UNIVERSITY OF ALBERTA

COUPLED EQUATIONS FOR MODELLING UNSTEADY FLOW IN CHANNELS WITH FLOODPLAINS

BY

DANIEL K. TUITOEK ©

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE AND RESEARCH IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE

OF DOCTOR OF PHILOSOPHY

IN

WATER ROURCES ENGINEERING

DEPARTMENT OF CIVIL ENGINEERING

EDMONTON, ALBERTA

SPRING, 1995

ISBN 0-612-01768-0

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: DANIEL K. TUITOEK

TITLE OF THESIS: COUPLED EQUATIONS FOR MODELLING UNSTEADY FLOW IN CHANNELS WITH FLOODPLAINS

DEGREE: DOCTOR OF PHILOSOPHY

YEAR THIS DEGREE GRANTED: 1995

D.K. Tuitoek

Daniel K. Tuitoek
Egerton University
P.O.Box 536
Njoro
Kenya

Date: 26th January, 1995

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the faculty of Graduate Studies and Research for acceptance, a thesis entitled MODELLING UNSTEADY FLOW IN RIVERS WITH FLOODPLAINS submitted by DANIEL TUITOEK in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY in WATER RESOURCES ENGINEERING.

Dr. F. E. Hicks (supervisor)

Dr. N. Rajaratnam

Dr. P. Steffler

Dr. J. Shaw

Prof. S. Stanley

Dr. B.G.Krishnappan, external examiner

Date: 25th January, 1995

ABSTRACT

The occurrence of floods causing extensive property damage and in some cases, loss of life has made the need to curb or control floods in the floodplain a high priority in many places. To effectively carry out any meaningful control or management of floods, there is a great need for computational models that can accurately predict the stage-discharge relationship in rivers with floodplains.

In past studies the conventional stage-discharge prediction methods in compound channels have been found either to underestimate or overestimate floodplain discharge when small depths occur on the floodplain. In this research study, a model based on the St. Venant equations of flow, with incorporation of terms to account for the momentum transfer phenomenon, was developed. For the main channel, the full dynamic equations were used, while in the floodplains a diffusive model was used. Both included mass and momentum transfer terms. The resulting model was called the "coupled characteristic-dissipative-Galerkin 1-D model" (CCDG 1-D model). The resulting equations were solved by the characteristic-dissipative-Galerkin (CDG) finite element method. This numerical technique was adopted over traditional finite difference schemes because the finite element method can handle subcritical and supercritical flow reaches simultaneously. The CDG method was chosen in particular because of its robust ability to provide accurate solutions for highly dynamic events.

Results from the CCDG 1-D model obtained when simulating steady and unsteady flow in compound channels were compared to observed experimental data. The steady state results in straight compound channels clearly showed that the CCDG 1-D model predicted the stage-discharge relationship as well as any existing method used to compute compound channel flow. This was especially true for low depths in the floodplain. The unsteady results showed that the inclusion of apparent shear had marginal effect.

The CCDG 1-D model was also able to simulate such practical problems as theflow through a dike breache and steady flow in a meandering compound channel. However, before the proposed formulation can be applied to natural channels with confidence, field data is required to determine how variance of river aspect ratios, channel shape, relative roughness and sediment transport affect the flow in compound channels.

ACKNOWLEDGMENT

I would like to express my deep gratitude to my supervisor Dr. F.E. Hicks for providing great support and guidance in this study. Special thanks are also extended to Dr. P. Steffler for his interest and advise in this work. The support from the Dr. N. Rajaratnam throughout my graduate program is also highly appreciated. I would also like to thank Dr. B.G. Krishnappan for agreeing to be the external examiner.

The financial support from commonwealth scholarship plan of Canada which enabled me to pursue this degree is highly appreciated. Additional financial support from the department of civil engineering at University of Alberta in form of teaching assistance and from my supervisor Dr. F.E. Hicks towards the end of my program is also acknowledged. My special thanks also goes to Egerton University, Kenya for availing me study leave to undertake this degree.

On a personal note, I would like to thank my wife Prisca and my son Kimalel, for their patience, moral support and love that enabled me to undertake this degree. Last but not least, I would like to thank my colleagues and friends for the help and encouragement throughout my studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

A, A_c	area of main channel
A_f	area of floodplain
A_l	area of left floodplain
A_r	area of right floodplain
A_t	total area of compound channel
B	bed width of main channel
B_f	bed width of floodplain
C	Chezy's conveyance coefficient
C_*	dimensionless Chezy's conveyance coefficient
C_{fa}	apparent friction factor at the interface between the main channel and the floodplain
F	Froude number
f_c	main channel friction factor
f_f	floodplain friction factor
f_i	test function
f_j	interpolation function
g	acceleration due to gravity
H	main channel depth
H_f	floodplain depth
H_l	left floodplain depth
H_r	right floodplain depth
H_w	channel step or weir height
i, j, k	integer indices
J	Jacobian (CCDG 1-D model)
k_e	effective channel roughness

LIST OF SYMBOLS (continued)

K	coefficient of conveyance
$\mathbf{K}$	stiffness matrix
M_{trf}	momentum transfer from the main channel to the floodplain
M_{tr}	the total momentum transfer from the main channel and the floodplains
M_{trl}	the momentum transfer between the main channnel and the left floodplain
M_{trr}	the momentum transfer between the main channnel and the right floodplain
n	Mannings coeffient
n_k	Mannings roughness for the kth section
n_e	equivalent Mannings roughness coefficient
P_t	compound channel wetted perimeter
P, P_c	main channel wetted perimeter
P_f	floodplain wetted perimeter
P_k	wetted perimeter for the kth section
Q, Q_c	discharge for the main channel
Q_f	floodplain dicharge
Q_f'	discharge for the floodplain with no flow interaction
Q_m'	discharge for the main channel with no flow interaction
Q_t	total compound channel flow
q	lateral outflow or inflow
q_f	lateral inflow into the flood plain
q_l	lateral inflow into the left floodplain
q_r	lateral inflow into the right floodplain
R, R_c	hydraulic radius for the main channel
R_f	hydraulic radius for the floodplain
R_k	hydraulic radius for the kth section

LIST OF SYMBOLS (continued)

R_t	hydraulic radius for the compound channel
$\boldsymbol{S}$	mass matrix
S_o	longitudinal main channel slope
S_f	main channel friction slope
S_{ff}	floodplain friction slope
t	time coordinate
T_w	compound channel top width
$\overline{V}$	average compound channel velocity
V, V_c	average main channel velocity
V_f	average velocity in the floodplain
V_l	average velocity in the left floodplain
V_r	average velocity in the right floodplain
V_x	component of the lateral out flow velocity in x-direction
ΔV	difference of average velocity of the main channel and floodplain
w	lateral velocity
$\boldsymbol{W}$	upwinding matrix
x	longitudinal direction coordinate
Δx	computational distance step
z	vertical coordinate
Z_1 ....Z_6	trapezoidal cross-sestion slope notations
α	kinematic wave speed coefficient
$\boldsymbol{\Phi}$	nodal values of solution vector
γ	specific weight of water
φ_f	a floodplain apparent shear force index
φ_m	a main channel apparent shear force index
θ	implicitness

LIST OF SYMBOLS (continued)

τ_a	average apparent shear stress at the interface between main channel and floodplain
τ_c, τ_o	main channel boundary shear stress
τ_f	floodplain boundary shear stress
$\tau_{f\infty}$	undisturbed boundary shear stress in the floodplain
ρ	density of water
ω	upwinding coefficient in cdg 1-D scheme

1.0 INTRODUCTION

Extensive development in river floodplains has made the need to minimize the potential for overbank flooding a high priority in many countries. Despite recent moves to limit floodplain development the fact remains that, in many areas, people and property are at tremendous risk from floods. This situation presents three practical problems for water resources engineers. First, they must be able to identify those areas of the floodplain which are at risks from floods. Second, they must be able to assess the impact of flood mitigation schemes, such as dikes, on flood hydrographs and flood levels, not only to design these mitigation works, but also to assess the effects of such schemes on unprotected areas both upstream (due to potential backwater effects) and downstream (due to the loss of flood storage area, and therefore reduced flood peak attenuation). New or extended flood mitigation schemes necessitate a new analysis to reassess both the adequacy of existing works and the increased impact on unprotected areas. Third, as the 1993 flood disaster in the midwestern United States has illustrated, engineers have to be able to provide updated flood forecasting information under situations when such flood control schemes fail (Williams, 1994).

Over the past three decades, extensive research efforts have been directed to the problem of quantifying the stage-discharge relationship in channels with inundated floodplains. Although great advances have been made in the understanding of the flow interactions between the main channel and the floodplains, most of these investigations have concentrated on steady flow situations. In practice, unsteady flow analyses are generally based on "hydrologic" flood routing techniques, with the use of "hydraulic" flood routing techniques generally reserved for very dynamic floods, such as those resulting from a dam break or an ice jam release. Furthermore, these dynamic models have been limited to one-dimensional analyses of open channels flows. Such models are incapable of handling flow situations in which floodplain flows are independent of the flow in the main channels, such as when dikes are overtopped and/or breached.

This study addresses this need by presenting a formulation of the problem which considers a channel and its inundated floodplains as three separate yet interdependent conveyance channels. The channel flow is modelled with the full (dynamic) one-dimensional equations of open channel flow, while the floodplains are modelled with diffusive wave equations. Interdependence between the channel and floodplains is established through mass and longitudinal momentum transfer functions. These momentum transfer functions include both convective momentum transport as well as the apparent shear force generated along the interfaces between the main channel and the floodplains. The proposed formulation provides for significantly less computational time as compared to a full two-dimensional dynamic model of a channel with floodplains.

The validity of the proposed formulation is established for steady flow situations through comparisons to observed laboratory experiments on compound channel flows in trapezoidal sections by Prinos and Townsend (1984), and Wallingford (1992). Comparisons to the results of existing calculation techniques are also provided.

The ability of the model to simulate unsteady flow in an open channel is then verified through a comparison with laboratory experiments conducted by Treske (1980) in a straight channel. Comparisons to the performance of more conventional one-dimensional models are also provided for comparative purposes. Next, the unique capabilities of the proposed formulation are illustrated for two practical situations. A hypothetical scenario involving floodplain inundation through a breached dike is simulated, illustrating the ability of the model to handle independent flow situations in the channel and floodplain. Finally, the unique ability of the formulation to model steady flow in a meandering channel and its floodplain is illustrated and compared with flume data collected by Smith (1978).

2.0 LITERATURE REVIEW

2.1 Introduction

This chapter examines past research into the flow in channels with inundated floodplains, beginning in section 2.2 with an overview of the geomorphic characteristics of rivers that define the practical situations encountered. A review of the investigations into the steady flow problem is presented in section 2.3, beginning with an overview of experimental studies of the flow characteristics in both straight and meandering channels with inundated floodplains. An overview of the steady flow analysis techniques currently used in practice, is also presented. Section 2.4 presents a summary of the research to date into unsteady flow modelling, including variations on the one-dimensional open channel flow equations. Finally sections 2.5 and 2.6 outlines the framework used in handling the mass and momentum transfer associated with interaction of flow between the main channel and the floodplain.

2.2 Practical Considerations in Modelling Channels with Floodplains

An examination of types of channel and floodplain geometries is necessary in order to define the situations to be modeled. This is done most directly by considering the planform geometry of the river channel and the relation of the channel to the valley wall.

Rivers are generally classified as young, mature or old (Schumn, 1963). "Young" rivers, that is rivers in the early stages of development tend to be relatively straight and confined within valley walls (Brice, 1964). The top of the valley wall typically exceeds all flood stages, so the channel never experiences overbank flow.

As rivers mature they begin to meander, eventually eroding the valley wall and widening the river valley. "Mature" rivers develop floodplains in the process but at this early stage these floodplains are discontinuous, alternating from bank to bank as the meandering channel impinges on the valley wall (Brice, 1964). If the valley wall is relatively inerodible, floodplain development and channel widening may be inhibited, leading to the development of confined meanders, as illustrated in Figure 2.1(a). Although the loops of the meanders are distorted by the confining valley walls, such channels are often surprisingly regular (Brice, 1964).

Mature rivers present a difficult modelling problem, because of these discontinuous floodplains. When the flow exceeds bankful, water passes from one floodplain, across the channel to the other floodplain. In this situation two complexities must be considered. First, the flow in the floodplain tends to be subjected to a steeper gradient than that in the channel, because of the greater distance traveled through the meandering channel for a similar drop in elevation (Leopold, Wolman, and Miller, 1964) The ratio of these slopes is defined as the sinuosity of the channel.

$$Sinuosity = \frac{Channel\ length}{Valley\ length} = \frac{Valley\ Slope}{Channel\ bed\ slope} \qquad [2.1]$$

The second complexity to be considered involves the fact that the flow in the channel is not aligned with the flow in the floodplain, particularly at the cross-over, where the difference may be as much as 90°.

Rivers are described as "old" when they have reached the stage of floodplain development where the valley walls have little or no influence on meander development (Brice, 1964). The planform pattern in this case tends to depend upon the homogeneity of the alluvium through which the river flows as well as the river gradient. If the alluvium is fairly homogeneous, then regular meanders may develop as illustrated in Figure 2.1(b) (Leopold, *et al.* 1964). Otherwise the meander pattern may be irregular with discontinuous loops (Brice, 1964), as the river meanders through alluvium of varying degrees of erodibility. This pattern is illustrated in Figure 2.1 (c). In cases where the river valley allows a high degree of channel migration, tortuous meanders may develop (Ritter, 1978). This feature is easily distinguished by the elongated and distorted meander loops which tend to cross the valley at angles in excess of 90° (Kellerhalls, Church, and Bray

1976) as illustrated in Figure 2.1 (d). This means that during floods that exceed bankful, the channel flow would have a velocity component against that of the floodplain flow.

Old rivers, particularly those with tortuous meander pattern as characterized by a high degree of lateral shifting and the floodplains are usually scarred with oxbow lakes resulting from the cutoff of tight meander loops (Chorley *et al.* 1984) usually during major floods. These scars form depressions in the floodplain but do not contribute to effective flow area (because they are no longer connected with the channel). However, they may enhance floodplain storage, if they are not already full of water prior to overbank flooding.

Other river types, identified by the planform pattern include braided (Brice, 1964), anabranching (Brice, 1964) and anastomosing rivers (Morisawa, 1985). These river types are not considered here because they involve multi-channel networks.

Another important consideration in modelling flood flows when the floodplain is inundated arises because of the different roughness associated with the channel and the floodplains, as well as the different flow depth (the floodplain flow being generally shallower). Channel roughness values depend upon the type of bed material as well as any bedforms that might be present. For rivers with sand beds, the Mannings roughness, n, of the channel generally ranges from 0.025 to 0.033 (Chow, 1959) and may be affected by changes in flow stage as a result of changing bedforms (Simons, Li and Associates, 1982). When the bed material consists of gravel and or boulders, the channel n is generally higher ranging from 0.03 to 0.050 (Chow, 1959). Vegetation along the channel banks may reduce the capacity of the river flow. However, this effect is generally marginal except in rivers of low aspect ratio (Simons *et al.* 1982). Floodplains generally have higher n values because they are usually vegetated with pasture, crops, bushes and trees, or contain structures associated with floodplain development. The n values can range from 0.03 for a floodplain with pasture to 0.12 for floodplains containing heavy bush (Chow, 1959).

Rougher floodplain cover and shallower floodplain depths mean that the relative roughness in the floodplain tends to be considerably higher than in the main channel. Consequently, velocities in the floodplain are typically much smaller than in the channel (Chow, 1959). The interaction between the slower floodplain flow and the faster channel flow results in momentum transfer between the channel and floodplains and an associated decrease in overall conveyance capacity. This interaction and momentum exchange leads to a reduction in discharge in the main channel and an increase of discharge in the floodplain (Sellin, 1964; Prinos and Townsend, 1984) compared to the case where for the same stage there is no flow interaction.

Floodplain development and flood control structures, such as dikes (or levees), and flood walls form an integral part of many river channels (Williams, 1994). These flood control structures protect development in the floodplain by confining flood flows to the channel area. They present an additional complexity because of the need to assess the effects of the structures themselves on the flood peak (given that overbank storage and associated attenuation of the flood peak have been eliminated). Also, as was seen during the Mississippi River flood of 1993, dynamic models capable of assessing the effects of structure failures are also needed (Williams, 1994).

2.3 Steady Flow in Channels with Floodplains

2.3.1 Introduction

The majority of the research in this area has focused on experimental studies and steady flow situations beginning with Sellin (1964) and continuing through to the present. Table 2.1 provides a summary of the various types of compound channels used by different researchers in their laboratory investigations where, as illustrated in Figure 2.2: H is the flow depth in the main channel; B is width of the channel bottom, H_f the depth of flow on the floodplain; B_f the width of the floodplain bottom; and T_w the total top width. The table shows that most of the experimental investigations were done on straight rectangular compound channels, with only a few done on trapezoidal and meandering channels. In Table 2.1, it is shown that most of the flows in both the main channel and floodplain had Reynolds numbers of about 10^4. This means that these flows were not fully turbulent. For those who used meandering channels, the sinuosity ranged from 1.2 to 3 which reflects typical meandering rivers (Kiely, 1989). Table 2.1 also shows that some of the compound channels had two symmetrical floodplains and others had only one floodplain. It also shows that the Wallingford research facility (1992) used the largest compound channels in which channel aspect ratios *(B/H)* ranged between 4.9 to 9.4 and relative depths *(H/H_f)* ranged from 2 to 20. It also has large floodplains with B_f/H_f being about 6.7. All researchers used physical models with low aspect ratios compared to real world rivers. For example, a river the like North Saskatchewan at Edmonton has an aspect ratio of about 100 for a two year flood (Kellerhalls, Neil and Bray, 1972).

2.3.2 Flow Characteristics in Straight Channels With Inundated Floodplains

2.3.2.1 Velocity Distribution

The velocity distributions in channels with floodplains have been investigated by Sellin (1964), Rajaratnam and Ahmadi (1979, 81), Ervine and Baird (1982), Prinos, Townsend and Tavoularies (1985), Wormleaton and Hadjipanos (1985), Kawahara (1985), Myers (1987), McKeogh, Kiely and Javan (1989), and Murota, Fukuhara and Seta (1990). A summary of their findings is presented below.

Sellin (1964) presented velocities in the form of isovel patterns, documenting that the region of highest velocity in the compound channel flow was skewed from the center of the main channel towards one of the banks. He explained this as being caused by the momentum transfer between the main channel and the floodplain. Rajaratnam and Ahmadi (1979, 81) reported on the behavior of the depth averaged velocity, finding that it decreased towards the banks from a maximum value at the centerline; increased slightly at the beginning of the floodplain; and then reduced to some undisturbed value towards the banks of the floodplain, as illustrated in Figure 2.3. Similar observations were noted by Prinos *et al.* (1985) and McKeogh *et al.* (1989). At low overbank depths, the depth averaged velocity profiles showed a steep velocity gradient across the interface between the main channel and the floodplain (Kiely, 1989). This velocity gradient reduced as the depth on the floodplain increased (Kiely, 1989). It is generally accepted that the effect of interaction is to reduce the velocities in the main channel and increase the velocities in the floodplain, in comparison with non-interacting conditions.

2.3.2.2 Shear Stress Distribution

Investigations of boundary shear stress in straight compound channels have been undertaken by several authors. These include Ghosh and Jena (1971), Myers and Elsawy (1975), and Rajaratnam and Ahmadi (1979, 81), Knight and Hamed (9[illegible]), Nalluri and Judi (1985), Holden and James (1989).

Using a smooth and artificially roughened rectangular compound straight channel, Ghosh and Jena (1971) showed that shear stress distribution is distinctly non-uniform in character. A typical boundary shear stress distribution, as reported by Rajaratnam and Ahmadi (1979, 81), is illustrated in Figure 2.4. For smooth surfaces, the maximum side shear stress was found to be located at some distance from the free surface of the main channel and the maximum bed shear stress shifted from the center towards the corner of the main channel bed. On the floodplain portion, the maximum bed shear stress occurred near the junction of the floodplain and the main channel.

Myers and Elsawy (1975) made a similar observation. They also found that a decrease of about 22% in bed shear occurred in the main channel while an increase of up to 260% in floodplain bed shear occurred at low overbank flows as compared to the case when there was no flow interaction. Rajaratnam and Ahmadi (1979, 81) also found that the flow interaction between the main channel and the floodplain increased bed shear stresses in the floodplain and reduced bed shear stresses in the main channel.

Holden and James (1989) further investigated lateral shear stress profiles in both rectangular and trapezoidal compound cross-sections. They observed that as the flow depth increased, the maximum shear stress for the floodplain shifted from near the junction to the floodplain a short distance away. They also found that the shape of the bank slope affected the interaction between the main channel and the floodplain flows. They observed that, the intensity of interaction for a given flow decreased slightly as the slope became milder and increased as the slope became steeper.

2.3.2.3 Apparent Shear Stress

This type of shear stress acts along the junction of the main channel and the floodplain during compound channel flow. Rajaratnam and Ahmadi (1981) called this type of shear stress "turbulent mean shear stress". Investigations of apparent shear stress has been documented by Cruff (1965), Myers (1978), Wormleaton, Allen and Hadjipanos (1982), Knight, Demetriou and Hamed (1983) and Prinos and Townsend (1984). The apparent shear stress causes an apparent shear force that opposes the flow motion in the main channel and assists floodplain flow. Myers (1978) defined this apparent shear force as the force due to the momentum transfer from the main channel to the floodplain. It is a measure of the net effect of viscous shear and turbulence together with the action of the vortices transferring momentum from the main channel to the floodplain(s).

Other researchers, such as Kawahara and Tamai (1989) used the concept of apparent shear stress to explain the significance of secondary currents in momentum transfer. Quantifying apparent shear stress is the main problem in understanding floodplain and main channel interaction. At present no one method is widely accepted to quantify the apparent shear stress.

2.3.2.4 Flow Interaction Mechanisms

The typical energy loss mechanisms in open channel flow are bed and wall friction. However, in compound channel flow, the interaction of slow moving flow in the floodplain and the faster main channel flow introduces an interactive flow mechanism. Sellin (1964) established that there are vortices rotating about vertical axes at the interface of a main channel and the floodplain during compound flow. Imamoto and Ishigaki (1989), and Tominaga, Nezu and Ezaki (1989) further observed that apart from the vortices identified by Sellin, vortices with longitudinal axes also existed. These vortices generally assisted in the transfer of momentum from the fast moving main channel flow into the slower moving floodplain flow. Both the floodplain and the main channel zones were affected by the turbulence mixing which resulted from the interaction of flow. The extent of the sub-regions affected, was found to depend on the channel aspect ratio. Rajaratnam and Ahmadi (1979) found that the width of this mixing layer, or interacting zone, was approximately six times the bank height, with most of it apportioned to the main channel.

The interactive flow mechanisms in straight, compound channels have been investigated by Kawahara and Tamai (1989), Tominaga (1989), Imamoto and Ishigaki (1989) and Kiely (1989). They all established that secondary currents contribute to the turbulent shear stress (apparent shear stress) at the interface of the main channel and the floodplain. Kawahara and Tamai (1989) suggested that momentum transfer was made up of two components; advection by secondary flow and turbulent diffusion due to the velocity gradient. They defined momentum transfer in the following way

$$M_{tr} = \int_0^{H_f} \rho u w \, dy - \int_0^{H_f} \rho\left(\overline{-u'w'}\right) dy = \tau_a H_f \qquad [2.2]$$

where:

M_{tr} = momentum transfer;

H_f = floodplain depth;

u = longitudinal velocity;

w = lateral velocity;

y = vertical ordinate;

$\rho\left(\overline{-u'w'}\right)$ = turbulent shear stress; and

τ_a = apparent shear stress.

Kawahara and Tamai (1989) found that the magnitudes of the secondary currents were in the order of 2% - 4% of the longitudinal velocity. At high flows, advection was found to dominate over turbulent diffusion throughout the entire depth of the floodplain. Their experiments showed that at relatively low depths, diffusion action transported about one half of the total momentum into the floodplain. This was because the lateral difference of the longitudinal velocity was large. Kawahara and Tamai (1989) also found that when the water depth increased, the advection component of the secondary flow decreased and the apparent shear stress diminished. The total momentum transfer also decreased. Kawahara and Tamai further observed that when the floodplain roughness was increased (relative to the channel), the turbulent diffusion component of secondary flow increased, resulting in an increase in apparent shear stress was the result. Kiely (1989) found that there were higher turbulence values on the floodplain bed than on the main channel bed. He suggested that these high turbulence values contributed to the retardation of velocity in the interaction region of the main channel.

2.3.2.5 Stage-Discharge Relationship

The need to know the stage-discharge relationships for rivers and canals in all situations is very important. Sellin (1964) and Zheleznyakov (1971) were the first to identify the anomaly in the stage-discharge relationship as the flow just exceeds bank level. Sellin (1964) found that for low overbank depths, the discharge reduced below that of bankful depths. He also noted that, as the floodplain depth continued to increase the discharge again began to increase. Smith (1978) and Wormleaton *et al.* (1982) noted the same observation as Sellin. Bhowmik and Demissie (1982) analyzed field data on floods for several streams in the United States and also observed a similar trend.

2.3.3 Flow Characteristics in Meandering Channels with Inundated Floodplains

2.3.3.1 Velocity Distribution

Investigations of velocity distributions in meandering compound channels have been limited, with most of the work done in smooth compound channels. The few researchers who have tried to study the velocity distribution in a meandering compound channel are Toebes and Sooky (1967), James and Brown (1977), Smith (1978), Ahmadi (1979), McKeogh *et al.* (1989) and Kiely (1989). Figure 2.5 shows the notation used in reviewing meandering compound channels.

Toebes and Sooky (1967) carried out investigations on a meandering compound channel with floodplain depths, H/H_f = 1.2 to 1.5. They presented details of isovel patterns, velocity vector distributions and secondary current patterns. Toebes and Sooky found that high velocity values occurred close to the inner bends of the main channel which differs from those observed in real world rivers. They explained this anomaly as being caused by geometric dissimilarities between the model and real rivers. The cross-section shape in the bend of a real river is nearly triangular, rather than rectangular, due to sediment deposition on the inside bend. This promotes an increase in velocity at the outside banks and a velocity decrease at the inside bank. By observing the directions of the velocity vectors, Toebes and Sooky showed that the dominant direction of floodplain flow was in the streamwise direction although flow exchanges occurred between the meandering channel and the floodplain.

James and Brown (1977) carried out investigations on a meandering compound channel with floodplain depths, H/H_f = 5.0 to 11.0. They reported that the depth averaged velocity profile changed dramatically throughout the meandering compound channel. The velocities were found to accelerate in diverging floodplain areas and to decelerate in converging floodplain areas. A study of surface currents by Toebes and Sooky (1967) indicated that flow was exchanged between floodplain and the meandering channel. At the cross-over section of the meandering channel, the highest velocities were on the diverging section of the floodplain. Rajaratnam and Ahmadi (1979) in a study of compound channel flow with a meandering channel for floodplain depths, H/H_f = 3.1 to 3.5, established that the main channel was not exclusively the location of high velocities at all sections. From contour plots of velocities, they showed that the highest velocities were located on the floodplain adjacent to the inside bend of the meandering channel.

McKeogh and Kiely, (1989) in their investigation of a meandering compound channel with floodplain depths, H/H_f = 2.0 to 2.3 also observed that the maximum velocity was on the floodplain adjacent to the inside bend. They found that the longitudinal velocities in the main channel were typically lower than those on the floodplains. The low velocities in the meandering channel are the result of flow expansion and contraction, horizontal shearing and the development of secondary currents taking place in the meandering cross-over sections. The longitudinal velocities on the floodplain at the outer bend were much lower than their opposite floodplain velocities. McKeogh and Kiely described the distribution of velocities at the cross-over section as very complex with contraction and expansion behaviors near the meander floodplain junctions. Their findings on the velocities vectors distributions were similar to those of Toebes and Sooky (1967).

2.3.3.2 Shear Stress Distribution

The number of studies on shear stress distribution in meandering channels is very limited. One such study is that of Ghosh and Kar (1975), who evaluated the boundary shear stress in a smooth meandering channel for floodplain depths, H/H_f = 1.4 to 2.3. Using velocity isovels, they constructed boundary shear distributions at two distinct sections of a meandering compound channel: the bend; and the cross-over section, as illustrated in Figure 2.6. Their results, when compared to those of straight compound sections, had a few notable features. For the flow in the bend, the side shear distribution was asymmetric and the maximum bed shear distribution in the meandering channel was skewed to the inside of the bend. The maximum floodplain bed shear was found on the floodplain away from the outside bend junction. For the cross-over section, the floodplain bed shear stresses values were generally higher than those of the main channel. Ghosh and Kar explained this behavior of the shear stress distribution as being caused by large scale effects of secondary circulation on the flow.

2.3.3.3 Flow Interaction Mechanisms

The flow interaction mechanism in meandering compound channels has been investigated by Kiely (1989). Kiely suggested that in addition to the interaction flow mechanisms found in straight compound channels, horizontal shearing; and flow expansion and contraction are unique to the meandering flow structure. These flow interaction mechanisms make meandering compound channel flows highly complex and much more difficult to analyze. Figure 2.7 show a representation of these interaction flow mechanisms as observed by Ervine, Willets, Sellin and Lorena (1993). It is noted that the vortex due to the bend has oposite rotation to the vortex caused by flow separation over the bank.

For in-bank meandering flow, secondary currents are driven by the imbalance between the centrifugal force and the transverse pressure force generated by super-elevation of the water surface. For overbank meandering flow, the overbank secondary currents are caused by the intense shear layer across the interface of the outer bend Kiely, (1989). The strength of secondary currents in overbank flow is much stronger than the secondary flow mechanism of in-bank flow (Imamoto and Ishikaki, 1989).

Horizontal shearing occurs when the flows in the main channel, both below and above bank level are impinged upon by the floodplain flow causing a horizontal shear layer between the upper and lower parts of the main channel. Flow expansion and contraction occur at the cross-over between two bends. Kiely (1989) observed that when the flow from one side of the floodplain impinged onto deeper main channel flow, it led to flow expansion. When the flow crossed over, the flow over the main channel encountered an abrupt rise on re-entering the other side of the floodplain, which caused low contraction.

2.3.3.4 Stage-Discharge Relationship

The conveyance capability of the main channel and floodplain in compound meandering channels is greatly affected by channel sinuosity, the size of the meander belt, and the channel aspect and depth ratios. The stage-discharge relationship in meandering compound channels has been researched by the U.S Army Corps of Engineers (1956), Toebes and Sooky (1967), Smith (1978) and Kiely (1989). Toebes and Sooky (1967) found that, for the meandering channel with floodplains described in Table 2.1, the main channel and the floodplain conveyed about roughly the same amount of flow. The in-bank main channel conveyed about 38% of the flow, the main channel above bank level conveyed about 13% while the wide floodplains conveyed about 33% and the narrow floodplain conveyed about 15% of the total flow. Kiely (1989) also observed that for the meandering compound channel stated in Table 2.1, the floodplain and main channel conveyed about the same amount of flow.

Smith (1978) noted in a study of meandering compound channel flow that, as the stage increased above bankful depth, the net percentage of total flow carried by the main channel decreased. He also showed that once the floodplain flow was deep, it dominated the flow in the meandering channel. Smith (1978) also found that greater stages would occur if the meandering main channel was absent. The U.S. Army Corps of Engineers

(1956) established that increased floodplain roughness, as compared to the channel, reduced discharges in both straight and meandering channels. They also showed that channel discharges reduced by about 8% - 10% when sinuosity increased from 1.2 to 1.4.

2.3.4 Calculation Techniques for Compound Channels

2.3.4.1 Introduction

When solving for flow in natural channels with inundated floodplains, the flow section is generally treated either as a composite channel or a compound channel. A composite channel consists of a flow section where the Manning (or Chezy) roughness coefficient can reasonably be considered constant across the cross-section (Chow, 1959). A situation where this would be justified occurs when the flow in the floodplain or valley is significantly larger than the flow in the river channel of the flow section, as illustrated in Figure 2.8 (a). However, when the main channel conveys a significant portion of the total flow, as illustrated in Figure 2.8 (b), then the section is better represented by a compound channel. In this case, the channel is divided into the left floodplain, the main channel, and the right floodplain (Chow, 1959) with the conveyance characteristics of each subsection considered separately. These two methods are further examined in the following sections.

2.3.4.2 Composite Channel Method

This method treats the whole compound channel as a single unit. The roughness across the flow section is represented by the roughness of the floodplain (Chow, 1959). For uniform flow, the Mannings equation (or the Chezy equation) can be used to compute the discharge. If the Mannings equation is used, then the discharge is calculated as follows:

$$Q_t = \frac{1}{n} A_t R_t^{2/3} S_f^{1/2} \quad \text{(S.I version)} \qquad [2.3]$$

where:

A_t = total flow area of channel and floodplains;

n = Mannings coefficient for the floodplain or valley;

Q_t = total discharge;

R_t = hydraulic radius of the whole section (A_t/P_t); and

P_t = wetted perimeter of the whole section;

S_f = longitudinal friction slope.

This method works well as long as the flow depth in the main channel remains relatively small compared to the flow in the floodplain or valley, and if the floodplains have comparable roughness. However once the flow in the main channel becomes significant, the method starts to underestimate the discharge. This is because the use of the floodplain roughness to represent the main channel underestimates the conveyance in the main channel.

2.4.4.3 Compound Channel Method

In cases where the channel and the floodplain carry comparable proportions of the flow, a compound section comprised of a main channel and floodplain zones may be considered (Wormleaton *et al.* 1982; and Prinos and Townsend, 1984). This method does not use the roughness of the floodplain as the roughness for the whole channel but rather an equivalent roughness, n_e , based on the main channel roughness and floodplain roughness. This equivalent roughness is then used in equation [2.3] to calculate the discharge in the compound channel. Chow (1959) suggested three methods of calculating an equivalent roughness, each based on different assumptions. These methods together with the assumptions made in their derivation, are shown in Table 2.2, where: n is Mannings roughness; n_e the equivalent roughness; k equal to 1 represents the left floodplain, equal to 2 represents the main channel, equal to 3 represents the right floodplain; n_k the Mannings roughness n for kth sub-section; P_k the wetted perimeter for

k^{th} sub-section; P the compound channel wetted perimeter; S_f the main channel friction slope; S_o, S_c the main channel bed slope; S_l the left floodplain bed slope; S_r the right floodplain bed slope; R the compound channel hydraulic radius; R_c the main channel hydraulic radius; R_l the left floodplain hydraulic radius; R_r the right floodplain hydraulic radius; Q_t the total compound section discharge; Q_c the discharge in the main channel sub-section; Q_l the discharge in the left floodplain sub-section; Q_r the discharge in the right floodplain sub-section; V_c the average main channel sub-section velocity; V_l the average left floodplain sub-section velocity; V_r the average right floodplain sub-section velocity; and V the average compound channel velocity. Some authors refer to these as variations of 'single channel methods' (Prinos and Townsend, 1984).

Another method, which has been referred to as the 'divided channel method' (Smith, 1978; Prinos and Townsend, 1984), also considers separate roughness values in the main channel and floodplains. The Manning formula is applied separately to the main channel and the floodplains and the resulting discharges are summed to obtain the total compound discharge. In fact, it can be shown that the 'single channel method' (3) is, in fact, equivalent to this 'divided channel method'. Since all of these methods assume the division of the compound channel into main channel and floodplain zones, in this study, they will be referred to as 'divided channel methods'.

As shown in Table 2.2, divided channel methods (1) and (2) generally underestimate the discharge, while divided channel method (3) generally overestimates the discharge. The magnitude of the overestimation or underestimation of discharge is directly proportional to the compound channel roughness. The discharge estimation errors shown in Table 2.2 for the three methods were established from experimental data observed by Prinos and Townsend (1984). It is stressed that the assumptions under which these methods are derived are not entirely correct. The assumption of equal bed slopes for the three channel sections is not always correct because the floodplains may have larger slopes than the main channel (because of channel sinuosity). The assumption of the average floodplain velocity being equal to the main channel velocity in divided channel methods (1) and (2) is also not reasonable. This is because as stated earlier, the floodplains tend to have a higher relative roughness than the main channel and, therefore, a lower average flow velocity. As the least unrealistic of the three divided channel methods, (3) has gained almost universal acceptance. For example, both the HEC-2 (U.S. Army Corps of Engineers, 1982) and DAMBRK (Fread, 1988) computational models use divided channel method (3) to compute the stage-discharge relationship in rivers with floodplains.

2.3.4.4 Dividing the Compound Channel

Dividing the compound channel into three distinct zones takes care of variations in roughness between the main channel and the floodplain zones but neglects the phenomenon which generates an apparent shear at the interface. Rajaratnam and Ahmadi (1979) and Wormleaton *et al.* (1982) suggested various ways of dividing the compound channel to either limit the effect of the apparent shear or to include it. Horizontal division at bankfull depth, vertical division at the edges of the main channel or diagonal (inclined) division can be adopted (Wormleaton *et al.* 1982). Figure 2.9 illustrates these alternatives.

One suggestion has been to include the vertical lines dividing the main channel and its floodplains in the channel wetted perimeter, as a means of approximating the effects of apparent shear from the floodplains resisting the channel flow. It has been found that this approach still overestimates the discharge at low floodplain depths (Prinos and Townsend, 1984), though not as significantly as when this additional wetted perimeter is excluded from the calculation. However, Wormleaton *et al.* (1982) demonstrated that the inclusion of the vertical boundary interface in the calculation of the wetted perimeter for the main channel did not significantly improve results.

Rajaratnam and Ahmadi (1979) suggested that the effect of the apparent shear could be avoided by adopting various channel divisions along shear free boundaries. They suggested that depending on the ratio of the total depth *(H)* to the depth of flow in the floodplain *(Hf)*, horizontal division or vertical division may be appropriate. When this ratio H/H_f is much larger than unity, then the introduction of shear-free vertical boundaries as extensions of the banks of the main channel are more appropriate. When H/H_f is only slightly larger than unity, a horizontal plane as an extension of the floodplain bed is a better method. The use of vertical divisions when H/H_f is much larger than unity is considered appropriate because the difference in velocities in the floodplain and the main channel will probably be great while for H/H_f slightly greater than unity, the depths in both sections are comparable and the difference in velocities between the main channel and floodplain will generally be small. However Wormleaton *et al.* (1982) found that the horizontal method tends to underestimate the discharge at higher flow depths.

Wormleaton and Merret (1990) further investigated the zero-shear interface approach and suggested a diagonal interface joining the banks of the main channel and the top of flow at the center line of the main channel. However for wide channels, diagonal division may not be reasonable because the adopted line of zero shear may not approximate the actual value in the field. Their results showed that this method performed better than the previous two methods when used to determine the total discharge. However, Wormleaton and Merret (1990) found that none of the three methods performed well in predicting individual main channel and floodplain components of flow. The performance of the vertical and diagonal methods improved when the floodplains were narrow while the horizontal method was more accurate when the floodplains were wider. In general, the vertical method was found to give the highest discharge and the horizontal method the lowest discharge.

Because of the inadequacies of the traditional methods for discharge calculation of compound channel flow, many researchers have agreed that the flow interaction and momentum transfer occurring between the main channel and the floodplain need to be taken into consideration. The momentum transfer mechanism generated in the region of high shear flow at the interaction region has the effect of reducing local and mean velocities, boundary shear stress and discharge in the main channel while increasing those properties in the floodplain zone near the junction. The next section reviews the progress that has been made in the alternative approach of incorporating the apparent shear stress into the discharge calculation.

2.3.5 Incorporating Apparent Shear Stress in Discharge Calculation

2.3.5.1 Introduction

Several methods which take into account the shear stress in the interface for the calculation of compound channel discharge have been proposed. Among those who have examined this problem include, Wormleaton *et al.* (1982), Ervine and Baird (1982), Noutsopoulos and Hadjipanos (1983), Knight and Hamed (1984), Prinos and Townsend (1984), Dracos and Hardegger (1987), Wormleaton and Merrett (1990), Stephenson and Kolovopoulos (1991) and Christodoulou (1992). Of the many approaches presented, two seem to have gained the greatest acceptance: the force balance method and the φ-index method.

2.3.5.2 Force Balance Method

Ervine and Baird (1982) and Prinos and Townsend (1984) suggested that, because there is a momentum drain from the main channel into the floodplain, the only way the compound channel flow remains in equilibrium is through the balance of the forces involved. Using a control volume formulation, one can establish the steady uniform flow discharge through a compound channel. Figure 2.10 shows the control volumes that can be considered.

Figure 2.10 shows that the compound channel is divided into three sections: one control volume is the main channel and two control volumes are the floodplain sections. In each control volume, three forces are shown as acting on each sub-section. These forces are the body force, the boundary shear force and the apparent shear force. In the steady state condition, these forces balance within each subsection.

The total discharge *(Q_t)* for a compound channel, when accounting for momentum transfer using this method, is given as:

$$Q_t = A_c V_c + 2 A_f V_f \qquad [2.4]$$

where the main channel velocity can be derived as:

$$V_c = \sqrt{\frac{8}{\rho f_c}\left[\gamma R_c S_o - \frac{2 \tau_a H_f}{P_c}\right]} \qquad [2.5]$$

while the floodplain velocity can be shown to be equal to:

$$V_f = \sqrt{\frac{8}{\rho f_f}\left[\gamma R_f S_{of} + \frac{2 \tau_a H_f}{P_f}\right]} \qquad [2.6]$$

where:

Q_t = total discharge;

A_C = area of main channel section;

V_C = average main channel velocity;

A_f = area of floodplain section;

V_f = average velocity in the floodplain;

f_c = main channel friction factor;

ρ = fluid density;

R_c = hydraulic radius of the main channel;

S_o = main channel bed slope;

τ_a = apparent shear stress;

H_f = floodplain flow depth;

P_c = main channel wetted perimeter;

f_f = floodplain friction factor;

S_{of} = floodplain bed slope;

R_f = hydraulic radius in the floodplain; and

P_f = wetted perimeter in the floodplain.

Although this method accounts for momentum transfer, the apparent shear stress (τ_a) is needed to evaluate the average velocities. A few empirical methods have been developed to evaluate the apparent shear stress. These methods are presented in section 2.4.3.

2.3.5.3 φ-index Method

This method is based on an index that characterizes the degree of interaction between the main channel and floodplain sub-sections. It was first introduced by Radojkovic and Djordevic (1976) and later adopted by Wormleaton and Merrett (1990). The φ-index in the main channel is defined as the ratio of the boundary shear force to the weight component of the fluid in the flow direction. The apparent shear force is taken into consideration through the value of τ_a.

The φ-indices for the main channel and the floodplain are given as follows:

$$\varphi_m = 1 - 2\frac{\tau_a H_f}{\rho g A S_o} \qquad [2.7]$$

$$\varphi_f = 1 + \frac{A}{A_f}(1 - \varphi_m) \qquad [2.8]$$

where:

φ_m = is the main channel index; and

φ_f = is the floodplain index.

The discharges of the subsections and the total discharge Q_t are then computed as follows:

$$Q_c = Q_c^{'}\ \varphi_m^{1/2} \qquad [2.9]$$

$$Q_f = Q_f^{'}\ \varphi_f^{1/2} \qquad [2.10]$$

$$Q_t = Q_c + Q_f \qquad [2.11]$$

where $Q_c^{'}$ and $Q_f^{'}$ are the main channel and the floodplain discharges with no interaction respectively.

$Q_c^{'}$ and $Q_f^{'}$ may be evaluated using the Manning formula. The predicted compound flow using equations [2.9] and [2.10] have been found by Wormleaton and Merrett (1990) and Christodulou (1992) to compare well with various experimental data. As in the force balance method, this method requires the evaluation of the apparent shear stress, τ_a.

2.3.5.4 Determination of Apparent Shear Stress *(τ_a)*

The magnitude of the apparent shear stress on the vertical interface has been indirectly determined in several experimental studies where measurements of boundary shear distribution allowed the solution of equation [2.12]. This equation was derived based on a balance of forces for uniform flow:

$$\tau_a = \frac{\gamma A_c S_o - \tau_c P_c}{2 H_f} \qquad [2.12]$$

This method has been used by Myers (1978), Ervine and Baird (1982) and Prinos and Townsend (1984). Knight and Demetriou (1983) have expressed the apparent shear stress as a percentage of the total boundary shear stress in smooth channels. Rajaratnam and Ahmadi (1981) also proposed an expression for the turbulent mean shear (apparent shear stress) for a compound channel with one floodplain as a function of the undisturbed floodplain boundary shear stress ($\tau_{f\infty}$), found near the outside edge of the floodplain:

$$\tau_a = 0.15\left(\frac{H}{H_f} - 1\right)^2 \tau_{f\infty} \quad \text{valid for} \quad 2.2 \le H/H_f \le 7.4 \qquad [2.13]$$

where:

H = main channel depth;

H_f = depth of floodplain, and

$\tau_{f\infty}$ = undisturbed boundary shear stress in the floodplain.

Wormleaton *et al.* (1982) proposed an expression for symmetrical rectangular channels with smooth or roughened boundaries:

$$\tau_a = 13.84(\Delta V)^{0.882}\left(\frac{H}{H_f}\right)^{3.123}\left(\frac{B}{T_w - B}\right)^{0.727} \quad \text{valid for} \quad 2.3 \le H/H_f \le 9.0 \qquad [2.14]$$

Where:

T_w = total width of the compound channel;

B = bed width of main channel; and

ΔV = velocity difference between the main channel and the floodplain.

When Wormleaton *et al.* (1982) compared results generated from the above equation to those obtained through experimental work, the agreement was quite good.

Prinos and Townsend (1984) proposed a similar equation for symmetrical compound channels with a trapezoidal main channel:

$$\tau_a = 0.874(\Delta V)^{0.92}\left(\frac{H}{H_f}\right)^{1.129}\left(\frac{B}{Tw - B}\right)^{0.514} \quad \text{valid for } 3.0 \le H/H_f \le 11.2 \qquad [2.15]$$

Equation [2.15] also gave reasonable agreement with observed data. Others who have suggested empirical relations include Wormleaton and Merret (1990) and Christodoulou (1992). Using a large scale experimental facility, Wormleaton and Merret (1990) proposed the following expression:

$$\tau_a = 3.325\, \Delta V^{1.451}\left(\frac{1}{H_f}\right)^{0.354} (T_w - B)^{0.519} \quad \text{valid for } 2.0 \le H/H_f \le 20.0 \qquad [2.16]$$

Christodoulou (1992) proposed a rather different expression from the previous researchers:

$$\tau_a = \frac{1}{2}\rho\, C_{fa}\, \Delta V^2 \qquad \text{valid for } 1.9 \le H/H_f \le 9.3 \qquad [2.17]$$

Where C_{fa} is the apparent friction factor at the interface between the main channel and floodplain.

For a symmetrical smooth channel, Christodoulou determined that C_{fa} can be expressed as:

$$C_{fa} = 0.01\frac{T_w}{B} \qquad [2.18]$$

Christodoulou found equation [2.18] to hold for $0 < H_f / H_w < 1$ and $1.7 < T_w / B < 6.7$ and for $H_w / B \approx 0.5$. He is also found that equation [2.18] could not be extended to asymmetrical compound channels. He found that asymmetric channels had a much stronger interaction between the floodplain and the main channel and equation [2.18] underestimated the value of C_{fa} considerably. He concluded that more studies were required to obtained a generalized relation for C_{fa} for non-symmetric shapes.

Equations [2.14], [2,15] and [2.16] suggest that the apparent shear is proportional to $(T_w - B)^{-0.727}$, $(T_w - B)^{-0.514}$ and $(T_w - B)^{0.519}$. This is contradictory. Since equations [2.17] and [2.18] suggest that the apparent shear is proportional to T_w, then equation [2.16] is probably more reasonable. The exponents for the ratio of H/H_f ramges from 0.354 to 3.123 while those for ΔV range from 0.882 to 2. This seem to suggest that these relations of apparent shear stress are influenced by the experimental set up adopted.

2.4 Unsteady Flow in Compound Channels

2.4.1 Introduction

Although a steady flow approximation may be reasonable or valid for a large number of practical situations, an unsteady flow analysis is essential in many situations, for example, when conducting inundation studies or evaluations of the impact of flood mitigation structures on flood waves, particularly when they fail. Although much progress has been made in the study of open channel flow modelling and in particular hydraulic flood routing, considerable limitations remain.

2.4.2 Investigations of Unsteady flow in Compound Channels

Experimental research involving unsteady flow tests in compound channels are almost non-existent. To the author's knowledge, only two such experimental studies have been conducted to date, one by (Treske, 1980) and the other by Rashid and Chaudhry, (1993). The data by Rashid and Chaudhry (1993) could not be obatined for this study, therefore Treske 's data were used. Treske conducted some tests in a 210 m long flume for both straight and meandering channels. His data, limited to details of the outflow hydrographs has been used for verification purposes in this thesis.

Field data is also limited. Although severe floods involving inundated floodplains occur frequently, data collection during such events is normally limited to high water mark surveys and discharge measurements at selected sites. The lack of detailed field data can be attributed to the difficulty and expense associated with the collection of research quality data at a time when resources are normally stretched to the limit by the emergency at hand.

2.4.3 Current Modelling Techniques for Unsteady Flow in Compound Channels

One-dimensional modelling of unsteady compound flow has been investigated by Radojkovic (1976); Fread, (1976, 1988); Ervine and Ellis (1987); Stephenson and Kovopovoulos (1990); and Abida and Townsend (1994). The dynamic wave models, used in dam breach inundation studies, are generally based on the 'divided channel methods' discussed in the last section. These solve a formulation of the St. Venant equations, as developed by St Venant in 1871 for one-dimensional unsteady flow, consisting of continuity and longitudinal momentum equations (Cunge *et al.* 1980). For a channel without lateral inflow or outflow, the continuity equation is given as:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0 \qquad [2.19]$$

Where: Q is the discharge; and A is the cross section flow area. Q is defined as the volume of water passing through a cross section per unit time.

The longitudinal momentum equation is given as:

$$\frac{\partial Q}{\partial t} + \frac{\partial (QV)}{\partial x} + gA\frac{\partial H}{\partial x} = gA(S_o - S_f) \qquad [2.20]$$

where:

g = is the acceleration due to gravity;

H = depth of flow ;

S_o = longitudinal channel bed slope;

S_f = longitudinal friction slope;

t = temporal coordinate;

x = longitudinal distance; and

V = cross-section longitudinal average velocity.

In equations [2.19] and [2.20], the first two terms describe temporal and local acceleration, respectively. They are often referred to as the inertial terms (Ferrick, 1985). The third term represents the net longitudinal pressure force acting on the control volume. The next two terms reflect the effects of gravity and friction in the longitudinal direction, respectively.

Recent researchers (DeLong, 1986; Fread, 1988) have sought to develop adaptations of these equations which would take into account practical and natural factors such as channel sinuosity, ineffective floodplain zones, and lateral inflows of mass or momentum, as evidenced by the formulation solved in the NWS DAMBRK (Fread, 1988) model:

$$\frac{\partial Q}{\partial x} + \frac{\partial s_c(A + A_o)}{\partial t} + q = 0 \qquad [2.21]$$

$$\frac{\partial(s_m Q)}{\partial t} + \frac{\partial(\beta Q^2 / A)}{\partial t} + gA\left(\frac{\partial h}{\partial x} + S_f + S_e\right) + L' = 0 \qquad [2.22]$$

where:

A_o = the inactive flow area (off-channel storage);

s_c, s_m = sinuosity factors which vary with stage (DeLong, 1986) ;

β = the momentum correction coefficient;

S_e = the expansion-contraction slope; and

L' = the momentum contribution of lateral inflows.

Although this model has a provision for lateral inflow and outflow, this only applies to lateral inflows in to and out of the compound channel, and not between the channel and its floodplains. Therefore, despite such enhancements to the basic equations, such models cannot consider situations where the flow in the floodplain is independent of that in the channel; for example when a dike is overtopped and/or breached and the flow does not return to the main channel. In fact, the floodplain inundation can even begin from a downstream low-dike location, such that floodplain waters actually flow in an upstream direction at least for a certain period of time (Cunge *et al.* 1980). Another practical example often occurs at river confluences, where the floodplain flow of the tributary enters the larger river's floodplain. 'Divided channel models' also cannot take into account the convective transport of momentum in to and out of a channel meandering through inundated floodplains.

An early alternative to these dynamic 'divided channel' models, were the pseudo two-dimensional models (Cunge *et al.* 1980) in which floodplains were either treated as storage areas (Yevjevich, 1975) or were divided into a number of cells (or storage basins) which communicated with neighbor cells and or the main channel through selected "hydraulic laws" neglecting inertial forces (Abbott and Cunge, 1975). For example, Zanobetti, Lorgere, Preismann and Cunge, (1970) employed this cell-type modelling in the Mekong floodplain study. These two-dimensional models, though capable of modelling floodplain storage and flow losses entirely independent of the main channel, could not incorporate dynamic terms. Therefore, the equations modelled for the channel had to neglect inertial terms as well (Cunge *et al.* 1980).

Even if momentum transfer due to flow interaction is shown not to have significant impact on unsteady compound flow, modelling of mass exchange between the main channel and floodplain have significant implications in modelling overtopping and breached dikes. The possibility of using one-dimensional models in modelling meandering compound channels could be enhanced if mass flow exchange between the floodplain and main channel were to be included. Although recent models by Stephenson and Kovopovoulos (1990) and Abida and Townsend (1994) include the effect of flow interaction, they have not allowed for the exchange of mass between the main channel and the floodplain.

At present, fully two-dimensional dynamic models of compound channel flow are the subject of much research (Lee and Froelich, 1986). However, the computational overhead for such models is generally quite high in comparison to one-dimensional models. Therefore, these are not yet a practical alternative for flood routing over long distances. In the following chapter a new model formulation is presented that seeks to include the effects of flow interaction with mass flow exchange being allowed between the main channel and floodplain. The procedure adopted in this study in the determination of mass transfer between the main channel and floodplain is discussed below.

2.5 Mass Transfer Between the Main Channel and Floodplain

2.5.1 Introduction

In the model, the channel and floodplain equations must be linked through mass transfer equations which represent the net transfer of water from the channel to the floodplains for rising stage and discharge, and the flow back into the channel from the floodplains for falling stage and discharge. As there is very little data available regarding this mass transfer process, an approximate model of mass transfer must be considered. Two approaches were used in this study, as discussed below.

2.5.2 Flow From the Main Channel to Floodplain

2.5.2.1 Linking Model

The mass conservation link between the inundated floodplain and the main channel is important in channel overflow representation for unsteady flows. When the overflow occurs at defined points along the bank and the length of the overflow section is short relative to the river length, then a weir section could be used to link the flow in the main channel and the floodplain as shown as in Figure 2.11 (Cunge, *et al.* 1980). In this case the weir section in the model would be defined by the crest elevation, width, and discharge coefficient which represent the physical situation as closely as possible (Cunge *et al.* 1980). The discharge calculation is based on the difference in the water level between the main channel and the floodplain.

When there are no well defined overflow sections, because there is a general spilling from the channel to the floodplain along the bank, the modelling problem is more complex. This situation is shown in Figure 2.12(a). Cunge *et al.* (1980) suggest two methods to deal with this problem. If the computational points are closely spaced, then the best approach is that which links the computational points in the main channel to each computational point in the floodplain. Figure 2.12(b) shows this option. The close computational points allow for true representation of flow depth changes along the channel in the longitudinal direction.

If the computational points are far apart, Cunge *et al.* (*1980*) proposes that the best way to link the main channel and the floodplain, is by linking about four computational points in the floodplain with one computational point in the main channel. Figures 2.12 (c) illustrates this approach. Cunge *et al.* (1980) point out that, a very long weir crest can lead to a situation where a small increase in the water surface elevation in the river could provoke a sudden large discharge to the floodplain which may violate the continuity equation. This could possibly lead to computational instabilities or could totally falsify the details of the flood overflows.

In this model, the 'one-to-one' approach, illustrated in Figure 2.12 (t... has been used to connect the floodplain and the main channel. In modelling the 'one-to-one' approach, a weir approximation is used to link the main channel and the inundated floodplain. Possible weir relations connecting the main channel and floodplain are discussed in the next sections.

2.5.2.2 Weir Discharge Estimation

2.5.2.2.1 Modular Flow

When the flow above the crest of a weir is dependent on the upstream depth only, the flow is said to be modular. Modular flow normally occur in perfect weirs as shown in Figure 2.13.

Govinda Rao and Muralidhar, (1963) classified the flow over weirs as a function of (H_l/B_w) as shown in Table 2.3, where: H_l is the head over the weir and B_w is the width of the weir in the longitudinal direction (as shown in Figure 2.13).

The discharge relationship for weirs is usually expressed as:

$$q = C_d \frac{2}{3}\sqrt{2g}H_l^{3/2} \quad \text{(Lakshmana Rao, 1975)} \qquad [2.23]$$

where:

C_d = discharge coefficient;

q = discharge over the weir per unit weir length, L;

L = the length of the weir, equal to the width of approach channel.

The coefficient of discharge is the main parameter used to differentiate weir types. Some widely used coefficients of discharge for sharp crested weirs are summarized in Table 2.4.

To compare the discharge coefficients for broad crested weirs, equation [2.23] is reformulated as:

$$q = C_l H_l^{3/2} \quad \text{(Govinda Rao and Muralidhar, 1963)} \qquad [2.24]$$

where C_l is the coefficient of discharge accounting for the effect of approach velocity. C_l has been determined empirically by many researchers, and a few of these equations are shown in Table 2.5.

When modelling using a weir to link the main channel and floodplain, the floodplain will form part of the weir crest (width of the weir and the bank of the river forms the step or height of the weir *(P)*. The flow in the river above the bank (H_l) is represented by typical depths in the floodplain. Typical parameters for rivers and floodplains in UK rivers (Samuels, 1985) shown in Table 2.6, are used to establish the possible kind of weir to use as a model.

The value of H_l/B_w for typical floodplains is about 0.002 which lies in the very very long crested range. It is also feasible that for small widths and large depths in the floodplain, the value H_l/B_w would mostly lie in the broad crested range. Therefore long and broad crested weirs are possible models to use in estimating lateral discharge if H_l/B_w is the only parameter to consider. However, other factors like submergence and flow angle into the floodplain also have to be considered.

2.5.2.2.2 Effect of Submergence (Non Modular Flow)

Most of the floodplain inundation and flow back into the river will probably be in the non-modular range (submerged), therefore the effects of submergence on the flow over broad or long crested weirs is examined.

When the discharge of a weir depends on both the upstream and downstream heads as shown in Figure 2.14, the flow is said to be in the non-modular range or submerged.

The effect of submergence on flow in a broad crested has been investigated by Smith, (1959); Kandaswamy and Rajaratnam (1959) and Clemmens *et al.* (1984). Smith (1959) investigated the effect of submergence as function of Froude number and upstream and downstream depths and presented the following formula:

$$q = H_1^{3/2}\sqrt{g\left[2.86 - 2.96\frac{H_3}{H_1}\right]\left[0.027 + 0.991\frac{H_3}{H_1}\right]^3} \qquad [2.25]$$

On comparing with equation [2.24], it can be seen that:

$$C_1 = \sqrt{g\left[2.86 - 2.96\frac{H_3}{H_1}\right]\left[0.027 + 0.991\frac{H_3}{H_1}\right]^3} \qquad [2.26]$$

Kandaswamy and Rajaratnam (1959), expressed the non-modular discharge as:

$$q = q_m.f \qquad [2.27]$$

where:

q_m = modular discharge and

f = a reduction factor dependent on H_3/H_1.

They established that f varied from 1.0 to 0.4 for values of H_3/H_1 ranging from 0.4 to 1.0.

Rijn (1990) suggested using the following relation:

$$q = C_d H_3\left[2g\left(H_o - H_3\right)\right]^{1/2} \qquad [2.28]$$

C_d in equation [2.28] accounts for losses due to expansion of flow at the downstream part of the weir. He suggested a C_d of 0.9 for a rough weir with a sharp bottom transition, and 1.3 for a smooth weir with a rounded bottom transition.

Clemmens *et al.* (1984) defined a limiting submergence factor called the modular limit *(ML)*, that divides the non-submerged conditions (modular flow) and the submerged conditions (non-modular flow) on a broad crested weir. This limit is defined as:

$$ML = \frac{H_3}{H_1} \qquad [2.29]$$

For ML less than 0.8, they said the flow is modular (not submerged) and the perfect weir equations should be applied while for *ML* above 0.8, the imperfect weir equation should be used (Clemmens *et al.* 1984). Ramamurthy, Tim and Rao (1988) said this value of *ML* is equal to 0.73 while Hager (1994) set it at 0.75 for a broad crested weir.

2.5.2.2.3 Effect of Flow Angle

The application of a perfect broad or long crested weir in approximating the spilling of flow into the floodplain may overestimate the lateral discharge for the following reasons. The discharge estimation using a weir assumes the flow approach is perpendicular to the weir and yet the spilling of flow from the main channel into the floodplain is expected to be at angle and only part of the flow in the main channel spills into the floodplain. Therefore a side weir flow model should be considered a possible model to estimate the lateral discharge.

A side weir, also known as a lateral weir, is a free over-flow weir set into the side of a channel as shown in Figure 2.15. The weir allows part of the flow to spill over the side when the surface of the flow in the channel rises above the weir crest.

In the basic approach, the flow through a side weir is assumed to be approximately two-dimensional and the pressure in the channel is assumed to approximately hydrostatic despite some curvature and irregularity of the water surface (El-Khasab and Smith, 1976). Although the flow over the side weir crest is at an angle with the direction normal to the weir, a conventional weir equation per unit length is normally used (Subramanya and Awasthy, 1972; Smith, 1973; El-Khasab and Smith, 1976; Kumar and Pathak, 1987; and Cheong, 1991). It is normally written as:

$$q = \frac{2}{3} C_d \sqrt{2g} \left(H - H_w \right)^{3/2} \qquad [2.30]$$

where:

C_d = the side weir discharge coefficient;

H_w = side weir height; and

H = the water surface elevation, which varies in the longitudinal direction as illustrated in Figure 2.15.

In the analysis of side weirs, it is normally assumed that specific energy is constant across the side length (Subramanya and Awasthy, 1972; Smith, 1973; El-Khasab and Smith, 1976; and Cheong, 1991). This means that, the longitudinal component of the velocity of the spill over is equal to the average channel velocity (El-Khasab and Smith, 1976).

Most of the studies done on side weirs have been on sharp crested weirs. The contributions by Ackers (1957), Collinge (1957), Frazer, (1957), Subramanya *et al.* (1972) and El-Khasab and Smith, (1976), Ranga Raju, Prasad, and Gupta (1979), Cheong (1991), and Manivannan and Satyanarayana (1994) are all on side weirs with sharp crested shapes. As in the previous analysis on sharp and broad crested weirs, the value of the weir discharge coefficient has been the major focus of research.

Ackers (1957) suggested that for subcritical flow, C_d is equal to 0.625 if H is measured at a remote distance from the plane of the weir (towards the center of the main channel) and 0.725 if measured at the plane of the side sharp crested weir. For supercritical flow, he found C_d to be about $0.36 - 0.08F_1$, where F_1 is the Froude number at section 1 shown in Figure 2.15. Some contributions on discharge coefficient for the side sharp crested weir are summarized in Table 2.7.

If spilling into the floodplain is assumed to occur over a broad or long crested weir, then the best modelling side weir would be one with a broad crested shape. Side weirs of broad crested shape seemed to have received minimal attention. Ranga Raju *et al.* (1979) extended the results obtained for a side sharp crested weir to obtain the discharge coefficient for a side broad crested weir also shown in Table 2.7.

The effect of flow submergence on side weirs seems also not to have received any attention as no information was found in a literature search.

2.5.2.2.4 Effects of Sloped Upstream Face

The upstream face of a broad crested weir influences the coefficient of discharge C_l (Rao *et al.* 1988, and Bos, 1989). Most of the coefficients shown in Table 2.6 are for a vertical upstream face. Normally the upstream face of the weir is rounded to offer a streamline transition of the flow into the crest of the weir and an increase in the coefficient of discharge. However when the upstream face has a significant slope beyond the small rounding of the nose as shown in Figure 2.16, Arunachalam (1964) suggests that an additional effect of the slope be included in the normal coefficient C_l.

Arunachalam (1964) found that the coefficient of discharge, C_l (where the velocity approach conditions have been lumped into the C_l coefficient) decreased with steeper slope of the upstream face of the weir. This means that for a vertical upstream face, the coefficient of discharge C_l is the lowest while large slopes have larger values of C_l. He suggested that C_l for a weir with any side slope can be given by the following equation:

$$C_l = C_s + \Delta C_u + \Delta C_d \qquad [2.31]$$

where:

C_s = the discharge coefficient for a weir having a vertical upstream face and a sloping downstream face of 1:1;

ΔC_u = effect of change in the upstream slope; and

ΔC_d = effect of change in the downstream slope.

Arunachalam, (1964), established that for flow in the modular limit and $H_l/B_w \leq 2.2$, an approximate relation for determining ΔC_u *is* given as:

$$\Delta C_u = 0.08\,S_u + 0.08 \qquad [2.32]$$

where S_u is the upstream weir face slope.

Equation [2.32] is valid for values of S_u more than 1/3:1. When Su=0, ΔC_u is also equal to zero. Arunachalam, (1964) also found that when $H_1 / Bw < 0.4$ (flow over a broad and long crest weir) and the downstream slope is more than the critical slope, any further increase in the downstream slope shows no effect at all.

2.5.3 Selection of the Discharge Coefficient

To estimate the lateral discharge into the floodplain, several factors are considered. The spilling of flow into the floodplain occurs at an angle and there is no constriction of flow as normally witnessed in a side weir. Therefore while they may be energy losses when the flow goes through a side weir, the spilling into the floodplain is associated with almost no loss of energy.

If H_1/B_w of a weir is the only criterion considered, then estimation of the lateral outflow should be made using long and broad crested weirs. However because of the need to consider the spilling of the flow at an angle, side weirs should also be considered. From the literature search, it became apparent that most of the studies on side weirs have been on side weirs of sharp crested shape. Only one study, that of Ranga Raju (1979) was on side weirs of broad crested shape.

The discharge coefficient for sharp crested weirs depends on the ratio of the weir head to the height of the weir (H_1/P) as shown in Figures 2.17 (a) and 2.17 (b) while C_d for broad crested weirs is influenced by the ratio of the weir head to the width of the weir (in the flow direction), H_1/B_w as shown in Figure 2.18(a). Figures 2.17 (c) and 2.18 (b) show that the Froude number is the major factor influencing discharge coefficients for side weirs.

The discharge coefficients for sharp and broad crested weirs generally show a close agreement as shown in Figures 2.17 (a), 2.17 (b) and 2.18 (a) while the estimation of coefficient of discharge for the side weirs shows a wide disagreement as shown in Figures 2.17 (c) and 2.18 (b). This means that whatever method adopted for estimating the lateral discharge will only give an approximate value. Furthermore, while the effect of submergence of sharp and broad crested has been researched, the effect of submergence on side weirs seems not to have been investigated. Clearly the problem of mass exchange between the floodplain and the main channel is a combination of flow in the modular and non-modular range. This then calls for using side weirs (broad crested shape) equations that account for the angle of spilling and broad crested weirs equations that account for some form of submergence. Alternatively, a range of C_d values could be determined and tested for sensitivity on the estimation of discharge.

Since only one equation developed for side weir of broad crested shape was available and no information on submergence of side crested weirs were found, it was decided that a range of C_l values from 1.45 (which covered the lower range of C_l values) to 1.90 (for high values) be used for estimating lateral outflow into the floodplain in order to assess the sensitivity of the model to this parameter.

2.5.4 Overtopping and Breaching of Dikes

Since part of this study was to consider flow over flood structures like dikes, the flow over a dike or a breaching dike is essentially treated as flow over a side weir with top width using the equations shown in Tables 2.5 and 2.7. If H_1/B_w is less or equal to 0.4 then a broad crested weir could be used to approximate the flow, whereas if it is greater than 0.4 then a side weir of sharp crested shape may be considered as a possible model to determine the flow discharge (Govinda Rao and Muralidhar, 1963).

A dike breach can also be treated as some sort of end-weir of broad crested shape as shown in Figure 2.19. Muralidhar (1964) investigated flow over an end weir with a top width and came up with an end weir discharge coefficient, C_e which he defined as:

$$C_e = C_l \lambda \qquad [2.33]$$

where:

λ = a multiplier accounting for lateral spreading of the nappe.

The variation of λ with H_l/B_w is given as :

$$\lambda = 0.018 log\left(\frac{H_l}{B_l}\right) + 1.048 \quad \text{for long crested weir range} \quad [2.34]$$

$$\lambda = 0.151\left(\frac{H_l}{B_w}\right) + 1.021 \quad \text{for broad crested weir range} \quad [2.35]$$

$$\lambda = 1.084 - 0.014\left(\frac{H_l}{B_w}\right) \quad \text{for the narrow crested weir range} \quad [2.36]$$

The use of λ was found to increase the discharge coefficient C_l by up to 8%.

The use of the end-weir discharge coefficient does not take into effect the case where the flow through a breached dike is at an angle. Therefore for this study, the equation developed for side weirs for a broad crested shape was considerd more appropriate.

2.5.5 Flow From Floodplain to Channel

The flow from the floodplain into the river happens during flood recession or subsidence. Figure 2.20 shows an illustration of flow from the floodplain into the river.

To estimate the lateral discharge into the river, a side weir with zero height ($H_w = 0$) is probably the best choice, as there is no step but a fall. Although an end depth weir could be a possible model, it is not considered because only part of the flow spills into the river at an angle, while the rest continue in the downstream direction. A side weir of the sharp crested shape is the ideal choice because the water spills over an edge. Therefore the side weir equations for side weirs of zero height as tabulated in Table 2.7 should be applicable as models to estimate the flow. Although the Froude number affects the value of C_d for side weirs of sharp crested shape, most of the coefficients lie in the range 0.45 to 0.65 as shown in Figure 2.17 (c) for the range of Froude numbers of 0 to about 0.3. This range of Froude numbers covers most of the expected flow in the floodplain.

Therefore for lateral inflow from the floodplain to the main channel, it was deemed reasonable to assume C_d values of 0.45 to 0.65 with sensitivity tests being carried out.

2.5.6 A New Method For Estimating the Lateral Flow Into The Floodplain

2.5.6.1 Flow into Inunudated Floodplain

A new method developed by Shome (1995) to estimate the lateral discharge to the floodplain was also considered. This method involves setting up a control volume extending from the edge of the river to the edge of the floodplain width, as shown in Figure 2.21, and applying conservation of lateral momentum.
where: H_f is the floodplain depth at any place along the floodplain width; H_3 the floodplain edge depth; y any arbitrary distance in the lateral direction; and B_f the total floodplain width.

Applying conservation of lateral momentum yields:

$$P_l - P_3 + M_l = F_\tau \qquad [2.37]$$

where:

P_l	= the pressure force at the bank interface;
P_3	= the pressure force at the end of the control volume in the floodplain;
M_l	= lateral momentum into the floodplain; and
F_τ	= the boundary shear force on the floodplain.

If it is assumed that the lateral momentum into floodplain is zero, equation [2.37] reduces to:

$$P_l - P_3 = F_\tau \qquad [2.38]$$

If further it is assumed that the water surface varies linearly across the floodplain. This means that the floodplain depth can be expressed in terms of H_3 and H_1 as:

$$H_f = \frac{H_1 + H_3}{2}$$ [2.39]

expressing:

$$P_1 = \frac{\gamma H_1^2}{2}$$ [2.40]

and

$$P_3 = \frac{\gamma H_3^2}{2}$$ [2.41]

then

$$F_p = P_1 - P_3 = \frac{\gamma}{2}\left(H_1^2 - H_3^2\right)$$ [2.42]

The boundary shear force in the floodplain is defined as:

$$F_\tau = \int_0^{B_f} \tau_f \, dy$$ [2.43]

where:

τ_f = boudary shear stress. It is defined as:.

$$\tau_f = \frac{\rho V W}{C_*^2}$$ [2.44]

where:

V =the longitudinal velocity;

W = lateral velocity; and

C_* = Chezy's nondimensional coefficient.

The lateral velocity W is assumed to vary linearly as:

$$W = W_o\left(1 - \frac{y}{B_f}\right) = \frac{q_o}{H_f}\left(1 - \frac{y}{B_f}\right)$$ [2.45]

and V can be defined using Chezy's equation as:

$$V = C_{\bullet}\sqrt{gRS_{of}} \approx C_{\bullet}\sqrt{gH_f S_{of}} \quad \text{for wide channels} \tag{2.46}$$

where:

S_{of} = longitudinal friction slope in the floodplain; and

R =hydraulic radius of the floodplain wetted section.

Then equation [2.44] becomes:

$$\tau_f = \frac{\rho\sqrt{gS_{of}}}{C_{\bullet}} \frac{q_o}{\sqrt{H_f}} \left(1 - \frac{y}{B_f} \right) \tag{2.47}$$

Defining

$$\chi = \frac{\rho\sqrt{gS_{of}}}{C_{\bullet}} \tag{2.48}$$

Then equation [2.43] becomes:

$$F_\tau = \int_0^{B_f} \frac{\chi q_o}{\sqrt{H_f}} \left(1 - \frac{y}{B_f} \right) dy \tag{2.49}$$

and integration yields:

$$F_\tau = \frac{\chi q_o}{\sqrt{H_f}} \left[y - \frac{y^2}{2B_f} \right]_0^{B_f} = \frac{\chi q_o}{\sqrt{H_f}} \frac{B_f}{2} \tag{2.50}$$

Equating equation [2.50] and [2.42], yields (after some rearrangement):

$$I_c = \frac{4C_{\bullet}\sqrt{g}}{B_c\sqrt{S_{of}}} H_f^{3/2} \left(H_1 - H_f \right) \tag{2.51}$$

2.5.6.2 Flow into a Partially Inundated Floodplain

When the floodplain is partially dry, the control volume extends to the floodplain as shown in Figure ? 22, where: B_e is effective flow width in the floodplain.

A force b... e in the control volume extending to the wetted surface in the floodplain is set up as:

$$P_1 = F_\tau \qquad [2.52]$$

It is noted that there is only one pressure force (P_1) that balances the boundary shear force (F_τ).

P_1 is given by equation [2.40] ... estated below for easy reference.

$$P_1 = \frac{\gamma H_1^2}{2} \qquad [2.40]$$

Boundary shear force is defined as:

$$F_\tau = \int_0^{B_e} \tau \, dy \qquad [2.53]$$

where B_e is given by the following relation:

$$B_e = 2\frac{H_f}{H_1} B_f \qquad [2.54]$$

The floodplain boundary shear stress (τ_f) is given by equation [2.44] and the floodplain longitudinal velocity (V) is defined by equation [2.46]. These equations are restated below for easy reference.

$$\tau_f = \frac{\rho V W}{C_*^2} \qquad [2.44]$$

$$V = C_* \sqrt{gRS_{of}} \approx C_* \sqrt{ghS_{of}} \quad \text{for wide channels} \qquad [2.46]$$

The lateral velocity *(W)* is defined for this control volume as:

$$W = \frac{qy}{h} \qquad [2.55]$$

where, qy is expressed as:

$$qy \cong q_o\left(1 - \frac{y}{B_e}\right) \qquad [2.56]$$

Therefore W becomes:

$$W \cong \frac{q_o}{h}\left(1 - \frac{y}{B_e}\right) \qquad [2.57]$$

where h is assumed to be defined by the relation:

$$h = H_I\left(1 - \frac{y}{B_e}\right) \qquad [2.58]$$

Substituting equations [2.46]; [2.57] ; and [2.58] into equation [2.44]: the floodplain boundary shear stress τ_f takes the form of:

$$\tau_f = \frac{\rho\sqrt{gS_{of}}}{C_*\sqrt{H_I}} q_o\left(1 - \frac{y}{B_e}\right)^{1/2} \qquad [2.59]$$

and the floodplain boundary shear force becomes:

$$F_\tau = \int_0^{B_e} \tau_f \, dy = \int_0^{B_e} \frac{\rho\sqrt{gS_{of}}q_o}{C_*\sqrt{H_I}}\left(1 - \frac{y}{B_e}\right)^{1/2} dy \qquad [2.60]$$

Integrating the above equation yields F_τ as:

$$F_{\tau} = \frac{2}{3}\frac{\rho\sqrt{gS_{of}}q_{o}B_{c}}{C.\sqrt{H_{l}}} \qquad [2.61]$$

and equating $P_l = F_\tau$ yields q_o as:

$$q_{o} = \frac{3}{8}\frac{\sqrt{g}C.}{B_{f}\sqrt{S_{of}}H_{f}}H_{l}^{7/2} \qquad [2.62]$$

Equation [2.51] and [2.62] were then used to estimate lateral flow into the floodplain and compared with the other conventional weir equations shown earlier.

2.6 Momentum Transfer Between the Main Channel and Floodplain Flow

When the flow spills into or out of the floodplain, the lateral convective momentum $V_x q$, is conveyed into and out of the floodplain through the lateral discharge q.

For flow from the main channel to the main floodplain, V_x would be defined as:

$$V_x = V \qquad [2.63]$$

where, V is the average channel velocity and for flow from the floodplain to the main channel, it is given as:

$$V_x = V_f \qquad [2.64]$$

where V_f is the average velocity in either floodplain.

The flow interaction momentum transfer between the main channel and the floodplain is accomplished through the apparent shear stresses. The momentum transfer term developed for steady state is also assumed to apply to unsteady flow. M_{tr} is defined as:

$$M_{a} = \tau_{a} H_{f} \qquad [2.65]$$

The value of the apparent shear stress can be based one of the relations given in equations [2.13] to [2.17]. These equations are requoted here for easy reference.

Rajaratnam and Ahmadi (1979)

$$\tau_{a} = 0.15\left(\frac{H}{H_{f}} - 1\right)^{2} \tau_{f\infty} \quad \text{valid for } 2.2 \leq H/H_{f} \leq 7.4 \qquad [2.13]$$

Wormleaton *et al.* (1982)

$$\tau_{a} = 13.84(\Delta V)^{0.882}\left(\frac{H}{H_{f}}\right)^{3.123}\left(\frac{B}{T_{w} - B}\right)^{0.727} \quad \text{valid for } 2.3 \leq H/H_{f} \leq 9.0 \qquad [2.14]$$

Prinos and Townsend (1984)

$$\tau_{a} = 0.874(\Delta V)^{0.92}\left(\frac{H}{H_{f}}\right)^{1.129}\left(\frac{B}{T_{w} - B}\right)^{0.514} \quad \text{valid for } 3.0 \leq H/H_{f} \leq 11.2 \qquad [2.15]$$

Wormleaton and Merret (1990)

$$\tau_{a} = 3.325\, \Delta V^{1.451}\left(\frac{1}{H_{f}}\right)^{0.354} (T_{w} - B)^{0.519} \quad \text{valid for } 2.0 \leq H/H_{f} \leq 20.0 \qquad [2.16]$$

Christodoulou (1992)

$$\tau_a = \frac{1}{2}\rho\, C_{fa}\, \Delta V^2 \qquad \text{valid for} \quad 1.9 \le H / H_f \le 9.3 \qquad [2.17]$$

Table 2.1 Compound channel sizes and types used by researcher(s).

Researcher(s)	H/H_f	T_w/B	B/H	B_f/H_f	Sinuosity	Channel Re $\times10^{-4}$	Floodplain Re $\times10^{-4}$	Cross section shape
Sellin (1964)	6.8 - 10.6	4	2.2 - 2.3	22.3 - 37.1	1	7	1.4	Rect.
Toebes and Sooky (1967)	1.2 - 1.5	5.8	2.7 - 5.5	1.6 - 7.7	1.9	14	10	Rect.
Ghosh and Jena (1971)	1.6 - 5.8	1.4	2.3 - 3.3	1.0 - 3.6	1	-	-	Rect.
Ghosh and Kar (1975)	1.4 - 2.3	5.3	0.5 - 0.8	1.2 - 5.3	1.2	-	-	Rect.
James and Brown (1977) *	5.0 - 11.0	8	2.8 - 3.2	45 - 112	1 & 3	-	-	Trap.
Myers (1978)†	2.5 - 11.5	2.4	1.5 - 2.3	5.4 - 32	1	4.5	-	Rect.
Smith (1978)*	2.3 - 6.0	10	0.9 - 1.3	7.7 - 31	1 & 1.2	37	2.6	Trap.
Rajaratnam and Ahmadi (1979)	3.1 - 3.5	5.9	2.3 -3.3	12.1 - 14.5	1	1.1	0.26	Rect.
Ahmadi (1979)	2.1 - 2.5	4.8	3.5 - 4.2	6.7 - 7.9	1.13	6.1	2.2	Rect.
Rajaratnam and Ahmadi (1981)	2.2 - 7.4	1.7	3.9 - 6.5	6.1 - 33.4	1	2.5	1.2	Rect.
Wormleaton et al (1982)	2.3 - 9.0	4.2	1.4 - 2.1	5.1 - 30.6	1	-	-	Rect.
Knight et al (1983, 84)	1.9 - 9.3	4	1.0 - 1.8	2.9 - 24.9	1	3.9	3.5	Rect.
Prinos and Townsend (1984)	3.0 - 11.2	6	1.3 - 1.8	10.2 - 50.8	1	2.1	1.1	Rect.
Holden and James (1989)†	2.3 - 9.8	2.5	3.8 - 6.1	6.9 - 46.7	1	-	-	Trap.
McKeogh et al (1989)	2.0 - 2.3	6	2.0 - 2.8	10 - 24.5	1	3.4	3	Rect.
Kiely (1989)	2.0 - 2.3	6	2.0 - 2.8	10 - 24.5	1.25	3.4	3	Rect.
Wallingford (1992)	2.0 - 20	6.7	4.9 - 9.4	26 - 455	1	51	2.6	Trap.

Notes:

* - the researcher(s) used both straight and meandering channels with the same geometry

† - one floodplain only

Table 2.2 Mannings equivalent roughness for compound channels.

Method 1	Method 2	Method 3
$n_e = \left[\dfrac{\sum_{1}^{k=3} P_k n_k^{3/2}}{P_t} \right]^{2/3}$	$n_e = \left[\dfrac{\sum_{1}^{k=3} P_k n_k^{2}}{P_t} \right]^{1/2}$	$n_e = \dfrac{P_t R_t^{5/3}}{\sum_{1}^{k=3} \dfrac{P_k R_k^{5/3}}{n_k}}$
Assumptions: (1) $V_l = V_c = V_r = \overline{V}$ (2) $S_l = S_c = S_r = S_f = S_o$	Assumptions: (1) total force resisting motion is equal to the sum of the subsection resisting forces (2) $V_l = V_c = V_r = \overline{V}$ (3) $R_l = R_c = R_r = R$	Assumptions: (1) $Q_t = Q_l + Q_c + Q_r$ (2) $S_l = S_c = S_r = S_f = S_o$
Q_f error: -10% to -25% (the underestimation increases with increasing n)	Q_f error: -15% to -40% (the underestimation increases with increasing n)	Q_f error: +10% to +35% (the overestimation increases with increasing n)

Table 2.3 Classification of weirs by Govinda Rao and Muralidhar, (1963)

Values of H_1/B_w	Type of Weir
$0 < \frac{H_1}{B_w} \leq 0.1$	Long crested
$0.1 < \frac{H_1}{B_w} \leq 0.4$	Broad crested
$0.4 < \frac{H_1}{B_w} \leq 1.5 \text{ to } 1.9$ (upper limit depends on H_1/P)	Narrow crested
$\frac{H_1}{B_w} \geq 1.5 \text{ to } 1.9$ (Lower limit depends on H_1/P)	Sharp crested

Table 2.4 Discharge coefficients for sharp crested weirs.

Sharp Crest		
Author	Discharge Coefficient	Valid Range
Bazin (1898)	$C_d = 0.608 + 0.334\left(\frac{H_1}{H_1 + P}\right)^2$	$0 < \frac{H_1}{P} < 6$
Rehbock (1929)	$C_d = 0.611 + 0.08\frac{H_1}{P}$	$0 < \frac{H_1}{P} < 6$
Kandaswamy and Rouse (1957)	$C_d = 1.06\left(1 + \frac{P}{H_1}\right)^{3/2}$	$\frac{H_1}{P} \geq 15$
Swamee (1988)	$C_d = 1.06\left[\left(\frac{14.14P}{8.15P + H_1}\right)^{10} + \left(\frac{H}{H + P}\right)^{15}\right]^{-0.1}$	All ranges

Table 2.5 Discharge coefficients for long, narrow and broad crested weirs.

Author	Discharge Coefficient	Valid Range
Long Crested Weirs		
Govinda Rao, Muralidhar, (1963)	$C_l = 1.79\left(3\left(\frac{H_l}{B_w}\right)^{0.022}\right)$	$0 < \frac{H_l}{B_w} \le 0.1$
Swamee, (1988)	$C_l = \frac{2}{3}\sqrt{2g}\left(0.5 + 0.1\left(\frac{H_l}{B_w}\right)^{0.5}\right)$	$\frac{H_l}{B_w} < 0.1$
Narrow Crested Weirs		
Govinda Rao & Muralidhar, (1963)	$C_l = 1.79\left(0.64\left(\frac{H_l}{B_w}\right) + 2.63\right)$	$0.45 < \frac{H_l}{B_w} \le 1.5$
Swamee, (1988)	$C_l = \frac{2}{3}\sqrt{2g}\left(0.5 + 0.11\left(\frac{H_l}{B_w}\right)\right)$	$0.45 < \frac{H_l}{B_w} \le 1.5$
Broad Crested Weirs		
Govinda Rao, & Muralidhar, (1963)	$C_l = 1.79\left(0.15\left(\frac{H_l}{B_w}\right) + 2.82\right)$	$0.1 < \frac{H_l}{H_w} \le 0.4$
Clemmens, Reploge and Boss, (1984)	$C_l = \left(\frac{H_l + V^2/2g}{B_w} - 0.07\right)^{0.018} \frac{2}{3}\left(\frac{2}{3}g\right)^{0.5}$	$0.1 \le \frac{H_l + V^2/2g}{B_w} \le 1.0$
Swamee, (1988)	$C_l = \left(0.5 + 0.05\left(\frac{H_l}{B_w}\right)^{0.2}\right)\frac{2}{3}\sqrt{2g}$	$0.1 < \frac{H_l}{H_w} \le 0.4$

Table 2.6 Typical floodplain parameters for UK rivers (Samuels, 1985)

Parameter	River		Floodplain	
	range	typical	range	typical
Width, m	5 to 200	30	0 to 2000	500
Depth of flow, m	1 to 10	5	0 to 4	1
Velocity, m/s	0.5 to 3	1	0 to 2	0.3
Longitudinal Surface slope	0.01 to 0.00001	0.0005	0.01 to 0.00001	0.0005

Table 2.7 Discharge coefficients for side weirs.

Side Weirs (sharp crested)		
Author	Discharge Coefficient	Valid Range
Subramanya & Awasthy, (1972)	$C_d = 0.611\sqrt{1-\left(\frac{3F_1^2}{F_1^2+2}\right)}$	$0 \le H_w \le 0.6$ m
Yu-Tech, (1972)	$C_d = 0.622 - 0.222F_1$	$0 \le H_w \le 0.6$ m
Ranga Raju, *et al.* (1979)	$C_d = 0.81 - 0.60F_1$	$0.2 \le H_w \le 0.5$ m
Cheong, (1991)	$C_d = 0.45 - 0.22F_1^2$	$H_w = 0$
Manivannan *et al.* (1994)	$C_d = 0.33 - 0.18F_1 + 0.49\frac{H_w}{H_1}$	$0.06 \le H_w \le 0.12$ m
Swamee, et al. (1994)	$C_d = 0.447\left[\left(\frac{44.7H_w}{49H_w + H}\right)^{6.67} + \left(\frac{H - H_w}{H}\right)^{6.67}\right]^{-0.15}$	$0 \le H_w \le 0.1$ m
Side Weir (broad crested)		
Ranga Raju, *et al.* 1979)	$C_m = (0.81 - 0.6F_1)\left(0.80 + 0.1\frac{H_1 - H_w}{B_w}\right)$	$0.05 \le H_w \le 0.25$ m

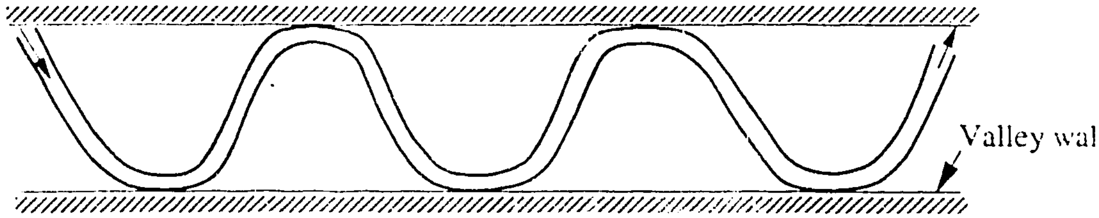


Figure 2.1 (a) confined meandering channel

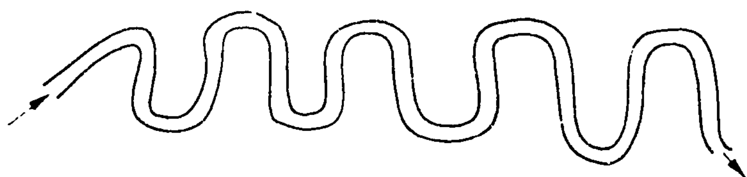


Figure 2.1 (b) regular meandering channel

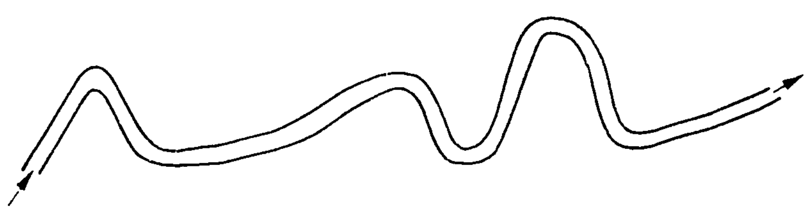


Figure 2.1 (c) irregular meandering channel

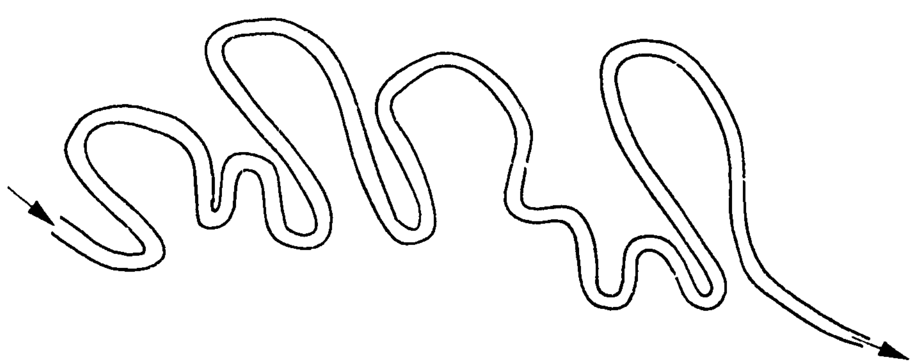
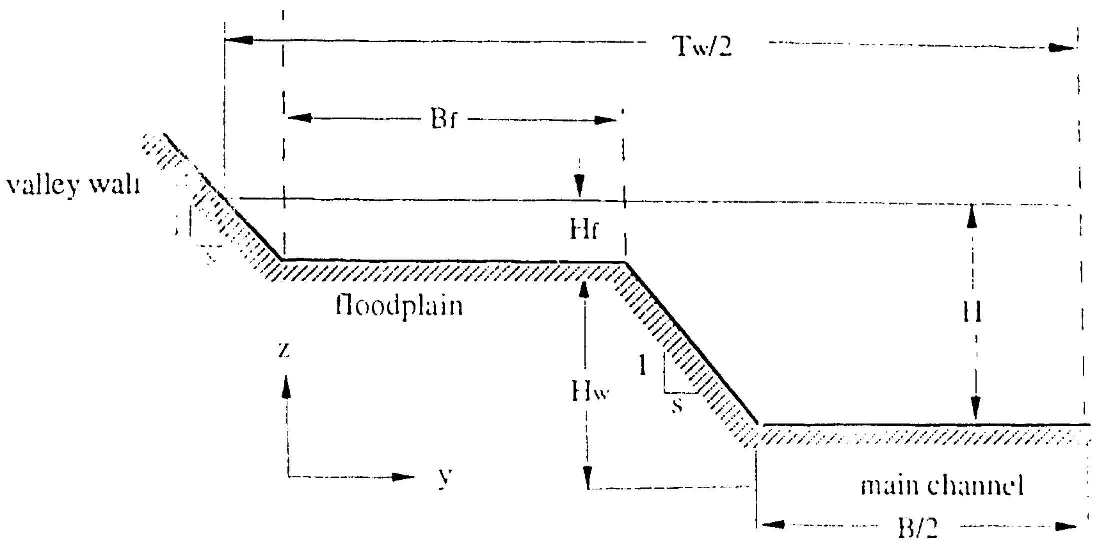
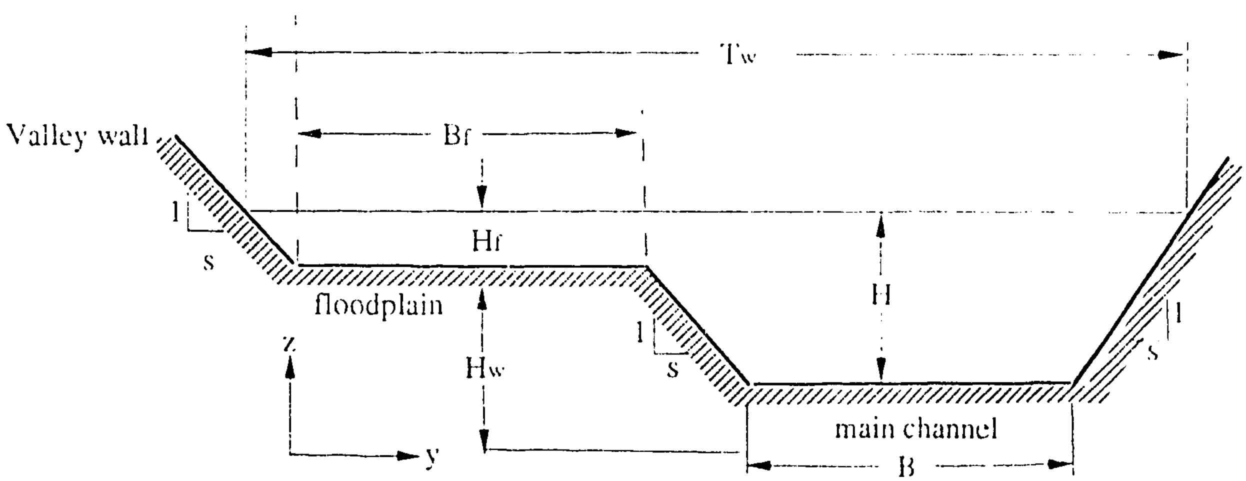


Figure 2.1 (d) Tortuous meandering channel



Symmetric (half channel shown)



Asymmetric (one floodplain)

Figure 2.2 Compound channel notation

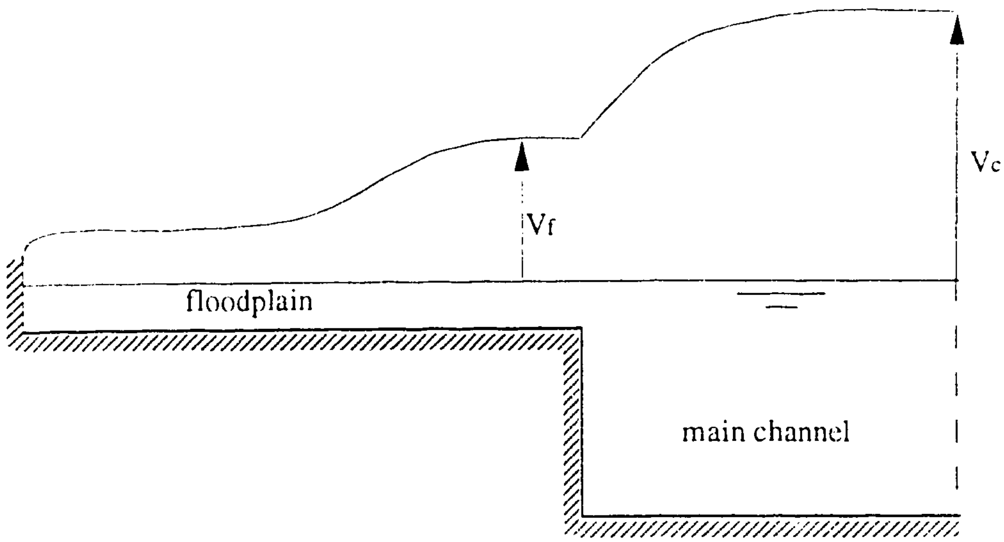


Figure 2.3 Typical velocity profile in a compound channel in a lateral direction as reported by Rajaratnam and Ahmadi (1981) for H/Hf =2.2 to 7.4 (half-channel shown).

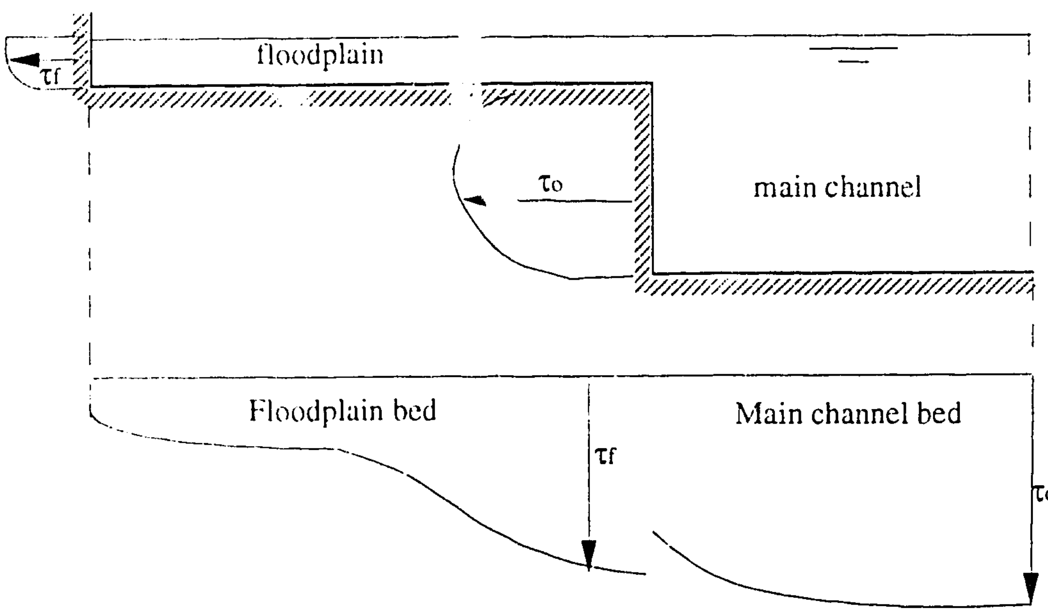


Figure 2.4 Typical shear stress profile in a compound channel in a lateral direction as reported by Rajaratnam and Ahmadi (1981) for H/Hf =2.2 to 7.4 (half-channel shown).

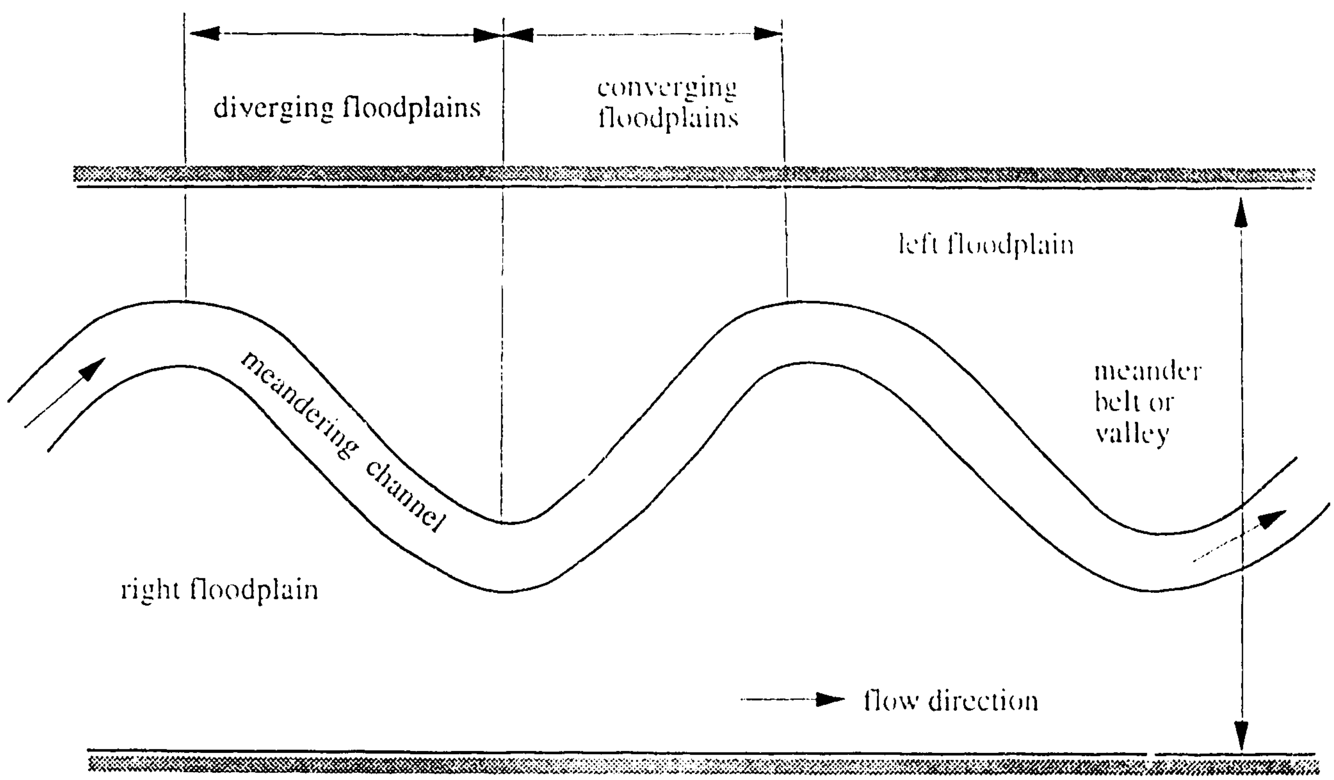


Figure 2.5 Meandering compound channel notation.

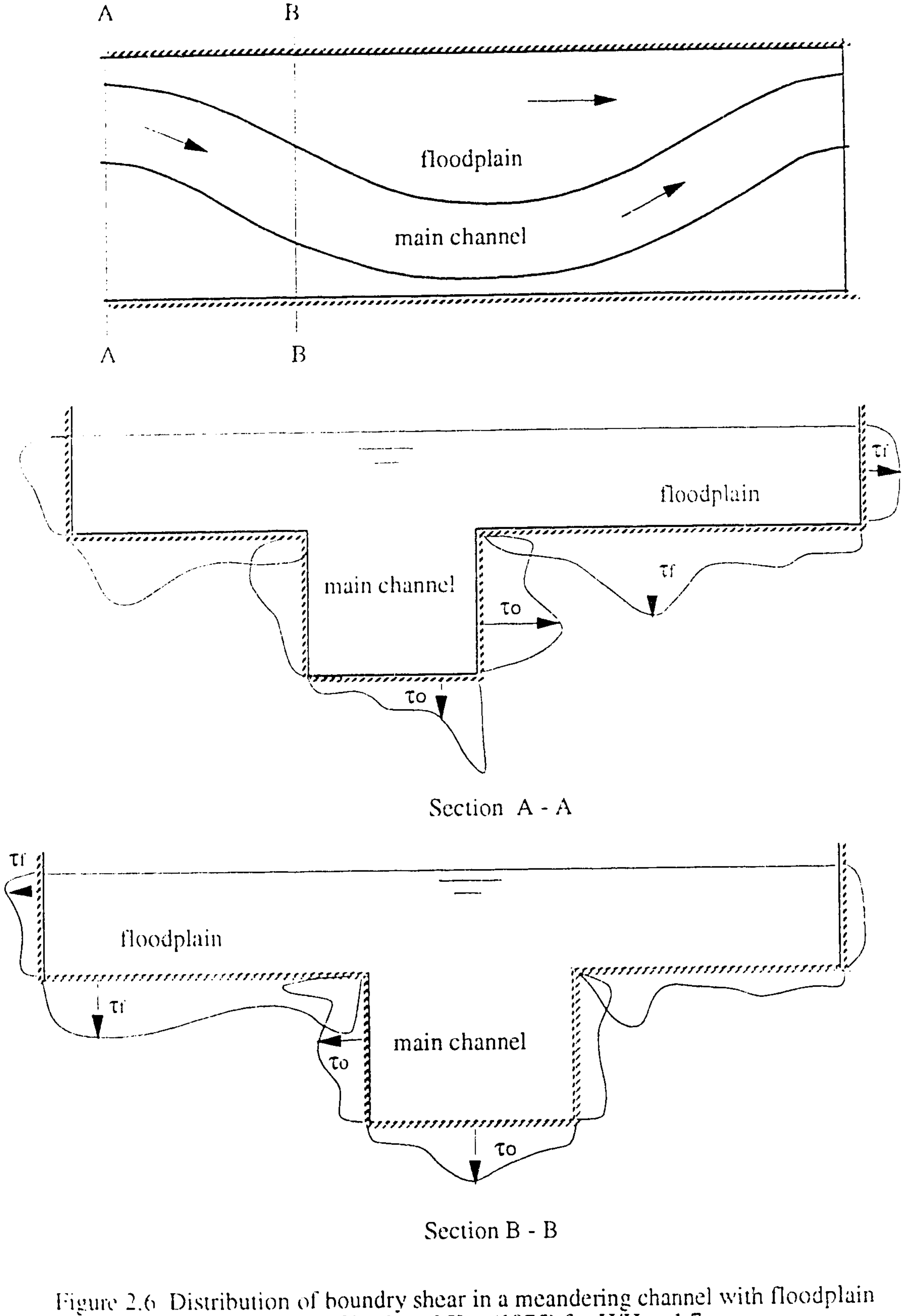


Figure 2.6 Distribution of boundry shear in a meandering channel with floodplain as presented by Ghosh and Kar (1975) for H/Hf = 1.7 (smooth boundary).

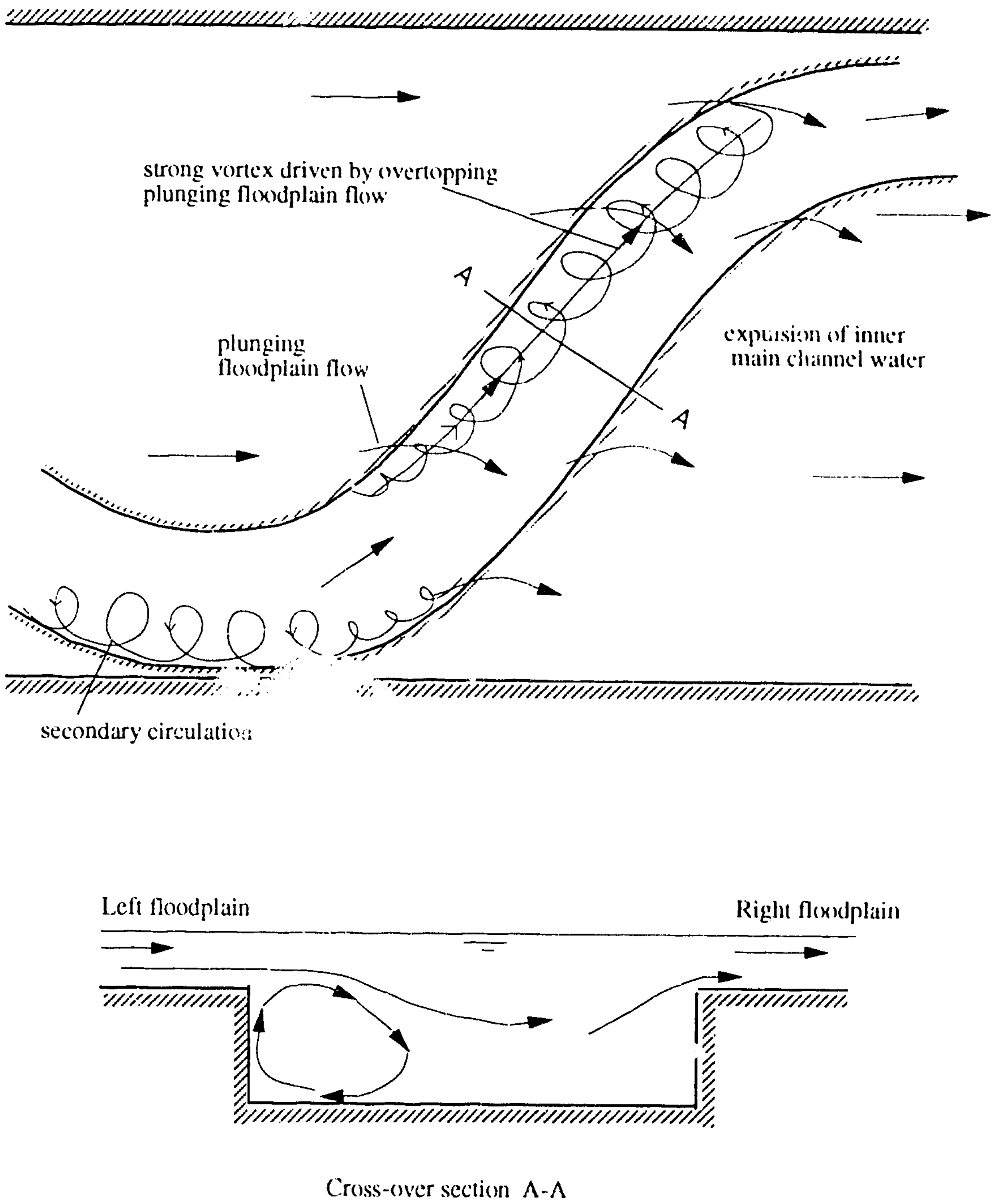


Figure 2.7 Representaion of interaction flow mechanisms in a meandering channel (after Ervine, Wiilets, Sellin and Lorena, 1993)

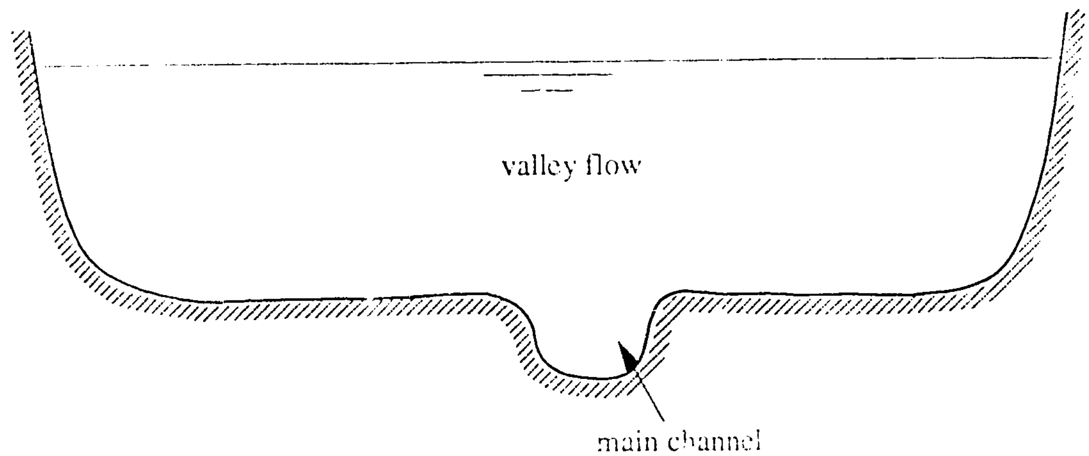


Figure 2.8 (a) Deep flow in the floodplains or valley.
(vertical scale exaggerated)

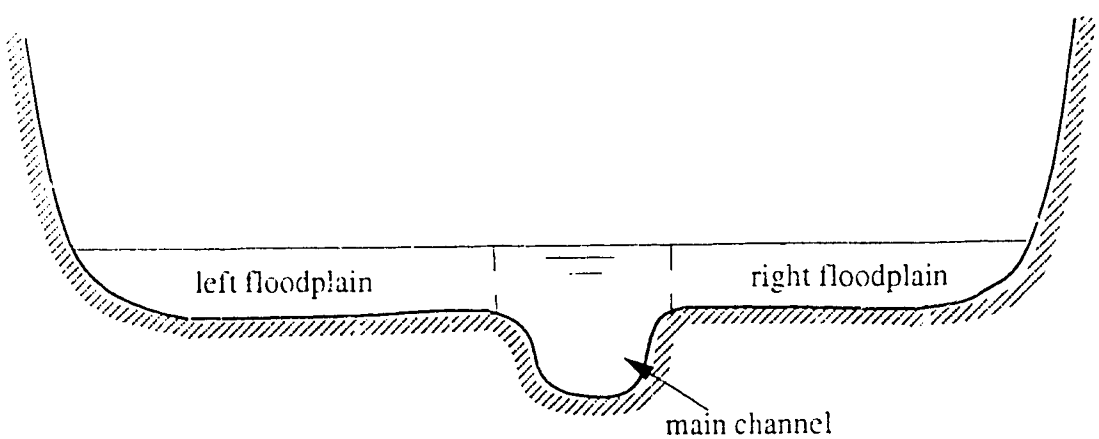


Figure 2.8 (b) Shallow flow in the floodplains or valley.
(vertical scale exaggerated)

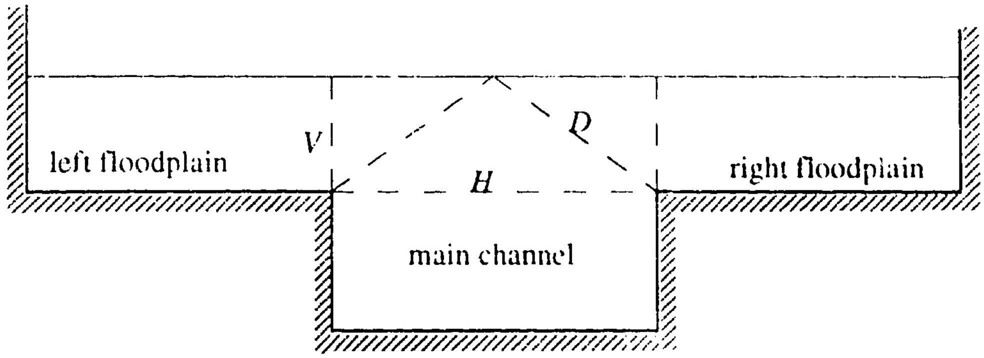


Figure 2.9 Compound channel cross-section showing possible division planes
(after Wormleaton, Allen, and Hadjipanos, 1982)

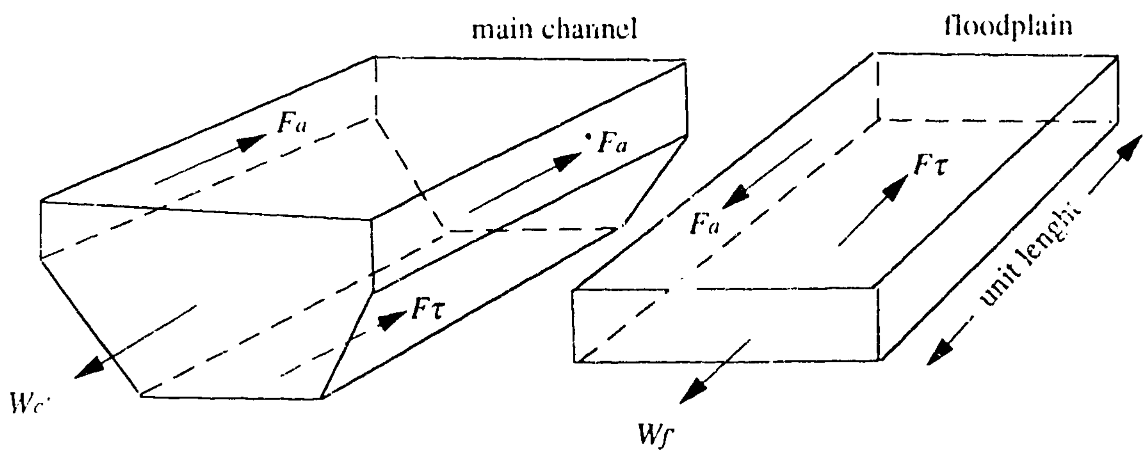


Figure 2.10 Main channel and floodplain control volumes.
(after Prinos and Townsend, 1984)

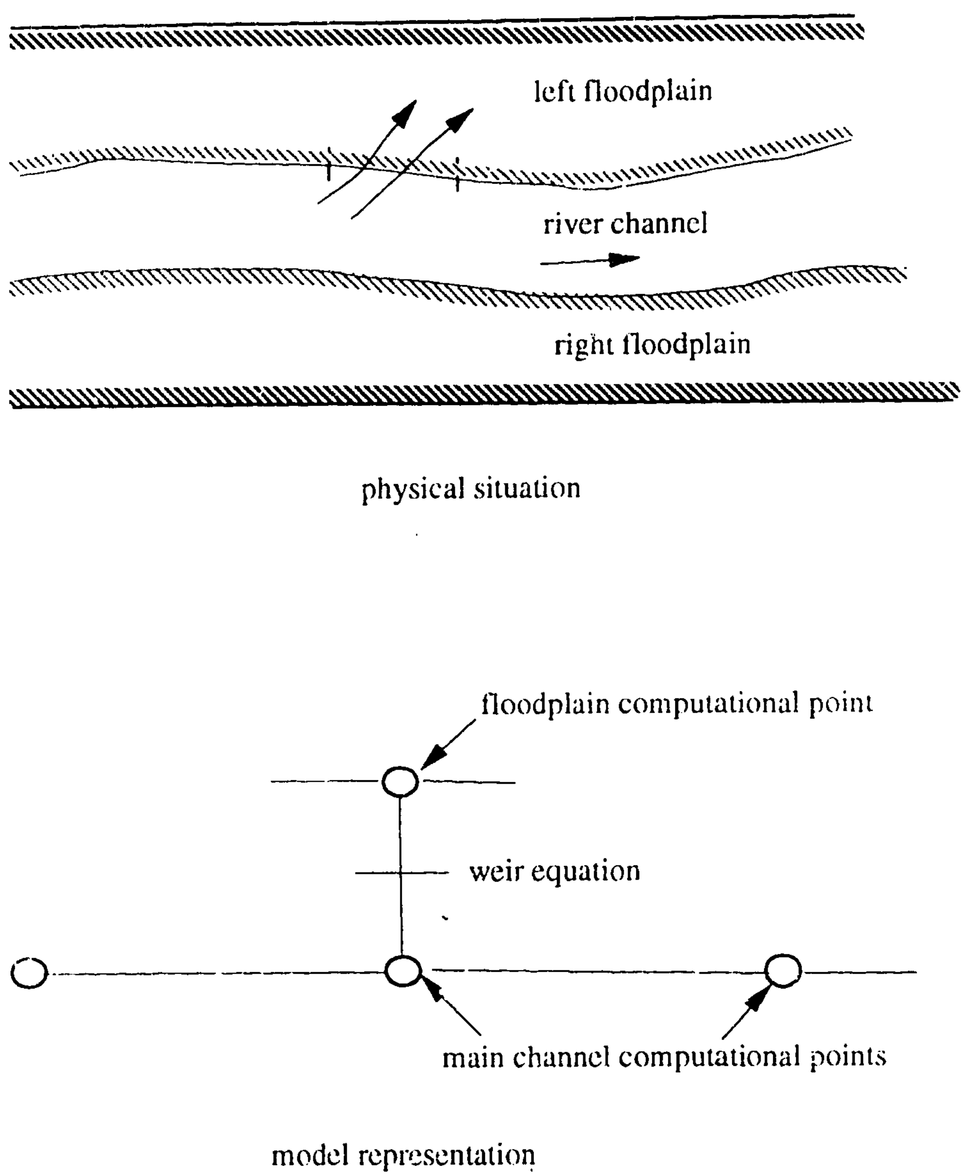


Figure 2.11 Model -1 representation of main channel-floodplain flow exchange.

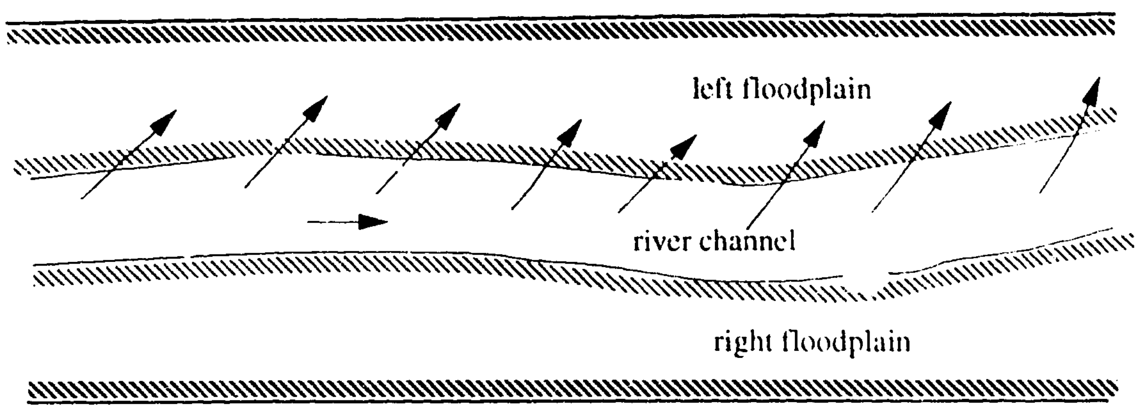


Figure 2.12 (a) General spilling of flow from main channel into the floodplain.

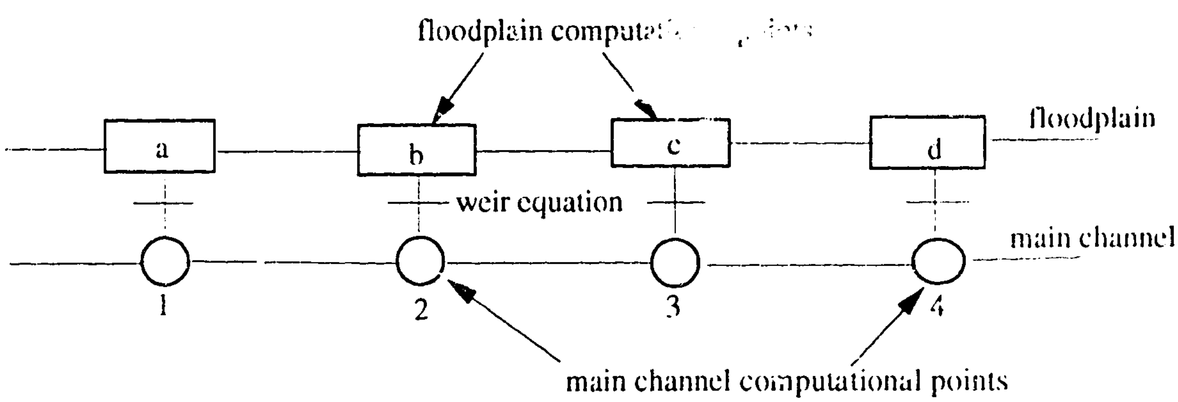


Figure 2.12 (b) Model-2 representation of main channel-floodplain flow exchange

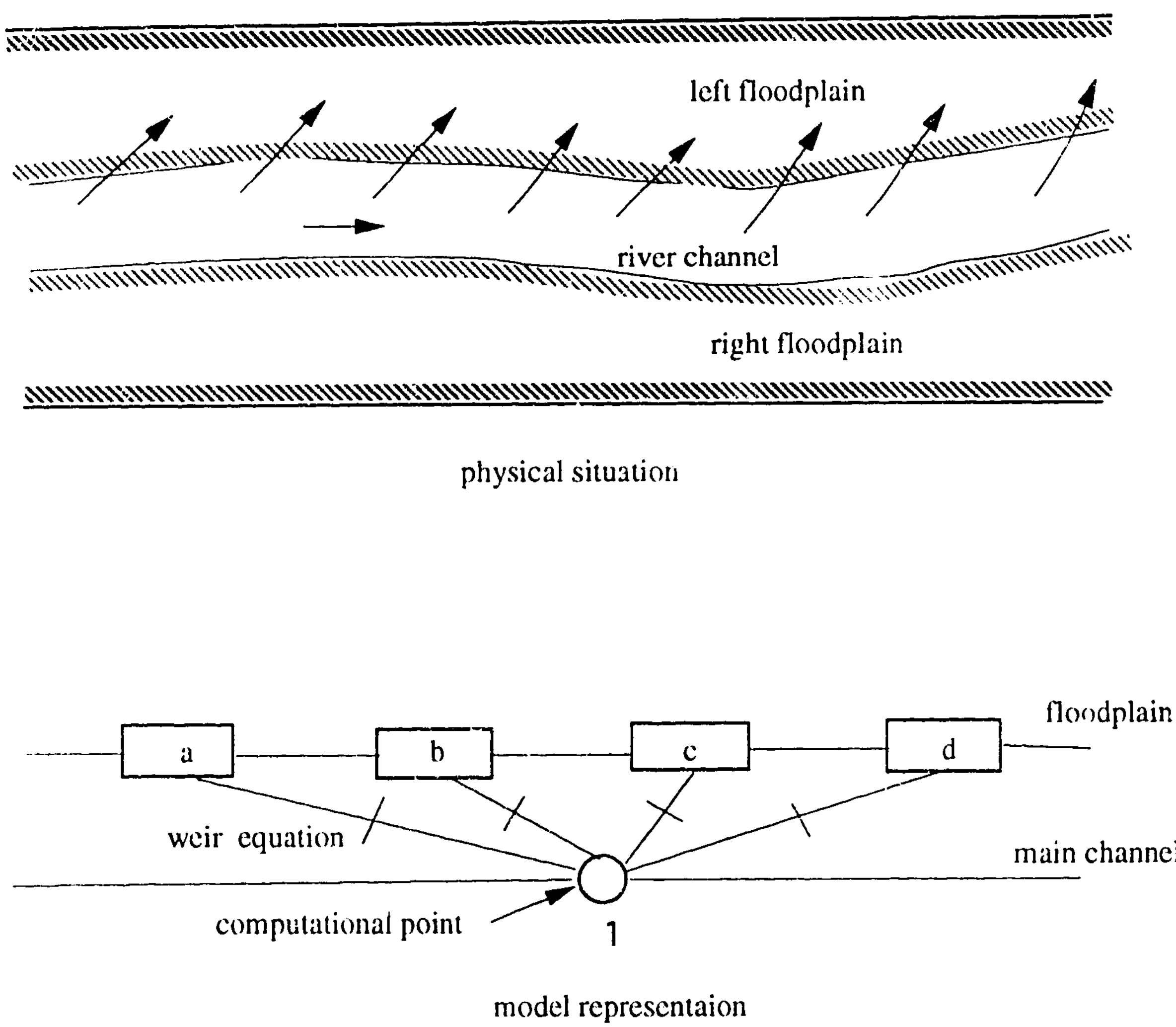


Figure 2.12 (c) Model-3 representaion of main channel-floodplain flow exchange

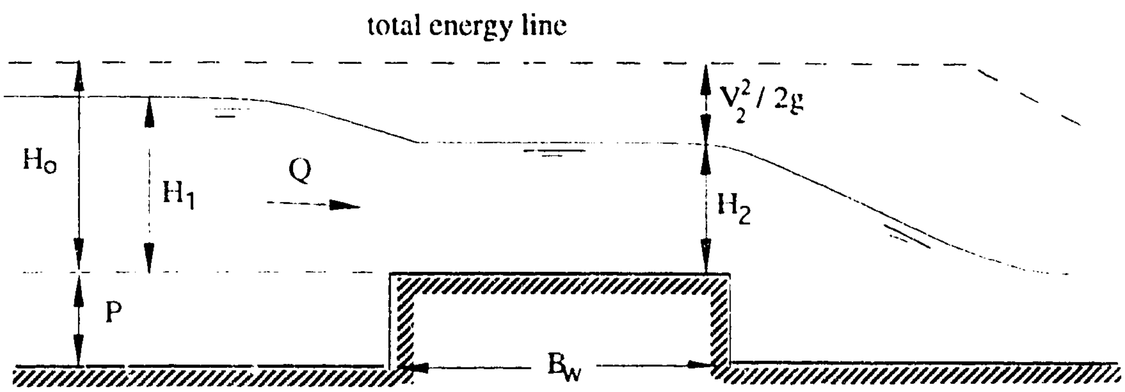


Figure 2.13 Perfect broad crested weir.

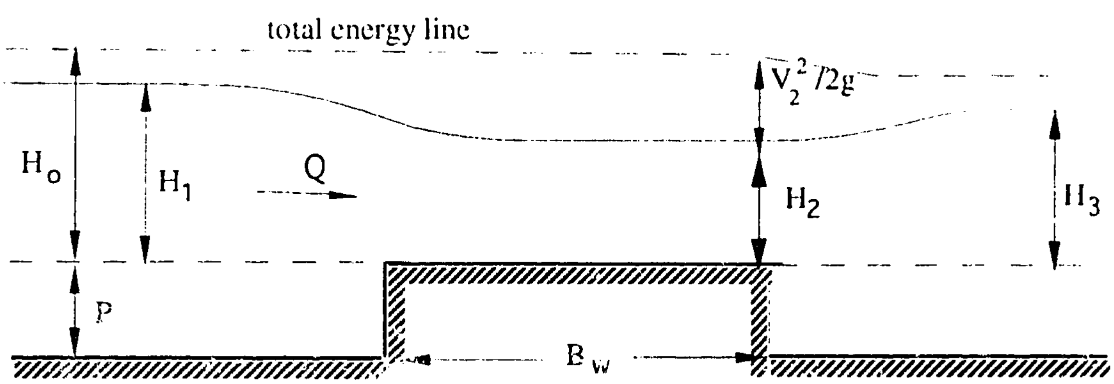


Figure 2.14 Non-modular flow over a broad crested weir.

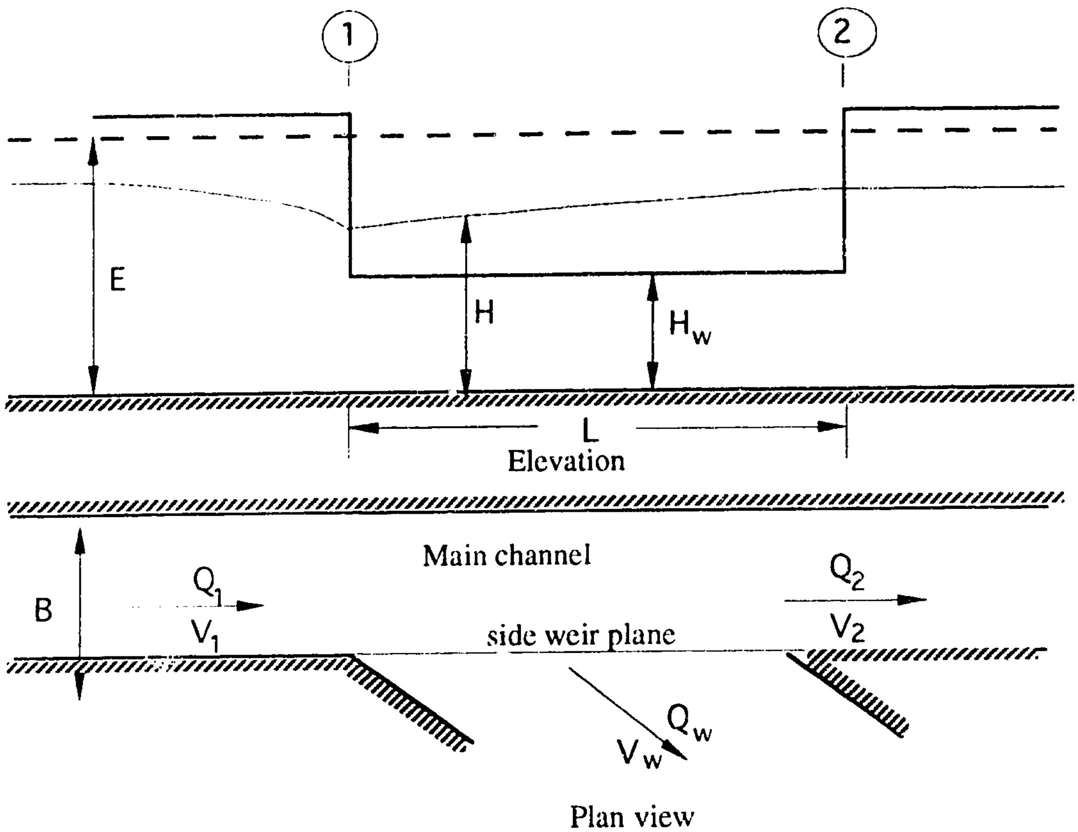


Figure 2.15 Side Weir.

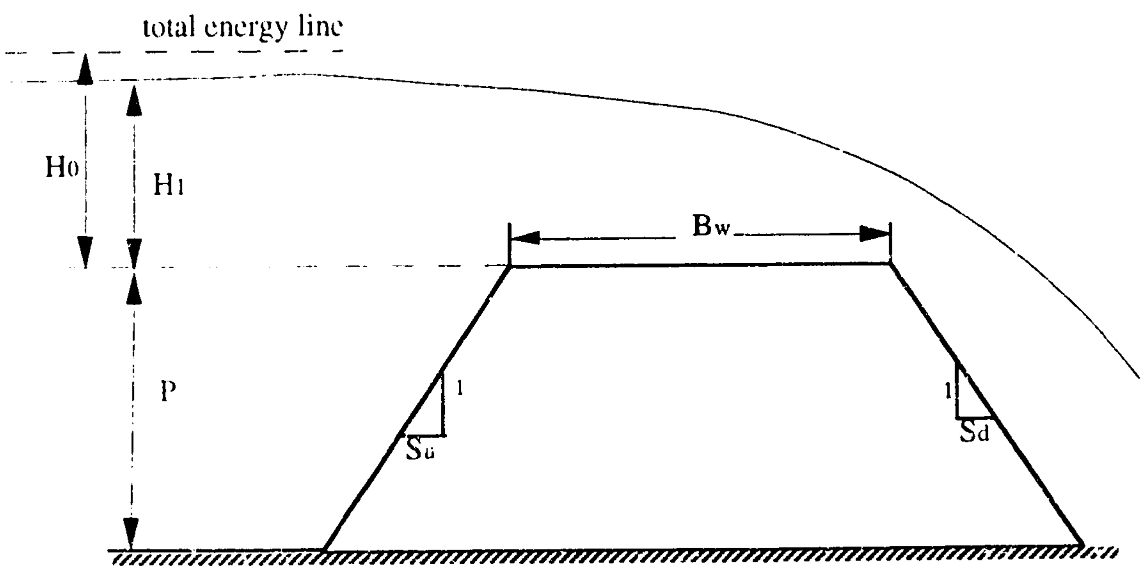


Figure 2.16 Weir with sloped upstream face

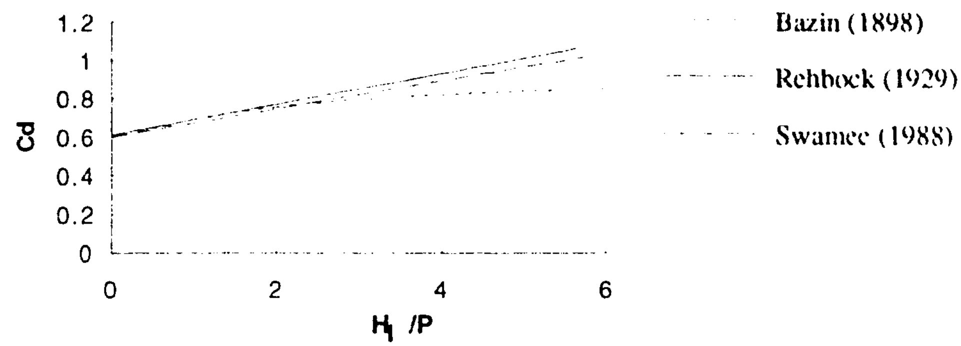


Figure 2.17 (a) Discharge coefficient for sharp crested weir

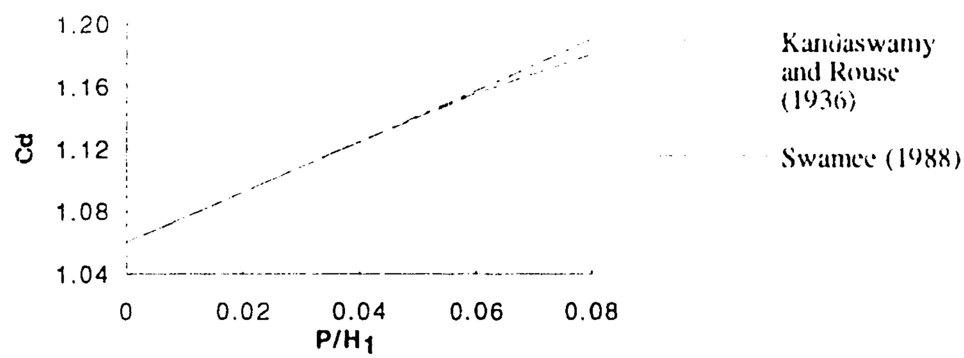


Figure 2.17(b) Discharge coefficients for sharp crested weirs in the sill range

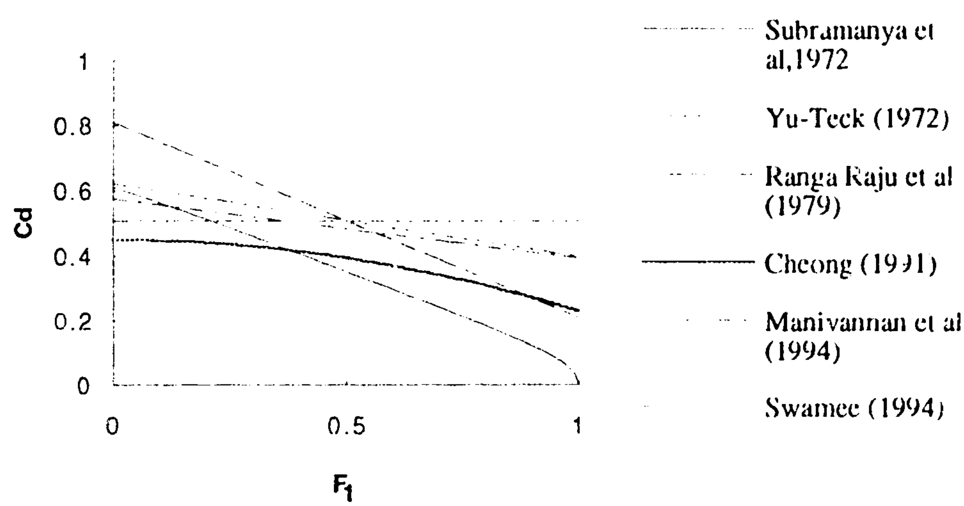


Figure 2.17(c) Comparison of side weir discharge coefficients

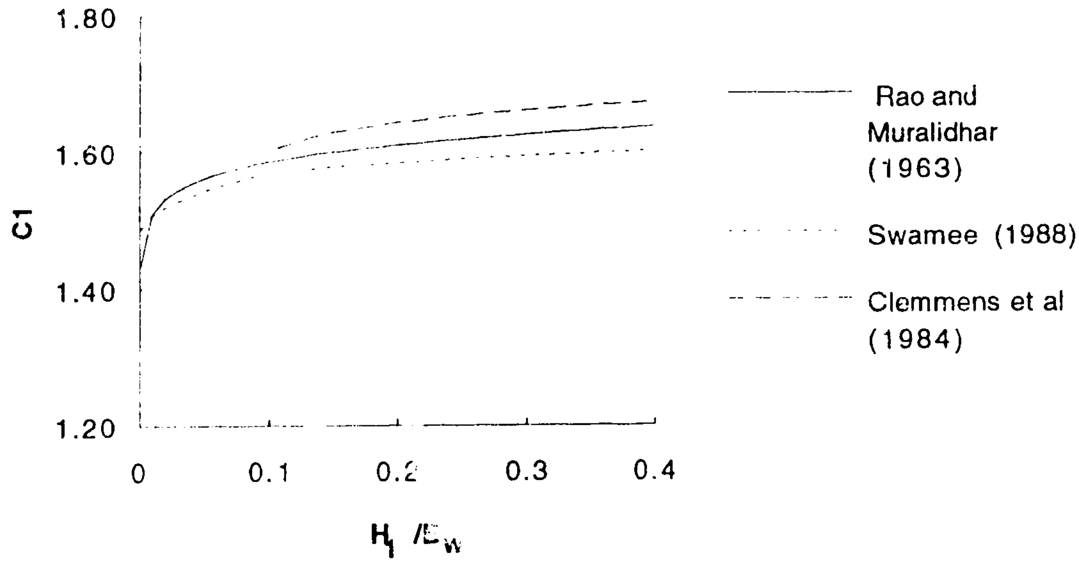


Figure 2.18 (a) Discharge coefficients for long and broad crested weirs

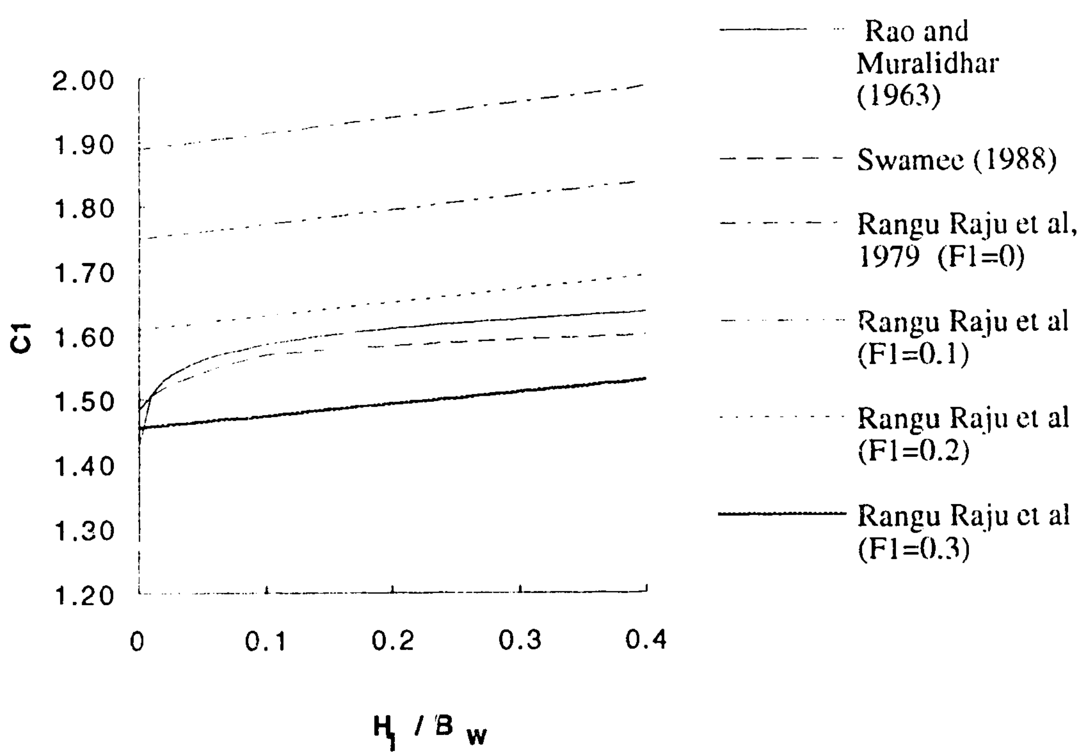


Figure 2.18 (b) Comparison of broad crested weir and side weir (broad crested shape)

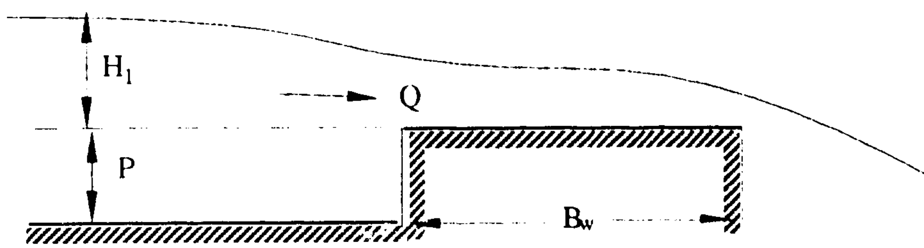


Figure 2.19 Flow over an end weir.

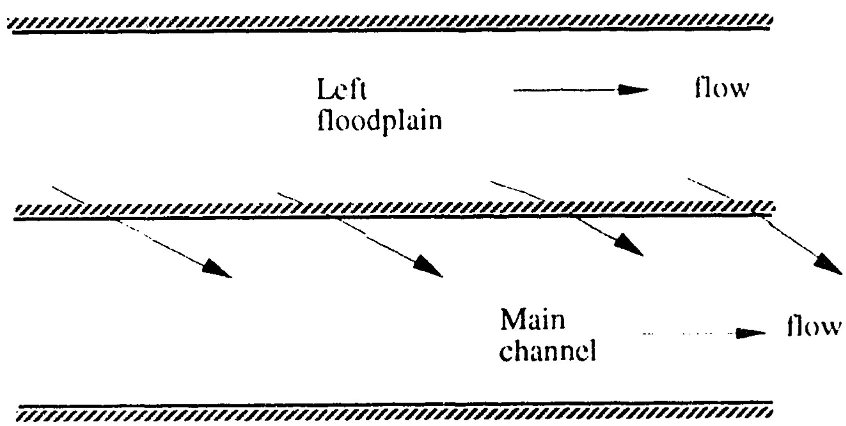


Figure 2.20 Flow from the floodplain into a river.

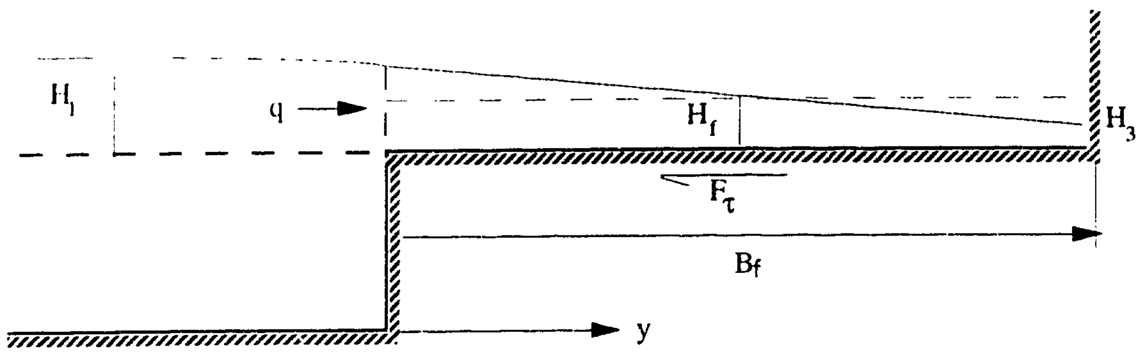


Figure 2.21. Lateral flow into the floodplain.

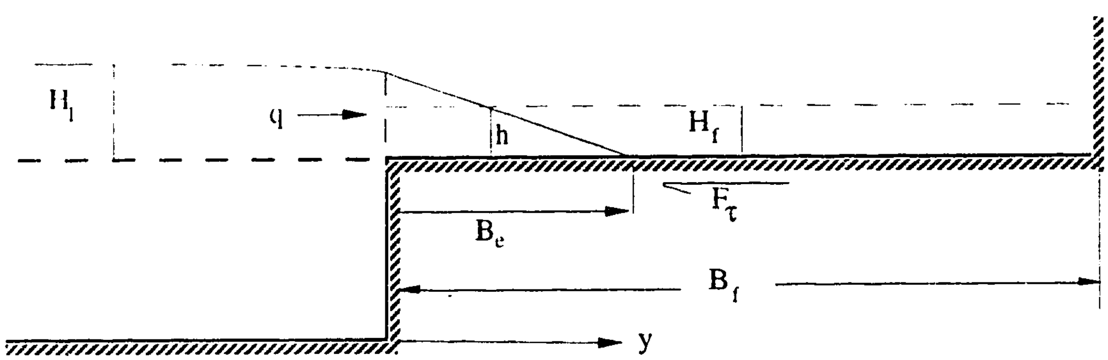


Figure 2.22. Lateral flow into partially inundated floodplain.

3.0 PROPOSED MODEL OF COMPOUND CHANNEL FLOW

3.1 Introduction

In many one-dimensional models, river floodplains are treated as storage areas (Yevjevich, 1975). When the conveyance capability of the floodplains is included, they are handled either as composite or compound flow sections. In this study, a one-dimensional model is proposed that handles the river and the floodplains as a compound channel while accounting for the flow interaction and mass transfer between the main channel and the floodplain through the introduction of an apparent shear force, and mass transfer equations.

It is recognized that as the flow rises above the channel bank and starts inundating the floodplain the flow is three-dimensional. However, if the dominant flow direction is in the longitudinal direction, a one-dimensional approximation can produce reasonable stage and discharge results, even though they may be approximate at certain times within the simulation (Cunge, Holly and Verwey, 1980).

3.2 Basic Equations of One-Dimensional Open Channel Flow

3.2.1 Basic Formulations of the St.Venant Equations

The basic formulation of the St. Venant equations; [2.19] and [2.20] for a channel with lateral inflow or outflow is given as (Chaudhry, 1993):

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = -q \qquad [3.1]$$

which represents conservation of mass and

$$\frac{\partial Q}{\partial t} + \frac{\partial (QV)}{\partial x} + gA\frac{\partial H}{\partial x} = gA(S_o - S_f) - V_x q \qquad [3.2]$$

which is the longitudinal momentum equation.

Where: V_{xq} is the convective momentum transport between the channel and the floodplain. The lateral outflow from the channel is considered negative while the lateral inflow is considered positive.

The x direction coordinate has been adopted as shown in Figure 3.1, where: S_1, S_2 and S_3 are cross-sections and l_f is a correction factor for the distance between cross-sections in the floodplain when the distance between cross-sections in the main channel is used as the measuring distance x. This correction factor is found to be equal to sinuosity. The assumptions that are required in the derivation of the above equations are (Yevjevich, 1975):

- streamlines are straight and parallel between cross-sections,
- the pressure distribution is hydrostatic;
- the channel width is allowed to change gradually;
- the velocity distribution is uniform cross the section;
- the channel bottom slope is small, so that the flow depths measured normal to the channel bed as those measured vertically are approximately the same;
- the water has constant density; and
- both dependent variables, that is A and Q, are continuous differentiable functions.

3.2.2 Other Forms of the St. Venant Equations

If the cross-section shape is approximated as a trapezoid, then the area is

$$A = BH + \frac{1}{2}(Z_1 + Z_2)\,H^2 = BH + \frac{1}{2}ZH^2 \qquad [3.3]$$

where:

B = main channel bottom width;

$$Z = Z_1 + Z_2; \qquad [3.4]$$

and Z_1 and Z_2 are the channel side slopes, as shown in Figure 3.2.

The pressure term in equation [3.2] may be rewritten using,

$$gA\frac{\partial H}{\partial x}=\frac{\partial}{\partial x}\left(g\frac{AH}{2}\right)-g\frac{H^2}{2}\frac{dB}{dx}-\frac{g}{4}ZH^2\frac{\partial H}{\partial x} \quad [3.5]$$

The full derivation of equation [3.5] is shown in Appendix A.

Neglecting convective momentum transfer and lateral mass transfer for now, equation [3.2] becomes:

$$\frac{\partial Q}{\partial t}+\frac{\partial (QV)}{\partial x}+\frac{\partial}{\partial x}\left(\frac{gAH}{2}\right)-\frac{g}{2}H^2\frac{dB}{dx}-\frac{g}{2}ZH^2\frac{\partial H}{\partial x}=gA(S_o-S_f) \quad [3.6]$$

Equation [3.6] is a quasi-conservation form of the longitudinal momentum equation for a trapezoidal cross section. The momentum equation may also be written in a completely non-conservation form using:

$$gA\frac{\partial H}{\partial x}=gH\frac{\partial A}{\partial x}-gH^2\frac{dB}{dx}-g\frac{ZH^2}{2}\frac{\partial H}{\partial x} \quad [3.7]$$

and

$$\frac{\partial (QV)}{\partial x}=2V\frac{\partial Q}{\partial x} \quad [3.8]$$

Substituting equ… …ation [3.2] one obtains:

$$\frac{\partial Q}{\partial x}+2V\frac{\partial Q}{\partial x}+\left(gH-V^2\right)\frac{\partial A}{\partial x}=gA(S_o-S_f)+gH^2\frac{dB}{dx}-g\frac{ZH^2}{2}\frac{\partial H}{\partial x} \quad [3.9]$$

If the cross section has a rectangular shape, then Z is equal to zero and $A=BH$. Then equation [3.6] can be reduced to:

$$\frac{\partial Q}{\partial t}+\frac{\partial (QV)}{\partial x}+\frac{\partial}{\partial x}\left(\frac{gAH}{2}\right)-g\frac{H^2}{2}\frac{dB}{dx}=gA(S_o-S_f) \quad [3.10]$$

while equation [3.9] takes the form of:

$$\frac{\partial Q}{\partial x} + 2V\frac{\partial Q}{\partial x} + \left(gH - V^2\right)\frac{\partial A}{\partial x} = gA(S_o - S_f) + gH^2\frac{dB}{dx} \qquad [3.11]$$

3.2.3 Diffusive Wave Approximation

The St.Venant equations simplify to a diffusive wave equation when the magnitudes of the inertial terms in equation [3.6] are very small in comparison to the pressure, slope and friction terms (Henderson, 1966). The reduced equations are:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0 \qquad [2.19]$$

and

$$\frac{\partial H}{\partial x} = S_o - S_f \qquad [3.12]$$

The friction slope (S_f) is the defined using conveyance (K) as:

$$S_f = \frac{|Q|Q}{K^2} \qquad [3.13]$$

here, the absolute value of discharge is taken so that the longitudinal friction slope retains the correct sign.

For the Chezys equation, K is expressed as:

$$K = C_* A\sqrt{gR} \qquad [3.14]$$

where: C_* is the nondimensional Chezy coefficient and R the hydraulic radius, defined as:

$$R = \frac{A}{P} = \frac{A}{B + H\left(\sqrt{1+Z_1^2} + \sqrt{1+Z_2^2}\right)} = \frac{A}{B + HZ_{mc}} \qquad [3.15]$$

for a trapezoidal cross section shape where P is the wetted perimeter and,

$$Z_{mc} = \left(\sqrt{1+Z_1^2} + \sqrt{1+Z_2^2}\right) \qquad [3.16]$$

For the Manning equation (in S.I units) K is given as:

$$K = \frac{1}{n} A R^{2/3} \qquad [3.17]$$

where: n is the Manning roughness coefficient.

Substituting equation [3.13] for the friction slope in equation [3.12] yields:

$$\frac{\partial H}{\partial x} = S_o - \frac{|Q|Q}{K^2} \qquad [3.18]$$

Equation [3.18] is differentiated with respect to x and the resulting expression for $\partial Q/\partial x$ is substituted into the continuity equation [3.1] to yield:

$$\frac{\partial A}{\partial t} + \alpha\, V \frac{\partial A}{\partial x} - \frac{Z_{mc}\, Q}{2(B + H\, Z_{mc})} \frac{\partial H}{\partial x} = D_l \frac{\partial^2 H}{\partial x^2} \qquad [3.19]$$

where D_l is a diffusion coefficient and α is a kinematic wave speed coefficient. D_l is defined for the Chezy formula as:

$$D_l = \frac{g\, C^2 A^3}{2|Q|\,(B + HZ_{mc})} \qquad [3.20]$$

and for the Manning formula as:

$$D_1 = \frac{A^{10/3}}{2n^2 \left(B + H\, Z_{mc}\right)^{4/3} |Q|} \qquad [3.21]$$

α for the Chezy equation is defined as:

$$\alpha \equiv \frac{3}{2} \qquad [3.22]$$

while for the Manning formula it is given as:

$$\alpha \equiv \frac{5}{3} \qquad [3.23]$$

The full derivation of equation [3.19] is shown in Appendix A.

Diffusive waves have only one wave component which propagates in the downstream direction. This is because there is only one convective velocity *(αV)*. They also attenuate as they propagate downstream. The rate at which this occurs is governed by the magnitude of the diffusion term D_1. Diffusive waves cause loop rating curves in which the water levels associated with the rising limb of a hydrograph are lower than those for the same discharge for the falling stage. This is because the diffusive velocity is a function of not only the depth but the slope of the water surface as well. This can be seen by the velocity equation below (for the Manning equation):

$$V = \frac{1}{n} R^{2/3} \sqrt{S_o - \frac{\partial H}{\partial x}} \qquad [3.24]$$

3.3 Proposed Formulation

3.3.1 Introduction

In the proposed formulation, a compound flow section is modelled by treating the flow in the floodplain (s) and the main channel as parallel, one-dimensional flows while allowing for the exchange of flow and longitudinal momentum between subsections. Although the flow exchanges suggest that the flow is two dimensional along the intersection of the main channel and the floodplain, the dominating flow direction is longitudinal.

The flow in the compound channel floodplain is modelled using a convection-diffusive wave model for the following reasons. First, a diffusive wave approximation facilitates the simulation of flow on a dry bed. This is particularly important to modelling the onset of floodplain inundation. Second, because of higher values of relative roughness on the floodplain (s) the magnitudes of the inertial terms in the equations are generally very small in comparison with the pressure, slope and friction terms. Further, Ponce (1977, 1978) showed that for a dimensionless wave number (σ_*) of approximately 0.1 or less, flood waves are predominantly kinematic or diffusive. He defined the dimensionless wave number as:

$$\sigma_* = \left(\frac{2\pi}{L}\right)L_o \qquad [3.25]$$

where

L = the wavelength of the wave disturbance; and

L_o = the horizontal length in which the steady uniform flow drops in head by an amount equal to its depth.

Ponce (1977) defined:

$$L_o = H_o/S_o \qquad [3.26]$$

where H_o and S_o are flow depth and slope, respectively. Therefore, equation [3.25] becomes:

$$\sigma_* = \frac{2\pi H_o}{LS_o} \qquad [3.27]$$

It is seen that for typical floodplain parameters, such as those shown previously in Table 2.6, a typical dimensionless wave number (σ_*) for flood waves in the floodplain can be determined. If H_o is taken as a typical depth , S_o as a typical slope, and L as a typical wavelength for flood waves in the floodplain then

$$\sigma_* = \frac{2\pi}{100x10^3} x \frac{1}{5x10^{-4}} = 0.12 \qquad [3.28]$$

where a typical L is about 100 km (Rijn, 1990). Since σ_* =0.12 is about 0.1, the diffusive wave approximation can be used to model flood waves in the floodplain.

It is proposed that the exchange of flow between the floodplain and the main channel be modelled using a side channel weir of either long or broad crested shape, as discussed in section 2.5. The accounting of the longitudinal momentum exchange between the floodplain and the main channel is done through the apparent shear force that is included in the forces acting on the control volumes of the sub-sections. The lateral convective momentum is assumed to be transfered into the floodplain without any loss.

This proposed formulation for modelling the compound flow section is restricted to rectangular and trapezoidal sections because the experimental work which has led to the evaluation of the apparent shear force has only been conducted for these types of channels. The method could be extended to natural sections once comparable apparent shear force relationshps are developed. The proposed model also does not consider the effect of sediment transport in compound channels which normally influences the bed channel roughness and formation of floodplains. Although this proposed formulation is limited in this aspect, its use on simulating real world situations is still expected to yield reasonable results. It is suggested that before the effect of sediment transport is included in this proposed formulation, the model should first be refined for situations of rigid boundaries.

3.3.2 Dynamic Equations for Main Channel Flow

The equations used to model the main channel are [3.1], [3.6] and [3.9] with the addition of new terms that account for the mass and momentum exchange between the main channel and floodplains. With these additional terms, the continuity equation becomes:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} + q_l + q_r = 0 \qquad [3.29]$$

and the momentum equation in a (pseudo-conservative form) takes the form of:

$$\frac{\partial Q}{\partial t} + \frac{\partial (QV)}{\partial x} + \frac{\partial}{\partial x}\left(\frac{gAH}{2}\right) - \frac{g}{2}H^2\frac{dB}{dx} - \frac{g}{4}ZH^2\frac{\partial H}{\partial x} =$$

$$gA(S_o - S_f) - V_x q_l - V_x q_r - M_{tr} \qquad [3.30]$$

where,

q_l = is the lateral discharge into (or out of) the left floodplain and

q_r = is the lateral discharge into (or out of) the right floodplain

$M_{tr} = M_{trl} + M_{trr}$ [3.31]

and,

M_{tr} = the total momentum transfer between the main channel and the the two floodplains;

M_{trl} = the momentum transfer between the main channel and the

the left floodplain ; and

M_{trr} = the momentum transfer between the main channel and the

the right floodplain.

M_{trl} and M_{trr} are positive when the lateral flow is to the floodplain (s) from main channel.

The longitudinal momentum equation in the non-conservative form is given as:

$$\frac{\partial Q}{\partial x} + 2V\frac{\partial Q}{\partial x} + \left(gH - V^2\right)\frac{\partial A}{\partial x} = \\ gA(S_o - S_f) + gH^2\frac{dB}{dx} - g\frac{ZH^2}{2}\frac{\partial H}{\partial x} - V_x q_l - V_x q_r - M_{tr} \quad [3.32]$$

where: V_x is the longitudinal component of the lateral velocity.

If there is lateral outflow into the floodplain, V_x is taken as the average channel velocity, because it is assumed that the lateral outflow goes with its full convective momentum. If there is lateral inflow into the main channel, V_x is assumed to be equal to respective floodplain longitudinal velocities for the same reason that the lateral inflow does not loose any of its momentum.

3.3.3 Diffusive Equation for Floodplain Flow

The derivation of the equations used to model the floodplain starts with equations [3.1] and [3.2] with the addition of new terms that account for the mass and momentum exchange between the main channel and the floodplain. The continuity equation [3.1] takes the form of:

$$\frac{\partial A_f}{\partial t} + \frac{\partial Q_f}{l_f \partial x} = q_f \quad [3.33]$$

where:

A_f = the floodplain cross section flow area;

Q_f = the floodplain discharge;

l_f = a correction factor for the distance between cross-sections in the floodplain when dx is the distance between cross-sections in the main channel; and

q_f = the lateral inflow from (or outflow to) the main channel.

The momentum equation becomes:

$$\frac{\partial Q_f}{\partial t} + \frac{\partial (Q_f V_f)}{l_f \partial x} + g A_f \frac{\partial H_f}{l_f \partial x} = g A_f (S_{of} - S_{ff}) + V_x q_f + M_{trf} \qquad [3.34]$$

where:

H_f = floodplain flow depth;

S_{of} = the longitudinal floodplain bed slope;

S_{ff} = the floodplain longitudinal friction slope;

M_{trf} = the momentum transfer between the main channel and one floodplain (M_{trl} or M_{trr})

As mentioned earlier, the floodplain equations are reduced to a convection-diffusion equation by neglecting the inertial terms. Equation [3.34] reduces to:

$$g A_f \frac{\partial H_f}{l_f \partial x} = g A_f \left(S_{of} - S_{ff}\right) + V_x q_f + M_{trf} \qquad [3.35]$$

Dividing through by gA_f, equation [3.35] becomes:

$$\frac{\partial H_f}{l_f \partial x} = S_{of} - S_{ff} + \frac{V_x}{g A_f} q_f + \frac{M_{trf}}{g A_f} \qquad [3.36]$$

Equation [3.36] is differentiated with respect to x and the resulting expression for $\partial Q_f / l_f \partial x$ is substituted in to the continuity equation [3.33]. If the Manning equation is used to define the conveyance (K) the equation becomes:

$$\frac{\partial A_f}{\partial t} + \frac{5}{3} V_f \frac{\partial A_f}{l_f \partial x} - \frac{2}{3} \frac{Z_f Q_f}{\left(B_f + H_f Z_f\right)} \frac{\partial H_f}{l_f \partial x} - D_2 \frac{\partial A_f}{\partial x} - D_3 \frac{\partial A_f}{\partial x} - q_f = D_1 \frac{\partial^2 H}{\partial x^2} \qquad [3.37]$$

$$D_1 = \frac{A_f^{10/3}}{2n^2\left(B_f + H_f Z_f\right)^{4/3} \left|Q_f\right| l_f^2} \qquad [3.38]$$

$$D_2 = \frac{A_f^{4/3} V_x q_f}{2n^2\left(B_f + H_f Z_f\right)^{4/3} \left|Q_f\right| l_f} \qquad [3.39]$$

$$D_3 = \frac{A_f^{4/3} M_L}{2n^2\left(B_f + H_f Z_f\right)^{4/3} \left|Q_f\right| l_f} \qquad [3.40]$$

where for the left floodplain:

$$Z_f = \left(\sqrt{1 + Z_3^2} + \sqrt{1 + Z_5^2}\right) \qquad [3.41]$$

and for the right floodplain:

$$Z_f = \left(\sqrt{1 + Z_4^2} + \sqrt{1 + Z_6^2}\right) \qquad [3.42]$$

where: B_f is floodplain bottom width; and Z_3; Z_4; Z_5; and Z_6 are side slopes as defined in Figure 3.2.

Rewriting equation [3.37] using the relations [3.38]-[3.40] gives:

$$\frac{\partial A_f}{\partial t} + \left(\frac{5}{3l_f} V_f - D_2 - D_3 \right) \frac{\partial A_f}{\partial x} - \frac{2}{3} \frac{Z_f Q_f}{\left(B_f + H_f Z_f\right) l_f} \frac{\partial H_f}{\partial x} - D_1 \frac{\partial^2 H_f}{\partial x^2} - q_f = 0 \qquad [3.43]$$

If one further, defines B_1 and B_2 as:

$$B_1 = \frac{5}{3\,l_f} V_f - D_2 - D_3 \qquad [3.44]$$

$$B_2 = \frac{Z_f Q_f}{2\left(B + H_f Z_f\right) l_f} \qquad [3.45]$$

Then equation [3.43] can be simplified to:

$$\frac{\partial A_f}{\partial t} + B_1 \frac{\partial A_f}{\partial x} - B_2 \frac{\partial H_f}{\partial x} - D_1 \frac{\partial^2 H_f}{\partial x^2} - q_f = 0 \qquad [3.46]$$

Equation [3.46] is the convection-diffusion equation modelled in the floodplains.

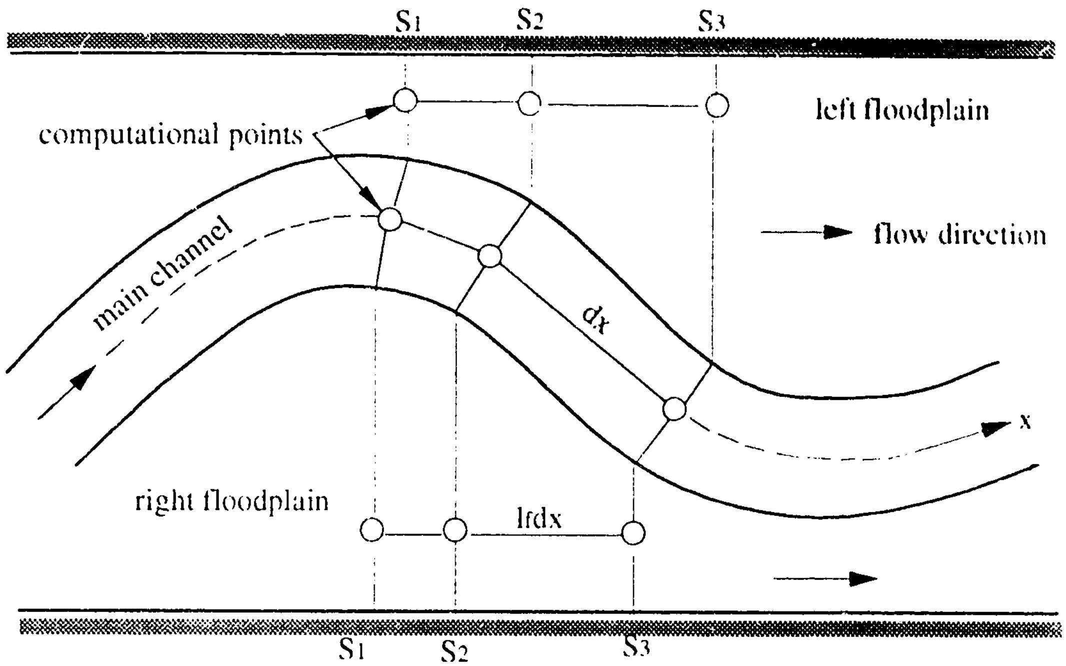


Figure 3.1 Definition sketch for one-dimensional flow in a meandering channel.

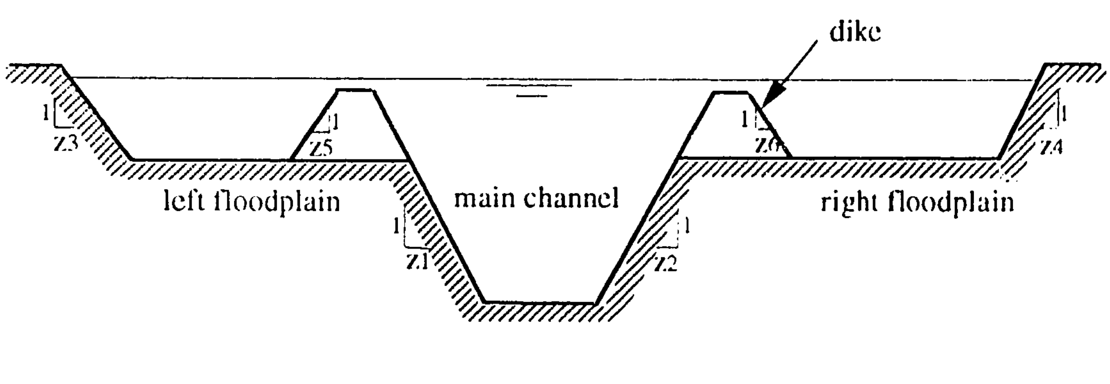


Figure 3.2 Compound channel showing the side slope notations.

4.0 Numerical Solution Technique

4.1 Introduction

As analytical solutions of the one-dimensional equations of open channel flow are only available for limited situations (Yevjevich, 1975), the equations formulated in Chapter 3 ([3.29], [3.30] and [3.46]) must be solved numerically. There are a number of numerical schemes which could be applied to solve this sytem of equations, as the application of numerical methods to the one-dimensional equations has been the subject of intense research for many years (Priessmann, 1961; Amien, 1968; Liggett and Cunge, 1975; Abbott, 1979; Cunge *et al*, 1980; Katopodes, 1984; and Hicks and Steffler, 1992, to name only a few). However, as it was the primary objective of this study to explore the formulation itself, the numerical technique may be considered as a merely a vehicle in obtaining the solution of the equations. Although this means that the proposed formulation could be implemented using any numerical technique suitable for the solution of non-linear hyperbolic systems, in order to ensure that the evaluation of the performance and validity of the proposed formulation was not obscured by limitations of the numericasl scheme, it was desirable to select a numerical technique that was computationally robust, as well as accurate. The characteristic-dissipative-Galerkin method (CDG) finite element scheme, first derived by Hughes and Mallet, (1986) and later adapted to open channel flow problems by Hicks and Steffler (1990, 1992), was considered ideal for this purpose.

4.2 The Characteristic Dissipative-Galerkin Method

The CDG scheme is a Streamline Upwind Petrov-Galerkin (SU/PG) based scheme (Hughes and Mallet, 1986) in which upwind weighted test functions are used to introduce selective dissipation based on the characteristic velocities of the propagating disturbances. Extensive investigations into the performance of this scheme in comparison to the popular "box" or "four-point implicit" finite difference scheme (Amien, 1968), have illustrated the superior stability and accuracy of the CDG finite element scheme for open channel flow applications, (Hicks and Steffler, 1990, 1992; Hicks, Steffler and Gerard, 1992). The characteristic-dissipative-Galerkin finite element method has also been shown to provide superior stability and more selective damping of numerical instabilities than the Taylor-Galerkin and Least Squares finite element schemes over a wide range of practical flow situations (Hicks and Steffler, 1990, 1995).

Implementation of the CDG on the continuity equation takes the form of:

$$\int_0^L \left(\mathbf{f}_i \, C + \omega \frac{\Delta x}{2} \left(W_{aa} \frac{d\mathbf{f}_i}{dx} \right) C + \omega \frac{\Delta x}{2} \left(W_{aq} \frac{d\mathbf{f}}{dx} \right) M \right) dx = 0 \qquad [4.1]$$

and the momentum equation is given as:

$$\int_0^L \left(\mathbf{f}_i \, M + \omega \frac{\Delta x}{2} \left(W_{qa} \frac{d\mathbf{f}_i}{dx} \right) C + \omega \frac{\Delta x}{2} \left(W_{qq} \frac{d\mathbf{f}_i}{dx} \right) M \right) dx = 0 \qquad [4.2]$$

Where: C and M refer to the continuity and momentum equations, respectively; $\mathbf{f}_i$ is the interpolating function; ω is the diffusion parameter termed the 'upwinding coefficient' while W_{aa}, W_{aq}, W_{qa}, and W_{qq} are components of the upwinding matrix $[\mathbf{W}]$ such that:

$$[\mathbf{W}] = \begin{bmatrix} W_{aa} & W_{aq} \\ W_{qa} & W_{qq} \end{bmatrix} \qquad [4.3]$$

The magnitudes of the components of the upwinding matrix $[\mathbf{W}]$ are determined by the characteristic velocities of both progressive and regressive dynamic wave components such that (Hicks and Steffler, 1992):

$$[W] = \frac{1}{2c}\begin{bmatrix} \dfrac{(c^2 - V^2)}{|V+c| - |V-c|} & \dfrac{(V+c)}{|V+c|} - \dfrac{(V-c)}{|V-c|} \\ (c^2 - V^2)\left(\dfrac{(V+c)}{|V+c|} - \dfrac{(V-c)}{|V-c|}\right) & \dfrac{(V+c)^2}{|V+c|} - \dfrac{(V-c)^2}{|V-c|} \end{bmatrix} \qquad [4.4]$$

in which $c = \sqrt{gH}$, represents the propagation velocity of a small disturbance in still water. It is important to note that although the upwinding matrix [W] is based on the dynamic wave characteristic propagation velocities, the scheme has been shown to provide excellent results for purely diffusive waves as well (Hicks, Yasmin and Chen, 1994).

Hicks and Steffler, (1990,1992) examined the sensitivity of the CDG scheme to the upwinding coefficient for three values of ω.: 0.25; 0.5; and 1.0, and found that ω = 0.5 optimized phase accuracy in the linear case, while at ω = 0.25 amplitude accuracy improved slightly while the reduction in phase accuracy was only marginal. Because of the limited effect of varying ω in the linear case, they recommended that a constant value of ω = 0.5 be used in modelling unsteady open channel flows. This value was adopted here.

The appropriate relationship between the spatial and ten.poral discretization for the CDG numerical scheme has been thoroughly examined through linear stability analyses (Hicks, 1990; Hicks and Steffler, 1992), and may be expressed in terms of the Courant number, C, which is defined as:

$$C = \frac{|V \pm c|\Delta t}{\Delta x} \qquad [4.5]$$

The CDG method was found to provide highly selective damping of high frequency (numerical) disturbances as well as good phase accuracy for values of the Courant number less than or equal to 0.5 (Hicks, 1990; Hicks and Steffler, 1992). In this study, the temporal and spatial discretizations in all test runs were designed to ensure the Courant number was within this range throughout the solution domain.

4.3 Implementation of the CDG for the Governing Equations

The equations to be solved for a trapezoidal compound section are given in equations [3.29], [3.30] and [3.46] and are restated here for easy reference.

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} + ql + qr = 0 \qquad [3.29]$$

$$\frac{\partial Q}{\partial t} + \frac{\partial (QV)}{\partial x} + \frac{\partial}{\partial x}\left(\frac{gAH}{2}\right) - \frac{g}{2}H^2\frac{dB}{dx} - \frac{g}{2}ZH^2 = gA(S_o - S_f) - V_x q_l - V_x q_r - M_{tr} \qquad [3.30]$$

$$\frac{\partial A_f}{\partial t} + B_1\frac{\partial A_f}{\partial x} - B_2\frac{\partial H_f}{\partial x} - D_f\frac{\partial^2 H_f}{\partial x^2} - q_f = 0 \qquad [3.46]$$

In the CDG method, the upwinding part uses the non-conservative form of the momentum equation given by equation [3.32].

$$\frac{\partial Q}{\partial x} + 2V\frac{\partial Q}{\partial x} + \left(gH - V^2\right)\frac{\partial A}{\partial x} = gA(S_o - S_f) + gH^2\frac{dB}{dx} - g\frac{ZH^2}{2}\frac{\partial H}{\partial x} - V_x q_l - V_x q_r - M_{tr} \qquad [3.32]$$

The weak statement is derived as:

$$[\mathbf{S}]\frac{\partial\{\Phi\}}{\partial t} + [\mathbf{K}]\{\Phi\} + \{\mathbf{F}\} = \{0\} \qquad [4.6]$$

where: [S] and [K] are mass and stiffness matrices, {F} is the load vector and {Φ} is defined as:

$$\{\Phi\} = \begin{Bmatrix} A_l \\ A \\ Q \\ A_r \end{Bmatrix} \qquad [4.7]$$

Using finite differences to discretize the time derivative, gives:

$$\frac{\partial \Phi}{\partial t} = \frac{\Phi^{n+1} - \Phi^{n}}{\Delta t} \qquad [4.8]$$

where Φ and F are defined as:

$$\Phi = \theta \Phi^{n+1} + (1-\theta)\Phi^{n} \qquad [4.9]$$

$$F = \theta F^{n+1} + (1-\theta)F^{n} \qquad [4.10]$$

and where: $n+1$ is the unknown time level, n is the known time level, and θ is the time weighting factor or implicitness, taken as θ=0.5 for the CDG scheme (Hicks and Steffler, 1992).

Rewriting equation [4.6], gives:

$$[S]\left(\frac{\Phi^{n+1} - \Phi^{n}}{\Delta t}\right) + [K]\left(\theta\Phi^{n+1} + (1-\theta)\Phi^{n}\right) + \left(\theta F^{n+1} + (1-\theta)F^{n}\right) = \{0\} \qquad [4.11]$$

Rearranging equation [4.11], yields:

$$\big[\,[S]+\Delta t\,\theta[K]\,\big]\left\{\Phi^{n+1}\right\}+\Delta t\,\theta\left\{F^{n+1}\right\}=$$

$$\big[\,[S]-\Delta t\,(1-\theta)[K]\,\big]\left\{\Phi^{n}\right\}-\Delta t\,(1-\theta)\left\{F^{n}\right\} \quad [4.12]$$

$[\mathbf{K}]$ and $\{\mathbf{F}\}$ depends upon the solution vector, $\{\,\Phi\,\}$. The matrices $[\mathbf{S}]$, $[\mathbf{K}]$ and $\{\mathbf{F}\}$ are defined as follows:

$$[\mathbf{S}]=\begin{bmatrix}[\mathbf{S}_{aa_L}] & 0 & 0 & 0\\ 0 & [\mathbf{S}_{aa}] & [\mathbf{S}_{aq}] & 0\\ 0 & [\mathbf{S}_{qa}] & [\mathbf{S}_{qq}] & 0\\ 0 & 0 & 0 & [\mathbf{S}_{aa_R}]\end{bmatrix}; \quad [4.13]$$

$$[\mathbf{K}]=\begin{bmatrix}[\mathbf{K}_{LF}] & 0 & 0 & 0\\ 0 & [\mathbf{K}_{aa}] & [\mathbf{K}_{aq}] & 0\\ 0 & [\mathbf{K}_{qa}] & [\mathbf{K}_{qq}] & 0\\ 0 & 0 & 0 & [\mathbf{K}_{RF}]\end{bmatrix} \text{ and} \quad [4.14]$$

$$\{\mathbf{F}\} = \begin{bmatrix} \mathbf{F}_{LF} \\ \mathbf{F}_a \\ \mathbf{F}_q \\ \mathbf{F}_{RF} \end{bmatrix} \quad [4.15]$$

where:

$[\mathbf{S}_{aa}]$ = mass matrix (continuity equation, area term);

$[\mathbf{S}_{aq}]$ = mass matrix (continuity equation, discharge term);

$[\mathbf{S}_{qa}]$ = mass matrix (momentum equation, area term);

$[\mathbf{S}_{qq}]$ = mass matrix (momentum equation, discharge term);

$[\mathbf{S}_{aaLF}]$ = mass matrix (left floodplain equation);

$[\mathbf{S}_{aaRF}]$ = mass matrix (right floodplain equation);

$[\mathbf{K}_{aa}]$ = stiffness matrix (continuity equation, area term);

$[\mathbf{K}_{aq}]$ = stiffness matrix (continuity equation, discharge term);

$[\mathbf{K}_{qa}]$ = stiffens matrix (momentum equation, area term);

$[\mathbf{K}_{qq}]$ = stiffness matrix (momentum equation, discharge term);

$[\mathbf{K}_{LF}]$ = stiffness matrix (left floodplain equation); and

$[\mathbf{K}_{RF}]$ = stiffness matrix (right floodplain equation).

The first subscript denotes the equation (a: continuity or 'area equation'; q: momentum or 'discharge equation'). The second subscript denotes the variable (a:area and q: discharge) in the case of main channel flow. The floodplain has only one equation and therefore the notations used are *LF* and *RF*. For the CDG method, the values of these matrices are defined as follows:

$$[\mathbf{S}_{aa}] \equiv \left[\int_e \left\{ (\mathbf{f}_i \, \mathbf{f}_j) + \left(w_{aa} \omega \frac{\Delta x}{2} \frac{d\mathbf{f}_i}{dx} \mathbf{f}_j \right) \right\} dx \right] \quad [4.16]$$

$$\left[\mathbf{S}_{aq}\right] \equiv \left[\int_e \left(w_{aq}\ \omega \frac{\Delta x}{2} \frac{d\mathbf{f}_i}{dx} \mathbf{f}_j \right) dx \right] \quad [4.17]$$

$$\left[\mathbf{S}_{qa}\right] \equiv \left[\int_e \left(w_{qa}\ \omega \frac{\Delta x}{2} \frac{d\mathbf{f}_i}{dx} \mathbf{f}_j \right) dx \right] \quad [4.18]$$

$$\left[\mathbf{S}_{qq}\right] \equiv \left[\int_e \left\{ (\mathbf{f}_i\ \mathbf{f}_j) + \left(w_{qq} \omega \frac{\Delta x}{2} \frac{d\mathbf{f}_i}{dx} \mathbf{f}_j \right) \right\} dx \right] \quad [4.19]$$

$$\left[\mathbf{S}aa_{LF}\right] \equiv \left[\int_e (\mathbf{f}_i\ \mathbf{f}_j)\ dx \right] \quad [4.20]$$

$$\left[\mathbf{S}aa_{RF}\right] \equiv \left[\int_e (\mathbf{f}_i\ \mathbf{f}_j)\ dx \right] \quad [4.21]$$

$$\left[\mathbf{K}_{aa}\right] \equiv \left[\int_e \omega \frac{\Delta x}{2} w_{aq} \left(-gS_o \frac{d\mathbf{f}_i}{dx} \mathbf{f}_j + \left(gH - V^2\right) \frac{d\mathbf{f}_i}{dx} \frac{d\mathbf{f}_j}{dx} \right) dx \right] \quad [4.22]$$

$$\left[\mathbf{K}_{aq}\right] \equiv \left[\int_e \left(-\frac{d\mathbf{f}_i}{dx} \mathbf{f}_j + \omega \frac{\Delta x}{2} \left(w_{aa} \frac{d\mathbf{f}_i}{dx} \frac{d\mathbf{f}_j}{dx} + w_{aq} f_c \frac{d\mathbf{f}_i}{dx} \mathbf{f}_j + 2 w_{aq} V \frac{d\mathbf{f}_i}{dx} \frac{d\mathbf{f}_j}{dx} \right) \right) dx \right] \quad [4.23]$$

$$\left[\mathbf{K}_{qa}\right] \equiv \left[\int_e \begin{pmatrix} \left(-\frac{gH}{2} \frac{d\mathbf{f}_i}{dx} \mathbf{f}_j - gS_o\ \mathbf{f}_i\ \mathbf{f}_j \right) \\ + \omega \frac{\Delta x}{2} w_{qq} \left(-gS_o \frac{d\mathbf{f}_i}{dx} \mathbf{f}_j + \left(gH - V^2\right) \frac{d\mathbf{f}_i}{dx} \frac{d\mathbf{f}_j}{dx} \right) \end{pmatrix} dx \right] \quad [4.24]$$

$$[\mathbf{K}_{qq}] \equiv \left[\int_e \left(\begin{matrix} \left(-V \dfrac{d\mathbf{f}_i}{dx} \mathbf{f}_j + f_c \, \mathbf{f}_i \, \mathbf{f}_j \right) \\ + \omega \dfrac{\Delta x}{2} \left(w_{qq} \, f_c \dfrac{d\mathbf{f}_i}{dx} \mathbf{f}_j + \left(w_{qa} + 2Vw_{qq} \right) \dfrac{d\mathbf{f}_i}{dx} \dfrac{d\mathbf{f}_j}{dx} \right) \end{matrix} \right) dx \right] \quad [4.25]$$

$$[\mathrm{K}_{lF}] \equiv \left[\int_e \left(\left(-\frac{5}{3} V_l + D_2 + D_3 \right) \frac{d\mathrm{f}_i}{dx} \mathrm{f}_j \right) dx \right] \quad [4.26]$$

$$[\mathrm{K}_{RF}] \equiv \left[\int_e \left(\left(-\frac{5}{3} V_r + D_2 + D_3 \right) \frac{d\mathrm{f}_i}{dx} \mathrm{f}_j \right) dx \right] \quad [4.27]$$

The contribution of the momentum transfer terms is reflected through the load terms.

$$\mathrm{F}_a \equiv \left\{ \int_e \left(\begin{matrix} \mathrm{f}_i \left(q_{lj} + q_{rj} \right) + \omega \dfrac{\Delta x}{2} \left(\left(w_{aa} \left(q_{lj} + q_{rj} \right) + \right. \right. \\ w_{aq} \left(q_l V_x + q_r V_x + M_{tr} \right) \dfrac{d\mathrm{f}_i}{dx} \mathrm{f}_i \\ - \omega \dfrac{\Delta x}{2} w_{aq} g H^2 \left(\dfrac{dB}{dx} + \dfrac{Z}{2} \dfrac{dH}{dx} \right) \dfrac{d\mathrm{f}_i}{dx} \end{matrix} \right) dx \right\} \quad [4.28]$$

$$F_q \equiv \left\{ \int_e \left(\begin{array}{l} f_i\left(V_x\left(o_{lj} + q_{rj}\right) + \left(M_{trlj} + M_{trrj}\right)\right) + \omega\frac{\Delta x}{2}((w_{qa}\left(q_{lj} + q_{lj}\right) + \\ w_{qq}\left(\left(q_{lj} + q_{rj}\right)V_x + \left(M_{trlj} + M_{trrj}\right)\right)\frac{df_i}{dx} \\ -\frac{gH^2}{2}\left(\frac{dB}{dx} + \frac{Z}{2}\frac{dH}{dx}\right)f_i \\ -\omega\frac{\Delta x}{2}w_{qq}gH^2\left(\frac{dB}{dx} + \frac{Z}{2}\frac{dH}{dx}\right)\frac{df_i}{dx} \end{array} \right) dx \right\} \quad [4.29]$$

$$F_{LF} \equiv \left\{ \int_e f_i\left(\left(D_l\frac{df_i}{dx} - B_2\right)\frac{dH_l}{dx} - q_l\right)dx \right\} \quad [4.30]$$

$$F_{RF} \equiv \left\{ \int_e f_i\left(\left(D_l\frac{df_i}{dx} - B_2\right)\frac{dH_r}{dx} - q_r\right)dx \right\} \quad [4.31]$$

4.4 Determination of Jacobian

The solution of equation [4.11] requires an iterative solution, because [**K**] and {**F**} depend upon the solution vector, { **Φ** }. In this study, a Newton-Raphson iteration scheme was employed. The Jacobian was evaluated analytically as:

$$[\mathbf{J}] = \left[\frac{\partial\{\mathbf{R}\}}{\partial\{\Phi\}}\right] \quad [4.32]$$

where: {**R**} is the residual vector and [**J**] is the Jacobian matrix.

The details of the determination of the residual {**R**} and Jacobian [**J**] are presented in Appendix A.

The convergence criteria was adapted after Hicks and Steffler, (1990) which was based on the root-mean square of the vector of corrections compared to a specified tolerance. If

$$\sqrt{\frac{\sum\left\{(\delta\Phi)^2\right\}}{\sum\left\{(\Phi)^2\right\}}} \leq \text{'tolerance'} \qquad [4.33]$$

then the solution would progress to the next step.

A complete listing of the finite element program is provided in Appendix B.

4.5 Boundary and Initial Conditions

Boundary and initial conditions are required to close the mathematical model. In this case the governing equations require that the initial values of discharge and flow area be specified both for the main channel and the floodplain. The geometric data for the compound flow cross section such as: the channel bed elevation; the floodplain bed elevations; and the step or dike heights are also required. The channel roughness can be specified through the Mannings n or through the relative roughness k_c.

The boundary conditions used in the model were problem specific. For subcritical flow, one boundary condition is specified at each end of the main channel. Normally, this would be a discharge at the upstream end and a water level or depth at the downstream end. Also, one boundary condition, depth, is specified at the upstream end of each floodplain. For supercritical flow, two boundary conditions must be specified at the upstream end of the channel domain.

5.0 SIMULATION ANALYSIS

5.1 Introduction

This chapter examines, and critically analyses, the performance of the proposed model in simulating stage discharge relationships, through the testing of reported laboratory data and field data. This includes the examination of both steady and unsteady flows observed in laboratory. Throughout this chapter, the proposed coupled formulation will be referred to as the CCDG 1-D model (Coupled Characteristic-Dissipative-Galerkin 1-D model).

As noted earlier, the main channel was modelled using the full dynamic equations in a pseudo-conservative form, while the floodplains were modelled using a diffusive wave model. All the tests were conducted with dependent variables for the main channel being the area and discharge, while that of the floodplain was area only. The discharge in the floodplain was determined through the solution of the momentum equation in the floodplain once the floodplain depths had been established. The upwinding matrix and upwinding parameter ω adapted after Hicks and Steffler (1992) were applied only to the main channel equations. The upwinding matrix was updated at every time step and ω of 0.5 was used for all tests as discussed in Chapter 4.

The iterative scheme used the analytical CDG Jacobian. These details are outlined in Appendix A. A specified tolerance of 10^{-5} (as defined in equation [4.33]) was imposed on the model except for steady flow simulations where a coarse tolerance of 0.1. Most of the tested problems converged in 3 to 4 iterations. Details of the initial and boundary conditions are provided with each test.

The momentum transfer term M_{tr} used to reflect the addition of the apparent shear force in the CCDG 1-D was tested to see how it affected the prediction of the steady state and unsteady state stage-discharge relationships. The methods investigated included those proposed by Prinos and Townsend (1984), Wormleaton and Merrett (1990) and Christodoulou (1992). For easy reference, these models will be differentiated as CCDG 1-

D (P-T) for the Prinos-Townsend model, CCDG 1-D (W-M) for the Wormleaton-Merrett model and CCDG 1-D (C) for the Christodoulou model. All the three methods were derived for symmetrical cases but both symmetric and asymmetric sections were tested in this study.

Sensitivity analyses were also carried out to see how the coefficient of discharge affects the lateral outflow or inflow into the floodplain of a compound channel and finally the CCDG 1-D model was used to test flow through a breached dike and a meandering channel to see how well it predicts the behavior of flow in both the main channel and the floodplain.

5.2 Verification of the Proposed Formulation for Steady Flow

5.2.1 Introduction

The set of tests used in the analysis were mainly that reported from laboratory data. Two different sets of experimental data were selected: the tests conducted by Prinos and Townsend (1984), and data from the Wallingford Research Institute in the UK. All data sets were obtained under uniform flow conditions. In order to study the impact of including the apparent shear at the interface of the main channel and floodplain, the CCDG 1-D model results were compared to the observed data. The observed discharges have also been compared to discharges calculated from conventional methods. The boundary conditions were adopted as area for both the upstream and downstream boundaries. All the steady state runs were fully implicit with a coarse convergence tolerance of 0.1.

5.2.2 Prinos-Townsend Experimental Data

Prinos and Townsend data were selected to test the CCDG 1-D model, because their data included a roughness variation in the floodplain part of the compound channel. Figure 5.1 shows the cross-section dimensions of the compound channel they used in their experiments. The main channel was 10.2 cm deep with the vertical to horizontal side slope ratio of 2V:1H. The bottom width of the main channel was set at 20.3, 30.5, 40.6 and 50.8 cm. In this study the data examined was that from the main channel whose bottom width was 20.3 cm. The floodplains on either side, were 38.1 cm wide. The first set of experiments tested on the model were performed with the main channel and floodplains having the same roughness (n =0.011) and then the floodplain roughness was varied to 0.014 and 0.022. The observed data were plotted together with the CCDG 1-D model results.

5.2.3 Methods Used for Comparison with CCDG 1-D model

The stage-discharge curves obtained using divided channel methods 2 and 3 and the diagonal method were compared to the observed data. The divided channel methods were chosen because they are traditionally used in steady state compound flow calculations, while the diagonal method is currently the best method in discharge prediction in compound channels. The divided channel method 1 was omitted because methods 1 and 2 are similar in predicting the stage-discharge relationship. In computing the discharge using these methods, the same compound channel conditions described in the Prinos-Townsend experiments were used. The results obtained using these methods are shown together with the CCDG 1-D model results.

5.2.4 CCDG 1-D Computational Model

The apparent shear stress models used in this study are the CCDG 1-D (P-T) model, CCDG 1-D (W-M) model and CCDG 1-D (C) model. The CCDG 1-D model reduces to the divided channel method 3 when apparent shear stress is excluded from the model. The compound channel conditions modelled were the same as those described in the Prinos-Townsend experimental setup. The results for the CCDG 1-D model and the divided channel methods are shown in Figures 5.2, 5.3 and 5.4 for the roughness in the floodplain of 0.011, 0.014 and 0.022 respectively. The discharge percentage error associated with the CCDG 1-D models and the divided channel methods are shown in Table 5.1 while Figures 5.5, 5.6 and 5.7 show the change of the percentage error with relative depth H_f/H. The percentage error has been calculated as:

$$\Delta Q \% = \frac{Q_p - Q_{ob}}{Q_{ob}} X 100 \qquad [5.1]$$

where:

Q_p = predicted or estimated discharge and

Q_{ob} = observed discharge.

The comparison of results for the diagonal method and CCDG 1-D model are shown in Figures 5.8, 5.9 and Figures 5.10 for the roughness in the floodplain of 0.011, 0.014 and 0.022 respectively.

5.2.5 Discussion of Results

The results obtained for uniform flow conditions were compared to the observed data. The stage-discharge results generally show that the divided channel method 3, the diagonal method and the CCDG 1-D models overestimated the discharge while the divided method 2 underestimated the discharge over all stages. It is, however, noted that the CCDG 1-D models overestimated the discharge by smaller values as compared to the divided channel method 3 and roughly behaves in the same manner as the diagonal method. Table 5.1 shows that for the floodplain roughness of 0.011, the divided channel method 3 overestimates the discharge by 22.7 % at low stage and 2.4 % at high stage while the worst of the three CCDG 1-D models overestimates by 16.7% at low stage and by 0.5 % at high stage. The diagonal method overestimated the flow by about 15% for low stages and underestimates the discharge by 3.5 % at high stages. This means that although the diagonal method is supposed to approximate lines of zero shear, at low flow over the floodplain, it does not perform any better than the other methods. The divided channel method 2 underestimates the discharge by 19.7 % at low stage and 5.4 % at high stage. The CCDG 1-D (P-T) model gave the best results because it only overestimated the discharge by 10.6% at low stage and 0.5 % at high stage. The CCDG 1-D(C) model gave the highest overestimation of about 17 % for the low stage among the CCDG 1-D models.

As the floodplain roughness was increased from 0.011 to 0.022, the prediction of discharge at low stage was poor by all methods. As the Table 5.1 shows, for the floodplain roughness of 0.022, the divided channel method 3 overestimates the discharge by 62.5 % while the divided channel method 2 underestimates by 35.4 %. Among the CCDG 1-D models, the CCDG 1-D (P-T) model overestimates by 37.5 % while the diagonal method overestimates by 51.2 %.

As the stage was increased, the divided channel methods and the CCDG 1-D models predicted the flow better. Figures 5.5, 5.6 and 5.7 show that, the trend of percentage error reduces with increasing relative depth. These figures also show that the CCDG 1-D models and the diagonal methods generally perform better than the divided channel methods because the CCDG 1-D models account for the effect of apparent shear on the flow, while the diagonal method attempts to account the effect of apparent shear by dividing the flow sections along lines of zero shear.

5.2.6 Wallingford SERC Flood Channel Facility Experimental Data

The data from Wallingford, UK (1992) were used in this study because it had more observed data at low stages. Figure 5.11 shows the cross-section dimensions of the Wallingford experimental setup, which consisted of a symmetrical compound channel with a trapezoidal main channel and a rectangular floodplain section. The bottom width of the main channel was 1.5 m and the channel depth was 0.15m. The main channel side slopes had a ratio of 1V:1H. The floodplain bed width was 4.1 m wide on each side of the main channel. The channel bed slope was 0.001027. The model tests were performed as the depths increased from 0.159 m to 0.25 m.

The compound channel was smooth. The Mannings *n* for the channel bed had different values for the bottom part of the main channel and the sloping part. It was given as varying from 0.0085 to 0.0122 for the sloping and the bottom parts respectively. The Mannings roughness for the floodplain varied from 0.0098 for low stages to 0.0092 for high stages.

5.2.7 Computational Tests

The compound channel conditions modelled were the same as those described in the Wallingford experimental facility. In this study, the floodplain roughness was used for both the main channel and the floodplain because the equivalent roughness for the main channel was close to the floodplain roughness. The Wallingford data were compared with results obtained using the divided channel methods, diagonal method and the results from the three CCDG 1-D models. Figure 5.12 shows the comparison of the stage-discharge curves with the observed data while Figure 5.13 and Table 5.2 show the percentage error associated with discharge estimation for the CCDG models and the divided channel methods. The comparison of the diagonal method and the CCDG 1-D model are shown in Figures 5.14 and 5.15.

5.2.8 Discussion of Results

As shown in Table 5.2, the divided channel method 2, underestimates the discharge by 69.3 % at low stage and by 41.6 % at high stage. The divided channel method 3, diagonal method and the CCDG 1-D models overestimate the discharge by about the same amount of about 10% to 15 %. Generally the results follow the same trend as the previous test, where the diagonal method and the CCDG 1-D models gave better discharge prediction than the divided methods.

5.2.9 Comparison of Methods

The steady state tests have shown that the divided channel method 2 underestimates the discharge generally, while the other methods overestimated the discharge.

Although the CCDG 1-D model and the diagonal method showed better prediction of the stage-discharge relationship over the traditional methods, it also showed that the two methods are also not entirely adequate. The inclusion of apparent shear in a divided method (in this case CCDG 1-D), improved the prediction of the discharge but this method only performed as well as the diagonal method. The tests also showed that increased roughness in the floodplain reduced the accuracy of these methods in estimating the compound channel discharge whether an apparent shear was included or lines of zero shear were adopted. Overall, the CCDG 1-D models improved the discharge prediction by 8-12 % (at low depths) for smooth surfaces and about 20 % for rough surfaces over the traditional methods. These results generally show that the CCDG 1-D model is as good as any existing method used for handling compound channel flow.

In flood studies, the issue of overestimation of discharge may not be as important as underestimation of stage. This is because high stages can cause excessive damage to human developments during flooding. In this respect, it is therefore important not to underestimate stage when modelling it. The results observed from the CCDG 1-D model showed that for a given discharge the stage was generally under predicted, and a freeboard should be considered when using this model for designing purposes. However it is believed that better prediction of stage-discharge relationships may be achieved if improved relations representing flow mechanism between the floodplain and main channel are found.

5.3 Verification of the Proposed Formulation for Unsteady Flow Tests

5.3.1 Introduction

In this section, the intention of the study was to examine the ability of the proposed model to simulate unsteady flow in compound channels. The tests involved comparisons to laboratory data. In these tests, θ of 0.5 and a tolerance of 10^{-5} were used.

The effect of momentum transfer and mass flow exchange on unsteady flow were investigated. Sensitivity analyses were conducted to determined whether the mass flow exchange was sensitive to the coefficients of discharge used. Also in this section the performance of DWOPER and ONE-D models (CSCE, 1993) in simulating the tested data was compared to the CCDG 1-D model.

5.3.2 Treske's Experimental Data

Treske's unsteady flow data (1980) was first reported by the CSCE task committee on river models (1987). The data was collected in an outdoor laboratory facility by Treske in Germany. Treske conducted experiments on three different types of compound channel configurations, namely, straight channel with and without floodplains, meandering channels with and without floodplains, and straight channels with lateral inflow. In this model the data collected on straight channels were used to test the model.

The cross-section of the straight channel is shown in Figure 5.16. The main channel had a bedwidth of 1.25 m and was 0.39 m in depth. The left floodplain was 3.0 m wide and the right floodplain was 1.5 m wide. The working length of the channel was 210 m. The bed slope was 0.019 % and the Manning roughness coefficient for both the main channel and floodplains was estimated to be 0.012 (Treske, 1980). The downstream measurement station was 210 m from the upstream measurement station. At the measurement stations, both the flow rate and stage were measured with time. Figures 5.17 and 5.18 show the depth and discharge inflow and outflow hydrographs of Treske's tenth experiment (used in this comparison)

5.3.3 CCDG 1-D Computational Model

The total length of the channel was divided into 14 elements, each of 15 m length. Based on the travel time of the observed peak discharge to reach downstream, it was established that the wave speed was about 1.17 m/s. Therefore a time step of 6 seconds was used in the simulation so that the Courant number was less than 0.5. This is because the CDG scheme was found to achieve excellent phase accuracy while successfully damping the shortest wavelengths when the Courant number was less than 0.5 (Hicks and Steffler, 1992). The model used an observed discharge hydrograph as the boundary condition upstream and a stage hydrograph as the boundary condition at the downstream boundary. The observed upstream discharge had to be split into three discharges using the Manning formula to distribute the main channel and the floodplains discharges. A Mannings n of 0.012 was used for both the main and floodplain sections. The model predictions such as the flow depth at the upstream and the flow rate at the downstream section were compared with the measured data. The addition of the momentum transfer terms were examined to see how the routing of the flow was affected. In this study unless otherwise stated, the term model discharge will refer to the total compound discharge. Other discharges that will be referred to are the main channel and floodplain discharges.

5.3.4 Sensitivity Analysis on the Coefficient of Discharge

The flow exchange between the floodplain and the main channel was linked through a long or broad crested side weir for the flow into the floodplain and a side weir of zero height for inflow into the channel. Because of certain factors, such as spilling of flow at an angle and effect of submergence, a range of coefficients was adopted as stated earlier in section 2.5.3. For flow from the main channel into the floodplain, C_f values from 1.45 to 1.90 were considered while the discharge coefficients associated with inflow to the main channel were taken from 0.45 to 0.64. Also the new method (shome, 1995) outlined in section 2.5.6 was tested to see how it performed against other methods for estimating lateral outflow. Figures 5.19, 5.20 and 5.21 show the effect of different lateral outflow discharge coefficients on total discharge when the inflow discharge coefficient is kept at 0.45. The two figures show that, the effect of variance is very minimal. Although a high coefficient of discharge, means more lateral discharge, the effect of submergence may have been significant.

The new method also performed equally well. Varying the inflow discharge coefficients in the floodplain, the solution also showed no sensitivity. Figures 5.22 and 5.23 showed a slight effect around 160 minutes for a high value of $C_d = 0.64$. Figure 5.24 also show that different coefficients of inflow discharge have no effect on the floodplain total discharge. Therefore for all other tests, a C_l value of 1.45 is adopted for the lateral outflow and C_d value of 0.45 for the lateral inflow.

5.3.5 Discussion of Results

The computed upstream depth and downstream discharge were compared to the observed data at these two boundaries. Treske's data werelimited in that there are no intermediate points at which to compare results. All the CCDG 1-D models generally reproduced the measured upstream depth very well as shown Figure 5.25. The comparison of observed discharge and computed discharge at the downstream boundary are shown in Figures 5.26. Here the agreement of the discharge hydrographs was good except for the fact that, the wave was slightly out of phase with the observed wave. This was probably caused by the poor representation of the upstream boundary condition since the distribution of flow between the channel and the floodplain had to be estimated. The CCDG 1-D (P-T) overestimated the peak discharge by about 4.6 %. The model peak discharge was 0.412 m^3/s while the observed was 0.394 m^3/s. The CCDG 1-D (P-T) model was not affected by the transition associated with the beginning of the inundation of the floodplain.

Treske's data were also compared with the CCDG 1-D model results when the momentum transfer terms were excluded from the model but the lateral inflow and outflow were allowed. As shown in Figure 5.26, the model without the momentum transfer terms generated a peak discharge slightly higher than the CCDG 1-D model with momentum transfer terms. The computed maximum peak discharge at the downstream boundary was 0.427 m^3/s while the observed was 0.394 m^3/s. The relative error was 8.4 % which was higher than that obtained from any of the apparent shear stress models. However, since the CCDG 1-D models also overestimated the peak discharge by about 4% to 7%, it can be said that including the apparent shear had a marginal effect on unsteady flow.

5.3.6 Selected Dynamic Models

A CSCE Task Force Committee on River Models (CSCE, 1993) evaluated some dynamic models to see how they perform in modelling rivers with floodplains. They tested these models with the same Treske data described in the previous section and some of their results have been used to compare to the results obtained from the CCDG 1-D model. Here, the results from DWOPER and ONE-D finite difference models were compared to the CCDG 1-D model results. These models were chosen because, DWOPER is widely used in United States of America and ONE-D is used widely in Canada.

5.3.6.1.1 DWOPER Computational Model

In the CSCE study, the channel was subdivided into 14 reaches, each of 15 m length. A time step of 1 minute (60 secs) and a θ weighting factor of 0.55 was used. The Manning *n* of 0.012 was modified because the DWOPER model was developed to simulate large rivers where flow depths are much smaller than the widths and the wetted perimeter (P) is approximated by the top width (T_w). To compensate for the under-estimation of the wetted perimeter (P), Mannings roughness n was modified by:

$$n_{eq} = n\left(\frac{P}{T_w}\right)^{2/3} \qquad [5.2]$$

The observed upstream discharge was input as the upstream boundary condition and a rating curve based on observed flows and stage was used as the downstream boundary condition. A table of Mannings n as a function of depth (H) was also input to DWOPER. The transition from the main channel to the floodplain was made to change gradually.

Figures 5.27 and 5.28 show the DWOPER results. The upstream depth was generally simulated well as shown by Figure 5.27. The observed discharge at the downstream was satisfactorily predicted by the DWOPER model. The peak discharge computed by the model at the downstream boundary was found to be 0.379 m^3/s which was an under-estimation of 3.8 %.

5.3.6.2 ONE-D Computational Model

In the CSCE study, the channel was again divided into reaches of 15 m for the ONE-D model, with a rating curve specified at the downstream end of the channel. A time step of 1 minute and a θ weighting factor of 0.55 was also used for this model. The observed discharge hydrograph was used as the upstream boundary condition. Mannings n was varied from 0.0116 to 0.0110 as the depth increased from the initial depth to the top of the main channel. The main channel width was restricted to 2 m for the first 0.025 m above the start of the floodplain. The ONE-D model was also run with ficitious lateral withdrawal to partially account for the difference in volume between the observed inflow and outflow. The transition from the main channel to the floodplain was also made to change gradually like the DWOPER model.

The results obtained using this model are also shown in Figures 5.27 and 5.28. Figure 5.27 shows that, like other models, there was a good agreement between the computed and observed depth at the upstream station. The computed discharge at the downstream boundary was also well predicted. The model computed peak discharge was 0.394 m^3/s which was the same as observed.

5.3.7 Comparison of Computational Models

The results show that the CCDG 1-D model generally handled the transition of flow from the main channel to the floodplain very well although its performance was only marginaly improved when the momentum transfer terms were included in the model. The CCDG 1-D (P-T) model performed best among all the CCDG 1-D models. It only overestimated the peak discharge by 4.6 % at the downstream boundary. The CCDG 1-D (W-M) and The CCDG 1-D (C) models overestimated the peak discharges by 6.8 % and 5.3 %, respectively, at the downstream boundary.

The model without the transfer terms overestimated the peak discharge by about 8.4 % at the downstream boundaries. The results showed that the inclusion of the apparent shear stress models slightly increased the prediction capability of the observed discharges in compound channels.

The performance of the CCDG 1-D model in comparison to the DWOPER and ONE-D model was fairly good. The peak discharge at the downstream station was slightly over predicted by the CCDG 1-D model as compared to the ONE-D model. The reason for this difference may have been the poor representation of the upstream boundary condition. It is also noted that the DWOPER and ONE-D models had to be specifically adjusted and the channel modified to be able to model this flow. With further improvement, the CCDG 1-D model can perform equally well in predicting compound channel flows.

5.4.0 Illustration of Model Performance for Practical Situations.

5.4.1 Simulation of Flow through a Dike Breach into the Floodplain.

TheTreske (1980) channel and inflows were used to set up the hypothetical case of flow through a breached dike. A dike of height 0.3 m was placed on the floodplain as an extension of the channel side walls from the upstream boundary to the downstream boundary. Beginning at 120 m from the upstream boundary, a breach was set such that the height of the dike was reduced to 0.06 m. The length of the breach was 30 m, which was exactly equal to two element lengths. Treske's inflow hydrograph was used as the upstream boundary condition. For illustrative purposes, the downstream boundary was extended to 420 m from the upstream boundary and a constant depth of 0.215 m was used as the downstream boundary condition. The same time step and upwinding coefficient adopted for other tests was used. The flow was routed using the CCDG 1-D model for 219 minutes and results at 150 m the downstream from upstream boundary were plotted and compared to the case when the breach was not present. Figures 5.30 to 5. 37 show these results.

5.4.2 Discussion of Results

The CCDG 1-D results from the breaching of the dike were compared to the case when the dike was not breached. As shown in Figure 5.30, the main channel discharge at 150 m downstream drops from 0.412 cms to 0.402 m^3/s when the dike was breached. This represented a reduction in discharge of 2.4 %. Figure 5.31 show that the depth along the entire length of the channel was lowered when the dike breached. Upstream of the breached dike, the depth reduced by about 2 %, but downstream of the breached dike the depth reduced by about 1.2%. The discharge that flowed into the left floodplain flowed in the upstream and downstream directions as shown in Figure 5.32. Figure 5.33 shows also that the depth hydrograph in the left floodplain increased in the upstream and downstream directions. The main channel discharge and depth distributions with time are shown in Figures 5.34 and 5.35. These Figures show that the peak discharge and peak depth are also reduced although by small percentages. Figures 5.36 and 5.37 show that there is some distribution of depth and discharge in the floodplain as opposed to the case of no flow associated with no breach conditions.

Although the amount of discharge passed into the floodplain was minimal in this case, the use of computational models like CCDG 1-D model can be useful in prediction of how the wave behaves in the main channel and the floodplain(s) when a dike breaches in case of extreme floods. Specifically, it helps in assessing the effects of flood waters passing through breached dikes on unprotected areas both upstream (due to potential backwater effects) and downstream (due to the loss of flood storage area, causing flood peak attenuation). It is also pointed out that, although this model can model breached dikes, it may not exactly reproduce the kind of situations observed in nature. This is because the flow may not flow far into the floodplain and the observed depth in the floodplain may not be deep as observed in the CCDG 1-D model, when floodplain roughness is high.

5.4.3 Simulating Steady Flow in a Meandering Compound Channel

To test the CCDG 1-D model on simulating the stage-discharge relationship for steady state in a meandering compound channel, Smith's experimental set up (1978) was tried. Figure 5.38 shows the plan view, dimensions and cross-section of the meandering compound channel Smith used. His experimental set up consisted of a valley of slope 0.001 and a sinusoid meandering channel of slope 0.00085 laid on a bed flume 24 m long and 1.22 m wide. The channel had a bottom width of 0.122 m, top width of 0.27 m and a depth of 0.076 m with side slope ratio of 1V:1H.

In simulating this experiment, the cross-section representing the length of an element was adopted such that the cross-sections are perpendicular to the downstream direction of flow and the main channel width varies gradually along the channel in the longitudinal direction. Because of a lack of information, the compound channel entrance and exit lengths were approximated as 5.0 m and the left and right floodplain widths were taken as shown in Figure 5.38. The valley slope was adopted as the slope of the left and right floodplains and the floodplains widths and distances between the cross-sections were determined based on the dimensions in the original figures. For the simulation of flow in a compound meandering channel, the effect of apparent shear stress was not included in the CCDG 1-D model because available relations are only valid for straight compound channels given that the flow interaction methods are entirely different in this case. The roughness in the main channel used by Smith (1978) varied from 0.0138 to 0.0127 as the stage increased while for the valley section of the compound channel, the Mannings roughness varied from 0.0117 to 0.0114. In this study, an average Mannings roughness of 0. 013 was used for both the main channel and the floodplain. The boundary conditions were adopted as area for both the upstream and downstream boundaries. The CCDG 1-D model simulated results were compared to Smith's observed stage and discharge.

5.4.4 Discussion of Results

The comparison of the CCDG 1-D model simulation results at end of the second wavelength of the meandering channel and observed stage-discharge results are shown in Figure 5.39. The simulation results shows a good agreement with the observed data, although the CCDG 1-D model overestimates the observed discharge by about 12.5 % at the low stages and 6.1 % at the high stages. The longitudinal distribution of discharge shown in Figure 5.40 shows some oscillations in the left and right floodplain. This is mainly due to the the fact that the width is always changing in the downstream direction. The total discharge, although expected to be constant, showed some small fluctuations.

The results obtained from this test on routing flow in meandering compound channels using the CCDG 1-D model showed that the model requires further refining in order to model meandering channels. In a general sense, the results showed that the CCDG 1-D model could be a good tool in simulating steady flow in meandering channels.

Table 5.1 Comparison of computed discharge error based on Prinos-Townsend Experiment (1984) for a main channel roughness n=0.011.

Floodplain Mannings n	Stage m	Divided channel method 2 %	Divided channel method 3 %	Diagonal method %	CCDG 1-D model: CCDG 1-D (P-T) %	CCDG 1-D model: CCDG 1-D (W-M) %	CCDG 1-D model: CCDG 1-D (C) %
0.011	0.112	-19.7	22.7	15.6	10.6	13.6	16.7
	0.132	-11.2	9.6	1.5	4.8	4.8	5.6
	0.152	-11.2	2.6	-3.5	0.5	0.0	0.5
0.014	0.112	-26.2	29.5	22.4	14.7	18.0	22.9
	0.132	-15.5	17.3	6.6	10.9	9.0	10.9
	0.152	-15.0	7.8	-14.1	3.3	1.7	2.2
0.022	0.112	-35.4	62.5	51.2	37.5	41.6	50.0
	0.132	-28.9	33.4	16.1	20.0	15.5	17.8
	0.152	-25.0	23.6	5.0	12.8	7.8	8.6

Table 5.2 Comparison of computed discharge error based on Wallingford Experiments (1992)
(n varies from 0.0098 to 0.0092 as the stage increases)

Stage m	Divided channel		Diagonal method	CCDG 1-D model		
	method 2 %	method 3 %	%	CCDG 1-D (P-T) %	CCDG 1-D (W-M) %	CCDG 1-D (C) %
0.159	-69.3	15.1	10.7	12.6	7.6	11.5
0.165	-63.3	17.7	11.3	15.6	9.4	13.0
0.176	-55.4	19.1	10.5	17.6	11.2	14.1
0.187	-51.2	16.6	7.5	15.5	10.3	12.2
0.199	-46.5	18.3	9.6	17.5	12.7	14.2
0.214	-44.8	14.9	7.4	14.3	10.6	11.6
0.250	-41.6	13.5	7.8	13.2	10.8	11.3

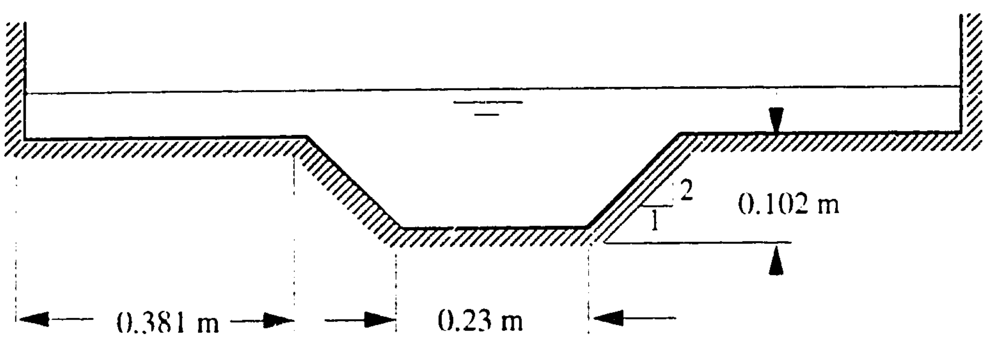


Figure 5.1 Cross-section of the Prinos-Townsend compound channel

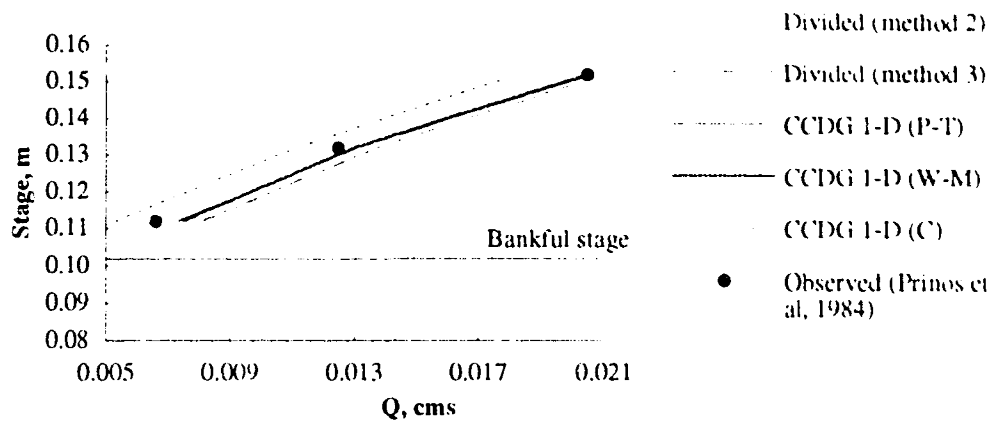


Figure 5.2 Stage-discharge relationship for observed discharge, divided channel methods and CCDG 1-D models (floodplain n=0.011, main channel n=0.011)

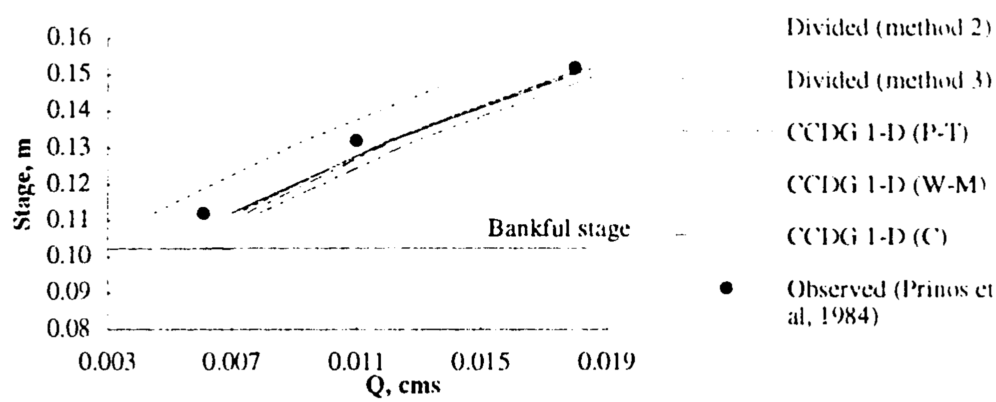


Figure 5.3 Stage-discharge relationships for observed discharge, divided channel methods and CCDG 1-D models (floodplain n=0.014, main channel n= 0.011)

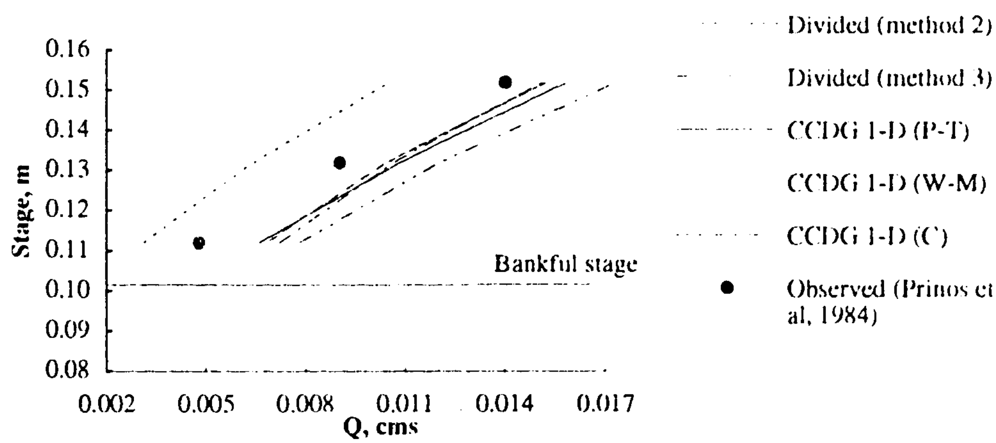


Figure 5.4 Stage-discharge relationships for observed discharge, divided channel methods and CCDG 1-D models (floodplain n=0.022, main channel n=0.011)

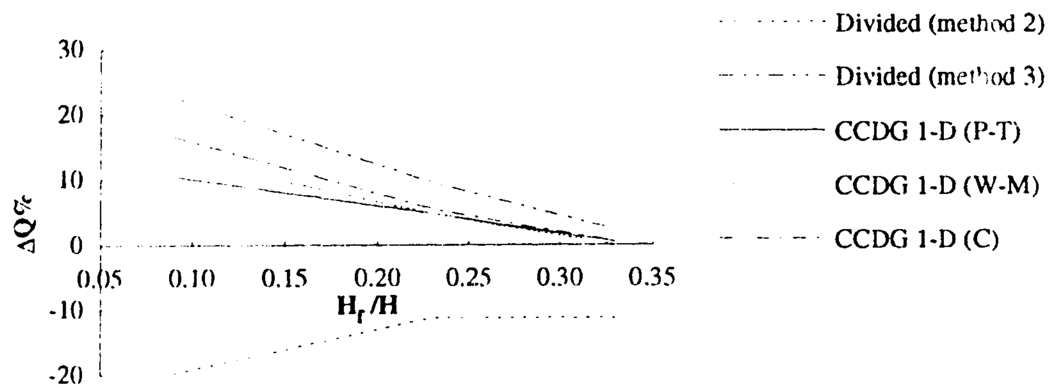


Figure 5.5 Percentage change in discharge versus relative depth for divided channel methods and CCDG 1-D models (floodplain and main channel has the same roughness of n=0.011)

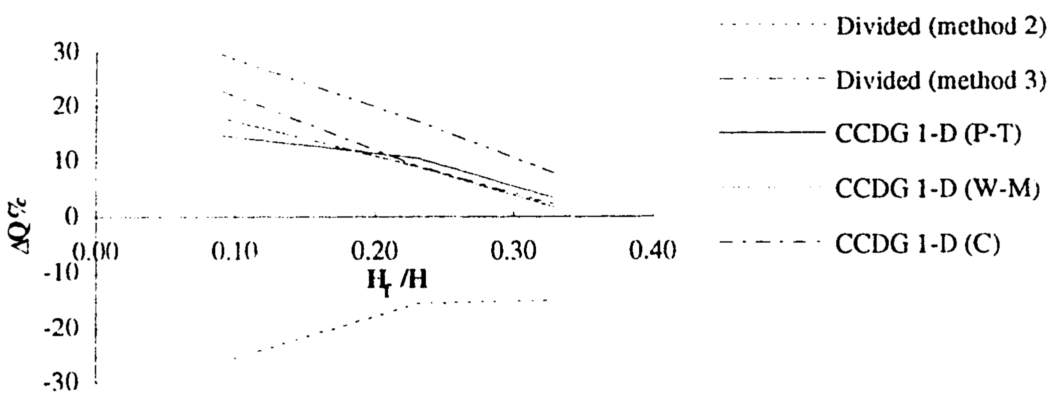


Figure 5.6 Percentage change in discharge versus relative depth for divided channel methods and CCDG 1-D models (floodplain n=0.014, main channel n=0.011)

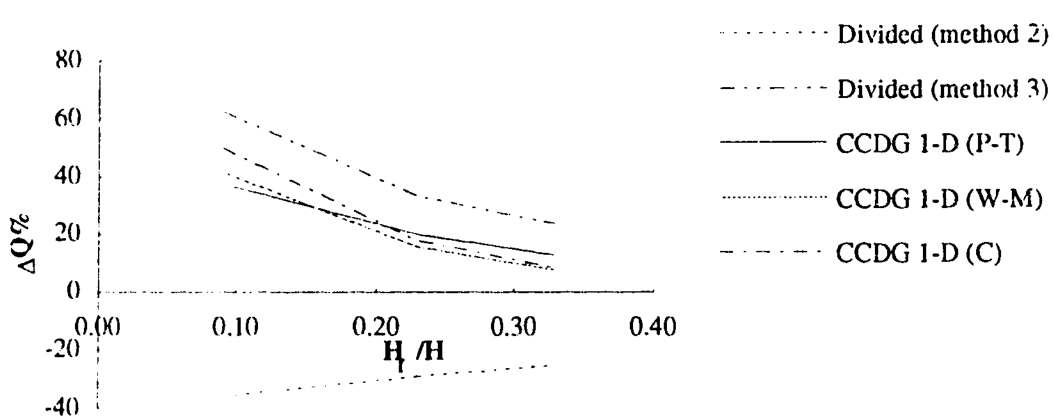


Figure 5.7 Percentage change in discharge versus relative depths for divided channel methods and CCDG 1-D models (floodplain n=0.022, main channel n=0.011)

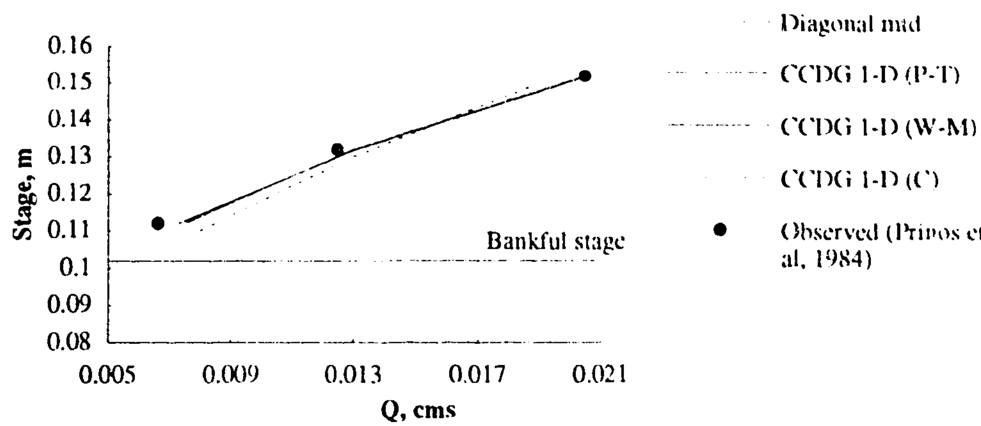


Figure 5.8 Stage-discharge relationship for observed discharge, diagonal channel method and CCDG 1-D models (floodplain n=0.011, main channel n=0.011)

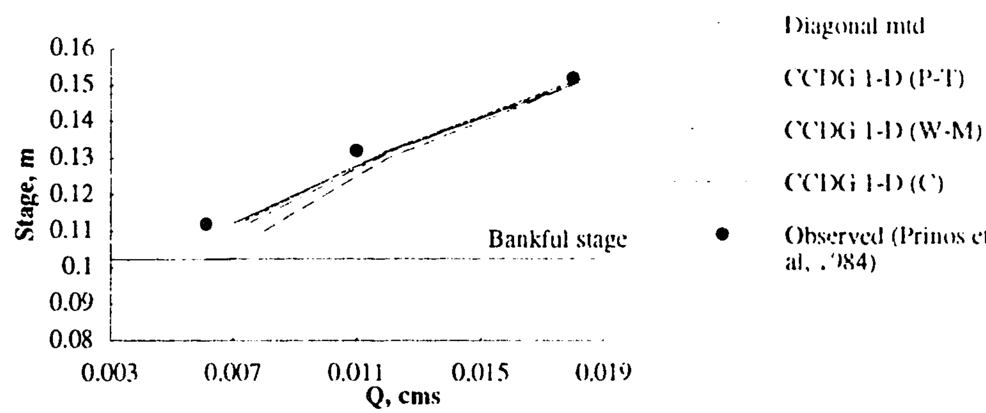


Figure 5.9 Stage-discharge relationships for observed discharge, diagonal channel method and CCDG 1-D models (floodplain n=0.014, main channel n= 0.011)

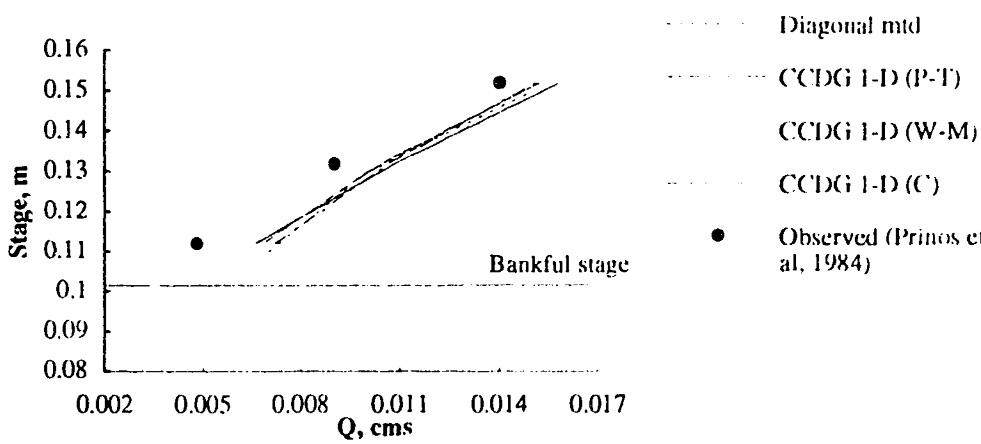


Figure 5.10 Stage-discharge relationships for observed discharge, diagonal channel method and CCDG 1-D models (floodplain n=0.022, main channel n=0.011)

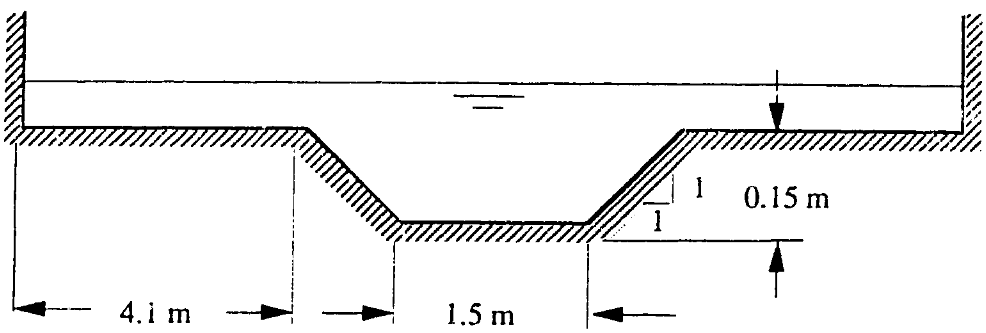


Figure 5.11 Cross-section of the Wallingford compound channel

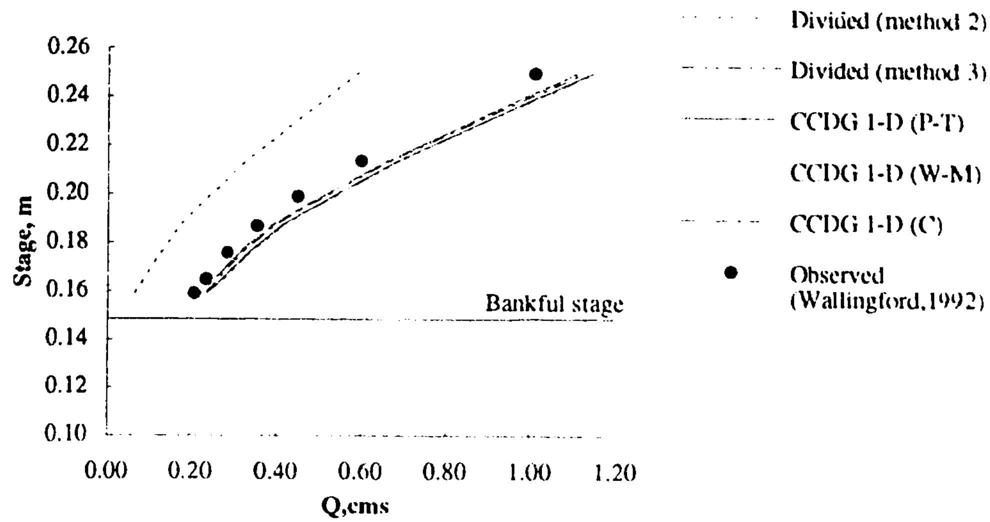


Figure 5.12 Stage-discharge relations for observed discharge, divided channel methods and CCDG 1-D models (n varies from 0.0098 to 0.0092 as the stage increases)

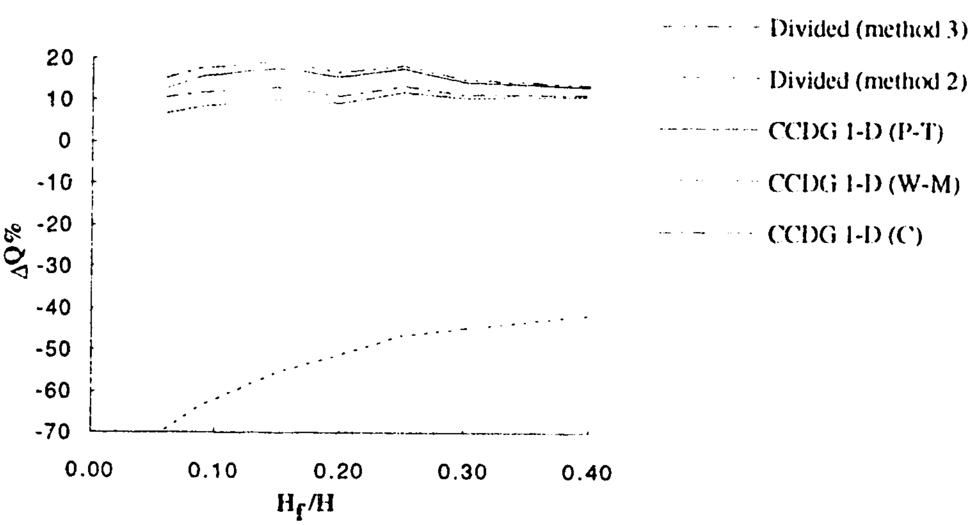


Figure 5.13 Percentage change in discharge versus relative depths for divided channel methods and CCDG 1-D models

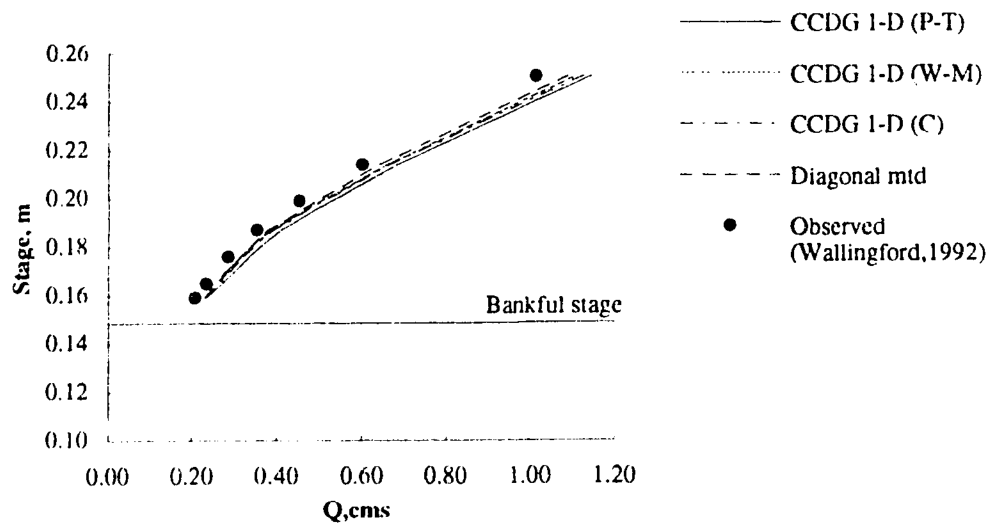


Figure 5.14 Stage-discharge relations for observed discharge, diagonal channel method and CCDG 1-D models
(n varies from 0.0098 to 0.0092 as the stage increases)

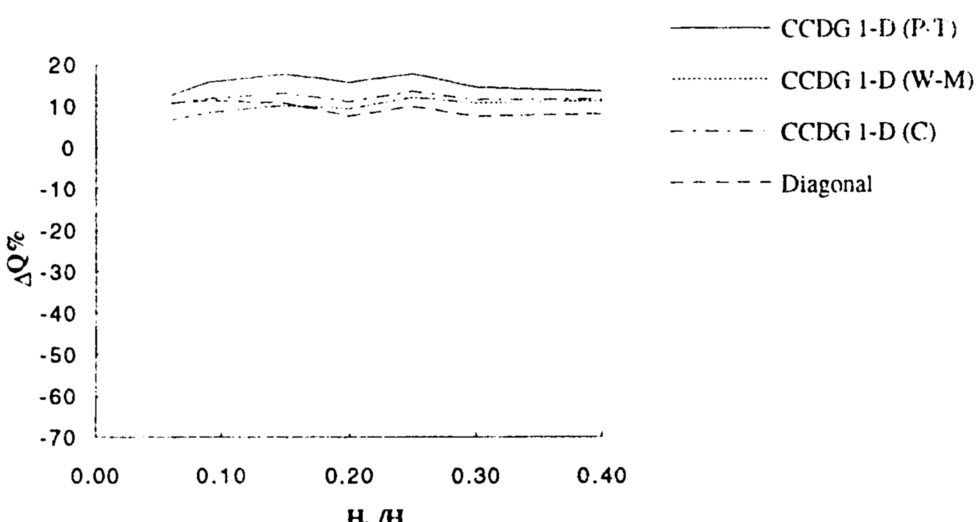


Figure 5.15 Percentage change in discharge versus relative depths for diagonal channel method and CCDG 1-D models

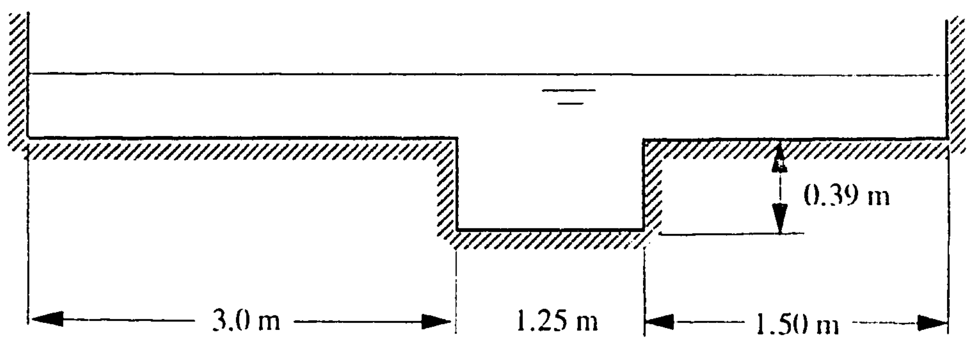


Figure 5.16 Cross-section of the Treske compound channel

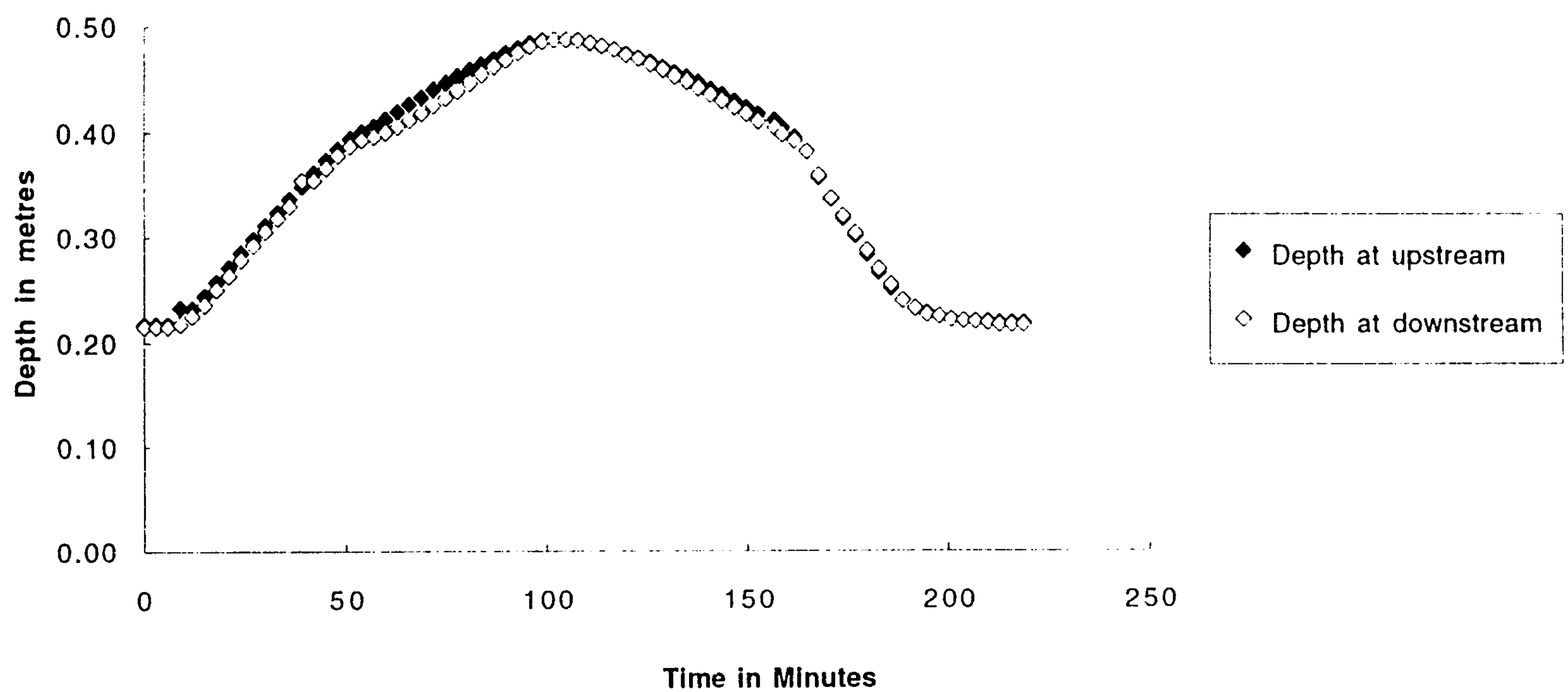


Figure 5.17 Treske's measured depths versus time at usptream and downstream boundaries

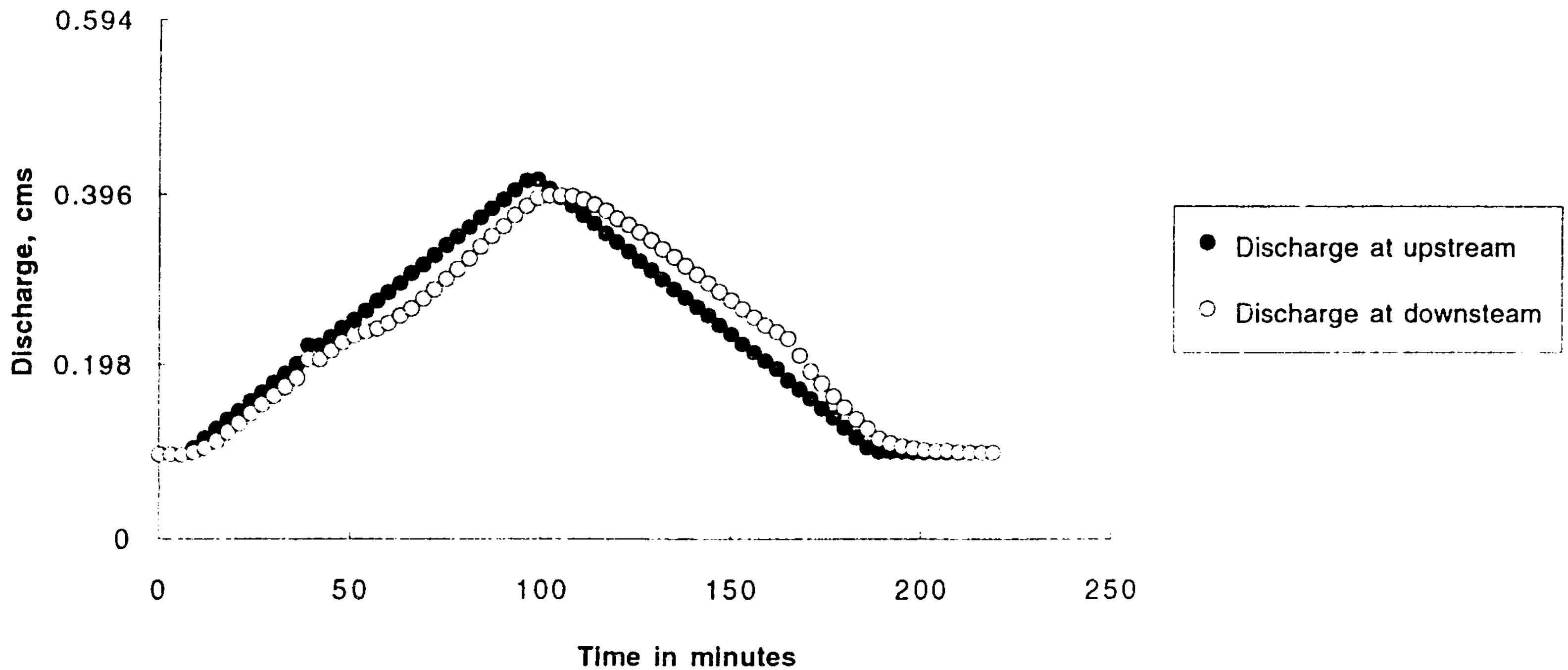


Figure 5.18 Treske's measured discharges versus time at the upstream and downstream boundaries

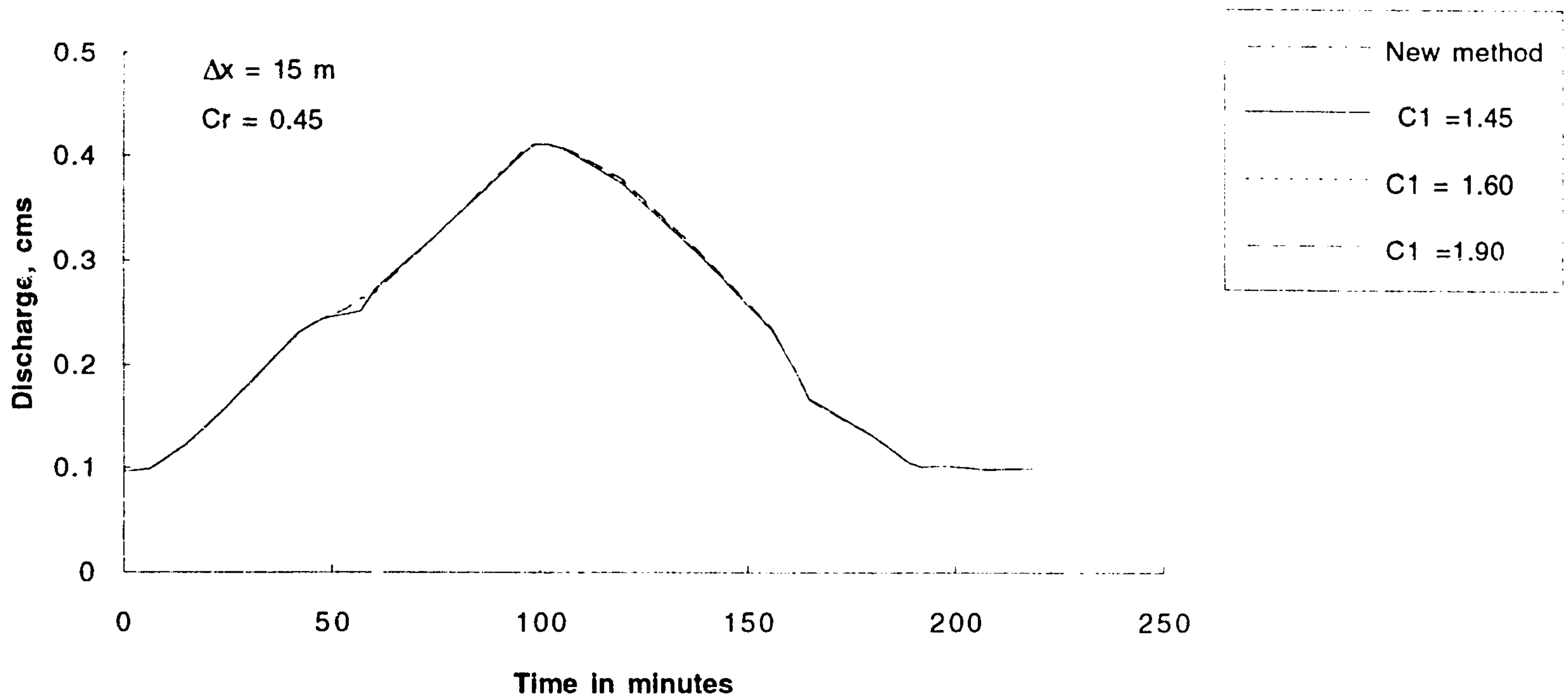


Figure 5.19 Effect of different lateral outflow discharge coefficients on on total discharge for CCDG 1-D (P-T) at the downstream boundary for an inflow discharge coefficient of 0.45

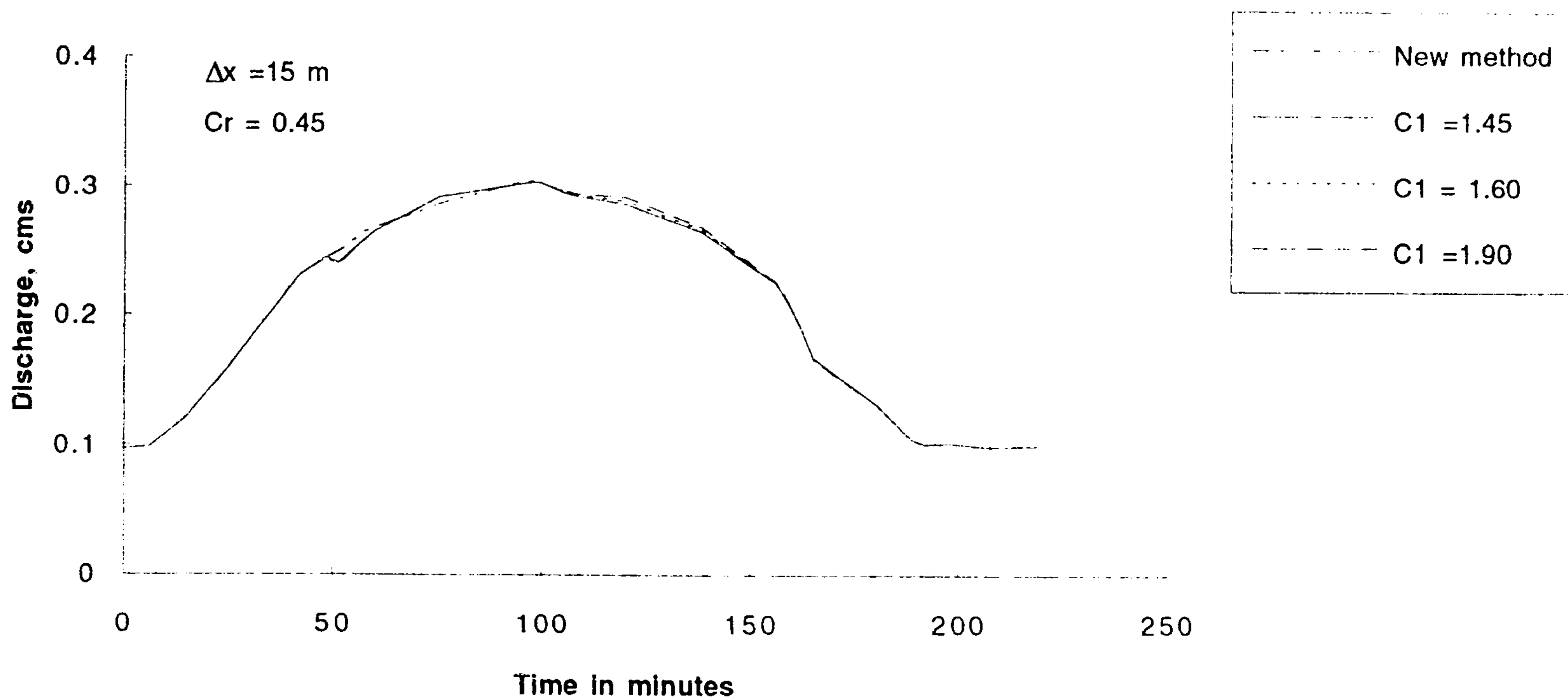


Figure 5.20 Effect of different lateral outflow discharge coefficients on the main channel flow for CCDG 1-D (P-T) at the downstream boundary for an inflow discharge coefficient of 0.45

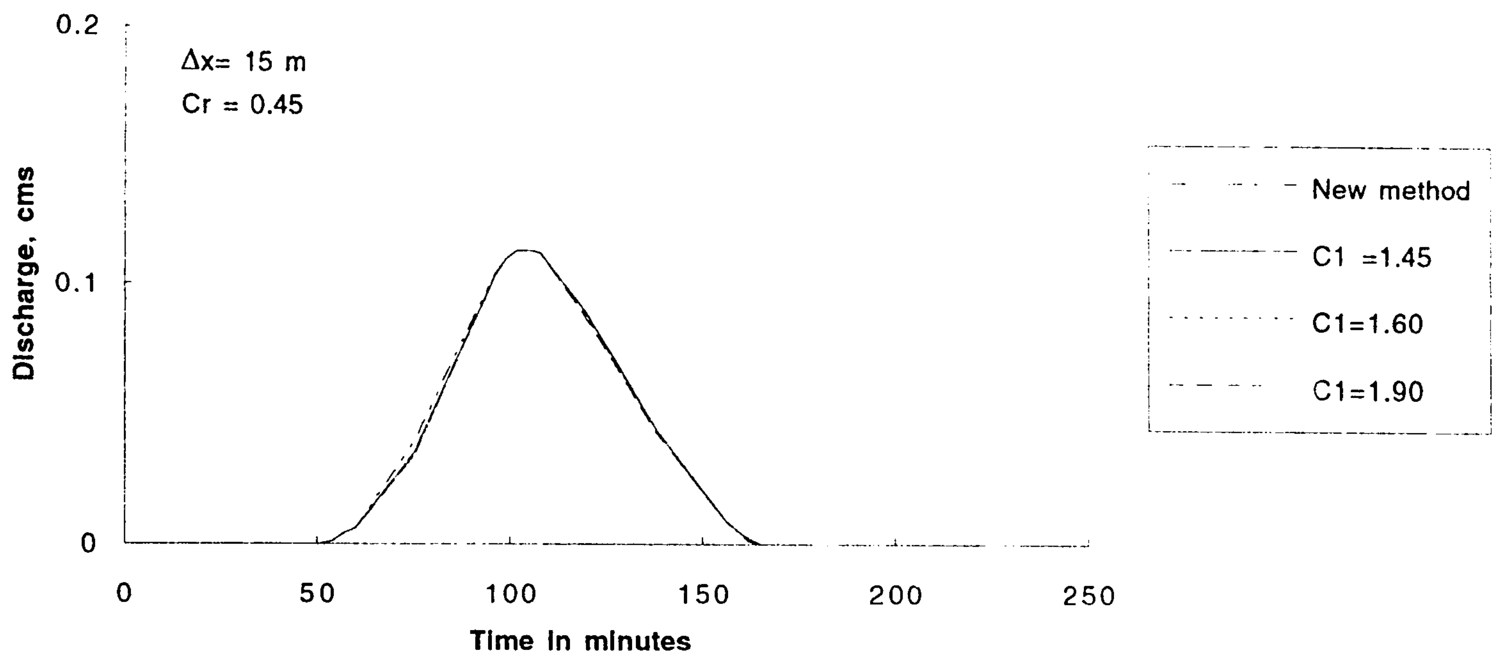


Figure 5.21 Effect of different outflow discharge coefficients on floodplain discharge for CCDG 1-D (P-T) at the downstream boundary for an inflow discharge coefficient of 0.45

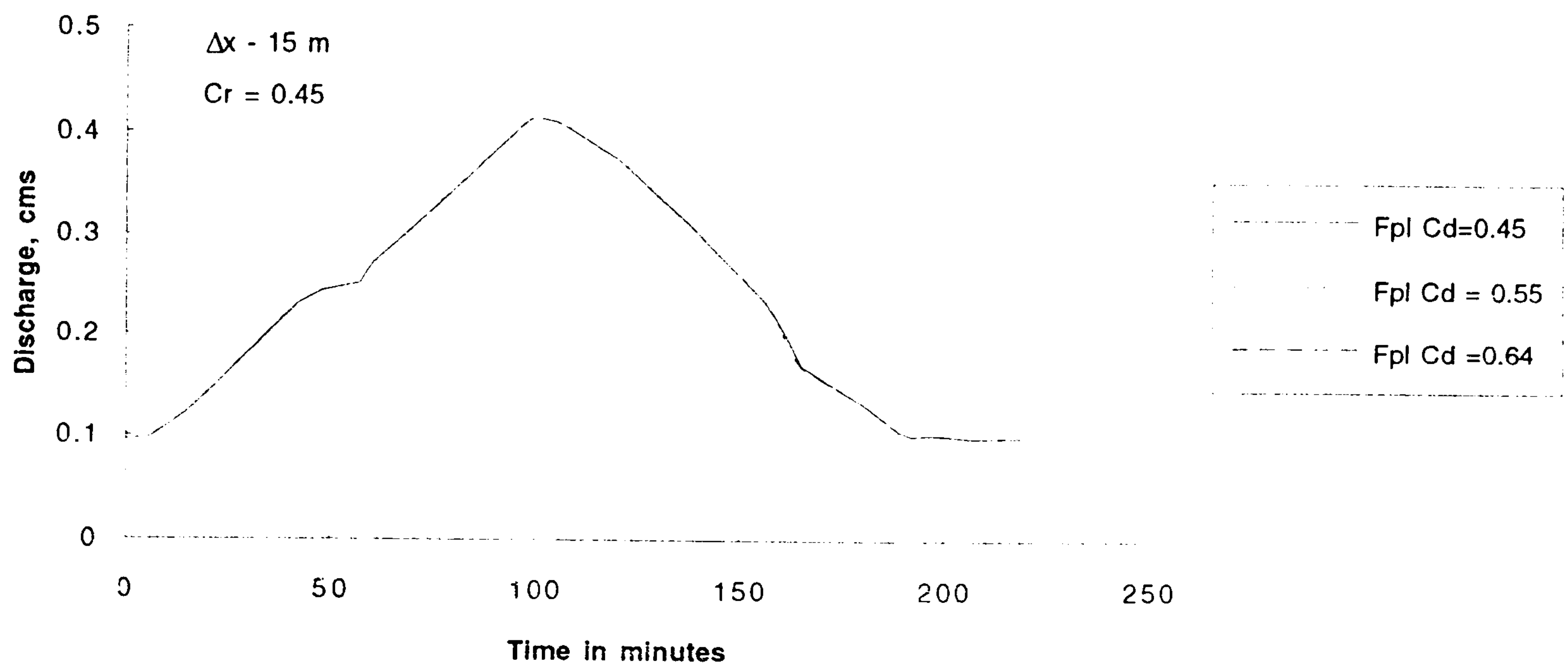


Figure 5.22 Effect of different inflow discharge coefficients on total discharge for CCDG 1-D (P-T) at the downstream boundary for an outflow discharge coefficient of 1.45

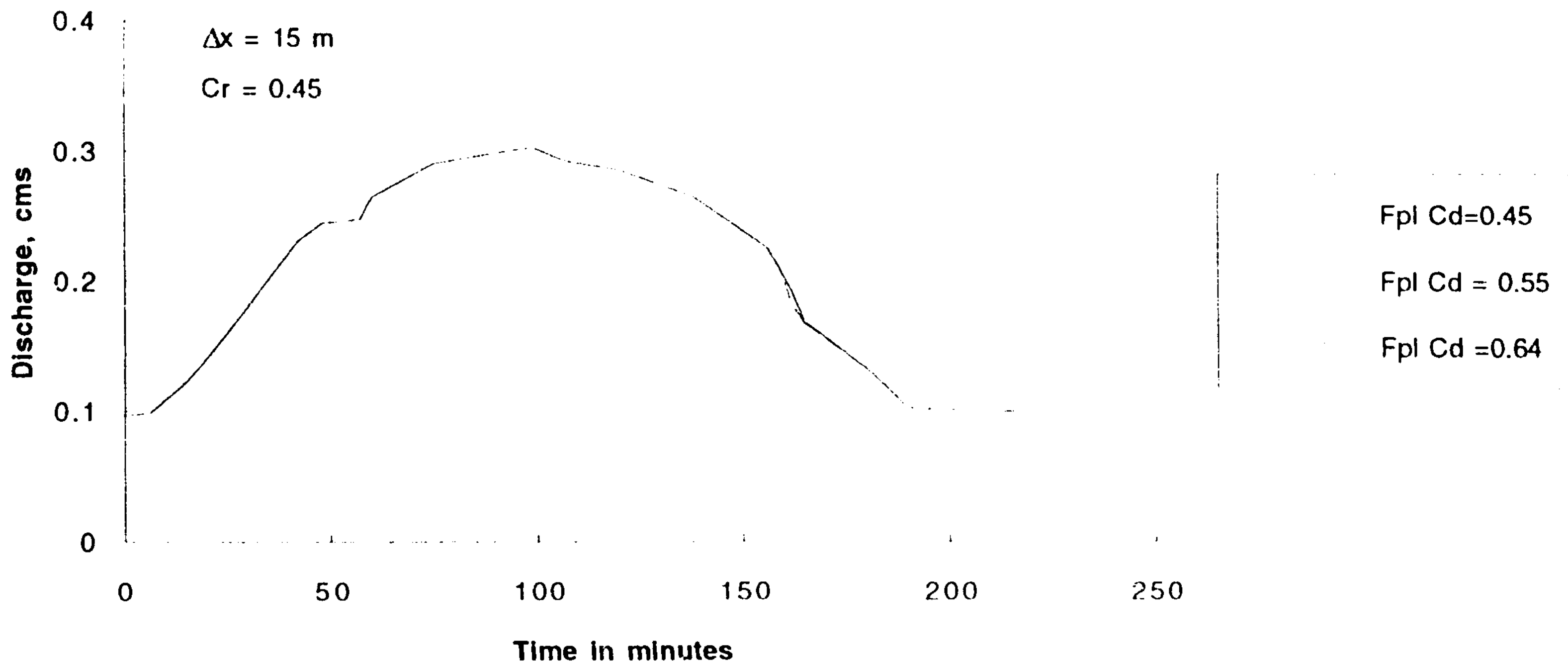


Figure 5.23 Effect of different inflow discharge coefficients on main channel discharge for CCDG 1-D (P-T) at the downstream boundary for an outflow discharge coefficient of 1.45

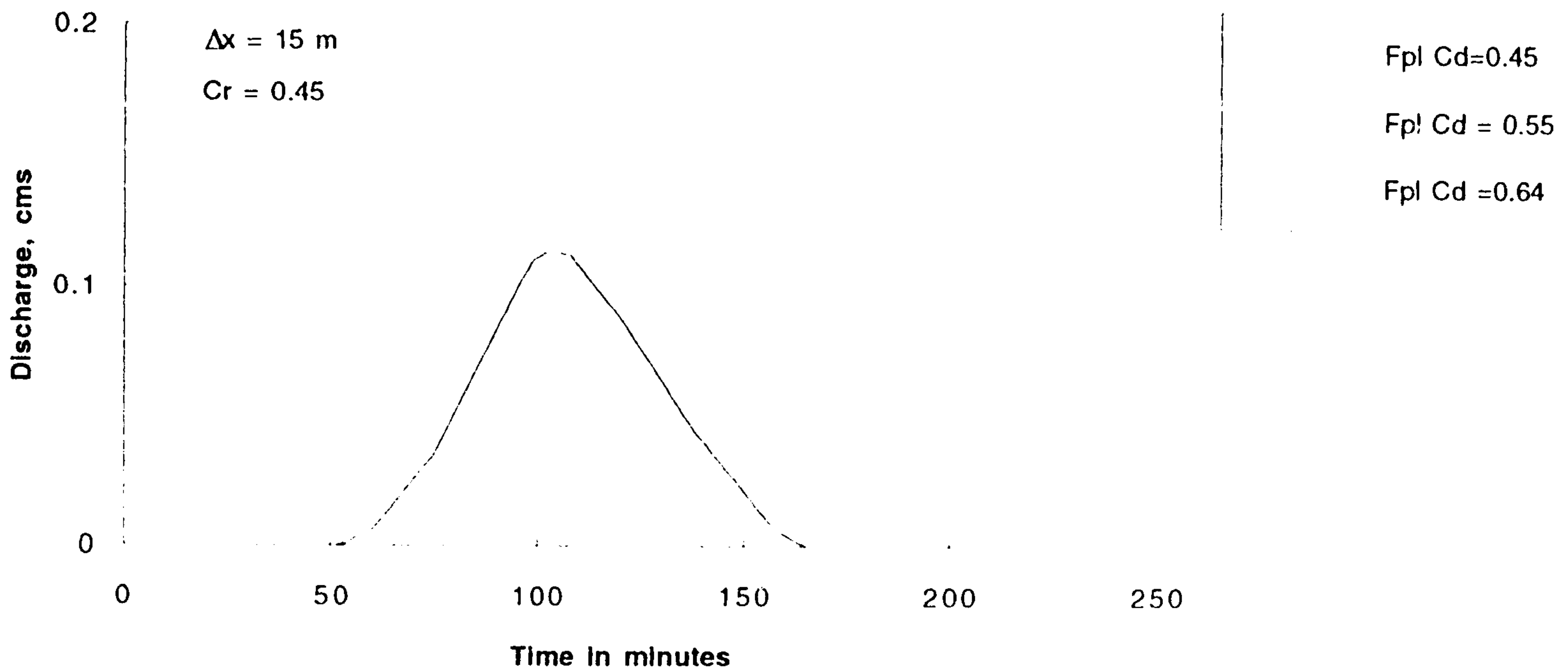


Figure 5.24 Effect of different inflow discharge coefficients on flooplain channel discharge for CCDG 1-D (P-T) at the downstream boundary for an outflow discharge coefficient of 1.45

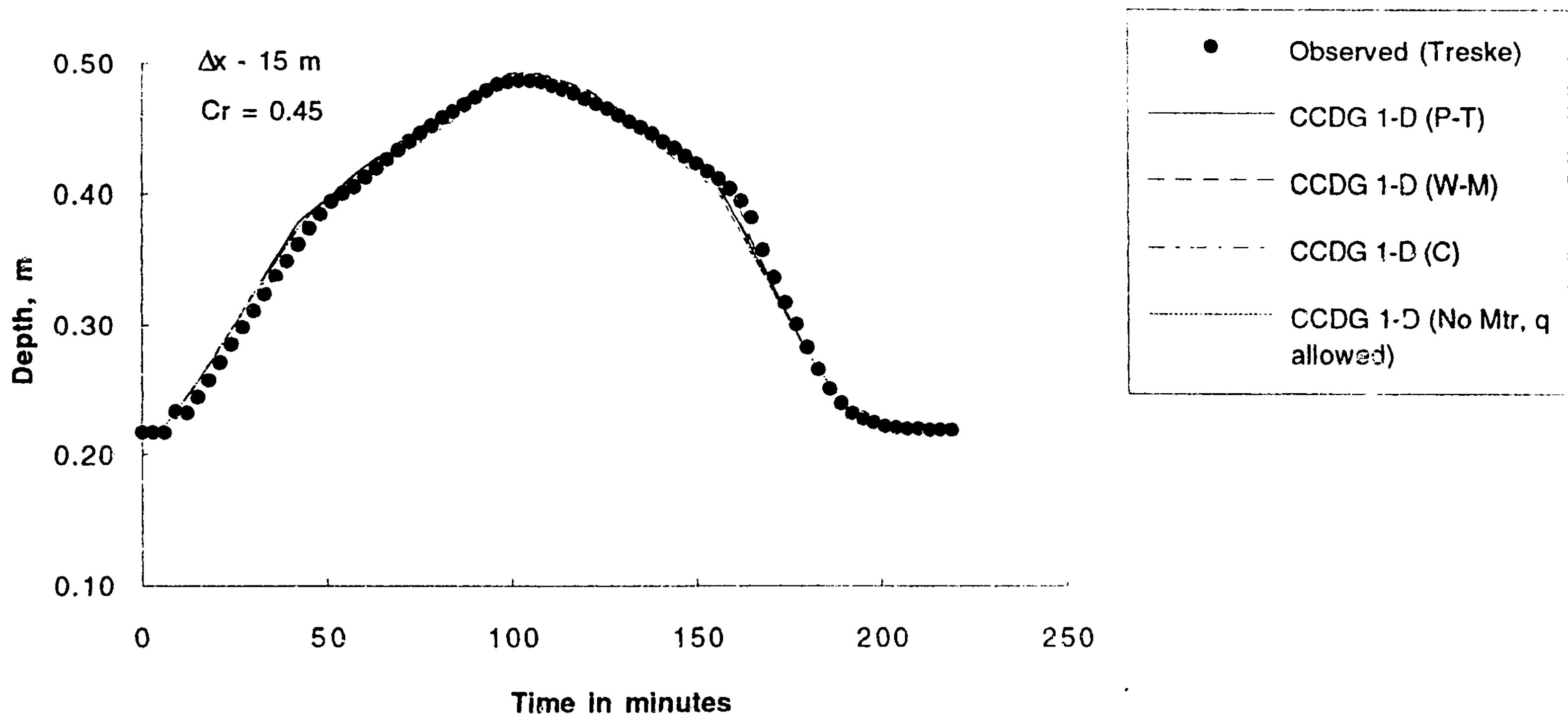


Figure 5.25 Comparison of observed depth and CCDG 1-D model results at the upstream boundary

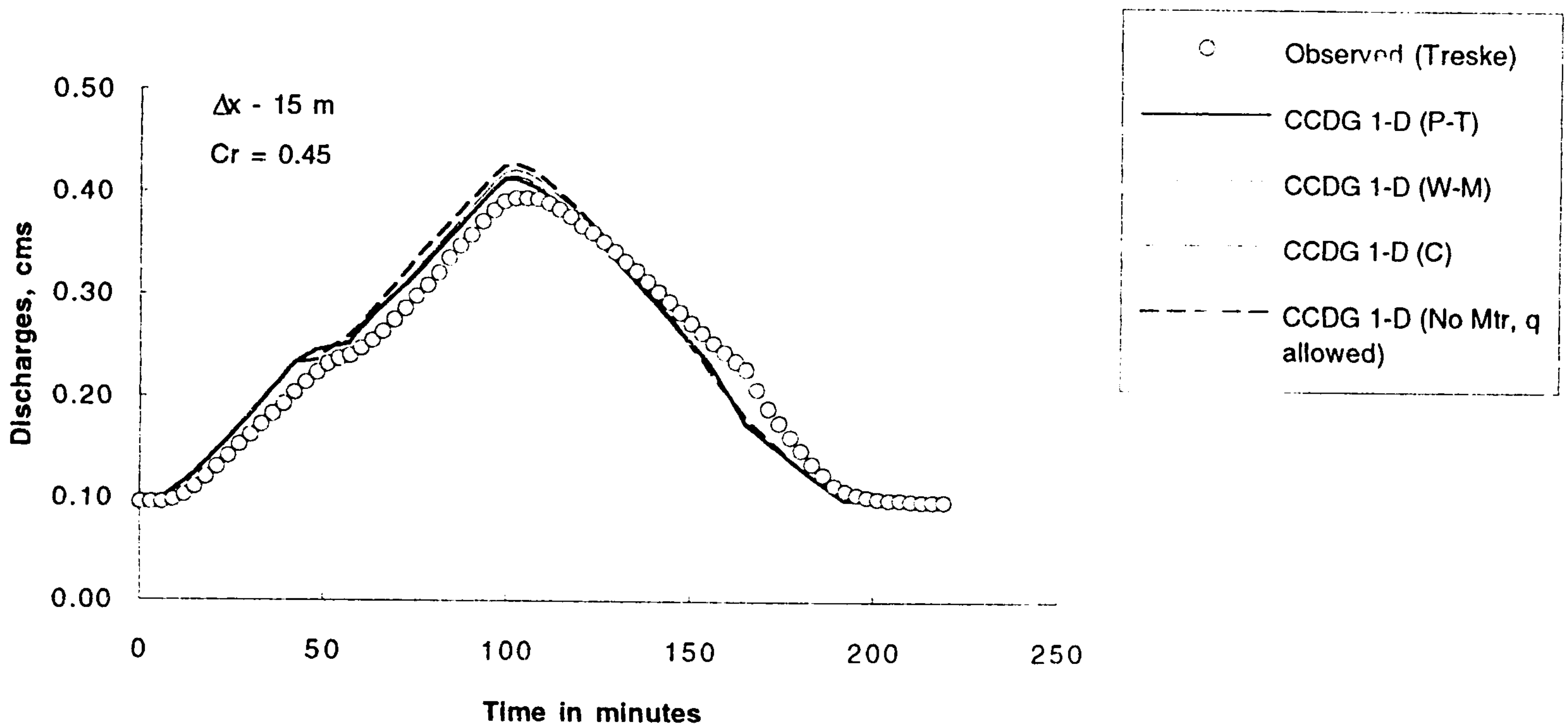


Figure 5.26 Comparison of observed discharge and CCDG 1-D models results at the downstream boundary

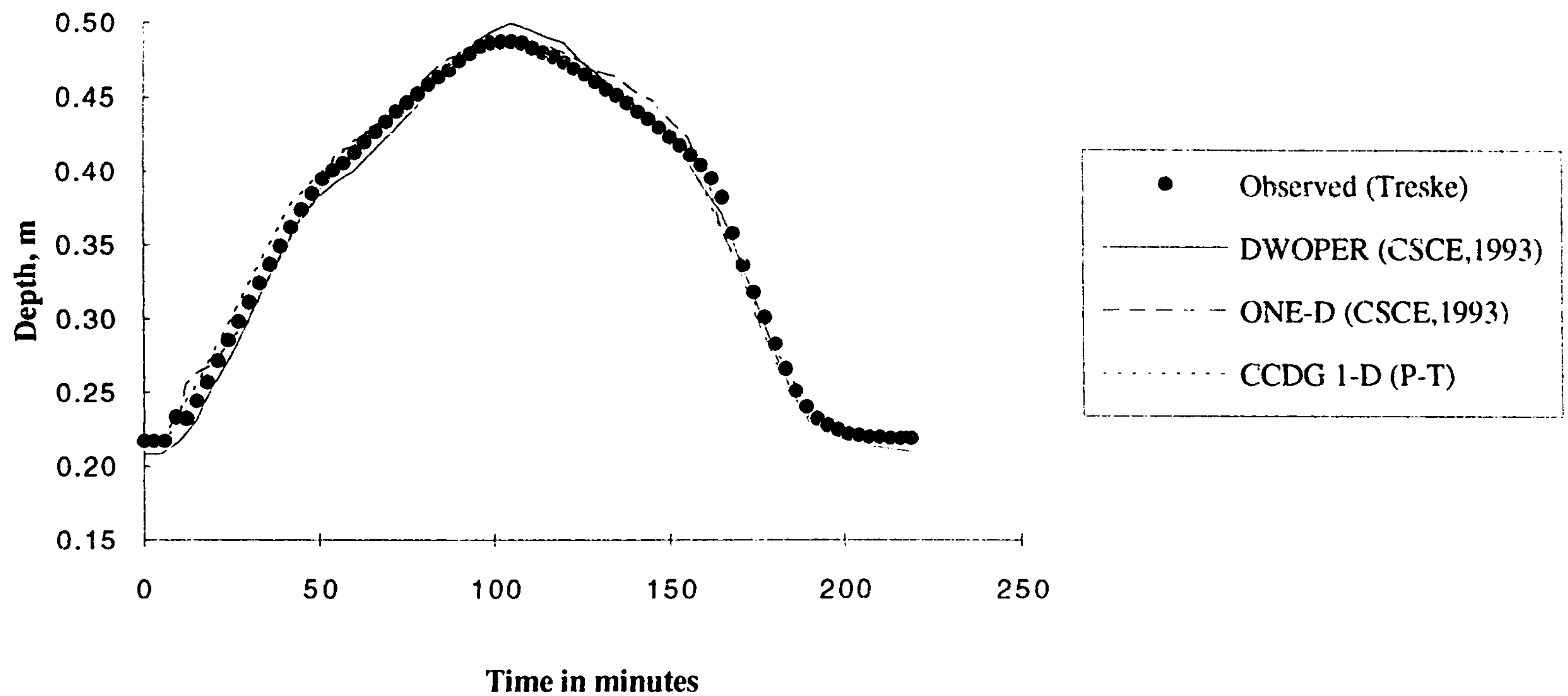


Figure 5.27 Comparison of observed depth and results from simulation models

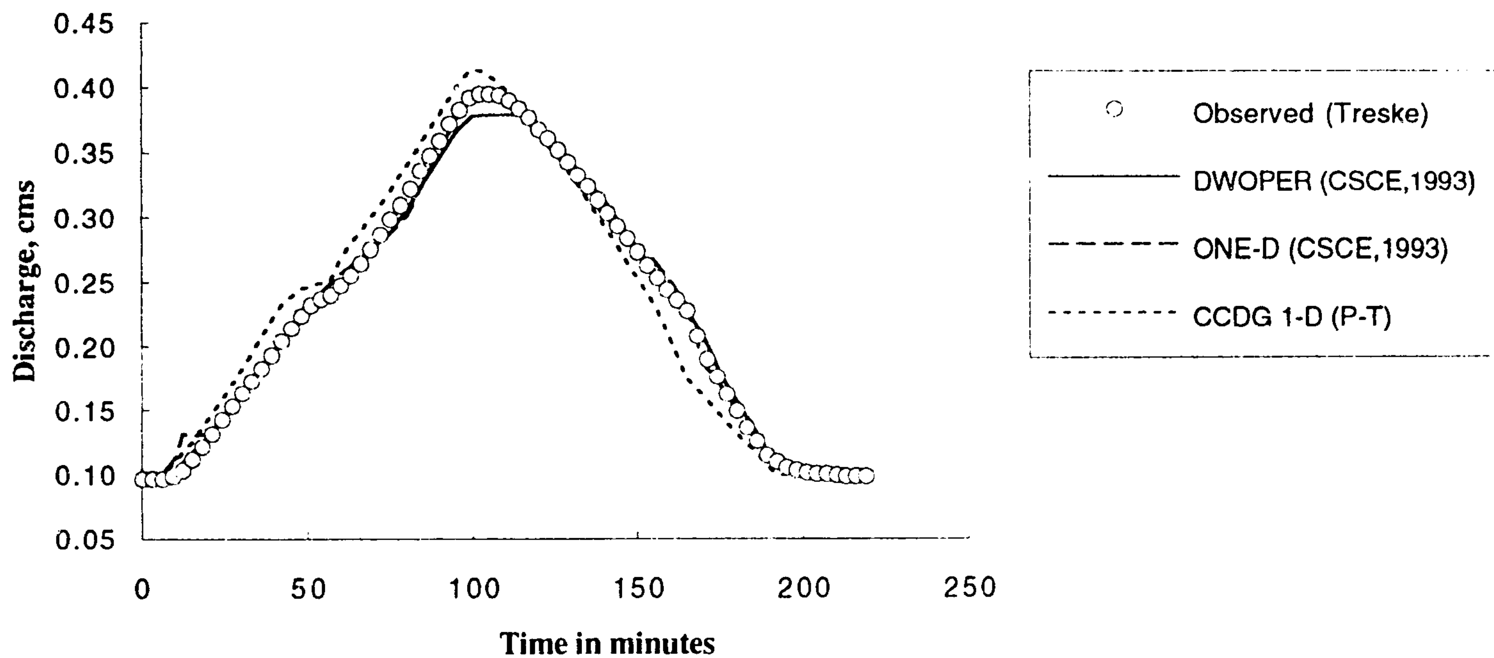


Figure 5.28 Comparison of observed discharge and results from simulation models at the downstream boundary

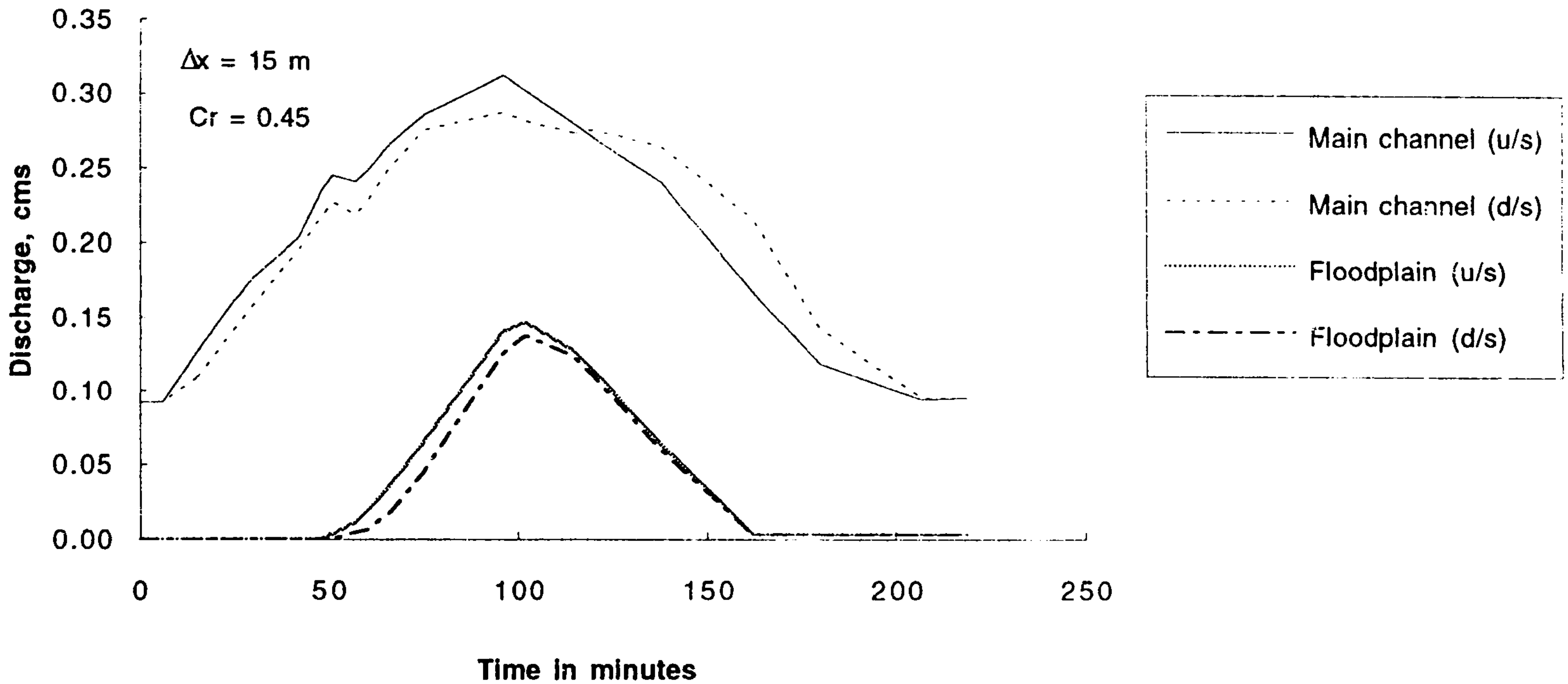


Figure 5.29 Distribution of main channel and floodplain discharges for CCDG 1-D (P-T) model at the upstream and downstream boundaries

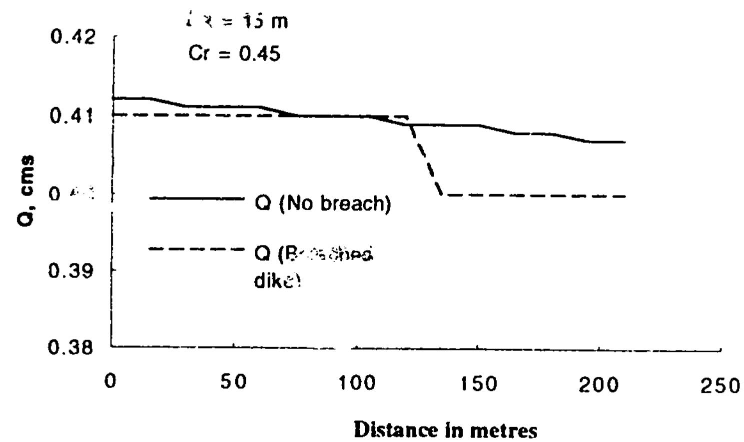


Figure 5.30 Distribution of main channel discharge at 99 minutes from start of flow (dike breach is located at 120 -150 m from upstream boundary)

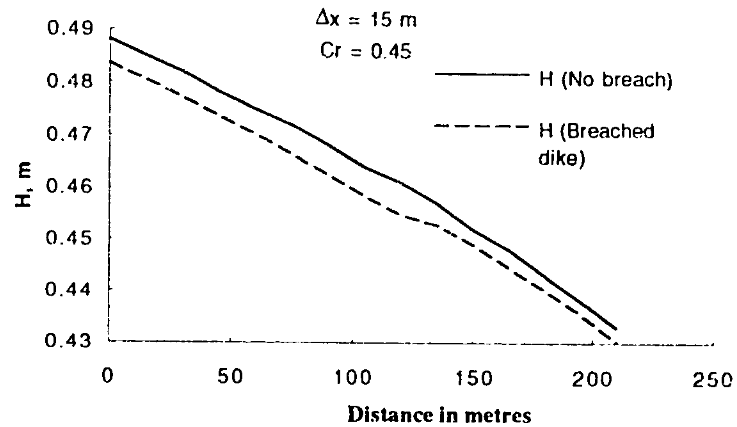


Figure 5.31 Distribution of main channel depth at 99 minutes from start of flow (the dike breach is located at 120 - 150 m from upstream boundary)

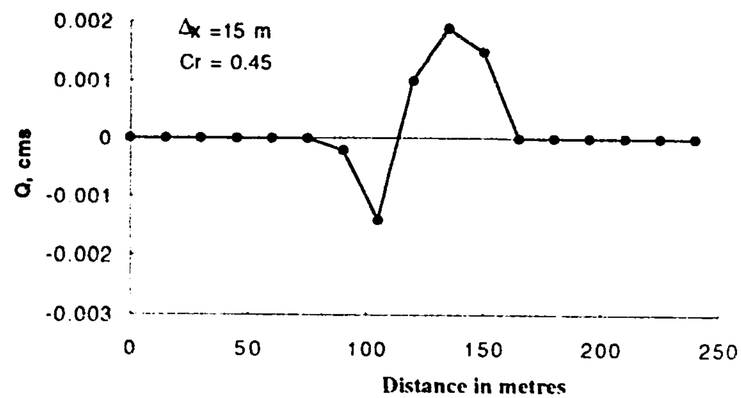


Figure 5.32 Discharge distribution in the left floodplain at 99 minutes from start of flow (dike breach is located at 120 - 150 m from upstream boundary)

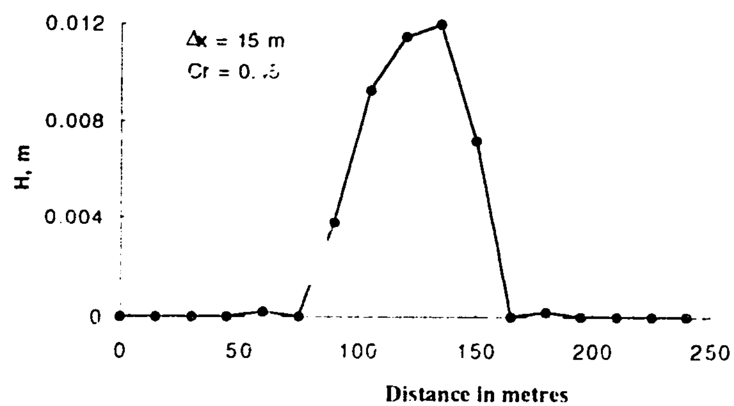


Figure 5.33 Distribution of left floodplain depth at 99 minutes from start of flow (the dike breaches is located at 120 -150 m from upstream boundary)

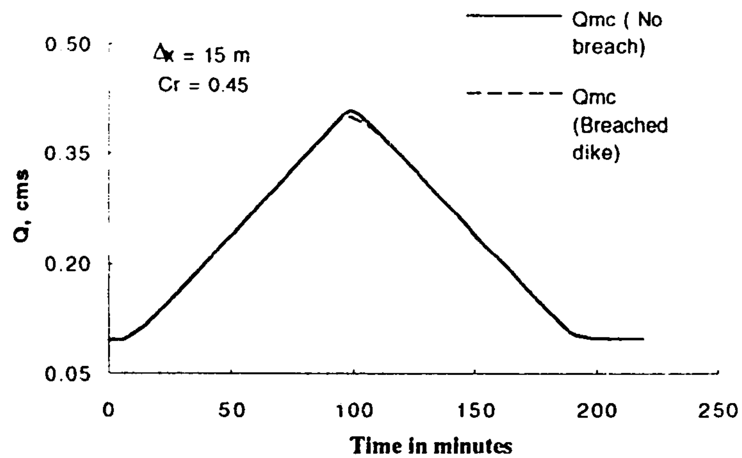


Figure 5.34 Main channel discharge distribution with time at 150 m from upstream boundary (dike breach is located at 120 - 150 m from upstream boundary)

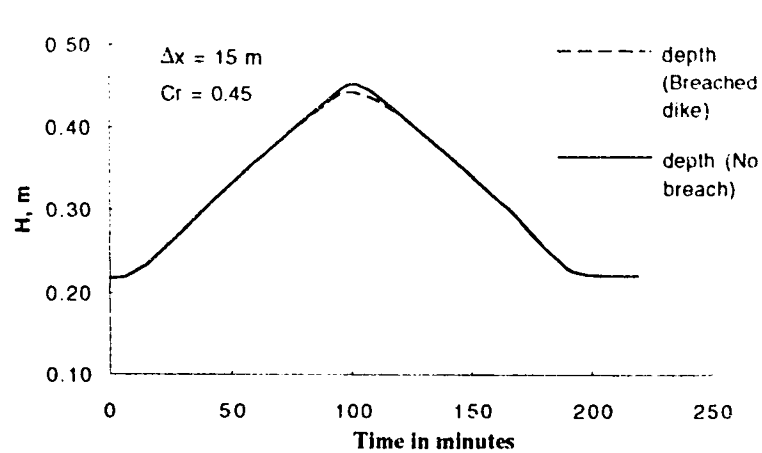


Figure 5.35 Distribution of main channel depth with time at 150 m from upstream boundary (the dike is located at 120 - 150 m downstream of the upstream boundary)

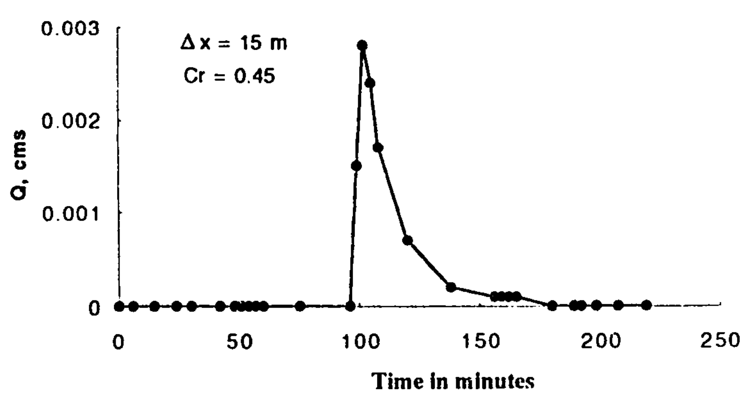


Figure 5.36 Distribution of left floodplain discharge with time at 150 m from upstream boundary (the dike breach is located at 120- 150 m from upstream boundary)

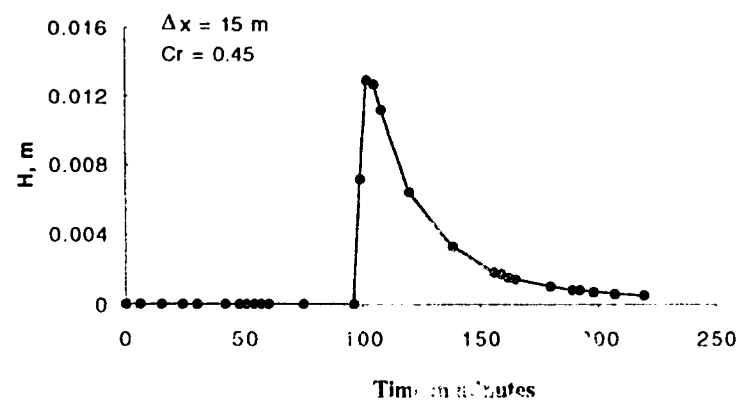
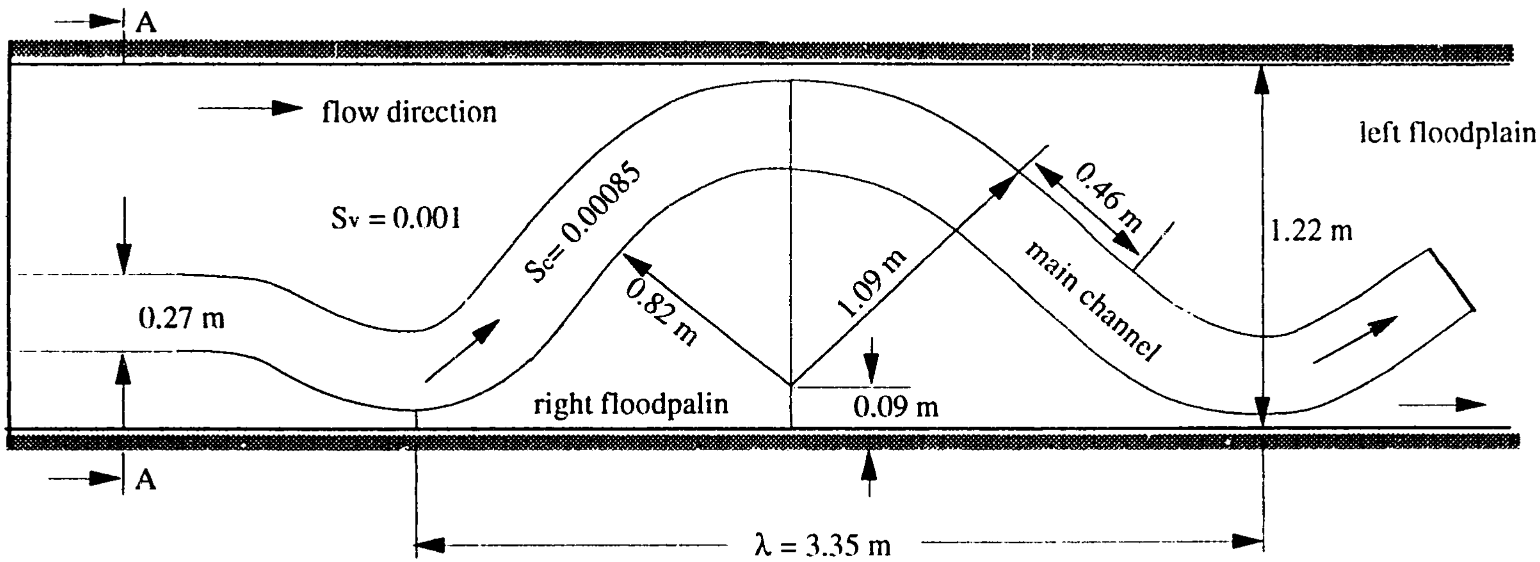
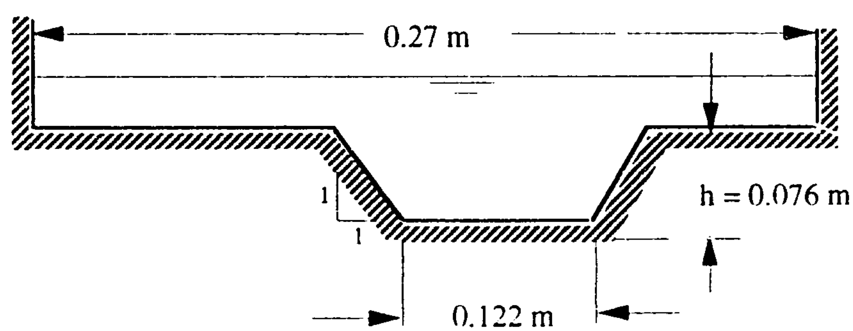


Figure 5.37 Distribution of left floodplain depth with time at 150 m from the upstream boundary (the dike breaches at 120-150 m from upstream boundary)



Plan view of Smith's meandering compound channel



Cross-section A-A

Figure 5.38 Smith's meandering compound channel.

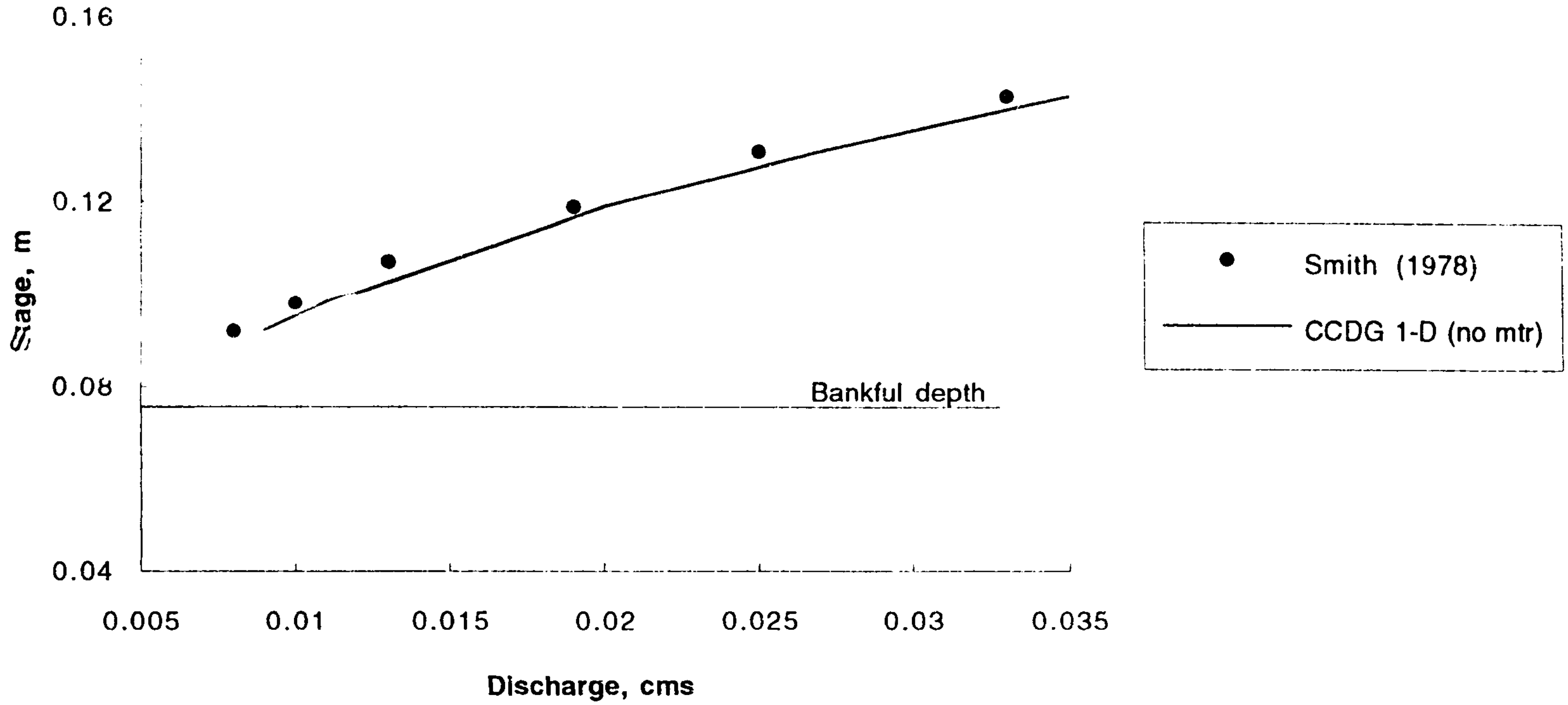


Figure 5.39 Comparison of stage-discharge relationship for observed flow and CCDG 1-D model results for flow in a meadering compound channel

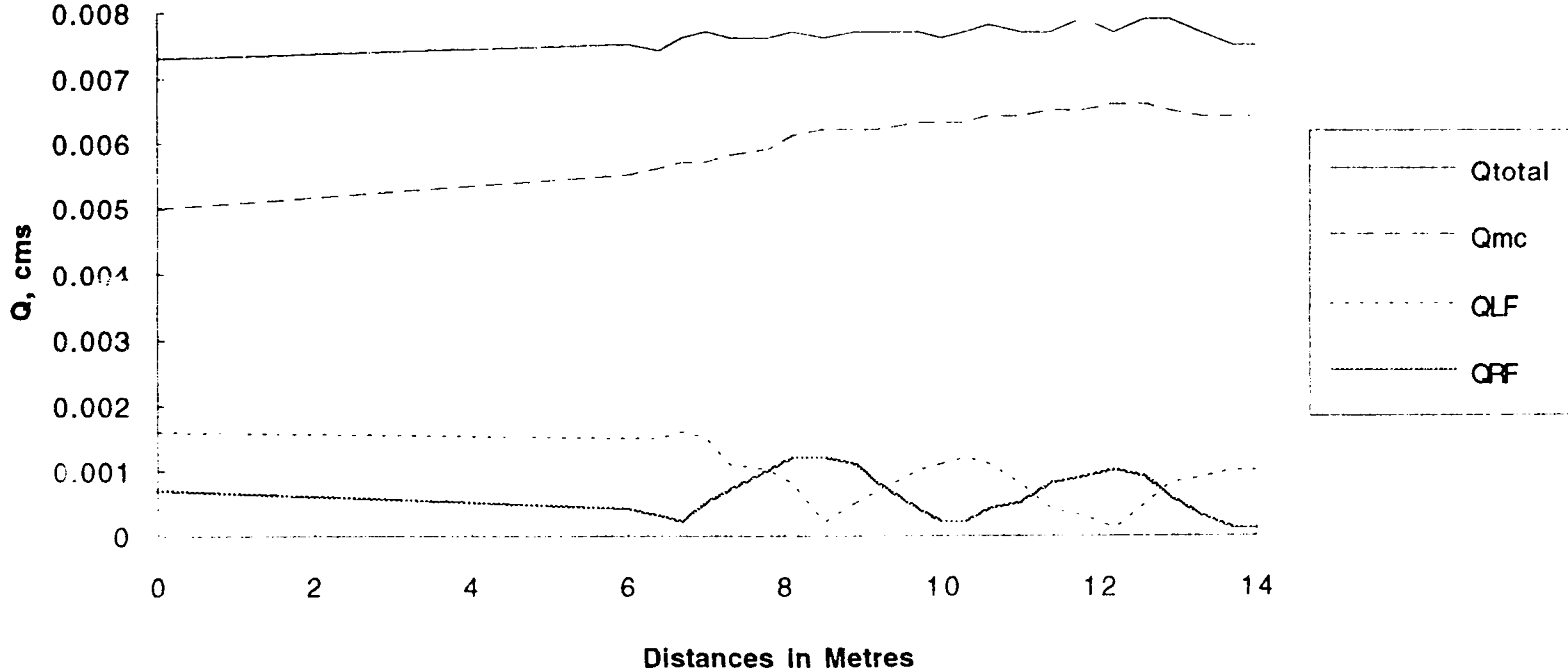


Figure 5.40 Longitudinal distribution of discharge in a meandering channel with floodplains

6.0 CONCLUSIONS AND RECOMMENDATIONS

The purpose of this research project was to evaluate the potential for a new coupled formulation of the equations of unsteady open channel flow in compound channels capable of handling the transport of mass and momentum between a channel and its floodplains. Considerable research effort has been devoted to the problem of open channel flow in compound channels in the past, primarily focused on developing an understanding of the flow structure and finding ways to improve the traditional methods used for stage-discharge prediction. However most of the work to date has concentrated on steady state problems.

Unsteady flow modelling is essential to assessing the impact of flood mitigation schemes, such as dikes, on flood hydrographs and flood levels if water resources engineers are to be able to: assess the effects of such schemes on unprotected areas, both due to potential backwater effects and reduced flood peak attenuation; reassess both the adequacy of existing works and their increased impact on unprotected areas; and provide updated flood forecasting information when flood control schemes fail, such as when a dike is overtopped and/or breached.

In this study a model of unsteady compound channel flow was formulated by modeling the channel with the full dynamic, one-dimensional (St. Venant) equations and by treating the floodplains as conveying channels with a diffusive wave approximation. The connection between the three separate flow systems was achieved by adapting the equations to include terms to account for mass and momentum transfer between the main channel and the floodplains, while considering a diffusive wave approximation in each of the floodplains. The momentum transfer terms consisted two effects: momentum transfer due to flow interaction (apparent shear); and convective momentum transport due to lateral outflow or inflow. Mass transfer to and from the channel was modelled in two ways. First as a side weir and second based on a simple lateral momentum balance. The resulting formulation provides a framework for assessing various models of mass and momentum transfer between the main channel and the floodplain for unsteady flow.

The resulting coupled formulation was solved numerically using the characteristic dissipative Galerkin finite element scheme, and was called the CCDG 1-D model. To assess the performance of the proposed formulation, steady and unsteady flow tests were performed on straight and meandering compound channels in a comparison to experimental measurements. Steady flow results in straight compound channels were also compared with conventional 'divided channel' methods, including the more recent zero-shear interface, or 'diagonal', method.

The steady state tests in straight channels showed that the inclusion of the momentum transfer terms, in the form of apparent shear stress, improved the predicted stage-discharge relationship over that of the traditional divided channel methods, with results being equally as good as those obtained using the diagonal method. It was found that the results were similar for a variety of apparent shear stress models, specifically those of Prinos and Townsend, Wormleaton, and Christoudolou. The proposed formulation was a particular improvement over conventional divided channel methods for the case when the stage in the floodplain was low and the floodplain roughness relative to the main channel was high, improving the discharge prediction by 8-15 %.

A sensitivity analysis on the effect of the weir discharge coefficient on the mass exchange between the main channel and the floodplain revealed that the solution was not particularly sensitive to variations in this parameter. The results showed only a marginal effect in the falling portion of the flood hydrograph. The lack of sensitivity may have been because the lateral outflow into the floodplain became quickly submerged by the floodplain flows being conveyed along the floodplain. The lateral discharge estimation using the simple lateral momentum conservation approach was found to perform very well. However it is stressed that there is no experimental data available with which to compare the validity of either approach.

Unsteady flow tests were run for comparison with experimental measurements. The proposed formulation was also compared to the performance of two unsteady flow models available in the public domain (DWOPER and ONE-D) which are based on conventional divided channel methods. The CCDG 1-D model (using the Prinos Townsend model to quantify the apparent shear effect) overestimated the peak discharge by about 4.6 % while the DWOPER model underestimated the peak discharge by 3.8 %. The ONE-D model exactly predicted the peak discharge. For unsteady flow, the test results showed that the inclusion of the momentum transfer terms affected the peak discharge, marginally. Although the DWOPER and ONE-D models showed slightly better results than the CCDG 1-D model, tit is stressed that modifications to the channel geometry, such as a gradual transition of the main channel bank into the floodplain and, in the case of the ONE-D model, flow withdrawal were required to achieve good results. The CCDG 1-D model required no such modification. However, a possible disadvantage of the CCDG 1-D model is that in coupling three separate flow systems, the flow distribution between the main channel and the floodplains at the upstream boundary, becomes a required boundary condition. Because this flow distribution has not been measured (nor is likely to be) this flow distribution must be approximated by conventional divided channel methods (at least at this time).

This study also illustrated that the proposed formulation has the potential to be used to simulate flow onto the floodplain through a breached dike. This capability is very important in that the proposed formulation model could help in determining the effects of such a dike failure on both the flood wave itself and in terms of floodplain inundation.

Accounting for the convective lateral momentum into the floodplain, showed a small increment in lateral discharge to the floodplain on flow through a breached dike. This means that in cases of large floods, the convective lateral momentum could be a significant term in increasing the floodplain inundation.

The study has established a framework of studying the problem of mass exchange between the floodplain and the main channel. This study has also shown that, the floodplain can be modelled as part of a conveying compound channel instead of as a storage area and that the inclusion of the apparent shear force into the coupled formulation provides a compound channel flow model which is as good as any current method in predicting the stage discharge relationship in channels with inundated floodplains. The demonstration of the coupled formulation for the breached dike scenario showed the possibility of using it to investigate the behavior of the flood wave when a dike is breached.

A preliminary test on steady flow in a meandering channel, showed that the proposed formulation shows promise for modelling flow in meandering channel. The simulation results showed a good agreement with the observed data, although the CCDG 1-D model overestimated the observed discharge by about 12.5 % at the low stages and 6.1 % at the high stages. Problems with mass conservation indicates that the proposed formulation requires further development before being used to model meandering compound channels.

The CCDG 1-D model, is still limited to rectangular and trapezoidal cross-sectional shapes but could be extended to other regular shapes, like parabolic cross-sections. It, however, offers itself as a useful investigative tool in providing some preliminary answers on compound flow problems. As recommendations for future research, it is suggested that the discharge coefficients used in determining lateral inflow or outflow be further investigated, especially regarding the case of submerged side weirs. The framework established to handle mass transfer between the main channel and the floodplain needs further development, especially in quantifying cross-over effects to extend the model for meandering channels. Before the proposed formulation can be applied to natural channels with confidence, field data is required to determine how river aspect ratios, channel shape, relative roughness and sediment transport affect the flow in compound channels.

REFERENCES

Abbott, M.B., and Cunge, J. A., 1975, "Two Dimensional Modelling of Tdal Deltas and Estuaries". *Unsteady Flow in Open Channels*, Water Resources Publication, Fort Collins, Colarado.

Abbott, M.B.,1979, *"Computational Hydraulics-Elements of the Theory of Free Surface Flows"*. Pitman Publishing Company, London, England, 324 pp.

Abida, H. and Townsend, R.D.,1994, " A model for routing unsteady flows in compound channels" *IAHR Journal of Hydraulic Research* , Vol. 32., No. 5.

Ahmadi, R., 1979, " An Experimental Study of Interaction Between Main Channel and Floodplain Flows". Thesis presented to the University of Alberta at Edmonton Alberta, Canada in partial fulfillment of the requirement for the degree of Doctor of Philosophy.

Amien, M., 1968, " An Implicit Method for Natural Flood Routing". *Water Resources Research*, Vol. 4, No.4, pp. 719 - 726.

Arunachalam, K., 1964, " Discharge Characterisctics of Weirs with Top Width". *Irrigation and Power*. pp. 435- 443.

Bazin, H.E., 1988, *Experiences Nouvelles sur L'ecoulement en Deversoir*, Paris.

Bhowmik, N. G., and Demissie, M., 1982, "Carrying Capacity of Flood Plains". *ASCE Journal of Hydraulic Division*, Vol. 108, No HY3, pp. 443-452.

Brice, J.C., 1964, "Channel Patterns and Terraces of Loup River in Nebraska". *U.S Geological Survey*, Project Paper 422-D.

Bos, M.G., 1989, *"Discharge Measurement Structures"*. International Institute for Land Reclamation and Improvement, Wageningen, The Netherlands, Publication 20.

Chaudhry, M.H., 1993, *"Open-Channel Flow"*. Prentice Hall, Englewood Cliffs, New Jersey, 483 pp.

Cheong, H. F., 1991, "Discharge Coefficient of Lateral Diversion From Trapezoidal Channel" *ASCE Journal of Irrigation and Drainage Engineering*, Vol.117, No.4, pp. 461-475.

Chorley, R.J., Schumm, S.A., Sugden, D.E., 1984, *"Geomorphology"*. Methuen, New York, NY, 605 pp.

Chow, V. T., 1959, *"Open channel Hydraulics"*. McGraw-Hill Book Company, New York N.Y., 680 pp.

Christodoulou, G.C., 1992, "Apparent Shear Stress in Smooth Compound Channels". *Water Resources Management*, No. 6, pp. 235- 247.

Clemmens, A.J., Replogle, M., and Bos, M.G, 1984, " Rectangular Measuring Flumes For Lined and Earthen Channels" *ASCE Journal of Irrigation and Drainage Engineering*, Vol.110, No.2, pp. 121-137.

CSCE Task Force on River Models., 1987, "One Dimensional River Models". *Proc. CSCE Annual Conference*, Montreal.

CSCE Task Force on River Models, Townsend, R., Chatila, J., and Gerges, H.,1993, "Evaluation of Dynamic Models for Rivers with Floodplains". *Proc. CSCE Annual Conference,* Fredericton, N.B.

Cruff, R.W., 1965, "Cross-Channel Transfer of Linear Momentum in Smooth Rectangular Channels". *Geological Survey water-supply paper,* 1592-B.

Cunge, J. A., Holly, F. M. and Verwey, A., 1980, *"Practical Aspects of Computational River Hydraulics"*. Pitman Publishing Company, London, England, 420 pp.

Delong, L.L., 1986, "Extension of the Unsteady One-Dimensional Open-Channel Flow Equations For Flow in Menadering Channels with Floodplains", *Selected Papers in Hydrological Science*, pp. 101-105.

Dracos, T. and Hardegger, P., 1987, "Steady Uniform Flow in Prismatic Channels with Floodplains". *IAHR Journal of Hydraulic Research* . Vol. 25, No.2.

El-Khashab, A., and Smith, K.V.H., 1976, " Experimental Investigation of Flow Over Side Weirs". *ASCE Journal of Hydraulic Division,* Vol. 102, No. 9, pp.1255-1268.

Ervine, D.A. and Baird, J. I., 1982, "Rating Curves for Rivers With Overbank Flow". *Proc. of Institute of Civil Engineers* (London) England.

Ervine, D.A. and Ellis, J., 1987, " Experimental Aspects of Overbank Floodplain Flow". *Transaction of the Royal society of Edinburg, Earth Sciences.* Vol 78, pp 315-325.

Ervine, D. A., Willets, B.B., Sellin, R.H.J., and Lorena, M., 1993, "Factors Affecting Conveyance in Meandering Compound Flows"*ASCE Journal Hydraulic Engineering*, Vol.119, No.12, pp. 1383 - 1399.

Ferrick, M.G., 1985, " Analysis of River Wave Types". *Water Resources*, Vol. 21 No.2 pp. 209 - 220.

Fread, D. L., 1976, "Flood Routing in Meandering Rivers with Flood Plains ". *Rivers, ASCE, 3rd Symposium of the Waterways, harbors and Coastal Engineering*, Colorado State University, Vol I, pp. 16-35.

Fread, D. L., 1988, *The NWS DAMBRK Model:* Theoretical Background /User Documentation. Office of Hydrology, National Weather Service (NWS), Maryland, USA.

Ghosh, S., and Jena, S.B., 1971, "Boundary Shear Distribution in Open Compound Channels". Proc. *of Institute of Civil Engineers* (London) England.

Ghosh, S. N. and Kar, S. K.,1975, "River Flood Plain Interaction and Distribution of Boundary Shear Stress in A Meander Channel with Flood Plain". *Proceedings of Institute of Civil engineers,* (London) England, Vol. 59, Part 2.

Govinda Rao, N.S., and Muralidhar, D., 1963), " Discharge Characteristics of Weirsof finite crest width". *La Houille Blanche*, 5 pp. 537-545.

Hager, W.H., 1987, "Lateral Outflow Over Side Weirs." *ASCE Journal Hydraulic Engineering*, Vol.113, No.4, pp. 491-504.

Hager, W. H., 1994, " Broad-Crested Weir". *ASCE Journal of Irrigation and Drainage Engineering*, Vol.120, No.1, pp. 13-26.

Henderson, F.M., 1966, *"Open Channel"*. MacMillan Publishing Co., Inc., New York, N.Y. pp. 522

Hicks, F.E., Steffler., P.M.,1990, "Finite Element Modeling of Open Channel Flow". *Water Resources Engineering Report* No. 90-6, Department of Civil Engineering, University of Alberta, Alberta.

Hicks, F.E., Steffler., P.M., 1992, "Characteristic Dissipative Galerkin Scheme for Open Channel flow". *ASCE Journal of Hydraulic Engineering*, Vol. 118, pp. 37-352.

Hicks, F.E., Yasmin, N. Chen, X., 1994, " Peace River Flow Analysis".*Water Resources Engineering Report 94-H2, NRBS Project 1154-C1*, 23 pp.

Hicks, F.E., Steffler., P.M., 1995, "Finite Element Methods for the St. Venant Equations". *International Journal for Numerical Methods in Fluids.* (in Press).

Holden, A.P. and James C. S., 1989, "Boundary Shear Distribution on Floodplains". *IAHR Journal of Hydraulic Research* . Vol. 27, No.1.

Hughes, T.J.R., and Mallet, M., 1986, " A New Finite Element Formulation for Computational Fluid Dynamics: III. The Generalized Streamline Operator for Multidimensional Advective-Diffusive Systems". *Computer Methods in Apllied Mechanics and Engineering*, Vol. 58. pp. 305-328.

Imamoto, H. and Ishigaki, T., 1989, "Secondary Flow in Compound Open channel". *Proc. of HYDROCOMP '89*, Elsevier Applied Science, pp. 234-243.

James, M. and Brown, B. J.,1977, "Geometric Parameters That Influence Floodplain Flow" . *U.S Army Engineer, Waterways Experimental Station, Vicksburg Mississippi. Research Report H-77-1.*

Kandaswamy, P.K., and Rouse, H., 1957, " Characteristics of Flow Over Terminal Weirs and Sills". *ASCE Journal of Hydraulic Division*, Vol.83, No.4, pp. 1345-1 - 1345-13.

Kandaswamy, P.K., and Rajaratnam, N., 1959, " Discussion on Calibration of Submerged Broad-crested Weirs".*ASCE Journal of Hydraulic Division,* Vol. 84, No. 9, pp. 119-124.

Katopodes, N.D., 1984, " A Dissipative Galerkin Scheme for Open-Channel Flow".*ASCE Journal of Hydraulic Engineering*, Vol.110, No.4, pp. 450-466.

Kawahara, Y. and Tamai, N.,1989, "Mechanism of Lateral Momentum Transfer in Compound Channel Clows". *Proc. of the 23rd Congress of IAHR,* Ottawa, Canada, pp. B 463 - B 467.

Kellerhals, R., Neil, C.R. and Bray, D.J., 1972, "*Hydraulic and Geomorphic Characteristics of Rivers in Alberta*". River Engineering and Surface Hydrology Report 72-1, Research Council of Alberta, 52 pp.

Kellerhals, R., Church, M. and Bray, D.J., 1976, "Classification and Analysis of River Process". *ASCE Journal of Hydraulic Division*, pp. 102 -813.

Keller, R. J. and Rodi, W.,1988, "Prediction of Flow Characteristics in Main Channel and Flood Plain Flows". *IAHR Journal of Hydraulic Research*, Vol. 26, No. 4, pp. 425-441.

Kiely, G.,1989, "A Comparison of Velocity Measurement in Straight, Single Meander and Multiple meander Compound Channels". *International Conference on Channel Flow And Catchment Runoff*, University of Virginia.

Kiely, G.,1990, "Overbank Flow in Meandering Compound Channels the Important Mechanisms". *Proceedings of International Conference on River Flood Hydraulics*, edited by White, W. R., paper F3.

Knight, D.W. and Demetriou, J. D.,1983, "Flood Plain and Main Channel Flow Interaction". *ASCE Journal of Hydraulic Engineering*, Vol.109, No.8, pp. 1073-1091.

Knight, D.W., Demetriou, J. D., and Hamed, M.E., 1983, "Hydraulic Analysis of Channels with Floodplains". *Proceedings of International Conference on Hydrologic Aspects of Floods*, Bedford: B.H.R.A. Fluid Engineering, pp. 129-144.

Knight, D.W. and Hamed E. M.,1984, "Boundary Shear in Symmetrical Compound Channel ".*ASCE Journal of Hydraulic Engineering*, Vol.110, No.10, pp. 1412-1430.

Kumar, C.P., and Pathak, S.K., 1987, " Triangular Side Weirs". *ASCE Journal of Irrigation and Drainage Engineering*, Vol.113, No.1, pp. 98-105.

Lakshamana, Rao, N.S.,1975, "Theory of Wiers"., Advances in Hydroscience Vol 10. Edited by V.T Chow, A.P New York.

Lee, J. K., and Froehlick, D.C., 1986, " Review of Literature on the Finite-Elements Solutions of the Equations of Two-Dimensional Surface-Water Flow in the Horizontal Plane". *U.S. Geological Survey Circular 1009*, 61 pp.

Leopold, L.B., Wolman, M.G., Miller, J.P., 1964," *Fluvial Processes in Geomorphology*" . W.H. Freeman and Company, San Fransisco,1964, 522 pp.

Liggett, J.A., and Cunge, J.A., 1975, "Numerical Methods of Solution of the Unsteady Flow Equations in Open Channels". *Unsteady Flow in Open Channels*, K. Mahmood and V. Yevjevich, Eds., Vol. 1, Water Resources Publications, Fort Collins, Colorado, pp. 89-182.

McKeogh, E. J., Kiely, G. K. and Javan, M.,1989, "Velocity and Turbulence Measurements in a Straight Channel with Interacting Flood Plains Using Laser Doppler Anemometry". *Proc. of the international conference on hydraulic and environmental Modeling of coastal, estuarine and river waters*, Bradford, Britain. pp. 429-440.

McKeogh, E. J. and Kiely, G. K., 1989, "Experimental Study of the Mechanisms of Flood Flow in Meandering Channels". *Proc. of the 23rd Congress of IAHR*, Ottawa, Canada, pp. B 491- B 498.

Muralidhar, D., 1964 " Flow Over End Weirs of FInite-Crest Width". *Irrigation and Power*. pp. 233-239.

Morisawa, M.,1985, *"Rivers"*. Longman House, Burnt Mill, Harlow, England, pp. 220.

Murota, A., Fukuhara, T. and Seta, M.,1990, "Effects of channel shape and flood plain roughness on flow structure in compound cross section". *Journal of Hydroscience and hydraulic engineering*. Vol. 8. No. 2., pp. 39-52.

Myers, R. C., and Elsawy, E. M.,1975, "Boundary Shear in Channel with Flood Plain". *ASCE Journal of Hydraulic Division*, Vol. 101, No HY7, pp. 933-946.

Myers, W.R.C., 1978, "Momentum Transfer in A Compound Channel". *Journal of IAHR Hydraulic Research* , Vol. 2. No. 16, PP. 139-150.

Myers, R. C., 1987, "Velocity and Discharge in Compound Channels". *ASCE Journal of Hydraulic Engineering*, Vol. 113, No HY6. pp. 753-766.

Myers W. R. C., 1990, "Physical Modeling of A Compound River Channel". *Proceedings of International Conference on river flood hydraulics*, edited by White, W. R., paper L1.

Nalluri, C. and Judy, N. D., 1985, "Interaction Between Main Channel and Flood Plain Flow". *Proc. of the 21st Congress of IAHR*, Melbourne, Australia, pp. 377-382.

Noutsopoulos, G., and Hadjipanos, P., 1983, " Discharge Computation in Compound Channels". Proc. *of the 20thCongress of IAHR*, Moscow.

Ponce, V.M., and Simons, D.B., 1977, " Shallow Water Wave Propagation in Open Channel Flow". *ASCE Journal of Hydraulic Division*, Vol. 103, No 12, pp. 1461-1476.

Ponce, V.M., Li, R.M., and Simons, D.B., 1978, "Applicability of Kinematic and Diffusion Wave Models". *ASCE Journal of Hydraulic Division*, Vol. 104, No 3, pp. 947-958.

Preissmann, A., 1961, " Propagation des Inntumescences Dans les Cannaux et les Rivieres". *le Congress de L'Association Francaise de Calcule*, Grenobe, France, pp. 433-442.

Prinos, P., Townsend, R.D., 1984, "Comparison of Methods for Predicting Discharge in Compound Open Channels". *Advances in Water Resources,*. Vol. 7, pp. 180-187.

Prinos, P., Townsend, R. D. and Tavoularis, R.,1985, "Structure of Turbulence in Compound Channel Flows". *ASCE Journal of Hydraulic Engineering*, Vol. 111, No 9, pp. 1246-1261.

Radojkovic, M.,1976 "Mathematical Modeling of Rivers with Floodplains". *Rivers, ASCE, 3rd Symposium of the Waterways, harbors and Coastal Engineering*, Colorado State University, Vol I, pp. 56-64.

Rajaratnam, N., and Ahmadi, R.,1981, "Hydraulics of Channels with Flood Plains". *ASCE Journal of Hydraulic Research*, Vol. 19, No. 1, pp. 43-59.

Rajaratnam, N., and Ahmadi, R. M.,1979, "Interaction Between Main Channel and Flood Plain Flows" *ASCE Journal of Hydraulic Division*, Vol. 105, No. HY5, pp. 573-588.

Rajaratnam, N., and Ahmadi, R.,1989, "Three Notes on Hydraulics of Channels with Flood plains". *Computational Modeling and Experimental Methods in Hydraulics*, (Hydrocomp '89), edited by Maksimovic C. and Radojkovic M., pp 224-233.

Ranga Raju, K.G., Prasad, B., and Gupta, S.K.,1979, " Side Weir in Rectangular Channel ". *ASCE Journal of Hydraulic Engineering*, Vol. 105, No. 5, pp. 547 - 554.

Ramamurthy, A.S., Tim, U.S., and Rao, M.V.J., 1988, " Characteristics of Square-Edged and Round-Nosed Broad-Crested Weirs". *ASCE Journal of Irrigation and Drainage Engineering*, Vol.114, No.1, pp. 61-73.

Rashid, R.S.M., and Chaudhry, M.H., 1993, " Numerical Modeling of Unsteady Compound Channel Flow". *Proc. ASCE Conference*, Hydraulic Engineering. pp. 1254-1259.

Rehbock, T., 1929, " Discussion of E.W. Schoder and K.B.Turner's, "Precise Weir Measurement", *Transactions A.S.C.E.*, Vol. 93.

Rijn van, L. C., 1990, " *Principles of Fluid Flow and Surface Waves in Rivers, Estuaries, Seas, and Oceans*". Aqua Publications, 335 pp.

Ritter. D.F., 1978, "Process Geomorphology". W.C. Brow Co., Dubuque, Iowa, 603 pp.

Samuels, P.G., 1985, " Modelling of River and Flood Plain Flow Using Finite Element Method". Hydraulics Research, Wallingford. Report No. SR 61.

Sellin, R. H. J.,1964, "A Laboratory Investigation into the Interaction Between the Flow in the Channel of A River and that Over its Floodplain" . *La Houille Blanche*, No 7, pp. 793-801.

Schumm, S.A., 1963,"A Tentative Classification of Alluvial River Channels". *U.S. Geological Survey Circular 477.*

Schumm, S.A., 1977, *"The Fluvial System"*. Wiley-Interscience, pp.338.

Shome, M. L., 1995, " A Computational Study of the Lateral Discharge Characteristics of Transient Compound Channel Flow". Thesis (unpublished) presented to the University of Alberta at Edmonton Alberta, Canada in partial fulfillment of the requirement for the degree of Doctor of Philosophy.

Simons, Li and Associates, 1982, "Engineering Analysis of Fluvial Systems". Fort Collins Colorado, USA, pp. 5.1-5.81.

Smith, R. A., 1959, " Calibration of Submerged Broad-crested Weir" *ASCE Journal of Hydraulic Division*, Vol. 85, No. 3, pp. 1-16.

Smith, K. V.H., 1973, "Computer Programming for Flow Over Side Weirs," *ASCE Journal of Hydraulic Division*, Vol. 99, No. 3, pp. 495-506.

Smith, C. D., 1978, "Effect of Channel Meanders on Flood Stage in Valley". *ASCE Journal of Hydraulic Division*, Vol. 104, No. HY1, pp. 49-58.

Stephenson, D. and Kolovopoulos, P.,1990, "Effects of Momentum Transfer in Compound Channels " *ASCE Journal of Hydraulic Engineering*, Vol. 116, No. 12, pp. 1512-1522.

Subramanya, K., and Awasthy, S.C., 1972," Spatially Varied Flow Over Side Weirs" *ASCE Journal of Hydraulic Engineering*, Vol. 98, No. 12, pp. 1-10.

Swamee, P.K., 1988, " Generalized Rectangular Weir equations". *ASCE Journal of Hydraulic Engineering*, Vol.114, No.8, pp. 945-949.

Swamee, P.K.,Pathak, S.K., Mohan, M., Agrawal, S. and Ali, M.S., 1994, "Subcritical Flow Over Rectangular Side Weir". *ASCE Journal of Irrigation and Drainage Engineering*, Vol.120, No.1, pp.213-217.

Toebes, G. H. and Sooky, A. A.,1967, "Hydraulics of Meandering Rivers with Flood-Plains". *ASCE Journal of waterways and Harbors Division*, pp. 213 - 236.

Tominaga, A. Nezu, I. and Ezaki, K., 1989, "Experimental Study on Secondary currents in Compound Open-channel Flows". Proc. *of the 23rd Congress of IAHR*, Ottawa, Canada, pp. A 15- A 22.

Treske, A., 1980., "Experimentelle Uberprufung Numerischer Berechrungsverfahven von Hachwasserwellen", in: Blind, Report of Hydraulics Research Station, TU Muchen, 44:1-33

U.S. Army Corps of Engineers., 1956, "Hydraulic Capacity of a Meandering Channel in Straight Floodways" . *Tech. memo 2-249 waterway experimental station Vicksburg* Mississippi.

U.S. Army Corps of Engineers., 1982, "HEC-2 Water Surface Profiles-Users Manual". *US Army Corps of Engineers*, Water Resources Support Center.

Wallingford, HR.,1992 "*SERC Flood Channel Facility*, "Experimental Data-Phase A", Volume 1, Report SR 314, Wallingford.

Williams, P.B., 1994, "Flood Control Vs.Flood Management". *Civil Engineering*, pp. 51-54

Wormleaton, P. R., Allen, J. and Hadjipanos, P.,1982, "Discharge Assessment in Compound Channels Flow". *ASCE Journal of Hydraulic Division,* Vol. 108, No. HY9, pp. 975-994.

Wormleaton, P. R., and Hadjipanos, P.,1985, "Flow Distribution in Compound Channels". *ASCE Journal of Hydraulic Division,* Vol. 111, No. 2, pp. 357-361.

Wormleaton, P. R., and Merret, D. J., 1990, "An Improved Method of Calculation for Steady Uniform Flow in Prismatic Main Channel/flood Plain Sections". *IAHR Journal of Hydraulic Research,* Vol. 28 No. 2, pp. 157-174.

Yevjevich V., 1975, "Introduction". *Unsteady Flow in Open Channels,* K. Mahmood and V. Yevjevich, Eds., Vol. 1,Water Resources Publications, Fort Collins, Colorado, pp. 1-27.

Yut-Tech, L. 1972, Discussion of " Spatially Varied Flow Over Side Weirs," by Subramanya, K., and Awasthy, S.C.*Journal of Hydraulic Engineering,* ASCE, Vol. 98, No. 11, pp. 2046 - 2048.

Zanobetti, D., Lorgere, H., Preissmann, A., and Cunge, J. A., 1970, " Mekong Delta Mathematical Model Program Construction". *ASCE Journal* of *Waterways and Harbors Division*, Vol. 96, No. WW2, pp. 181-199.

Zheleznyakov, G.V., 1971, "Interaction of Channel and Flood-plain Streams". *Proc. of the 14th Congress of IAHR, Paris, France, Vol. 5,* pp. 144-148.

APPENDIX A

A.1 The Pressure Term in the Momentum Equation

The pressure force through a centroid of trapezoid can be expressed as:

$$P = \overline{P}A = \gamma y_c A \qquad [A.1]$$

where:

P =pressure force acting through the centroid;

$\overline{P}$ =hydrostatic pressure; and

y_c = centroid of a trapezoid.

$$y_c = \left(\frac{BH^2}{2} + \frac{ZH^3}{6}\right)/A = \left(\frac{AH}{2} - \frac{ZH^3}{12}\right)/A \qquad [A.2]$$

where:

$Z = Z_1 + Z_2$. [A.3]

Z_1, Z_2 =the slopes of the trapezoid and

area for trapezoid is expressed as:

$$A = BH + \frac{ZH^2}{2} \qquad [A.4]$$

Then equation [A.1] becomes:

$$P = \gamma\left(\frac{AH}{2} - \frac{ZH^3}{12}\right) \qquad [A.5]$$

The longitudinal momentum equation is given as

$$\frac{\partial Q}{\partial t} + \frac{\partial (QV)}{\partial x} + gA\frac{\partial H}{\partial x} = gA(S_o - S_f) - V_x q \qquad [3.2]$$

and

The pressure term in equation [3.2] can the be rewritten for a trapezoid as:

$$\frac{\partial}{\partial x}\left(g\frac{AH}{2}-g\frac{ZH^2}{12}\right)=\frac{\partial}{\partial x}\left(g\frac{AH}{2}\right)-\frac{\partial}{\partial x}\left(g\frac{ZH^2}{12}\right) \qquad [A.6]$$

The last two terms in equation [A.6] are then expressed out as:

$$\frac{\partial}{\partial x}\left(\frac{gAH}{2}\right)=\frac{g}{2}\left(A\frac{\partial H}{\partial x}+H\frac{\partial A}{\partial x}\right)=\frac{g}{2}\left[\left(BH+\frac{Z}{2}H^2\right)\frac{\partial H}{\partial x}+H\frac{\partial}{\partial x}\left(BH+\frac{Z}{2}H^2\right)\right]$$

$$=\frac{g}{2}BH\frac{\partial H}{\partial x}+\frac{g}{4}ZH^2\frac{\partial H}{\partial x}+\frac{g}{2}BH\frac{\partial H}{\partial x}+\frac{g}{2}H^2\frac{dB}{dx}+\frac{g}{2}ZH^2\frac{\partial H}{\partial x}$$

$$=g\left(BH+\frac{1}{2}ZH^2\right)\frac{\partial H}{\partial x}+\frac{g}{2}ZH^2\frac{\partial H}{\partial x}+\frac{g}{2}H^2\frac{dB}{dx}$$

$$\frac{\partial}{\partial x}\left(g\frac{AH}{2}\right)=gA\frac{\partial H}{\partial x}+g\frac{H^2}{2}\frac{dB}{dx}+\frac{g}{2}ZH^2\frac{\partial H}{\partial x} \qquad [A.7]$$

and

$$\frac{\partial}{\partial x}\left(g\frac{ZH^3}{12}\right)=g\frac{ZH^2}{4}\frac{\partial H}{\partial x} \qquad [A.8]$$

Substitutin equations [A.7] and [A.8] into [A.6] and rearranging yields:

$$gA\frac{\partial H}{\partial x}=\frac{\partial}{\partial x}\left(g\frac{AH}{2}\right)-g\frac{H^2}{2}\frac{dB}{dx}-\frac{g}{4}ZH^2\frac{\partial H}{\partial x} \qquad [A.9]$$

Equation [A.9] is used in the pseudo-conservative momentum equation [3.].

The momentum equation may also be written in a non-conservation form using:

$$gH\frac{\partial A}{\partial x} = gH\frac{\partial}{\partial x}\left(BH + \frac{Z}{2}H^2\right) = gH\left(B\frac{\partial H}{\partial x} + H\frac{dB}{dx} + ZH\frac{\partial H}{\partial x}\right)$$

$$= g\left(BH + \frac{Z}{2}H^2\right)\frac{\partial H}{\partial x} + gH^2\frac{dB}{dx} + g\frac{ZH^2}{2}\frac{\partial H}{\partial x} \quad [A.10]$$

$$= gA\frac{\partial H}{\partial x} + gH^2\frac{dB}{dx} + g\frac{ZH^2}{2}\frac{\partial H}{\partial x}$$

then

$$gA\frac{\partial H}{\partial x} = gH\frac{\partial A}{\partial x} - g\frac{ZH^2}{2}\frac{\partial H}{\partial x} - g\frac{H^2}{2}\frac{dB}{dx} \quad [A.11]$$

A.2 Formulation of the Diffusive Equation.

The continuity and momentum equatuons for a diffusive equation is expesses respectively as:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0 \quad [2.19]$$

and

$$\frac{\partial H}{\partial x} = S_o - S_f \quad [3.12]$$

where the friction slope *(S_f)* is defined using c nveyance *(K)* as:

$$S_f = \frac{|Q|Q}{K^2} \quad [3.13]$$

Substituting equation [3.13] for the friction slope in equation [3.12] yields:

$$\frac{\partial H}{\partial x} = S_o - \frac{|Q|Q}{K^2} \quad [3.18]$$

Differentiating equation [3.12] with respect to x gives:

$$\frac{\partial^2 H}{\partial x^2} = -\frac{2|Q|}{K^2}\frac{\partial Q}{\partial x} + \frac{2|Q|Q}{K^3}\frac{\partial K}{\partial x} \qquad [A.12]$$

Multiplying equation [3.22] throughout by $K^2/2|Q|$ and writing an expression for $\partial Q/\partial x$ yields:

$$\frac{\partial Q}{\partial x} = \frac{Q}{K}\frac{\partial K}{\partial x} - \frac{K^2}{2|Q|}\frac{\partial^2 H}{\partial x^2} \qquad [A.13]$$

Substituting equation [A.13] into the continuity equation [2 19] yields:

$$\frac{\partial A}{\partial t} + \frac{Q}{K}\frac{\partial K}{\partial x} - \frac{K^2}{2|Q|}\frac{\partial^2 H}{\partial x^2} = 0 \qquad [A.14]$$

For Mannings equation (in S.I units) K is given as:

$$K = \frac{1}{n}AR^{2/3} \qquad [3.17]$$

$$R = \frac{A}{P} = \frac{A}{B + H\left(\sqrt{1+Z_1^2} + \sqrt{1+Z_2^2}\right)} = \frac{A}{B + HZ_{mc}} \qquad [3.15]$$

then $$K = \frac{A^{5/3}}{n(B + HZ_{mc})^{2/3}} \qquad [A.15]$$

and
$$\frac{\partial K}{\partial x} = \frac{5}{3}\frac{A^{2/3}}{n(B + HZ_{mc})^{2/3}}\frac{\partial A}{\partial x} - \frac{2}{3}\frac{A^{5/3}Z_{mc}}{n(B + HZ_{mc})^{5/3}}\frac{\partial H}{\partial x} \qquad [A.16]$$

substituting equations [A.15] and [A.16] into [A.14] and simplifying, yields:

$$\frac{\partial A}{\partial t}+\frac{5}{3}V\frac{\partial A}{\partial A}-\frac{Z_{mc}Q}{2(B+HZ_{mc})}\frac{\partial H}{\partial x}=D_l\frac{\partial^2 H}{\partial x^2} \quad [A.17]$$

For Chezy's equation,

$$K=C.A\sqrt{gR}=C.A\sqrt{g}\,\frac{A^{1/2}}{(B+HZ_{mc})^{1/2}}=C.\sqrt{g}\,\frac{A^{3/2}}{(B+HZ_{mc})^{1/2}} \quad [A.18]$$

and

$$\frac{\partial K}{\partial x}=\frac{3}{2}C.\sqrt{g}\frac{A^{1/2}}{(B+HZ_{mc})^{1/2}}\frac{\partial A}{\partial x}-\frac{1}{2}C.\sqrt{g}\frac{A^{3/2}Z_{mc}}{(B+HZ_{mc})^{3/2}}\frac{\partial H}{\partial x} \quad [A.19]$$

$$\frac{\partial A}{\partial t}+\frac{3}{2}V\frac{\partial A}{\partial A}-\frac{Z_{mc}Q}{2(B+HZ_{mc})}\frac{\partial H}{\partial x}=D_l\frac{\partial^2 H}{\partial x^2} \quad [A.20]$$

where D_l is a diffusion coefficient adefined for Chezy'ss formula as:

$$D_1=\frac{g\,C^2 A^3}{2|Q|(B+HZ_{mc})} \quad [3.22]$$

and for Mannigs formula as:

$$D_1=\frac{A^{10/3}}{2n^2\,(B+H\,Z_{mc})^{4/3}|Q|} \quad [3.23]$$

Equations [A.17] and [A.20] can be represented by one equation in the form of:

$$\frac{\partial A}{\partial t}+\alpha\,V\frac{\partial A}{\partial x}-\frac{Z_{mc}Q}{2(B+H\,Z_{mc})}\frac{\partial H}{\partial x}=D_l\frac{\partial^2 H}{\partial x^2} \quad [3.21]$$

α for Chezy"s is defined as:

$$\alpha \equiv \frac{3}{2}$$ [3.24]

while for Mannings formula it is given as:

$$\alpha \equiv \frac{5}{3}$$ [3.23]

A.3 Determinatuion of Jacobian

The solution of equation [4.12] requires an iterative solution, becuase **[K]** and **{F}** depends upon the solution vector, {Φ}. In this study, a Newton-Raphson iteration scheme is employed and the Jacobian used in this scheme is evaluated analytically as:

$$[\mathbf{J}] = \left[\frac{\partial\{\mathbf{R}\}}{\partial\{\Phi\}}\right]$$ [A.21]

where: {**R**} is the residual vector and **[J]** is the Jacobian matrix.

The equations needed tobe solved can be put in the fonn below:

$$\frac{\partial\{\Phi_i\}}{\partial t} + \frac{\partial\{F_i\}}{\partial x} + \{G_i\} = \{0\}$$ [A.22]

Deriving the weak statement from equation [A.22] yields:

$$\int_0^L \left([v_{ki}]\frac{\partial\Phi_i}{\partial t} - \frac{\partial[v_{ki}]}{\partial x}\{\mathbf{F}_i\} + [v_{ki}]\{\mathbf{G}_i\} \right) dx = \{\mathbf{0}\} - [v_{ki}]\{\mathbf{F}_i\}\Big|_0^L$$ [A.23]

where: $[v_{ki}]$ is the test functin matrix.

For Bubnov-Galerkin approximation, the test function is given as:

$$\left[v_{ki}\right]=\left[\mathbf{f}_{ki}\right] \quad [A.24]$$

The test function in Petrov-Galerkin formulation is presented as:

$$\left[v_{ki}\right]=\left[\mathbf{f}_{ki}\right]+\left[\frac{d\mathbf{f}_{km}}{dx}\right]\left[W_{mi}\right] \quad [A.25]$$

The test function of the Petrov-Galerkin formulation will hencefort be used to derive the the residual $\{\mathbf{R}\}$. Therefore equation [A.23] becomes:

$$\int_{0}^{L}\left(\begin{array}{c}\left(\left[\mathbf{f}_{ki}\right]+\left[\frac{d\mathbf{f}_{km}}{dx}\right]\left[W_{mi}\right]\right)\frac{\partial\Phi_{i}}{\partial t}-\frac{\partial}{\partial t}\left(\left[\mathbf{f}_{ki}\right]+\left[\frac{d\mathbf{f}_{km}}{dx}\right]\left[W_{mi}\right]\right)\left\{\mathbf{F}_{i}\right\} \\ \left(\left[\mathbf{f}_{ki}\right]+\left[\frac{d\mathbf{f}_{km}}{dx}\right]\left[W_{mi}\right]\right)\left\{\mathbf{G}_{i}\right\}\end{array}\right)dx=\{0\}+\left[\mathbf{f}_{ki}\right]\left\{\mathbf{F}_{i}\right\} \quad [A.26]$$

The $\{\Phi_j)$ is defines as:

$$\left\{\Phi_{i}\right\}=\left[\mathbf{f}_{ij}\right]\left\{\phi_{j}\right\} \quad [A.27]$$

Then equation [A.26] becomes:

$$\int_{0}^{L}\left(\begin{array}{c}\left(\left[\mathbf{f}_{ki}\right]+\left[\frac{d\mathbf{f}_{km}}{dx}\right]\left[W_{mi}\right]\right)\left[\mathbf{f}_{ji}\right]\frac{\partial\phi_{j}}{\partial t}- \\ \frac{\partial}{\partial t}\left(\left[\mathbf{f}_{ki}\right]+\left[\frac{d\mathbf{f}_{km}}{dx}\right]\left[W_{mi}\right]\right)\left\{\mathbf{F}_{i}\right\}+ \\ \left(\left[\mathbf{f}_{ki}\right]+\left[\frac{d\mathbf{f}_{km}}{dx}\right]\left[W_{mi}\right]\right)\left\{\mathbf{G}_{i}\right\}\end{array}\right)dx=\{0\}+\left[\mathbf{f}_{ki}\right]\left\{\mathbf{F}_{i}\right\}\Big|_{0}^{L} \quad [A.28]$$

Equation [A.28] can be rewritten as:

$$[\mathbf{S}_{kj}]\frac{d\{\phi_j\}}{dt}+\{\mathbf{K}_k\}=\{0\}+BT \qquad [A.29]$$

where:

$$[\mathbf{S}_{kj}]=\int_e\left([\mathbf{f}_{ki}][\mathbf{f}_{ij}]+\left[\frac{d\mathbf{f}_{km}}{dx}\right][\mathbf{W}_{mi}][\mathbf{f}_{ij}]\right)dx \qquad [A.30]$$

$$\{\mathbf{K}_k\}=\int_e\begin{pmatrix}-\left[\frac{d\mathbf{f}_{ki}}{dx}\right]\{\mathbf{F}_i\}_c+\left[\frac{d\mathbf{f}_{km}}{dx}\right][\mathbf{W}_{mi}]\frac{\partial}{\partial x}\{\mathbf{F}_i\}_{nc}\\ \\ +\ [\mathbf{f}_{ki}]\{\mathbf{G}_i\}_c+\left[\frac{d\mathbf{f}_{km}}{dx}\right][\mathbf{W}_{mi}]\frac{\partial}{\partial x}\{\mathbf{G}_i\}_{nc}\end{pmatrix}dx+B.T \qquad [A.31]$$

$$B.T=[\mathbf{f}_{ki}]\{\mathbf{F}_i\}_c\Big|_0^L \qquad [A.32]$$

To account for implicitness, $\{\mathbf{K}_k\}$ is define as:

$$\{\mathbf{K}_k\}=\theta\{\mathbf{K}_k^{n+1}\}+(1-\theta)\{\mathbf{K}_k^n\} \qquad [A.33]$$

and

$$\frac{\partial\phi_j}{\partial t}=\frac{\phi_j^{n+1}-\phi_j^n}{\Delta t} \qquad [A.34]$$

Substituting equations [A.33] and [A.34] into equation [A.29] and doing some rearrangement yields:

$$[\mathbf{S}_{kj}]\{\phi_j^{n+1}\}+\theta\Delta\{\mathbf{K}_k^{n+1}\}-[\mathbf{S}_{kj}]\{\phi_j^n\}+(1-\theta)\Delta\{\mathbf{K}_k^n\}=\{0\}+B.T. \qquad [A.35]$$

Then the residual $\{\mathbf{R}_k\}$ becomes:

$$\{\mathbf{R}_k\} = [\mathbf{S}_{kj}]\{\phi_j^{n+1}\} + \theta\Delta\{\mathbf{K}_k^{n+1}\} - \{\mathbf{F}_k^n\} \qquad [A.36]$$

where:

$$\{\mathbf{F}_k^n\} = [\mathbf{S}_{kj}]\{\phi_j^n\} - (1-\theta)\Delta\{\mathbf{K}_k^n\} + B.T. \qquad [A.37]$$

The residual $\{\mathbf{R}_k\}=\{\mathbf{0}\}$ when the problem is solved.

From equation [A21] the Jacobian is then calculated as:

$$[\mathbf{J}_{km}] = \left[\frac{\partial\{\mathbf{R}_k\}}{\partial\{\phi_j^{n+1}\}}\right] = \left[\frac{\partial\{\mathbf{R}_k\}}{\partial\{\phi_m\}}\right] \qquad [A.38]$$

$$[J_{km}] = \frac{\partial}{\partial\{\phi_m^{n+1}\}}\{[\mathbf{S}_{kj}]\{\phi_j^{n+1}\}\} + \theta\Delta t\frac{\partial}{\partial\{\phi_m^{n+1}\}}\{\mathbf{K}_k^{n+1}\} \qquad [A.39]$$

Finding the derivative of the first part of equation [A.39] leads to:

$$\frac{\partial}{\partial\{\phi_m^{n+1}\}}\{[\mathbf{S}_{kj}]\{\phi_j^{n+1}\}\} = \frac{[\mathbf{S}_{kj}]}{\partial\{\phi_j^{n+1}\}}\{\phi_j^{n+1}\} + [\mathbf{S}_{kj}]\frac{\{\phi_j^{n+1}\}}{\partial\{\phi_j^{n+1}\}} = [\mathbf{S}_{kj}] \qquad [A.40]$$

The mass matrix $[\mathbf{S}_{kj}]$ has no $\{\phi\}$ in it, therefore its derivative with respect to $\{\phi\}$ is zero.

The derivative of the second part is given as:

$$\frac{\partial}{\partial\{\phi_j^{n+1}\}}\{\mathbf{K}_k^{n+1}\} = \theta\Delta t \frac{\partial}{\partial\{\phi_j^{n+1}\}}\{\mathbf{K}^{n+1}\}$$

$$= \theta\Delta t \left\{ \int_c \left(\begin{array}{c} -\frac{d\mathbf{f}_{ki}}{dx}\frac{\partial\{\mathbf{F}_i\}_c}{\partial\{\phi_j^{n+1}\}} + [\mathbf{f}_{ki}]\frac{\partial\{\mathbf{G}_i\}_c}{\partial\{\phi_j^{n+1}\}} \\ + [\mathbf{W}_{mi}]\left[\frac{d\mathbf{f}_{kmi}}{dx}\right]\frac{\partial\{\mathbf{F}_i\}_{nc}}{\partial\{\phi_j^{n+1}\}} \\ + [\mathbf{W}_{mi}]\left[\frac{d\mathbf{f}_{kmi}}{dx}\right]\frac{\partial\{\mathbf{G}_i\}_{nc}}{\partial\{\phi_j^{n+1}\}} \end{array} \right) dx + [\mathbf{f}_{ki}]\frac{\partial\{\mathbf{F}_i\}_c}{\partial\{\phi_j^{n+1}\}}\Bigg|_0^L \right\} \quad \text{[A.41]}$$

If ϕ_l is define as follows:

$$\phi_j = [\mathbf{f}_{jl}]\{\phi_l\} \quad \text{[A.42]}$$

Then, $\frac{\partial\phi_j}{\partial\phi_l^{n+1}} = [\mathbf{f}_{jl}]\frac{\partial\{\phi_l\}}{\partial\{\phi_l^{n+1}\}} = [\mathbf{f}_{jl}]$ [A.43]

$$\frac{\partial\{F_i\}}{\partial\{\phi_l^{n+1}\}} = \frac{\partial F_i}{\partial\phi_j}\times\frac{\partial\phi_j}{\partial\phi_l} = \left[\frac{\partial F_i}{\partial\phi_j}\right][\mathbf{f}_{jl}] \quad \text{[A.44]}$$

$$\frac{\partial\{G_i\}}{\partial\{\phi_l^{n+\cdot}\}} = \frac{\partial G_i}{\partial\phi_j}\times\frac{\partial\phi_j}{\partial\phi_l} = \left[\frac{\partial G_i}{\partial\phi_j}\right][\mathbf{f}_{jl}] \quad \text{[A.45]}$$

Therefore,

$$\frac{\partial\{\mathbf{K}_k^{n+1}\}}{\partial\{\phi_l^{n+1}\}} = \int_e \begin{pmatrix} -\left[\frac{d\mathbf{f}_{ki}}{dx}\right]\left[\frac{\partial F_{ic}}{\partial \phi_j}\right]\left[\mathbf{f}_{jl}\right] + \left[f_{ki}\right]\left[\frac{\partial G_{ic}}{\partial \phi_j}\right]\left[\mathbf{f}_{jl}\right] \\ +\left[W_{mi}\right]\left[\frac{d\mathbf{f}_{kmi}}{dx}\right]\left[\frac{\partial F_{inc}}{\partial \phi_j}\right]\left[\frac{d\mathbf{f}_{jl}}{dx}\right] \\ +\left[W_{mi}\right]\left[\frac{d\mathbf{f}_{kmi}}{dx}\right]\left[\frac{\partial G_{inc}}{\partial \phi_j}\right]\left[\mathbf{f}_{jl}\right] \end{pmatrix} dx \qquad [A.46]$$

Substituting equations [A.40] and [A.46] into equation [A. 39] yields the Jacobian as:

$$\left[\mathbf{J}_{kl}\right] = \left[\mathbf{S}_{kl}\right] + \theta\Delta t \left\{ \int_e \begin{pmatrix} -\left[\frac{d\mathbf{f}_{ki}}{dx}\right]\left[\frac{\partial F_{ic}}{\partial \phi_j}\right]\left[\mathbf{f}_{jl}\right] \\ +\left[\frac{d\mathbf{f}_{kmi}}{dx}\right]\left[\mathbf{W}_{mi}\right]\left[\frac{\partial F_{inc}}{\partial \phi_j}\right]\left[\frac{d\mathbf{f}_{jl}}{dx}\right] \\ +\left[\mathbf{f}_{ki}\right]\left[\frac{\partial G_{inc}}{\partial \phi_j}\right]\left[\mathbf{f}_{jl}\right] \\ +\left[\frac{d\mathbf{f}_{kmi}}{dx}\right]\left[\mathbf{W}_{mi}\right]\left[\frac{\partial G_{inc}}{\partial \phi_j}\right]\left[\mathbf{f}_{jl}\right] \end{pmatrix} \right\} dx \qquad [A.47]$$

where:

$$\left[\frac{\partial F_i}{\partial \phi_j}\right] = \begin{bmatrix} \frac{\partial F_1}{\partial A_l} & \frac{\partial F_1}{\partial A_c} & \frac{\partial F_1}{\partial Q_c} & \frac{\partial F_1}{\partial A_r} \\ \frac{\partial F_2}{\partial A_l} & \frac{\partial F_2}{\partial A_c} & \frac{\partial F_2}{\partial Q_c} & \frac{\partial F_2}{\partial A_r} \\ \frac{\partial F_3}{\partial A_l} & \frac{\partial F_3}{\partial A_c} & \frac{\partial F_3}{\partial Q_c} & \frac{\partial F_3}{\partial A_r} \\ \frac{\partial F_4}{\partial A_l} & \frac{\partial F_4}{\partial A_c} & \frac{\partial F_4}{\partial Q_c} & \frac{\partial F_4}{\partial A_r} \end{bmatrix} \quad [A.48]$$

$$\left[\frac{\partial G_i}{\partial \phi_j}\right] = \begin{bmatrix} \frac{\partial G_1}{\partial A_l} & \frac{\partial G_1}{\partial A_c} & \frac{\partial G_1}{\partial Q_c} & \frac{\partial G_1}{\partial A_r} \\ \frac{\partial G_2}{\partial A_l} & \frac{\partial G_2}{\partial A_c} & \frac{\partial G_2}{\partial Q_c} & \frac{\partial G_2}{\partial A_r} \\ \frac{\partial G_3}{\partial A_l} & \frac{\partial G_3}{\partial A_c} & \frac{\partial G_3}{\partial Q_c} & \frac{\partial G_3}{\partial A_r} \\ \frac{\partial G_4}{\partial A_l} & \frac{\partial G_4}{\partial A_c} & \frac{\partial G_4}{\partial Q_c} & \frac{\partial G_4}{\partial A_r} \end{bmatrix} \quad [A.49]$$

APPENDIX B

```
C **********PROGRAM MAIN WHICH WRITES THE MASS ELEMENT MATIX********
       PROGRAM ELEMENT MASS  MATRIX
       IMPLICIT REAL *8(A-H,O-Z)
      CHARACTER*12 FILE5,FILE6
       DIMENSION RHS(200),P2(200),DPHI(200),ES(8,8),FCL(200),FCR(200),
     +P3(200),P4(200),TH(200),Zo(200),Ho(200),TZ(200)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     + Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     + PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     + HRSTEP(200),CASEL(200),CASER(200),WALL
C
       PRINT *,'GIVE DATA FILE NAME '
       READ(*,'(A)') FILE5
       PRINT *,' '
       PRINT *,'GIVE OUTPUT FILENAME'
       READ(*,'(A)') FILE6
      OPEN(UNIT=5,FILE =FILE5,STATUS='OLD')
      OPEN(UNIT=6,FILE=FILE6,STATUS='UNKNOWN')
C
       DATA DPHI/200*0.0D+00/
C
       CALL INPUT(NSTEP,NITER,TOL,MTD,K,KUW,KLP,KFL,Cr,DT,NGP,NQc,
     + IY1,IY2,IY3,IY4,IY5,IY6,IY7,IY8,IY9,IY10,IY11,IY12,IY13,IY14,
     +IY15,IY16,IY17,IY18,IY19,IY20,IY21,IY22,IY23,IY24,IY25,IY26,
     + IY27,IY28,JQF,ITAO,TM,DST,TAG1,TAG2,PET)
C
C   COMPUTE THE MINIMUM ELEMENT LENGTH (DXMIN)
C-----------------------------------------
C
      WRITE(6,12)THETA
      WRITE(6,13)DT
11      FORMAT(/6X,'DX IS = ',F12.2,1X,'M')
12      FORMAT(/6X,'THETA IS = ',F3.1)
13      FORMAT(/6X,'DT IS = ',F8.2,1X,'SEC')
C
      GRAV=9.81D+00
      RHO=1000.0D+00
      NTRY=0
      NTEST=0
      T = 0.0D+00
C
       Z= Z1 + Z2
       CALL SLOPE(P2,P3,P4,
C
C    READ ESSENTIAL BOUNDARY CONDITIONS***
```

```
C     --------------------------------
C
      WRITE(6,314)
314   FORMAT(/2X,'NATURE OF BOUNDARY CONDITIONS')
C
       READ(5,*)IBC(1),IBC(2),IBC(3),IBC(4),NBCUS,II
       WRITE(6,315)IBC(1),IBC(2),IBC(3),IBC(4),NBCUS,II
C
      DO 310 I=1,NBCUS
      READ(5,*)TH(I),Zo(I)
C      WRITE(6,313)I,TH(I),Zo(I)
310    CONTINUE
C
       READ(5,*)IBC(5),IBC(6),IBC(7),IBC(8),NBCDS,III
       WRITE(6,315)IBC(5),IBC(6),IBC(7),IBC(8),NBCDS,III
C
      DO 312 I=1,NBCDS
      READ(5,*)TZ(I),Ho(I)
C      WRITE(6,313)I,TZ(I),Ho(I)
312    CONTINUE
C
315    FORMAT(2X,6(2X,I3))
313    FORMAT(2X,I2,F10.1,3(2X,F10.3))
C
C
C      WRITE(6,316)
C316   FORMAT(/2X,'INITIAL CONDITIONS')
C
C      IF(MTD.EQ.2)THEN
C      WRITE(6,317)
317   FORMAT(18X,'AL',12X,'Amc',10X,'Qmc',12X,'AR')
C      ELSE
C      WRITE(6,318)
C318   FORMAT(18X,'Amc',12X,'Qmc')
C      ENDIF
C
       CALL CHPROP(IELNO,NITER,NTEST,T,DPHI,NTRY,MTD,K,P2,P3,P4,
     + KLP,NQc,JQF,ITAO,TAG1,TAG2,PET)
C
      DO  600 N = 1,NSTEP
C
       T = T + DT
       T1 = ELVLP(1)+HLSTEP(1)-ELVMc(1)
       Tw = PAR(1,2) + Z*T1
      CALL INTERPO(TH,Zo,T,NBCUS,UNK1,TM)
       IF((II.EQ.1).AND.(K.EQ.2))THEN
       PHI(1)=UNK1*PAR(1,2)
       Acnew(1)=PHI(1)
       Hcnew(1)=UNK1
       ELSEIF((II.EQ.2).AND.(K.EQ.2))THEN
       PHI(2)=UNK1
       ELSEIF((II.EQ.1).AND.(K.EQ.4))THEN
       IF(UNK1.GT.T1)THEN
       ATP= PAR(1,2)*T1 + Z*T1**2/2.0
       HN= UNK1 - T1
       ATOP = HN*Tw
       PHI(2)= ATP + ATOP
       ELSE
```

```
       PHI(2)=UNK1*PAR(1,2) + Z*UNK1**2/2.0
       ENDIF
       Hcnew(1)=UNK1
       Acnew(1)=PHI(2)
       ELSE
       PHI(3) =  UNK1
       ENDIF
C
       TIML=T/TM
C      WRITE(6,115)TIME,ATP,HN,ATOP,PHI(2),Hcnew(1),Acnew(1)
C
       HSTEP=ELVLP(1)+HLSTEP(1)-ELVMc(1)
C
         IF(II.EQ.2)THEN
      CALL QNITIAL(UNK1,P2,P3,P4,HSTEP,TOL,N,COTR)
        IF(COTR.EQ.1.0) GO TO 700
         ENDIF
C
      T1= ELVLP(NNODES)+HLSTEP(NNODES)-ELVMc(NNODES)
C
       Tw = PAR(NNODES,2) + Z*T1
C
      CALL INTERPO(TZ,Ho,T,NBCDS,UNK1,TM)
       IF((III.EQ.1).AND.(K.EQ.2))THEN
       PHI(NNODES*2-1)=UNK1*PAR(NNODES,2)
       Acnew(NNODES)=PHI(NNODES*2-1)
       Hcnew(NNODES)=UNK1
       ELSEIF((III.EQ.2).AND.(K.EQ.2))THEN
       PHI(NNODES*2)=UNK1
       ELSEIF((III.EQ.1).AND.(K.EQ.4))THEN
       IF(UNK1.GT.T1)THEN
       ATP=PAR(NNODES,2)*T1 + Z*T1**2/2.0
       HN= UNK1 - T1
       ATOP = HN*Tw
       PHI(NNODES*4-2)= ATP + ATOP
       ELSE
       PHI(NNODES*4-2)=UNK1*PAR(NNODES,2) + Z*UNK1**2/2.0
       ENDIF
       Hcnew(NNODES)=UNK1
       Acnew(NNODES)=PHI(NNODES*4-2)
       ELSE
       PHI(NNODES*4-1) = UNK1
       ENDIF
C
C      WRITE(6,115)TIME,ATP,HN,ATOP,PHI(NNODES*4-2),Hcnew(NNODES),
C     + Acnew(NNODES)
C
       IF(II.EQ.1)THEN
       T2 = ELVLP(1)+HLSTEP(1)-ELVMc(1)
       IF(Hcnew(1).LE.T2)THEN
        HLnew(1)=HLnew(1)
        PHI(1)=PHI(1)
        ELSE
       HLnew(1)=Hcnew(1) - T2
       PHI(1)=HLnew(1)*PARF(1,2) + Z3*HLnew(1)**2/2.0
       ENDIF
       T3 = ELVRP(1)+HRSTEP(1)-ELVMc(1)
       IF(Hcnew(1).LE.T3)THEN
```

```
        HRnew(1)=HRnew(1)
        PHI(4)=PHI(4)
        ELSE
       HRnew(1)=Hcnew(1)-T3
       PHI(4)=HRnew(1)*PARL(1,2)  + Z4*HRnew(1)**2/2.0
          ENDIF
          ENDIF
C
C           IF(JQF.EQ.1)THEN
       T4 = ELVLP(NNODES)+HLSTEP(NNODES)-ELVMc(NNODES)
       IF(Hcnew(NNODES).LE.T4)THEN
        HLnew(NNODES)=HLnew(NNODES)
        PHI(NNODES*4-3)=PHI(NNODES*4-3)
        ELSE
       HLnew(NNODES)=Hcnew(NNODES)  - T4
       PHI(NNODES*4-3)=HLnew(NNODES)*PARF(NNODES,2)  + Z3*
     + HLnew(NNODES)**2/2.0
       ENDIF

       T5 = ELVRP(NNODES)+HRSTEP(NNODES)-ELVMc(NNODES)
       IF(Hcnew(NNODES).LE.T5)THEN
        HRnew(NNODES)=HRnew(NNODES)
        PHI(NNODES*4)=PHI(NNODES*4)
         ELSE
       HRnew(NNODES*4)=Hcnew(NNODES*4)-T5
       PHI(NNODES*4)=HRnew(NNODES)*PARL(NNODES,2)   + Z4*
     + HRnew(NNODES)**2/2.0
          ENDIF
C             ENDIF

C             JJ=NNODES-1
C             PHI(NNODES*4-3)=PHI(JJ*4-3)
C             HLnew(NNODES)=PHI(NNODES*4-3)/PARF(NNODES,2)
C             PHI(NNODES*4)=PHI(JJ*4)
C             ALnew(NNODES)=PHI(NNODES*4-3)
C             ARnew(NNODES)=PHI(NNODES*4)
C             HRnew(NNODES)=PHI(NNODES*4)/PARL(NNODES,2)
C             ENDIF
C
C       WRITE(6,115)TIME,PHI(NNODES-3),PHI(NNODES-2),PHI(NNODES),
C     + HLnew(NNODES),HRnew(NNODES),Acnew(NNODES)
115     FORMAT(2X,7(2X,F10.3))

      IF(T.EQ.DT)THEN
       WRITE(6,430)
      IF(II.EQ.1)THEN
       WRITE(6,435)
       ELSE
       WRITE(6,436)
       ENDIF
       ENDIF
430     FORMAT(/2X,'**** SOLUTIONS AT DIFFERENT TIMES *****')
435     FORMAT(/2X,'STAGE HYDROGRAPH IS INPUT UPSTREAM')
436     FORMAT(/2X,'DISCHARGE HYDROGRAPH IS INPUT UPSTREAM')
C
       IF(T.EQ.DT) GO TO 225
C
220    CALL CHPROP(IELNO,NITER,NTEST,T,DPHI,NTRY,MTD,K,P2,P3,P4,
```

```
     + KLP,NQc,JQF,ITAO,TAG1,TAG2,PET)
225      CONTINUE
14      FORMAT(2X,F6.3)
C
370      CONTINUE
C
      CALL ASSEMB(FCL,FCR,DT,NTEST,MTD,K,KUW,P2,P3,P4,
     + KLP,ES,KFL,NGP,T)
C
       CALL ASSJACOB(DT,MTD,K,KUW,P2,P3,P4,ES,KLP,KFL,NGP,HSTEP,T)
C
      CALL RESIDUAL(FCL,FCR,RHS,TOL,DPHI,NITER,NTEST,
     + NTRY,MTD,K)
C
       CALL DEPTH(NTEST,T,K,MTD,TAG1,TAG2)
C
       T2 = ELVLP(1)+HLSTEP(1)-ELVMc(1)
       T3 = ELVRP(1)+HRSTEP(1)-ELVMc(1)
        IF(II.EQ.2)THEN
       IF(Hcnew(1).LE.T2)THEN
        HLnew(1)=HLnew(1)
        PHI(1)=PHI(1)
        HRnew(1)=HRnew(1)
        PHI(4)=PHI(4)
        ELSE
       HLnew(1)=Hcnew(1)-T2
       PHI(1)=HLnew(1)*PARF(1,3) + Z3*HLnew(1)**2/2.0
       HRnew(1)=Hcnew(1)-T3
       PHI(4)=HRnew(1)*PARL(1,2) + Z4*HRnew(1)**2/2.0
        ENDIF
        ENDIF
C
        CALL QFLPL7                    4,KLP,TM,JQF,TAG1,TAG2)
C
       IF(NTEST
C
       IF(NTF
       WRITE(

400    FORMAT(                           , ',',2X,'# OF ITER. ARE',2X,I4)
        GO TO
         ENDIF
       IF((N.EQ.           N.EQ.IY2).OR.(N.EQ.IY3).OR.(N.EQ.IY4).OR.
     + (N.EQ.IY5).OR.(N.EQ.IY6).OR.(N.EQ.IY7).OR.(N.EQ.IY8).OR.
     + (N.EQ.IY9).OR.(N.EQ.IY10).OR.(N.EQ.IY11).OR.(N.EQ.IY12).OR.
     + (N.EQ.IY13).OR.(N.EQ.IY14).OR.(N.EQ.IY15).OR.(N.EQ.IY16).OR.
     + (N.EQ.IY17).OR.(N.EQ.IY18).OR.(N.EQ.IY19).OR.(N.EQ.IY20).OR.
     + (N.EQ.IY21).OR.(N.EQ.IY22).OR.(N.EQ.IY23).OR.(N.EQ.IY24).OR.
     +(N.EQ.IY25).OR.(N.EQ.IY26).OR.(N.EQ.IY27).OR.(N.EQ.IY28))THEN
C
       WRITE(6,460) N,NTRY
       WRITE(6,470)TIME
C
       IF(MTD.EQ.2)THEN
       WRITE(6,464)
       ELSE
       WRITE(6,462)
       ENDIF
```

```
C
        DO 450 I=1,NNODES
       DIST=PAR(I,1)/DST
       Qt(I)= PHI(I*K-1)+QLnew(I)+QRnew(I)
       QtF(I)= QLnew(I)+QRnew(I)
        IF(MTD.EQ.1)THEN
       WRITE(6,465)DIST,Hcnew(I),PHI(I*K-1),PHI(I*K)
           ELSE
C       WRITE(6,465)DIST,HLnew(I),Hcnew(I),HRnew(I),PHI(I*K-3),
C     + PHI(I*K-2),PHI(I*K-1),PHI(I*K),QLnew(I),QRnew(I),
C     + AMTL(I),AMTR(I)
C
       WRITE(6,465)DIST,HLnew(I),Hcnew(I),HRnew(I),Qt(I),PHI(I*K-1),
     + QtF(I),QLnew(I),QRnew(I)
C
       ENDIF
450      CONTINUE
       ENDIF
       NTRY=0
       NTEST=0
C
460      FORMAT(/2X,'N',I6,2X,'NTRY= ',I2)
462      FORMAT(5X,'DIST',7x,'Hmc',7X,'Amc',9X,'Qmc')
C464       FORMAT(5X,'DIST',7X,'HL',9X,'Hmc',9X,'HR',10X,'AL',10X,'Amc',
C     + 9X,'Qmc',9X,'AR',9X,'QLF',9X,'QRF',11X,'MTL',9X,'MTR')
C
464      FORMAT(5X,'DIST',5X,'HL',11X,'Hmc',10X,'HR',8X,'Qtotal',
     +  7X,'Qmc',8X,'QtFlp',8X,'QLF',9X,'QRF')
C
465      FORMAT(2X,F6.1,11(2X,F10.4))
470      FORMAT(2X,'TIME = ',F10.4)
600     CONTINUE
700     STOP
C
      END
C
      SUBROUTINE INTERPO(X,Y1,T,NBCUS,UNK1,TM)
       IMPLICIT REAL *8(A-H,O-Z)
       DIMENSION X(200),Y1(200)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     + Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     + PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     + HRSTEP(200),CASEL(200),CASER(200),WALL
C
C        WRITE(6,1)T,TM
1       FORMAT(2X,'I AM IN SUBROUTINE INTERPO',2(2X,F10.6))
C
       TIME=T/TM
       IF(TIME.EQ.X(NBCUS))THEN
```

```
      UNK1= Y1(NBCUS)
      ELSEIF(TIME.GT.X(NBCUS))THEN
      WRITE(6,10)
10    FORMAT(2X,'EXTRAPOLATION HAS BEEN REQUESTED')
      ELSE
C      WRITE(6,15)X(1),Y1(1)
      DO 20 I=1,NBCUS-1
      IF(TIME.LT.X(I+1))THEN
      UNK1= Y1(I) + (TIME -X(I))*(Y1(I+1) - Y1(I))/(X(I+1) -X(I))
C      WRITE(6,15) X(I+1),Y1(I+1)
      GO TO 25
      ENDIF
20    CONTINUE
25      ENDIF
15    FORMAT(2X,4(2X,F10.3))
C
      RETURN
       END
C
     SUBROUTINE QNITIAL(UNK1,P2,P3,P4,HSTEP,TOL,N,COTR)
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION P2(200),P3(200),P4(200)
C
     COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
     COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
     COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
     COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
     COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
     COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
     COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
     COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
     COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
    + Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
    + PARR(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
    + HRSTEP(200),CASEL(200),CASER(200),WALL
C
C       WRITE(6,1)N,UNK1
1      FORMAT(2X,'I AM IN SUBROUTINE QNITIAL',I3,F10.4)

       Hcnew(1)=PHI(2)/PAR(1,2)
C
      I=0
2       I= I+1
      IF(I.GT.100)THEN
      WRITE(6,15)I
      COTR=1.0
      WRITE(6,10)N,X,Hcnew(1)
      GO TO 20
      ENDIF
      COTR=2.0
C
      PHI(2)=PAR(1,2)*Hcnew(1)
      IF(Hcnew(1).LT.HSTEP)THEN
      RC= PHI(2)/(PAR(1,2)+2.0*Hcnew(1))
       ELSE
      RC = PHI(2)/(PAR(1,2)+2.0*HSTEP)
      ENDIF
C
```

```
      A1=Hcnew(1) - HSTEP
      IF(A1.LT.0.0D+0)THEN
      AL=0.0D+00
      AR=0.0D+00
      RL=0.0D+00
      RR=0.0D+00
      GO TO 3
      ENDIF
C
      AL=PARF(1,2)*A1
      AR=PARL(1,2)*A1
      RL= AL/(PARF(1,2) +A1)
      RR=AR/(PARL(1,2) +A1)
C
3       ERROR=0.0D+00
      PHI(3)=PHI(2)*RC**0.66667*P2(1)**0.5/CN1(1)
      QL=AL*RL**0.66667*P3(1)**0.5/CN2(1)
      QR=AR*RR**0.66667*P4(1)**0.5/CN3(1)
C
      X=UNK1-PHI(3)-QL-QR
      Acnew(1)=PHI(2)
C
      ERROR=ABS(X)
      IF(ERROR.GT.0.001)THEN
      Hcnew(1)= Hcnew(1)+ 0.5*X
C         IF(N.EQ.510)THEN
C         WRITE(6,10)I X,Hcnew(1)
C         ENDIF
      GO TO 2
      ENDIF
      IF(Hcnew(1).LT.HSTEP)THEN
      HLX=0.0D+00
      HRX=0.0D+00
      ELSE
      HLX=AL/PARF(1,2)
      HRX=AR/PARL(1,2)
      ENDIF
C
C       IF(N.EQ.300)THEN
C       WRITE(6,10)I,UNK1,PHI(3),QL,QR
C       WRITE(6,10)I,Hcnew(1),HLX,HRX
C       ENDIF
10    FORMAT(2X,I2,4F15.4)
15    FORMAT(2X, 'NUMBER OF ITERATION EXCEEDS',I2)
20       RETURN
      END
C
C **** CALCULATING DEPTH AND VELOCITY FOR THE PURPOSE OF CALCULATING
DT****
C    -------------------------------------------------------
C
      SUBROUTINE CHPROP(IELNO,NITER,NTEST,T,DPHI,NTRY,MTD,K,P2,P3,P4,
     + KLP,NQc,JQF,ITAO,TAG1,TAG2,PET)
      IMPLICIT REAL *8(A-H,O-Z)
     DIMENSION DPHI(200),P2(200),P3(200),P4(200)
C
     COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
     COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
```

```
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     + PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     + HRSTEP(200),CASEL(200),CASER(200),WALL
C
C      REVISE PHI BY ADDING DPHI (DPHI=0 FOR FIRST TRIAL)
C       ----------------------------------------------
C
C        WRITE(6,1)
1       FORMAT(2X,'I AM IN SUBROUTINE CHPROP')
C
       IF((NTEST.EQ.0).AND.(T.EQ.0.0D+00))THEN
       DO   IELNO = 1,NELEM
       DO 5 J= 1,NELTYP(IELNO)
       CALL AREAS(IELNO,J,NQc,P2,KLP,TAG1,TAG2)
       IF(MTD.EQ.2)THEN
      HLnew(NODNUM(IELNO,J)) = ALM(NODNUM(IELNO,J))
      PHI((NODNUM(IELNO,J))*K-2) = Acnew(NODNUM(IELNO,J))
      PHI((NODNUM(IELNO,J))*K-1) = Qcnew(NODNUM(IELNO,J))
      HRnew(NODNUM(IELNO,J)) =  ARM(NODNUM(IELNO,J))
C
      Ucnew(NODNUM(IELNO,J)) = PHI((NODNUM(IELNO,J))*K-1)/
     +  PHI((NODNUM(IELNO,J))*K-2)
C
       PHI((NODNUM(IELNO,J))*K-3)= HLnew(NODNUM(IELNO,J))*
     + PARF((NODNUM(IELNO,J)),2) + Z3*HLnew(NODNUM(IELNO,J))**2/2.0
C
      ALnew(NODNUM(IELNO,J))=PHI((NODNUM(IELNO,J))*K-3)
       ZLF=DSQRT(1.0 + Z3**2)
C
       R=PHI((NODNUM(IELNO,J))*K-3)/(PARF((NODNUM(IELNO,J)),2) +
     + HLnew(NODNUM(IELNO,J))*ZLF)
C
       IF(R.EQ.0.0D+00)THEN
      VEL=0.0D+00
      QLnew(NODNUM(IELNO,J))=0.0D+00
      GO TO 11
      ENDIF
      IF(KLP.EQ.0)THEN
       CS= 5.75*DLOG10(R/CN2(IELNO)) + 6.2D+00
       VEL=CS*DSQRT(GRAV*R*P3(IELNO))
      ELSE
      VEL=R**0.667*P3(IELNO)**0.5/CN2(IELNO)
      ENDIF
      QLnew(NODNUM(IELNO,J))=PHI((NODNUM(IELNO,J))*K-3)*VEL
11      CONTINUE
C
       PHI((NODNUM(IELNO,J))*K)=HRnew(NODNUM(IELNO,J))*
     + PARL((NODNUM(IELNO,J)),2)+ Z4*HRnew(NODNUM(IELNO,J))**2/2.0
C
      ARnew(NODNUM(IELNO,J))=PHI((NODNUM(IELNO,J))*K)
       ZRF=DSQRT(1.0 + Z4**2)
```

```
C
       R=PHI((NODNUM(IELNO,J))*K)/(PARL((NODNUM(IELNO,J)),2) +
     + HRnew(NODNUM(IELNO,J))*ZRF)
C
       IF(R.EQ.0.0D+00)THEN
      VER=0.0D+00
      QRnew(NODNUM(IELNO,J))=0.0D+00
      GO TO 13
      ENDIF
C
      IF(KLP.EQ.0)THEN
       CS= 5.75*DLOG10(R/CN3(IELNO)) + 6.2D+00
       VER=CS*DSQRT(GRAV*R*P4(IELNO))
      ELSE
      VER=R**0.667*P4(IELNO)**0.5/CN3(IELNO)
      ENDIF
C
      QRnew(NODNUM(IELNO,J))=PHI((NODNUM(IELNO,J))*K)*VER
13      CONTINUE
C       WRITE(6,15)IELNO,J,PHI((NODNUM(IELNO,J))*K-3),
C     + PHI((NODNUM(IELNO,J))*K-2),PHI((NODNUM(IELNO,J))*K-1),
C     + PHI((NODNUM(IELNO,J))*K)
C
15      FORMAT(2X,2I3,2X,5(2X,F12.6))
C
        ELSE
      PHI((NODNUM(IELNO,J))*K-1) = Acnew(NODNUM(IELNO,J))
      PHI((NODNUM(IELNO,J))*K) = Qcnew(NODNUM(IELNO,J))
C
C       WRITE(6,15)IELNO,J,PHI((NODNUM(IELNO,J))*K-1),
C     + PHI((NODNUM(IELNO,J))*K)
      ENDIF
C
5       CONTINUE
C
         ELSE
C
      IF(NTRY.GE.NITER)THEN
      DO 220 I = 1,NNODES
       IF(MTD.EQ.1)THEN
      WRITE(6,21)PAR(I),PHI(I*K-1),PHI(I*K),DPHI(I*K-1),DPHI(I*K)
       ELSE
C      WRITE(6,21)PAR(I),PHI(I*K-3),PHI(I*K-2),PHI(I*K-1),
C     + PHI(I*K)
       WRITE(6,21)PAR(I),DPHI(1*K-3),DPHI(I*K-2),DPHI(I*K-1),
     + DPHI(I*K)
        ENDIF
220     CONTINUE
21      FORMAT(2X,F8.1,5(2X,F12.8))
        ENDIF
C
C       SET DPHI = 0
C--------------------------
        DO 400 I=1,NNODES*K
        DPHI(I)=0.0D+00
400      CONTINUE
        ENDIF
C
```

```
      IF(MTD.EQ.1)GO TO 24

      CALL FLOODPROP(IELNO,J,ITAO,NTEST)
C
24     CONTINUE
C
      IF(T.EQ.0.0D+00)THEN
       CALL OUTFLOW(JQF,TAG1,TAG2,T,PET)
       ENDIF
C
       RETURN
      END
C
       SUBROUTINE DEPTH(NTEST,T,K,MTD,TAG1,TAG2)
       IMPLICIT REAL *8(A-H,O-Z)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
       TIME =T/60.0
C        IF(TIME.GT.46.30)THEN
C        WRITE(6,1)
1       FORMAT(2X,'I AM IN SUBROUTINE DEPTH')
C        ENDIF
      IF((NTEST.EQ.0).AND.(T.EQ.0.0D+00))GO TO 10
C
      Z=Z1+Z2
C
      DO 5  IELNO=1,NELEM
      DO 5 J= 1, NELTYP(IELNO)
      IF(MTD.EQ.2)THEN
      ALnew(NODNUM(IELNO,J))  = PHI((NODNUM(IELNO,J))*K-3)
      Acnew(NODNUM(IELNO,J))  = PHI((NODNUM(IELNO,J))*K-2)
      Qcnew(NODNUM(IELNO,J))  = PHI((NODNUM(IELNO,J))*K-1)
      ARnew(NODNUM(IELNO,J))  = PHI((NODNUM(IELNO,J))*K)
      ELSE
      Acnew(NODNUM(IELNO,J))  = PHI((NODNUM(IELNO,J))*K-1)
      Qcnew(NODNUM(IELNO,J))  = PHI((NODNUM(IELNO,J))*K)
       ENDIF
C
       Ucnew(NODNUM(IELNO,J))  = Qcnew(NODNUM(IELNO,J))/
     +  Acnew(NODNUM(IELNO,J))
C
       IF(NTEST.EQ.0)THEN
      Qold(NODNUM(IELNO,J))=Qcnew(NODNUM(IELNO,J))
      Aold(NODNUM(IELNO,J))=Acnew(NODNUM(IELNO,J))
      Hold(NODNUM(IELNO,J))=Hcnew(NODNUM(IELNO,J))
       ENDIF
```

```
C
C
        IF(Z.EQ.0.0D+00)THEN
       Hcnew(NODNUM(IELNO,J))=Acnew(NODNUM(IELNO,J))/
     + PAR((NODNUM(IELNO,J)),2)
       GO TO 4
       ENDIF
C
        IF(IELNO.LT.TAG1)THEN
        HLSTEP(NODNUM(IELNO,2))=HLSTEP(1)
        ENDIF
C
        IF(IELNO.GT.TAG2)THEN
        HLSTEP(NODNUM(IELNO,1))=HLSTEP(1)
        ENDIF
C
        IF((IELNO.EQ.TAG1).OR.(IELNO.EQ.TAG2))THEN
        HLSTEP(NODNUM(IELNO,J))=WALL
        ENDIF

C
WRITE(6,15)IELNO,J,HLSTEP(NODNUM(IELNO,J)),Acnew(NODNUM(IELNO,J))
C
       T1= ELVLP(NODNUM(IELNO,J))+HLSTEP(NODNUM(IELNO,J))-
     +  ELVMc(NODNUM(IELNO,J))
C
       TOPWIDTH=PAR((NODNUM(IELNO,J)),2) + Z*T1
C
       A1=PAR((NODNUM(IELNO,J)),2)*T1 + Z*T1**2/2.0
C
       A2=Acnew(NODNUM(IELNO,J)) - A1
C         WRITE(6,15)IELNO,J,TOPWIDTH,A1,A2
C
        IF((A2.GT.0.0D+  ).AND.(HLSTEP(NODNUM(IELNO,J)).EQ.WALL))THEN
C
        A3= TOPWIDTH**2 + 2.0*(ABS(A2))*Z
C
        H1=(DSQRT(A3)-TOPWIDTH)/Z
C
        Hcnew(NODNUM(IELNO,J))= H1 + T1
       GO TO 4
       ENDIF
C
        IF(A2.LE.0.0D+00)THEN
        A3=PAR((NODNUM(IELNO,J)),2)**2 + 2.0*Acnew(NODNUM(IELNO,J))*Z

        Hcnew(NODNUM(IELNO,J))=(DSQRT(A3)-PAR((NODNUM(IELNO,J)),2))/Z
C
       GO TO 4
       ENDIF
       A4=A2/TOPWIDTH
       Hcnew(NODNUM(IELNO,J))= A4 + T1
4         CONTINUE
C        WRITE(6,15)IELNO,J,Acnew(NODNUM(IELNO,J)),Hcnew(NODNUM(IELNO,J))
15        FORMAT(2X,2I3,3(2X,F10.6))
5         CONTINUE
10        CONTINUE
C
```

```
      RETURN
      END
C
      SUBROUTINE AREAS(IELNO,J,NQc,P2,KLP,TAG1,TAG2)
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION P2(200)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C       WRITE(6,1)
1       FORMAT(2X,'I AM IN SUBROUTINE AREAS')
C
      HW=HLSTEP(1)
      Z= Z1 + Z2
      Hcnew(NODNUM(IELNO,J))= APHI(NODNUM(IELNO,J))
C
       IF(IELNO.GT.TAG2)THEN
       HLSTEP(NODNUM(IELNO,1))=HW
       ENDIF
C
       IF((IELNO.EQ.TAG1).OR.(IELNO.EQ.TAG2))THEN
       HLSTEP(NODNUM(IELNO,J))=WALL
C       WRITE(6,15)IELNO,J,WALL,HLSTEP(NODNUM(IELNO,J))
C
       T1= ELVLP(NODNUM(IELNO,J))+HLSTEP(NODNUM(IELNO,J))-
     + ELVMc(NODNUM(IELNO,J))

C
      IF(Hcnew(NODNUM(IELNO,J)).GT.T1)THEN
C
      A1=PAR((NODNUM(IELNO,J)),2)*T1 +Z*T1**2/2
C
      TOPW1=PAR((NODNUM(IELNO,J)),2) + Z*T1
C
       T2= Hcnew(NODNUM(IELNO,J)) - T1
C
      IF(WALL.EQ.0.0D+00)THEN
       A2=TOPW1*T2
       ELSE
       A2=TOPW1*T2 + Z*T2**2/2
      ENDIF
C
      Acnew(NODNUM(IELNO,J)) = A1 + A2
      ELSE
      GO TO 7
       ENDIF
C
```

```
      A1=1.0D+00 + Z1**2
      A2=DSQRT(A1)
      B1=1.0D+00 + Z2**2
      B2=DSQRT(B1)
C
      IF(WALL.EQ.0.0D+00)THEN
      P=PAR((NODNUM(IELNO,J)),2) +T1*(A2+B2)
      ELSE
      P=PAR((NODNUM(IELNO,J)),2) +T1*(A2+B2) +T2*B2
      ENDIF
C
C       WRITE(6,15)IELNO,J,Acnew(NODNUM(IELNO,J))
      GO TO 12
      ENDIF
C
7      T1= ELVLP(NODNUM(IELNO,J))+HLSTEP(NODNUM(IELNO,J))-
     + ELVMc(NODNUM(IELNO,J))
C
      IF(Hcnew(NODNUM(IELNO,J)).LT.T1)THEN
      T1=Hcnew(NODNUM(IELNO,J))
        ENDIF
      TOPWIDTH=PAR((NODNUM(IELNO,J)),2) + Z*T1
C
      T3=TOPWIDTH*(Hcnew(NODNUM(IELNO,J))-T1)
C
      Acnew(NODNUM(IELNO,J)) =PAR((NODNUM(IELNO,J)),2)*T1 +Z*T1**2/
     + 2.0 + T3
C
      A1=1.0D+00 + Z1**2
      A2=DSQRT(A1)
      B1=1.0D+00 + Z2**2
      B2=DSQRT(B1)
C
      P=PAR((NODNUM(IELNO,J)),2) +T1*(A2+B2)
C
12      R=Acnew(NODNUM(IELNO,J))/P
      IF(KLP.EQ.0)THEN

      A3=R/CN1(IELNO)
      CS= 5.75*DLOG10(A3) + 6.2D+00
      VEL=CS*DSQRT(GRAV*R*P2(IELNO))
      ELSE
      VEL=R**0.667*P2(IELNO)**0.5/CN1(IELNO)
      ENDIF
C       WRITE(6,15)IELNO,J,T1,TOPWIDTH,T3
C       WRITE(6,15)IELNO,J,P,R,VEL
C
      IF(NQc.EQ.1)THEN
      Qcnew(NODNUM(IELNO,J))=QPHI(NODNUM(IELNO,J))
      ELSE
      Qcnew(NODNUM(IELNO,J))=Acnew(NODNUM(IELNO,J))*VEL
      ENDIF
C
C       WRITE(6,15)IELNO,J,Acnew(NODNUM(IELNO,J)),
C     + Qcnew(NODNUM(IELNO,J))
15     FORMAT(2X,2I3,2X,3(2X,F12.6))
      RETURN
      END
```

```
C
      SUBROUTINE FLOODPROP(IELNO,J,ITAO,NTEST)
      IMPLICIT REAL *8(A-H,O-Z)
C
     COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
     COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
     COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
     COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
     COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
     COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
     COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
     COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
     COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA
    +Qt(200),QtF(200),VXL(200),V   00),CML(200),CMR(200),CF1,CF2,
    +PARF(200,2),PARL(200,2),D    .DXR(200),DXM(200),HLSTEP(200),
    +HRSTEP(200),CASEL(200),CA       0),WALL
C
C       WRITE(6,1)
1       FORMAT(2X,'I AM IN SUBROUTINE FLOODPROP')
C
     DO 24   IELNO=1,NELEM
     DO 24 J= 1, NELTYP(IELNO)
C     WRITE(6,20)IELNO,J,HLnew(NODNUM(IELNO,J)),HRnew(NODNUM(IELNO,J)),
C    + Hcnew(NODNUM(IELNO,J))
C
      IF(NTEST.EQ.0)THEN
     Qold(NODNUM(IELNO,J))=Qcnew(NODNUM(IELNO,J))
     Aold(NODNUM(IELNO,J))=Acnew(NODNUM(IELNO,J))
     Hold(NODNUM(IELNO,J))=Hcnew(NODNUM(IELNO,J))
      ENDIF
C
      CALL APPARENT(IELNO,J,ITAO)
C
19    AMTL(NODNUM(IELNO,J))=TAL(NODNUM(IELNO,J))*
    + HLnew(NODNUM(IELNO,J))/RHO
C
      AMTR(NODNUM(IELNO,J))=TAR(NODNUM(IELNO,J))*
    + HRnew(NODNUM(IELNO,J))/RHO
C
C      WRITE(6,20)IELNO,J,TAL(NODNUM(IELNO,J)),TAR(NODNUM(IELNO,J))
C      WRITE(6,20)IELNO,J,AMTL(NODNUM(IELNO,J)),AMTR(NODNUM(IELNO,J))
C
20    FORMAT(2X,2I3,3(2X,F12.6))
24    CONTINUE
C
25       RETURN
      END

      SUBROUTINE APPARENT(IELNO,J,ITAO)
      IMPLICIT REAL *8(A-H,O-Z)
C
     COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
     COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
     COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
     COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
     COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
     COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
     COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
```

```
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C         WRITE(6,1)
1        FORMAT(2X,'I AM IN SUBROUTINE APPARENT')
C
C        WRITE(6,20)IELNO,ITAO,QLnew(NODNUM(IELNO,J)),
C     + QRnew(NODNUM(IELNO,J))
C
      Z=Z1 + Z2
C
      TL1= ELVLP(NODNUM(IELNO,J))+HLSTEP(NODNUM(IELNO,J))-
     +  ELVMc(NODNUM(IELNO,J))
C
      IF(ITAO.EQ.0)THEN
       TAL(NODNUM(IELNO,J))=0.0D+00
       TAR(NODNUM(IELNO,J))=0.0D+00
       GO TO 19
       ENDIF
C
      BB=PAR((NODNUM(IELNO,J)),2) + Z*TL1
      T1=PAR((NODNUM(IELNO,J)),3)+BB + PAR((NODNUM(IELNO,J)),2)
C
       IF(TAA.EQ.1)THEN
      IF(HLnew(NODNUM(IELNO,J)).LE.0.001D+00)THEN
      TAL(NODNUM(IELNO,J))=0.0D+00
       GO TO 7
       ENDIF

C        T2=Qcnew(NODNUM(IELNO,J))/Acnew(NODNUM(IELNO,J)) -
C     + QLnew(NODNUM(IELNO,J))/ALnew(NODNUM(IELNO,J))
C
       T2=Qold(NODNUM(IELNO,J))/Aold(NODNUM(IELNO,J)) -
     + QLnew(NODNUM(IELNO,J))/ALnew(NODNUM(IELNO,J))
C
       T4=(Hcnew(NODNUM(IELNO,J))/HLnew(NODNUM(IELNO,J)))**1.129
       T5=(PAR((NODNUM(IELNO,J)),2)/
     + (T1-PAR((NODNUM(IELNO,J)),2)))**0.514
       A5=T2/DABS(T2)
       A6=DABS(T2)
       T6=A5*A6**0.92
      TAL(NODNUM(IELNO,J))=0.874*T4*T5*T6
C
7      IF(HRnew(NODNUM(IELNO,J)).LE.0.001D+00)THEN
       TAR(NODNUM(IELNO,J))=0.0D+00
       GO TO 19
       ENDIF
C
C        T3=Qcnew(NODNUM(IELNO,J))/Acnew(NODNUM(IELNO,J)) -
C     + QRnew(NODNUM(IELNO,J))/ARnew(NODNUM(IELNO,J))

       T3=Qold(NODNUM(IELNO,J))/Aold(NODNUM(IELNO,J)) -
     + QRnew(NODNUM(IELNO,J))/ARnew(NODNUM(IELNO,J))

       T4=(Hcnew(NODNUM(IELNO,J))/ARnew(NODNUM(IELNO,J)))**1.129
```

```
        A5=T3/DABS(T3)
        A6=DABS(T3)
        T6=A5*A6**0.92
        TAR(NODNUM(IELNO,J))=0.874*T4*T5*T6
C
       ELSEIF(TAA.EQ.2)THEN
C
       IF(HLnew(NODNUM(IELNO,J)).LE.0.001D+00)THEN
        TAL(NODNUM(IELNO,J))=0.0D+00
        GO TO 8
        ENDIF
C
C         T2=Qcnew(NODNUM(IELNO,J))/Acnew(NODNUM(IELNO,J)) -
C      + QLnew(NODNUM(IELNO,J))/ALnew(NODNUM(IELNO,J))
C
        T2=Qold(NODNUM(IELNO,J))/Aold(NODNUM(IELNO,J)) -
     + QLnew(NODNUM(IELNO,J))/ALnew(NODNUM(IELNO,J))

        T4=(1.0/HLnew(NODNUM(IELNO,J)))**0.354
        T5=(T1-PAR((NODNUM(IELNO,J)),2))**0.519
        A5=T2/DABS(T2)
        A6=DABS(T2)
        T6=A5*A6**1.451
       TAL(NODNUM(IELNO,J))=3.325*T4*T5*T6
C
8       IF(HRnew(NODNUM(IELNO,J)).LE.0.001D+00)THEN
        TAR(NODNUM(IELNO,J))=0.0D+00
        GO TO 19
        ENDIF
C
C         T3=Qcnew(NODNUM(IELNO.J))/Acnew(NODNUM(IELNO,J)) -
C      + QRnew(NODNUM(IELNO,J))/ARnew(NODNUM(IELNO,J))
C
        T3=Qold(NCDNUM(IELNO,J))/Aold(NODNUM(IELNO,J)) -
     + QRnew(NODNUM(IELNO,J))/ARnew(NODNUM(IELNO,J))
C
        T4=(1.0/HRnew(NODNUM(IELNO,J)))**0.354
        A5=T3/DABS(T3)
        A6=DABS(T3)
        T6=A5*A6**1.451
        TAR(NODNUM(IELNO,J))=3.325*T4*T5*T6
       ELSE
        CFA=0.01*T1/PAR((NODNUM(IELNO,J)),2)
C
       IF(HLnew(NODNUM(IELNO,J)).LE.0.001D+00)THEN
        TAL(NODNUM(IELNO,J))=0.0D+00
        GO TO 10
        ENDIF
C
C         T2=Qcnew(NODNUM(IELNO,J))/Acnew(NODNUM(IELNO,J)) -
C      + QLnew(NODNUM(IELNO,J))/ALnew(NODNUM(IELNO,J))
C
        T2=Qold(NODNUM(IELNO,J))/Aold(NODNUM(IELNO,J)) -
     + QLnew(NODNUM(IELNO,J))/ALnew(NODNUM(IELNO,J))
C
        TAL(NODNUM(IELNO,J))=0.5*RHO*CFA*T2**2

10      IF(HRnew(NODNUM(IELNO,J)).LE.0.001D+00)THEN
```

```
        TAR(NODNUM(IELNO,J))=0.0D+00
        GO TO 19
        ENDIF
C
C         T3=Qcnew(NODNUM(IELNO,J))/Acnew(NODNUM(IELNO,J)) -
C       + QRnew(NODNUM(IELNO,J))/ARnew(NODNUM(IELNO,J))
C
        T3=Qold(NODNUM(IELNO,J))/Aold(NODNUM(IELNO,J)) -
      + QRnew(NODNUM(IELNO,J))/ARnew(NODNUM(IELNO,J))
C
        TAR(NODNUM(IELNO,J))=0.5*RHO*CFA*T3**2
       ENDIF
19       CONTINUE
C        WRITE(6,20)IELNO,J,TAL(NODNUM(IELNO,J)),TAR(NODNUM(IELNO,J))
20      FORMAT(2X,2I3,3(2X,F12.6))
C
C
       RETURN
       END
C

C     ***** CALCULATING FLOOD PLAIN DISCHARGES (DURING INTERATIONS
********
C
       SUBROUTINE QFLPLAIN(T,K,NGP,P3,P4,KLP,TM,JQF,TAG1,TAG2)
       IMPLICIT REAL *8(A-H,O-Z)
        DIMENSION  DFIDS(2),FI(2),S(2),W(3),P3(200),P4(200)
C
       COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
       COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
       COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
       COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
       COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
       COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
       COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
       COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
       COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
      +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
      +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
      +HRSTEP(200),CASEL(200),CASER(200),WALL
C
       TIME=T/TM
C          IF(TIME.GT.46.30)THEN
C           WRITE(6,1)TIME
1        FORMAT(2X,'I AM IN SUBROUTINE QFLPLAIN',2X,F10.2)
C           ENDIF
C
         IF(T.EQ.0.0D+00) GO TO 45
         DO 5 I=1, NNODES
       IF(Z3.EQ.0.0D+00)THEN
       HLnew(I)=PHI(I*K-3)/PARF(I,2)
       ALnew(I)=PHI(I*K-3)
       ELSE
       A3=PARF(I,2)**2 + 2.0*PHI(I*K-3)*Z3
       HLnew(I)=(DSQRT(A3)-PARF(I,2))/Z3
       ALnew(I)=PHI(I*K-3)
       ENDIF
C
```

```
      IF(Z4.EQ.0.0D+00)THEN
      HRnew(I)=PHI(I*K)/PARL(I,2)
      ARnew(I)=PHI(I*K)
      ELSE
      A3=PARL(I,4)**2 + 2.0*PHI(I*K)*Z4
      HRnew(I)=(DSQRT(A3)-PARL(I,2))/Z4
      ARnew(I)=PHI(I*K)
      ENDIF
C         IF(TIME.GT.46.30)THEN
C         WRITE(6,15)I,PHI(I*K-3),PHI(I*K),HLnew(I),HRnew(I)
C         ENDIF
C        CONTINUE
C
        CALL OUTFLOW(JQF,TAG1,TAG2,T,PET)
C
        DO 40 I=1, NNODES
        IF(I.EQ.NNODES)THEN
        CN2(NNODES)=CN2(NNODES-1)
        CN3(NNODES)=CN3(NNODES-1)
        ENDIF
C
      DO  10 L=1,NGP
      CALL GAUSS(NGP,L,W,S)
      CALL SHAPE(L,S,FI,DFIDS)
10    CONTINUE
C
      R=PHI(I*K-3) /(PARF(I,2) + HLnew(I))
      IF(R.LE.0.0D+00)THEN
      VL=0.0D+00
      CONST=0.0D+00
      GO TO 12
      ENDIF
      A1=R/CN2(I)
      IF(KLP.EQ.0)THEN
      CS= 5.75*DLOG10(A1) + 6.2D+00
      VL=CS*DSQRT(GRAV*R*PHI(I*K-3))
      B1=GRAV*R
      CONST=CS*PHI(I*K-3)*DSQRT(B1)
      ELSE
      VL=R**0.6666667*P3(I)**0.5/CN2(I)
      CONST=PHI(I*K-3)*R**0.6666667/CN2(I)
      ENDIF
C
12      B1=GRAV*PHI(I*K-3)
       IF(B1.EQ.0.0D+00)THEN
       T2=0.0D+00
       T3=0.0D+00
       GO TO 23
       ENDIF
       T1=VXL(I)*QfL(I)
       T2=T1/B1
       T3=AMTL(I)/B1
C
C         IF((I.EQ.1).OR.(I.EQ.NNODES)) GO TO 25
23       IF((I.EQ.1).OR.(I.EQ.NNODES))THEN
C          QLnew(I)=VL*PHI(I*K-3)
       A2= P3(I) + T2 +T3
       A3=A2/DABS(A2)
```

```
       QLnew(I)=CONST*A3*DSQRT(DABS(A2))
C          WRITE(6,15)I,R,VL,PHI(I*K-3),CONST,QLnew(I)
      GO TO 25
      ENDIF
C
      SoL=P3(I)
      DHLdX=(HLnew(I-1)*DFIDS(1)+HLnew(I)*DFIDS(2))*2./DXL(I-1)
C        WRITE(6,15)I,DXL(I-1),dHLdX
C
      A2= SoL- DHLdX + T2 +T3
C
      A3=A2/DABS(A2)
      QLnew(I)=CONST*A3*DSQRT(DABS(A2))
C
25       R=PHI(I*K) /(PARL(I,2) + HRnew(I))
      IF(R.LE.0.0D+00)THEN
      VR=0.0D+00
      CONSTR=0.0D+00
      GO TO 26
      ENDIF
      A1=R/CN3(I)
      IF(F[illegible].EQ.0)THEN
      CS= 5.[illegible]5*DLOG10(A1) + 6.2D+00
      VR=CS*DSQRT(GRAV*R*PHI(I*K))
      B1=GRAV*R
      CONSTR=CS*PHI(I*K)*DSQRT(B1)
      ELSE
      VR=R**0.6666667*P4(I)**0.5/CN3(I)
      CONSTR=PHI(I*K)*R**0.6666667/CN3(I)
      ENDIF
C
26     B1=GRAV*PHI(I*K)
       IF(B1.EQ.0.0D+00)THEN
       T2=0.0D+00
       T3=0.0D+00
       GO TO 27
       ENDIF
       T1=VXR(I)*QfR(I)
       T2=T1 [illegible]
       T3=AM[illegible] B1
C
C        IF((I.EQ.1).OR.(I.EQ.NNODES)) GO TO 30
27         IF((I.EQ.1).OR.(I.EQ.NNODES))THEN
C           QRnew(I)=VR*PHI(I*K)
       A2= P4(I) + T2 +T3
       A3=A2/DABS(A2)
       QRnew(I)=CONSTR*A3*DSQRT(DABS(A2))
C         WRITE(6,15)I,R,VR,PHI(I*K),CONSTR,QRnew(I)
      GO TO 30
      ENDIF
C
      SoR=P4(I)
       DHRdX=(HRnew(I-1)*DFIDS(1)+HRnew(I)*DFIDS(2))*2./DXR(I-1)
C       WRITE(6,15)I,DXR(I-1),dHRdX
C
      A2= SoR-DHRdX + T2 +T3
C
      A3=A2/DABS(A2)
```

```
C
 15     FORMAT(2X,I3,5(2X,F12.6))
C
      QRnew(I)=CONSTR*A3*DSQRT(DABS(A2))
30     CONTINUE
C        IF(TIME.GT.46.30)THEN
C          WRITE(6,50)I,QLnew(I),QRnew(I)
C        ENDIF
40     CONTINUE
45      CONTINUE
50     FORMAT(2X,I3,4(2X,F15.6
       RETURN
       END
C
       SUBROUTINE OUTFLOW(JQF,TAG1,TAG2,T,PET)
       IMPLICIT REAL *8(A-H,O-Z)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
       TIME=T/60.0
C        IF(TIME.GT.48.0)THEN
C        WRITE(6,1)TIME
1       FORMAT(2X,'I AM IN SUBROUTINE OUTFLOW',3(2X,F10.3))
C        ENDIF
C
      HW=HLSTEP(1)
C
      DO 24  IELNO=1,NELEM
      DO 24 J= 1, NELTYP(IELNO)
C        IF(TIME.GT.46.0)THEN
C
WRITE(6,20)IELNO,J,HLnew(NODNUM(IELNO,J)),HRnew(NODNUM(IELNO,J)),
C     + Hcnew(NODNUM(IELNO,J))
C        ENDIF
C
       IF(IELNO.LT.TAG1)THEN
       HLSTEP(NODNUM(IELNO,2))=HW
       ENDIF
C
       IF(IELNO.GT.TAG2)THEN
       HLSTEP(NODNUM(IELNO,1))=HW
       ENDIF
C
       IF((IELNO.EQ.TAG1).OR.(IELNO.EQ.TAG2))THEN
       HLSTEP(NODNUM(IELNO,J))=WALL
       ENDIF
C
```

```
C       WRITE(6,20)IELNO,J,HLnew(NODNUM(IELNO,J)),HRnew(NODNUM(IELNO,J)),
C      + Hcnew(NODNUM(IELNO,J)),HLSTEP(NODNUM(IELNO,J))
C
        IF(JQF.EQ.0)THEN
       DHL(NODNUM(IELNO,J))=0.0D+00
       DHR(NODNUM(IELNO,J))=0.0D+00
       QfL(NODNUM(IELNO,J))=0.0D+00
       QfR(NODNUM(IELNO,J))=0.0D+00
       GO TO 24
         ENDIF
C
       IF(HLnew(NODNUM(IELNO,J)).LE.0.0D+00)THEN
        HLnew(NODNUM(IELNO,J))=0.0D+00
        ALnew(NODNUM(IELNO,J))=0.0D+00
        ENDIF
C
       IF(HRnew(NODNUM(IELNO,J)).LE.0.0D+00)THEN
        HRnew(NODNUM(IELNO,J))=0.0D+00
        ARnew(NODNUM(IELNO,J))=0.0D+00
        ENDIF
C
       T1=ELVMc(NODNUM(IELNO,J))  + Hcnew(NODNUM(IELNO,J))
C
       T2=ELVLP(NODNUM(IELNO,J))  + HLnew(NODNUM(IELNO,J))
C
       T3=ELVRP(NODNUM(IELNO,J))  + HRnew(NODNUM(IELNO,J))
C
       T4=ELVLP(NODNUM(IELNO,J))  + HLSTEP(NODNUM(IELNO,J))
C
       T5=ELVRP(NODNUM(IELNO,J))  + HRSTEP(NODNUM(IELNO,J))
C
       T1=ELVMc(NODNUM(IELNO,J))  + Hcnew(NODNUM(IELNO,J))
C
       A1=ELVMc(NODNUM(IELNO,J))  + HSTEP
       A2=T1-A1

C        IF(HLnew(NODNUM(IELNO,J)).EQ.0.0D+00) GO TO 3
C        IF(A2.EQ.0.0D+00) GO TO 3
C         IF((HLnew(NODNUM(IELNO,J))/A2).GT.0.75D+00)THEN
C          DHL(NODNUM(IELNO,J))=0.0D+00
C          GO TO 4
C          ENDIF
C
 3         IF((T1.GT.T2).AND.(T2.GE.T4))THEN
           DHL(NODNUM(IELNO,J))=T1 - T2
           CASEL(NODNUM(IELNO,J))=1
           CML(IELNO)=CF1
         ELSEIF((T2.GT.T1).AND.(T1.GE.T4))THEN
           DHL(NODNUM(IELNO,J))=-(T2 - T1)
           CASEL(NODNUM(IELNO,J))=2
            CML(IELNO)=CF2
         ELSEIF((T2.GT.T4).AND.(T1.LT.T4))THEN
            DHL(NODNUM(IELNO,J))=-(T2 - T4)
            CASEL(NODNUM(IELNO,J))=3
             CML(IELNO)=CF1
         ELSEIF((T1.GT.T4).AND.(T2.LT.T4))THEN
            DHL(NODNUM(IELNO,J))=T1 - T4
            CASEL(NODNUM(IELNO,J))=4
```

```
            CML(IELNO)=CF2
         ELSEIF((T1.EQ.T2).AND.(T1.GT.T4))THEN
            DHL(NODNUM(IELNO,J))=0.0D+00
            CASEL(NODNUM(IELNO,J))=5
            CML(IELNO)=0.0D+00
          ELSE
         DHL(NODNUM(IELNO,J))=0.0D+00
         CASEL(NODNUM(IELNO,J))=6
         CML(IELNO)=0.0D+00
         ENDIF
C
C            IF(DABS(DHL(NODNUM(IELNO,J))).LT.0.00001)THEN
C            DHL(NODNUM(IELNO,J))=0.0D+00
C              ENDIF
C
C4          IF(HRnew(NODNUM(IELNO,J)).EQ.0.0D+00) GO TO 5
C        IF(A2.EQ.0.0D+00) GO TO 5
C         IF((HRnew(NODNUM(IELNO,J))/A2).GT.0.75D+00)THEN
C          DHR(NODNUM(IELNO,J))=0.0D+00
C          GO TO 6
C          ENDIF

5           IF((T1.GT.T3).AND.(T3.GE.T5))THEN
          DHR(NODNUM(IELNO,J))=T1 - T3
          CASER(NODNUM(IELNO,J))=1
          CMR(IELNO)=CF1
        ELSEIF((T3.GT.T1).AND.(T1.GE.T5))THEN
          DHR(NODNUM(IELNO,J))=-(T3 - T1)
          CASER(NODNUM(IELNO,J))=2
           CMR(IELNO)=CF2
        ELSEIF((T3.GT.T5).AND.(T1.LT.T5))THEN
           DHR(NODNUM(IELNO,J))=-(T3 - T5)
           CASER(NODNUM(IELNO,J))=3
           CMR(IELNO)=CF1
        ELSEIF((T1.GT.T5).AND.(T3.LT.T5))THEN
           DHR(NODNUM(IELNO,J))=T1 - T5
           CASER(NODNUM(IELNO,J))=4
           CMR(IELNO)=CF2
        ELSEIF((T1.EQ.T3).AND.(T3.GT.T5))THEN
           DHR(NODNUM(IELNO,J))=0.0D+00
           CASER(NODNUM(IELNO,J))=5
        CMR(IELNO)=0.0D+00
         ELSE
        DHR(NODNUM(IELNO,J))=0.0D+00
        CASER(NODNUM(IELNO,J))=6
        CMR(IELNO)=0.0D+00
        ENDIF
C
C            IF(DABS(DHR(NODNUM(IELNO,J))).LT.0.00001)THEN
C            DHR(NODNUM(IELNO,J))=0.0D+00
C          ENDIF
6          CONTINUE
C
C            IF(TIME.GT.51.0)THEN
C
WRITE(6,20)IELNO,J,DHL(NODNUM(IELNO,J)),DHR(NODNUM(IELNO,J))
C           WRITE(6,20)IELNO,J,CML(IELNO),CMR(IELNO)
C           ENDIF
```

```
C
         CALL COEFDIS(IELNO,J,T)
C
        B1=2.0*GRAV
C
      IF(DHL(NODNUM(IELNO,J)).EQ.0.0D+00)THEN
       QfL(NODNUM(IELNO,J))=0.0D+00
       GO TO 11
       ELSE
       A1=DHL(NODNUM(IELNO,J))/DABS(DHL(NODNUM(IELNO,J)))
       A3=DABS(DHL(NODNUM(IELNO,J)))
       QfL(NODNUM(IELNO,J))=CML(IELNO)*2.0*DSQRT(B1)*A1*A3**1.5/3.0
       ENDIF
C
11      IF(DHR(NODNUM(IELNO,J)).EQ.0.0D+00)THEN
      QfR(NODNUM(IELNO,J))=0.0D+00
       GO TO 23
       ELSE
       A2=DHR(NODNUM(IELNO,J))/DABS(DHR(NODNUM(IELNO,J)))
       A4=DABS(DHR(NODNUM(IELNO,J)))
       QfR(NODNUM(IELNO,J))=CMR(IELNO)*2.0*DSQRT(B1)*A2*A4**1.5/3.0
       ENDIF
C
23      CONTINUE
C
       HC=Hcnew(NODNUM(IELNO,J))-HSTEP

        IF((PET.EQ.0).OR.(HC.LT.0.0D+00)) GO TO 24
C
       AS=8.41*DSQRT(GRAV)*CN2(IELNO)
       AKS= AS**6
       R=ALnew(NODNUM(IELNO,J)) /(PAR((NODNUM(IELNO,J)),3)+
     +  HLnew(NODNUM(IELNO,J)))
       R=HLnew(NODNUM(IELNO,J))
       AL=R/AKS
       IF(AL.LE.1.0D+0)THEN
       CSL=6.2D+00
       ELSE
       CSL=5.75*LOG(AL) + 6.2
       ENDIF
C
       AS=8.41*DSQRT(GRAV)*CN3(IELNO)
       AKS= AS**6
       R=ARnew(NODNUM(IELNO,J)) /(PAR((NODNUM(IELNO,J)),4)+
     +   HRnew(NODNUM(IELNO,J)))
       R=HRnew(NODNUM(IELNO,J))
       AR=R/AKS
       IF(AR.LE.1.0D+0)THEN
       CSR=6.2D+00
       ELSE
       CSR=5.75*LOG(AR) + 6.2
       ENDIF
C
      IF(TIME.GT.48.7D+00)THEN
        WRITE(6,20)IELNO,J,AS,AKS,R,AR
      WRITE(6,20)IELNO,J,CSL,CSR,DHL(NODNUM(IELNO,J)),P3(IELNO)
      ENDIF
C
```

```
      IF((HC.GT.0.00.D+00).AND.(HLnew(NODNUM(IELNO,J)).EQ.0.0D+00))THEN
     HLnew(NODNUM(IELNO,J))=0.001
     A2=HC**3.5
     A3=8.0*PAR((NODNUM(IELNO,J)),3)*DSQRT(P3(IELNO))*
    + HLnew(NODNUM(IELNO,J))
     A4=3.0*CSL*DSQRT(GRAV)/A3
     QfL(NODNUM(IELNO,J))=A4*A2
      ELSEIF((HC.GT.0.001D+00).AND.(HLnew(NODNUM(IELNO,J)).NE.0.0D+00)
    + .AND.(Hcnew(NODNUM(IELNO,J)).GT.HLnew(NODNUM(IELNO,J))))THEN
C
       IF(HC.GT.0.001D+00)THEN
       IF(HLnew(NODNUM(IELNO,J)).EQ.0.0D+00)THEN
       HLnew(NODNUM(IELNO,J))=0.001D+00
       ENDIF
     HLC=Hcnew(NODNUM(IELNO,J)) - (HSTEP + HLnew(NODNUM(IELNO,J)))
     A2=HLnew(NODNUM(IELNO,J))**1.5*HLC
     A3=PAR((NODNUM(IELNO,J)),3)*DSQRT(P3(IELNO))
     A4=4.0*CSL*DSQRT(GRAV)/A3
     QfR(NODNUM(IELNO,J))=A4*A2
     ENDIF
C
      IF((HC.GT.0.005D+00).AND.(HRnew(NODNUM(IELNO,J)).EQ.0.0D+00))THEN
     HRnew(NODNUM(IELNO,J))=0.001
     A2=HC**3.5
     A3=8.0*PAR((NODNUM(IELNO,J)),4)*DSQRT(P4(IELNO))*
    + HRnew(NODNUM(IELNO,J))
     A4=3.0*CSR*DSQRT(GRAV)/A3
     QfR(NODNUM(IELNO,J))=A4*A2
      ELSEIF((HC.GT.0.001D+00).AND.(HRnew(NODNUM(IELNO,J)).EQ.0.0D+00)
    + .AND.(Hcnew(NODNUM(IELNO,J )).GT.HRnew(NODNUM(IELNO,J))))THEN
C
      IF(HC.GT.0.001D+00)THEN
      IF(HRnew(NODNUM(IELNO,J)).EQ.0.0D+00)THEN
       HRnew(NODNUM(IELNO,J))=0.001D+00
       ENDIF
     HRC=Hcnew(NODNUM(IELNO,J)) - (HSTEP + HRnew(NODNUM(IELNO,J)))
     A2=HRnew(NODNUM(IELNO,J))**1.5*HRC
     A3=PAR((NODNUM(IELNO,J)),4)*DSQRT(P4(IELNO))
     A4=4.0*CSR*DSQRT(GRAV)/A3
     QfR(NODNUM(IELNO,J))=A4*A2
     ENDIF
24       CONTINUE
C    IF(TIME.GT.48.7D+00)THEN
C    WRITE(6,20)IELNO,J,QfL(NODNUM(IELNO,J)),QfR(NODNUM(IELNO,J))
C    ENDIF
25      CONTINUE
20    FORMAT(2X,2I3,4(2X,F12.6))
20    FORMAT(2X,2I3,3(2X,F12.6))
      RETURN
      END
C
     SUBROUTINE COEFDIS(IELNO,J,T)
     IMPLICIT REAL *8(A-H,O-Z)
C
     COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
     COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
     COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
     COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
```

```
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
      TIME=T/60.0
C        IF(TIME.GT.46.0)THEN
C        WRITE(6,1)TIME
1       FORMAT(2X,'I AM IN SUBROUTINE COEFDIS',F10.3)
C        ENDIF
C
         IF(DHL(NODNUM(IELNO,J)).EQ.0.0D+00)THEN
          VXL(NODNUM(IELNO,J))=0.0
          GO TO 5
          ENDIF
C
         IF(DHL(NODNUM(IELNO,J)).GT.0.0D+00)THEN
         VXL(NODNUM(IELNO,J))=Qcnew(NODNUM(IELNO,J))/
     +     Acnew(NODNUM(IELNO,J))
         ELSE
        VXL(NODNUM(IELNO,J))=QLnew(NODNUM(IELNO,J))/
     +    ALnew(NODNUM(IELNO,J))
        ENDIF
C
5         IF(DHR(NODNUM(IELNO,J)).EQ.0.0D+00)THEN
          VXR(NODNUM(IELNO,J))=0.0
          GO TO 10
          ENDIF
C
         IF(DHR(NODNUM(IELNO,J)).GT.0.0D+00)THEN
         VXR(NODNUM(IELNO,J))=Qcnew(NODNUM(IELNO,J))/
     +     Acnew(NODNUM(IELNO,J))
         ELSE
        VXR(NODNUM(IELNO,J))=QRnew(NODNUM(IELNO,J))/
     +    ARnew(NODNUM(IELNO,J))
        ENDIF
10        CONTINUE
C
C       WRITE(6,20)IELNO,J,VXL(NODNUM(IELNO,J))
C       WRITE(6,20)IELNO,J,F1L,H1,CML
C      WRITE(6,20)IELNO,J,VXR(NODNUM(IELNO,J))
C       WRITE(6,20)IELNO,J,F1R,H2,CMR
20     FORMAT(2X,2I3,3(2X,F12.6))
       RETURN
       END
C
C    ***** CALCULATING THE SLOPE BETWEEN ELEMENTS ********
C
      SUBROUTINE SLOPE(P2,P3,P4)
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION P2(200),P3(200),P4(200)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
```

```
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
      WRITE(6,1)
1       FORMAT(2X,'I AM IN SUBROUTINE SLOPE')
C
      DO 10  I=2,NNODES
      UP= ELVMc(I-1)
      DN= ELVMc(I)
      P2(I) = (UP-DN)/DXM(I-1)
      P2(1)=P2(2)
      P2(NNODES)=P2(NNODES-1)
C
      UP= ELVLP(I-1)
      DN= ELVLP(I)
      P3(I) = (UP-DN)/DXL(I-1)
      P3(1)=P3(2)
      P3(NNODES)=P3(NNODES-1)
C
      UP= ELVRP(I-1)
      DN= ELVRP(I)
      P4(I) = (UP-DN)/DXR(I-1)
      P4(1)=P4(2)
      P4(NNODES)=P4(NNODES-1)
C
       WRITE(6,20)I,P2(I),P3(I),P4(I)
10       CONTINUE
20       FORMAT(2X,I2,3(2X,F10.5))
       RETURN
        END
C
C **** ASSSEMBLING THE MATRICES **************
C     ES  ----- IS THE MASS MATRIX
C     EKA ----- IS THE LEFT STIFNESS MATRIX
C     EKB ------ IS THE RIGHT STIFNESS MATRIX
C      ----------------------------------------------
C
      SUBROUTINE ASSEMB(FCL,FCR,DT,NTEST,MTD,K,KUW,P2,P3,P4,
     + KLP,ES,KFL,NGP,T)
C
       IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION ESaa(2,2),ESaq(2,2),FE1(4),FE2(4),P2(200),ESqa(2,2),
     + ESqq(2,2),ES(8,8),EKaa(2,2),EKaq(2,2),EKqa(2,2),EKqq(2,2),
     + EK(8,8),EKA(8,8),EKB(8,8),ESL(8,8),ESR(8,8),SLaa(2,2),SRaa(2,2),
     + ALK(2,2),ARK(2,2),FL(4),FR(4),FCL(200),FCR(200),P3(200),P4(200)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
```

```
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C****** BEGIN LOOP OVER ALL ELEMENTS*********
C
       TIME=T/60.0
C         IF(TIME.GT.46.30)THEN
C          WRITE(6,1)
1        FORMAT(2X,'I AM IN SUBROUTINE ASSEMB')
C         ENDIF
C
      DO 500 IELNO = 1,NELEM
      DO 10 I = 1, K*NELTYP(IELNO)
      DO 10 J = 1, K*NELTYP(IELNO)
      ES(I,J) = 0.0D+00
      EK(I,J) = 0.0D+00
      EKA(I,J) = 0.0D+00
      ESL(I,J) = 0.0D+00
       IF(NTEST.EQ.1) GO TO 10
      ESR(I,J) = 0.0D+00
      EKB(I,J) = 0.0D+00
10       CONTINUE
C
      CALL INTEGRALS(IELNO,ESaa,ESaq,ESqa,ESqq,EKaa,EKaq,
     + EKqa,EKqq,SLaa,SRaa,ALK,ARK,P2,P3,P4,NGP,KUW,KLP,KFL,MTD)
C
         IF(MTD.EQ.1) GO TO 5
C
      CALL SOURCE(IELNO,FE1,FE2,FL,FR,P2,P3,P4,NGP,KUW,KLP)
C
C    *** COMPOUND CHANNEL MATRICES ****
C
         ES(1,1) = SLaa(1,1)
         ES(1,5) = SLaa(1,2)
C
         ES(2,2) = ESaa(1,1)
         ES(2,3) = ESaq(1,1)
         ES(2,6) = ESaa(1,2)
         ES(2,7) = ESaq(1,2)
C
         ES(3,2) = ESqa(1,1)
         ES(3,3) = ESqq(1,1)
         ES(3,6) = ESqa(1,2)
         ES(3,7) = ESqq(1,2)
C
         ES(4,4) = SRaa(1,1)
         ES(4,8) = SRaa(1,2)
C
         ES(5,1) = SLaa(2,1)
         ES(5,5) = SLaa(2,2)
C
         ES(6,2) = ESaa(2,1)
```

```
        ES(6,3) = ESaq(2,1)
        ES(6,6) = ESaa(2,2)
        ES(6,7) = ESaq(2,2)
C
        ES(7,2) = ESqa(2,1)
        ES(7,3) = ESqq(2,1)
        ES(7,6) = ESqa(2,2)
        ES(7,7) = ESqq(2,2)
C
        ES(8,4) = SRaa(2,1)
        ES(8,8) = SRaa(2,2)
C
        EK(1,1) = ALK(1,1)
        EK(1,5) = ALK(1,2)
C
        EK(2,2) = EKaa(1,1)
        EK(2,3) = EKaq(1,1)
        EK(2,6) = EKaa(1,2)
        EK(2,7) = EKaq(1,2)
C
        EK(3,2) = EKqa(1,1)
        EK(3,3) = EKqq(1,1)
        EK(3,6) = EKqa(1,2)
        EK(3,7) = EKqq(1,2)
C
        EK(4,4) = ARK(1,1)
        EK(4,8) = ARK(1,2)
C
        EK(5,1) = ALK(2,1)
        EK(5,5) = ALK(2,2)
C
        EK(6,2) = EKaa(2,1)
        EK(6,3) = EKaq(2,1)
        EK(6,6) = EKaa(2,2)
        EK(6,7) = EKaq(2,2)
C
        EK(7,2) = EKqa(2,1)
        EK(7,3) = EKqq(2,1)
        EK(7,6) = EKqa(2,2)
        EK(7,7) = EKqq(2,2)
C
        EK(8,4) = ARK(2,1)
        EK(8,8) = ARK(2,2)
C
C     ****** SINGLE CHANNEL MATRICES *****
C
       IF(MTD.NE.1) GO TO 17

5         ES(1,1) = ESaa(1,1)
        ES(1,2) = ESaq(1,1)
        ES(1,3) = ESaa(1,2)
        ES(1,4) = ESaq(1,2)
C
        ES(2,1) = ESqa(1,1)
        ES(2,2) = ESqq(1,1)
        ES(2,3) = ESqa(1,2)
        ES(2,4) = ESqq(1,2)
C
```

```
         ES(3,1)  = ESaa(2,1)
         ES(3,2)  = ESaq(2,1)
         ES(3,3)  = ESaa(2,2)
         ES(3,4)  = ESaq(2,2)
C
         ES(4,1)  = ESqa(2,1)
         ES(4,2)  = ESqq(2,1)
         ES(4,3)  = ESqa(2,2)
         ES(4,4)  = ESqq(2,2)
C
         EK(1,1)  = EKaa(1,1)
         EK(1,2)  = EKaq(1,1)
         EK(1,3)  = EKaa(1,2)
         EK(1,4)  = EKaq(1,2)
C
         EK(2,1)  = EKqa(1,1)
         EK(2,2)  = EKqq(1,1)
         EK(2,3)  = EKqa(1,2)
         EK(2,4)  = EKqq(1,2)
C
         EK(3,1)  = EKaa(2,1)
         EK(3,2)  = EKaq(2,1)
         EK(3,3)  = EKaa(2,2)
         EK(3,4)  = EKaq(2,2)
C
         EK(4,1)  = EKqa(2,1)
         EK(4,2)  = EKqq(2,1)
         EK(4,3)  = EKqa(2,2)
         EK(4,4)  = EKqq(2,2)
C
17         CONTINUE
C
C          DO 15 I= 1,K*NELTYP(IELNO)
C          DO 15 J = 1,K*NELTYP(IELNO)
C          WRITE(6,16)I,J,ES(I,J),EK(I,J)
15        CONTINUE
16       FORMAT(2X,2I3,2(2X,F12.6))
C
       DO 20 I= 1,K*NELTYP(IELNO)
       DO 20 J = 1,K*NELTYP(IELNO)
         EKA(I,J) = THETA*DT*(EK(I,J))
         IF(NTEST.EQ.1) GO TO 20
         EKB(I,J) = (1.0 - THETA)*DT*(EK(I,J))
20         CONTINUE
C
        DO 30 I= 1,K*NELTYP(IELNO)
        DO 30 J= 1,K*NELTYP(IELNO)
          ESL(I,J) = ES(I,J) + EKA(I,J)
         IF(NTEST.EQ.1) GO TO 30
         ESR(I,J) = ES(I,J) - EKB(I,J)
C           WRITE(6,34)I,J,ESL(I,J),ESR(I,J)
30         CONTINUE
34         FORMAT(2X,2I3,2(2X,F12.6))
C
        IF(IELNO.EQ.1)THEN
C
        DO 35 I = 1,NNODES*K
          FC(I) = 0.0D+00
```

```
           FCL(I)= 0.0D+00
          IF(NTEST.EQ.1) GO TO 35
           FCR(I)= 0.0D+00
 35        CONTINUE
C
         DO 40 I = 1,NNODES*K
         DO 40 J= 1,NNODES*K
          GSL(I,J) = 0.0D+00
         IF(NTEST.EQ.1) GO TO 40
          GSR(I,J) =  0.0D+00
 40       CONTINUE
C
          DO 50 I= 1,NELTYP(IELNO)*K
          DO 50 J =1,NELTYP(IELNO)*K
          GSL(I,J) = ESL(I,J)
          IF(NTEST.EQ.1) GO TO 50
          GSR(I,J) = ESR(I,J)
C           WRITE(6,55)I,J,GSR(I,J)
50         CONTINUE
55        FORMAT(2X,2I3,2X,F10.3)
C
           ELSEIF(IELNO.LE.NELEM)THEN
           NND=K*(IELNO+1)
           M=IELNO*K+1
C
          IF(MTD.EQ.1) GO TO 80
C
          DO 60 I= M,NND,4
          GSL(I-4,I) =  ESL(1,5)
          GSL(I-3,I+1) = ESL(2,6)
          GSL(I-3,I+2) = ESL(2,7)
          GSL(I-2,I+1) = ESL(3,6)
          GSL(I-2,I+2) = ESL(3,7)
          GSL(I-1,I+3) = ESL(4,8)
C
          GSL(I,I-4) = ESL(5,1)
          GSL(I-4,I-4) = GSL(I-4,I-4) + ESL(1,1)
          GSL(I+1,I-3) = ESL(6,2)
          GSL(I+1,I-2) = ESL(6,3)
C
          GSL(I-3,I-3) = GSL(I-3,I-3) + ESL(2,2)
          GSL(I-3,I-2) = GSL(I-3,I-2) + ESL(2,3)
          GSL(I+2,I-3) = ESL(7,2)
          GSL(I+2,I-2) = ESL(7,3)
C
          GSL(I-2,I-3) = GSL(I-2,I-3) + ESL(3,2)
          GSL(I-2,I-2) = GSL(I-2,I-2) + ESL(3,3)
          GSL(I+3,I-1) = ESL(8,4)
          GSL(I-1,I-1) = GSL(I-1,I-1) + ESL(4,4)
  60        CONTINUE
C
           GSL(NND-3,NND-3) = ESL(5,5)
           GSL(NND-2,NND-2) = ESL(6,6)
           GSL(NND-2,NND-1) = ESL(6,7)
           GSL(NND-1,NND-2) = ESL(7,6)
           GSL(NND-1,NND-1) = ESL(7,7)
           GSL(NND,NND) = ESL(8,8)
C
```

```
        IF(NTEST.EQ.1)GO TO 75
C
        DO 70 I= M,NND,4
        GSR(I-4,I) =  ESR(1,5)
        GSR(I-3,I+1) = ESR(2,6)
        GSR(I-3,I+2) = ESR(2,7)
        GSR(I-2,I+1) = ESR(3,6)
        GSR(I-2,I+2) = ESR(3,7)
        GSR(I-1,I+3) = ESR(4,8)
C
        GSR(I,I-4) = ESR(5,1)
        GSR(I-4,I-4) = GSR(I-4,I-4) + ESR(1,1)
        GSR(I+1,I-3) = ESR(6,2)
        GSR(I+1,I-2) = ESR(6,3)
C
        GSR(I-3,I-3) = GSR(I-3,I-3) + ESR(2,2)
        GSR(I-3,I-2) = GSR(I-3,I-2) + ESR(2,3)
        GSR(I+2,I-3) = ESR(7,2)
        GSR(I+2,I-2) = ESR(7,3)
C
        GSR(I-2,I-3) = GSR(I-2,I-3) + ESR(3,2)
        GSR(I-2,I-2) = GSR(I-2,I-2) + ESR(3,3)
        GSR(I+3,I-1) = ESR(8,4)
        GSR(I-1,I-1) = GSR(I-1,I-1) + ESR(4,4)
  70      CONTINUE
C
         GSR(NND-3,NND-3) = ESR(5,5)
         GSR(NND-2,NND-2) = ESR(6,6)
         GSR(NND-2,NND-1) = ESR(6,7)
         GSR(NND-1,NND-2) = ESR(7,6)
         GSR(NND-1,NND-1) = ESR(7,7)
         GSR(NND,NND) = ESR(8,8)
C
75          CONTINUE
C
        IF(MTD.NE.1)GO TO 105
C
80         DO 90 I= M,NND,2
        GSL(I-2,I) =  ESL(1,3)
        GSL(I-2,I+1) = ESL(1,4)
C
        GSL(I-1,I) = ESL(2,3)
        GSL(I-1,I+1) = ESL(2,4)
C
        GSL(I,I-2) = ESL(3,1)
        GSL(I,I-1) = ESL(3,2)
C
        GSL(I-2,I-2) = GSL(I-2,I-2) + ESL(1,1)
        GSL(I-2,I-1) = GSL(I-2,I-1) + ESL(1,2)
        GSL(I-1,I-2) = GSL(I-1,I-2) + ESL(2,1)
        GSL(I-1,I-1) = GSL(I-1,I-1) + ESL(2,2)
C
        GSL(I+1,I-2) = ESL(4,1)
        GSL(I+1,I-1) = ESL(4,2)
C
  90      CONTINUE
C
         GSL(NND-1,NND-1) = ESL(3,3)
```

```
          GSL(NND-1,NND) = ESL(3,4)
          GSL(NND,NND-1) = ESL(4,3)
          GSL(NND,NND) = ESL(4,4)
C
C               IF(IELNO.EQ.NELEM)THEN
C        WRITE(6,91)IELNO,GSL(NND-1,NND),GSL(NND,NND-1),
C      + GSL(NND,NND)
C91       FORMAT(2X,I2,3(2X,F8.  )
C              ENDIF
C
         IF(NTEST.EQ.1)GO TO 105
C
         DO 100 I= M,NND,2
         GSR(I-2,I) =  ESR(1,3)
         GSR(I-2,I+1) = ESR(1,4)
C
         GSR(I-1,I) = ESR(2,3)
         GSR(I-1,I+1) = ESR(2,4)
C
         GSR(I,I-2) = ESR(3,1)
         GSR(I,I-1) = ESR(3,2)
C
         GSR(I-2,I-2) = GSR(I-2,I-2) + ESR(1,1)
         GSR(I-2,I-1) = GSR(I-2,I-1) + ESR(1,2)
         GSR(I-1,I-2) = GSR(I-1,I-2) + ESR(2,1)
         GSR(I-1,I-1) = GSR(I-1,I-1) + ESR(2,2)
C
         GSR(I+1,I-2) = ESR(4,1)
         GSR(I+1,I-1) = ESR(4,2)
  100       CONTINUE
C
          GSR(NND-1,NND-1) = ESR(3,3)
          GSR(NND-1,NND) = ESR(3,4)
          GSR(NND,NND-1) = ESR(4,3)
          GSR(NND,NND) = ESR(4,4)
C
C             IF(IELNO.EQ.NELEM)THEN
C        WRITE(6,91)IELNO,GSR(NND-1,NND),GSR(NND,NND-1),
C      + GSR(NND,NND)
C91      FORMAT(2X,I2,3(2X,F8.3))
C             ENDIF
105         ENDIF
C
         IF(MTD.EQ.1)GO TO 110
C
C  *****APPLYING BOUNDARY TERMS AT THE FIRST BOUNDARY *******
C    FOR COMPOUND FLOW *****
C
        IF(KLP.EQ.1)THEN
         AX=5/3
         ELSE
         AX=3/2
         ENDIF
C
         DO 108 J= 1,NELTYP(IELNO)
          IF((IELNO.EQ.1).AND.(J.EQ.1))THEN
      IF((ALnew(NODNUM(IELNO,J)).LE.0.0).OR.(ARnew(NODNUM(IELNO,J))
     + .LE.0.0))THEN
```

```
        UL=0.0D+00
        UR=0.0D+00
        GO TO 117
        ELSE
       UL=QLnew(NODNUM(IELNO,J))/ALnew(NODNUM(IELNO,J))
       UR=QRnew(NODNUM(IELNO,J))/ARnew(NODNUM(IELNO,J))
        ENDIF
C
117        CONTINUE
C            WRITE(6,123)UL,UR
C
       GSL(1,1) = GSL(1,1) -THETA*DT*AX*UL
       GSL(2,3) = GSL(2,3) -1.0*THETA*DT
       GSL(3,2)= GSL(3,2)-THETA*DT*GRAV*Hcnew(NODNUM(IELNO,J))/2.
       GSL(3,3) = GSL(3,3) - THETA*DT*Ucnew(NODNUM(IELNO,J))
       GSL(4,4) = GSL(4,4) -THETA*DT*AX*UR
C
          IF(NTEST.EQ.1) GO TO 106
       GSR(1,1) = GSR(1,1) +(1.0-THETA)*DT*AX*UL
       GSR(2,3) = GSR(2,3) +1.0*(1.0-THETA)*DT
       GSR(3,2)=GSR(3,2)+(1.0-THETA)*DT*GRAV*Hcnew(NODNUM(IELNO,J))/2.
       GSR(3,3)=GSR(3,3) +(1.0-THETA)*DT*Ucnew(NODNUM(IELNO,J))
       GSR(4,4) = GSR(4,4) +(1.0-THETA)*DT*AX*UR
106         CONTINUE
          ENDIF
C      *** APPLYING BOUNDARY CONDITINS AT THE LAST BOUNDARY
C       FOR COMPOUND FLOW ****
C
            NND=K*(IELNO+1)
           IF((IELNO.EQ.NELEM).AND.(J.EQ.2))THEN
       IF((ALnew(NODNUM(IELNO,J)).LE.0.0).OR.(ARnew(NODNUM(IELNO,J))
     + .LE.0.0))THEN
        UL=0.0D+00
        UR=0.0D+00
        GO TO 127
        ELSE
       UL=QLnew(NODNUM(IELNO,J))/ALnew(NODNUM(IELNO,J))
       UR=QRnew(NODNUM(IELNO,J))/ARnew(NODNUM(IELNO,J))
        ENDIF
C
127         CONTINUE
C            WRITE(6,123)UL,UR
123        FORMAT(2X,2(2X,F12.6))
           GSL(NND-3,NND-3) = ESL(5,5) + THETA*DT*AX*UL
           GSL(NND-2,NND-1) = ESL(6,7) + 1.0*THETA*DT
           GSL(NND-1,NND-2) = ESL(7,6) + THETA*DT*GRAV*
     +  Hcnew(NODNUM(IELNO,J))/2.
C
        GSL(NND-1,NND-1)= ESL(7,7)+THETA*DT*
     +   Ucnew(NODNUM(IELNO,J))
C
        GSL(NND,NND) = ESL(8,8) + THETA*DT*AX*UR
C
          IF(NTEST.EQ.1) GO TO 107
         GSR(NND-3,NND-3) = ESR(5,5) - (1.0-THETA)*DT*AX*UL
         GSR(NND-2,NND-1) = ESR(6,7) - 1.0*(1.0-THETA)*DT
         GSR(NND-1,NND-2) = ESR(7,6) - (1.0-THETA)*DT*GRAV*
     +  Hcnew(NODNUM(IELNO,J))/2.
```

```
C
         GSR(NND-1,NND-1)=ESR(7,7)-(1.0-THETA)*DT*
     +  Ucnew(NODNUM(IELNO,J))
C
        GSR(NND,NND) = ESR(8,8) - (1.0-THETA)*DT*AX*UR
107       CONTINUE
         ENDIF
C       WRITE(6,109)IELNO,J,GSL(1,1),GSL(2,3),GSL(3,2),GSL(3,3),
C     + GSL(4,4),GSL(NND-3,NND-3),GSL(NND-2,NND-1),GSL(NND-1,NND-2),
C     + GSL(NND-1,NND-1),GSL(NND,NND)
108      CONTINUE
109     FORMAT(2X,2I3,10(2X,F12.6))
C
         IF(MTD.NE.1)GO TO 120
C
C ****APPLYING BOUNDARY TERMS AT THE FIRST BOUNDARY (SINLGE CHANNEL)
*****
C
110         DO 115 J= 1,NELTYP(IELNO)
           IF((IELNO.EQ.1).AND.(J.EQ.1))THEN
      GSL(1,2) = GSL(1,2) -1.0*THETA*DT
      GSL(2,1) = GSL(2,1) - THETA*DT*GRAV*Hcnew(NODNUM(IELNO,J))/2.
      GSL(2,2) = GSL(2,2) - THETA*DT*Ucnew(NODNUM(IELNO,J))
C
         IF(NTEST.EQ.1) GO TO 112
      GSR(1,2) = GSR(1,2) +1.0*(1.0-THETA)*DT
      GSR(2,1) = GSR(2,1) +(1.0-THETA)*DT*GRAV*Hcnew(NODNUM(IELNO,J))/2.
      GSR(2,2) = GSR(2,2) +(1.0-THETA)*DT*Ucnew(NODNUM(IELNO,J))
112      CONTINUE
         ENDIF
C *** APPLYING BOUNDARY CONDITINS AT THE LAST BOUNDARY (SINLGE
CHANNEL)****
C
          NND=K*(IELNO+1)
          IF((IELNO.EQ.NELEM).AND.(J.EQ.2))THEN
          GSL(NND-1,NND) = ESL(3,4) + 1.0*THETA*DT
          GSL(NND,NND-1) = ESL(4,3) + THETA*DT*GRAV*
     +  Hcnew(NODNUM(IELNO,J))/2.
      GSL(NND,NND) = ESL(4,4)+THETA*DT*Ucnew(NODNUM(IELNO,J))
C
         IF(NTEST.EQ.1) GO TO 114
          GSR(NND-1,NND) = ESR(3,4) - 1.0*(1.0-THETA)*DT
          GSR(NND,NND-1) = ESR(4,3) - (1.0-THETA)*DT*GRAV*
     +  Hcnew(NODNUM(IELNO,J))/2.
       GSR(NND,NND) = ESR(4,4) - (1.0-THETA)*DT*Ucnew(NODNUM(IELNO,J))
114       CONTINUE
         ENDIF
C        WRITE(6,116)IELNO,J,GSL(1,1),GSL(2,3),GSL(2,2),
C     + GSL(NND-1,NND),GSL(NND,NND-1),GSL(NND,NND)
115      CONTINUE
116     FORMAT(2X,2I3,6(2X,F10.3))
C
C
120     CONTINUE
C
       IF(MTD.EQ.1) GO TO 500
C
        IF(IELNO.EQ.1)THEN
```

```
C
        DO 130 I= 1,2
        FC(I*4-3) = FL(I)
        FC(I*4-2) = FE1(I)
        FC(I*4-1) = FE2(I)
        FC(I*4) = FR(I)
130       CONTINUE
C
         ELSEIF(IELNO.NE.1)THEN
          II=IELNO
          N=IELNO+1
C
         DO 145 I=II,N
         FC(I*4-3)=FC(I*4-3)+FL(1)
         FC(I*4-2)= FC(I*4-2)+FE1(1)
         FC(I*4-1)= FC(I*4-1)+FE2(1)
         FC(I*4)= FC(I*4)+FR(1)
145        CONTINUE
C
         FC(N*4-3)=FL(2)
         FC(N*4-2)= FE1(2)
         FC(N*4-1)=FE2(2)
         FC(N*4)= FR(2)
         ENDIF
500       CONTINUE
         DO 180 I =1,K*NNODES
         DO 180 J = 1,K*NNODES
C           WRITE(6,190)I,J,GSL(I,J),GSR(I,J)
180       CONTINUE
190      FORMAT(2X,'(',I5,1X,',',I5,')',2X,2(4X,F12.6))
C
C           DO 215 I=1,NNODES
C          IF(MTD.EQ.1)GO TO 210
C       WRITE(6,220)I,FC(I*4-3),FC(I*4-2),FC(I*4-1),FC(I*4)
C        IF(MTD.NE.1)GO TO 215
C210    WRITE(6,220)I,FC(I*2-1),FC(I*2)
215      CONTINUE
C
         Z = Z1 + Z2
C
         DO 230 I=1,NNODES*K
         FCL(I)= THETA*DT*FC(I)
        IF(NTEST.EQ.1) GO TO 224
         FCR(I)= -(1.0-THETA)*DT*FC(I)
224       CONTINUE
C225         WRITE(6,240)I,FCL(I),FCR(I)
230       CONTINUE
C
220       FORMAT(2X,I2,4(2X,F12.6))
240       FORMAT(2X,I2,4(2X,F12.6))
C
      RETURN
      END
C
C************ ASSEBLING THE GLOBAL MATRIX  OF THE JACOBIAN ****
C
       SUBROUTINE ASSJACOB(DT,MTD,K,KUW,P2,P3,P4,ES,KLP,KFL,NGP,HSTEP,T)
C
```

```
        IMPLICIT REAL *8(A-H,O-Z)
       DIMENSION P2(200),AJB(8,8),CJA(8,8),ES(8,8),P3(200),P4(200)
C
       COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
       COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
       COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
       COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
       COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
       COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
       COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
       COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
       COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
      + Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
      +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
      +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C****** BEGIN LOOP OVER ALL ELEMENTS*********
C
C         IF(K.EQ.4)THEN
C          WRITE(6,1)
1        FORMAT(2X,'I AM IN SUBROUTINE ASSJACOB')
C         ENDIF
C
       DO 500 IELNO = 1,NELEM
       DO 10 I = 1, K*NELTYP(IELNO)
       DO 10 J = 1, K*NELTYP(IELNO)
       AJB(I,J) = 0.0D+00
10       CONTINUE
C
       CALL JACOBIANS(IELNO,CJA,P2,P3,P4,NGP,MTD,KUW,KLP,KFL,K,HSTEP,T)
C
C          IF(IELNO.EQ.NELEM)THEN
C          DO 30 I = 1, K*NELTYP(IELNO)
C          DO 30 J = 1, K*NELTYP(IELNO)
C          WRITE(6,33) I,J,CJA(I,J)
30       CONTINUE
33       FORMAT(2X,2I3,4(2X,F10.6))
C         ENDIF
C
       DO 40 I = 1, K*NELTYP(IELNO)
       DO 40 J = 1, K*NELTYP(IELNO)
       CJA(I,J) =THETA*DT*CJA(I,J)
40       CONTINUE
C
C         WRITE(6,51)IELNO
       DO 50 I = 1, NELTYP(IELNO)*K
       DO 50 J = 1, NELTYP(IELNO)*K
       AJB(I,J) = ES(I,J) + CJA(I,J)
C         IF(IELNO.EQ.NELEM)THEN
C         WRITE(6,55) I,J,ES(I,J),CJA(I,J),AJB(I,J)
50       CONTINUE
C            ENDIF
51       FORMAT('ELEMENT NUMBER IS',I2,/)
55       FORMAT(2X,2I3,3(4X,F12.6))
C
         IF(IELNO.EQ.1)THEN
C
         DO 60 I = 1,NNODES*K
```

```
      DO 60 J= 1,NNODES*K
       GJC(I,J) =0.0D+00
 60    CONTINUE
C
       DO 70 I= 1,NELTYP(IELNO)*K
       DO 70 J =1,NELTYP(IELNO)*K
       GJC(I,J) = AJB(I,J)
 70     CONTINUE
C
        ELSEIF(IELNO.LE.NELEM)THEN
C
       NND=K*(IELNO+1)
       M= K*IELNO +1
C
      IF(MTD.EQ.1) GO TO 85
C
       DO 80 I= M,NND,4
       GJC(I-4,I) = AJB(1,5)
       GJC(I-4,I+1) = AJB(1,6)
       GJC(I-4,I+2) = AJB(1,7)
C
       GJC(I-3,I)   = AJB(2,5)
       GJC(I-3,I+1) = AJB(2,6)
       GJC(I-3,I+2) = AJB(2,7)
       GJC(I-3,I+3) = AJB(2,8)
C
       GJC(I-2,I) = AJB(3,5)
       GJC(I-2,I+1) = AJB(3,6)
       GJC(I-2,I+2) = AJB(3,7)
       GJC(I-2,I+3) = AJB(3,8)
C
       GJC(I-1,I+1) = AJB(4,6)
       GJC(I-1,I+2) = AJB(4,7)
       GJC(I-1,I+3) = AJB(4,8)
C
       GJC(I,I-4) = AJB(5,1)
       GJC(I,I-3) = AJB(5,2)
       GJC(I,I-2) = AJB(5,3)
       GJC(I-4,I-4) =  GJC(I-4,I-4) + AJB(1,1)
       GJC(I-4,I-3) =  GJC(I-4,I-3) + AJB(1,2)
       GJC(I-4,I-2) =  GJC(I-4,I-2) + AJB(1,3)
C
       GJC(I+1,I-4) = AJB(6,1)
       GJC(I+1,I-3) = AJB(6,2)
       GJC(I+1,I-2) = AJB(6,3)
       GJC(I+1,I-1) = AJB(6,4)
       GJC(I-3,I-4) = GJC(I-3,I-4) + AJB(2,1)
       GJC(I-3,I-3) = GJC(I-3,I-3) + AJB(2,2)
       GJC(I-3,I-2) = GJC(I-3,I-2) + AJB(2,3)
       GJC(I-3,I-1) = GJC(I-3,I-1) + AJB(2,4)
C
       GJC(I+2,I-4) = AJB(7,1)
       GJC(I+2,I-3) = AJB(7,2)
       GJC(I+2,I-2) = AJB(7,3)
       GJC(I+2,I-1) = AJB(7,4)
       GJC(I-2,I-4) = GJC(I-2,I-4) + AJB(3,1)
       GJC(I-2,I-3) = GJC(I-2,I-3) + AJB(3,2)
       GJC(I-2,I-2) = GJC(I-2,I-2) + AJB(3,3)
```

```
      GJC(I-2,I-1) = GJC(I-2,I-1) + AJB(3,4)
C
      GJC(I+3,I-3) = AJB(8,2)
      GJC(I+3,I-2) = AJB(8,3)
      GJC(I+3,I-1) = AJB(8,4)
      GJC(I-1,I-3) = GJC(I-1,I-3) + AJB(4,2)
      GJC(I-1,I-2) = GJC(I-1,I-2) + AJB(4,3)
      GJC(I-1,I-1) = GJC(I-1,I-1) + AJB(4,4)
  80    CONTINUE
C
       GJC(NND-3,NND-3) = AJB(5,5)
       GJC(NND-3,NND-2) = AJB(5,6)
       GJC(NND-3,NND-1) = AJB(5,7)
C
       GJC(NND-2,NND-3) = AJB(6,5)
       GJC(NND-2,NND-2) = AJB(6,6)
       GJC(NND-2,NND-1) = AJB(6,7)
       GJC(NND-2,NND) = AJB(6,8)
C
       GJC(NND-1,NND-3) = AJB(7,5)
       GJC(NND-1,NND-2) = AJB(7,6)
       GJC(NND-1,NND-1) = AJB(7,7)
       GJC(NND-1,NND) = AJB(7,8)
C
       GJC(NND,NND-2) = AJB(8,6)
       GJC(NND,NND-1) = AJB(8,7)
       GJC(NND,NND) = AJB(8,8)
C
      IF(MTD.NE.1) GO TO 490
C
85        DO 90 I= M,NND,2
      GJC(I-2,I) =  AJB(1,3)
      GJC(I-2,I+1) = AJB(1,4)
C
      GJC(I-1,I) = AJB(2,3)
      GJC(I-1,I+1) = AJB(2,4)
C
      GJC(I,I-2) = AJB(3,1)
      GJC(I,I-1) = AJB(3,2)
C
      GJC(I-2,I-2) = GJC(I-2,I-2) + AJB(1,1)
      GJC(I-2,I-1) = GJC(I-2,I-1) + AJB(1,2)
      GJC(I-1,I-2) = GJC(I-1,I-2) + AJB(2,1)
      GJC(I-1,I-1) = GJC(I-1,I-1) + AJB(2,2)
C
      GJC(I+1,I-2) = AJB(4,1)
      GJC(I+1,I-1) = AJB(4,2)
C
  90    CONTINUE
C
       GJC(NND-1,NND-1) = AJB(3,3)
       GJC(NND-1,NND) = AJB(3,4)
       GJC(NND,NND-1) = AJB(4,3)
       GJC(NND,NND) = AJB(4,4)
C
490       ENDIF
C
      IF(MTD.NE.1) GO TO 495
```

```
C
C *****APPLYING BOUNDARY TERMS AT THE FIRST BOUNDARY *****
C
          DO 108 J= 1,NELTYP(IELNO)
           IF((IELNO.EQ.1).AND.(J.EQ.1))THEN
        GJC(1,2) = GJC(1,2) -1.0*THETA*DT
        GJC(2,1) = GJC(2,1) -THETA*DT*(GRAV*Hcnew(NODNUM(IELNO,J)) -
      + Ucnew(NODNUM(IELNO,J))**2)
C
        GJC(2,2) = GJC(2,2) - THETA*DT*2.0*Ucnew(NODNUM(IELNO,J))
         ENDIF
C      *** APPLYING BOUNDARY CONDITINS AT THE LAST BOUNDARY ****
C
           IF((IELNO.EQ.NELEM).AND.(J.EQ.2))THEN
            NND=K*(IELNO+1)
           GJC(NND-1,NND) = AJB(3,4) + 1.0*THETA*DT
       GJC(NND,NND-1)=AJB(4,3)+THETA*DT*(GRAV*Hcnew(NODNUM(IELNO,J))-
      + Ucnew(NODNUM(IELNO,J))**2)
C
       GJC(NND,NND) = AJB(4,4)+THETA*DT*2.0*Ucnew(NODNUM(IELNO,J))
          ENDIF
C         WRITE(6,109)IELNO,J,GJC(1,2),GJC(2,1),GJC(2,2),
C      + GJC(NND-1,NND),GJC(NND,NND-1),GJC(NND,NND)
108       CONTINUE
109      FORMAT(2X,2I3,6(2X,F12.6))
C
        IF(MTD.EQ.1) GO TO 500
C
C  *****APPLYING BOUNDARY TERMS AT THE FIRST BOUNDARY *******
C    FOR COMPOUND FLOW *****
C
495        IF(KLP.EQ.1)THEN
          AX=5/3
          ELSE
          AX=3/2
          ENDIF
C
        DO 496 J= 1,NELTYP(IELNO)
        IF((IELNO.EQ.1).AND.(J.EQ.1))THEN
       IF((ALnew(NODNUM(IELNO,J)).EQ.0.0).OR.(ARnew(NODNUM(IELNO,J))
      + .EQ.0.0))THEN
        UL=0.0D+00
        UR=0.0D+00
        GO TO 117
        ELSE
       UL=QLnew(NODNUM(IELNO,J))/ALnew(NODNUM(IELNO,J))
       UR=QRnew(NODNUM(IELNO,J))/ARnew(NODNUM(IELNO,J))
        ENDIF
117     CONTINUE
       GJC(1,1) = GJC(1,1) -THETA*DT*AX*UL
       GJC(2,3) = GJC(2,3) -1.0*THETA*DT
       GJC(3,2)= GJC(3,2)-THETA*DT*(GRAV*Hcnew(NODNUM(IELNO,J))-
      + Ucnew(NODNUM(IELNO,J))**2)
       GJC(3,3) =GJC(3,3)-THETA*DT*2.0*Ucnew(NODNUM(IELNO,J))
       GJC(4,4) = GJC(4,4) -THETA*DT*AX*UR
          ENDIF
C      *** APPLYING BOUNDARY CONDITINS AT THE LAST BOUNDARY
C       FOR COMPOUND FLOW ****
```

```
C
           NND=K*(IELNO+1)
          IF((IELNO.EQ.NELEM).AND.(J.EQ.2))THEN
      IF((ALnew(NODNUM(IELNO,J)).EQ.0.0).OR.(ARnew(NODNUM(IELNO,J))
     + .EQ.0.0))THEN
       UL=0.0D+00
       UR=0.0D+00
       GO TO 127
       ELSE
      UL=QLnew(NODNUM(IELNO,J))/ALnew(NODNUM(IELNO,J))
      UR=QRnew(NODNUM(IELNO,J))/ARnew(NODNUM(IELNO,J))
       ENDIF
127   CONTINUE
          GJC(NND-3,NND-3) = AJB(5,5) + THETA*DT*AX*UL
          GJC(NND-2,NND-1) = AJB(6,7) + 1.0*THETA*DT
       GJC(NND-1,NND-2)=AJB(7,6)+THETA*DT*(GRAV*Hcnew(NODNUM(IELNO,J)) -
     + Ucnew(NODNUM(IELNO,J))**2)
C
       GJC(NND-1,NND-1)=AJB(7,7)+THETA*DT*2.0*Ucnew(NODNUM(IELNO,J))
C
       GJC(NND,NND) = AJB(8,8) + THETA*DT*AX*UR
         ENDIF
C        WRITE(6,499)IELNO,J,GJC(1,1),GJC(2,3),GJC(3,2),GJC(3,3),
C     +GJC(4,4),GJC(NND-3,NND-3),GJC(NND-2,NND-1),GJC(NND-1,NND-2),
C     + GJC(NND-1,NND-1),GJC(NND,NND)
496      CONTINUE
499     FORMAT(2X,2I3,10(2X,F12.6))
C
500      CONTINUE
C
C            DO 199 I =1,K*NNODES
C            DO 199 J = 1,K*NNODES
C            GJC(I,J)= GSL(I,J) + GJC(I,J)
C            WRITE(6,190)I,J,GJC(I,J),GSL(I,J)
199        CONTINUE
C
         DO 200 I =1,K*NNODES
         DO 200 J = 1,K*NNODES
C           WRITE(6,190)I,J,GJC(I,J)
200        CONTINUE
190       FORMAT(2X,'(',I5,1X,',',I5,')',2(2X,F15.10))
C
        RETURN
         END
C
C *****      CALCULATION OF THE RESIDUAL ***********
C
      SUBROUTINE RESIDUAL(FCL,FCR,RHS,TOL,DPHI,NITER,NTEST,
     + NTRY,MTD,K)
C
       IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION RES(200),RHS(200),DPHI(200),FCL(200),FCR(200)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
```

```
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3.Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C        WRITE(6,1)
1       FORMAT(2X,'I AM IN SUBROUTINE RESIDUAL',2X,F10.6)
C
       CALL MATMUL(FCR,FCL,RES,RHS,NTEST,MTD,K)
C
       CALL BOUND(RES,MTD)

        NEQ=NNODES*K

      DO 130 I=1,NEQ
        RES(I)= -RES(I)
C        WRITE(6,150)I,RES(I)
130      CONTINUE
150     FORMAT(2X,I4,E20.6)
C
      CALL SOLVE(GJC,RES.NEQ)
C
C       CHECK TO SEE IF DPHI=0.0
C  ------------------------------
         NTEST=0
        SUMA=0.0D+00
        SUMB=0.0D+00
        ERRMAX=0.0D+00
         DO 200 I=1,NEQ
         DPHI(I) = RES(I)
          IF(NTRY.GT.20)THEN
          PHI(I) = PHI(I) + 0.5*DPHI(I)
          TOL=0.001
         ELSE
         PHI(I) = PHI(I) + DPHI(I)
         TOL=0.00001
         ENDIF
C
         SUMA = SUMA + DPHI(I)**2
         SUMB = SUMB + PHI(I)**2
200       CONTINUE
C
         ERRMAX= SQRT(SUMA/SUMB)
C
      IF(ERRMAX.GT.TOL) NTEST=1
C
       IF(NTEST.NE.1) GO TO 350
C
       IF(NTRY.EQ.NITER) GO TO 250
        NTRY= NTRY+1
         RETURN
C
250       NTEST=2
C
        WRITE(6,260) NTRY
```

```
260   FORMAT('NTRY IS EQUAL TO NUMBER OF ITERATION',2X,I2)
C
350      CONTINUE
C
        RETURN
        END
C
C ***** MATRIX MULTIPLICATION **********
C
      SUBROUTINE MATMUL(FCR,FCL,RES,RHS,NTEST,MTD,K)
C
       IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION RES(200),FCL(200),FCR(200),RHS(200),AHS(200)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     + Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C        WRITE(6,1)
1        FORMAT(2X,'I AM IN SUBROUTINE MATMUL',2X,I2)
C
       N= NNODES*K
C
      DO 5 I=1,N
      AHS(I) = 0.0D+00
      RES(I)= 0.0D+00
       IF(NTEST.EQ.1)GO TO 5
      RHS(I) =0.0D+00
5        CONTINUE
C
       IF(MTD.EQ.2)THEN
       NBAND = 11
       ELSE
       NBAND = 6
       ENDIF
       L=NBAND/2
C
        DO 20 I=1,N
        AHS(I) = FCL(I)
       IF(NTEST.EQ.1)GO TO 20
        RHS(I) = FCR(I)
20       CONTINUE
C
       IF(NTEST.EQ.1) GO TO 60
        DO 50 I = 1,N
        DO 40 J = 1,N
        IF((ABS(J-I)).GT.L) GO TO 40
         RHS(I)= RHS(I) + GSR(I,J)*PHI(J)
40        CONTINUE
```

```
50        CONTINUE
C
60        DO 80 I = 1,N
        DO 70 J = 1,N
        IF((ABS(J-I)).GT.L) GO TO 70
         AHS(I)= AHS(I)+ GSL(I,J)*PHI(J)
70        CONTINUE
C           WRITE(6,81)I,RHS(I),AHS(I),PHI(I)
80        CONTINUE
81        FORMAT(2X,I3,3(2X,F12.6))
C
         DO 90 I=1,N
         RES(I) = AHS(I)-RHS(I)
C
90       CONTINUE
100      FORMAT(2X,I3,2X,F25.20)
C
       RETURN
       END
C
C********** PROGRAM TO DO LU MATRIX SOLVING ****************
C     -------------------------------------------------------
C
       SUBROUTINE SOLVE(AB,B,NEQ)
C
       IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION AB(200,200),B(200)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     + Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C         WRITE(6,1)
1        FORMAT(2X,'I AM IN SOLVE',2X,I2)
C
C     PERFORM THE LU DECOMPOSITION
C     ----------------------------
      DO 15 J=2,NEQ
       AB(J,1)=AB(J,1)/AB(1,1)
       DO 10 I=2,(J-1)
          SUML=0.0D+00
          SUMU=0.0D+00
          DO 5 M=1,(I-1)
             SUML=SUML+AB(J,M)*AB(M,I)
             SUMU=SUMU+AB(I,M)*AB(M,J)
    5        CONTINUE
          AB(J,I)=(AB(J,I)-SUML)/AB(I,I)
          AB(I,J)=AB(I,J)-SUMU
   10     CONTINUE
```

```
      SUMU=0.0D+00
      DO 12 M=1,(J-1)
      SUMU=SUMU+AB(J,M)*AB(M,J)
  12    CONTINUE
      AB(J,J)=AB(J,J)-SUMU
  15 CONTINUE
C
C     FORWARD SWEEP
C     -------------
     DO 30 I=2,NEQ
      SUML=0.0D+00
      DO 20 J=1,(I-1)
         SUML=SUML+AB(I,J)*B(J)
  20    CONTINUE
      B(I)=B(I)-SUML
  30 CONTINUE
C
C     BACKWARD SWEEP
C     --------------
     B(NEQ)=B(NEQ)/AB(NEQ,NEQ)
     DO 60 I=1,(NEQ-1)
      SUMU=0.0D+00
      J=NEQ-I
      DO 50 K=(J+1),NEQ
         SUMU=SUMU+AB(J,K)*B(K)
  50    CONTINUE
      B(J)=(B(J)-SUMU)/AB(J,J)
  60 CONTINUE
C
     RETURN
     END
C
C *****  BOUNDARY CONDITIONS  ***********
C
     SUBROUTINE BOUND(RES,MTD)
C
      IMPLICIT REAL *8(A-H,O-Z)
     DIMENSION RES(200)
C
     COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
     COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
     COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
     COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
     COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
     COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
     COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
     COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
     COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
    +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
    +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
    +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C       WRITE(6,1)
1       FORMAT(2X,'I AM IN SUBROUTINE BOUND')
C
C    *** PENALTY METHOD,IF Ac OR Qc MAY BE KNOWN AT THE BOUNDARIES ****
C
     IF(MTD.EQ.1)THEN
```

```
      N2= NNODES*2
      IF(IBC(2).EQ.0) GO TO 70
      GJC(1,1)=1.0D+30
      RES(1)=0.0D+00
C
70      IF(IBC(3).EQ.0) GO TO 80
      GJC(2,2)=1.0D+30
      RES(2)=0.0D+00
C
80      IF(IBC(6).EQ.0) GO TO 90
      GJC(N2-1,N2-1)=1.0D+30
      RES(N2-1)=0.0D+00
C
90      IF(IBC(7).EQ.0) GO TO 180
      GJC(N2,N2)=1.0D+30
      RES(N2)=0.0D+00
       ELSE
      N2= NNODES*4
      IF(IBC(1).EQ.0) GO TO 100
      GJC(1,1)=1.0D+30
      RES(1)=0.0D+00
C
100      IF(IBC(2).EQ.0) GO TO 110
      GJC(2,2)=1.0D+30
      RES(2)=0.0D+00
C
110      IF(IBC(3).EQ.0) GO TO 120
      GJC(3,3)=1.0D+30
      RES(3)=0.0D+00
C
120      IF(IBC(4).EQ.0) GO TO 130
      GJC(4,4)=1.0D+30
      RES(4)=0.0D+00
C
130      IF(IBC(5).EQ.0) GO TO 140
      GJC(N2-3,N2-3)=1.0D+30
      RES(N2-3)=0.0D+00
C
140      IF(IBC(6).EQ.0) GO TO 150
      GJC(N2-2,N2-2)=1.0D+30
      RES(N2-2)=0.0D+00
C
150      IF(IBC(7).EQ.0) GO TO 160
      GJC(N2-1,N2-1)=1.0D+30
      RES(N2-1)=0.0D+00
C
160      IF(IBC(8).EQ.0) GO TO 180
      GJC(N2,N2)=1.0D+30
      RES(N2)=0.0D+00
C
C        WRITE(6,170) GJC(2,2),RES(2),GJC(N2,N2),
C     + RES(N2)
170      FORMAT(2X,4(2X,F20.3))
180      ENDIF
       RETURN
       END
C
C****** SAMPLING POINT VALUES ******
```

```
      SUBROUTINE GAUSS(NGP,K,W,S)
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION W(3),S(3)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C       WRITE(6,1)
C1      FORMAT(2X,'I AM IN GAUSS')
C
      IF(NGP.EQ.2)THEN
      IF(K.EQ.1)THEN
       W(1)=1.0D+00
       S(1)= -0.577350269189626D+00
       ELSEIF(K.EQ.2)THEN
       W(2)=1.0D+00
       S(2)= 0.577350269189626D+00
       ENDIF
       ELSE
      IF(K.EQ.1)THEN
       W(1)= 0.555555555555556D+00
       S(1)= -0.774596669241483D+00
      ELSEIF(K.EQ.2)THEN
       W(2)= 0.888888888888889D+00
       S(2)= 0.0D+00
      ELSEIF(K.EQ.3)THEN
       W(3)= 0.555555555555556D+00
       S(3)= 0.774596669241483D+00
       ENDIF
       ENDIF
C       WRITE(6,10)W(K),S(K)
10     FORMAT(2X,F4.1,F20.15)
       RETURN
       END
C
       SUBROUTINE UPW(IELNO,KUW,Uold,HH,WK1,WK2,WK3,WK4)
C
       IMPLICIT REAL *8(A-H,O-Z)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
```

```
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C         WRITE(6,1)
1        FORMAT(2X,'I AM IN UPWMATRIX')
C
       A1=GRAV*HH
       C = DSQRT(A1)
       A = Uold + C
       B = Uold - C
       D= 1.0/(2.0*C)
       E= C**2 - Uold**2
C
C          WRITE(6,25) GRAV,HH,C,A,B,D,E
25       FORMAT(2X,7(2X,F20.15))
C
      Waa = D*(E*(1.0/DABS(A) -1.0/DABS(B)))
      Waq = D*(A/DABS(A) - B/DABS(B))
      Wqa = D*(E*(A/DABS(A) - B/DABS(B)))
      Wqq = D*(A**2/DABS(A) - B**2/DABS(B))
C
       WK1 = Waa*OMEGA*DXM(IELNO)/2.0
       WK2 = Waq*OMEGA*DXM(IELNO)/2.0
       WK3 = Wqa*OMEGA*DXM(IELNO)/2.0
       WK4 = Wqq*OMEGA*DXM(IELNO)/2.0
C
       IF(KUW.EQ.0)THEN
       WK1= 0.0D+00
       WK2= 0.0D+00
       WK3= 0.0D+00
       WK4= 0.0D+00
       ENDIF
C
C         WRITE(6,30)WK1,WK2,WK3,WK4
30       FORMAT(2X,5(2X,F10.6))
C
        RETURN
        END
C
C **** THIS SUBROUTINE CHANGES VALUES OF VARIABLES ******
C
      SUBROUTINE CHANGEVAR(IELNO,B,AA,QQ,dBdX,U,H,FI,AJa,AJq,
     + DFIDS,Uold,HH,P2,So,dHdX)
C
       IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION FI(2),DFIDS(2),P2(200)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
```

```
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C         WRITE(6,1)
1        FORMAT(2X,'I AM IN CHANGEVAR'2(2X,F10.6))
C
C
      AA=Acnew(NODNUM(IELNO,1))*FI(1)+Acnew(NODNUM(IELNO,2))*FI(2)
      QQ=Qcnew(NODNUM(IELNO,1))*FI(1)+Qcnew(NODNUM(IELNO,2))*FI(2)
C
      A=Aold(NODNUM(IELNO,1))*FI(1)+Aold(NODNUM(IELNO,2))*FI(2)
      Q=Qold(NODNUM(IELNO,1))*FI(1)+Qold(NODNUM(IELNO,2))*FI(2)
C
      dBdX=(PAR((NODNUM(IELNO,1)),2)*DFIDS(1) +
     + PAR((NODNUM(IELNO,2)),2)*DFIDS(2))*2./DXM(IELNO)
C
      B= PAR((NODNUM(IELNO,1)),2)*FI(1) +
     + PAR((NODNUM(IELNO,2)),2)*FI(2)
C
      So=P2(NODNUM(IELNO,1))*FI(1)+P2(NODNUM(IELNO,2))*FI(2)
C
      H=Hcnew(NODNUM(IELNO,1))*FI(1)+Hcnew(NODNUM(IELNO,2))*FI(2)
      HH=Hold(NODNUM(IELNO,1))*FI(1)+Hold(NODNUM(IELNO,2))*FI(2)
C
      dHdX=(Hcnew(NODNUM(IELNO,1))*DFIDS(1) +
     + Hcnew(NODNUM(IELNO,2))*DFIDS(2))*2./DXM(IELNO)
C
      U=QQ/AA
      Uold=Q/A
C
        A1=(Acnew(NODNUM(IELNO,1))*DFIDS(1) +
     + Acnew(NODNUM(IELNO,2))*DFIDS(2))*2./DXM(IELNO)
C
       A2=(Qcnew(NODNUM(IELNO,1))*DFIDS(1) +
     + Qcnew(NODNUM(IELNO,2))*DFIDS(2))*2./DXM(IELNO)
C
      C=DSQRT(GRAV*H)
      A3=C**2 + 2.0*U**2
      AJa=(A1*A3 - 2.0*U*A2)/AA
      AJq=2.0*(A2 -U*A1)/AA
C
C         WRITE(6,10)AA,QQ,dBdX,B,U,H,Uold,HH,So
C        WRITE(6,26) AJa,AJq
10     FORMAT(2X,10(2X,F12.6))
26      FORMAT(2X,5(2X,F10.6))
C
      RETURN
      END
C
C **** THIS SUBROUTINE CHANGES VALUES OF VARIABLES IN THE FLOOD PLAIN
******
C
       SUBROUTINE FLOODVAR(IELNO,AL,AR,QL,QR,BL,BR,HL,HR,SoL,SoR,
     + QQL,QQR,DDHL,DDHR,TL,TR,TML,TMR,FI,P3,P4,HLdX,HRdX,dTML,dTMR,
     + dQFL,dQFR,DFIDS,VL,VR)
C
       IMPLICIT REAL *8(A-H,O-Z)
       DIMENSION FI(2),DFIDS(2),P3(200),P4(200)
```

```
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C        WRITE(6,1)IELNO
1       FORMAT(2X,'I AM IN FLOODVAR FOR ELEMENT #',I2)
C
      AL=ALnew(NODNUM(IELNO,1))*FI(1)+ALnew(NODNUM(IELNO,2))*FI(2)
      AR=ARnew(NODNUM(IELNO,1))*FI(1)+ARnew(NODNUM(IELNO,2))*FI(2)
C
      QL=QLnew(NODNUM(IELNO,1))*FI(1)+QLnew(NODNUM(IELNO,2))*FI(2)
      QR=QRnew(NODNUM(IELNO,1))*FI(1)+QRnew(NODNUM(IELNO,2))*FI(2)
C
      BL=PARF((NODNUM(IELNO,1)),2)*FI(1)+PARF((NODNUM(IELNO,1)),2)*FI(2)
      BR=PARL((NODNUM(IELNO,1)),2)*FI(1)+PARL((NODNUM(IELNO,1)),2)*FI(2)
C
       HL=HLnew(NODNUM(IELNO,1))*FI(1)+HLnew(NODNUM(IELNO,2))*FI(2)
       HR=HRnew(NODNUM(IELNO,1))*FI(1)+HRnew(NODNUM(IELNO,2))*FI(2)
C
       SoL=P3(NODNUM(IELNO,1))*FI(1)+P3(NODNUM(IELNO,2))*FI(2)
C
       SoR=P4(NODNUM(IELNO,1))*FI(1)+P4(NODNUM(IELNO,2))*FI(2)
C
       DDHL=DHL(NODNUM(IELNO,1))*FI(1)+DHL(NODNUM(IELNO,2))*FI(2)
       DDHR= DHR(NODNUM(IELNO,1))*FI(1)+DHR(NODNUM(IELNO,2))*FI(2)
C
       QQL= QfL(NODNUM(IELNO,1))*FI(1)+QfL(NODNUM(IELNO,2))*FI(2)
       QQR= QfR(NODNUM(IELNO,1))*FI(1)+QfR(NODNUM(IELNO,2))*FI(2)
C
       TL= TAL(NODNUM(IELNO,1))*FI(1)+TAL(NODNUM(IELNO,2))*FI(2)
       TR= TAR(NODNUM(IELNO,1))*FI(1)+TAR(NODNUM(IELNO,2))*FI(2)
C
       TML= AMTL(NODNUM(IELNO,1))*FI(1)+AMTL(NODNUM(IELNO,2))*FI(2)
       TMR= AMTR(NODNUM(IELNO,1))*FI(1)+AMTR(NODNUM(IELNO,2))*FI(2)
C
      HLdX=(HLnew(NODNUM(IELNO,1))*DFIDS(1) +
     + HLnew(NODNUM(IELNO,2))*DFIDS(2))*2./DXL(IELNO)
C
      HRdX=(HRnew(NODNUM(IELNO,1))*DFIDS(1) +
     + HRnew(NODNUM(IELNO,2))*DFIDS(2))*2./DXR(IELNO)
C
      dTML=(AMTL(NODNUM(IELNO,1))*DFIDS(1) +
     + AMTL(NODNUM(IELNO,2))*DFIDS(2))*2./DXL(IELNO)
C
      dTMR=(AMTR(NODNUM(IELNO,1))*DFIDS(1) +
     + AMTR(NODNUM(IELNO,2))*DFIDS(2))*2./DXR(IELNO)
C
       VL=VXL(NODNUM(IELNO,1))*FI(1)+VXL(NODNUM(IELNO,2))*FI(2)
```

```
      VR=VXR(NODNUM(IELNO,1))*FI(1)+VXR(NODNUM(IELNO,2))*FI(2)
C
      dQFL=(QfL(NODNUM(IELNO,1))*DFIDS(1) +
     + QfL(NODNUM(IELNO,2))*DFIDS(2))*2./DXL(IELNO)
C
      dQFR=(QfR(NODNUM(IELNO,1))*DFIDS(1) +
     + QfR(NODNUM(IELNO,2))*DFIDS(2))*2./DXR(IELNO)
C
C        WRITE(6,10)AL,AR,BL,BR,QL,QR,HL,HR
C        WRITE(6,10)DDHL,DDHR,QQL,QQR,TL,TR,TML,TMR
C        WRITE(6,10)SoL,SoR
10    FORMAT(2X,8(2X,F10.6))
      RETURN
      END
C
C **** THIS SUBROUTINE CHANGES VALUES OF FRICTION f ******
C
       SUBROUTINE SHEAR(IELNO,B,AA,FF,U,H,KLP,KFL)
C
       IMPLICIT REAL *8(A-H,O-Z)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C         WRITE(6,1)
1        FORMAT(2X,'I AM IN SHEAR')
C
       T1= ELVLP(NODNUM(IELNO,1))-ELVMc(NODNUM(IELNO,1))
       IF(T1.GT.H)THEN
        T1=H
        ENDIF
C
       A1=1.0D+00 + Z1**2
       A2=DSQRT(A1)
       B1=1.0D+00 + Z2**2
       B2=DSQRT(B1)
C
       P= B + T1*(A2+B2)
       R=AA/P
C
       A3= R/CN1(IELNO)
       CS= 5.75*DLOG10(A3) + 6.2D+00
C
       IF(KLP.EQ.1)THEN
       FF= GRAV*DABS(U)*CN1(IELNO)**2/(R**1.333)
       ELSE
       FF= DABS(U)/(R*CS**2)
       ENDIF
```

```
C
      IF(KFL.EQ.0)THEN
      FF=0.0D+00
      ENDIF
C        WRITE(6,25)FF
25      FORMAT(2X,F10.6)
      RETURN
      END
C
C **** THIS SUBROUTINE CALCULATES VARIABLES ******
C
       SUBROUTINE INTEGRALS(IELNO,ESaa,ESaq,ESqa,ESqq,EKaa,EKaq,
     + EKqa,EKqq,SLaa,SRaa,ALK,ARK,P2,P3,P4,NGP,KUW,KLP,KFL,MTD)
C
       IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION FI(2),DFIDS(2),ESaa(2,2),ESaq(2,2),ESqa(2,2),ESqq(2,2),
     +EKaa(2,2),EKaq(2,2),EKqa(2,2),EKqq(2,2),W(3),S(3),SLaa(2,2),
     +SRaa(2,2),ALK(2,2),ARK(2,2),P2(200),P3(200),P4(200)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C        WRITE(6,1)IELNO,NGP
1     FORMAT(2X,'I AM IN INTEGRALS FOR ELEMENT # ',2X,I2,1X,I2)
C
      DO 10 I = 1, NELTYP(IELNO)
      DO 10 J = 1, NELTYP(IELNO)
      ESaa(I,J) = 0.0D+00
      ESaq(I,J) = 0.0D+00
      ESqa(I,J) = 0.0D+00
      ESqq(I,J) = 0.0D+00
      EKaa(I,J) = 0.0D+00
      EKaq(I,J) = 0.0D+00
      EKqa(I,J) = 0.0D+00
      EKqq(I,J) = 0.0D+00
      SLaa(I,J) = 0.0D+00
      SRaa(I,J) = 0.0D+00
      ALK(I,J) = 0.0D+00
      ARK(I,J) = 0.0D+00
10      CONTINUE
C
      DO  60 L=1,NGP
      CALL GAUSS(NGP,L,W,S)
      CALL SHAPE(L,S,FI,DFIDS)
C
      CALL CHANGEVAR(IELNO,B,AA,QQ,dBdX,U,H,FI,AJa,AJq,
     +DFIDS,Uold,HH,P2,So,dHdX)
C
```

```
      CALL UPW(IELNO,KUW,Uold,HH,WK1,WK2,WK3,WK4)
      CALL SHEAR(IELNO,B,AA,FF,U,H,KLP,KFL)
C
       IF(MTD.EQ.1) GO TO 30
C
      CALL FLOODVAR(IELNO,AL,AR,QL,QR,BL,BR,HL,HR,SoL,SoR,QQL,QQR,DDHL,
     + DDHR,TL,TR,TML,TMR,FI,P3,P4,HLdX,HRdX,dTML,dTMR,dQFL,dQFR,DFIDS,
     +VL,VR)
C
30     CONTINUE
C
      DO 50 I = 1, NELTYP(IELNO)
      DO 50 J = 1, NELTYP(IELNO)
C
      ESaa(I,J) = ESaa(I,J) + W(L)*(FI(I)*FI(J)*DXM(IELNO)/2.+
     + WK1*DFIDS(I)*FI(J))
C
      ESaq(I,J) = ESaq(I,J) + W(L)*WK2*DFIDS(I)*FI(J)
      ESqa(I,J) = ESqa(I,J) +  W(L)*WK3*DFIDS(I)*FI(J)
      ESqq(I,J) =  ESqq(I,J) + W(L)*(FI(I)*FI(J)*DXM(IELNO)/2.+
     + WK4*DFIDS(I)*FI(J))
C
      CALL ELMKaa(IELNO,I,J,FI,DFIDS,WK2,U,So,H,t)
C
      EKaa(I,J) = EKaa(I,J) + W(L)*t
C
      CALL ELMKaq(IELNO,I,J,FI,DFIDS,WK1,WK2,U,FF,t)
C
      EKaq(I,J) = EKaq(I,J) + W(L)*t
C
      CALL ELMKqa(IELNO,I,J,FI,DFIDS,WK4,U,So,H,t)
C
      EKqa(I,J) = EKqa(I,J) + W(L)*t
C
      CALL ELMKqq(IELNO,I,J,FI,DFIDS,WK3,WK4,U,FF,t)
C
      EKqq(I,J) = EKqq(I,J) + W(L)*t
C
      IF(MTD.EQ.1) GO TO 40
C
      SLaa(I,J) = SLaa(I,J) + W(L)*FI(I)*FI(J)*DXL(IELNO)/2.
      SRaa(I,J) = SRaa(I,J) + W(L)*FI(I)*FI(J)*DXR(IELNO)/2.
C
       CALL ELMALK(IELNO,I,J,AL,QL,BL,HL,QQL,TML,FI,DFIDS,
     +t,KLP,VL)
C
      ALK(I,J) = ALK(I,J) + W(L)*t
C
       CALL ELMARK(IELNO,I,J,AR,QR,BR,HR,QQR,TMR,FI,DFIDS,
     +t,KLP,VR)
C
      ARK(I,J) = ARK(I,J) + W(L)*t
C
40     CONTINUE
C
C          IF(L.EQ.NGP)THEN
C          WRITE(6,70)I,J,ESaa(I,J),ESaq(I,J),ESqa(I,J),ESqq(I,J)
C           WRITE(6,70)I,J,EKaa(I,J),EKaq(I,J),EKqa(I,J),EKqq(I,J)
```

```
C          WRITE(6,70)I,J,SLaa(I,J),SRaa(I,J),ALK(I,J),ARK(I,J)
C          ENDIF
50      CONTINUE
60      CONTINUE
70        FORMAT(2X,2I3,4(2X,F12.6))
       RETURN
       END
C
C **** THIS SUBROUTINE CALCULATES VARIABLES ******
C
       SUBROUTINE SOURCE(IELNO,FE1,FE2,FL,FR,P2,P3,P4,NGP,KUW,KLP)
C
       IMPLICIT REAL *8(A-H,O-Z)
       DIMENSION FI(2),DFIDS(2),FL(4),W(3),FE1(4),FE2(4),P2(200),
     + S(3),FR(4),P3(200),P4(200)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C         WRITE(6,1)IELNO,NGP
1     FORMAT(2X,'I AM IN SOURCE FOR ELEMENT # ',2X,I2,1X,I2)
C
      DO 10 J = 1, 2*NELTYP(IELNO)
      FE1(J)= 0.0D+00
      FE2(J)= 0.0D+00
      FL(J)= 0.0D+00
      FR(J)= 0.0D+00
10        CONTINUE
C
      DO    30 L=1,NGP
      CALL GAUSS(NGP,L,W,S)
      CALL SHAPE(L,S,FI,DFIDS)
C
       CALL CHANGEVAR(IELNO,B,AA,QQ,dBdX,U,H,FI,AJa,AJq,
     +DFIDS,Uold,HH,P2,So,dHdX)
C
       CALL UPW(IELNO,KUW,Uold,HH,WK1,WK2,WK3,WK4)
C
      CALL FLOODVAR(IELNO,AL,AR,QL,QR,BL,BR,HL,HR,SoL,SoR,QQL,QQR,DDHL,
     + DDHR,TL,TR,TML,TMR,FI,P3,P4,HLdX,HRdX,dTML,dTMR,dQFL,dQFR,DFIDS,
     + VL,VR)
C
        DO 20 I=1,NELTYP(IELNO)
C
      CALL FMC(IELNO,I,QQL,QQR,FI,DFIDS,WK1,WK2,WK3,WK4,TML,TMR,R1,R2,
     + dBdX,H,dHdX,VL,VR)
C
          FE1(I)= FE1(I) + W(L)*R1
```

```
           FE2(I)= FE2(I)  + W(L)*R2
C
      CALL FLK(IELNO,I,AL,QL,BL,HL,FI,DFIDS,QQL,t,KLP,HLdX)
C
          FL(I)= FL(I)  + W(L)*t
C
      CALL FRK(IELNO,I,AR,QR,BR,HR,FI,DFIDS,QQR,t,KLP,HRdX)
C
           FR(I)= FR(I)  + W(L)*t
C
C         IF(L.EQ.NGP)THEN
C          WRITE(6,50)I,FE1(I),FE2(I),FL(I),FR(I)
C           ENDIF
20       CONTINUE
30       CONTINUE
50        FORMAT(2X,I2,4(2X,F12.6))
       RETURN
       END
C
C **** THIS SUBROUTINE CALCULATES  JACOBIAN INTEGRALS******
C
      SUBROUTINE JACOBIANS(IELNO,CJE,P2,P3,P4,NGP,MTD,KUW,KLP,KFL,K,
     + HSTEP,XT)
C
       IMPLICIT REAL *8(A-H,O-Z)
       DIMENSION CJA(8,8),CJB(8,8),CJC(8,8),CJD(8,8),CJE(8,8),
     + W(3),P2(200),S(3),FI(2),DFIDS(2),P3(200),P4(200)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C         IF(K.EQ.4)THEN
C        WRITE(6,1)IELNO,MTD
1     FORMAT(2X,'I AM IN JACOBIANS FOR ELEMENT # ',2X,I2,1X,I2)
C       ENDIF
C
      DO 10 I = 1, K*NELTYP(IELNO)
      DO 10 J = 1, K*NELTYP(IELNO)
      CJA(I,J) = 0.0D+00
      CJB(I,J) = 0.0D+00
      CJC(I,J) = 0.0D+00
      CJD(I,J) = 0.0D+00
      CJE(I,J) = 0.0D+00
10        CONTINUE
C
       DO  60 L=1,NGP
      CALL GAUSS(NGP,L,W,S)
      CALL SHAPE(L,S,FI,DFIDS)
```

```
C
        CALL CHANGEVAR(IELNO,B,AA,QQ,dBdX,U,H,FI,AJa,AJq,
     +DFIDS,Uold,HH,P2,So,dHdX)
C
        CALL UPW(IELNO,KUW,Uold,HH,WK1,WK2,WK3,WK4)
C
        IF(MTD.EQ.1) GO TO 55
C
       CALL FLOODVAR(IELNO,AL,AR,QL,QR,BL,BR,HL,HR,SoL,SoR,QQL,QQR,DDHL,
     + DDHR,TL,TR,TML,TMR,FI,P3,P4,HLdX,HRdX,dTML,dTMR,dQFL,dQFR,DFIDS,
     + VL,VR)
C
C  *** THIS PART OF THE DERIVATIVE IS FROM THE CONSERVATIVE EQUATIONS
(F1i) ***
C
        CALL DERIVF(IELNO,AL,AR,QL,QR,BL,BR,HL,HR,QQ,AA,H,
     +B,tAL1,tAL2,dF1dA,dF1dQ,dF1dAR,dF2dAL,dF2dA,dF2dQ,dF2dAR,dF3dAL,
     +dF3dA,dF3dQ,dF3dAR,dF4dAL,dF4dA,dF4dQ,tAR1,tAR2,KLP,TML,TMR,
     +QQR,QQL,XT,VL,VR)
C
C  *** THIS PART OF THE DERIVATIVE IS FROM THE NON-CONSERVATIVE
EQUATIONS (F2i) ***
C
       CALL DERIVD(U,H,B,AA,dD1dAL,dD1dA,dD1dQ,dD1dAR,dD2dAL,dD2dA,dD2dQ,
     + dD2dAR,dD3dAL,dD3dA,dD3dQ,dD3dAR,dD4dAL,dD4dA,dD4dQ,dD4dAR,XT)
C
C  *** THIS PART OF THE DERIVATIVE IS FROM THE CONSERVATIVE EQUATIONS
(G1i) ***
C
       CALL DERIVG(IELNO,So,QQ,AA,B,dBdX,KFL,KLP,AL,AR,QL,QR,BL,BR,
     +HL,HR,dG1dAL,dG1dA,dG1dQ,dG1dAR,dG2dAL,dG2dA,dG2dQ,dG2dAR,dG3dAL,
     +dG3dA,dG3dQ,dG3dAR,TML,TMR,dG4dAL,dG4dA,dG4dQ,dG4dAR,DDHL,DDHR,
     +QQR,QQL,HSTEP,VL,VR,L)
C
C  *** THIS PART OF THE DERIVATIVE IS FROM THE NON-CONSERVATIVE
EQUATIONS (G2i) ***
C
       CALL DERIVE(IELNO,So,QQ,AA,B,dBdX,KFL,KLP,AL,AR,QL,QR,BL,BR,
     +HL,HR,dE1dAL,dE1dA,dE1dQ,dE1dAR,dE2dAL,dE2dA,dE2dQ,dE2dAR,dE3dAL,
     +dE3dA,dE3dQ,dE3dAR,TML,TMR,dE4dAL,dE4dA,dE4dQ,dE4dAR,DDHL,DDHR,
     +QQR,QQL,HSTEP,VL,VR,L)
C
       CALL PART1(IELNO,L,W,FI,DFIDS,tAL1,tAL2,dF1dA,dF1dQ,
     + dF1dAR,dF2dAL,dF2dA,dF2dQ,dF2dAR,dF3dAL,dF3dA,dF3dQ,
     + dF3dAR,dF4dAL,dF4dA,dF4dQ,tAR1,tAR2,CJA)
C
       CALL PART2(IELNO,L,W,WK1,WK2,WK3,WK4,DFIDS,
     + dD2dAL,dD2dA,dD2dQ,dD2dAR,dD3dAL,dD3dA,
     + dD3dQ,dD3dAR,CJB)
C
        CALL PART3(IELNO,L,W,FI,dG1dAL,dG1dA,dG1dQ,
     + dG1dAR,dG2dAL,dG2dA,dG2dQ,dG2dAR,dG3dAL,dG3dA,dG3dQ.
     + dG3dAR,dG4dAL,dG4dA,dG4dQ,dG4dAR,CJC)
C
       CALL PART4(L,W,FI,DFIDS,dE2dAL,dE2dA,
     + dE2dQ,dE2dAR,dE3dAL,dE3dA,dE3dQ,
     + dE3dAR,WK1,WK2,WK3,WK4,CJD)
C
```

```
        IF(L.EQ.NGP)THEN
         DO 30 I=1,8
         DO 30 J=1,8
       CJE(I,J)=CJA(I,J)+CJB(I,J)+CJC(I,J)+CJD(I,J)
C        WRITE(6,50)I,J,CJA(I,J),CJB(I,J),CJC(I,J),CJD(I,J),CJE(I,J)
30      CONTINUE
       ENDIF
50        FORMAT(2X,2I3,2X,5(2X,F12.6))
C
       IF(MTD.EQ.2) GO TO 60
C
55     CALL DERIVB(IELNO,So,dBdX,QQ,AA,U,H,B,KLP,KFL,dF1dA,
     + dF1dQ,dF2dA,dF2dQ,dG1dA,dG1dQ,dG2dA,dG2dQ)
C
      CALL DERIVP(IELNO,So,dBdX,QQ,AA,U,H,B,KLP,KFL,dB1dA,
     + dB1dQ,dB2dA,dB2dQ,dE1dA,dE1dQ,dE2dA,dE2dQ)
C
       CALL MCJBIANS(IELNO,L,W,CJE,FI,DFIDS,dF1dA,dF1dQ,
     + dF2dA,dF2dQ,dG1dA,dG1dQ,dG2dA,dG2dQ,dB1dA,dB1dQ,dB2dA,
     + dB2dQ,dE1dA,dE1dQ,dE2dA,dE2dQ,WK1,WK2,WK3,WK4)
C
60      CONTINUE
         RETURN
          END
C
C    *** THE JACOBIAN FOR THE MAIN CHANNEL ONLY ******
C
       SUBROUTINE MCJBIANS(IELNO,K,W,CJA,FI,DFIDS,dF1dA,dF1dQ,
     + dF2dA,dF2dQ,dG1dA,dG1dQ,dG2dA,dG2dQ,dB1dA,dB1dQ,dB2dA,
     + dB2dQ,dE1dA,dE1dQ,dE2dA,dE2dQ,WK1,WK2,WK3,WK4)
C
       IMPLICIT REAL *8(A-H,O-Z)
       DIMENSION CJA(8,8),W(3),FI(2),DFIDS(2)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHC,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C         WRITE(6,1)IELNO
1     FORMAT(2X,'I AM IN MCJBIANS ',2X,I2,4(2X,F10.6))
C
      t1= -DFIDS(1)*dF1dA*FI(1)
      t2= FI(1)*dG1dA*FI(1)*DXM(IELNO)/2.
      t3=(DFIDS(1)*WK1*dB1dA + DFIDS(1)*WK2*dB2dA)*DFIDS(1)*2./
     +   DXM(IELNO)
      t4=(DFIDS(1)*WK1*dE1dA + DFIDS(1)*WK2*dE2dA)*FI(1)
      t=t1 + t2 + t3 + t4
       CJA(1,1)= CJA(1,1) + W(K)*t
C
```

```
      t1= -DFIDS(1)*dF1dQ*FI(1)
      t2= FI(1)*dG1dQ*FI(1)*DXM(IELNO)/2.
      t3=(DFIDS(1)*WK1*dB1dQ + DFIDS(1)*WK2*dB2dQ)*DFIDS(1)*2./
     +   DXM(IELNO)
      t4=(DFIDS(1)*WK1*dE1dQ + DFIDS(1)*WK2*dE2dQ)*FI(1)
      t= t1 + t2 + t3 + t4
       CJA(1,2)= CJA(1,2) + W(K)*t
C
      t1= -DFIDS(1)*dF1dA*FI(2)
      t2= FI(1)*dG1dA*FI(2)*DXM(IELNO)/2.
      t3=(DFIDS(1)*WK1*dB1dA + DFIDS(1)*WK2*dB2dA)*DFIDS(2)*2./
     +   DXM(IELNO)
      t4=(DFIDS(1)*WK1*dE1dA + DFIDS(1)*WK2*dE2dA)*FI(2)
      t= t1 + t2 + t3 + t4
       CJA(1,3)= CJA(1,3) + W(K)*t
C
      t1= -DFIDS(1)*dF1dQ*FI(2)
      t2= FI(1)*dG1dQ*FI(2)*DXM(IELNO)/2.
      t3=(DFIDS(1)*WK1*dB1dQ + DFIDS(1)*WK2*dB2dQ)*DFIDS(2)*2./
     +   DXM(IELNO)
      t4=(DFIDS(1)*WK1*dE1dQ + DFIDS(1)*WK2*dE2dQ)*FI(2)
      t= t1 + t2 + t3 + t4
       CJA(1,4)= CJA(1,4) + W(K)*t
C
      t1= -DFIDS(1)*dF2dA*FI(1)
      t2= FI(1)*dG2dA*FI(1)*DXM(IELNO)/2.
      t3=(DFIDS(1)*WK3*dB1dA + DFIDS(1)*WK4*dB2dA)*DFIDS(1)*2./
     +   DXM(IELNO)
      t4=(DFIDS(1)*WK3*dE1dA + DFIDS(1)*WK4*dE2dA)*FI(1)
      t= t1 + t2 + t3 + t4
       CJA(2,1)= CJA(2,1) + W(K)*t
C
      t1= -DFIDS(1)*dF2dQ*FI(1)
      t2= FI(1)*dG2dQ*FI(1)*DXM(IELNO)/2.
      t3=(DFIDS(1)*WK3*dB1dQ + DFIDS(1)*WK4*dB2dQ)*DFIDS(1)*2./
     +   DXM(IELNO)
      t4=(DFIDS(1)*WK3*dE1dQ + DFIDS(1)*WK4*dE2dQ)*FI(1)
      t= t1 + t2 + t3 + t4
       CJA(2,2)= CJA(2,2) + W(K)*t
C
      t1= -DFIDS(1)*dF2dA*FI(2)
      t2= FI(1)*dG2dA*FI(2)*DXM(IELNO)/2.
      t3=(DFIDS(1)*WK3*dB1dA + DFIDS(1)*WK4*dB2dA)*DFIDS(2)*2./
     +   DXM(IELNO)
      t4=(DFIDS(1)*WK3*dE1dA + DFIDS(1)*WK4*dE2dA)*FI(2)
      t= t1 + t2 + t3 + t4
       CJA(2,3)= CJA(2,3) + W(K)*t
C
      t1= -DFIDS(1)*dF2dQ*FI(2)
      t2= FI(1)*dG2dQ*FI(2)*DXM(IELNO)/2.
      t3=(DFIDS(1)*WK3*dB1dQ + DFIDS(1)*WK4*dB2dQ)*DFIDS(2)*2./
     +   DXM(IELNO)
      t4=(DFIDS(1)*WK3*dE1dQ + DFIDS(1)*WK4*dE2dQ)*FI(2)
      t= t1 + t2 + t3 + t4
       CJA(2,4)= CJA(2,4) + W(K)*t
C
      t1= -DFIDS(2)*dF1dA*FI(1)
      t2= FI(2)*dG1dA*FI(1)*DXM(IELNO)/2.
```

```
      t3=(DFIDS(2)*WK1*dB1dA + DFIDS(2)*WK2*dB2dA)*DFIDS(1)*2./
     +    DXM(IELNO)
      t4=(DFIDS(2)*WK1*dE1dA + DFIDS(2)*WK2*dE2dA)*FI(1)
      t= t1 + t2 + t3 + t4
       CJA(3,1)= CJA(3,1) + W(K)*t
C
      t1= -DFIDS(2)*dF1dQ*FI(1)
      t2= FI(2)*dG1dQ*FI(1)*DXM(IELNO)/2.
      t3=(DFIDS(2)*WK1*dB1dQ + DFIDS(2)*WK2*dB2dQ)*DFIDS(1)*2./
     +    DXM(IELNO)
      t4=(DFIDS(2)*WK1*dE1dQ + DFIDS(2)*WK2*dE2dQ)*FI(1)
      t= t1 + t2 + t3 + t4
       CJA(3,2)= CJA(3,2) + W(K)*t
C
      t1= -DFIDS(2)*dF1dA*FI(2)
      t2= FI(2)*dG1dA*FI(2)*DXM(IELNO)/2.
      t3=(DFIDS(2)*WK1*dB1dA + DFIDS(2)*WK2*dB2dA)*DFIDS(2)*2./
     +    DXM(IELNO)
      t4=(DFIDS(2)*WK1*dE1dA + DFIDS(2)*WK2*dE2dA)*FI(2)
      t= t1 + t2 + t3 + t4
       CJA(3,3)= CJA(3,3) + W(K)*t
C
      t1= -DFIDS(2)*dF1dQ*FI(2)
      t2= FI(2)*dG1dQ*FI(2)*DXM(IELNO)/2.
      t3=(DFIDS(2)*WK1*dB1dQ + DFIDS(2)*WK2*dB2dQ)*DFIDS(2)*2./
     +    DXM(IELNO)
      t4=(DFIDS(2)*WK1*dE1dQ + DFIDS(2)*WK2*dE2dQ)*FI(2)
      t= t1 + [illegible] t3 + t4
       CJA(3 [illegible] (3,4) + W(K)*t
C
      t1= -DF[illegible]*dF2dA*FI(1)
      t2= FI(2)*dG2dA*FI(1)*DXM(IELNO)/2.
      t3=(DFIDS(2)*WK3*dB1dA + DFIDS(2)*WK4*dB2dA)*DFIDS(1)*2./
     +    DXM(IELNO)
      t4=(DFIDS(2)*WK3*dE1dA + DFIDS(2)*WK4*dE2dA)*FI(1)
      t= t1 + t2 + t3 + t4
       CJA(4,1)= CJA(4,1) + W(K)*t
C
      t1= -DFIDS(2)*dF2dQ*FI(1)
      t2= FI(2)*dG2dQ*FI(1)*DXM(IELNO)/2.
      t3=(DFIDS(2)*WK3*dB1dQ + DFIDS(2)*WK4*dB2dQ)*DFIDS(1)*2./
     +    DXM(IELNO)
      t4=(DFIDS(2)*WK3*dE1dQ + DFIDS(2)*WK4*dE2dQ)*FI(1)
      t= t1 + t2 + t3 + t4
       CJA(4,2)= CJA(4,2) + W(K)*t
C
      t1= -DFIDS(2)*dF2dA*FI(2)
      t2= FI(2)*dG2dA*FI(2)*DXM(IELNO)/2.
      t3=(DFIDS(2)*WK3*dB1dA + DFIDS(2)*WK4*dB2dA)*DFIDS(2)*2./
     +    DXM(IELNO)
      t4=(DFIDS(2)*WK3*dE1dA + DFIDS(2)*WK4*dE2dA)*FI(2)
      t= t1 + t2 + t3 + t4
       CJA(4,3)= CJA(4,3) + W(K)*t
C
      t1= -DFIDS(2)*dF2dQ*FI(2)
      t2= FI(2)*dG2dQ*FI(2)*DXM(IELNO)/2.
      t3=(DFIDS(2)*WK3*dB1dQ + DFIDS(2)*WK4*dB2dQ)*DFIDS(2)*2./
     +    DXM(IELNO)
```

```fortran
      t4=(DFIDS(2)*WK3*dE1dQ + DFIDS(2)*WK4*dE2dQ)*FI(2)
      t= t1 + t2 + t3 + t4
       CJA(4,4)= CJA(4,4) + W(K)*t
C
C         IF(K.EQ.NGP)THEN
C         DO 30 I=1,4
C         DO 30 J=1,4
C         WRITE(6,50)I,J,CJA(I,J)
C 30      CONTINUE
C         ENDIF
 50     FORMAT(2X,2I3,2X,F12.6)
         RETURN
          END
C
C ***** CALCULATION OF PARTIAL DERIVATIVES FOR MAIN CHANNEL ********
C
       SUBROUTINE DERIVB(IELNO,So,dBdX,QQ,AA,U,H,B,KLP,KFL,dF1dA,
     + dF1dQ,dF2dA,dF2dQ,dG1dA,dG1dQ,dG2dA,dG2dQ)
C
       IMPLICIT REAL *8(A-H,O-Z)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C        WRITE(6,1)
1       FORMAT(2X,'I AM IN SUBROUTINE DERIVB')
C
      IF(KFL.EQ.1)THEN
      F=1.0D+00
      ELSE
      F=0.0D+00
      ENDIF
C
       IF(KLP.EQ.1)THEN
       A1=2.*AA+B**2
      A2=GRAV*QQ*DABS(QQ)*CN1(IELNO)**2*A1**1.333/(AA**3.333*B**1.333)
      A3=GRAV*QQ*DABS(QQ)*CN1(IELNO)**2*A1**0.333/(AA**2.333*B**1.333)
        ENDIF
C
       T1= ELVLP(NODNUM(IELNO,1))-ELVMc(NODNUM(IELNO,1))
       IF(T1.GT.H)THEN
        T1=H
        ENDIF
C
       A4=1.0D+00 + Z1**2
       A5=DSQRT(A4)
       B4=1.0D+00 + Z2**2
       B5=DSQRT(B4)
```

```
C
       P= B + T1*(A5+B5)
       R=AA/(2.0*H + B)
C
      A4=R/CN1(IELNO)
      CS= 5.75*DLOG10(A4) + 6.2D+00
      dF1dA=0.0D+00
      dF1dQ=1.0D+00
C
      dF2dA= -(QQ**2/AA**2) + GRAV*H
      dF2dQ= 2.0*U
      dG1dA=0.0D+00
      dG1dQ=0.0D+00
C
C        A3=GRAV*QQ*DABS(QQ)*CN1(IELNO)**2/(AA**2*R**1.333)
C
            IF(KLP.EQ.1)THEN
       dG2dA= -GRAV*AA*dBdX/B**2 - GRAV*So -(7./3.)*A2*F +(8./3.)*A3*F
C        dG2dA= -GRAV*AA*dBdX/B**2 - GRAV*So - A3*F
C
       dG2dQ= 2.*F*GRAV*QQ*CN1(IELNO)**2*A1**1.333/(AA**2.333*B**1.333)
C        dG2dQ=2.*F*GRAV*QQ*CN1(IELNO)**2/(AA*R**1.333)
              ELSE
      dG2dA= -GRAV*AA*dBdX/B**2 - GRAV*So -2.*QQ*DABS(QQ)*F*B/
     + (AA**3*CS**2)- 5.0*QQ*DABS(QQ)*F/(AA**2*CS**3*R)
C
       dG2dQ= 2.*F*QQ*B/(AA**2*CS**2)
            ENDIF
C       WRITE(6,20)dF1dA,dF1dQ,dF2dA,dF2dQ
C       WRITE(6,20)dG1dA,dG1dQ,dG2dA,dG2dQ
20     FORMAT(2X,6(2X,F12.6))
      RETURN
      END
C
C  *** DERIVATIVES IS FROM THE NON-CONSERVATIVE PART EQ.(MAIN CHANEL
ONLY )***
C
       SUBROUTINE  DERIVP(IELNO,So,dBdX,QQ,AA,U,H,B,KLP,KFL,dB1dA,
     + dB1dQ,dB2dA,dB2dQ,dE1dA,dE1dQ,dE2dA,dE2dQ)
C
       IMPLICIT REAL *8(A-H,O-Z)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C        WRITE(6,1)
1       FORMAT(2X,'I AM IN SUBROUTINE DERIVP')
C
```

```
      IF(KFL.EQ.1)THEN
      F=1.0D+00
      ELSE
      F=0.0D+00
      ENDIF
C
       IF(KLP.EQ.1)THEN
       A1=2.*AA + B**2
       A2=GRAV*QQ*DABS(QQ)*CN1(IELNO)**2*
     +  A1**1.333/(AA**3.333*B**1.333)
C
       A3=GRAV*QQ*DABS(QQ)*CN1(IELNO)**2*A1**0.333/
     + (AA**2.333*B**1.333)
      ENDIF
C
       T1= ELVLP(NODNUM(IELNO,1))-ELVMc(NODNUM(IELNO,1))
       IF(T1.GT.H)THEN
        T1=H
        ENDIF
C
C
       A4=1.0D+00 + Z1**2
       A5=DSQRT(A4)
       B4=1.0D+00 + Z2**2
       B5=DSQRT(B4)
C
       P= B + T1*(A5+B5)
       R=AA/(2.0*H + B)
C
      A4=R/CN1(IELNO)
      CS= 5.75*DLOG10(A4) + 6.2D+00
      dB1dA=0.0D+00
      dB1dQ=1.0D+00
      dB2dA = 2.0*GRAV*H -U**2
      dB2dQ = 2.0*U
      dE1dA =0.0D+00
      dE1dQ =0.0D+00
C
C         A3=GRAV*QQ*DABS(QQ)*CN1(IELNO)**2/(AA**2*R**1.333)
C
            IF(KLP.EQ.1)THEN
       dE2dA= -GRAV*AA*dBdX/B**2 - GRAV*So -(7./3.)*A2*F +(8./3.)*A3*F
C         dE2dA= -2.0*GRAV*AA*dBdX/B**2 - GRAV*So - A3*F
C
       dE2dQ= 2.*F*GRAV*QQ*CN1(IELNO)**2*A1**1.333/(AA**2.333*B**1.333)
C         dE2dQ = 2.*F*GRAV*QQ*CN1(IELNO)**2/(AA*R**1.333)
              ELSE
      dE2dA = -2.0*GRAV*AA*dBdX/B**2 - GRAV*So -2.*QQ*
     + DABS(QQ)*F*B/(AA**3*CS**2) -5.0*QQ*DABS(QQ)*F/
     +(AA**2*CS**3*R)
C
       dE2dQ = 2.*F*QQ*B/(AA**2*CS**2)
           ENDIF
C
C      WRITE(6,20) dB1dA,dB1dQ,dB2dA,dB2dQ
C      WRITE(6,20) dE1dA,dE1dQ,dE2dA,dE2dQ
20     FORMAT(2X,5(2X,F12.6))
C
```

```
      RETURN
      END
C
C   ******** CALCULATION OF THE JACOBIAN PART 1 *******
C
      SUBROUTINE PART1(IELNO,K,W,FI,DFIDS,tAL1,tAL2,dF1dA,dF1dQ,
     + dF1dAR,dF2dAL,dF2dA,dF2dQ,dF2dAR,dF3dAL,dF3dA,dF3dQ,
     + dF3dAR,dF4dAL,dF4dA,dF4dQ,tAR1,tAR2,CJA)
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION CJA(8,8),DFIDS(2),FI(2),W(3)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C       WRITE(6,1)
1      FORMAT(2X,'I AM IN SUBROUTINE PART1')
C
       CJA(1,1)= CJA(1,1)  + W(K)*(-DFIDS(1)*(tAL1 +
     + tAL2*DFIDS(1)*2.0/DXL(IELNO))*FI(1))
       CJA(1,2)= CJA(1,2)  + W(K)*(-DFIDS(1)*dF1dA*FI(1))
       CJA(1,3)= CJA(1,3)  + W(K)*(-DFIDS(1)*dF1dQ*FI(1))
       CJA(1,4)= CJA(1,4)  + W(K)*(-DFIDS(1)*dF1dAR*FI(1))
C
       CJA(1,5)= CJA(1,5)  + W(K)*(-DFIDS(1)*(tAL1 +
     + tAL2*DFIDS(2)*2.0/DXL(IELNO))*FI(2))
       CJA(1,6)= CJA(1,6)  + W(K)*(-DFIDS(1)*dF1dA*FI(2))
       CJA(1,7)= CJA(1,7)  + W(K)*(-DFIDS(1)*dF1dQ*FI(2))
       CJA(1,8)= CJA(1,8)  + W(K)*(-DFIDS(1)*dF1dAR*FI(2))
C
       CJA(2,1)= CJA(2,1)  + W(K)*(-DFIDS(1)*dF2dAL*FI(1))
       CJA(2,2)= CJA(2,2)  + W(K)*(-DFIDS(1)*dF2dA*FI(1))
       CJA(2,3)= CJA(2,3)  + W(K)*(-DFIDS(1)*dF2dQ*FI(1))
       CJA(2,4)= CJA(2,4)  + W(K)*(-DFIDS(1)*dF2dAR*FI(1))
C
       CJA(2,5)= CJA(2,5)  + W(K)*(-DFIDS(1)*dF2dAL*FI(2))
       CJA(2,6)= CJA(2,6)  + W(K)*(-DFIDS(1)*dF2dA*FI(2))
       CJA(2,7)= CJA(2,7)  + W(K)*(-DFIDS(1)*dF2dQ*FI(2))
       CJA(2,8)= CJA(2,8)  + W(K)*(-DFIDS(1)*dF2dAR*FI(2))
C
       CJA(3,1)= CJA(3,1)  + W(K)*(-DFIDS(1)*dF3dAL*FI(1))
       CJA(3,2)= CJA(3,2)  + W(K)*(-DFIDS(1)*dF3dA*FI(1))
       CJA(3,3)= CJA(3,3)  + W(K)*(-DFIDS(1)*dF3dQ*FI(1))
       CJA(3,4)= CJA(3,4)  + W(K)*(-DFIDS(1)*dF3dAR*FI(1))
C
       CJA(3,5)= CJA(3,5)  + W(K)*(-DFIDS(1)*dF3dAL*FI(2))
       CJA(3,6)= CJA(3,6)  + W(K)*(-DFIDS(1)*dF3dA*FI(2))
       CJA(3,7)= CJA(3,7)  + W(K)*(-DFIDS(1)*dF3dQ*FI(2))
```

```
      CJA(3,8)= CJA(3,8)  + W(K)*(-DFIDS(1)*dF3dAR*FI(2))
C
      CJA(4,1)= CJA(4,1)  + W(K)*(-DFIDS(1)*dF4dAL*FI(1))
      CJA(4,2)= CJA(4,2)  + W(K)*(-DFIDS(1)*dF4dA*FI(1))
      CJA(4,3)= CJA(4,3)  + W(K)*(-DFIDS(1)*dF4dQ*FI(1))
      CJA(4,4)= CJA(4,4)  + W(K)*(-DFIDS(1)*(tAR1 +
     + tAR2*DFIDS(1)*2.0/DXR(IELNO))*FI(1))
C
      CJA(4,5)= CJA(4,5)  + W(K)*(-DFIDS(1)*dF4dAL*FI(2))
      CJA(4,6)= CJA(4,6)  + W(K)*(-DFIDS(1)*dF4dA*FI(2))
      CJA(4,7)= CJA(4,7)  + W(K)*(-DFIDS(1)*dF4dQ*FI(2))
      CJA(4,8)= CJA(4,8)  + W(K)*(-DFIDS(1)*(tAR1 +
     + tAR2*DFIDS(2)*2.0/DXR(IELNO))[illegible]I(2))
C
      CJA(5,1)= CJA(5,1)  + W(K)[illegible](2)*(tAL1 +
     + tAL2*DFIDS(1)*2.0/DXL(IELN[illegible]1))
      CJA(5,2)= CJA(5,2)  + W(K)*([illegible]S(2)*dF1dA*FI(1))
      CJA(5,3)= CJA(5,3)  + W(K)*(-DFIDS(2)*dF1dQ*FI(1))
      CJA(5,4)= CJA(5,4)  + W(K)*(-DFIDS(2)*dF1dAR*FI(1))
C
      CJA(5,5)= CJA(5,5)  + W(K)*(-DFIDS(2)*(tAL1 +
     + tAL2*DFIDS(2)*2.0/DXL(IELNO))*FI(2))
      CJA(5,6)= CJA(5,6)  + W(K)*(-DFIDS(2)*dF1dA*FI(2))
      CJA(5,7)= CJA(5,7)  + W(K)*(-DFIDS(2)*dF1dQ*FI(2))
      CJA(5,8)= CJA(5,8)  + W(K)*(-DFIDS(2)*dF1dAR*FI(2))
C
      CJA(6,1)= CJA(6,1)  + W(K)*(-DFIDS(2)*dF2dAL*FI(1))
      CJA(6,2)= CJA(6,2)  + W(K)*(-DFIDS(2)*dF2dA*FI(1))
      CJA(6,3)= CJA(6,3)  + W(K)*(-DFIDS(2)*dF2dQ*FI(1))
      CJA(6,4)= CJA(6,4)  + W(K)*(-DFIDS(2)*dF2dAR*FI(1))
C
      CJA(6,5)= CJA(6,5)  + W(K)*(-DFIDS(2)*dF2dAL*FI(2))
      CJA(6,6)= CJA(6,6)  + W(K)*(-DFIDS(2)*dF2dA*FI(2))
      CJA(6,7)= CJA(6,7)  + W(K)*(-DFIDS(2)*dF2dQ*FI(2))
      CJA(6,8)= CJA(6,8)  + W(K)*(-DFIDS(2)*dF2dAR*FI(2))
C
      CJA(7,1)= CJA(7,1)  + W(K)*(-DFIDS(2)*dF3dAL*FI(1))
      CJA(7,2)= CJA(7,2)  + W(K)*(-DFIDS(2)*dF3dA*FI(1))
      CJA(7,3)= CJA(7,3)  + W(K)*(-DFIDS(2)*dF3dQ*FI(1))
      CJA(7,4)= CJA(7,4)  + W(K)*(-DFIDS(2)*dF3dAR*FI(1))
C
      CJA(7,5)= CJA(7,5)  + W(K)*(-DFIDS(2)*dF3dAL*FI(2))
      CJA(7,6)= CJA(7,6)  + W(K)*(-DFIDS(2)*dF3dA*FI(2))
      CJA(7,7)= CJA(7,7)  + W(K)*(-DFIDS(2)*dF3dQ*FI(2))
      CJA(7,8)= CJA(7,8)  + W(K)*(-DFIDS(2)*dF3dAR*FI(2))
C
      CJA(8,1)= CJA(8,1)  + W(K)*(-DFIDS(2)*dF4dAL*FI(1))
      CJA(8,2)= CJA(8,2)  + W(K)*(-DFIDS(2)*dF4dA*FI(1))
      CJA(8,3)= CJA(8,3)  + W(K)*(-DFIDS(2)*dF4dQ*FI(1))
      CJA(8,4)= CJA(8,4)  + W(K)*(-DFIDS(2)*(tAR1 +
     + tAR2*DFIDS(1)*2.0/DXR(IELNO))*FI(1))
C
      CJA(8,5)= CJA(8,5)  + W(K)*(-DFIDS(2)*dF4dAL*FI(2))
      CJA(8,6)= CJA(8,6)  + W(K)*(-DFIDS(2)*dF4dA*FI(2))
      CJA(8,7)= CJA(8,7)  + W(K)*(-DFIDS(2)*dF4dQ*FI(2))
      CJA(8,8)= CJA(8,8)  + W(K)*(-DFIDS(2)*(tAR1 +
     + tAR2*DFIDS(2)*2.0/DXR(IELNO))*FI(2))
C
```

```
      RETURN
       END
C
C   ******** CALCULATION OF THE JACOBIAN PART 2 *******
C
      SUBROUTINE PART2(IELNO,K,W,WK1,WK2,WK3,WK4,DFIDS,
     + dD2dAL,dD2dA,dD2dQ,dD2dAR,dD3dAL,dD3dA,
     + dD3dQ,dD3dAR,CJB)
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION CJB(8,8),DFIDS(2),W(3)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C       WRITE(6,1)
1       FORMAT(2X,'I AM IN SUBROUTINE PART2',4(2X,F10.6))
C
      t=(WK1*DFIDS(1)*dD2dAL + WK2*DFIDS(1)*dD3dAL)*DFIDS(1)*2./
     +  DXM(IELNO)
      CJB(2,1)= CJB(2,1)  + W(K)*t
C
      t=(WK1*DFIDS(1)*dD2dA + WK2*DFIDS(1)*dD3dA)*DFIDS(1)*2./
     +   DXM(IELNO)
      CJB(2,2)= CJB(2,2)  + W(K)*t
C
      t=(WK1*DFIDS(1)*dD2dQ + WK2*DFIDS(1)*dD3dQ)*DFIDS(1)*2./
     +   DXM(IELNO)
      CJB(2,3)= CJB(2,3)  + W(K)*t
C
      t=(WK1*DFIDS(1)*dD2dAR + WK2*DFIDS(1)*dD3dAR)*DFIDS(1)*2./
     +   DXM(IELNO)
      CJB(2,4)= CJB(2,4)  + W(K)*t
C
      t=(WK1*DFIDS(1)*dD2dAL + WK2*DFIDS(1)*dD3dAL)*DFIDS(2)*2./
     +   DXM(IELNO)
      CJB(2,5)= CJB(2,5)  + W(K)*t
C
      t=(WK1*DFIDS(1)*dD2dA + WK2*DFIDS(1)*dD3dA)*DFIDS(2)*2./
     +   DXM(IELNO)
      CJB(2,6)= CJB(2,6)  + W(K)*t
C
      t=(WK1*DFIDS(1)*dD2dQ + WK2*DFIDS(1)*dD3dQ)*DFIDS(2)*2./
     +   DXM(IELNO)
      CJB(2,7)= CJB(2,7)  + W(K)*t
C
      t=(WK1*DFIDS(1)*dD2dAR + WK2*DFIDS(1)*dD3dAR)*DFIDS(2)*2./
     +   DXM(IELNO)
```

```
      CJB(2,8)= CJB(2,8) + W(K)*t
C
      t=(WK3*DFIDS(1)*dD2dAL + WK4*DFIDS(1)*dD3dAL)*DFIDS(1)*2./
     +   DXM(IELNO)
      CJB(3,1)= CJB(3,1) + W(K)*t
C
      t=(WK3*DFIDS(1)*dD2dA + WK4*DFIDS(1)*dD3dA)*DFIDS(1)*2./
     +   DXM(IELNO)
      CJB(3,2)= CJB(3,2) + W(K)*t
C
      t=(WK3*DFIDS(1)*dD2dQ + WK4*DFIDS(1)*dD3dQ)*DFIDS(1)*2./
     +   DXM(IELNO)
      CJB(3,3)= CJB(3,3) + W(K)*t
C
      t=(WK3*DFIDS(1)*dD2dAR + WK4*DFIDS(1)*dD3dAR)*DFIDS(1)*2.
     +   DXM(IELNO)
      CJB(3,4)= CJB(3,4) + W(K)*t
C
      t=(WK3*DFIDS(1) dD2dAL + WK4*DFIDS(1)*dD3dAL)*DFIDS(2)*2./
     +   DXM(IELNO)
      CJB(3,5)= CJB(3,5) + W(K)*t
C
      t=(WK3*DFIDS(1)*dD2dA + WK4*DFIDS(1)*dD3dA)*DFIDS(2)*2./
     +   DXM(IELNO)
      CJB(3,6)= CJB(3,6) + W(K)*t
C
      t=(WK3*DFIDS(1)*dD2dQ + WK4*DFIDS(1)*dD3dQ)*DFIDS(2)*2./
     +   DXM(IELNO)
      CJB(3,7)= CJB(3,7) + W(K)*t
C
      t=(WK3*DFIDS(1)*dD2dAR + WK4*DFIDS(1)*dD3dAR)*DFIDS(2)*2./
     +   DXM(IELNO)
      CJB(3,8)= CJB(3,8) + W(K)*t
C
      t=(WK1*DFIDS(2)*dD2dAL + WK2*DFIDS(2)*dD3dAL)*DFIDS(1)*2./
     +   DXM(IELNO)
      CJB(6,1)= CJB(6,1) + W(K)*t
C
      t=(WK1*DFIDS(2)*dD2dA + WK2*DFIDS(2)*dD3dA)*DFIDS(1)*2./
     +   DXM(IELNO)
      CJB(6,2)= CJB(6,2) + W(K *t
C
      t=(WK1*DFIDS(2)*dD2dQ + WK2*DFIDS(2)*dD3dQ)*DFIDS(1)*2./
     +   DXM(IELNO)
      CJB(6,3)= CJB(6,3) + W(K)*t
C
      t=(WK1*DFIDS(2)*dD2dAR + WK2*DFIDS(2)*dD3dAR)*DFIDS(1)*2./
     +   DXM(IELNO)
      CJB(6,4)= CJB(6,4) + W(K)*t
C
      t=(WK1*DFIDS(2)*dD2dAL + WK2*DFIDS(2)*dD3dAL)*DFIDS(2)*2./
     +   DXM(IELNO)
      CJB(6,5)= CJB(6,5) + W(K)*t
C
      t=(WK1*DFIDS(2)*dD2dA + WK2*DFIDS(2)*dD3dA)*DFIDS(2)*2./
     +   DXM(IELNO)
      CJB(6,6)= CJB(6,6) + W(K)*t
C
```

```
      t=(WK1*DFIDS(2)*dD2dQ + WK2*DFIDS(2)*dD3dQ)*DFIDS(2)*2./
     +    DXM(IELNO)
      CJB(6,7)= CJB(6,7) + W(K)*t
C
      t=(WK1*DFIDS(2)*dD2dAR + WK2*DFIDS(2)*dD3dAR)*DFIDS(2)*2./
     +    DXM(IELNO)
      CJB(6,8)= CJB(6,8) + W(K)*t
C
      t=(WK3*DFIDS(2)*dD2dAL + WK4*DFIDS(2)*dD3dAL)*DFIDS(1)*2./
     +    DXM(IELNO)
      CJB(7,1)= CJB(7,1) + W(K)*t
C
      t=(WK3*DFIDS(2)*dD2dA + WK4*DFIDS(2)*dD3dA)*DFIDS(1)*2./
     +    DXM(IELNO)
      CJB(7,2)= CJB(7,2) + W(K)*t
C
      t=(WK3*DFIDS(2)*dD2dQ + WK4*DFIDS(2)*dD3dQ)*DFIDS(1)*2./
     +    DXM(IELNO)
      CJB(7,3)= CJB(7,3) + W(K)*t
C
      t=(WK3*DFIDS(2)*dD2dAR + WK4*DFIDS(2)*dD3dAR)*DFIDS(1)*2./
     +    DXR(IELNO)
      CJB(7,4)= CJB(7,4) + W(K)*t
C
      t=(WK3*DFIDS(2)*dD2dAL + WK4*DFIDS(2)*dD3dAL)*DFIDS(2)*2./
     +    DXM(IELNO)
      CJB(7,5)= CJB(7,5) + W(K)*t
C
      t=(WK3*DFIDS(2)*dD2dA + WK4*DFIDS(2)*dD3dA)*DFIDS(2)*2./
     +    DXM(IELNO)
      CJB(7,6)= CJB(7,6) + W(K)*t
C
      t=(WK3*DFIDS(2)*dD2dQ + WK4*DFIDS(2)*dD3dQ)*DFIDS(2)*2./
     +    DXM(IELNO)
      CJB(7,7)= CJB(7,7) + W(K)*t
C
      t=(WK3*DFIDS(2)*dD2dAR + WK4*DFIDS(2)*dD3dAR)*DFIDS(2)*2./
     +    DXM(IELNO)
      CJB(7,8)= CJB(7,8) + W(K)*t
C
         RETURN
          END
C
C   ******** CALCULATION OF THE JACOBIAN PART 3 *******
C
      SUBROUTINE PART3(IELNO,K,W,FI,dG1dAL,dG1dA,dG1dQ,
     + dG1dAR,dG2dAL,dG2dA,dG2dQ,dG2dAR,dG3dAL,dG3dA,dG3dQ,
     + dG3dAR,dG4dAL,dG4dA,dG4dQ,dG4dAR,CJC)
C
       IMPLICIT REAL *8(A-H,O-Z)
       DIMENSION CJC(8,8),FI(2),W(3)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
```

```
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C         WRITE(6,1)
1        FORMAT(2X,'I AM IN SUBROUTINE PART3')
C
        CJC(1,1)= CJC(1,1)  + W(K)*FI(1)*dG1dAL*FI(1)*DXL(IELNO)/2.0
        CJC(1,2)= CJC(1,2)  + W(K)*FI(1)*dG1dA*FI(1)*DXL(IELNO)/2.0
        CJC(1,3)= CJC(1,3)  + W(K)*FI(1)*dG1dQ*FI(1)*DXL(IELNO)/2.0
        CJC(1,4)= CJC(1,4)  + W(K)*FI(1)*dG1dAR*FI(1)*DXL(IELNO)/2.0
C
        CJC(1,5)= CJC(1,5)  + W(K)*FI(1)*dG1dAL*FI(2)*DXL(IELNO)/2.0
        CJC(1,6)= CJC(1,6)  + W(K)*FI(1)*dG1dA*FI(2)*DXL(IELNO)/2.0
        CJC(1,7)= CJC(1,7)  + W(K)*FI(1)*dG1dQ*FI(2)*DXL(IELNO)/2.0
        CJC(1,8)= CJC(1,8)  + W(K)*FI(1)*dG1dAR*FI(2)*DXL(IELNO)/2.0
C
        CJC(2,1)= CJC(2,1)  + W(K)*FI(1)*dG2dAL*FI(1)*DXM(IELNO)/2.0
        CJC(2,2)= CJC(2,2)  + W(K)*FI(1)*dG2dA*FI(1)*DXM(IELNO)/2.0
        CJC(2,3)= CJC(2,3)  + W(K)*FI(1)*dG2dQ*FI(1)*DXM(IELNO)/2.0
        CJC(2,4)= CJC(2,4)  + W(K)*FI(1)*dG2dAR*FI(1)*DXM(IELNO)/2.0
C
        CJC(2,5)= CJC(2,5)  + W(K)*FI(1)*dG2dAL*FI(2)*DXM(IELNO)/2.0
        CJC(2,6)= CJC(2,6)  + W(K)*FI(1)*dG2dA*FI(2)*DXM(IELNO)/2.0
        CJC(2,7)= CJC(2,7)  + W(K)*FI(1)*dG2dQ*FI(2)*DXM(IELNO)/2.0
        CJC(2,8)= CJC(2,8)  + W(K)*FI(1)*dG2dAR*FI(2)*DXM(IELNO)/2.0
C
        CJC(3,1)= CJC(3,1)  + W(K)*FI(1)*dG3dAL*FI(1)*DXM(IELNO)/2.0
        CJC(3,2)= CJC(3,2)  + W(K)*FI(1)*dG3dA*FI(1)*DXM(IELNO)/2.0
        CJC(3,3)= CJC(3,3)  + W(K)*FI(1)*dG3dQ*FI(1)*DXM(IELNO)/2.0
        CJC(3,4)= CJC(3,4)  + W(K)*FI(1)*dG3dAR*FI(1)*DXM(IELNO)/2.0
C
        CJC(3,5)= CJC(3,5)  + W(K)*FI(1)*dG3dAL*FI(2)*DXM(IELNO)/2.0
        CJC(3,6)= CJC(3,6)  + W(K)*FI(1)*dG3dA*FI(2)*DXM(IELNO)/2.0
        CJC(3,7)= CJC(3,7)  + W(K)*FI(1)*dG3dQ*FI(2)*DXM(IELNO)/2.0
        CJC(3,8)= CJC(3,8)  + W(K)*FI(1)*dG3dAR*FI(2)*DXM(IELNO)/2.0
C
        CJC(4,1)= CJC(4,1)  + W(K)*FI(1)*dG4dAL*FI(1)*DXR(IELNO)/2.0
        CJC(4,2)= CJC(4,2)  + W(K)*FI(1)*dG4dA*FI(1)*DXR(IELNO)/2.0
        CJC(4,3)= CJC(4,3)  + W(K)*FI(1)*dG4dQ*FI(1)*DXR(IELNO)/2.0
        CJC(4,4)= CJC(4,4)  + W(K)*FI(1)*dG4dAR*FI(1)*DXR(IELNO)/2.0
C
        CJC(4,5)= CJC(4,5)  + W(K)*FI(1)*dG4dAL*FI(2)*DXR(IELNO)/2.0
        CJC(4,6)= CJC(4,6)  + W(K)*FI(1)*dG4dA*FI(2)*DXR(IELNO)/2.0
        CJC(4,7)= CJC(4,7)  + W(K)*FI(1)*dG4dQ*FI(2)*DXR(IELNO)/2.0
        CJC(4,8)= CJC(4,8)  + W(K)*FI(1)*dG4dAR*FI(2)*DXR(IELNO)/2.0
C
        CJC(5,1)= CJC(5,1)  + W(K)*FI(2)*dG1dAL*FI(1)*DXL(IELNO)/2.0
        CJC(5,2)= CJC(5,2)  + W(K)*FI(2)*dG1dA*FI(1)*DXL(IELNO)/2.0
        CJC(5,3)= CJC(5,3)  + W(K)*FI(2)*dG1dQ*FI(1)*DXL(IELNO)/2.0
        CJC(5,4)= CJC(5,4)  + W(K)*FI(2)*dG1dAR*FI(1)*DXL(IELNO)/2.0
C
        CJC(5,5)= CJC(5,5)  + W(K)*FI(2)*dG1dAL*FI(2)*DXL(IELNO)/2.0
        CJC(5,6)= CJC(5,6)  + W(K)*FI(2)*dG1dA*FI(2)*DXL(IELNO)/2.0
        CJC(5,7)= CJC(5,7)  + W(K)*FI(2)*dG1dQ*FI(2)*DXL(IELNO)/2.0
```

```
      CJC(5,8)= CJC(5,8)  + W(K)*FI(2)*dG1dAR*FI(2)*DXL(IELNO)/2.0
C
      CJC(6,1)= CJC(6,1)  + W(K)*FI(2)*dG2dAL*FI(1)*DXM(IELNO)/2.0
      CJC(6,2)= CJC(6,2)  + W(K)*FI(2)*dG2dA*FI(1)*DXM(IELNO)/2.0
      CJC(6,3)= CJC(6,3)  + W(K)*FI(2)*dG2dQ*FI(1)*DXM(IELNO)/2.0
      CJC(6,4)= CJC(6,4)  + W(K)*FI(2)*dG2dAR*FI(1)*DXM(IELNO)/2.0
C
      CJC(6,5)= CJC(6,5)  + W(K)*FI(2)*dG2dAL*FI(2)*DXM(IELNO)/2.0
      CJC(6,6)= CJC(6,6)  + W(K)*FI(2)*dG2dA*FI(2)*DXM(IELNO)/2.0
      CJC(6,7)= CJC(6,7)  + W(K)*FI(2)*dG2dQ*FI(2)*DXM(IELNO)/2.0
      CJC(6,8)= CJC(6,8)  + W(K)*FI(2)*dG2dAR*FI(2)*DXM(IELNO)/2.0
C
      CJC(7,1)= CJC(7,1)  + W(K)*FI(2)*dG3dAL*FI(1)*DXM(IELNO)/2.0
      CJC(7,2)= CJC(7,2)  + W(K)*FI(2)*dG3dA*FI(1)*DXM(IELNO)/2.0
      CJC(7,3)= CJC(7,3)  + W(K)*FI(2)*dG3dQ*FI(1)*DXM(IELNO)/2.0
      CJC(7,4)= CJC(7,4)  + W(K)*FI(2)*dG3dAR*FI(1)*DXM(IELNO)/2.0
C
      CJC(7,5)= CJC(7,5)  + W(K)*FI(2)*dG3dAL*FI(2)*DXM(IELNO)/2.0
      CJC(7,6)= CJC(7,6)  + W(K)*FI(2)*dG3dA*FI(2)*DXM(IELNO)/2.0
      CJC(7,7)= CJC(7,7)  + W(K)*FI(2)*dG3dQ*FI(2)*DXM(IELNO)/2.0
      CJC(7,8)= CJC(7,8)  + W(K)*FI(2)*dG3dAR*FI(2)*DXM(IELNO)/2.0
C
      CJC(8,1)= CJC(8,1)  + W(K)*FI(2)*dG4dAL*FI(1)*DXR(IELNO)/2.0
      CJC(8,2)= CJC(8,2)  + W(K)*FI(2)*dG4dA*FI(1)*DXR(IELNO)/2.0
      CJC(8,3)= CJC(8,3)  + W(K)*FI(2)*dG4dQ*FI(1)*DXR(IELNO)/2.0
      CJC(8,4)= CJC(8,4)  + W(K)*FI(2)*dG4dAR*FI(1)*DXR(IELNO)/2.0
C
      CJC(8,5)= CJC(8,5)  + W(K)*FI(2)*dG4dAL*FI(2)*DXR(IELNO)/2.0
      CJC(8,6)= CJC(8,6)  + W(K)*FI(2)*dG4dA*FI(2)*DXR(IELNO)/2.0
      CJC(8,7)= CJC(8,7)  + W(K)*FI(2)*dG4dQ*FI(2)*DXR(IELNO)/2.0
      CJC(8,8)= CJC(8,8)  + W(K)*FI(2)*dG4dAR*FI(2)*DXR(IELNO)/2.0
C
        RETURN
         END
C
C   ******** CALCULATION OF THE JACOBIAN PART 4 *******
C
     SUBROUTINE PART4(K,W,FI,DFIDS,dE2dAL,dE2dA,
    + dE2dQ,dE2dAR,dE3dAL,dE3dA,dE3dQ,dE3dAR,WK1,
    + WK2,WK3,WK4,CJD)
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION CJD(8,8),DFIDS(2),FI(2),W(3)
C
     COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
     COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
     COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
     COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
     COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
     COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
     COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
     COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
     COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
    +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
    +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
    +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C       WRITE(6,1)
```

```
1        FORMAT(2X,'I AM IN SUBROUTINE PART4',4(2X,F10.6))
C
      t=(WK1*DFIDS(1)*dE2dAL + WK2*DFIDS(1)*dE3dAL)*FI(1)
       CJD(2,1)= CJD(2,1)  + W(K)*t
C
      t=(WK1*DFIDS(1)*dE2dA + WK2*DFIDS(1)*dE3dA)*FI(1)
      CJD(2,2)= CJD(2,2)  + W(K)*t
C
      t=(WK1*DFIDS(1)*dE2dQ + WK2*DFIDS(1)*dE3dQ)*FI(1)
       CJD(2,3)= CJD(2,3)  + W(K)*t
C
      t=(WK1*DFIDS(1)*dE2dAR + WK2*DFIDS(1)*dE3dAR)*FI(1)
      CJD(2,4)= CJD(2,4)  + W(K)*t
C
      t=(WK1*DFIDS(1)*dE2dAL + WK2*DFIDS(1)*dE3dAL)*FI(2)
       CJD(2,5)= CJD(2,5)  + W(K)*t
C
      t=(WK1*DFIDS(1)*dE2dA + WK2*DFIDS(1)*dE3dA)*FI(2)
      CJD(2,6)= CJD(2,6)  + W(K)*t
C
      t=(WK1*DFIDS(1)*dE2dQ + WK2*DFIDS(1)*dE3dQ)*FI(2)
      CJD(2,7)= CJD(2,7)  + W(K)*t
C
      t=(WK1*DFIDS(1)*dE2dAR + WK2*DFIDS(1)*dE3dAR)*FI(2)
      CJD(2,8)= CJD(2,8)  + W(K)*t
C
      t=(WK3*DFIDS(1)*dE2dAL + WK4*DFIDS(1)*dE3dAL)*FI(1)
       CJD(3,1)= CJD(3,1)  + W(K)*t
C
      t=(WK3*DFIDS(1)*dE2dA + WK4*DFIDS(1)*dE3dA)*FI(1)
      CJD(3,2)= CJD(3,2)  + W(K)*t
C
      t=(WK3*DFIDS(1)*dE2dQ + WK4*DFIDS(1)*dE3dQ)*FI(1)
      CJD(3,3)= CJD(3,3)  + W(K)*t
C
      t=(WK3*DFIDS(1)*dE2dAR + WK4*DFIDS(1)*dE3dAR)*FI(1)
       CJD(3,4)= CJD(3,4)  + W(K)*t
C
      t=(WK3*DFIDS(1)*dE2dAL + WK4*DFIDS(1)*dE3dAL)*FI(2)
       CJD(3,5)= CJD(3,5)  + W(K)*t
C
      t=(WK3*DFIDS(1)*dE2dA + WK4*DFIDS(1)*dE3dA)*FI(2)
       CJD(3,6)= CJD(3,6)  + W(K)*t
C
      t=(WK3*DFIDS(1)*dE2dQ + WK4*DFIDS(1)*dE3dQ)*FI(2)
       CJD(3,7)= CJD(3,7)  + W(K)*t
C
      t=(WK3*DFIDS(1)*dE2dAR + WK4*DFIDS(1)*dE3dAR)*FI(2)
       CJD(3,8)= CJD(3,8)  + W(K)*t
C
      t=(WK1*DFIDS(2)*dE2dAL + WK2*DFIDS(2)*dE3dAL)*FI(1)
       CJD(6,1)= CJD(6,1)  + W(K)*t
C
      t=(WK1*DFIDS(2)*dE2dA + WK2*DFIDS(2)*dE3dA)*FI(1)
       CJD(6,2)= CJD(6,2)  + W(K)*t
C
      t=(WK1*DFIDS(2)*dE2dQ + WK2*DFIDS(2)*dE3dQ)*FI(1)
      CJD(6,3)= CJD(6,3)  + W(K)*t
```

```
C
      t=(WK1*DFIDS(2)*dE2dAR + WK2*DFIDS(2)*dE3dAR)*FI(1)
       CJD(6,4)= CJD(6,4) + W(K)*t
C
      t=(WK1*DFIDS(2)*dE2dAL + WK2*DFIDS(2)*dE3dAL)*FI(2)
       CJD(6,5)= CJD(6,5) + W(K)*t
C
      t=(WK1*DFIDS(2)*dE2dA + WK2*DFIDS(2)*dE3dA)*FI(2)
       CJD(6,6)= CJD(6,6) + W(K)*t
C
      t=(WK1*DFIDS(2)*dE2dQ + WK2*DFIDS(2)*dE3dQ)*FI(2)
       CJD(6,7)= CJD(6,7) + W(K)*t
C
      t=(WK1*DFIDS(2)*dE2dAR + WK2*DFIDS(2)*dE3dAR)*FI(2)
      CJD(6,8)= CJD(6,8) + W(K)*t
C
      t=(WK3*DFIDS(2)*dE2dAL + WK4*DFIDS(2)*dE3dAL)*FI(1)
       CJD(7,1)= CJD(7,1) + W(K)*t
C
      t=(WK3*DFIDS(2)*dE2dA + WK4*DFIDS(2)*dE3dA)*FI(1)
      CJD(7,2)= CJD(7,2) + W(K)*t
C
      t=(WK3*DFIDS(2)*dE2dQ + WK4*DFIDS(2)*dE3dQ)*FI(1)
      CJD(7,3)= CJD(7,3) + W(K)*t
C
      t=(WK3*DFIDS(2)*dE2dAR + WK4*DFIDS(2)*dE3dAR)*FI(1)
      CJD(7,4)= CJD(7,4) + W(K)*t
C
      t=(WK3*DFIDS(2)*dE2dAL + WK4*DFIDS(2)*dE3dAL)*FI(2)
       CJD(7,5)= CJD(7,5) + W(K)*t
C
      t=(WK3*DFIDS(2)*dE2dA + WK4*DFIDS(2)*dE3dA)*FI(2)
      CJD(7,6)= CJD(7,6) + W(K)*t
C
      t=(WK3*DFIDS(2)*dE2dQ + WK4*DFIDS(2)*dE3dQ)*FI(2)
      CJD(7,7)= CJD(7,7) + W(K)*t
C
      t=(WK3*DFIDS(2)*dE2dAR + WK4*DFIDS(2)*dE3dAR)*FI(2)
      CJD(7,8)= CJD(7,8) + W(K)*t
C
         RETURN
          END
C
C  *** THIS PART  . THE DERIVATIVES FROM THE CONSERVATIVE EQUATIONS
***
C
      SUBROUTINE DERIVF(IELNO,AL,AR,QL,QR,BL,BR,HL,HR,QQ,AA,H,
     +B,tAL1,tAL2,dF1dA,dF1dQ,dF1dAR,dF2dAL,dF2dA,dF2dQ,dF2dAR,dF3dAL,
     +dF3dA,dF3dQ,dF3dAR,dF4dAL,dF4dA,dF4dQ,tAR1,tAR2,KLP,TML,TMR,
     +QQR,QQL,XT,VL,VR)
C
       IMPLICIT REAL *8(A-H,O-Z)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVR(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),ARM(200),QPHI(200),ARM(200),TETA,FC(200)
```

```
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),Q[illegible]0),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,[illegible]ARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
      TIME=XT/60.0
C        IF(TIME.GT.46.30)THEN
C        WRITE(6,1)
1       FORMAT(2X,'I AM IN SUBROUTINE DERIVF')
C         ENDIF
C
      AL1=AL
      AR1=AR
      HL1=HL
      HR1=HR
      QL1=QL
      QR1=QR
C
C        IF(TIME.GT.46.30)THEN
C       WRITE(6,9)0,AL1,HL1,QL1,COS(TETA)
C       ENDIF
C
       Z = Z1 + Z2
C
       IF(HL1.LE.0.0D+00)THEN
       HL1=0.001D+00
       ENDIF
C
       IF(AL1.LE.0.0D+00)THEN
       AL1=HL1*BL
       ENDIF
C
C        IF(TIME.GT.46.30)THEN
C       WRITE(6,9)0,AL1,HL1,QL1
C       ENDIF
C
        IF(KLP.EQ.1) GO TO 5
        R=AL1/(PAR(1,3) + HL1)
       IF(R.LE.0.0D+00)THEN
       CS=6.2
       ELSE
       A1= R/CN2(IELNO)
       CS= 5.75*DLOG10(A1) + 6.2D+00
       ENDIF
5        CONTINUE
C
       UL=QL1/AL1
C
       IF(QL1.EQ.0.0D+00)THEN
       tAL1= 0.0D+00
       tAL2 =0.0D+00
      GO TO 13
      ENDIF
C
       A1=BL**2 + AL1
```

```
         IF(KLP.EQ.1)THEN
         A2=-2.0*QL1/(3.0*A1)
         A3=2.0*QL1*AL1/(3.0*A1**2)
         A4=-5.0*AL1**2.3333*BL**1.333/(3.0*CN2(IELNO)**2
     +   *A1**1.333*ABS(QL1))
         A5=2.0*AL1**3.3333*BL**1.333/(3.0*CN2(IELNO)**2
     +   *A1**2.333*ABS(QL1))
         A6= -7.0*TML*AL1**2.333*BL**1.333/(6.0*CN2(IELNO)**2
     +   *GRAV*A1**2.333*ABS(QL1))
         A7= -7.0*VL*QQL*AL1**2.333*BL**1.333/
     +   (GRAV*6.0*CN2(IELNO)**2*A1**2.333*ABS(QL1))
        ELSE
         A2= -QL1/A1
         A3=QL1*AL1/A1**2
         A4=-3.0*GRAV*CS**2*AL1**2*BL/(2.0*A1*ABS(QL1))
         A5= GRAV*CS**2*AL1**3*BL/(A1**2*2.0*ABS(QL1))
         ENDIF
         tAL1= A2 + A3 + A6 + A7
         tAL2 =(A4 + A5)*HL1
C
13        dF1dA=0.0D+00
        dF1dQ=0.0D+00
        dF1dAR=0.0+00
C
        dF2dAL=0.0D+00
        dF2dA=0.0D+00
        dF2dQ= 1.0D+00
        dF2dAR=0.0+00
C
         IF(Z.EQ.0.0D+00)THEN
        AB=0.0D+00
        ELSE
        AD= B**2+2.0*AA*Z
        AB=-3.0*GRAV*Z*H**2/(DSQRT(AD)*4.0)
C          AB=-3.0*GRAV*Z*H**2/(4.0* B)
        ENDIF
C
C          AB=0.0D+00
C
        dF3dAL=0.0D+00
        dF3dA= -(QQ**2/AA**2) + GRAV*H + AB
        dF3dQ= 2.0*QQ/AA
        dF3dAR=0.0+00
C
        dF4dAL=0.0D+00
        dF4dA=0.0D+00
        dF4dQ= 0.0D+00
C
         IF(HR1.LE.0.0D+00)THEN
         HR1=0.001D+00
         ENDIF
C
         IF(AR1.LE.0.0D+00)THEN
         AR1=HR1*BR
         ENDIF
C
C          IF(TIME.GT.46.30)THEN
C         WRITE(6,9)0,AR1,HR1,QR1
```

```
C         ENDIF
C
         IF(KLP.EQ.1) GO TO 6
         R=AR1/(PAR(1,3) + 2*HR1)
        IF(R.LE.0.0D+00)THEN
         CS=6.2
         ELSE
         A1= R/CN3(IELNO)
         CS= 5.75*DLOG10(A1) + 6.2D+00
         ENDIF
6         CONTINUE
C
        UR=QR1/AR1
C
        IF(QR1.EQ.0.0D+00)THEN
        tAR1= 0.0D+00
        tAR2 =0.0D+00
       GO TO 8
       ENDIF
C
        A1=BR**2 + AR1
        IF(KLP.EQ.1)THEN
C          WRITE(6,9)1,A1,QR1,CN3(IELNO),BR
        A2=-2.0*QR1/(3.0*A1)
        A3=2.0*QR1*AR1/(3.0*A1**2)
        A4=-5.0*AR1**2.3333*BR**1.333/(3.0*CN3(IELNO)**2
     + *A1**1.333*ABS(QR1))
        A5=2.0*AR1**3.3333*BR**1.333/(3.0*CN3(IELNO)**2
     + *A1**2.333*ABS(QR1))
        A6= -7.0*TMR*AR1**2.333*BR**1.333/(6.0*CN3(IELNO)**2
     + *GRAV*A1**2.333*ABS(QR1))
        A7= -7.0*VR*QQR*AR1**2.333*BR**1.333/
     + (6.0*GRAV*CN3(IELNO)**2*A1**2.333*ABS(QR1))
C        WRITE(6,9)2
       ELSE
        A2= -QR1/A1
        A3= QR1*AR1/A1**2
        A4=-3.0*GRAV*CS**2*AR1**2*BR/(2.0*A1*ABS(QR1))
        A5= GRAV*CS**2*AR1**3*BR/(A1**2*2.0*ABS(QR1))
        ENDIF
        tAR1= A2 + A3 + A6 + A7
        tAR2 =(A4 + A5)*HR1
C
8      CONTINUE
9       FORMAT(2X,'I AM HERE',I2,5(2X,F12.6))
C         WRITE(6,20)tAL1,tAL2,dF1dA,dF1dQ,dF1dAR
C        WRITE(6,20)dF2dAL,dF2dA,dF2dQ,dF2dAR
C         WRITE(6,20)dF3dAL,dF3dA,dF3dQ,dF3dAR
C         WRITE(6,20)dF4dAL,dF4dA,dF4dQ,tAR1,tAR2
20      FORMAT(2X,6(2X,F12.6))
       RETURN
       END
C
C  *** THIS PART OF THE DERIVATIVE IS FROM THE NON-CONSERVATIVE
EQUATIONS ***
C
       SUBROUTINE DERIVD(U,H,B,AA,dD1dAL,dD1dA,dD1dQ,dD1dAR,dD2dAL,dD2dA,
     +dD2dQ,dD2dAR,dD3dAL,dD3dA,dD3dQ,dD3dAR,dD4dAL,dD4dA,dD4dQ,dD4dAR,
```

```
     +XT)
C
       IMPLICIT REAL *8(A-H,O-Z)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
      TIME=XT/60.0
C        IF(TIME.GT.46.30)THEN
C        WRITE(6,1)
1       FORMAT(2X,'I AM IN SUBROUTINE DERIVD')
C        ENDIF
C
      Z = Z1 + Z2
C
      dD1dAL=0.0D+00
      dD1dA=0.0D+00
      dD1dQ=0.0D+00
      dD1dAR=0.0D+00
C
      dD2dAL=0.0D+00
      dD2dA=0.0D+00
      dD2dQ=1.0D+00
      dD2dAR=0.0D+00
C
       IF(Z.EQ.0.0D+00)THEN
      AB=0.0D+00
      ELSE
      AD= B**2+2.0*AA*Z
      AB=-3.0*GRAV*Z*H**2/(DSQRT(AD)*2.0)
C        AB=-3.0*GRAV*Z*H**2/(B*2.0)
      ENDIF
C
      AB=0.0D+00
C
      dD3dAL=0.0D+00
      dD3dA= 2.0*GRAV*H-U**2 + AB
      dD3dQ= 2.0*U
      dD3dAR=0.0D+00
C
      dD4dAL=0.0D+00
      dD4dA=0.0D+00
      dD4dQ=0.0D+00
      dD4dAR=0.0D+00
C
C      WRITE(6,20)dD1dAL,dD1dA,dD1dQ,dD1dAR
C      WRITE(6,20)dD2dAL,dD2dA,dD2dQ,dD2dAR
C       WRITE(6,20)dD3dAL,dD3dA,dD3dQ,dD3dAR
```

```
C       WRITE(6,20)dD4dAL,dD4dA,dD4dQ,dD4dAR
20      FORMAT(2X,4(2X,F12.6))
      RETURN
      END
C
C  *** THIS PART OF THE DERIVATIVE IS FROM THE NON-CONSERVATIVE
EQUATIONS (G2i) ***
C
      SUBROUTINE DERIVE(IELNO,So,QQ,AA,B,dBdX,KFL,KLP,AL,AR,QL,QR,BL,
     +BR,HL,HR,dE1dAL,dE1dA,dE1dQ,dE1dAR,dE2dAL,dE2dA,dE2dQ,dE2dAR,
     +dE3dAL,dE3dA,dE3dQ,dE3dAR,TML,TMR,dE4dAL,dE4dA,dE4dQ,dE4dAR,DDHL,
     +DDHR,QQR,QQL,HSTEP,VL,VR,L)
C
       IMPLICIT REAL *8(A-H,O-Z)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C        WRITE(6,1)
1       FORMAT(2X,'I AM IN SUBROUTINE DERIVE'4(2X,F12.6))
C
C        WRITE(6,4)QQL,QQR,AA,QQ
C        WRITE(6,4)QL,QR
      HL1=HL
      HR1=HR
      AL1=AL
      AR1=AR
      QL1=QL
      QR1=QR
C
      Z = Z1 + Z2
C
      IF(HL1.LE.0.0D+00)THEN
      HL1=0.001D+00
      ENDIF
C
       IF(AL1.LE.0.0D+00)THEN
       AL1= HL1*BL
       ENDIF
C
      IF(HR1.LE.0.0D+00)THEN
      HR1=0.001D+00
      ENDIF
C
       IF(AR1.LE.0.0D+00)THEN
       AR1= HR1*BR
       ENDIF
C
```

```
      UL=QL1/AL1
      UR=QR1/AR1
C
      dE1dAL=0.0D+00
      dE1dA=0.0D+00
      dE1dQ=0.0D+00
      dE1dAR=0.0D+00
      H=AA/B
C
      IF(KFL.EQ.1)THEN
      F=1.0D+00
      ELSE
      F=0.0D+00
      ENDIF
C
C        IF(KLP.EQ.1)THEN
         AX= 2.0*AA+B**2
C         BX= AA**2.333*B**1.333
C       B2=GRAV*QQ*DABS(QQ)*CN1(IELNO)**2*AX**1.333/(AA**3.333*B**1.333)
C       B3=GRAV*QQ*DABS(QQ)*CN1(IELNO)**2*AX**0.333/(AA**2.333*B**1.333)
C       ENDIF
C
       IF(KLP.EQ.1)THEN
       AX= B**1.333
       BX= AA**3.333
       B2=GRAV*QQ*DABS(QQ)*CN1(IELNO)**2*B**1.333/(AA**3.333)
       B3=-10.0*GRAV*QQ*DABS(QQ)*CN1(IELNO)**2*B**1.333/(3.0*AA**3.333)
        ENDIF
C
       T1= ELVLP(NODNUM(IELNO,L))+ HLSTEP(NODNUM(IELNO,L))-
     + ELVM (NODNUM(IELNO,L))
       IF(T1.GT.H)THEN
        T1=H
        ENDIF
C
C      BL=1.0
C      B=1.0
C      BR=1.0
C      AL1=1.0
C      AR1=1.0
C      AA=1.0
C
       A1=1.0D+00 + Z1**2
       A2=DSQRT(A1)
       B4=1.0D+00 + Z2**2
       B5=DSQRT(B4)
C
       P= B + T1*(A2+B5)
C
       R=AA/P
       A4=R/CN1(IELNO)
       CS= 5.75*DLOG10(A4) + 6.2D+00
C
       BA=2.0*GRAV
       A1=BL**2 + 2.0*Z3*AL1
       A2=B**2 + 2.0*Z*AA
       A3=BR**2 + 2.0*Z4*AR1
C
```

```
       IF(DDHL.EQ.0.0D+00)THEN
       dQLdAL=0.0D+00
       dQLdA=0.0D+00
       dQLdQ=0.0D+00
       dQLdAR=0.0D+00
       GO TO 2
       ELSEIF(DDHL.GT.0.0D+00)THEN
       dQLdAL=0.0D+00
       dQLdA= CML(IELNO)*DSQRT(BA)*DDHL**0.5/(A2**0.5)
       dQLdQ=0.0D+00
       dQLdAR=0.0D+00
       ELSE
       dQLdAL= -CML(IELNO)*DSQRT(BA)*(DABS(DDHL))**0.5/(A1**0.5)
       dQLdA=0.0D+00
       dQLdQ=0.0D+00
       dQLdAR=0.0D+00
       ENDIF
2      CONTINUE
C       WRITE(6,20)dQLdAL,dQLdA,dQLdQ,dQLdAR
C
      IF(DDHR.EQ.0.0D+00)THEN
       dQRdAL=0.0D+00
       dQRdA=0.0D+00
       dQRdQ=0.0D+00
       dQRdAR=0.0D+00
       GO TO 5
       ELSEIF(DDHR.GT.0.0D+00)THEN
      dQRdAL=0.0D+00
      dQRdA= CMR(IELNO)*DSQRT(BA)*DDHR**0.5/(A2**0.5)
      dQRdQ=0.0D+00
      dQRdAR=0.0D+00
      ELSE
      dQRdAL=0.0D+00
      dQRdA= 0.0D+00
      dQRdQ=0.0D+00
       dQRdAR= -CMR(IELNO)*DSQRT(BA)*(DABS(DDHR))**0.5/(A3**0.5)
      ENDIF
5       CONTINUE
C       WRITE(6,20)dQRdAL,dQRdA,dQRdQ,dQRdAR
C
      dE2dAL= dQLdAL + dQRdAL
      dE2dA= dQLdA + dQRdA
      dE2dQ= dQLdQ + dQRdQ
      dE2dAR= dQLdAR +dQRdAR
C
C         IF(TML.EQ.0.0D+00)THEN
      dMLdAL=0.0D+00
      dMLdA=0.0D+00
      dMLdQ=0.0D+00
C          ENDIF
4      FORMAT(2X,5(2X,F12.6))
C
10        CONTINUE
C
C          WRITE(6,20)QL1,VL
       IF(QQL.GT.0.0D+00)THEN
       dE3dAL= VL*dQLdAL + dMLdAL
       ELSEIF(QQL.LT.0.0D+00)THEN
```

```
      dE3dAL= -QL1*QQL/AL1**2 + VL*dQLdAL +dMLdAL
      ELSE
      dE3dAL=0.0D+00
      ENDIF
C
C        IF(TMR.EQ.0.0D+00)THEN
     dMRdA=0.0D+00
     dMRdQ=0.0D+00
C        ENDIF
12     CONTINUE
C
      IF(QQL.GT.0.0D+00)THEN
      Y1= -QQ*QQL/AA**2 + VL*dQLdA
      R1= QQL/AA +VL*dQLdQ
      ELSEIF(QQL.LT.0.0D+00)THEN
      Y1= VL*dQLdA
      R1=0.0D+00
      ELSE
      Y1= 0.0D+00
      R1= 0.0D+00
      ENDIF
C
       IF(QQR.GT.0.0D+00)THEN
      Y2= -QQ*QQR/AA**2 + VR*dQRdA
      R2= QQR/AA +VR*dQRdQ
      ELSEIF(QQR.LT.0.0D+00)THEN
      Y2= VR*dQLdA
      R2=0.0D+00
      ELSE
      Y2= 0.0D+00
      R2= 0.0D+00
      ENDIF
C
C        WRITE(6,20)Y1,Y2,R1,R2
      ZZ1= Y1 + Y2
      RR1= R1 + R2
C
C       B3=GRAV*QQ*DABS(QQ)*CN1(IELNO)**2/(AA**2*R**1.333)
C
     IF(Z.EQ.0.0D+00)THEN
     AB=-2.0*GRAV*AA*dBdX/B**2
     ELSE
     AD= B**2+2.0*AA*Z
     AB= -2.0*GRAV*H*dBdX/DSQRT(AD)
     ENDIF
C
           IF(KLP.EQ.1)THEN
      dE3dA= AB-GRAV*So + B2*F + B3*F+ZZ1+dMLdA+dMRdA
C
      dE3dQ= 2.*F*GRAV*QQ*CN1(IELNO)**2*AX/BX +RR1 + dMLdQ + dMRdQ
            ELSE
     dE3dA= AB - GRAV*So -2.*QQ*DABS(QQ)*F*B/
    + (AA**3*CS**2)- 5.0*QQ*DABS(QQ)*F/(AA**2*CS**3*R) + ZZ1 +
    +  dMLdA+ dMRdA
C
      dE3dQ=2.*F*QQ*B/(AA**2*CS**2)+RR1+dMLdQ+dMRdQ
         ENDIF
C
```

```
C         IF(TMR.EQ.0.0D+00)THEN
       dMRdAR=0.0D+00
C         ENDIF
C
15      CONTINUE
C
C          WRITE(6,20)QR1,VR
        IF(QQR.GT.0.0D+00)THEN
        dE3dAR= VR*dQRdAR + dMRdAR
        ELSEIF(QQR.LT.0.0D+00)THEN
        dE3dAR= -QR1*QQR/AR1**2 + VR*dQRdAR + dMRdAR
        ELSE
        dE3dAR=0.0D+00
        ENDIF
C
       dE4dAL=0.0D+00
       dE4dA=0.0D+00
       dE4dQ=0.0D+00
       dE4dAR=0.0D+00
C
C        WRITE(6,20)dE1dAL,dE1dA,dE1dQ,dE1dAR
C         WRITE(6,20)dE2dAL,dE2dA,dE2dQ,dE2dAR
C          WRITE(6,20)dE3dAL,dE3dA,dE3dQ,dE3dAR
C        WRITE(6,20)dE4dAL,dE4dA,dE4dQ,dE4dAR
20      FORMAT(2X,5(2X,F12.6))
C
           RETURN
            END
C
C  *** THIS PART OF THE DERIVATIVE IS FROM THE CONSERVATIVE EQUATIONS
***************
C
       SUBROUTINE DERIVG(IELNO,So,QQ,AA,B,dBdX,KFL,KLP,AL,AR,QL,QR,BL,
      +BR,HL,HR,dG1dAL,dG1dA,dG1dQ,dG1dAR,dG2dAL,dG2dA,dG2dQ,dG2dAR,
      +dG3dAL,dG3dA,dG3dQ,dG3dAR,TML,TMR,dG4dAL,dG4dA,dG4dQ,dG4dAR,
      +DDHL,DDHR,QQR,QQL,HSTEP,VL,VR,L)
C
        IMPLICIT REAL *8(A-H,O-Z)
C
       COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
       COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
       COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
       COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
       COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
       COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
       COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
       COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
       COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
      +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
      +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
      +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C           WRITE(6,1)
1        FORMAT(2X,'I AM IN SUBROUTINE DERIVG'4(2X,F12.6))
C
C         WRITE(6,4)QQL,QQR,AA,QQ
C         WRITE(6,4)QL,QR
```

```
C
      HL1=HL
      HR1=HR
      AL1=AL
      AR1=AR
      QL1=QL
      QR1=QR
C

      Z= Z1 + Z2
C
      IF(HL1.LE.0.0D+00)THEN
      HL1=0.001D+00
      ENDIF
C
       IF(AL1.LE.0.0D+00)THEN
       AL1= HL1*BL
       ENDIF

      IF(HR1.LE.0.0D+00)THEN
      HR1=0.001D+00
      ENDIF
C
       IF(AR1.LE.0.0D+00)THEN
       AR1= HR1*BR
       ENDIF

      H=AA/B
      IF(KFL.EQ.1)THEN
      F=1.0D+00
      ENDIF
      IF(KFL.EQ.0)THEN
      F=0.0D+00
      ENDIF
C
C      IF(KLP.EQ.1)THEN
C       AX=2.0*AA+B**2
C        BX= AA**2.333*B**1.333
C      B2=GRAV*QQ*DABS(QQ)*CN1(IELNO)**2*AX**1.333/(AA**3.333*B**1.333)
C      B3=GRAV*QQ*DABS(QQ)*CN1(IELNO)**2*AX**0.333/(AA**2.333*B**1.333)
C          ENDIF
C
       IF(KLP.EQ.1)THEN
       AX= B**1.333
       BX= AA**3.333
       B2=GRAV*QQ*DABS(QQ)*CN1(IELNO)**2*B**1.333/(AA**3.333)
       B3=-10.0*GRAV*QQ*DABS(QQ)*CN1(IELNO)**2*B**1.333/(3.0*AA**3.333)
       ENDIF
C
       T1= ELVLP(NODNUM(IELNO,L))+ HLSTEP(NODNUM(IELNO,L))-
     + ELVMc(NODNUM(IELNO,L))
       IF(T1.GT.H)THEN
        T1=H
        ENDIF
C
       A1=1.0D+00 + Z1**2
       A2=DSQRT(A1)
       B4=1.0D+00 + Z2**2
```

```
      B5=DSQRT(B4)
C
      P= B + T1*(A2+B5)
C
      RC=AA/P
      A4=RC/CN1(IELNO)
      CS1= 5.75*DLOG10(A4) + 6.2D+00
C
      UL=QL1/AL1
      UR=QR1/AR1
C
C     BL=1.0
C     B=1.0
C     BR=1.0
C     AL1=1.0
C     AR1=1.0
C     AA=1.0
C
      BA=2.0*GRAV
      A1=BL**2 + 2.0*Z3*AL1
      A2=B**2 + 2.0*Z*AA
      A3=BR**2 + 2.0*Z4*AR1
C
C      WRITE(6,4)DDHL,HL1,DDHR,HR1
      IF(DDHL.EQ.0.0D+00)THEN
      dQLdAL=0.0D+00
      dQLdA=0.0D+00
      dQLdQ=0.0D+00
      dQLdAR=0.0D+00
      GO TO 2
      ELSEIF(DDHL.GT.0.0D+00)THEN
      dQLdAL=0.0D+00
      dQLdA= -CML(IELNO)*DSQRT(BA)*DDHL**0.5/(A2**0.5)
      dQLdQ=0.0D+00
      dQLdAR=0.0D+00
      ELSEIF(DDHL.LT.0.0D+00)THEN
      dQLdAL= CML(IELNO)*DSQRT(BA)*(DABS(DDHL))**0.5/(A1**0.5)
      dQLdA=0.0D+00
      dQLdQ=0.0D+00
      dQLdAR=0.0D+00
      ENDIF
2      CONTINUE
C      WRITE(6,4)dQLdAL,dQLdA,dQLdQ,dQLdAR
      IF(DDHR.EQ.0.0D+00)THEN
      dQRdAL=0.0D+00
      dQRdA=0.0D+00
      dQRdQ=0.0D+00
      dQRdAR=0.0D+00
      GO TO 5
      ELSEIF(DDHR.GT.0.0D+00)THEN
     dQRdAL=0.0D+00
     dQRdA= -CMR(IELNO)*DSQRT(BA)*DDHR**0.5/(A2**0.5)
     dQRdQ=0.0D+00
     dQRdAR=0.0D+00
     ELSE
     dQRdAL=0.0D+00
     dQRdA= 0.0D+00
     dQRdQ=0.0D+00
```

```
      dQRdAR= CMR(IELNO)*DSQRT(BA)*(DABS(DDHR))**0.5/(A1**0.5)
      ENDIF
5       CONTINUE
C        WRITE(6,4)dQRdAL,dQRdA,dQRdQ,dQRdAR
4       FORMAT(2X,5(2X,F12.6))
C
      dG1dAL= dQLdAL
      dG1dA= dQLdA
      dG1dQ= dQLdQ
      dG1dAR= dQLdAR
C
      dG2dAL= -(dQLdAL + dQRdAL)
      dG2dA=  -(dQLdA + dQRdA )
      dG2dQ=  -(dQLdQ + dQRdQ)
      dG2dAR= -(dQRdAR +dQLdAR)
C
C        IF(TML.EQ.0.0D+00)THEN
      dMLdAL=0.0D+00
      dMLdQ=0.0D+00
      dMLdA=0.0D+00
C        ENDIF
C
C       WRITE(6,9)dMLdAL,dMLdQ,dMLdA
9      FORMAT(2X,3(2X,F12.6))
10    CONTINUE
C
C        IF(TMR.EQ.0.0D+00)THEN
      dMRdA=0.0D+00
      dMRdQ=0.0D+00
C        ENDIF
15     CONTINUE
C
C        WRITE(6,20)QL1,VL
       IF(QQL.GT.0.0D+00)THEN
       dG3dAL= -VL*dQLdAL + dMLdAL
       ELSEIF(QQL.LT.0.0D+00)THEN
       dG3dAL= -QL*QQL/AL1**2 - VL*dQLdAL + dMLdAL
       ELSE
       dG3dAL=0.0D+00
       ENDIF
C
C        IF(TMR.EQ.0.0D+00)THEN
      dMRdA=0.0D+00
      dMRdQ=0.0D+00
C        ENDIF
12      CONTINUE
C
       IF(QQL.GT.0.0D+00)THEN
       Y1= -QQ*QQL/AA**2 - VL*dQLdA
       R1= QQL/AA -VL*dQLdQ
       ELSEIF(QQL.LT.0.0D+00)THEN
       Y1= -VL*dQLdA
       R1=0.0D+00
       ELSE
       Y1=0.0D+00
       R1=0.0D+00
       ENDIF
C
```

```
        IF(QQR.GT.0.0D+00)THEN
        Y2= -QQ*QQR/AA**2 - VR*dQRdA
        R2= QQR/AA - VR*dQRdQ
        ELSEIF(QQR.LT.0.0D+00)THEN
        Y2= -VR*dQLdA
        R2=0.0D+00
        ELSE
        Y2=0.0D+00
        R2=0.0D+00
        ENDIF
C
C         WRITE(6,20)Y1,Y2,R1,R2
        ZZ1= Y1 + Y2
        RR1= R1 + R2
C
        IF(Z.EQ.0.0D+00)THEN
       AB=-GRAV*AA*dBdX/B**2
       ELSE
       AD= B**2+2.0*AA*Z
       AB= -GRAV*H*dBdX/DSQRT(AD)
       ENDIF
C
        IF(KLP.EQ.1)THEN
        dG3dA= AB-GRAV*So + B2*F + B3*F+ZZ1+dMLdA+dMRdA
C
        dG3dQ= 2.*F*GRAV*QQ*CN1(IELNO)**2*AX/BX +RR1 + dMLdQ + dMRdQ
              ELSE
       dG3dA= AB - GRAV*So -2.0*QQ*DABS(QQ)*F*B/
     + (AA**3*CS1**2)- 5.0*QQ*DABS(QQ)*F/(AA**2*CS1**3*RC)+ZZ1 +
     + dMLdA + dMRdA
C
       dG3dQ=2.0*F*QQ*B/(AA**2*CS1**2)+RR1 + dMLdQ + dMRdQ
            ENDIF
C
C         IF(TMR.EQ.0.0D+00)THEN
        dMRdAR=0.0D+00
C         ENDIF
16      CONTINUE
C
C        WRITE(6,17)dMRdAR,dMRdQ,dMRdA
17       FORMAT(2X,3(2X,F12.6))
C
C          WRITE(6,20)QR1,VR
        IF(QQR.GT.0.0D+00)THEN
        dG3dAR= -VR*dQRdAR + dMRdAR
        ELSEIF(QQR.LT.0.0D+00)THEN
        dG3dAR= -QR1*QQR/AR1**2 - VR*dQRdAR + dMRdAR
        ELSE
        dG3dAR=0.0D+00
        ENDIF
C
       dG4dAL= dQRdAL
       dG4dA= dQRdA
       dG4dQ= dQRdQ
       dG4dAR= dQRdAR
C
C          WRITE(6,20) dG1dAL,dG1dA,dG1dQ,dG1dAR
C          WRITE(6,20) dG2dAL,dG2dA,dG2dQ,dG2dAR
```

```
C         WRITE(6,20) dG3dAL,dG3dA,dG3dQ,dG3dAR
C         WRITE(6,20) dG4dAL,dG4dA,dG4dQ,dG4dAR
20     FORMAT(2X,5(2X,F12.6))
C
          RETURN
           END
C
C ******* CALCULATION OF STIFFNESS MATRIX [EKaa]*****
C
      SUBROUTINE ELMKaa(IELNO,I,J,FI,DFIDS,WK2,U,So,H,t)
C
       IMPLICIT REAL *8(A-H,O-Z)
       DIMENSION FI(2),DFIDS(2)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C        WRITE(6,1)DXM(IELNO)
1       FORMAT(2X,'I AM IN SUBROUT..E ELMKaa',F10.3)
C
      t1 = (GRAV*H-U**2)*WK2*DFIDS   )*DFIDS(J)*2./DXM(IELNO)
      t2 = -GRAV*So*WK2*DFIDS(I)*F   J)
      t= t1+t2
      RETURN
      END
C
C ******* CALCULATION OF STIFFNESS MA...I. [EKaq]*****
C
      SUBROUTINE ELMKaq(IELNO,I,J,FI,DFIDS,WK1,WK2,U,FF,t)
C
       IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION FI(2),DFIDS(2)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C        WRITE(6,1)DXM(IELNO)
1       FORMAT(2X,'I AM IN SUBROUTINE ELMKaq',F10.3)
```

```
C
      t1 = -DFIDS(I)*FI(J)
      t2 = WK1*DFIDS(I)*DFIDS(J)*2./DXM(IELNO)
      t3 = WK2*FF*DFIDS(I)*FI(J)
      t4 = WK2*2.0*U*DFIDS(I)*DFIDS(J)*2./DXM(IELNO)
      t = t1 + t2 +t3 + t4
C      WRITE(6,10)t
10    FORMAT(2X,F8.2)
      RETURN
      END
C
C ******* CALCULATION OF STIFFNESS MATRIX [EKqa]*****
C
      SUBROUTINE ELMKqa(IELNO,I,J,FI,DFIDS,WK4,U,So,H,t)
C
      IMPLICIT REAL *8(A-H,O-Z)
     DIMENSION FI(2),DFIDS(2)
C
     COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
     COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
     COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
     COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
     COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
     COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
     COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
     COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
     COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
    + Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
    +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
    +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C        WRITE(6,1)U,So,H,DXM(IELNO)
1      FORMAT(2X,'I AM IN SUBROUTINE ELMKqa',4(2X,F12.6))
C
      t1 = -GRAV*H/2.*DFIDS(I)*FI(J)
      t2 = -GRAV*So*FI(I)*FI(J)*DXM(IELNO)/2.
      t3 = -WK4*GRAV*So*DFIDS(I)*FI(J)
      t4 = (GRAV*H-U**2)*WK4*DFIDS(I)*DFIDS(J)*2./DXM(IELNO)
     t =t1 +t2 +t3 +t4
C        WRITE(6,10)t1,t2,t3,t4,t
10    FORMAT(2X,5(2X,F12.6))

      RETURN
      END
C
C ******* CALCULATION OF STIFFNESS MATRIX [EKqq]*****
C
     SUBROUTINE ELMKqq(IELNO,I,J,FI,DFIDS,WK3,WK4,U,FF,t)
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION FI(2),DFIDS(2)
C
     COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
     COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
     COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
     COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
     COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
     COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
```

```
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C        WRITE(6,1)DXM(IELNO)
1       FORMAT(2X,'I AM IN SUBROUTINE ELMKqq',F10.3)
C
      t1= -U*DFIDS(I)*FI(J) + FF*FI(I)*FI(J)*DXM(IELNO)/2.
      t2= WK4*FF*DFIDS(I)*FI(J)
      t3= WK3*DFIDS(I)*DFIDS(J)*2./DXM(IELNO)
      t4= WK4*2.0*U*DFIDS(I)*DFIDS(J)*2./DXM(IELNO)
       t= t1 + t2 + t3 + t4
C        WRITE(6,10)t1,t2,t3,t4,t
10      FORMAT(2X,5(2x,F10.6))
       RETURN
       END
C
C ******* CALCULATION OF STIFFNESS MATRIX [ALK]*****
C
      SUBROUTINE ELMALK(IELNO,I,J,AL,QL,BL,HL,QQL,TML,FI,DFIDS,
     +t,KLP,VL)
C
       IMPLICIT REAL *8(A-H,O-Z)
       DIMENSION FI(2),DFIDS(2)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C        WRITE(6,1)QQL,VL
1       FORMAT(2X,'I AM IN  SUBROUTINE ELMALK',2(2X,F10.6))
C
      IF((AL.LE.0.0D+00).OR.(QL.EQ.0.0D+00))THEN
C        IF(AL.LE.0.0D+00)THEN
         t=0.0D+00
         GO TO  15
         ENDIF
C
        UL=QL/AL
        ZL=DSQRT(1.0 + Z3**2)
        A1=BL + HL*ZL
        IF(KLP.EQ.1)THEN
        D1=AL**3.3333/(CN2(IELNO)**2*A1**1.333*2.0*ABS(QL))
C
      D2=AL**1.333*QQL*VL/(2.0*GRAV*CN2(IELNO)**2*A1**1.333*ABS(QL))
C
```

```
      D3  [illegible]**1.333*TML/(2.0*GRAV*CN2(IELNO)**2*A1**1.333*ABS(QL))
C
        F1  ( 5.0/3.0)*UL*DFIDS(I)*FI(J)
        F2  (D2+D3)*DFIDS(I)*FI(J)
       ELSE
        R  AL/A1
        R1  R/CN2(IELNO)
        CS  5.75*DLOG10(R1) + 6.2D+00
       D1 GRAV*CS**2*AL**3/(A1*2.0*ABS(QL))
       D2  CS**2*AL*QQL*VL/(A1*2.0*ABS(QL))
       D3  CS**2*AL*TML/(A1*2.0*ABS(QL))
       F1  1.5*UL*DFIDS(I)*FI(J)
       F2  (D2+D3)*DFIDS(I)*FI(J)
       ENDIF
        F  F1 + F2
C
C           WRITE(6,10)D1,D2,D3
C           WRITE(6,15)F1,F2,F
10      FORMAT(2X,8(F15.10))
15      RETURN
      END
C
C  ******* CALCULATION OF STIFFNESS MATRIX [ARF]*****
C
      SUBROUTINE ELMARF(IELNO,I,J,AR,QR,BR,HR,QQR,TMR,FI,DFIDS,
     +F,FLP,VR)
C
        IMPLICIT REAL *8(A [illegible]
       DIMENSION FI(2),DF[illegible]
C
      COMMON THETA,CN[illegible]2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,A[illegible](200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QIL(200),QIR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +QL(200),QLR(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARR(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),BLSTEP(200),
     +BRSTEP(200),CASEL(200),CASER(200),WALL
C
C         WRITE(6,1)QQR,VR
1       FORMAT(2X,'I AM IN SUBROUTINE ELMARK'2(2X,F10.6))
C
       IF((AR.LE.0.0D+00).OR.(QR.EQ.0.0D+00))THEN
C          IF(AR.LE.0.0D+00)THEN
         F  0.0D+00
         GO TO  15
         ENDIF
C
        UR  QR/AR
        ZR  DSQRT(1.0 + Z4**2)
        A1  BR + HR*ZR
        IF(FLP.EQ.1)THEN
        D1  AR**3.3333/(CN3(IELNO)**2*A1**1.333*2.0*ABS(QR))
C
```

```
      D2=AR**1.333*QQR*VR/(2.0*GRAV*CN3(IELNO)**2*
     + A1**1.333*ABS(QR))
C
      D3=AR**1.333*TMR/(2.0*GRAV*CN3(IELNO)**2*A1**1.333*ABS(QR))
C
        t1= (-5.0/3.0)*UR*DFIDS(I)*FI(J)
        t2= (D2+D3)*DFIDS(I)*FI(J)
       ELSE
        R=AR/A1
        B1= R/CN3(IELNO)
        CS= 5.75*DLOG10(B1) + 6.2D+00
       D1=GRAV*CS**2*AR**3/(A1*2.0*ABS(QR))
       D2= CS**2*AR*QQR*VR/(A1*2.0*ABS(QR))
       D3= CS**2*AR*TMR/(A1*2.0*ABS(QR))
       t1=-1.5*UR*DFIDS(I)*FI(J)
       t2= (D2+D3)*DFIDS(I)*FI(J)
       ENDIF
        t= t1 + t2
C
C         WRITE(6,10)D1,D2,D3
C         WRITE(6,10)t1,t2,t
10      FORMAT(2X,6(F15.10))
15      RETURN
      END
C
C ******* CALCULATION OF STIFFNESS MATRIX [FMC]*****
C
       SUBROUTINE FMC(IELNO,I,QQL,QQR,FI,DFIDS,WK1,WK2,WK3,WK4,TML,
     + TMR,R1,R2,dBdX,H,dHdX,VL,VR)
C
       IMPLICIT REAL *8(A-H,O-Z)
       DIMENSION FI(2),DFIDS(2)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C        WRITE(6,1)DXM(IELNO)
1       FORMAT(2X,'I AM IN SUBROUTINE FMC',2(2X,F10.6))
C
       Z= Z1 + Z2
C
       t1= FI(I)*(QQL+QQR)*DXM(IELNO)/2.
       t2= WK1*DFIDS(I)*(QQL+QQR)
       t3= WK2*DFIDS(I)*(VL*QQL+VR*QQR)
       t4= WK2*DFIDS(I)*(TML+TMR)
       t5= -GRAV*H**2*dBdX*WK2*DFIDS(I)
       t6= - GRAV*Z*H**2*WK2*DFIDS(I)*dHdX/2.0
C
```

```
       R1= t1 + t2 + t3 + t4 + t5 + t6
C        WRITE(6,10)t1,t2,t3,t4,R1
C         WRITE(6,10)t5,t6
C
       t1= FI(I)*(VL*QQL+VR*QQR)*DXM(IELNO)/2.0
       t2= FI(I)*(TML+TMR)*DXM(IELNO)/2.
       t3= WK3*DFIDS(I)*(QQL+QQR)
       t4= WK4*DFIDS(I)*(VL*QQL+VR*QQR)
       t5= WK4*DFIDS(I)*(TML+TMR)
       t6= -GRAV*H**2*dBdX*FI(I)*DXM(IELNO)/4.0
       t7= -GRAV*Z*H**2*FI(I)*dHdX*DXM(IELNO)/8.0
       t8= -GRAV*WK4*H**2*dBdX*DFIDS(I)
       t9= -GRAV*WK4*Z*H**2*DFIDS(I)*dHdX/2.0
C
       R2= t1 + t2 + t3 + t4 +t5 + t6 + t7 + t8 + t9
C
C         WRITE(6,10)t1,t2,t3,t4,t5,R2
C          WRITE(6,10)t6,t7,t8,t9
10       FORMAT(2X,6(2X,F12.6))
       RETURN
      END
C
C ******* CALCULATION OF STIFFNESS MATRIX [FL]*****
C
      SUBROUTINE FLK(IELNO,I,AL,QL,BL,HL,FI,DFIDS,QQL,t,KLP,HLdX)
C
       IMPLICIT REAL *8(A-H,O-Z)
       DIMENSION FI(2),DFIDS(2)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C         WRITE(6,1)DXL(IELNO)
1        FORMAT(2X,'I AM IN SUBROUTINE FLK',F10.3)
C
      IF((AL.LE.0.0D+00).OR.(QL.EQ.0.0D+00))THEN
C         IF(AL.LE.0.0D+00)THEN
          t=0.0D+00
          GO TO  15
          ENDIF
C
        UL=QL/AL
        ZL=DSQRT(1.0 + Z3**2)
        A1=BL + HL*ZL
C
        IF(KLP.EQ.0)THEN
        R=AL/A1
        B1= R/CN2(IELNO)
```

```
       CS= 5.75*DLOG10(B1) + 6.2D+00
      D1=CS**2*GRAV*AL**3/(2.0*A1*DABS(QL))
      B3= ZL*QL/(2.0*A1)
      ELSE
     D1=AL**3.333/(2.0*CN2(IELNO)**2*A1**1.333*ABS(QL))
      B3= 2.0*ZL*QL/(3.0*A1)
      ENDIF
      t3 = -FI(I)*QQL*DXL(IELNO)/2.0
      t4 = -B3*FI(I)*HLdX*DXL(IELNO)/2.0
      t5 = D1*HLdX*DFIDS(I)
      t= t3 + t4 + t5
C
C        WRITE(6,10)t3,t4,t5,t
10      FORMAT(2X,6(2X,F15.10))
15      RETURN
     END
C
C ******* CALCULATION OF STIFFNESS MATRIX [FR]*****
C
     SUBROUTINE FRK(IELNO,I,AR,QR,BR,HR,FI,DFIDS,QQR,t,KLP,HRdX)
C
      IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION FI(2),DFIDS(2)
C
     COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
     COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
     COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
     COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
     COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
     COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
     COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
     COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
     COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
    +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
    +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
    +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C       WRITE(6,1)DXR(IELNO)
1       FORMAT(2X,'I AM IN SUBROUTINE FRK',F10.3)
C
      IF((AR.LE.0.0D+00).OR.(QR.EQ.0.0D+00))THEN
C        IF(AR.LE.0.0D+00)THEN
        t=0.0D+00
        GO TO  15
        ENDIF
C
       UR=QR/AR
       ZR=DSQRT(1.0 + Z4**2)
       A1=BR + HR*ZR
C
       IF(KLP.EQ.0)THEN
       R=AR/A1
       B1= R/CN3(IELNO)
       CS= 5.75*DLOG10(B1) + 6.2D+00
      D1=CS**2*GRAV*AR**3/(2.0*A1*DABS(QR))
      B3= ZR*QR/(2.0*A1)
      ELSE
     D1=AR**3.333/(2.0*CN3(IELNO)**2*A1**1.333*ABS(QR))
```

```
       B3= 2.0*ZR*QR/(3.0*A1)
       ENDIF
       t3 = -FI(I)*QR*DXR(IELNO)/2.0
       t4 = -B3*FI(I)*HRdX*DXR(IELNO)/2.0
       t5 = D1*HRdX*DFIDS(I)
       t = t3 + t4 + t5
C        WRITE(6,10)t3,t4,t5,t
10     FORMAT(2X,6(2X,F15.10))
15      RETURN
      END
C
C********* SUBROUTINE SHAPE ******************************
C
C       THIS PROGRAM DEFINES THE TYPE OF FUNCTION   ONG THE
C       ELEMENT.
C
      SUBROUTINE SHAPE(K,S,FI,DFIDS)
       IMPLICIT REAL *8(A-H,O-Z)
      DIMENSION FI(2),DFIDS(2),S(3)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200) PAR(200,4),PHI(200)
      COMMON QRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DX[illegible]TEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C******* LINEAR FUNCTION ELEMENT *************
C
C        WRITE(6,1)
C1       FORMAT(2X,'I AM IN SUBROUTINE SHAPE')
C
      FI(1) = 0.5 * (1.0 - S(K))
      FI(2) = 0.5 * (1.0 + S(K))
      DFIDS(1) = -0.5D+00
      DFIDS(2) = 0.5D+00
      RETURN
      END
C
C ******** SUBROUTINE INPUT *******
C
       SUBROUTINE INPUT(NSTEP,NITER,TOL,MTD,K,KUW,KLP,KFL,[illegible],DT,NGP,NQc,
     + IY1,IY2,IY3,IY4,IY5,IY6,IY7,IY8,IY9,IY10,IY11,IY12,IY13,IY14,
     +IY15,IY16,IY17,IY18,IY19,IY20,IY21,IY22,IY23,IY24,IY25,IY26,
     + IY27,IY28,JQF,ITAO,TM,DST,TAG1,TAG2,PET)
C
       IMPLICIT REAL *8(A-H,O-Z)
C
      COMMON THETA,CN1(200),CN2(200),CN3(200),OMEGA,GRAV,Qold(200)
      COMMON IBC(8),NBC,ALM(200),ELVRP(200),GSL(200,200),GSR(200,200)
      COMMON NELEM,NNODES,NELTYP(200),XL(200),GJC(200,200),Aold(200)
      COMMON NODNUM(200,2),ELVMc(200),ELVLP(200),PAR(200,4),PHI(200)
```

```
      COMMONQRM(200),QLM(200),APHI(200),QPHI(200),ARM(200),TETA,FC(200)
      COMMON Acnew(200),Qcnew(200),Ucnew(200),Hcnew(200),HLnew(200)
      COMMON ALnew(200),ARnew(200),QLnew(200),QRnew(200),HRnew(200)
      COMMON QfL(200),QfR(200),AMTR(200),AMTL(200),TAL(200),TAR(200)
      COMMON DHL(200),DHR(200),RHO,Z1,Z2,Z3,Z4,Hold(200),COEFF,ITAA,
     +Qt(200),QtF(200),VXL(200),VXR(200),CML(200),CMR(200),CF1,CF2,
     +PARF(200,2),PARL(200,2),DXL(200),DXR(200),DXM(200),HLSTEP(200),
     +HRSTEP(200),CASEL(200),CASER(200),WALL
C
C        WRITE(6,1)
C1       FORMAT(2X,'I AM IN SUBROUTINE INPUT')
C
      READ(5,*) MTD,K,KUW,KLP,KFL,DT,NGP,TETA,NQc,TM,DST,PET
      READ(5,*)Z1,Z2,Z3,Z4,ITAA,CF1,CF2
      READ(5,*)WALL,TAG1,TAG2
      READ(5,*) NSTEP,NITER,OMEGA,Cr,THETA,TOL,COEFF,JQF,ITAO
      READ(5,*) IY1,IY2,IY3,IY4,IY5,IY6,IY7,IY8,IY9,IY10,IY11,IY12,
     + IY13,IY14,IY15,IY16,IY17,IY18,IY19,IY20,IY21,IY22,IY23,IY24,IY25,
     + IY26,IY27,IY28
C     GENERAL INFORMATION
C-------------------------
      READ(5,*) NNODES,NELEM,NBC
      WRITE(6,710)NNODES,NELEM
C        WRITE(6,720)
      DO 20 I=1,NNODES
      READ(5,*)(PAR(I,J),J=1,2),ELVMc(I),APHI(I),QPHI(I)
C
      WRITE(6,750)I,(PAR(I,J),J=1,2),ELVMc(I),APHI(I),QPHI(I)
 20   CONTINUE
C
      DO 25 I=1,NNODES
      READ(5,*)(PARF(I,J),J=1,2),ELVLP(I),ALM(I),QLM(I),HLSTEP(I)
C
       WRITE(6,750)I,(PARF(I,J),J=1,2),ELVLP(I),ALM(I),QLM(I),HLSTEP(I)
 25   CONTINUE
C
      DO 26 I=1,NNODES
      READ(5,*)(PARL(I,J),J=1,2),ELVRP(I),ARM(I),QRM(I),HRSTEP(I)

C
       WRITE(6,750)I,(PARL(I,J),J=1,2),ELVRP(I),ARM(I),QRM(I),HRSTEP(I)
 26   CONTINUE

C     SET UP CONNECTIVITY TABLE
C-----------------------------
C         WRITE(6,760)
       DO 50 I=1,NELEM
       READ(5,*)NELTYP(I),(NODNUM(I,J),J=1,(NELTYP(I))),CN1(I),
     + CN2(I),CN3(I)
       X1=PAR((NODNUM(I,1)),1)
       X2=PAR((NODNUM(I,(NELTYP(I)))),1)
      DXM(I)=DABS(X2-X1)
C
       X1=PARF((NODNUM(I,1)),1)
       X2=PARF((NODNUM(I,(NELTYP(I)))),1)
       DXL(I)=DABS(X2-X1)
C
       X1=PARL((NODNUM(I,1)),1)
```

```
      X2=PARL((NODNUM(I,(NELTYP(I))))),1)
      DXR(I)=DABS(X2-X1)
C
     WRITE(6,770) I,(NODNUM(I,J),J=1,2),DXM(I),DXL(I),DXR(I),CNI(I)
 50     CONTINUE
C
C       DO 60 I= 1,NELEM
C       DO 55 J= 1,NELTYP(I)
C      WRITE(6,780)I,APHI(NODNUM(I,1)),APHI(NODNUM(I,(NELTYP(I))))
C       WRITE(6,780)I,J,APHI(NODNUM(I,J))
55     CONTINUE
60     CONTINUE
C
      IF(KUW.EQ.0)THEN
      WRITE(6,790)
      ELSE
      WRITE(6,795)OMEGA
      ENDIF
C
      IF(ITAO.EQ.0)THEN
      WRITE(6,800)
      ELSE
      WRITE(6,810)
      ENDIF
C
      IF(JQF.EQ.0)THEN
      WRITE(6,820)
      ELSE
      WRITE(6,830)
      ENDIF
 710 FORMAT(/6X,'SOLUTION FOR A ONE - DIMENSIONAL PROBLEM (COMPOUND CH
    +ANNEL FLOW)'//6X,'UNKNOWN PARAMETERS: Amc,Qmc,AL,AR AT EACH NODE'
    +//6X,'TOTAL NUMBER OF NODES =',I3, 'FOR ',I3,'ELEMENTS'/)
 720 FORMAT(//6X,'NODE',7X,'X',7X,'AC',5X,'QC',7X,'AL',6X,'QL',
    + 6X,'AR',6X,'QR'/6X, 62('-'))
 750 FORMAT(6X,I3,2X,13(2X,F8.3))
 760 FORMAT(///6X,'CONNECTIVITY TABLE'/6X,18('-')//6X,'ELEMENT',4X,'TYP
    1E',13X,'NODES',12X,'LENGTH'/6X,51('-'))
 770     FORMAT(4X,I3,3X,'LINEAR',2X,2I3,1X,F7.2,3(1X,F10.6))
 780    FORMAT(/,2X,2I3,2(2X,F6.1))
 790    FORMAT(/6X,'THE UPWINDING PARAMETER OMEGA IS EQUAL TO ZERO')
 795   FORMAT(/6X,'THE UPWINDING PARAMETER OMEGA IS EQUAL TO',F6.2)
 800    FORMAT(/6X,'THE MOMENTUM TRANSFER Mtf IS NOT ALLOWED TO
    + TAKE PLACE')
 810    FORMAT(/6X,'THE MOMENTUM TRANSFER Mtf IS ALLOWED TO TAKE PLACE')
 820    FORMAT(/6X,'FLOW EXCHANGE q BETWEEN MAIN CHANNEL AND FLOOD PLAIN
    + IS NOT ALLOWED')
 830    FORMAT(/6X,'FLOW EXCHANGE q BETWEEN MAIN CHANNEL AND FLOOD PLAIN
    + IS ALLOWED')
      RETURN
      END
```