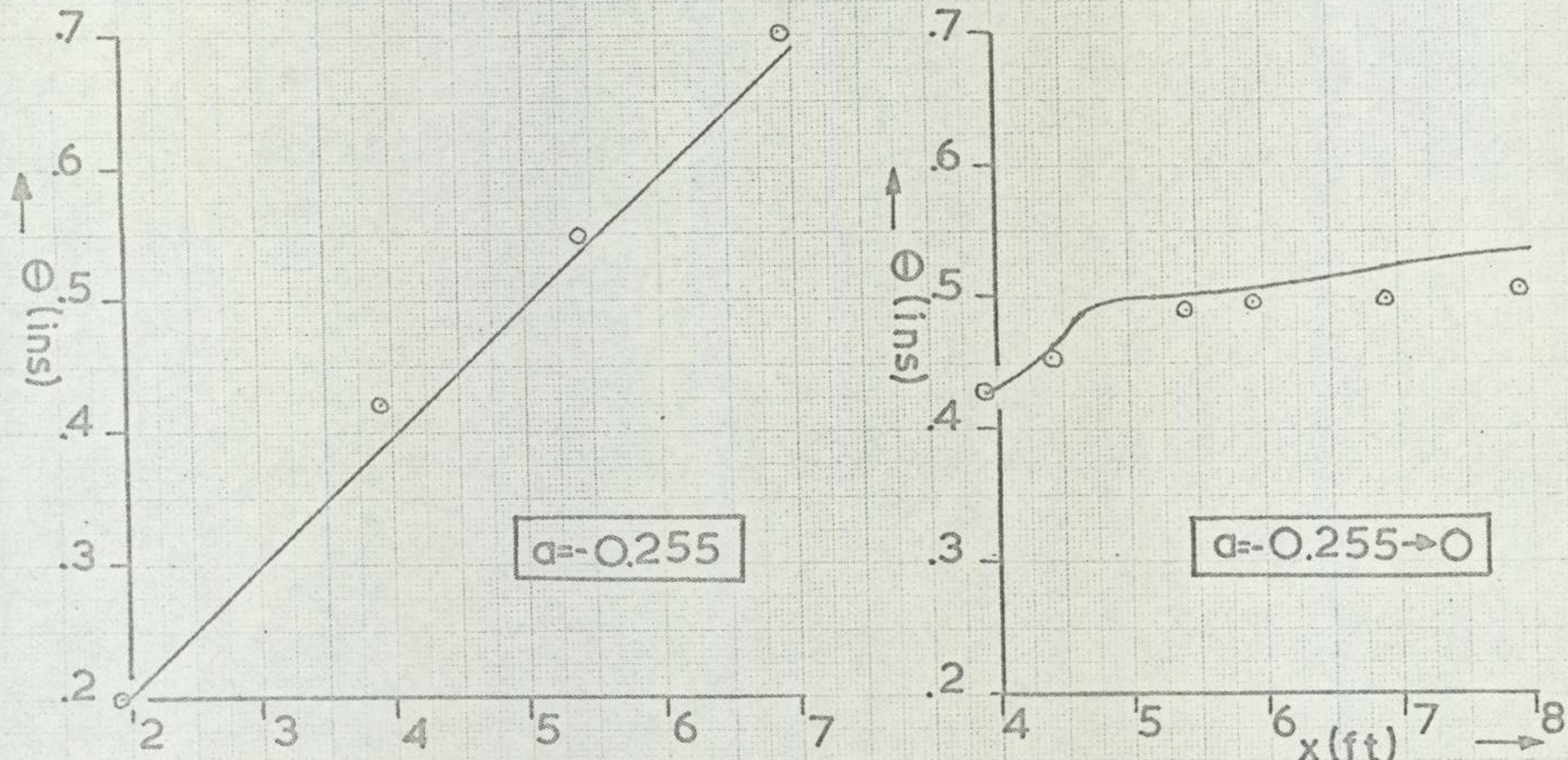
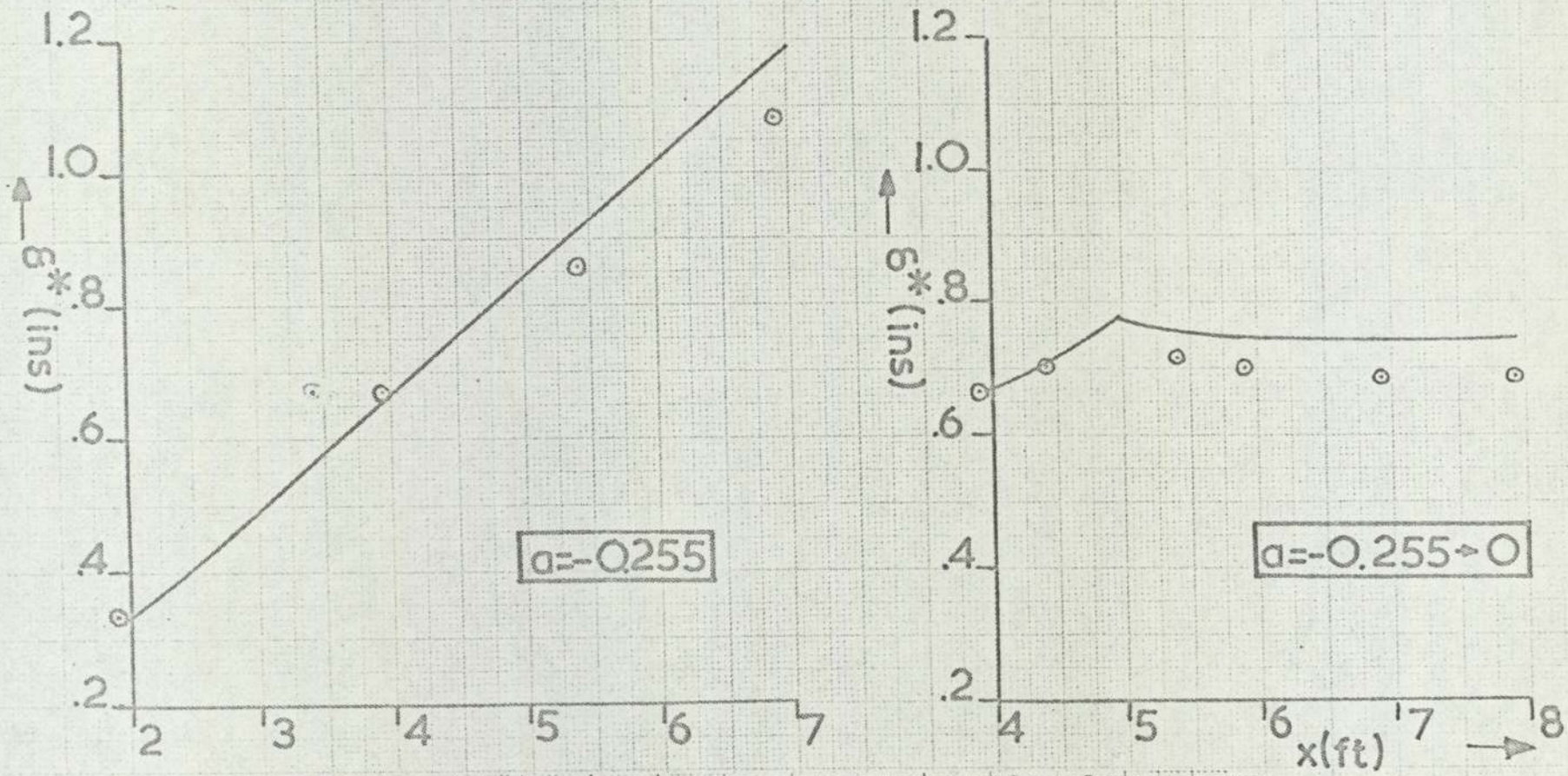


FIG (5.1.9) Schubauer and Spangenberg [26], experiment 'E'
Notation as in figure (5.1.1)



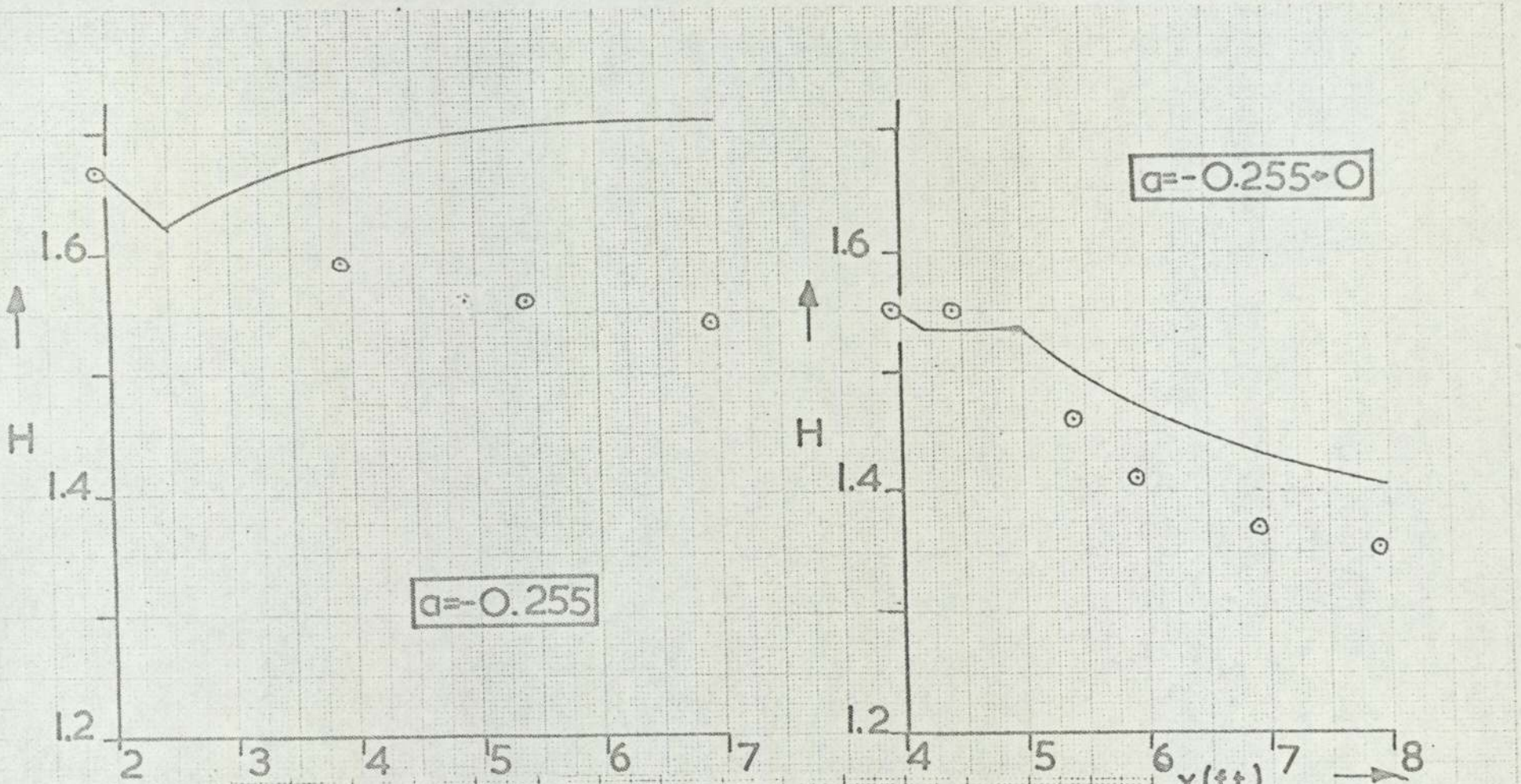
FIG(5.21) Bradshaw and Ferriss [27]

⊙ experiment —— present theory



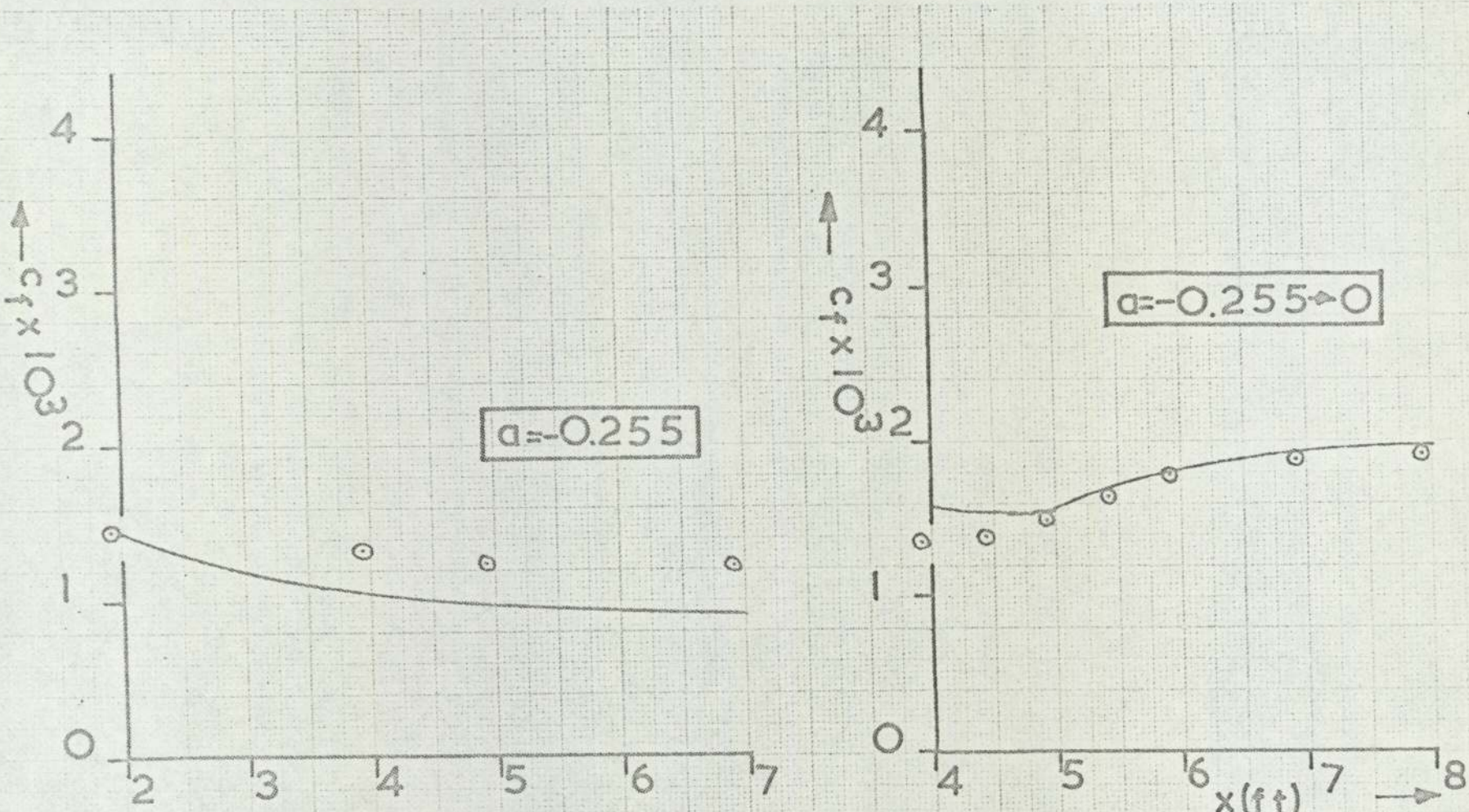
FIG(5.22) Bradshaw and Ferriss [27]

⊙ experiment —— present theory



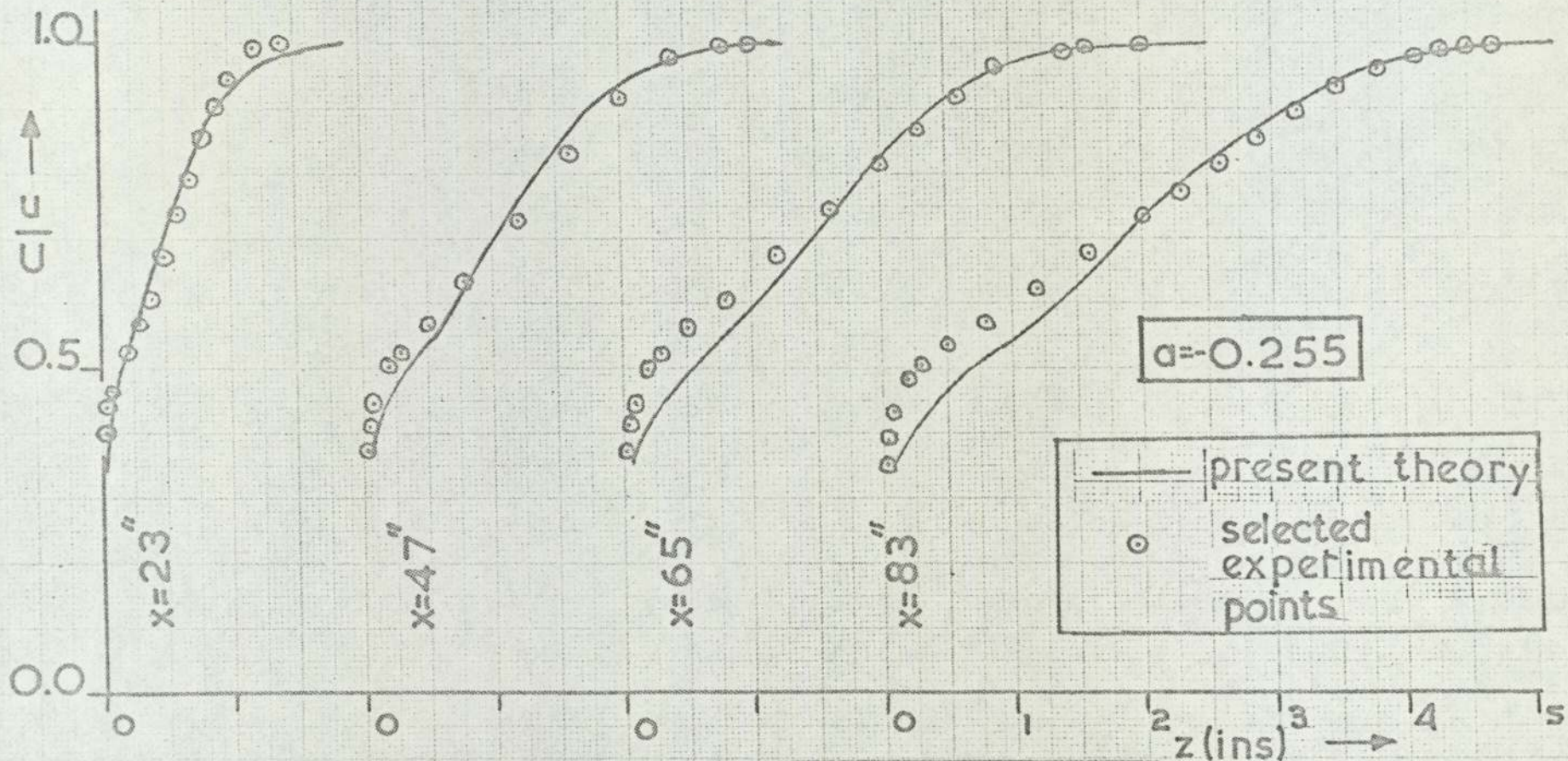
FIG(5.23) Bradshaw and Ferriss [27]

⊙ experiment —— present theory

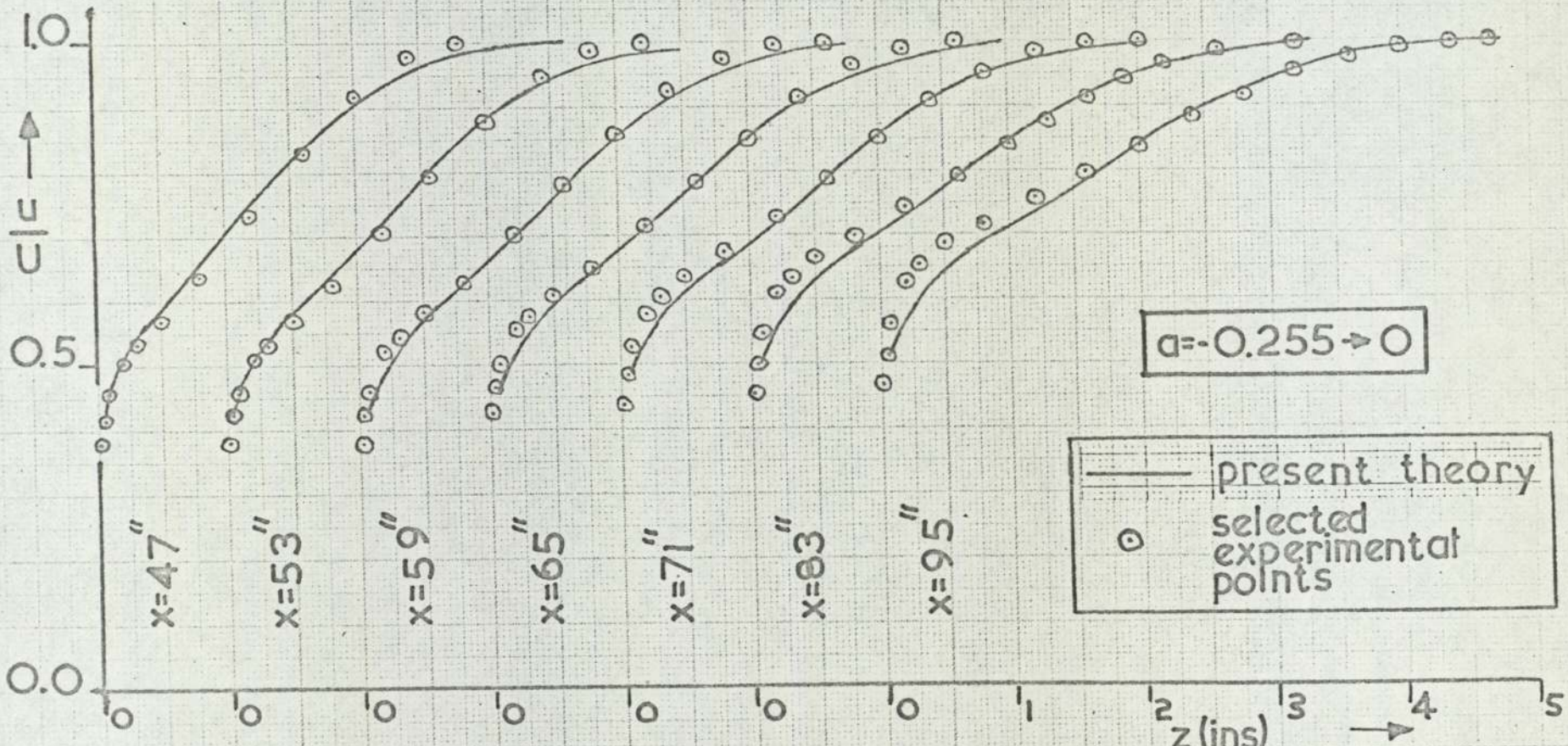


FIG(5.24) Bradshaw and Ferriss [27]

⊙ experiment —— present theory



FIG(525) Bradshaw and Ferriss [27]



FIG(5.26) Bradshaw and Ferriss [27]

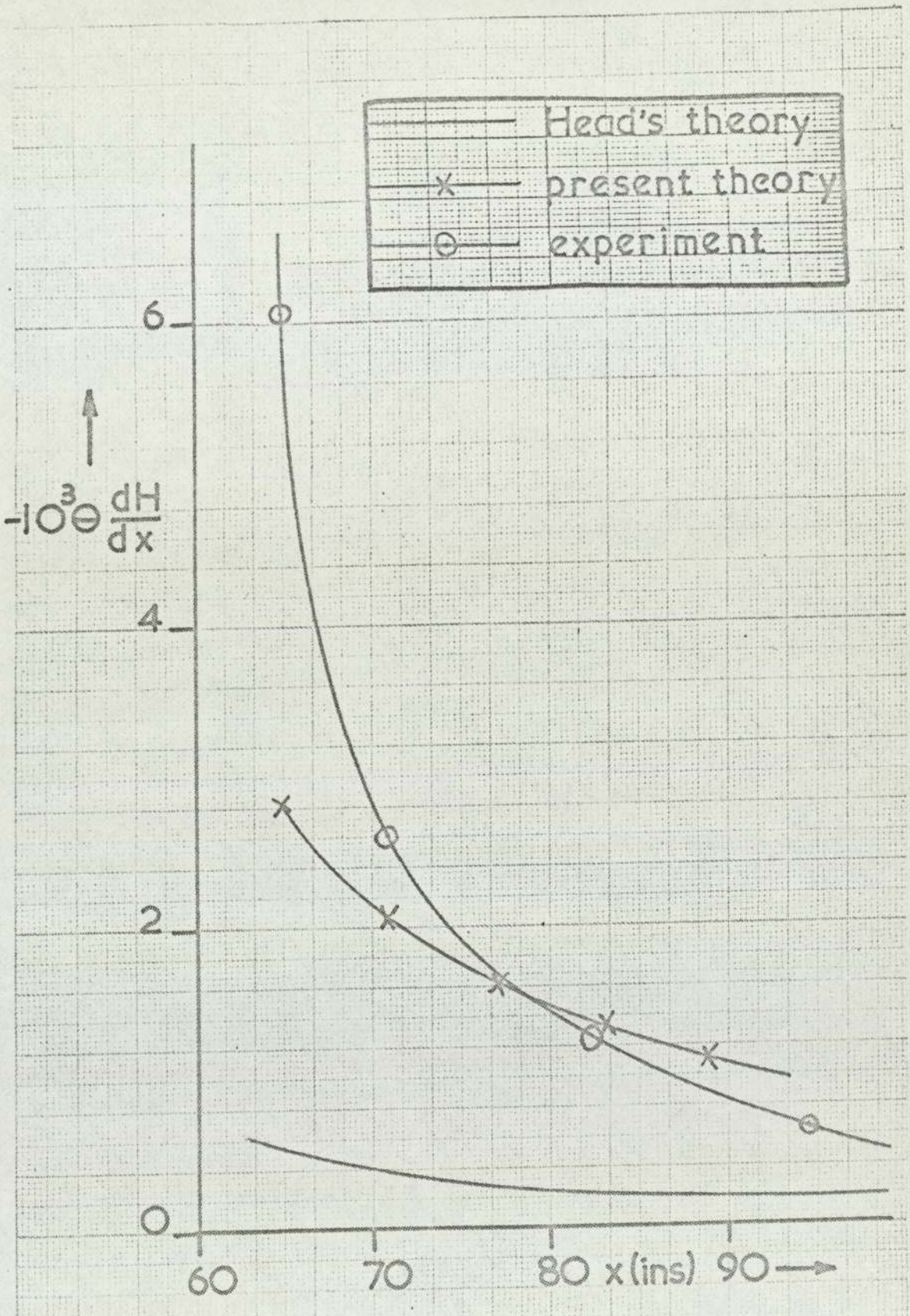


FIG (5.27) Bradshaw and Ferriss [27]

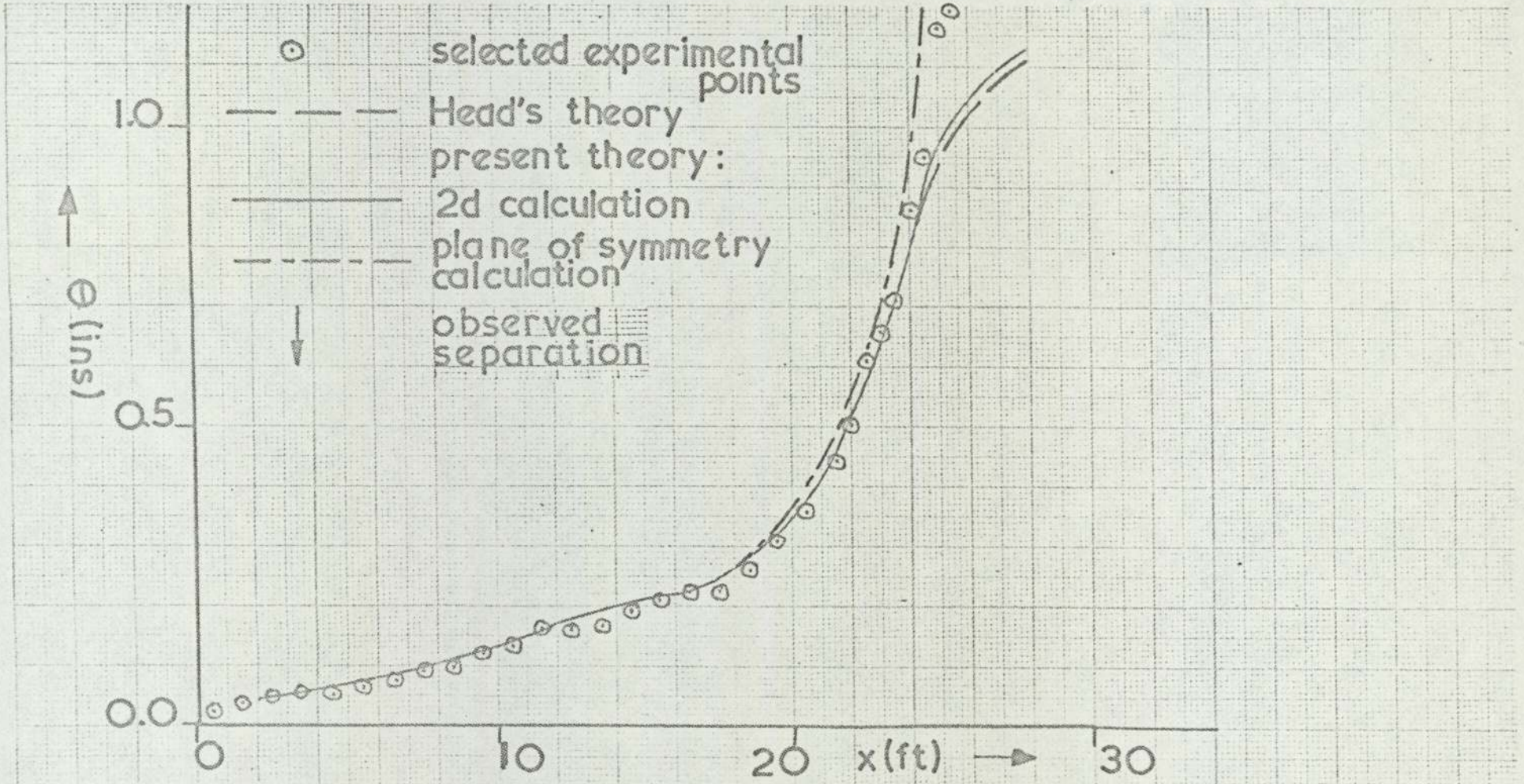


FIG (5.3.1) Schubauer and Klebanoff [28]

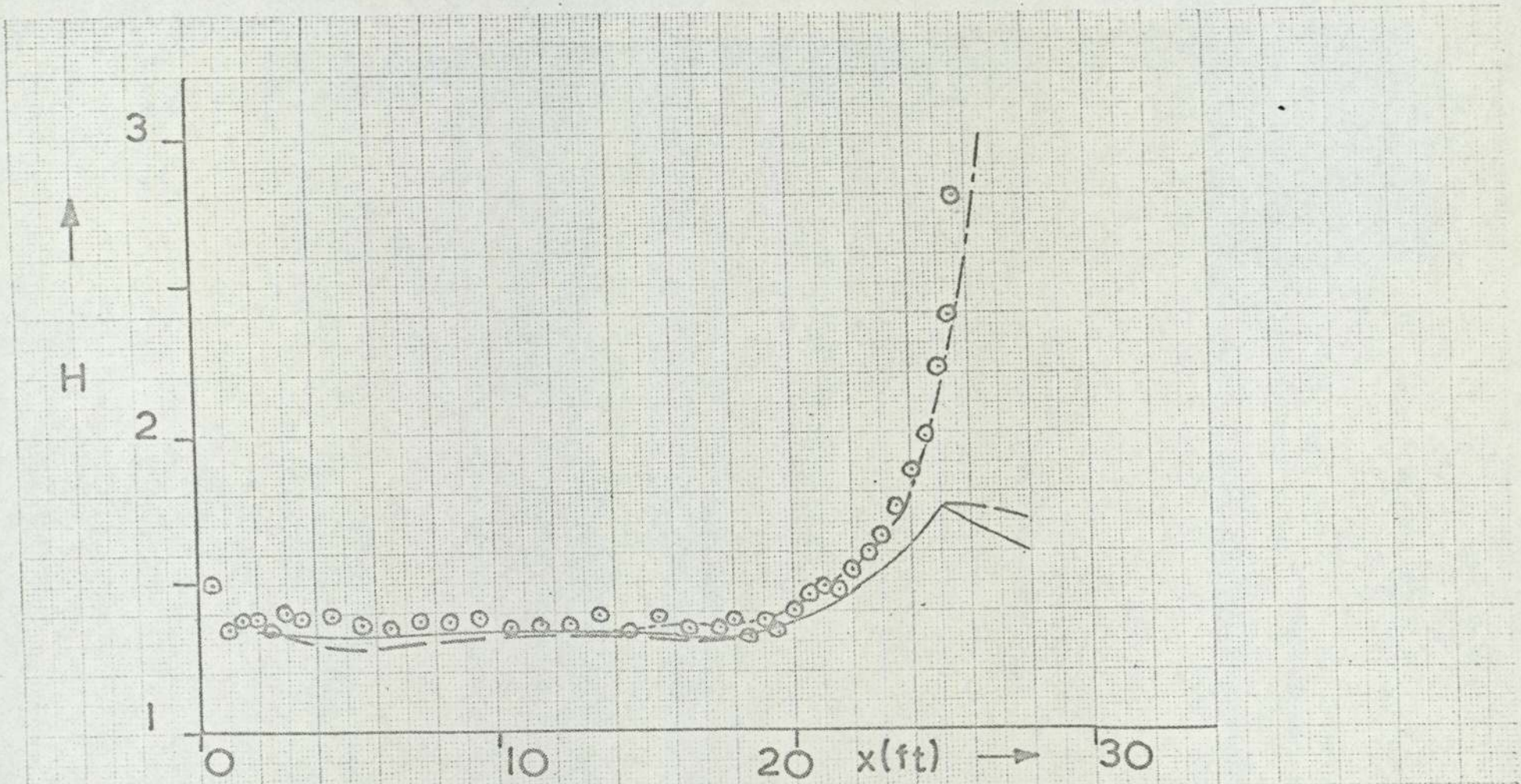


FIG (5.3.2) Schubauer and Klebanoff [28]
Notation as in figure (5.3.1)

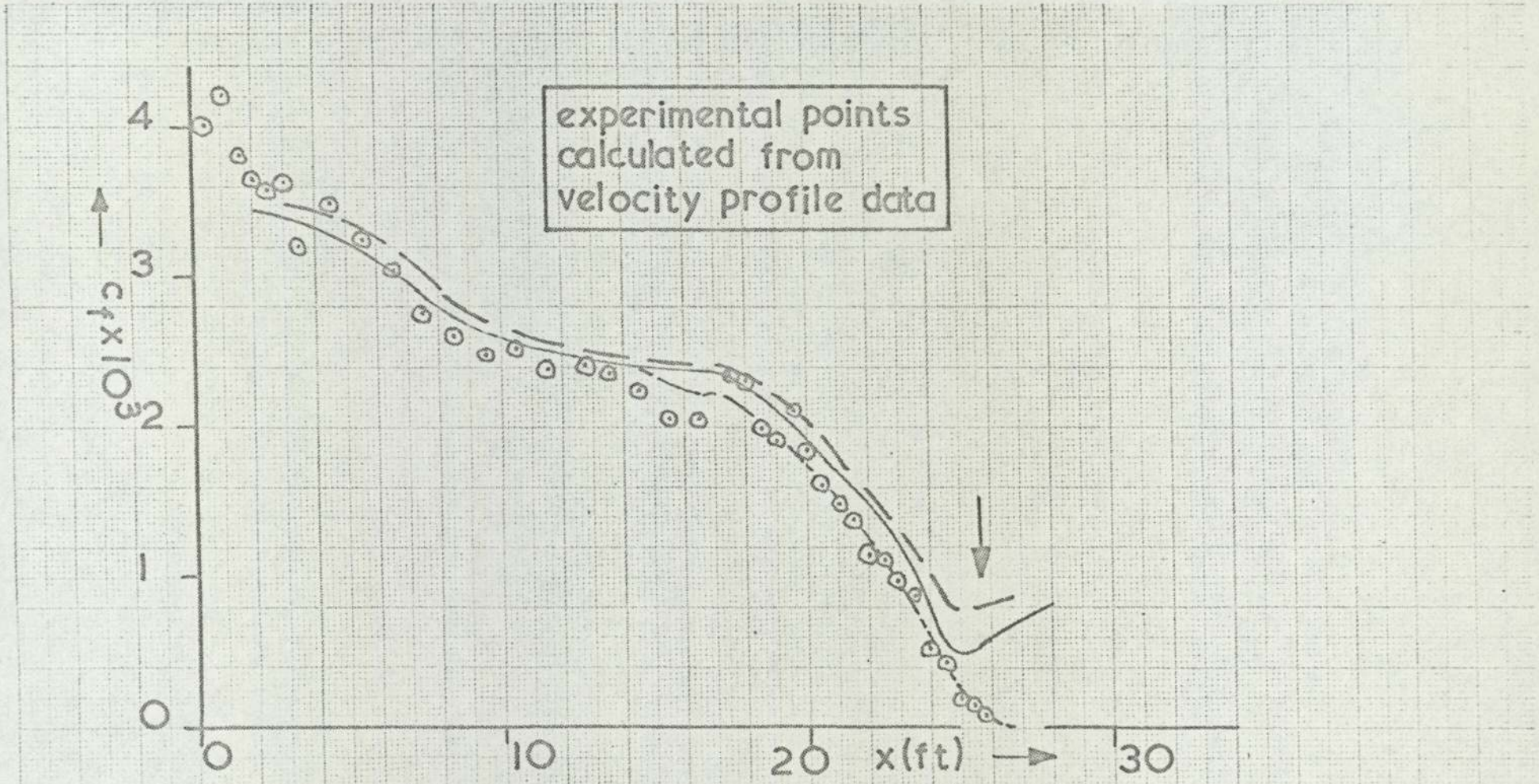


FIG (5.3.3) Schubauer and Klebanoff [28]
Notation as in figure (5.3.1)

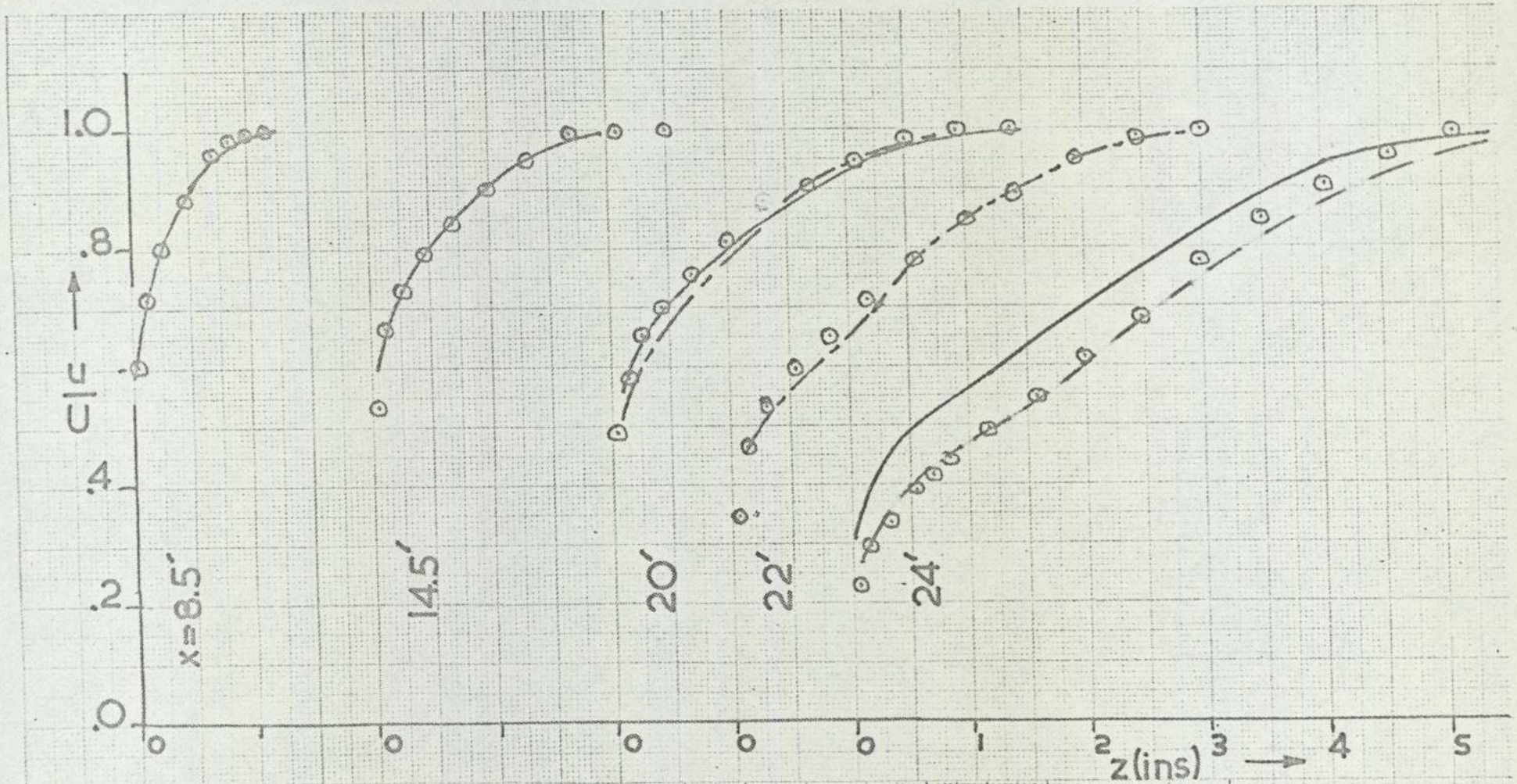
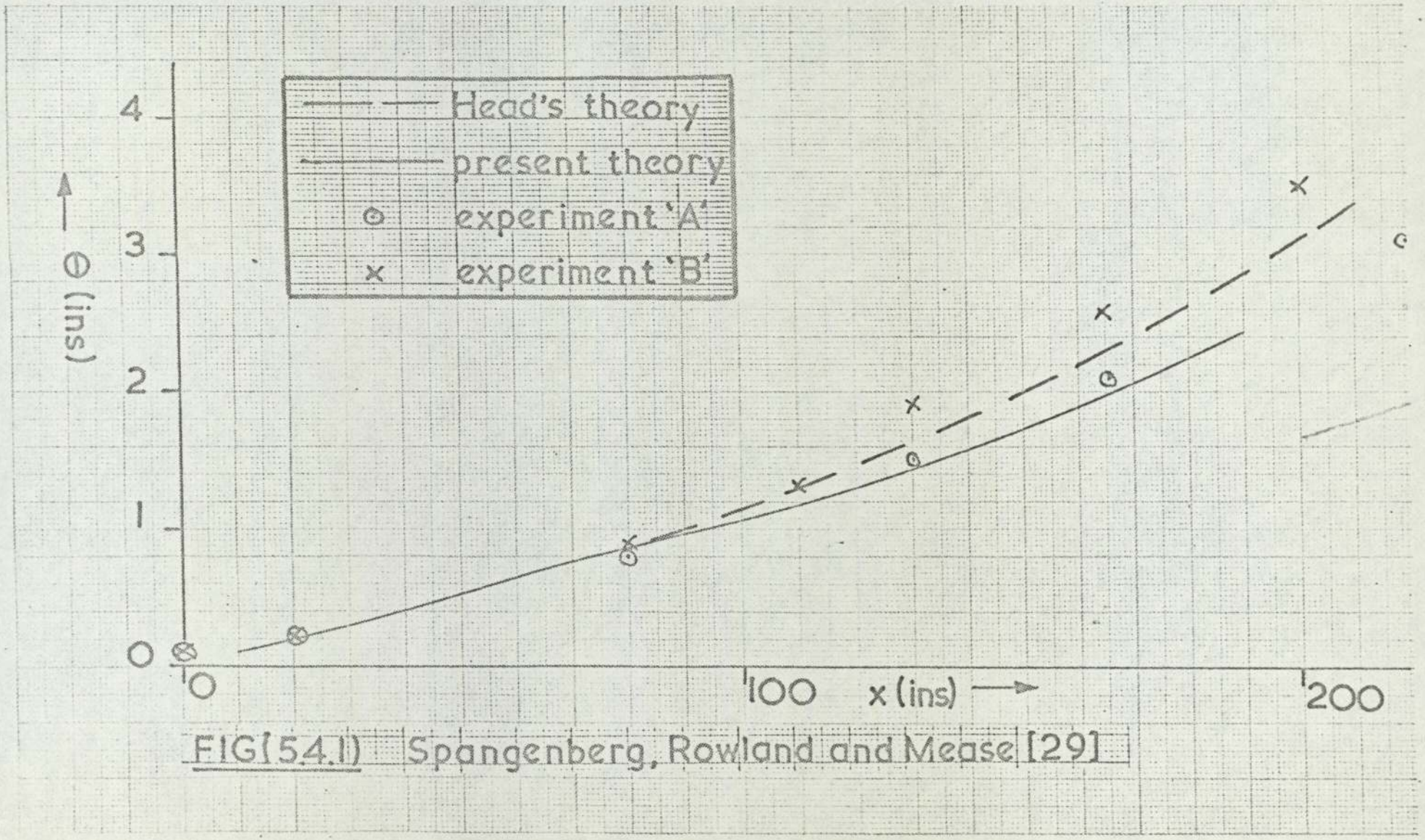
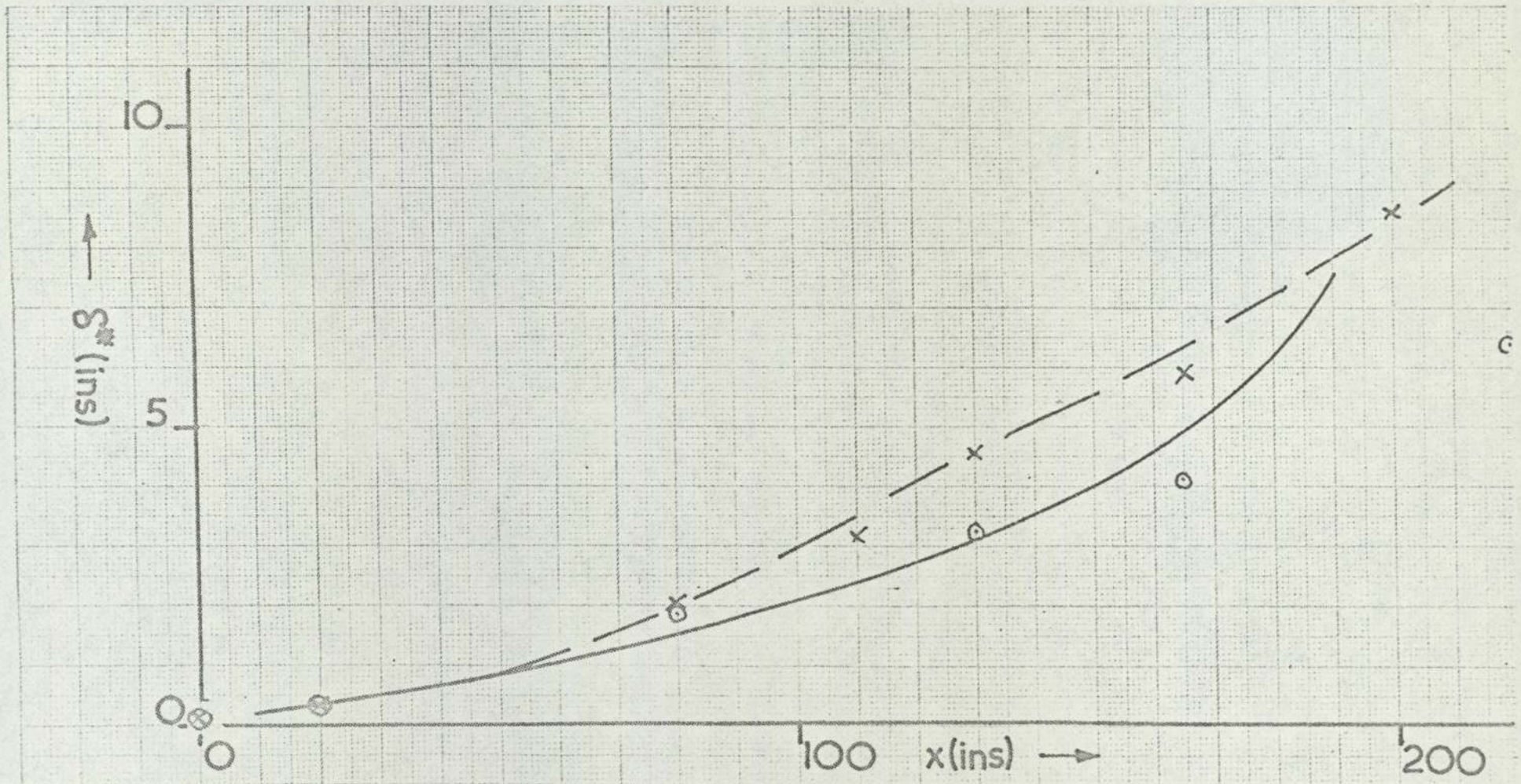


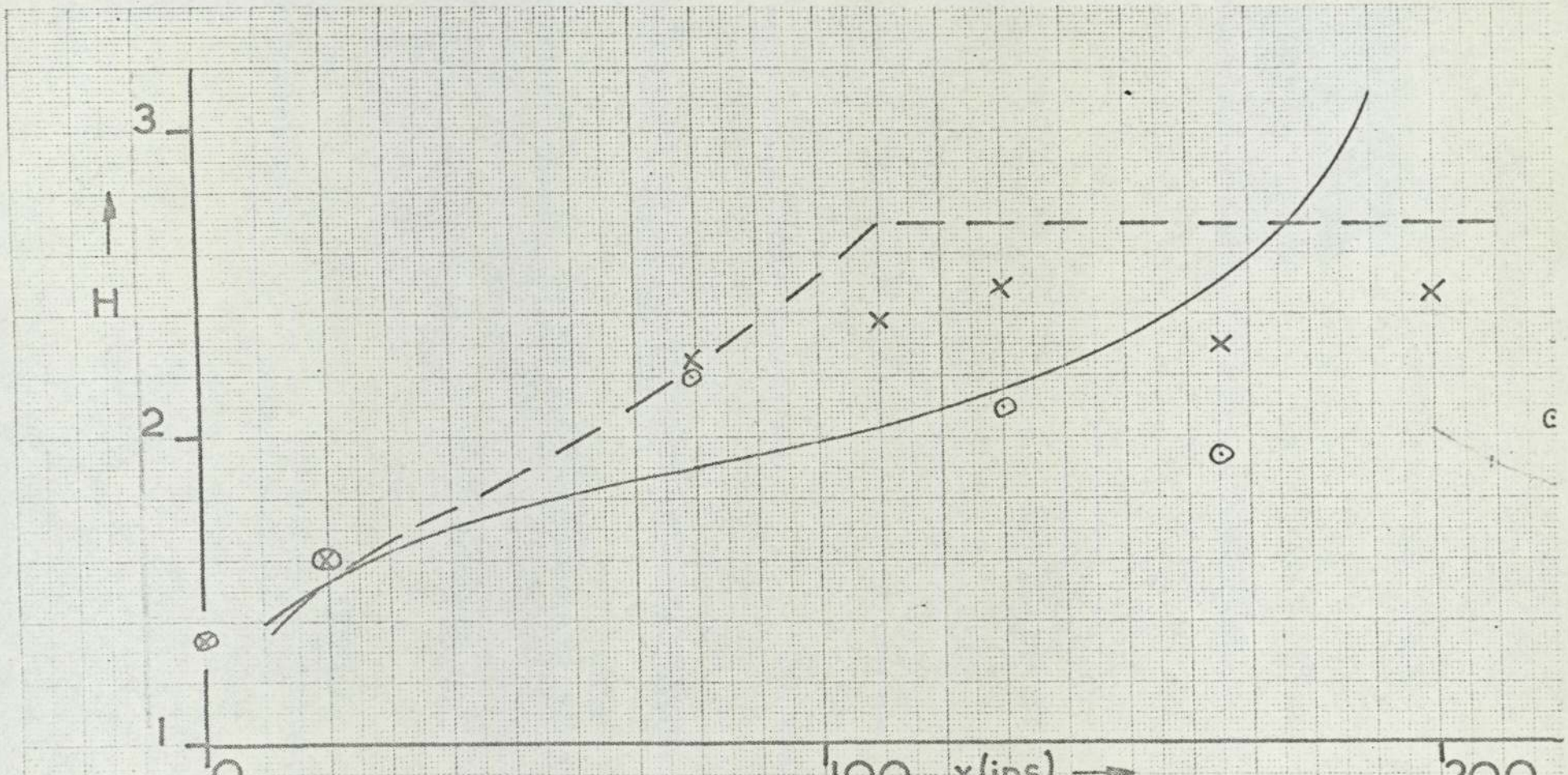
FIG (5.34) Schubauer and Klebanoff [28]
Notation as in figure (5.3.1)



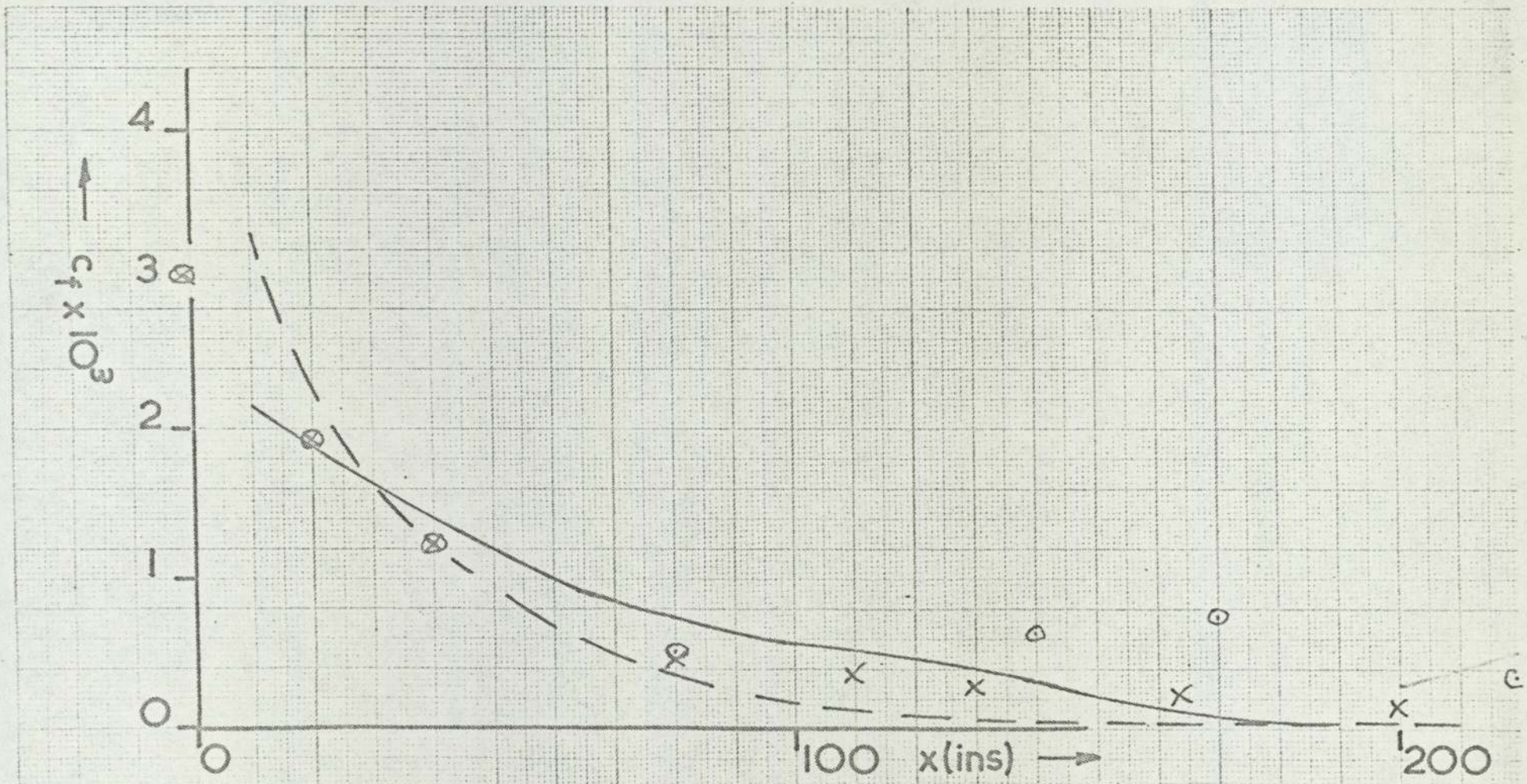
FIG(5.4.1) Spangenberg, Rowland and Mease [29]



FIG(5.4.2) Spangenberg, Rowland and Mease [29]
Notation as in figure (5.4.1)



FIG(5.4.3) Spangenberg, Rowland and Mease [29]
Notation as in figure (5.4.1)



FIG(5.4.4) Spangenberg, Rowland and Mease [29]
Notation as in figure (5.4.1)

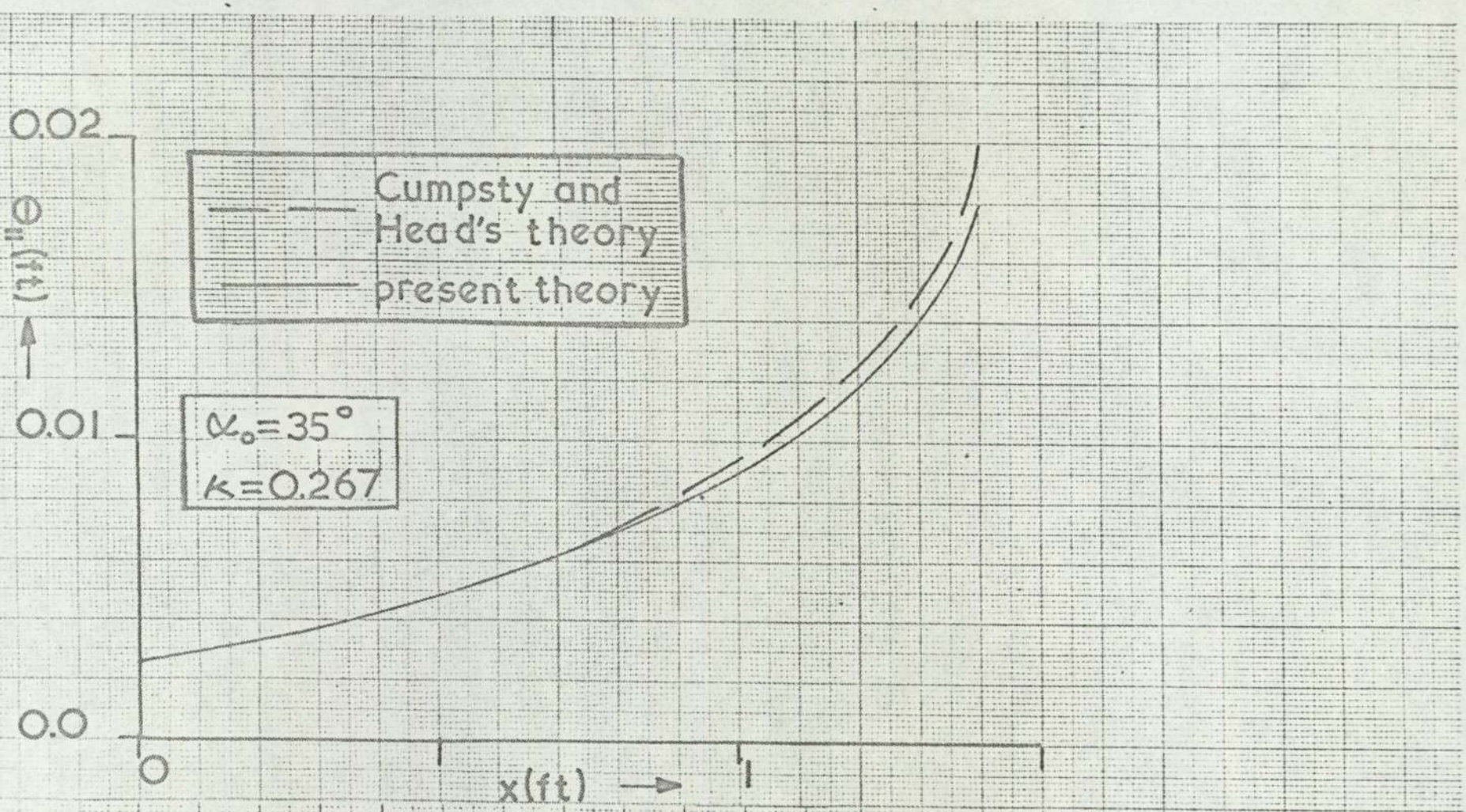


FIG (6.1.1) Cumpsty and Head [30]
The hypothetical infinite swept wing

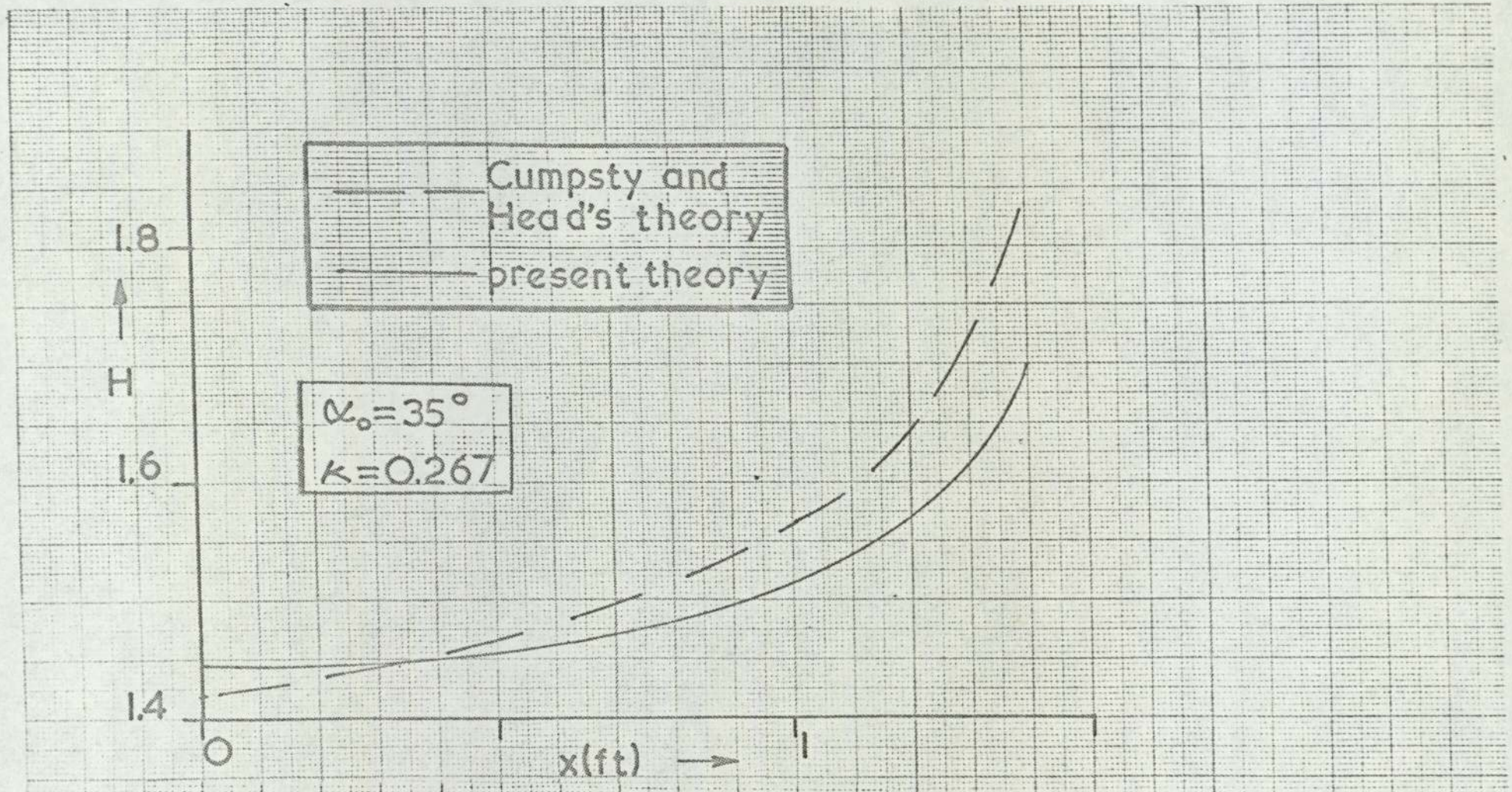


FIG (6.1.2) Cumpsty and Head [30]
The hypothetical infinite swept wing

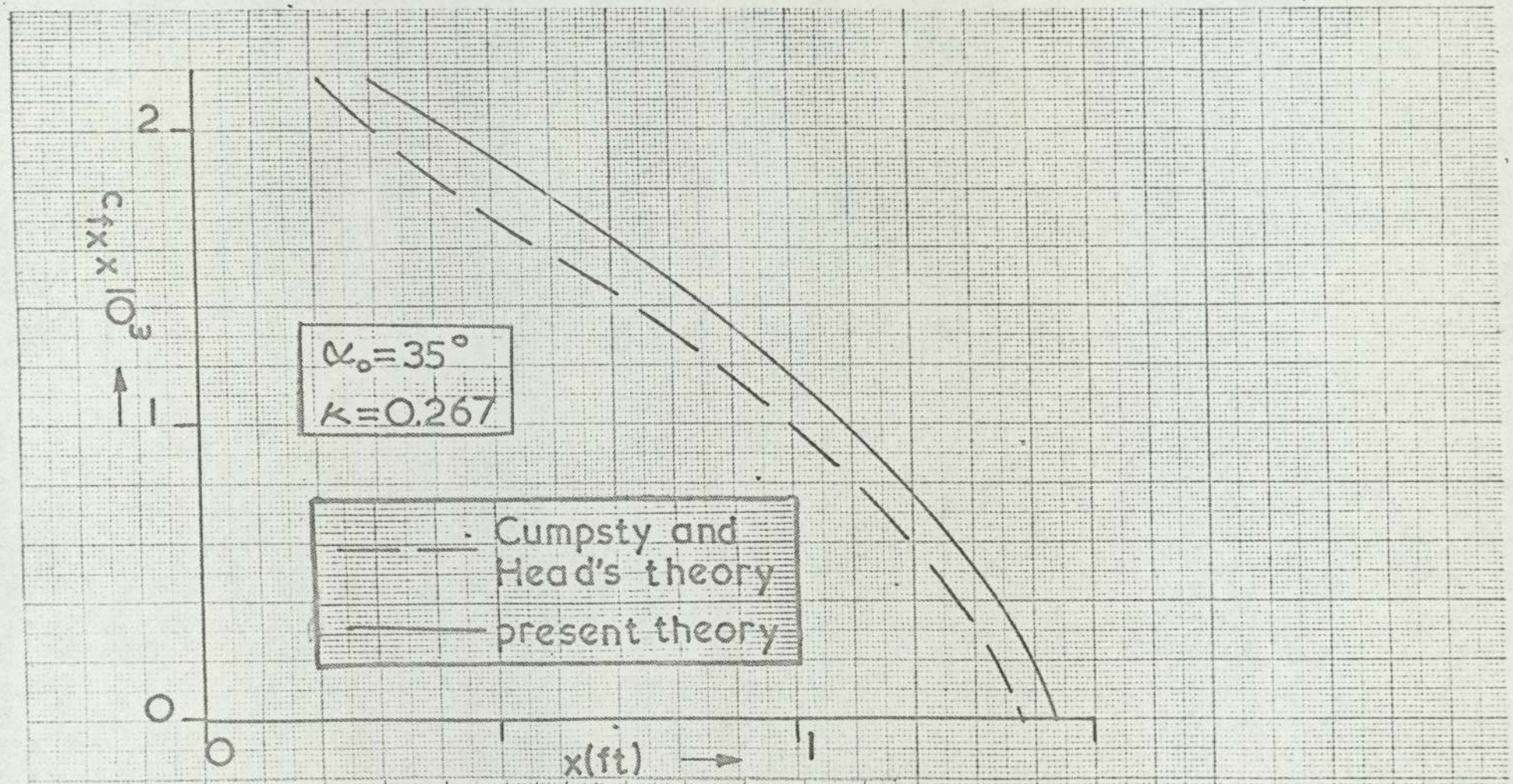


FIG (6.13) Cumpsty and Head [30]
The hypothetical infinite swept wing

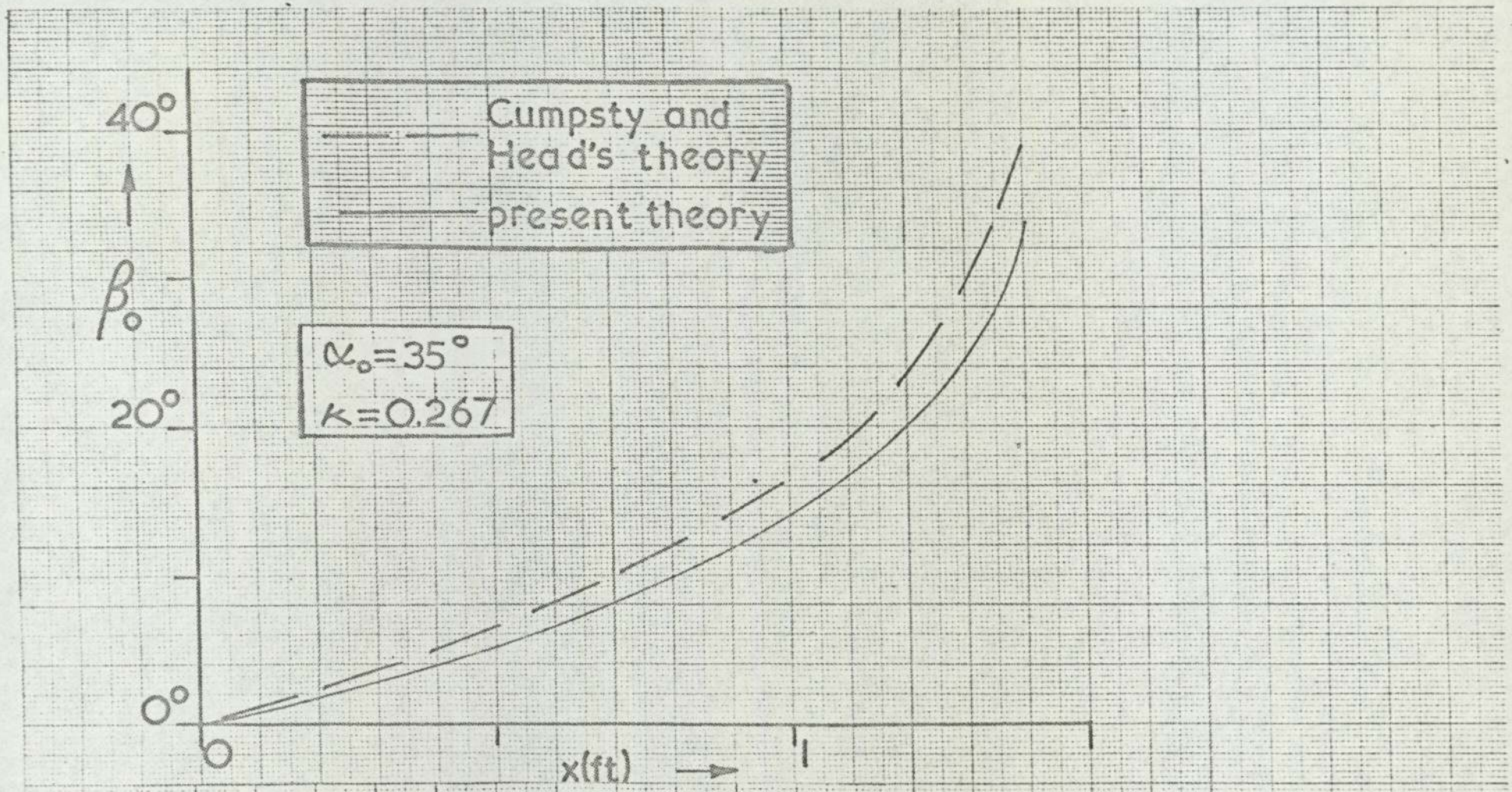


FIG (6.1.4) Cumpsty and Head [30]
The hypothetical infinite swept wing

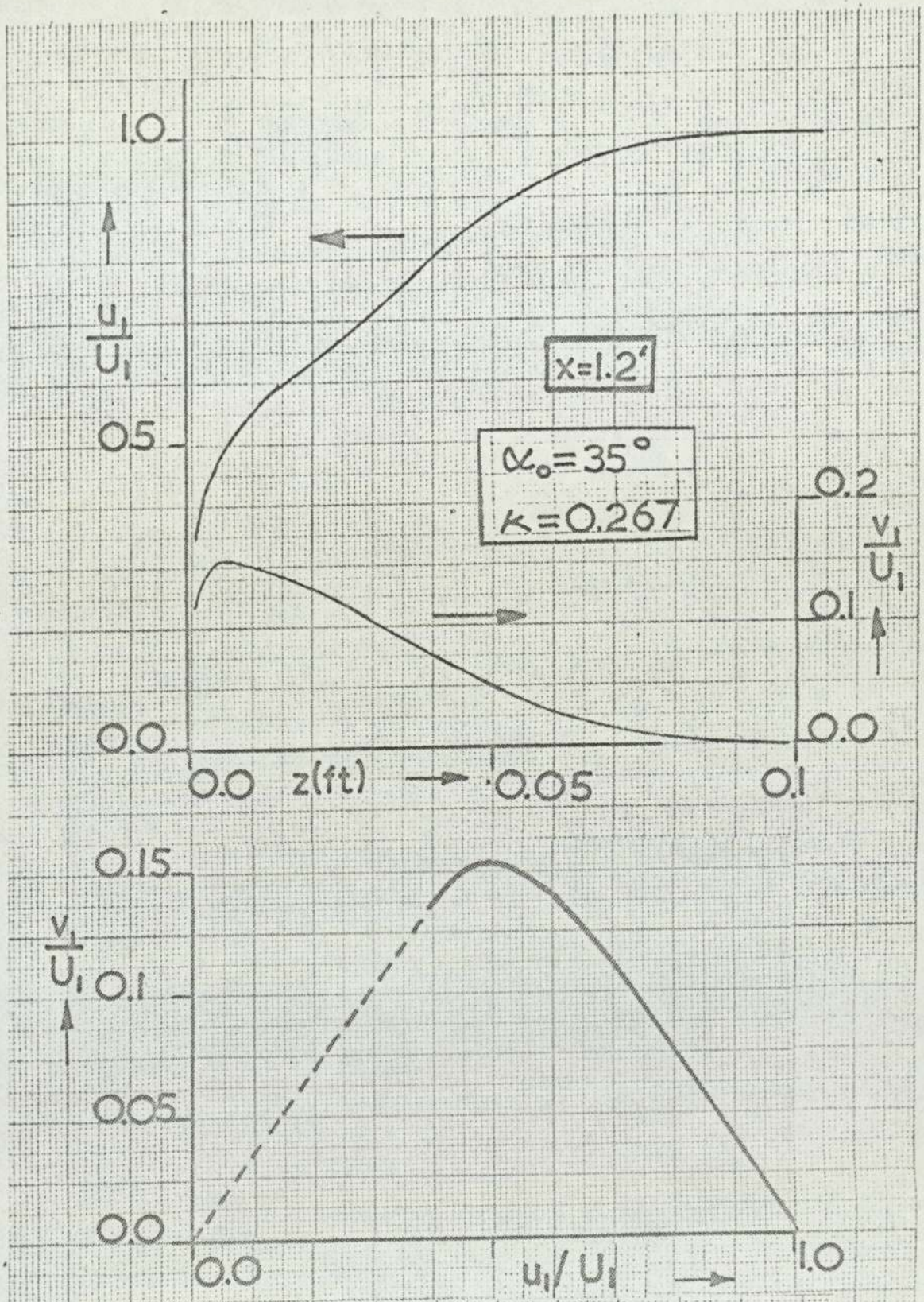


FIG (6.1.5) Cumpsty and Head [30]
The hypothetical infinite swept wing

present theory

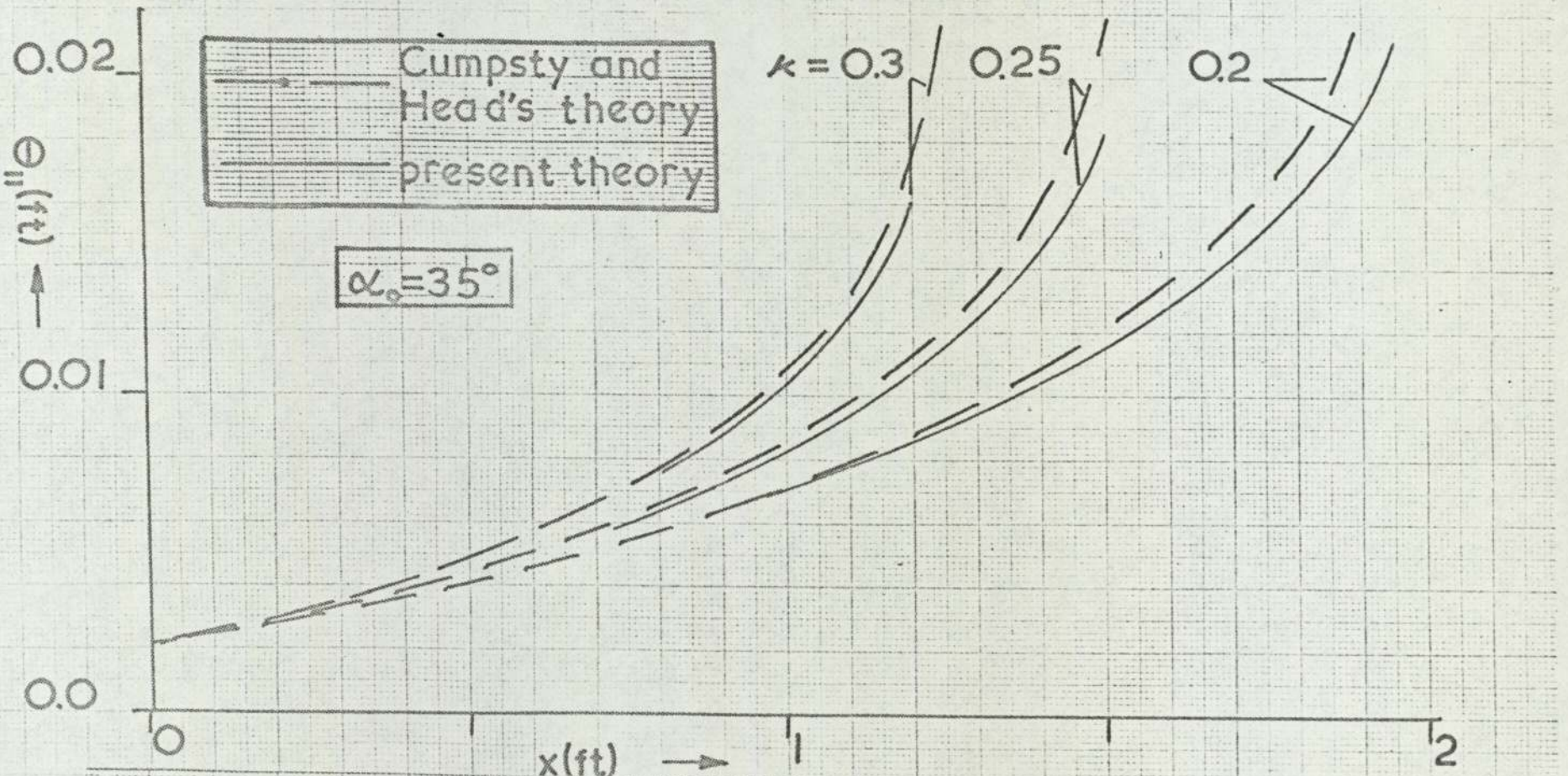
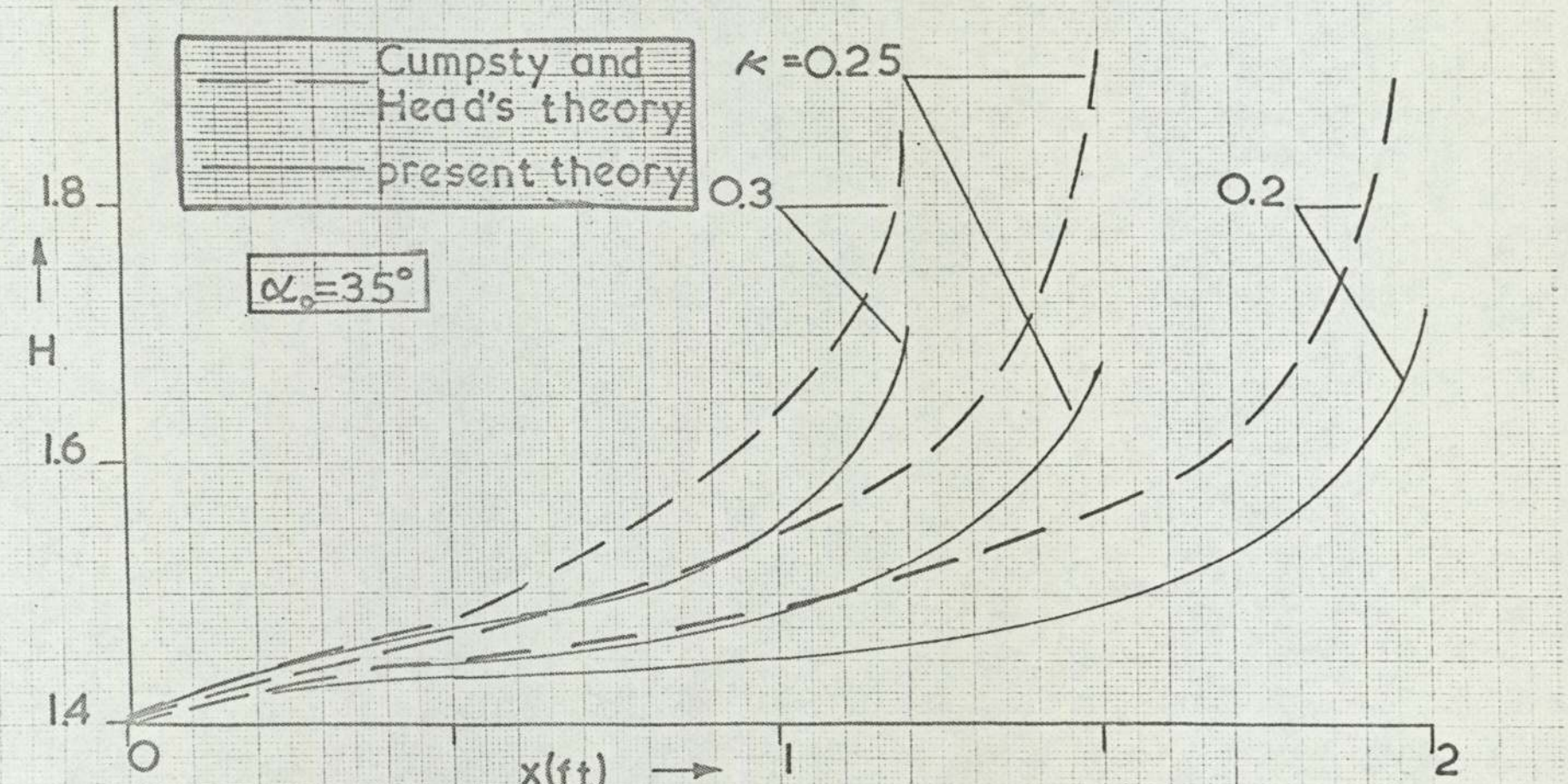
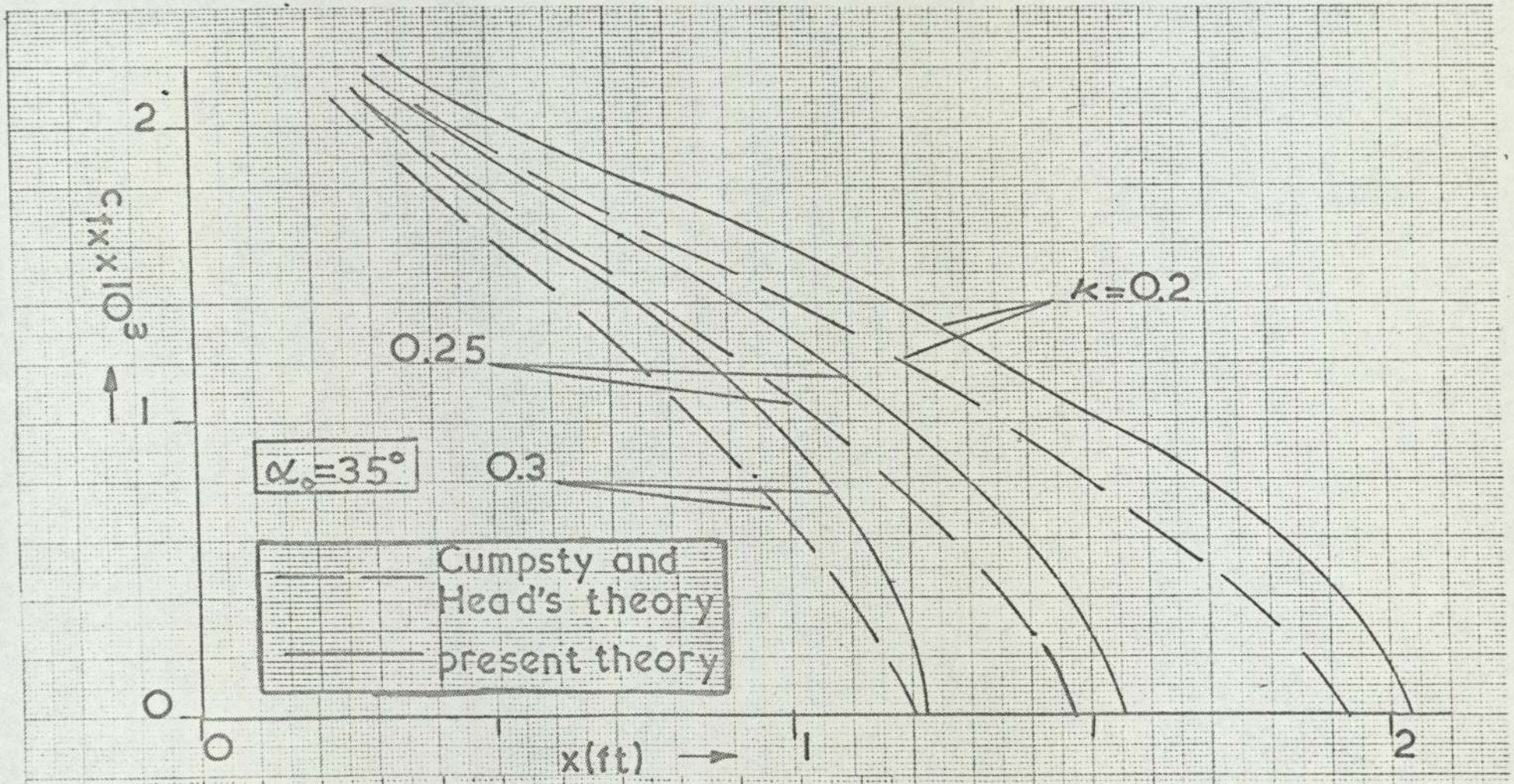


FIG (6.1.6) Cumpsty and Head [30]
The hypothetical infinite swept wing



FIG(6.1.7) Cumpsty and Head [30]
The hypothetical infinite swept wing



FIG(6.1.8) Cumpsty and Head [30]
The hypothetical infinite swept wing

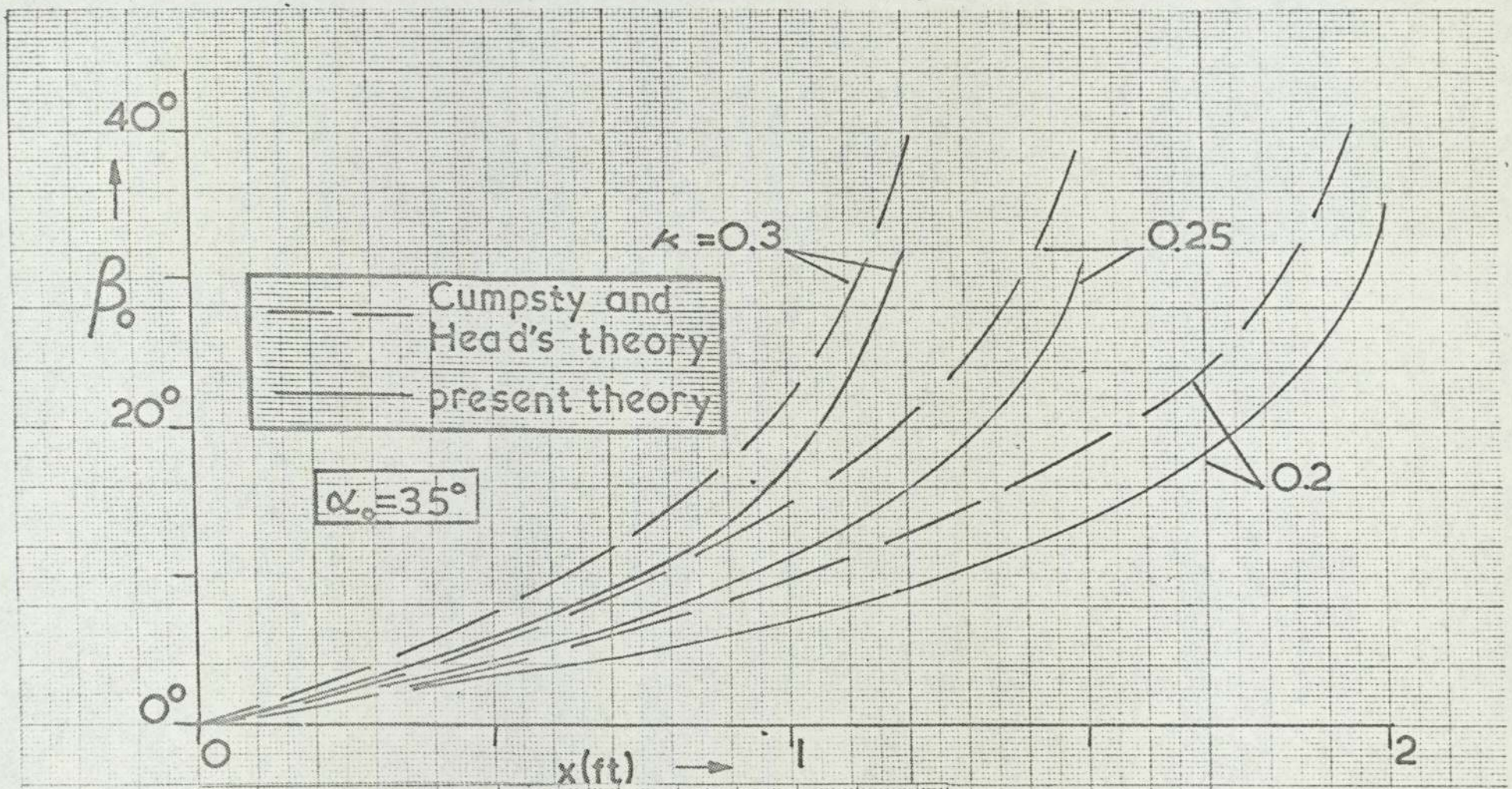


FIG (6.1.9) Cumpsty and Head [30]
The hypothetical infinite swept wing

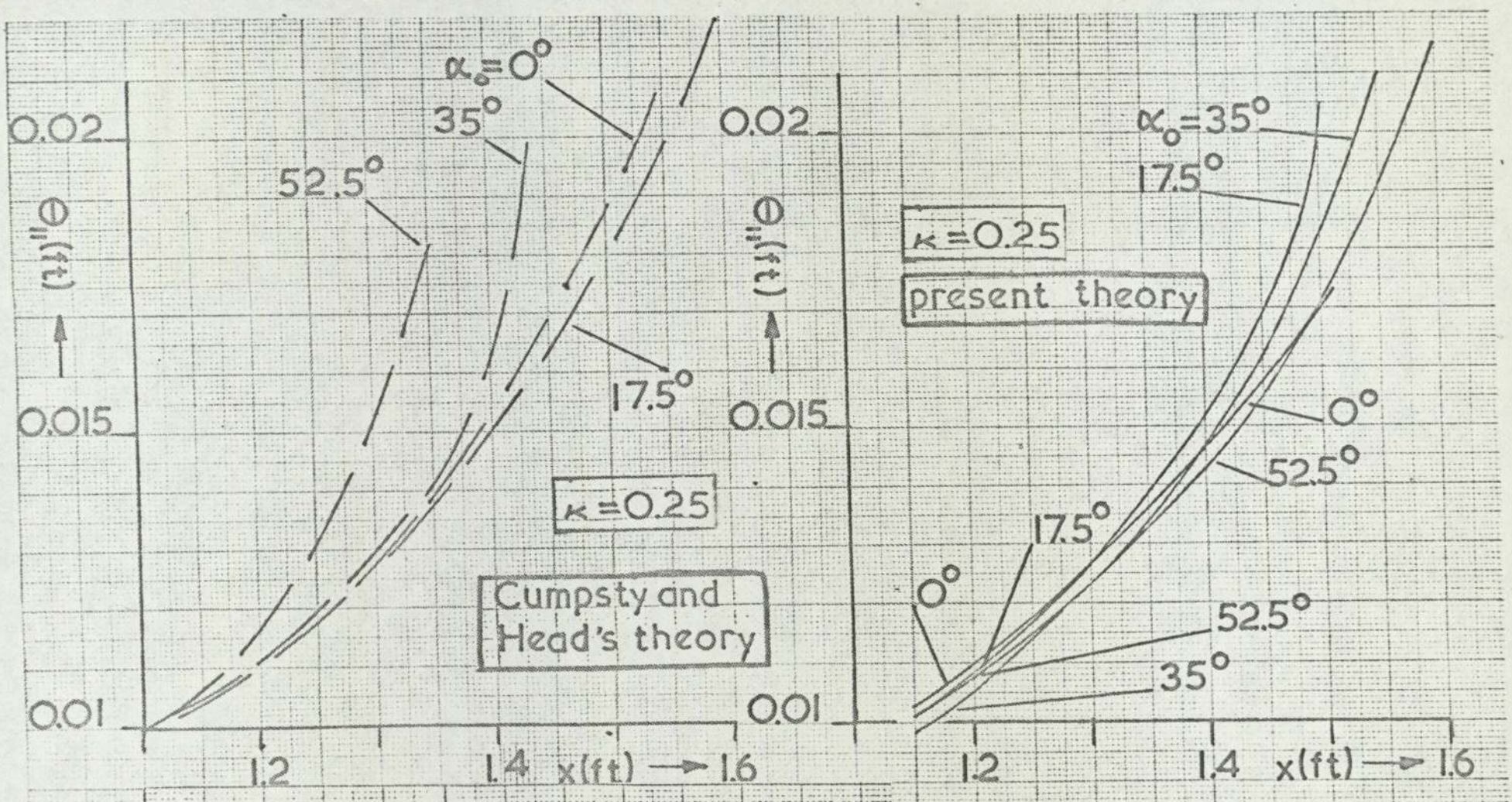


FIG (6.11(0)) Cumpsty and Head [30]
The hypothetical infinite swept wing

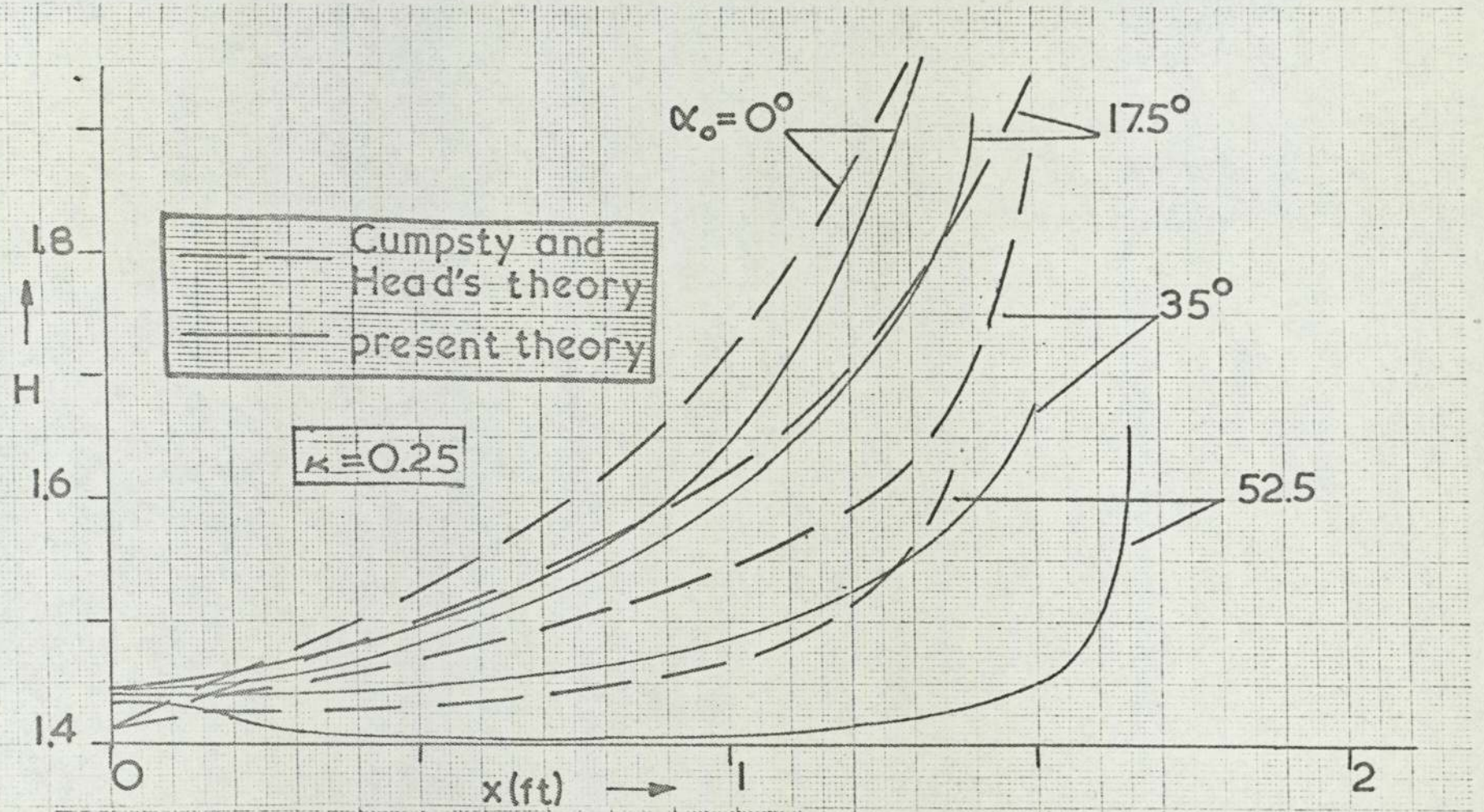


FIG (6.1.11) Cumpsty and Head [30]
The hypothetical infinite swept wing

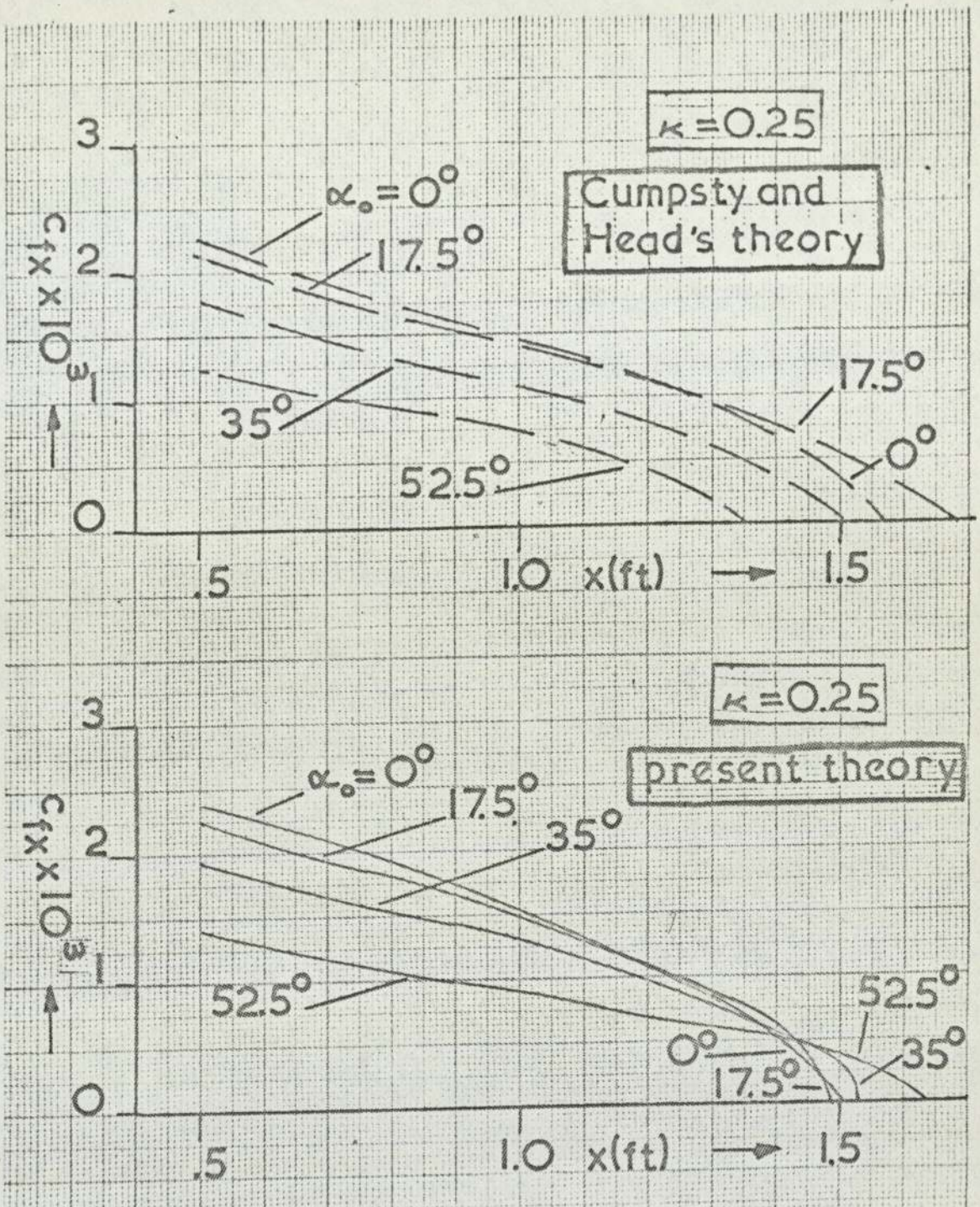


FIG (6.1.12) Cumpsty and Head [30] The hypothetical infinite swept wing

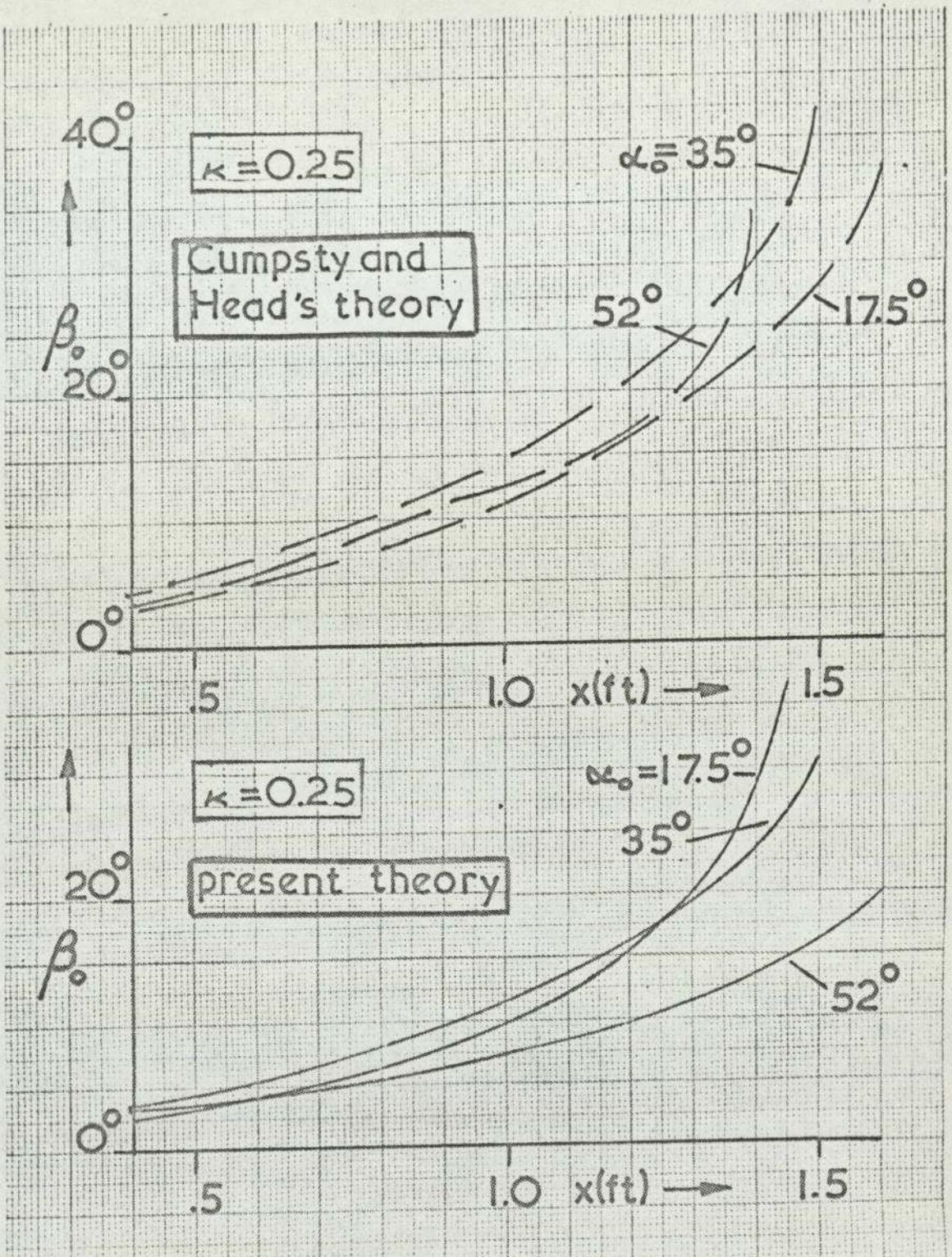


FIG (6.1.13) Cumpsty and Head [30] The hypothetical infinite swept wing

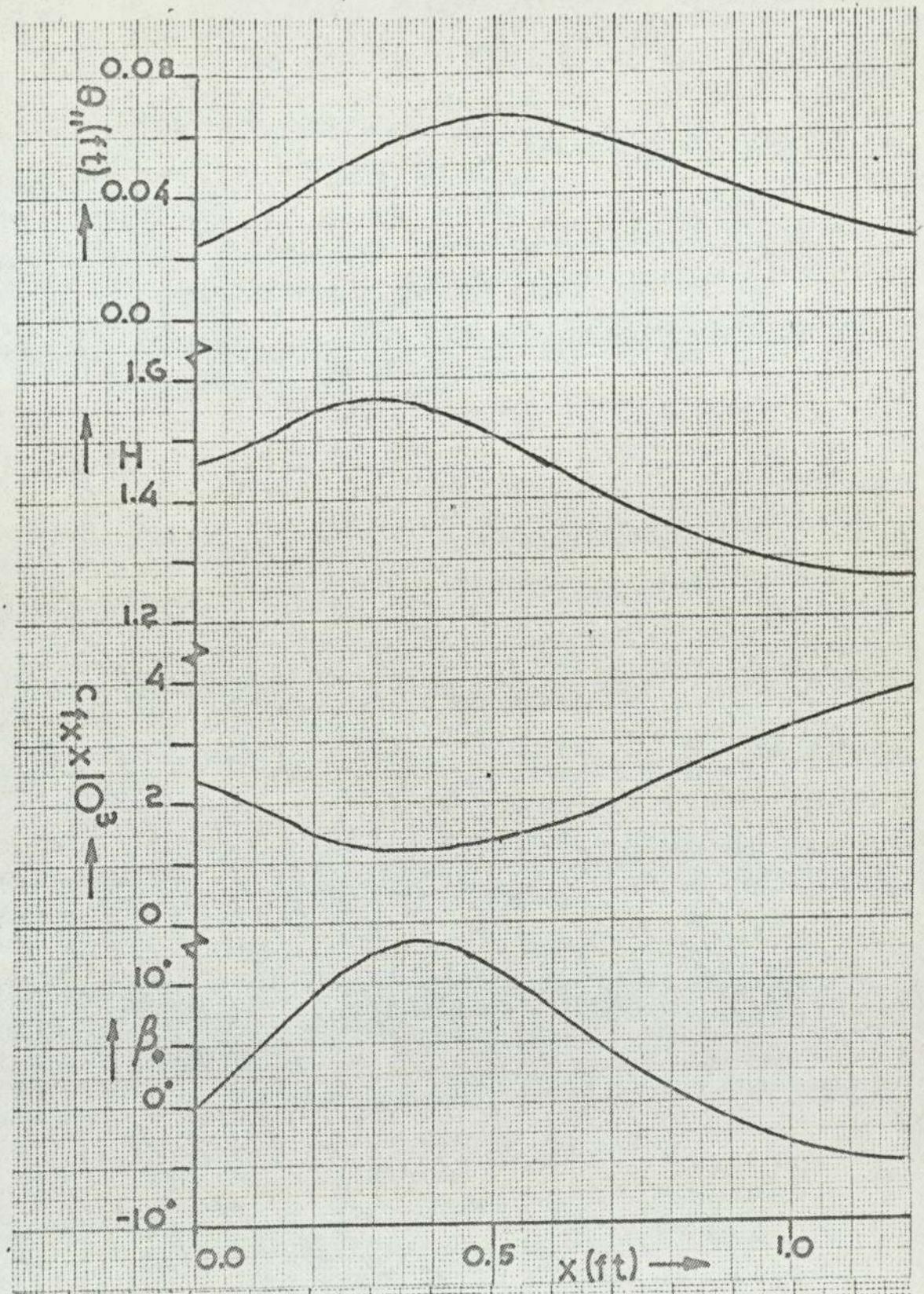


FIG (6.1.14) Cumpsty and Head [30]
The hypothetical 'crossover' flow
($\alpha_0 = 35°, \kappa = 1-x$)

present theory

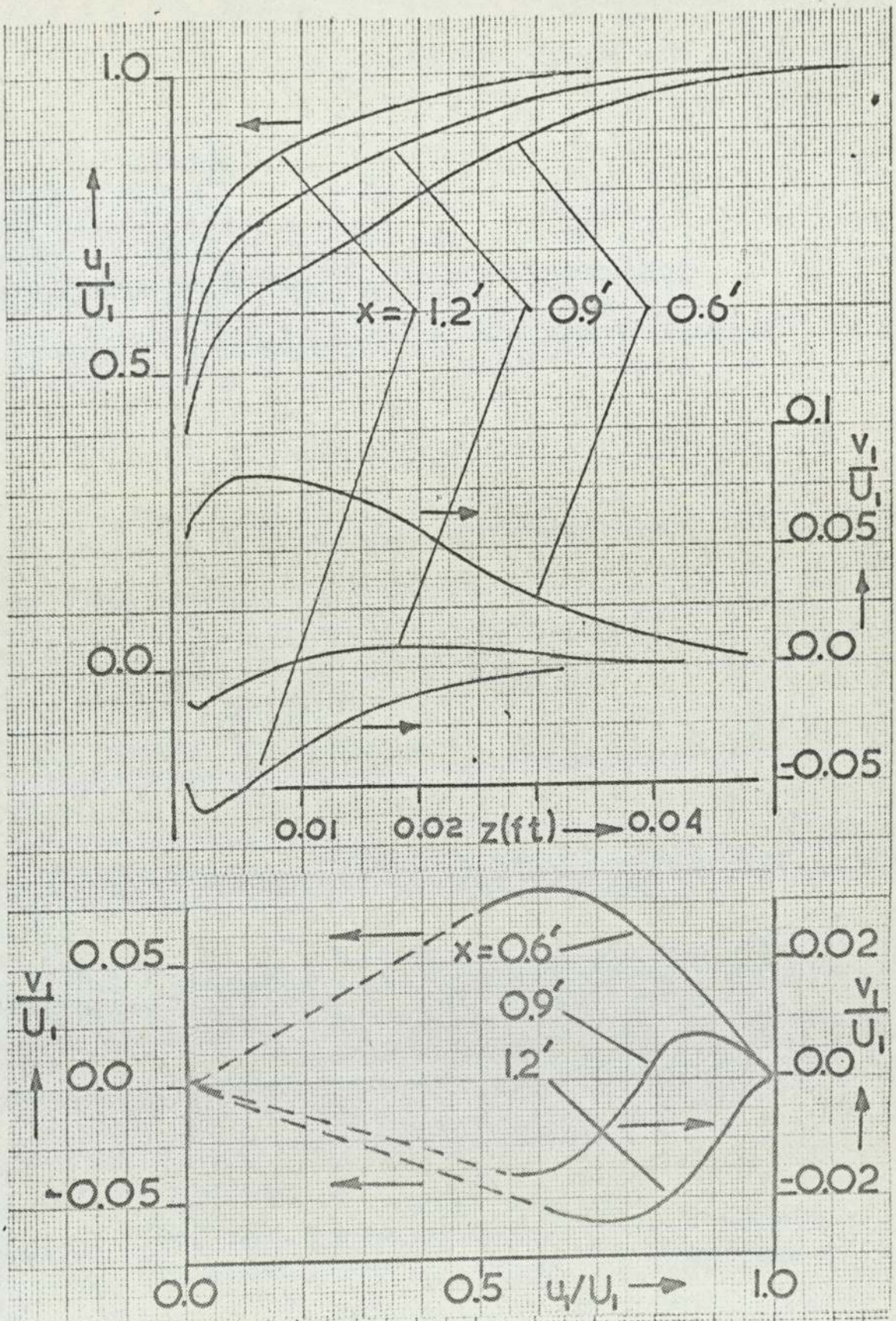


FIG (6.1.15) Cumpsty and Head [30]
The hypothetical 'crossover' flow
($\alpha_a = 35°$, $\kappa = 1-x$)
present theory

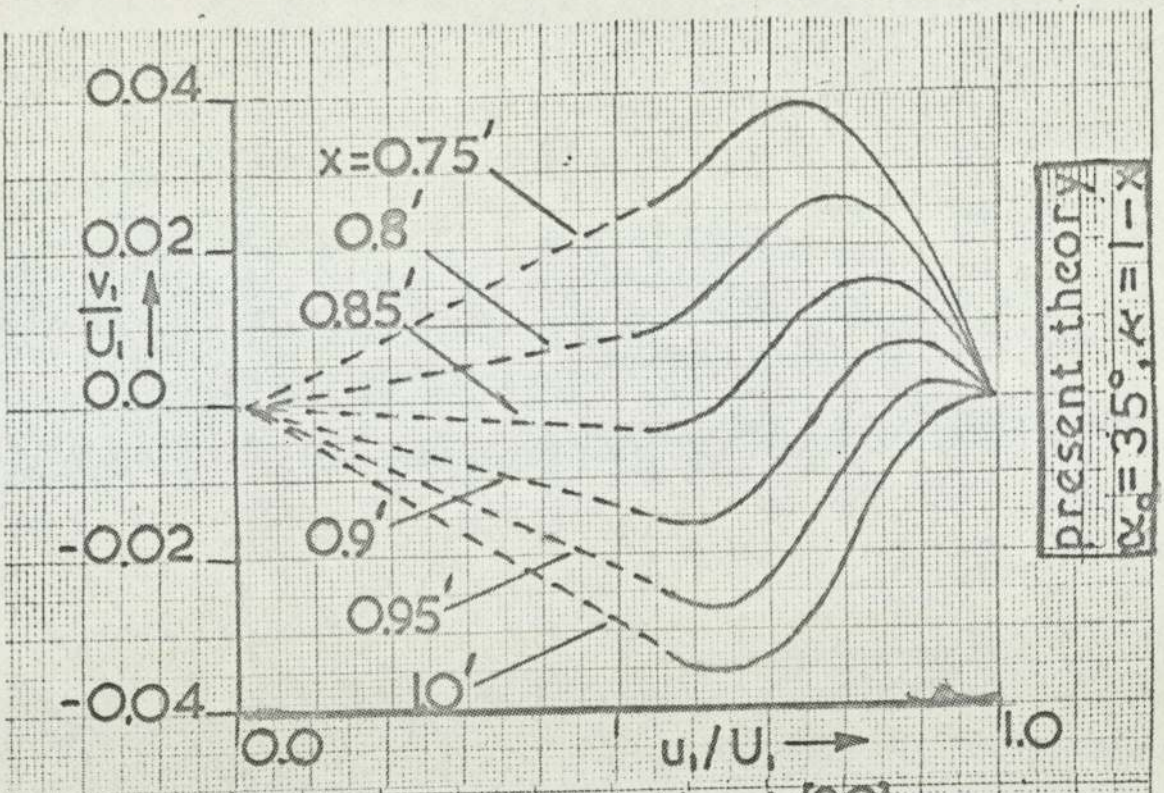


FIG (6.1.16) Cumpsty and Head [30]
The hypothetical 'crossover' flow

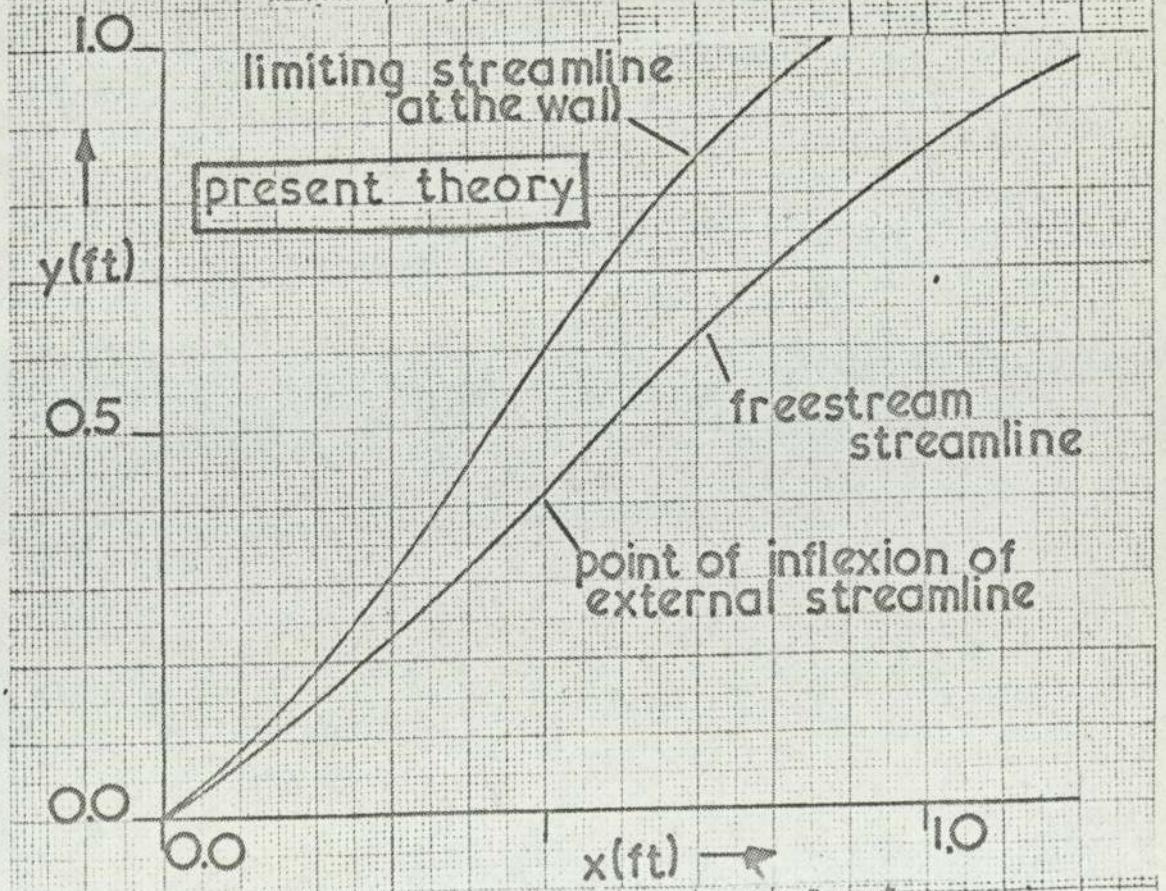


FIG (6.1.17) Cumpsty and Head [30]
The hypothetical 'crossover' flow
($\alpha_0 = 35°, \kappa = 1-x$)

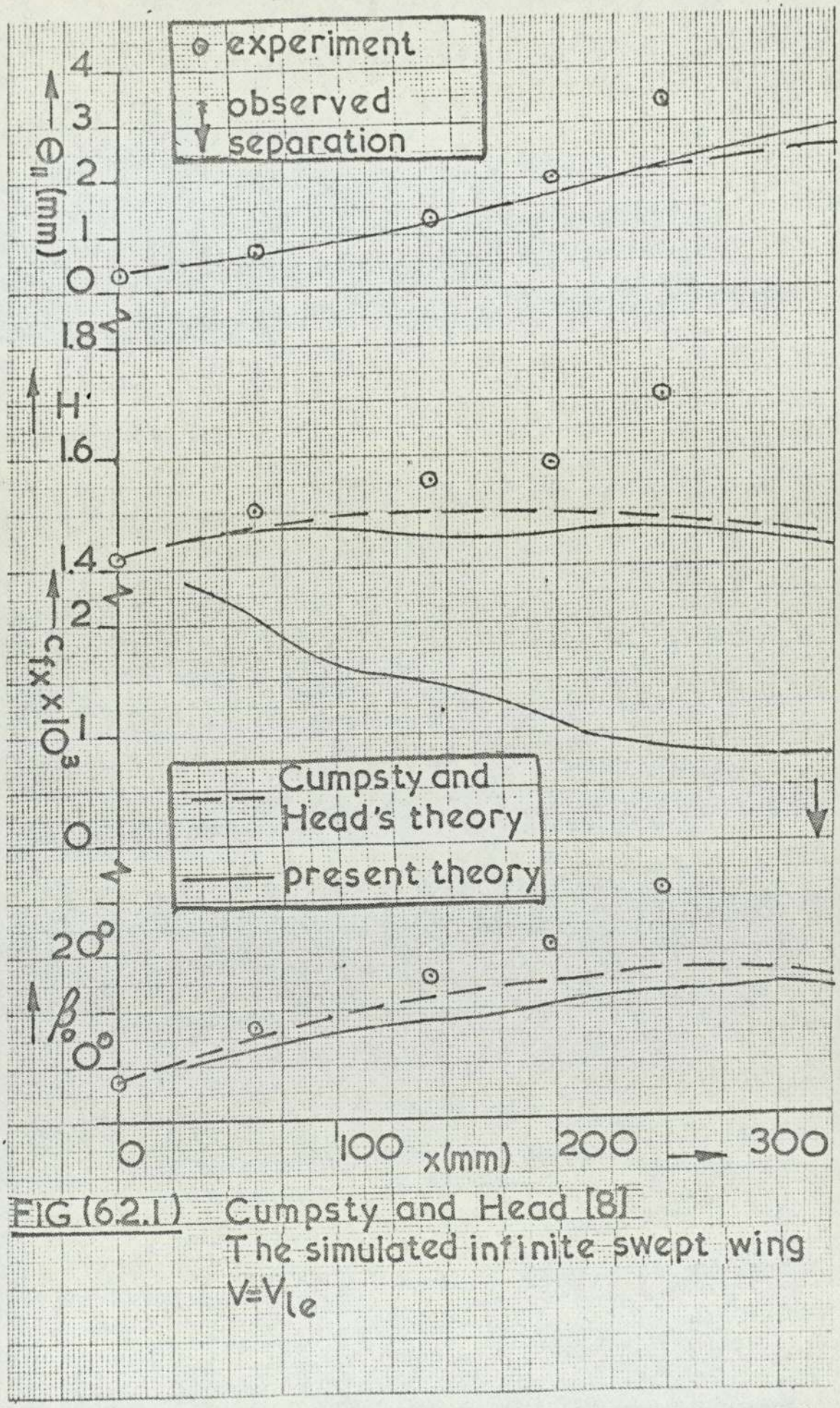


FIG (6.2.1) Cumpsty and Head [8]
The simulated infinite swept wing
$V = V_{le}$

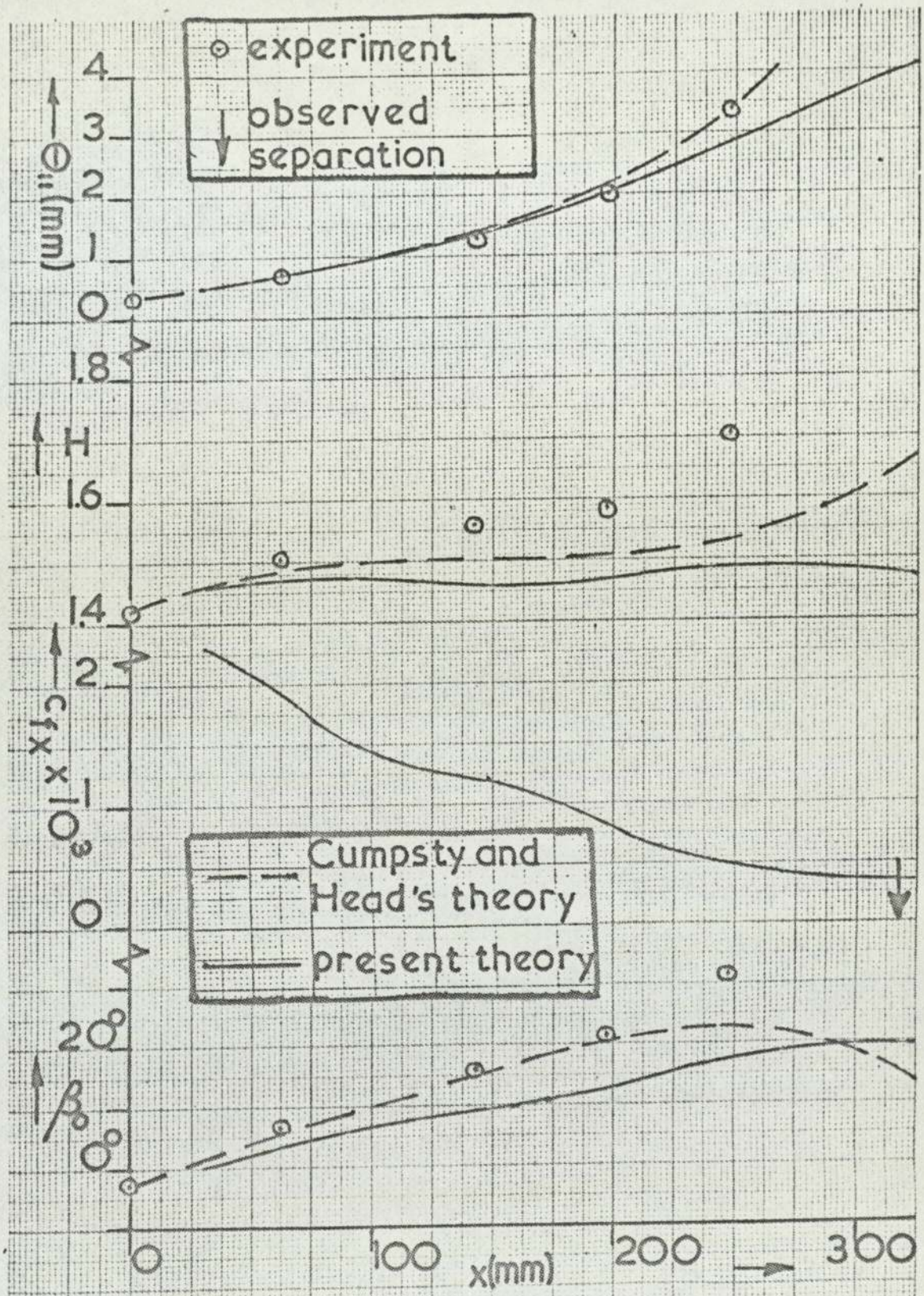


FIG (6.2.2) Cumpsty and Head [8]
The simulated infinite swept wing
$V = 1.05 V_{le}$

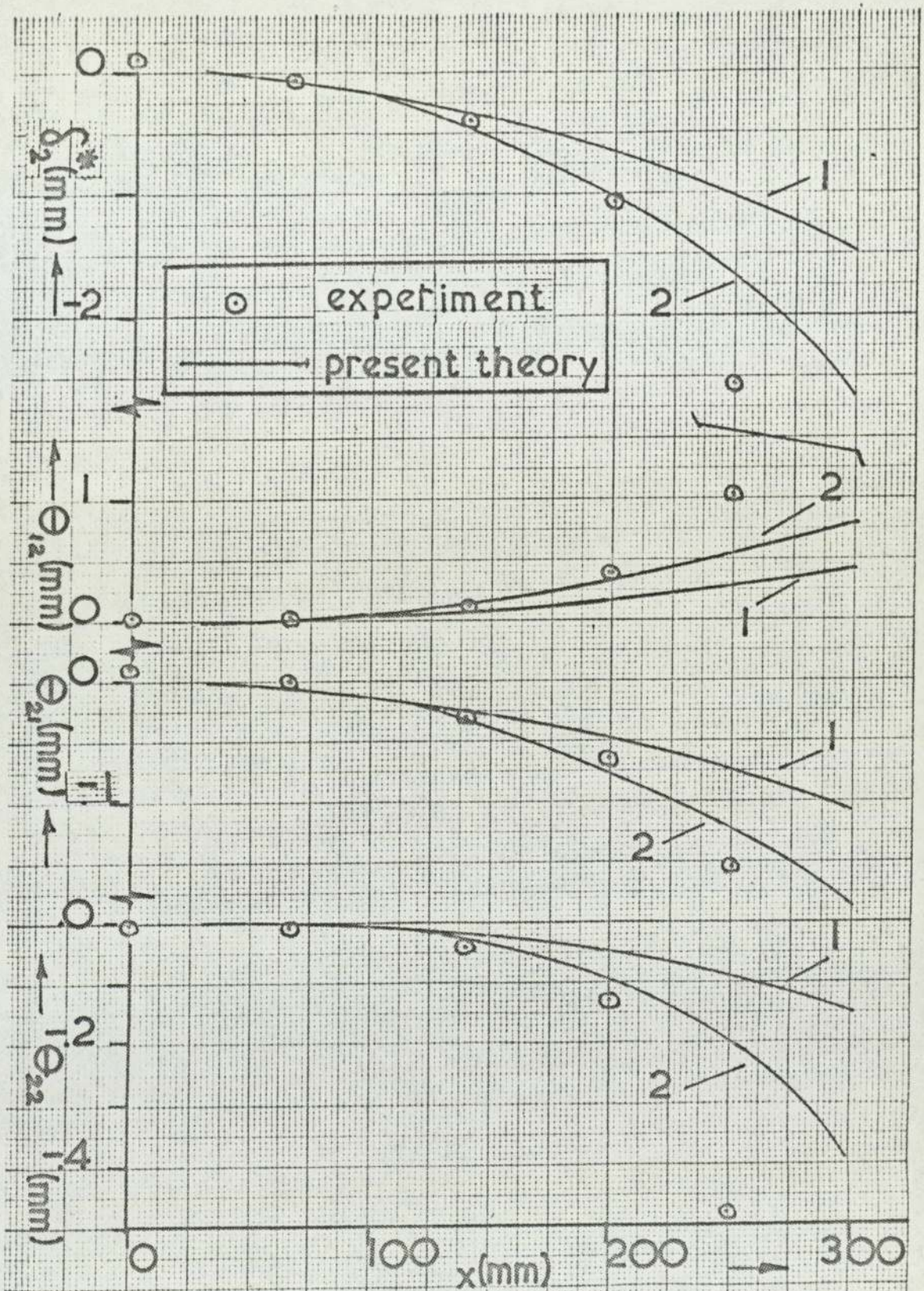


FIG (6.2.3) Cumpsty and Head [8]
The simulated infinite swept wing
curve 1 $V = V_{le}$
2 $V = 1.05 V_{le}$

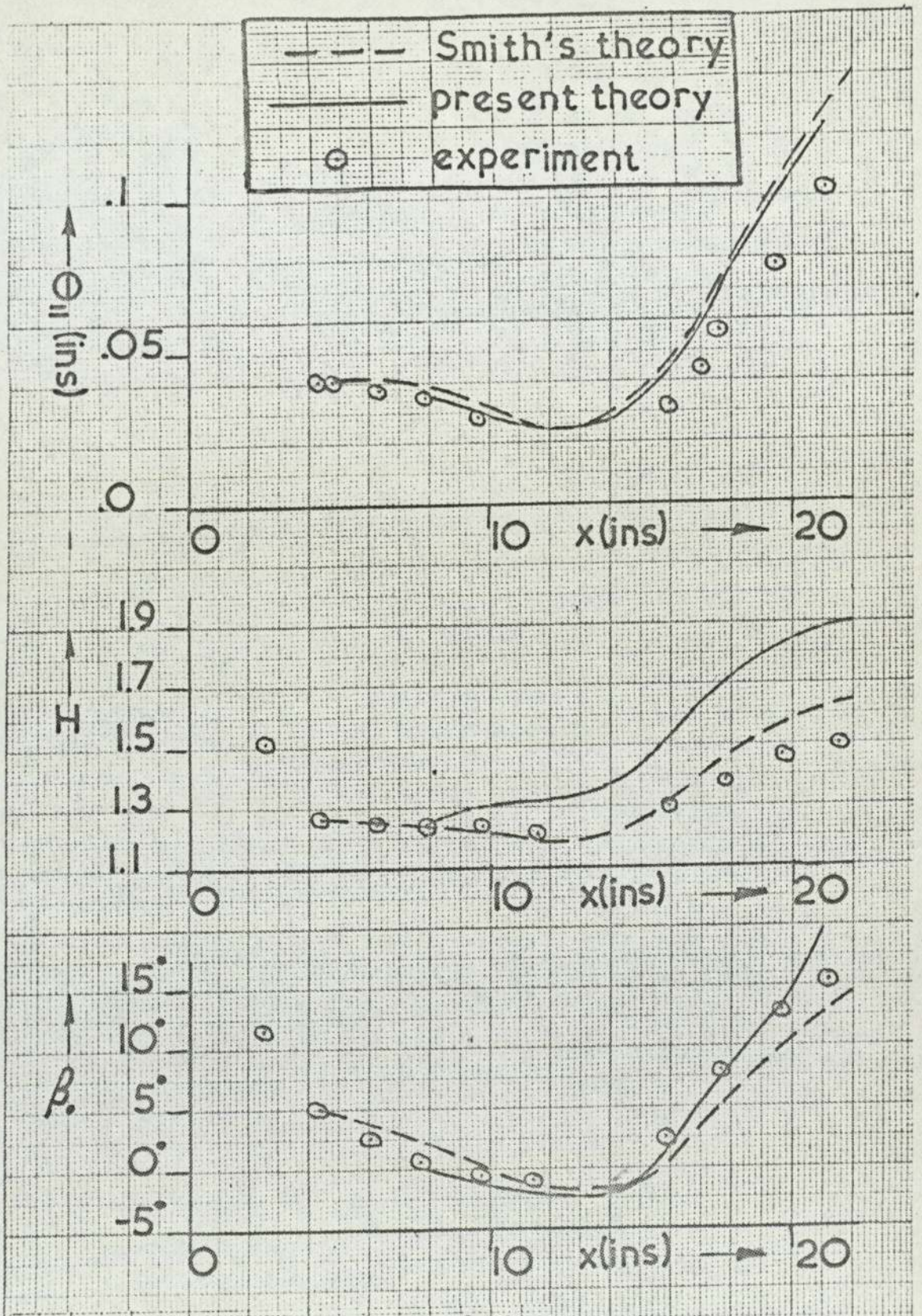


FIG (6.3.1) P. D. Smith [7]
Run 1

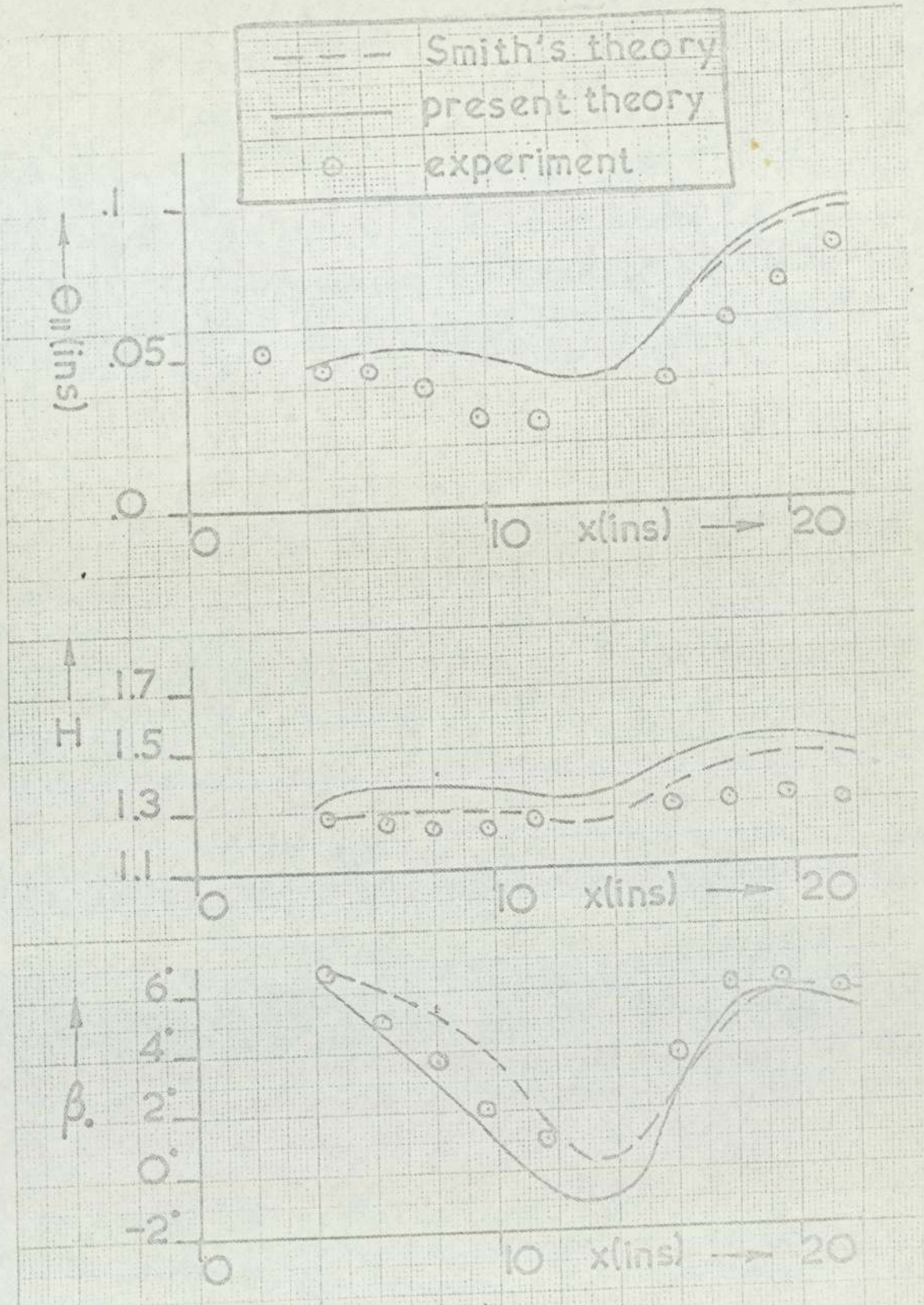


FIG (6.3.2) P. D. Smith [7]
Run 5

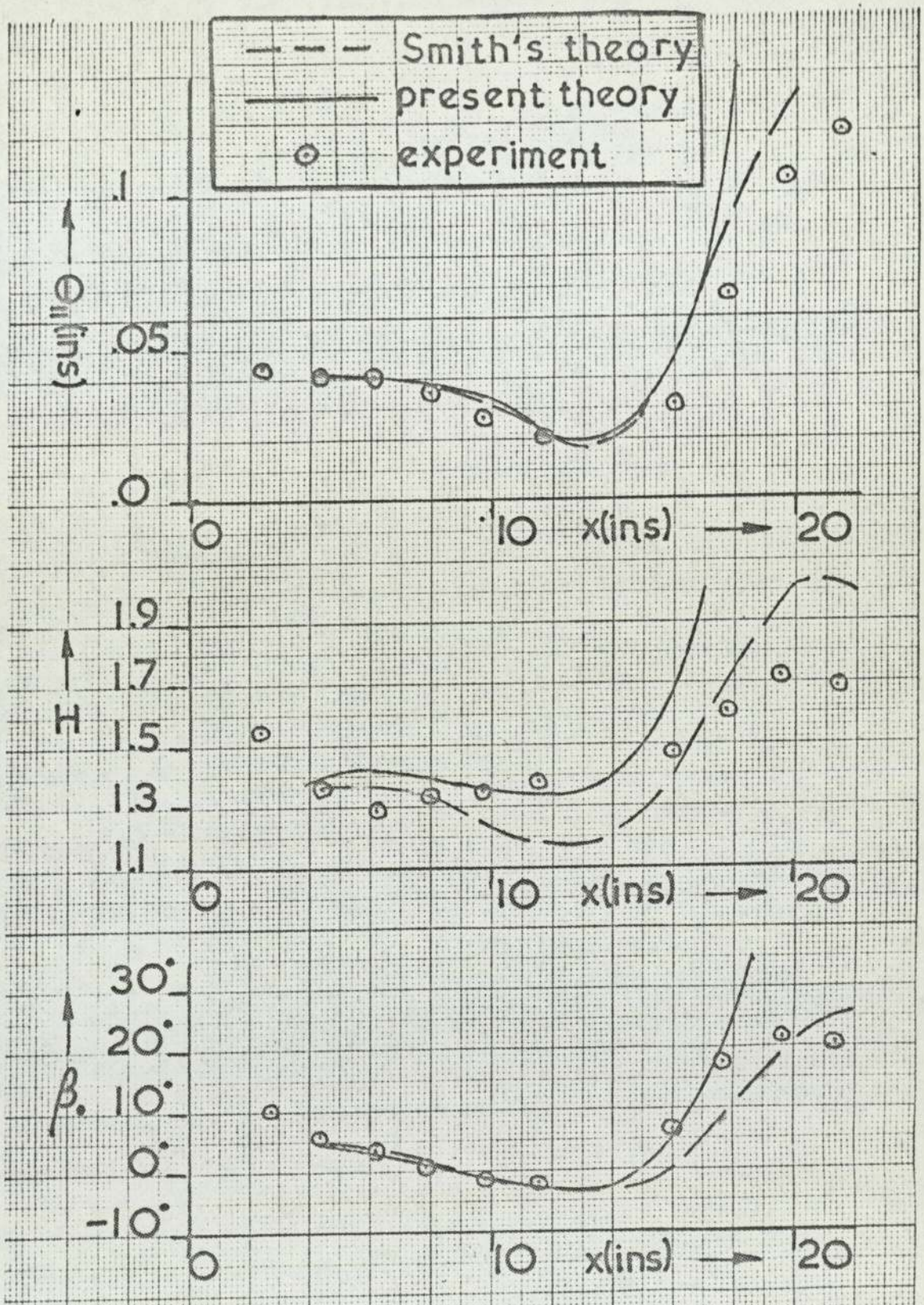


FIG (6.33) P. D. Smith [7]
Run 6

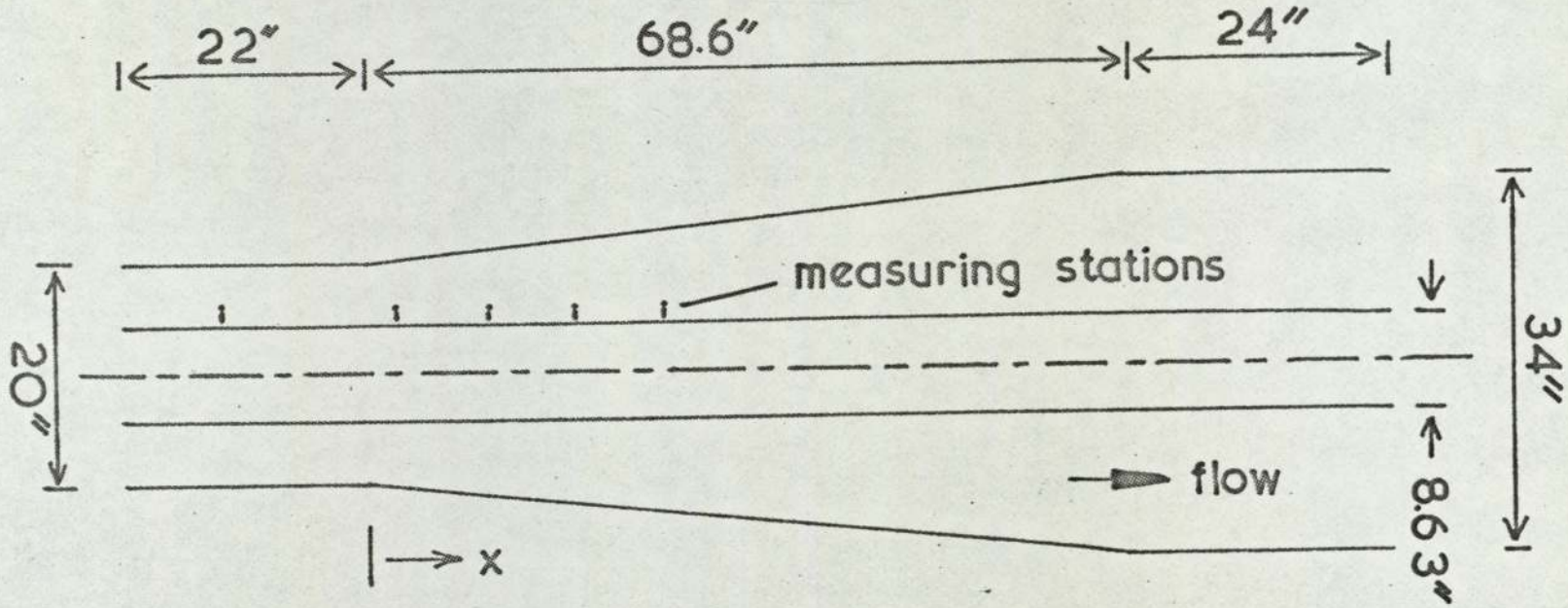


FIG 6.41 Hoadley's Diffuser

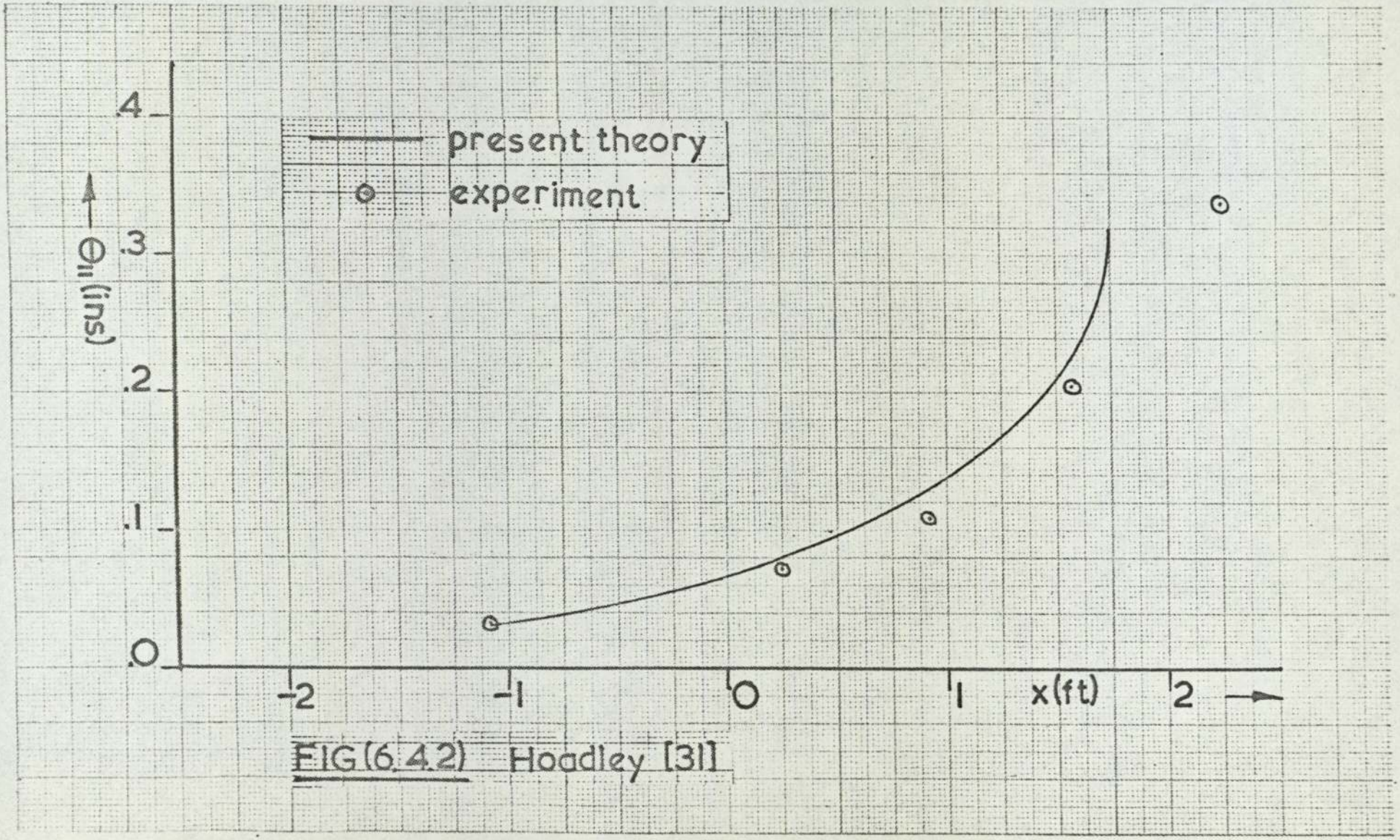
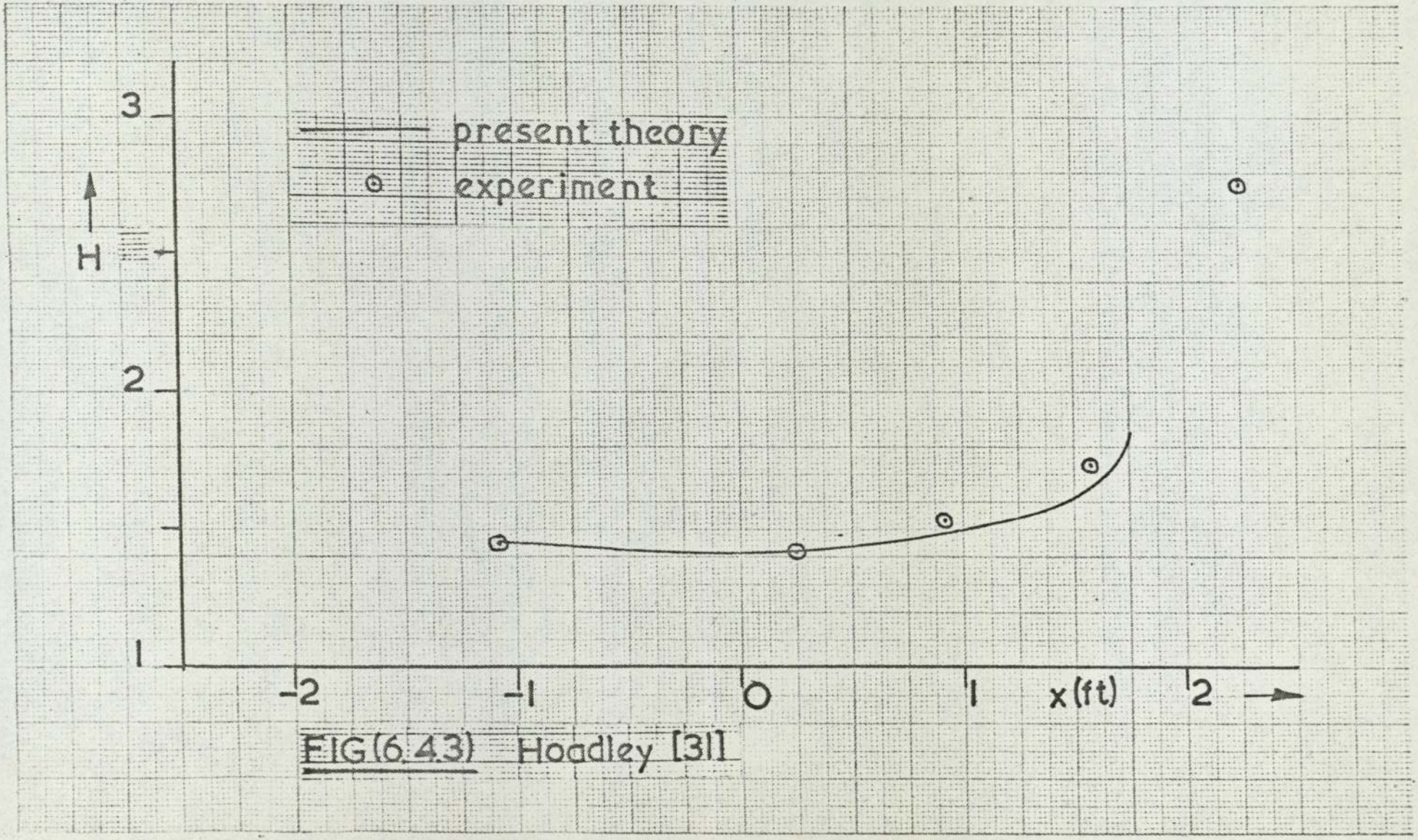
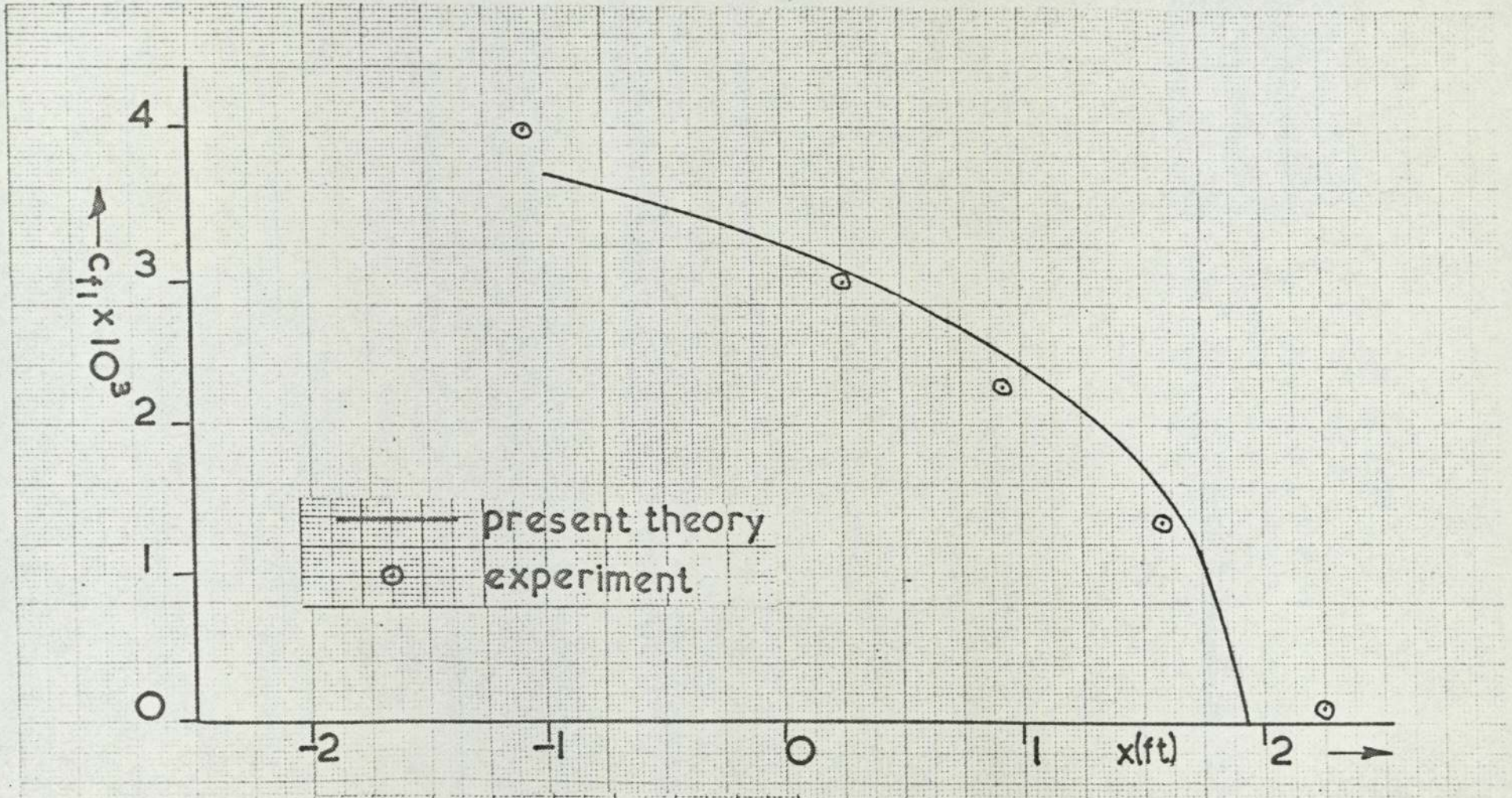


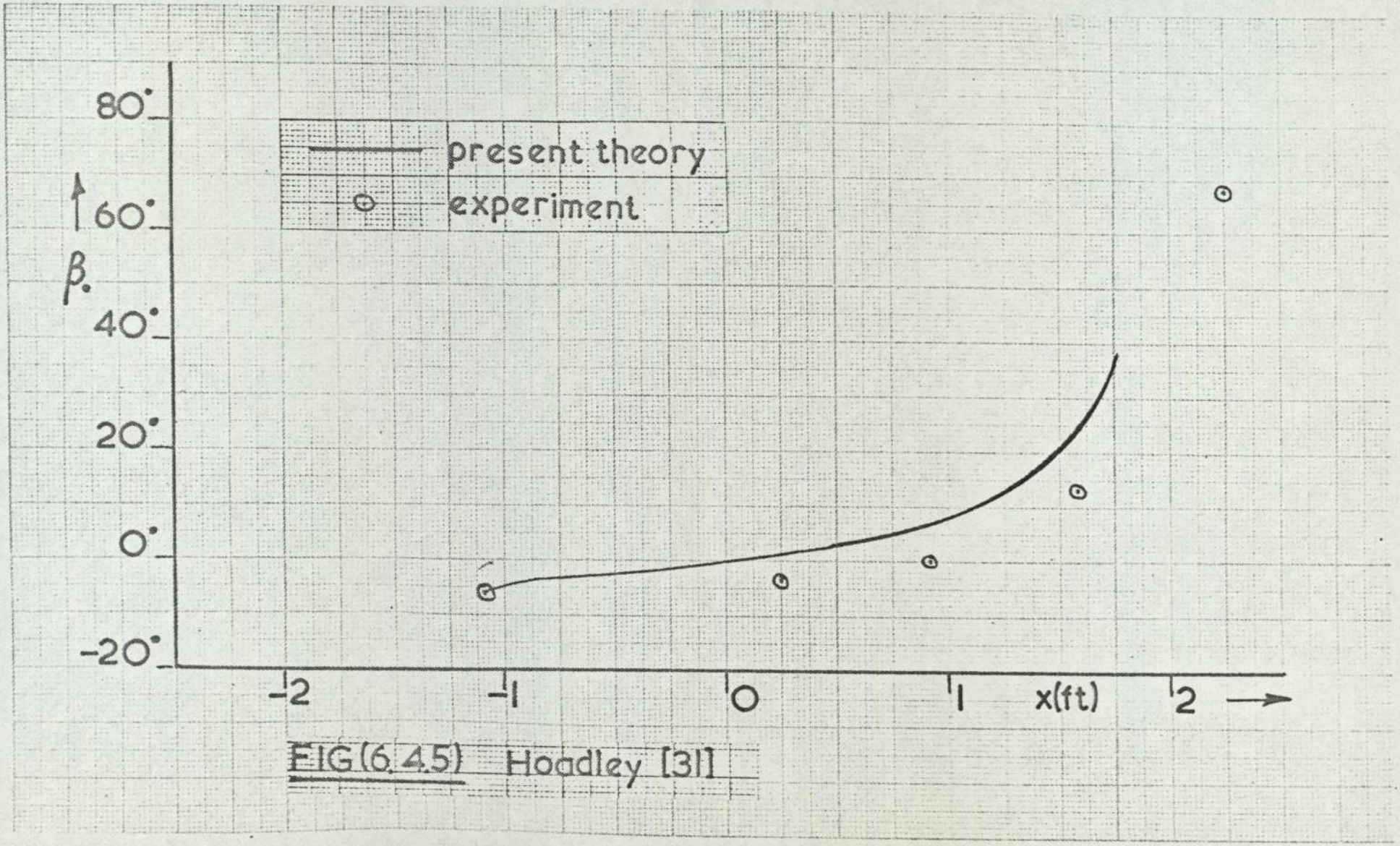
FIG (6.4.2) Hoadley [31]



FIG(6.4.3) Hoadley [31]



FIG(6.4.4) Hoadley [31]



FIG(6.4.5) Hoadley [31]

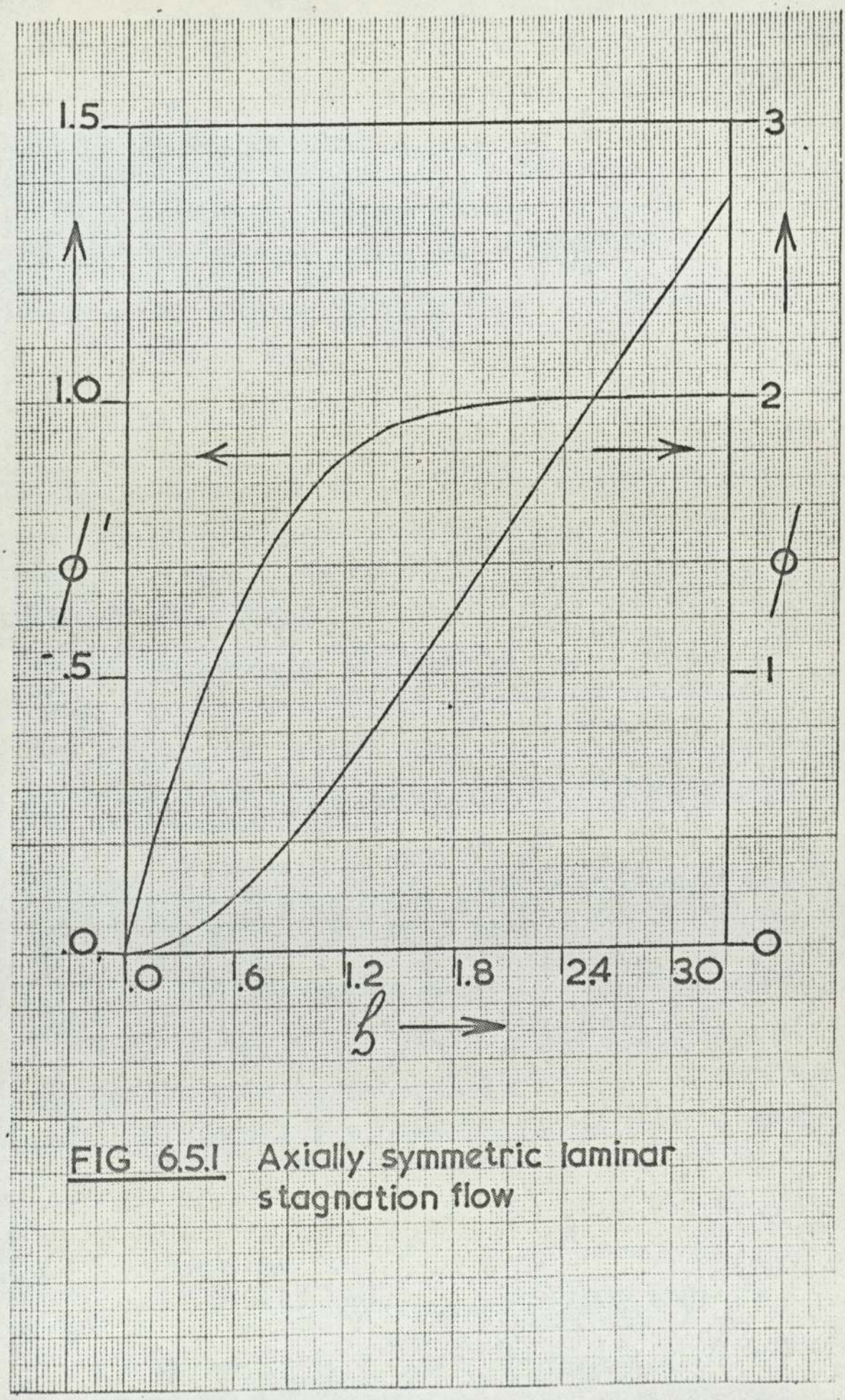


FIG 6.5.1 Axially symmetric laminar stagnation flow

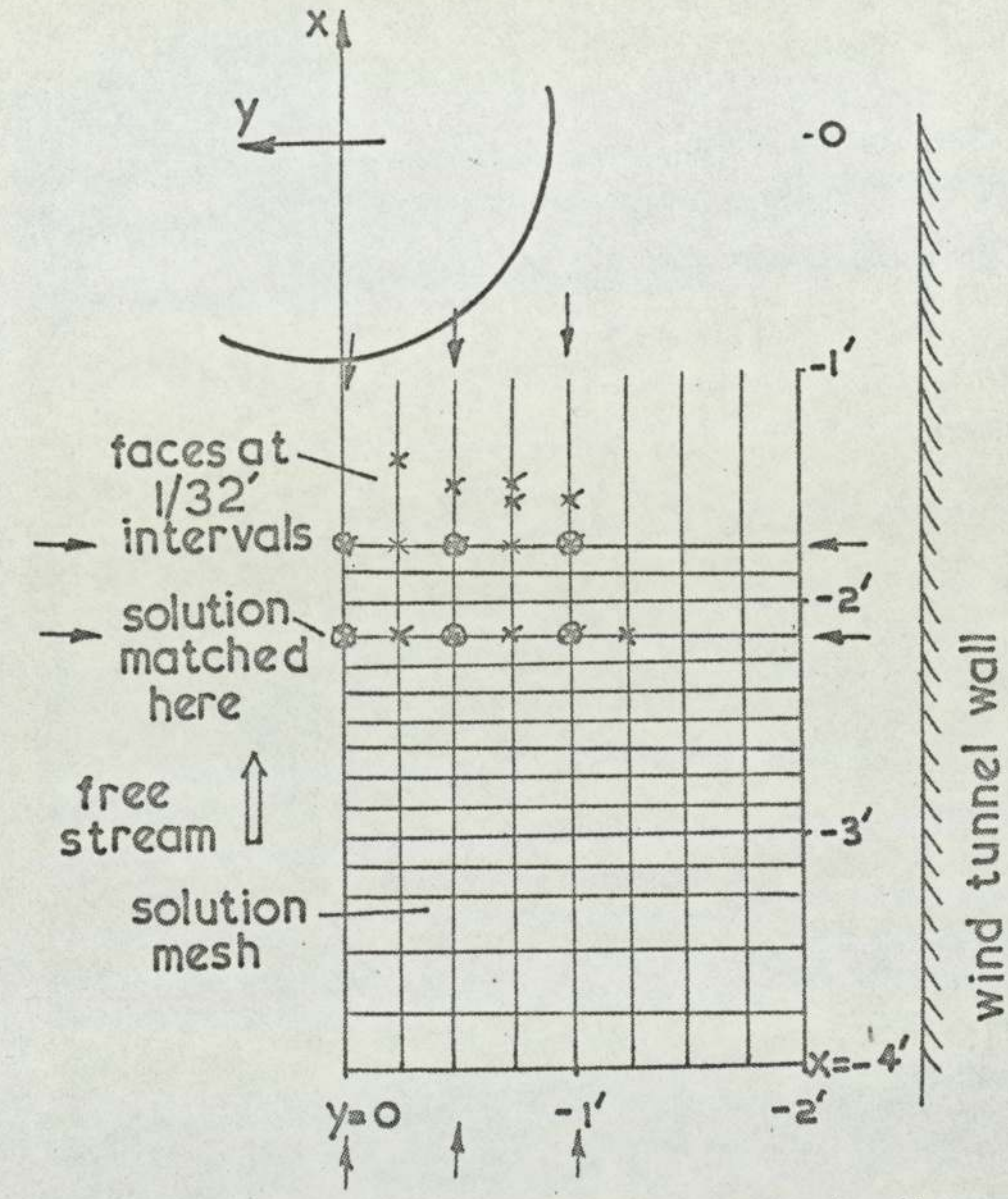


FIG (6.6.1) Hornung and Joubert [13]

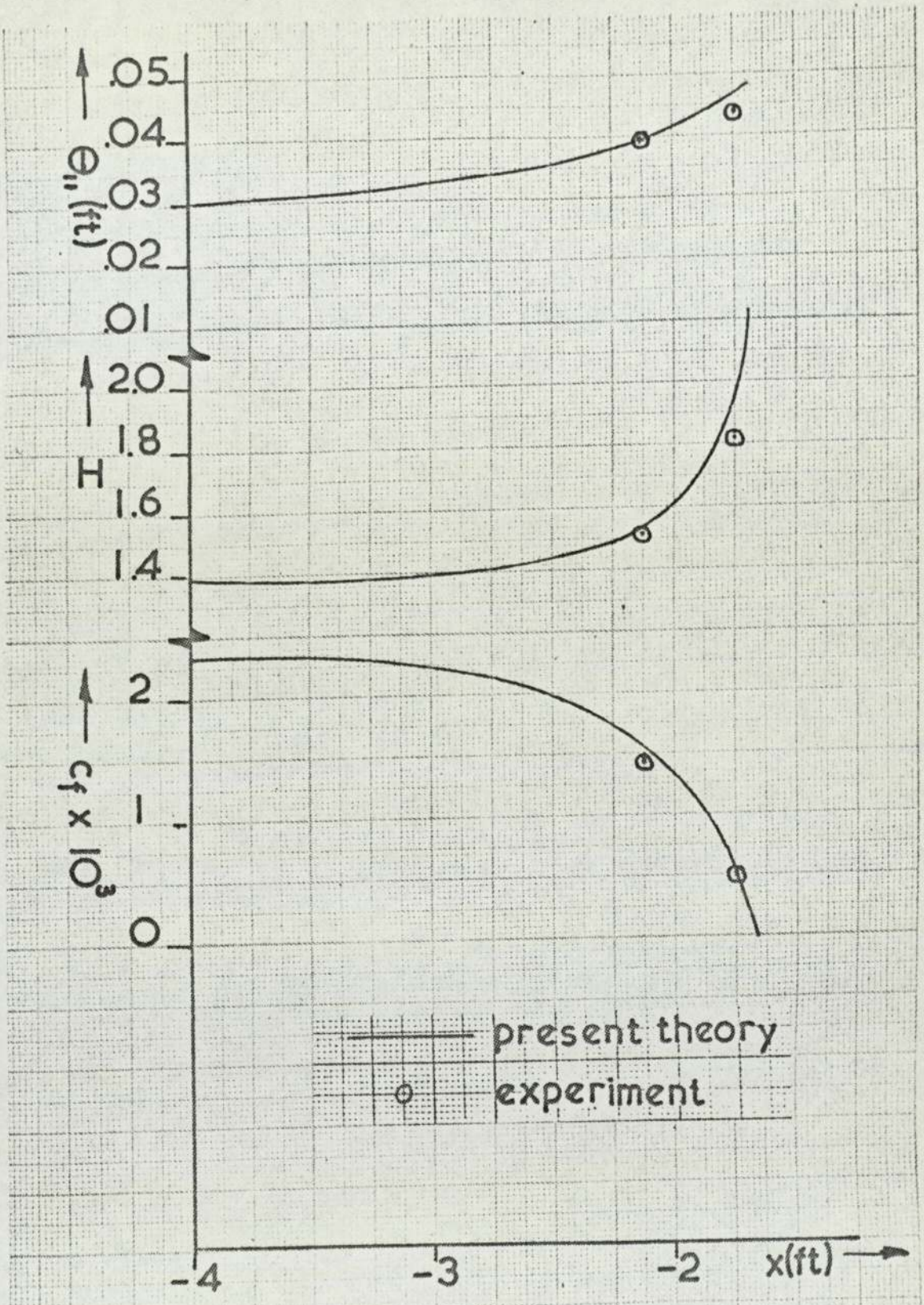


FIG (6.6.2) Hornung and Joubert [13]

y=0

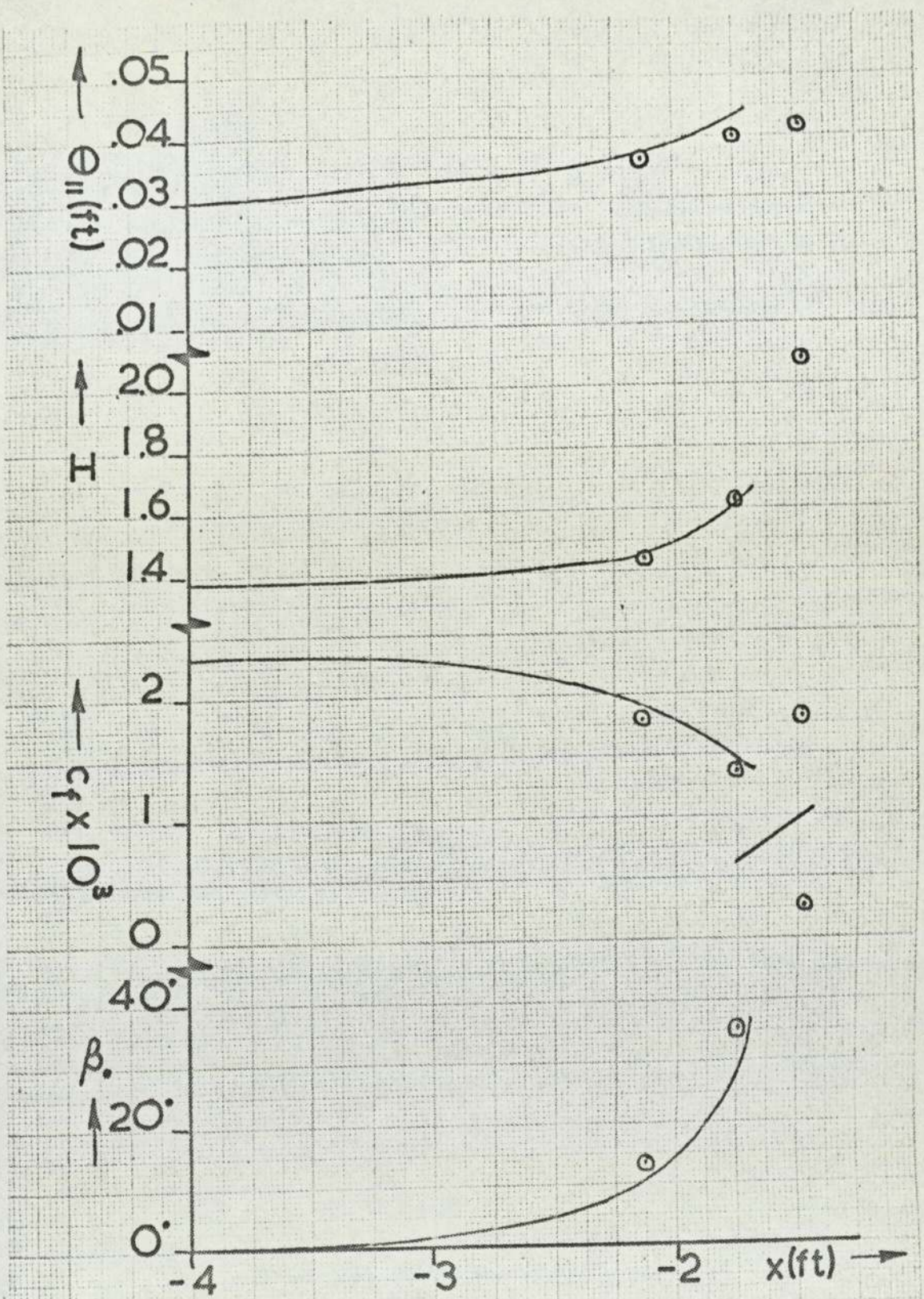


FIG (6.6.3) Hornung and Joubert [13]
Notation as in figure (6.6.2)
y=-.5'

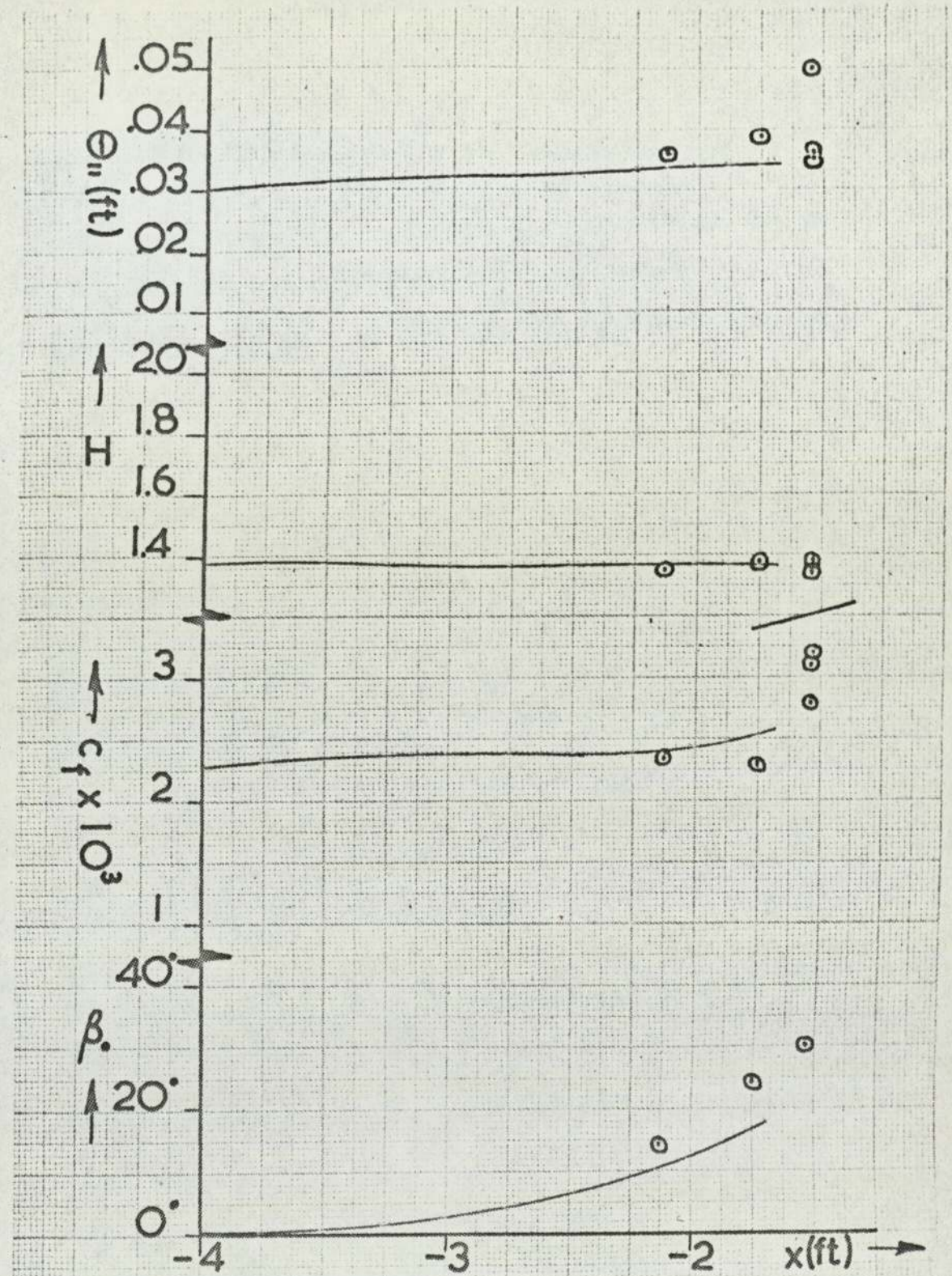


FIG (6.6.4) Hornung and Joubert [13]
Notation as in figure (6.6.2)
y=-1'

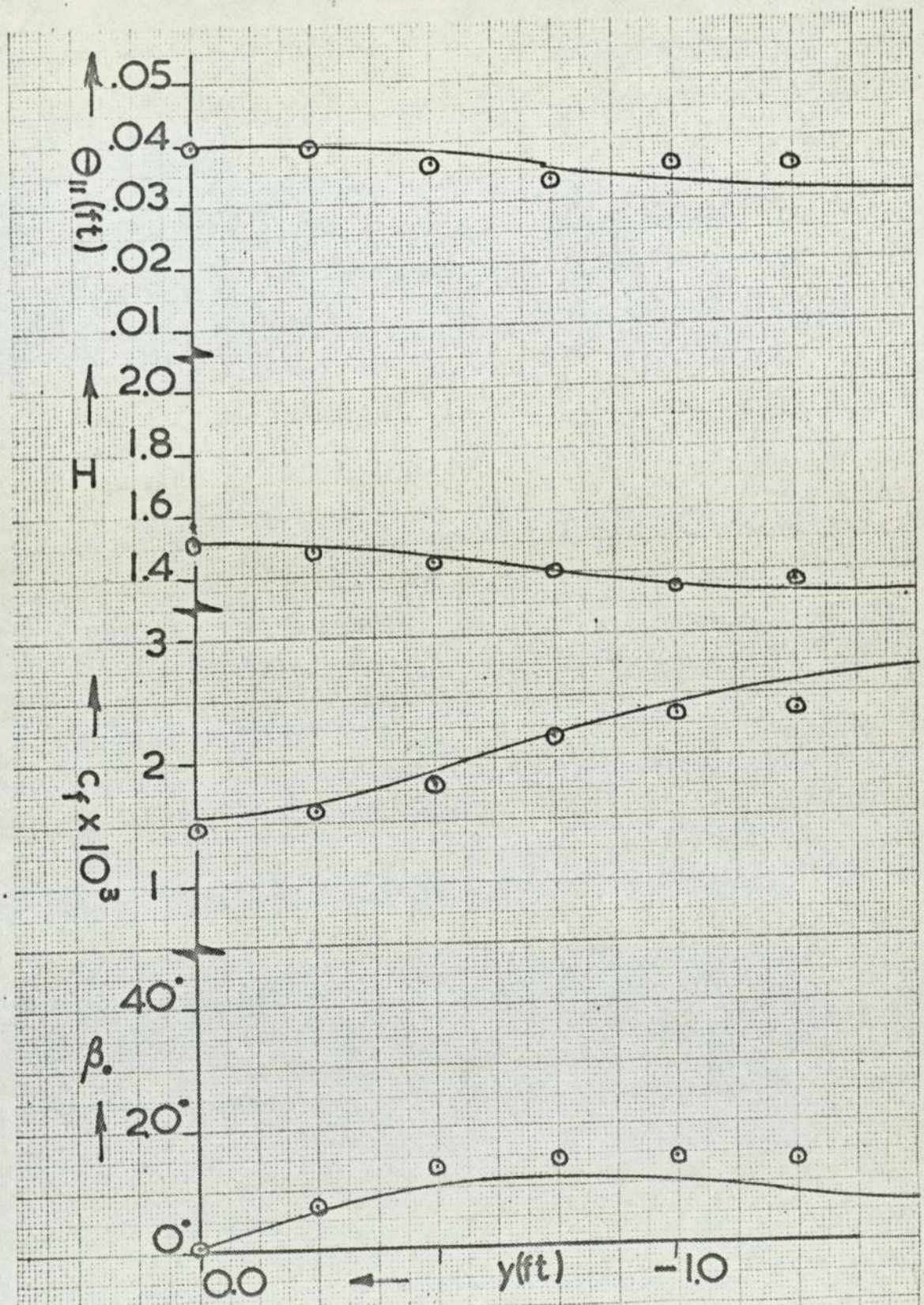


FIG (6.6.5) Hornung and Joubert [13]
Notation as in figure (6.6.2)
x=-2.125': solution matched to expt at y=0

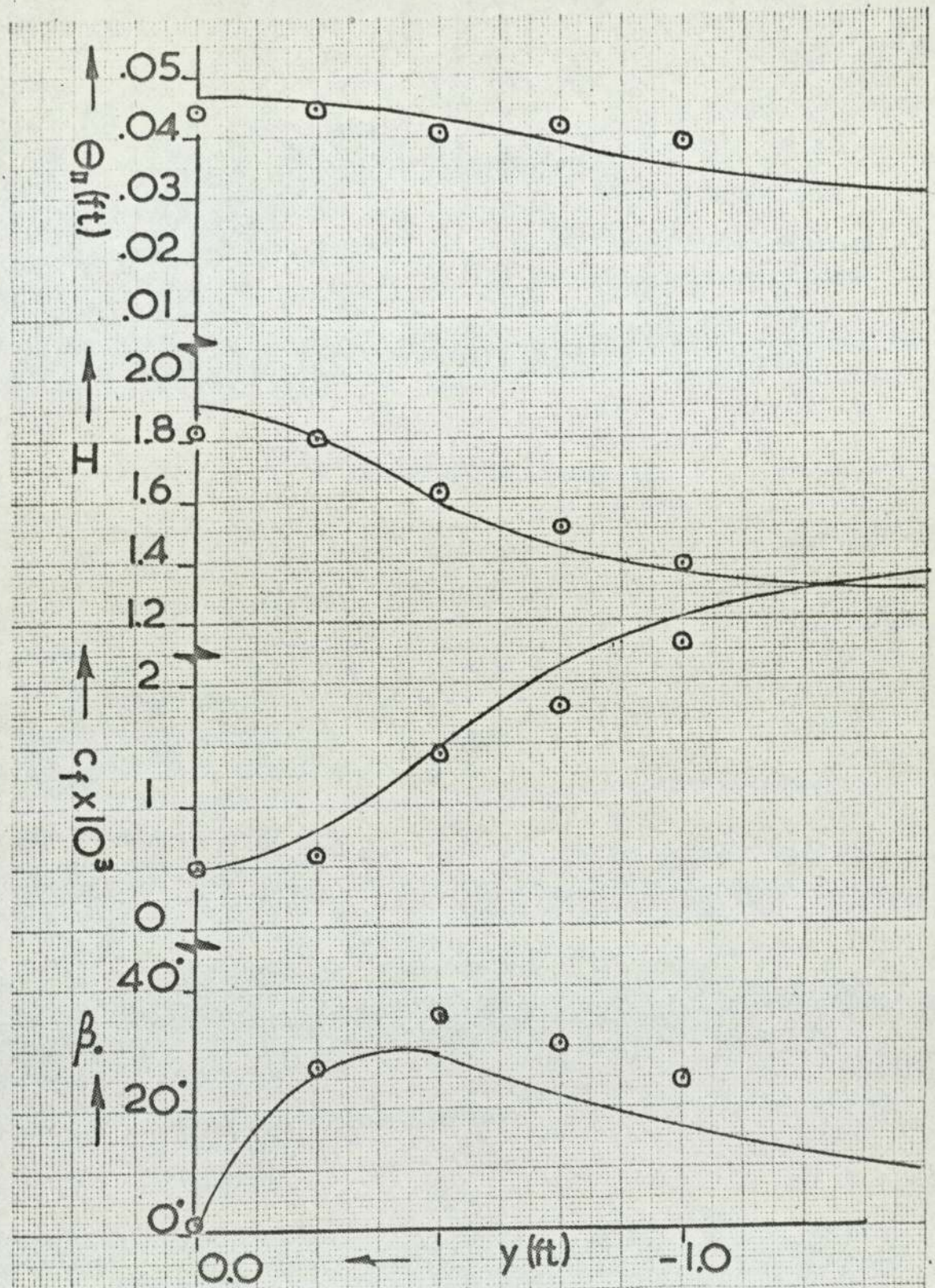


FIG (6.6.6) Hornung and Joubert [13]
Notation as in figure (6.6.2)
x=-1.75'

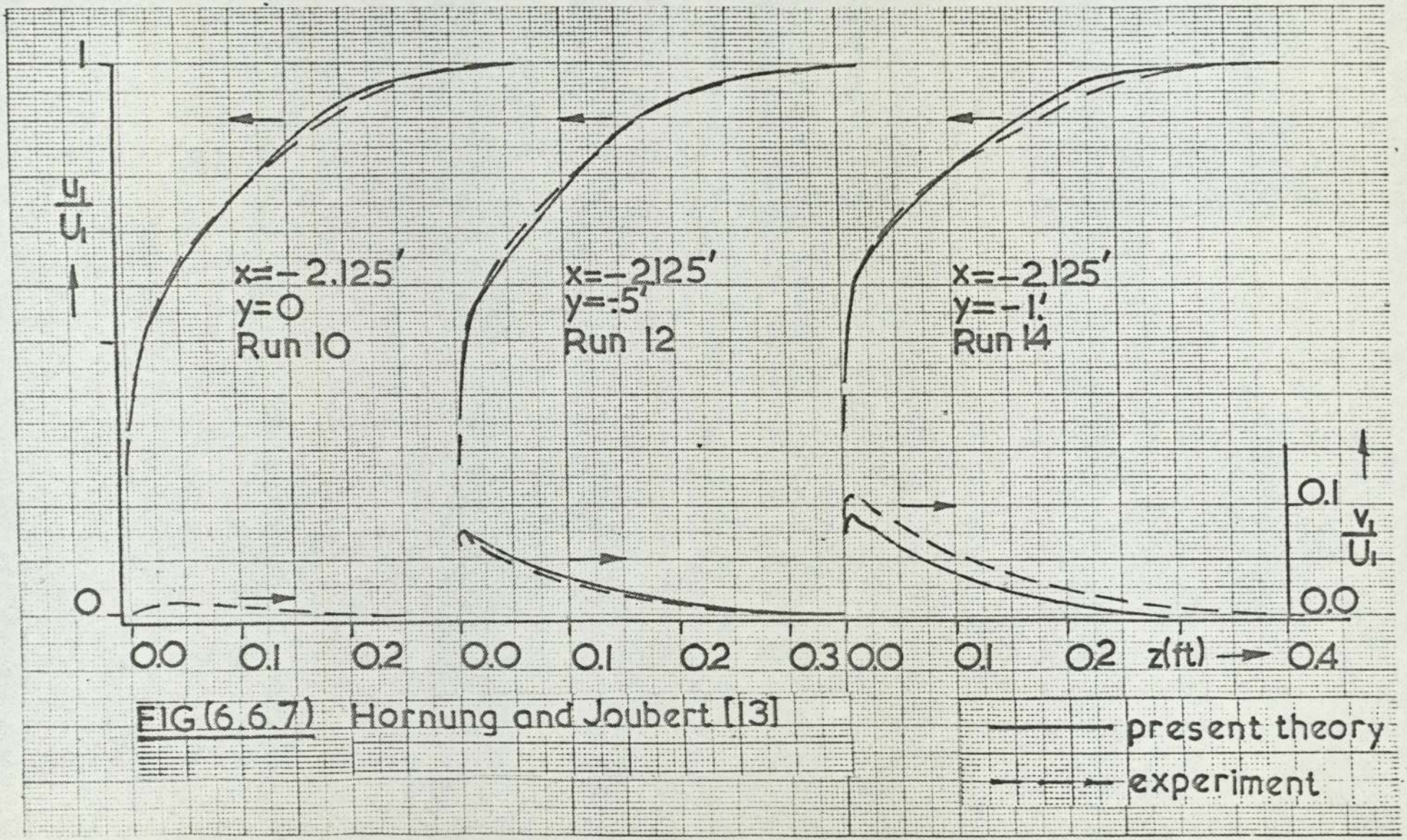


FIG (6.6.7) Hornung and Joubert [13]

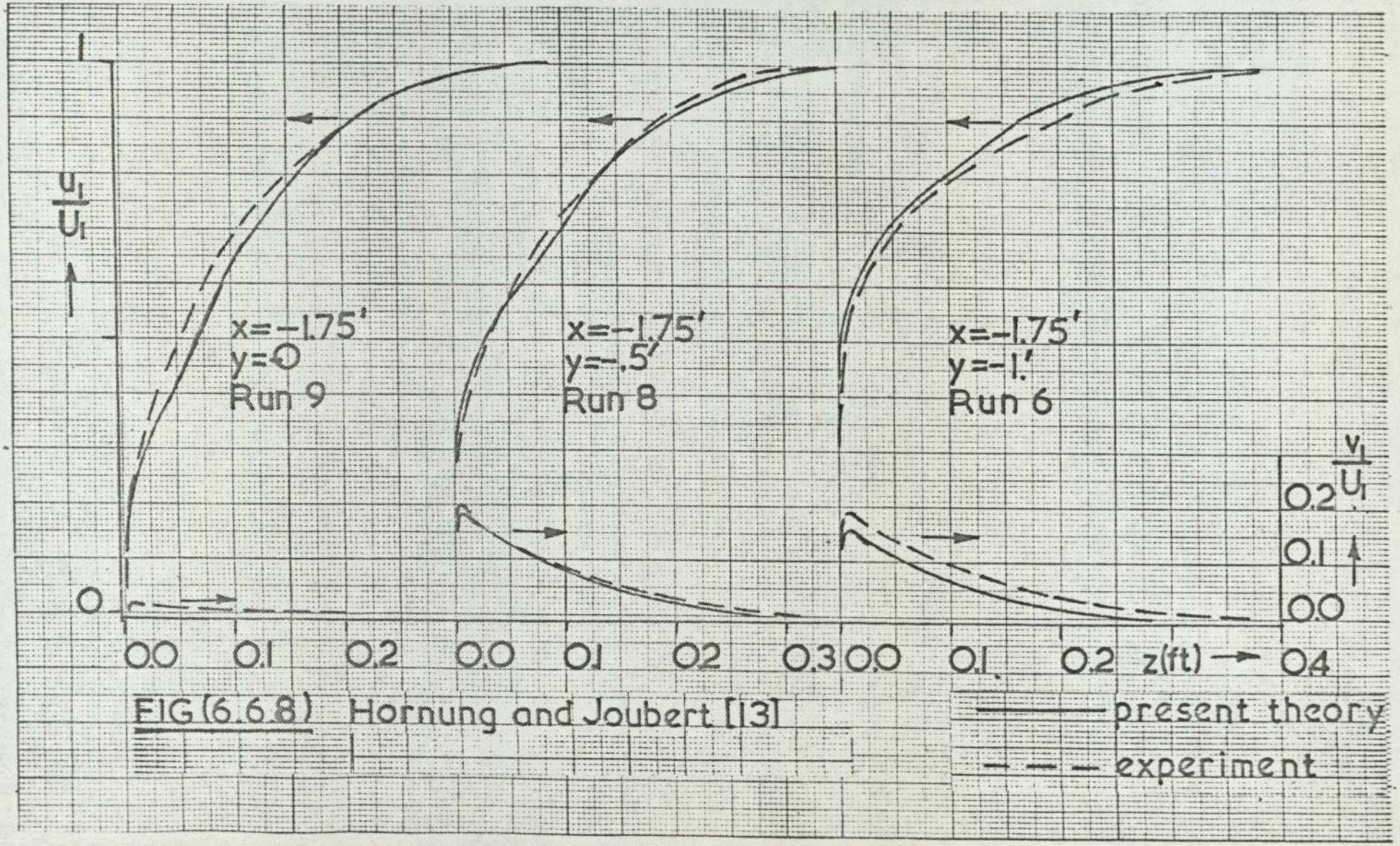


FIG (6.6.8) Hornung and Joubert [13]

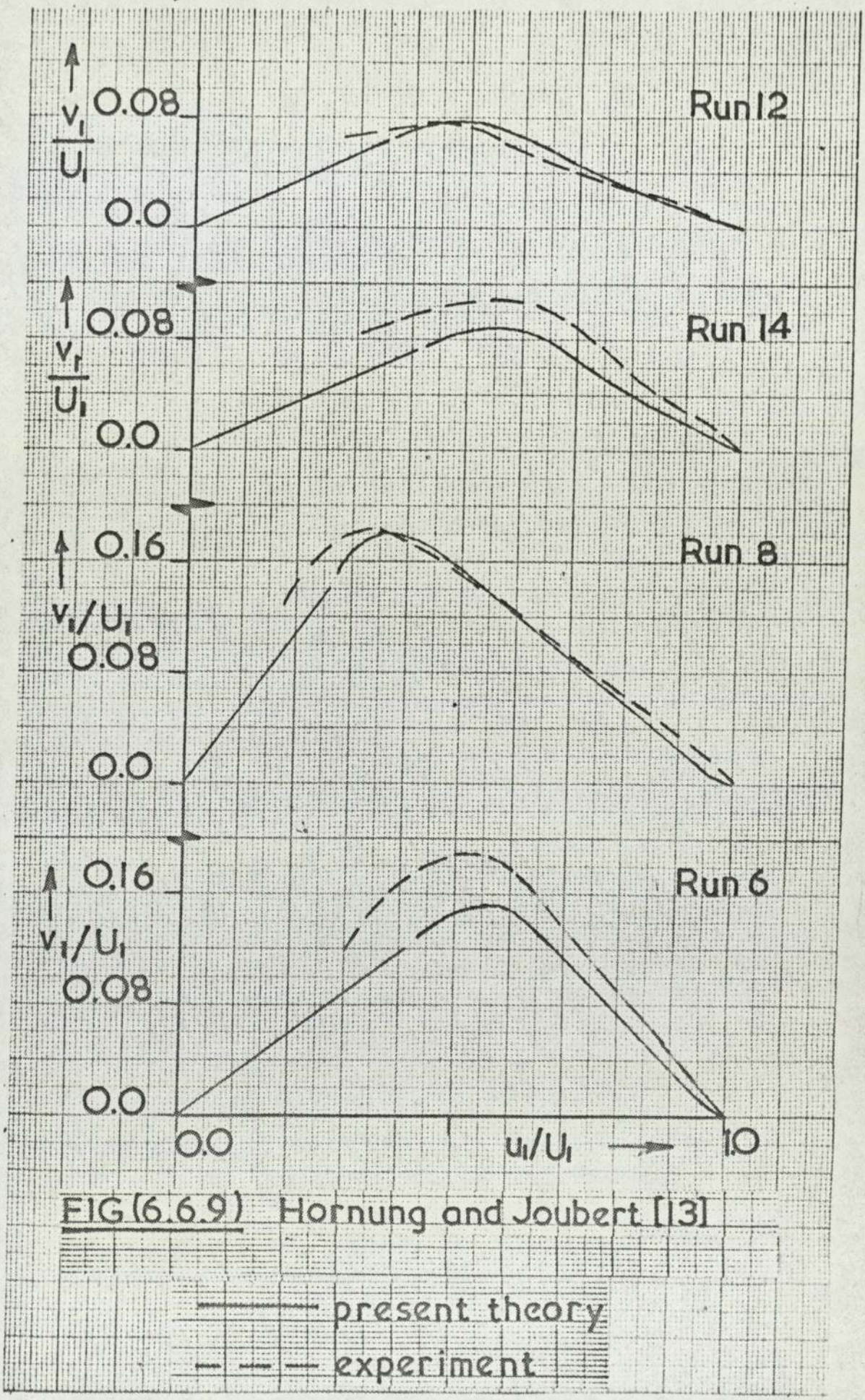
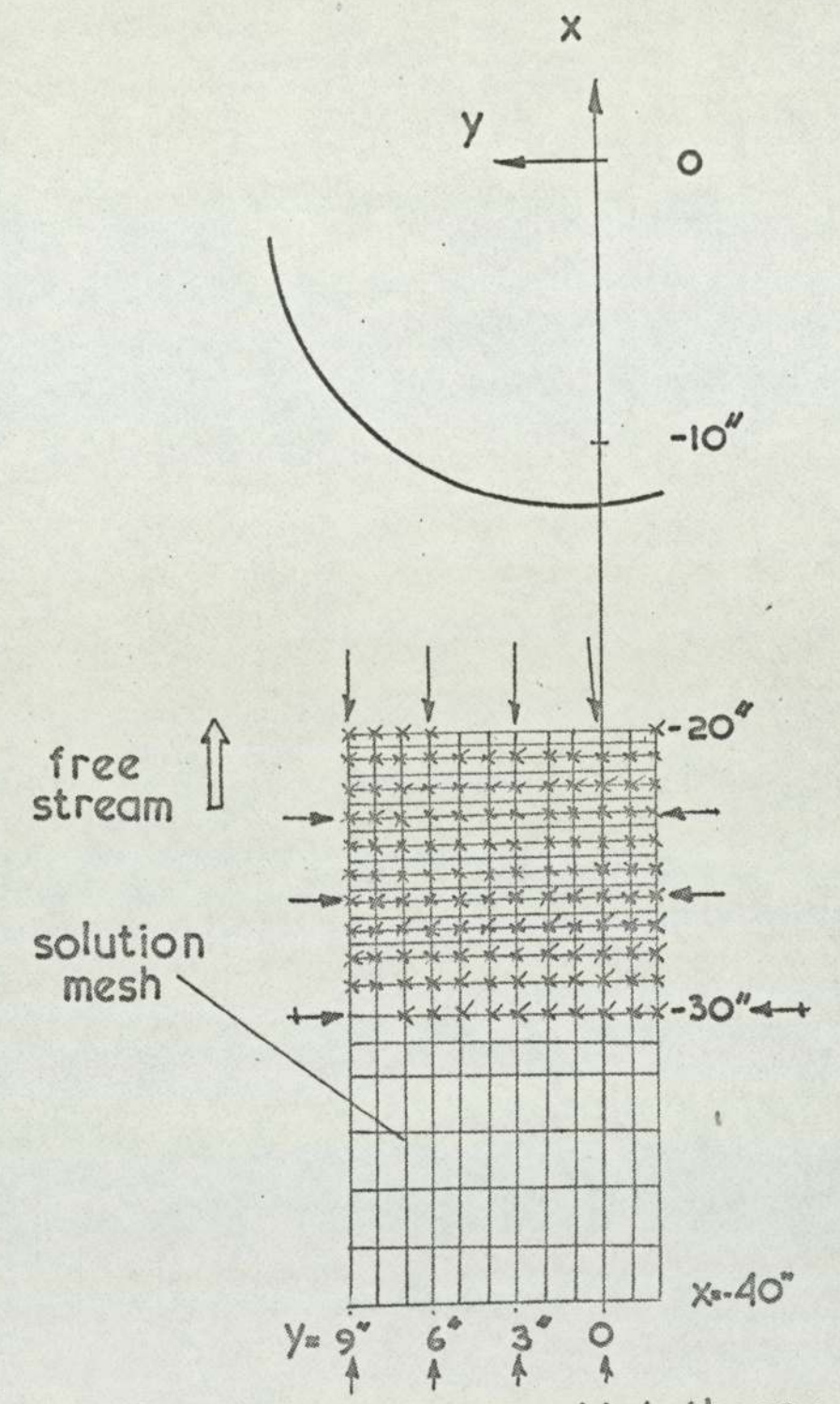
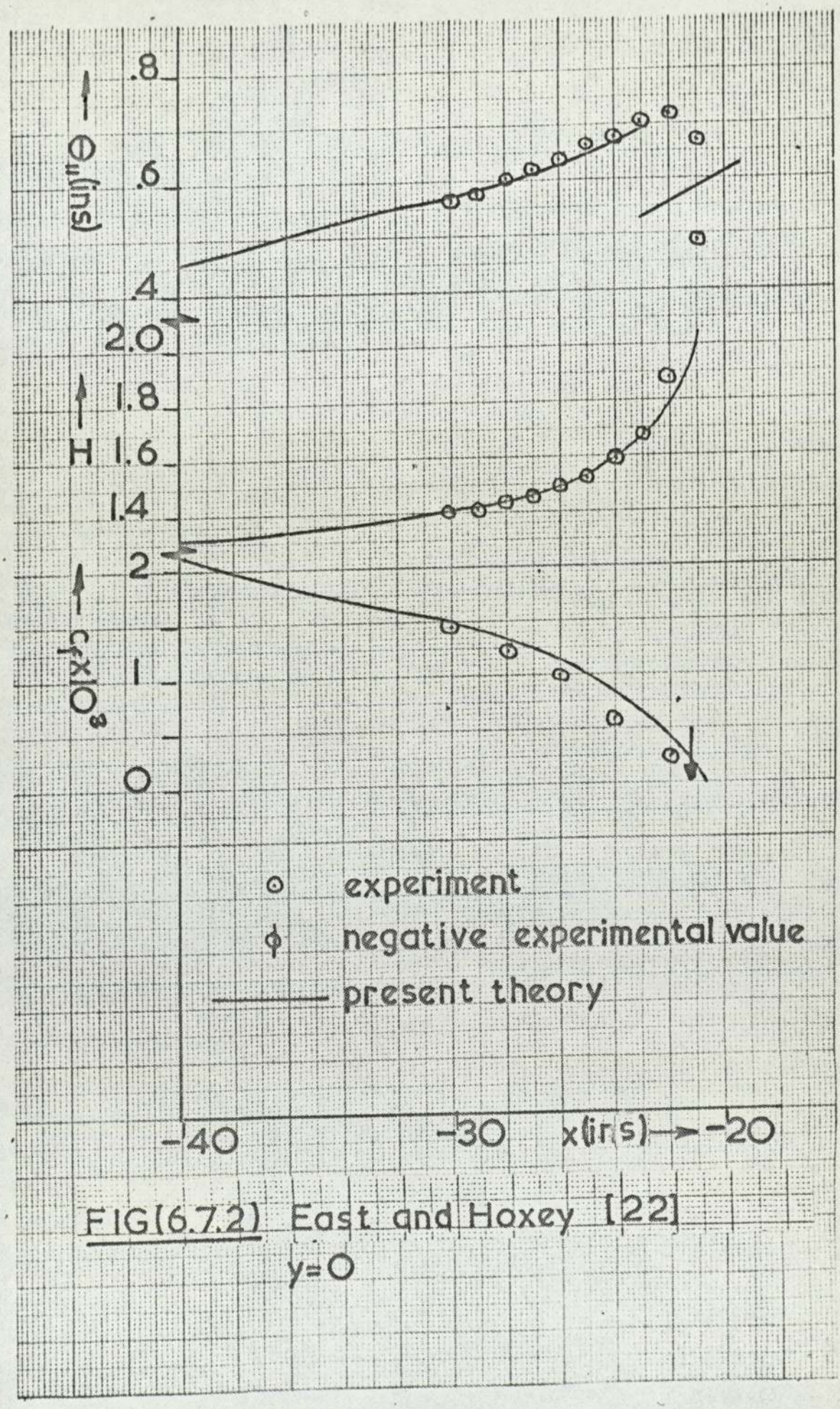


FIG (6.6.9) Hornung and Joubert [13]

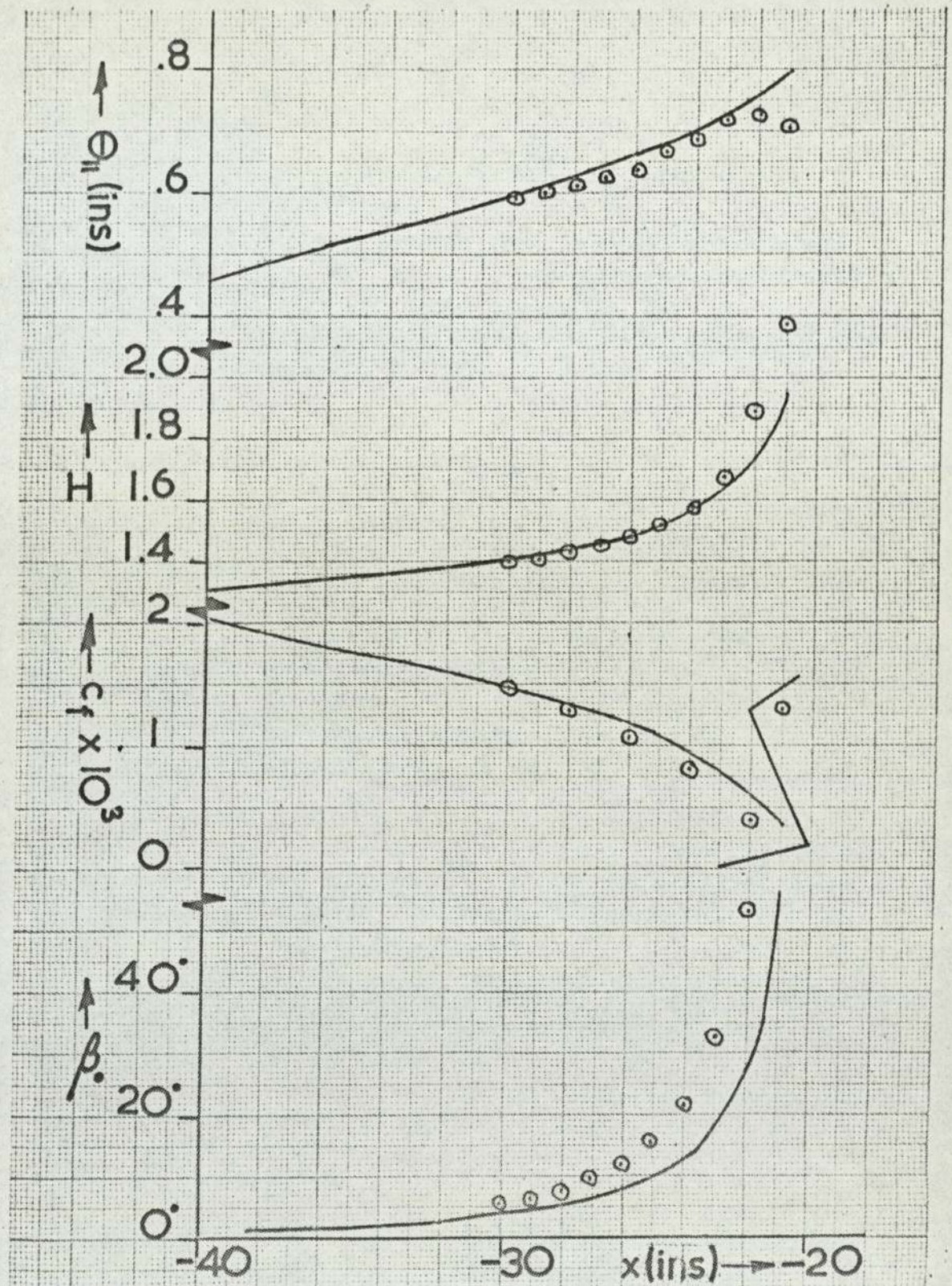


remaining notation as in figure (6.6.1)

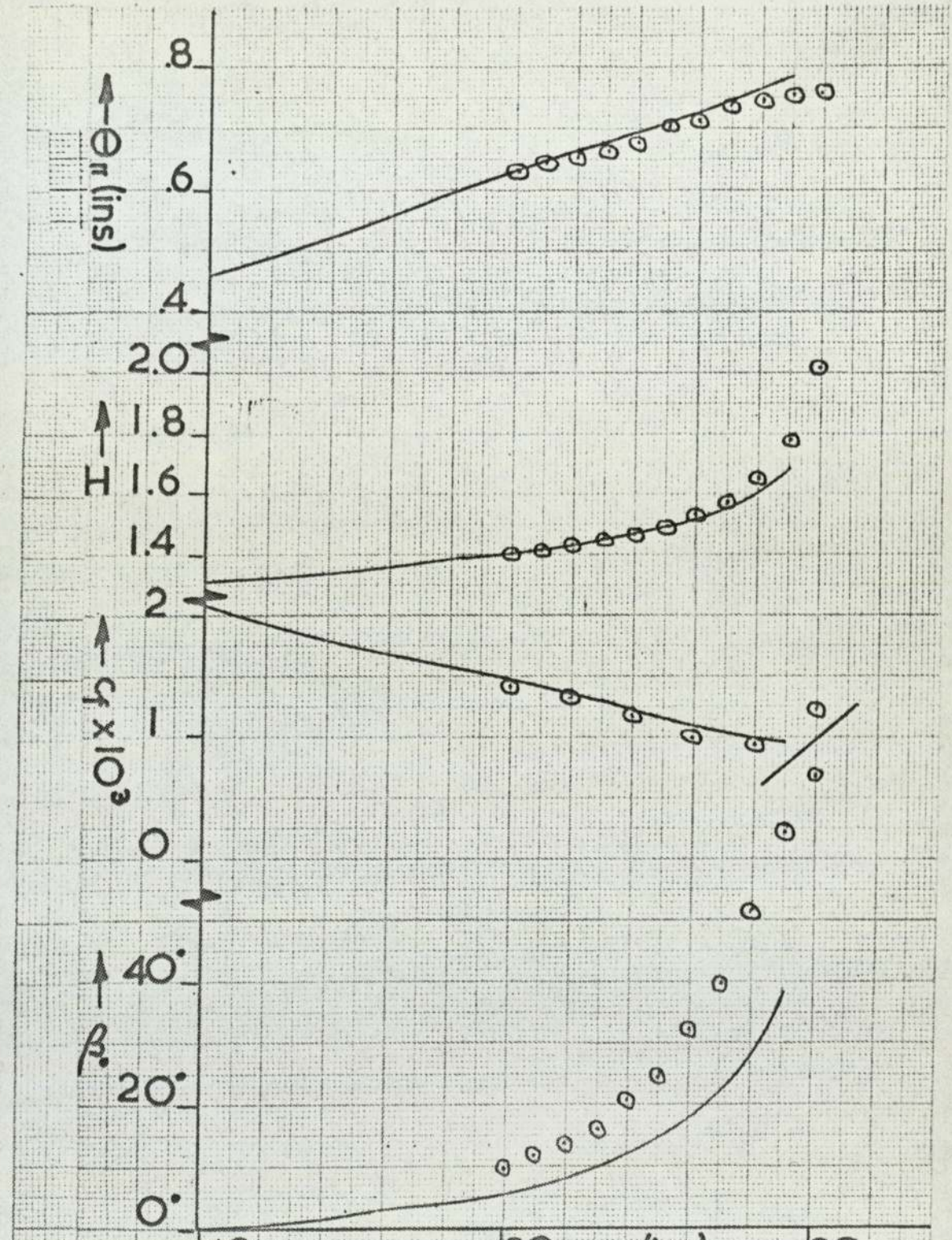
FIG(6.7.1) East and Hoxey [22]



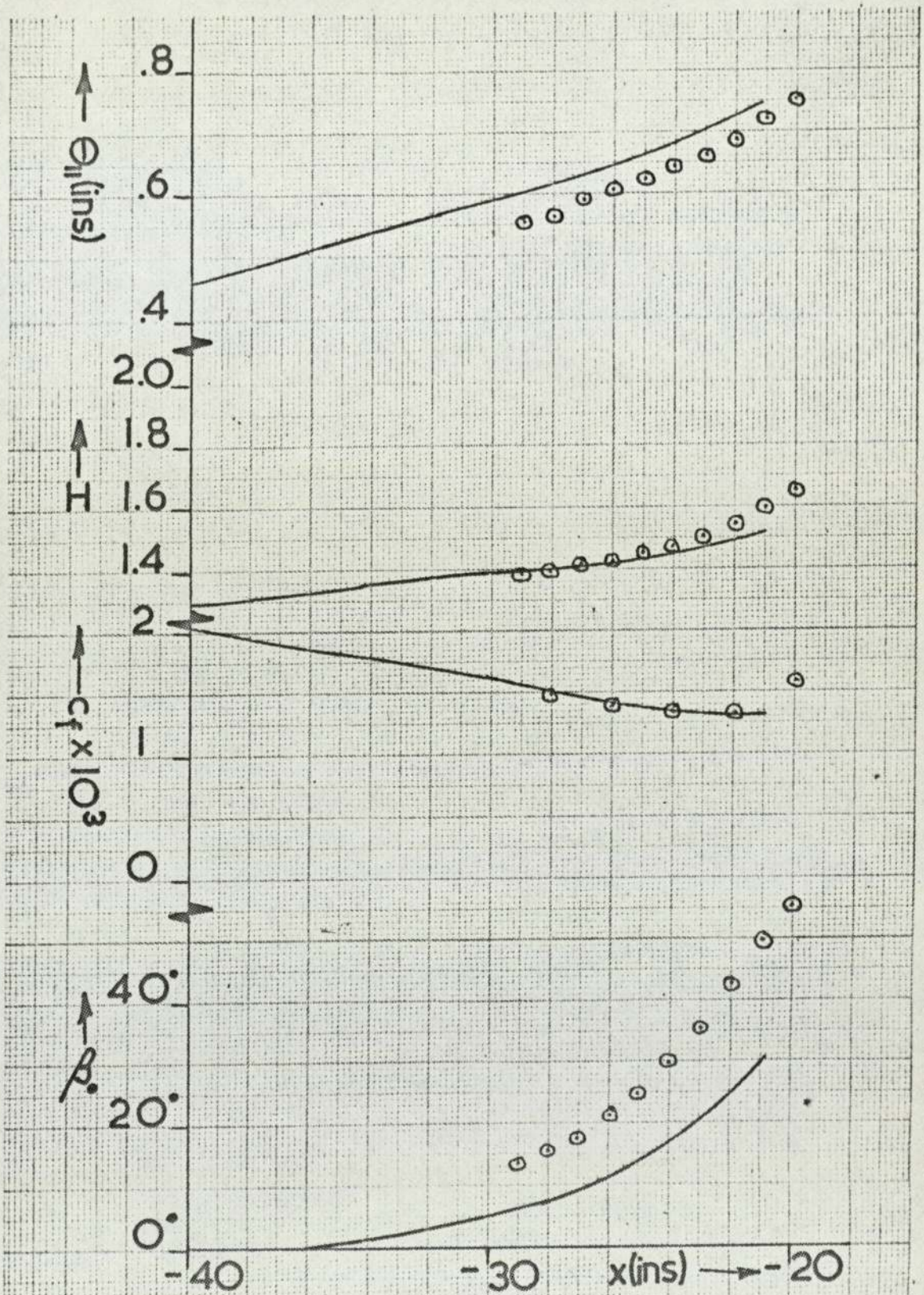
FIG(6.7.2) East and Hoxey [22]
y=0



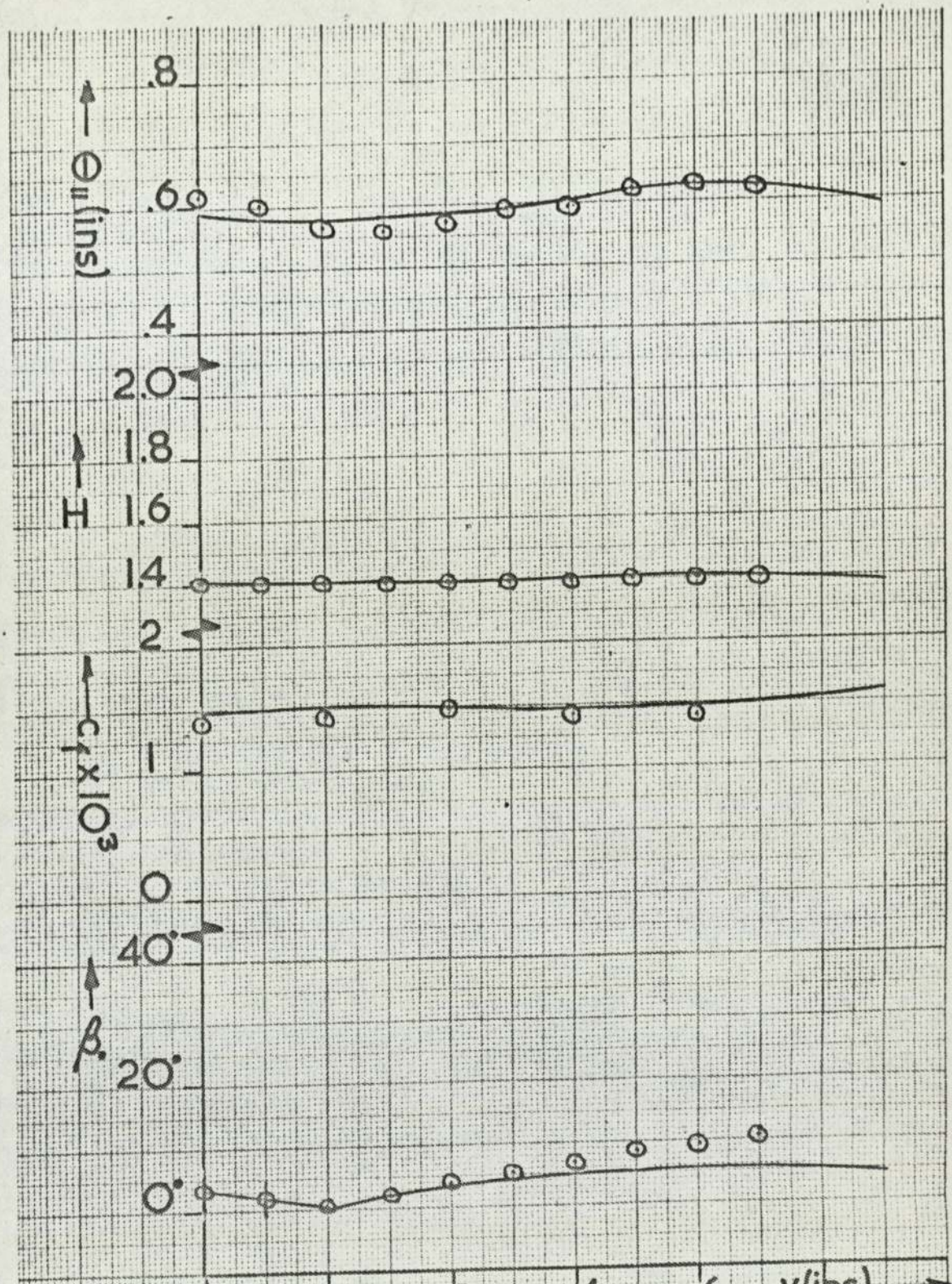
FIG(6.7.3) East and Hoxey [22]
Notation as in figure (6.7.2)
y=3"



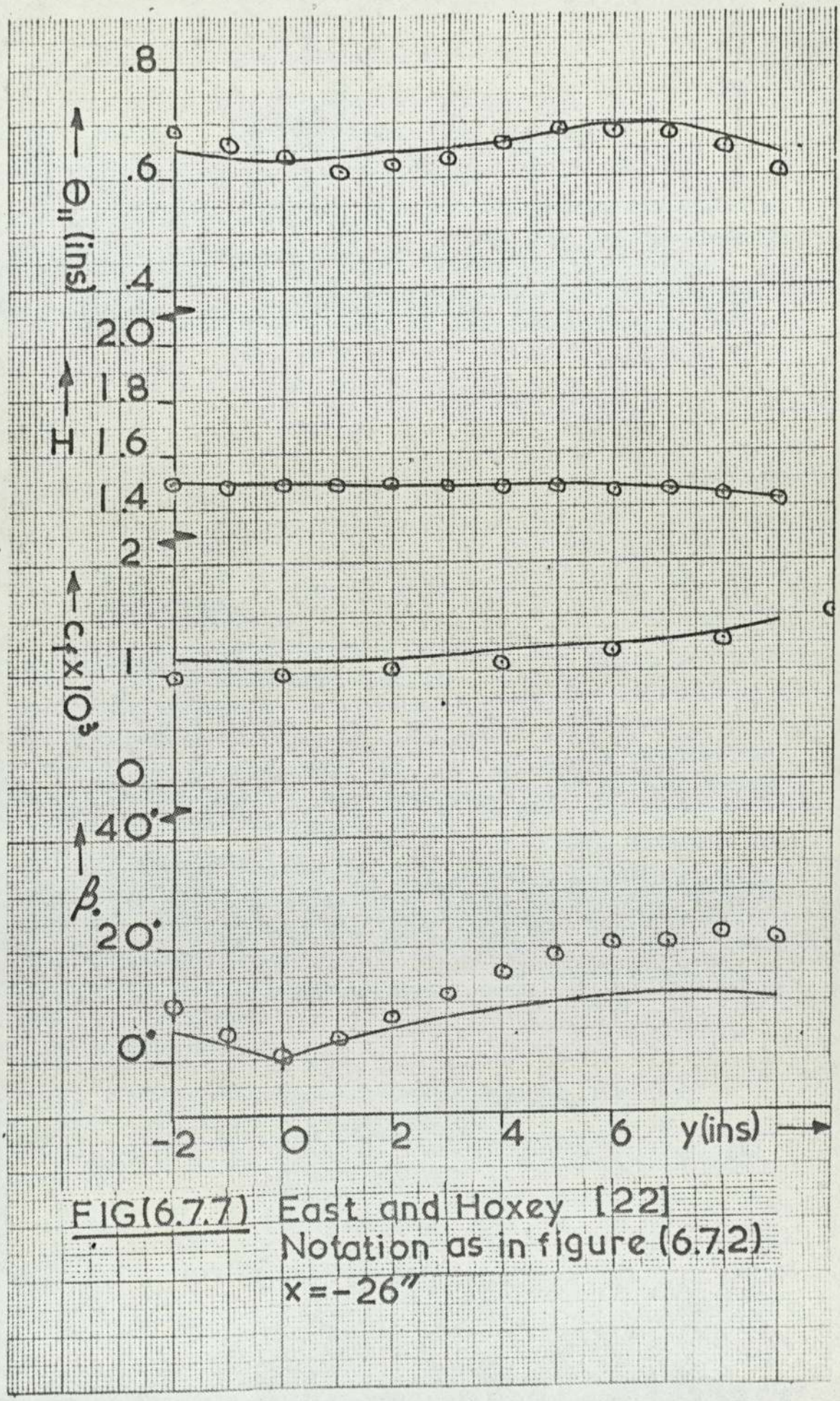
FIG(6.7.4) East and Hoxey [22]
Notation as in figure (6.7.2)
y=6"



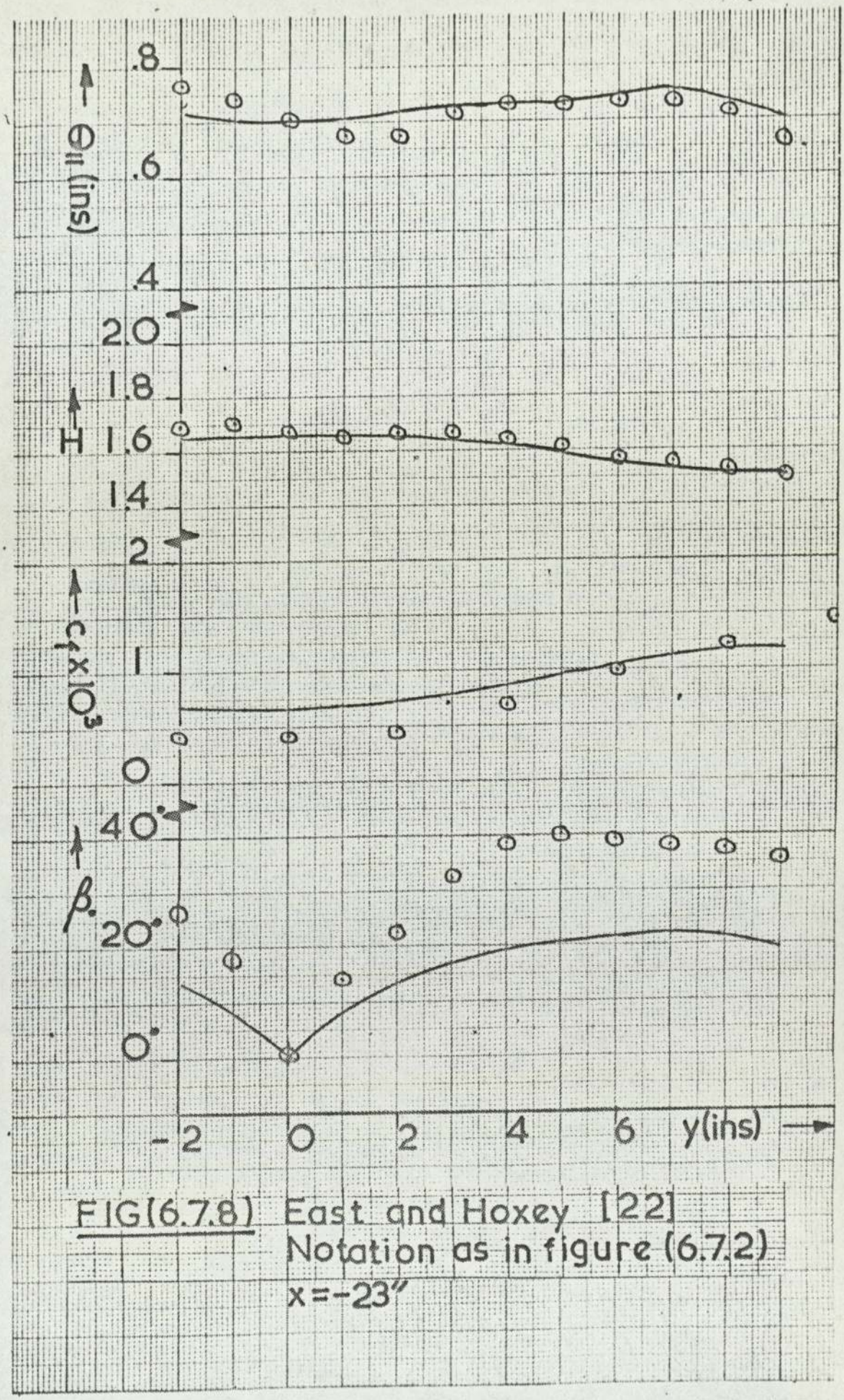
FIG(6.7.5) East and Hoxey [22]
Notation as in figure (6.7.2)
y=9"



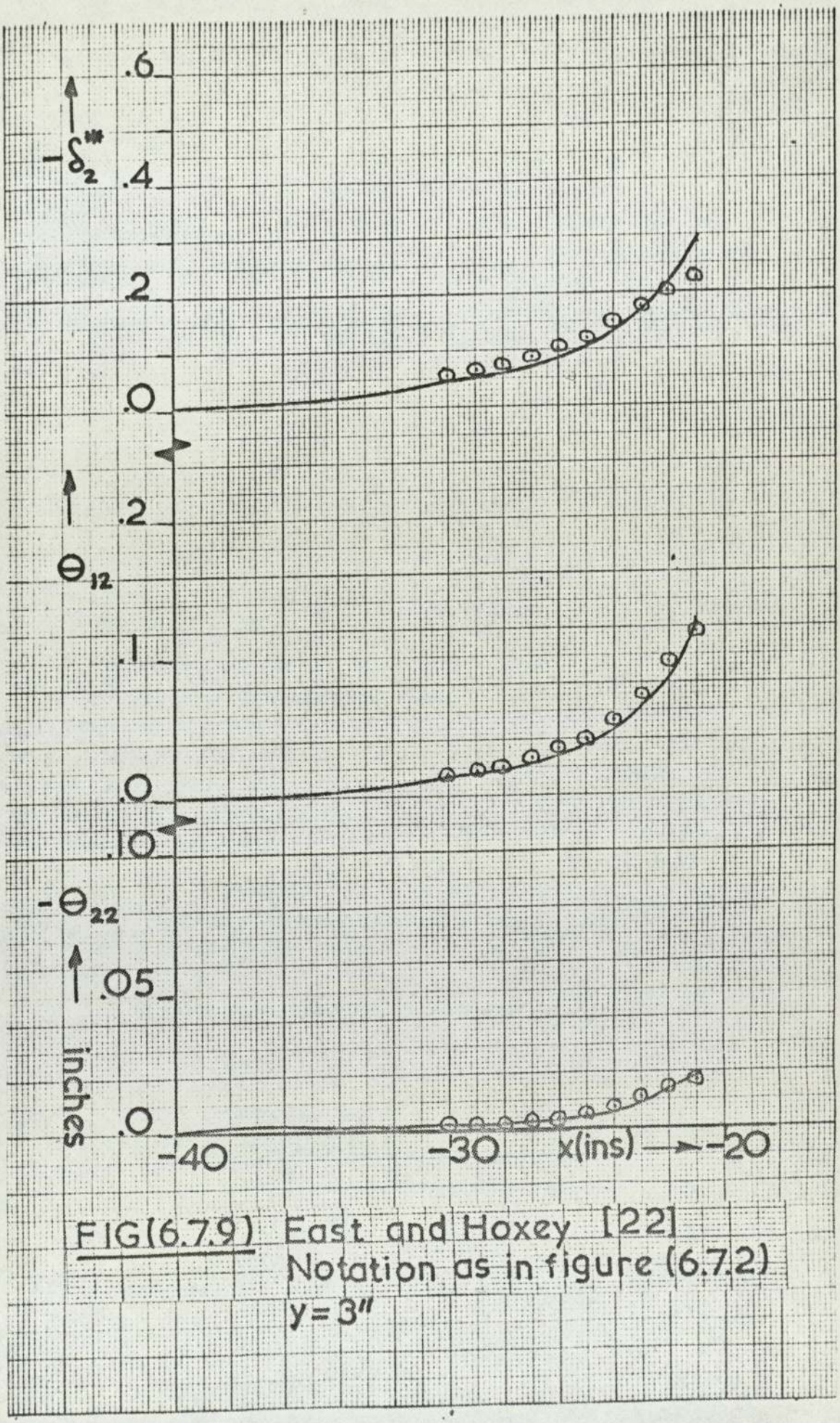
FIG(6.7.6) East and Hoxey [22]
Notation as in figure (6.7.2)
x = -30": solution matched to expt at this face



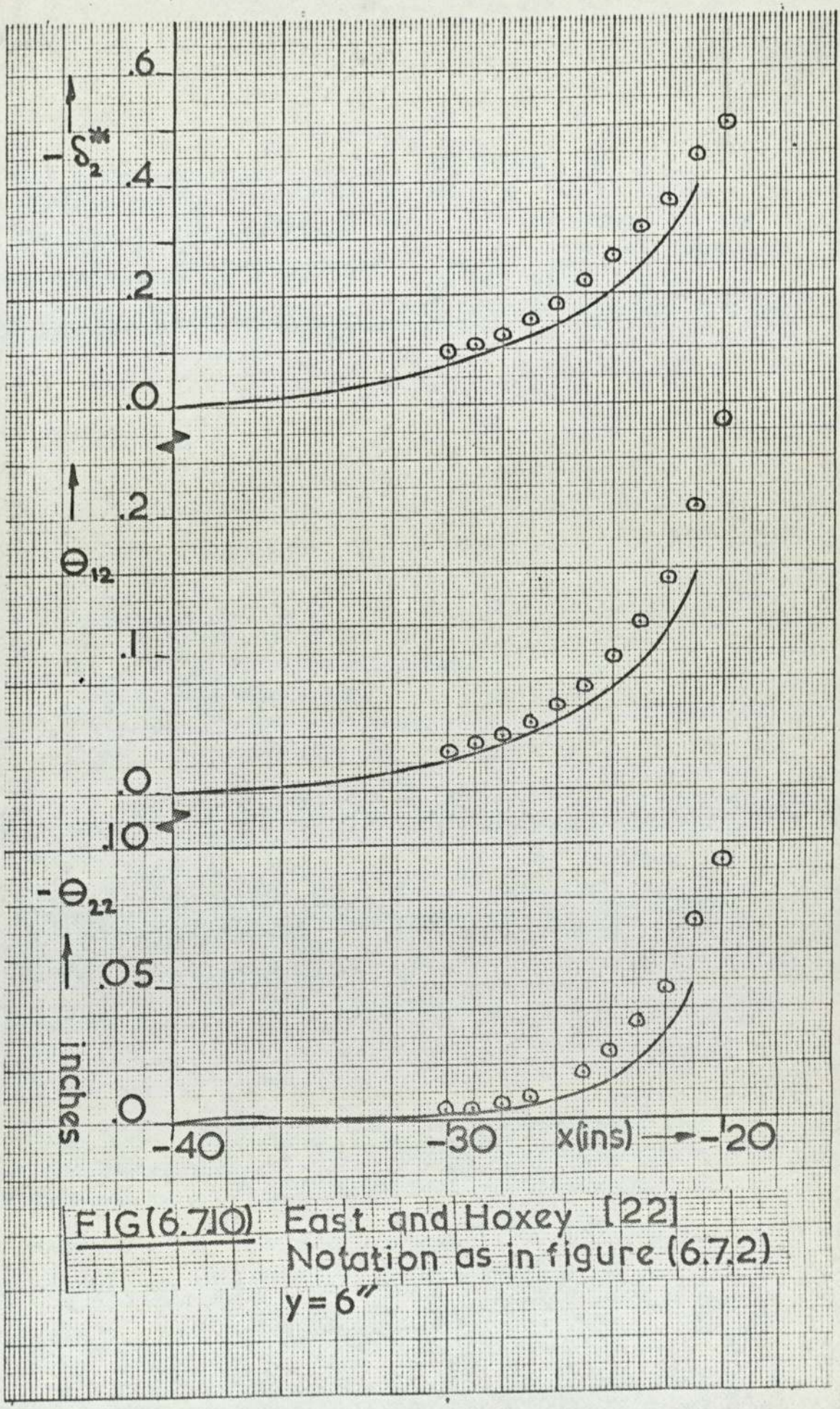
FIG(6.7.7) East and Hoxey [22]
Notation as in figure (6.7.2)
x = -26"



FIG(6.7.8) East and Hoxey [22]
Notation as in figure (6.7.2)
x=-23"



FIG(6.7.9) East and Hoxey [22]
Notation as in figure (6.7.2)
y = 3"



FIG(6.7.10) East and Hoxey [22]
Notation as in figure (6.7.2)
y = 6″

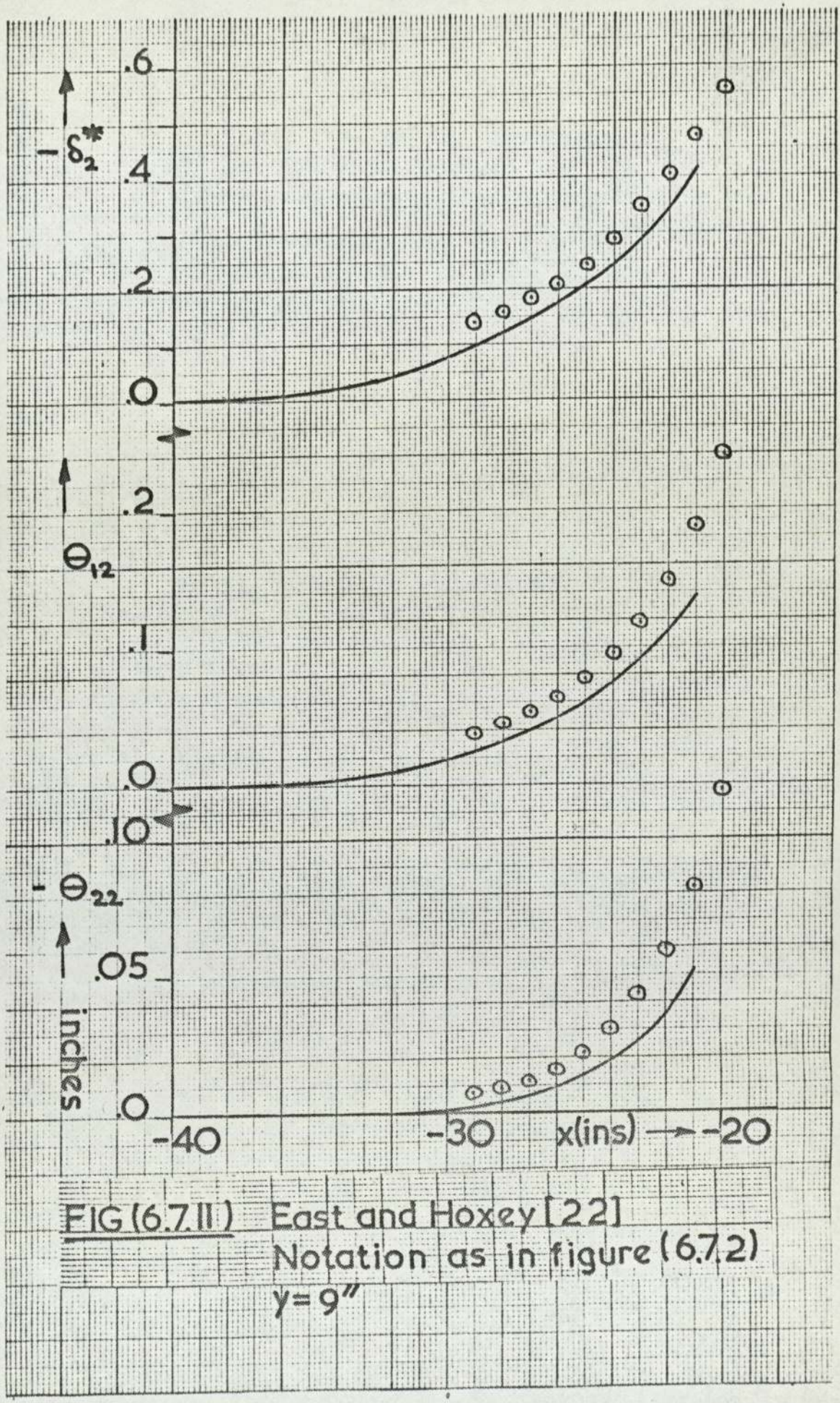


FIG (6.7.11) East and Hoxey [22]
Notation as in figure (6.7.2)
y = 9″

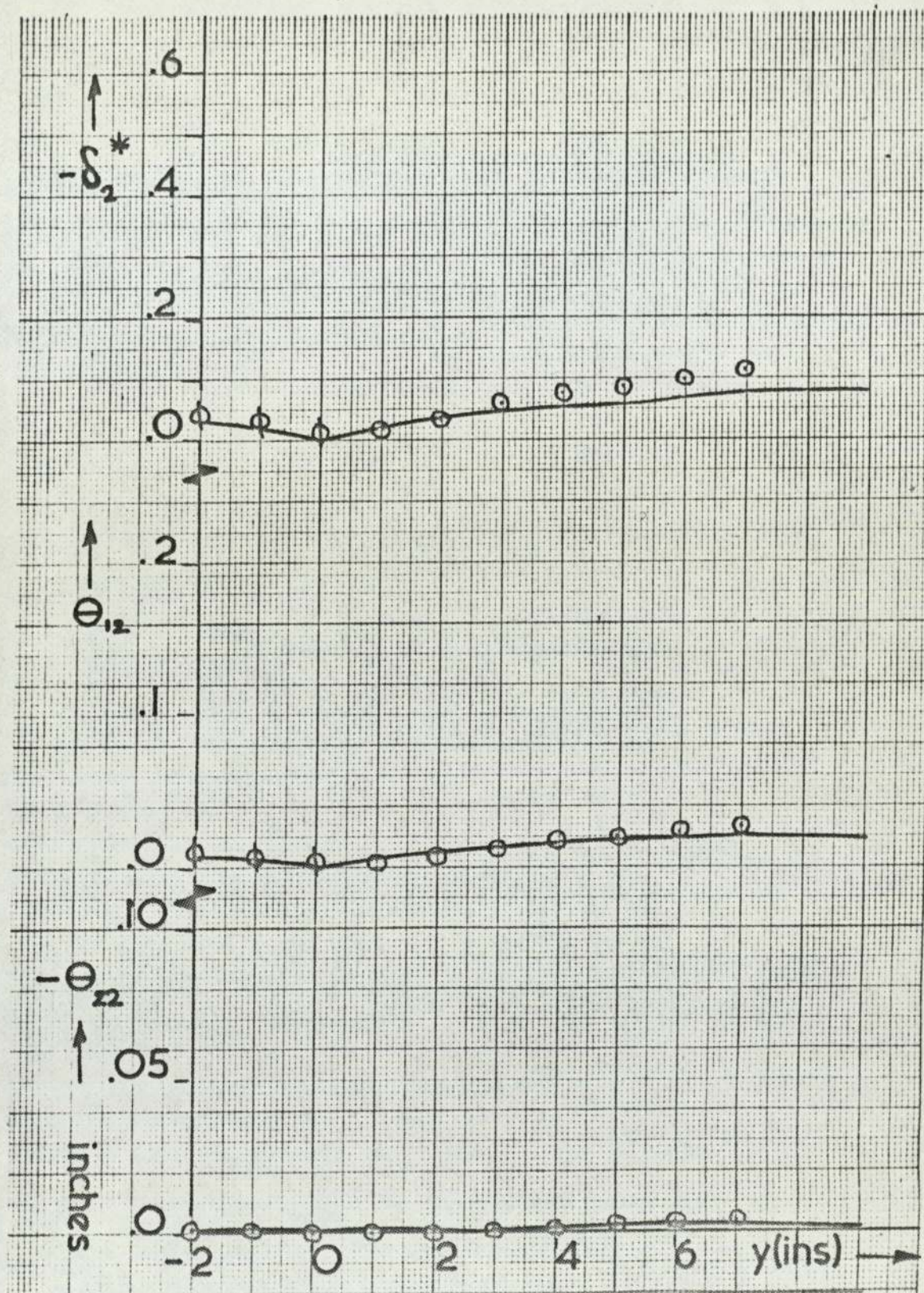


FIG (6.7.12) East and Hoxey [22]
Notation as in figure (6.7.2)
x = -30"

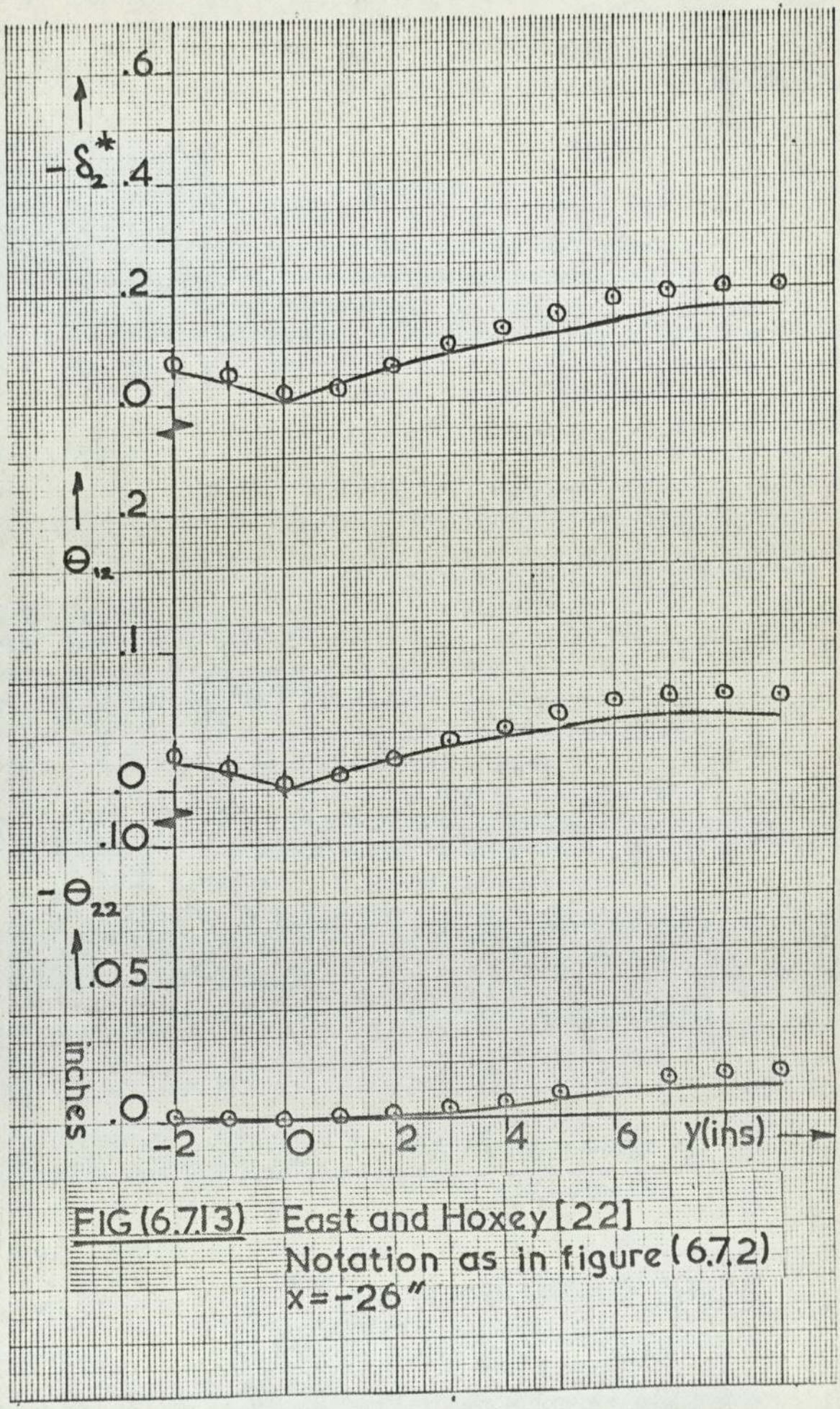


FIG (6.7.13) East and Hoxey [22]
Notation as in figure (6.7.2)
x = -26"

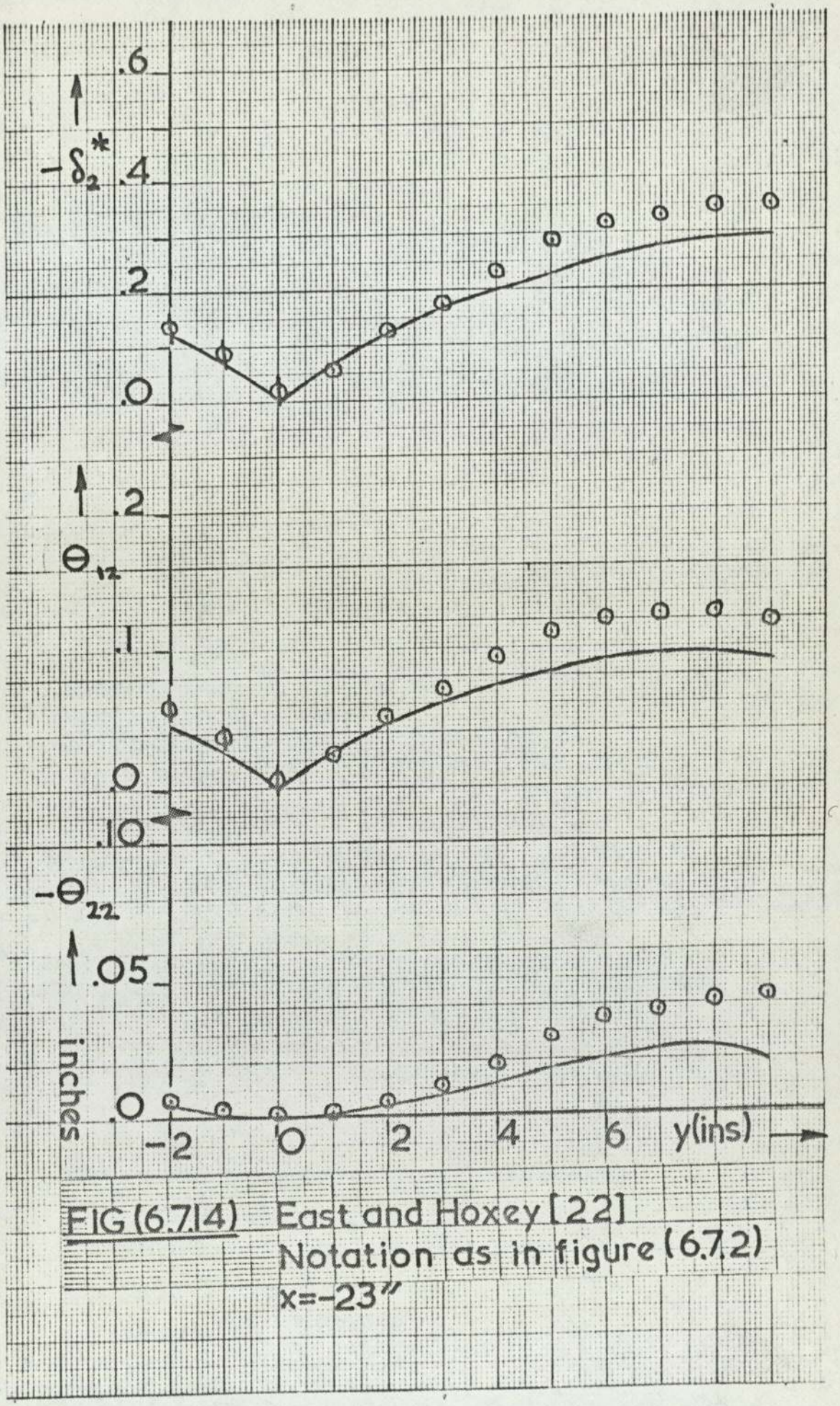


FIG (6.7.14) East and Hoxey [22]
Notation as in figure (6.7.2)
x=-23″