THE RIVER LARK PROJECT
MODEL MANUAL

PREPARED FOR
THE NATIONAL RIVERS AUTHORITY
BY THE DEPARTMENT OF CIVIL ENGINEERING
UNIVERSITY OF BIRMINGHAM

APRIL 1991

NATIONAL RIVERS AUTHORITY:

RIVER LARK PROJECT

MODEL MANUAL

TABLE OF CONTENTS

FIGURES:

TABLES:

NATIONAL RIVERS AUTHORITY:

RIVER LARK PROJECT

MODEL MANUAL

1 Introduction

This manual is for use in simulating groundwater flow behaviour in the River Lark Chalk aquifer in Suffolk. The extent of the aquifer and the finite difference mesh representing it in the model, are superimposed in Fig.(1.1).

The model has been developed for the National Rivers Authority by the Department of Civil Engineering of the University of Birmingham. The model is suitable for use in management studies of the aquifer.

This manual aims to provide the reader with all of the information necessary for using the model to simulate field conditions. It is expected that the reader is familiar with FORTRAN77 and also with finite difference groundwater models as described in Rushton and Redshaw (1979). However, brief discussions of the assumptions and the pertinent theory used in developing the model are included where necessary.

2 Recharge Model

2.1 Overview of the Recharge Model

Penman (1949) proposed a method to determine potential evaporation which has been applied frequently to the problem of calculating natural recharge to an aquifer. In this theory, recharge is calculated as the remainder after the potential evaporation and direct runoff have been subtracted from precipitation. Penman's work has been augmented by further

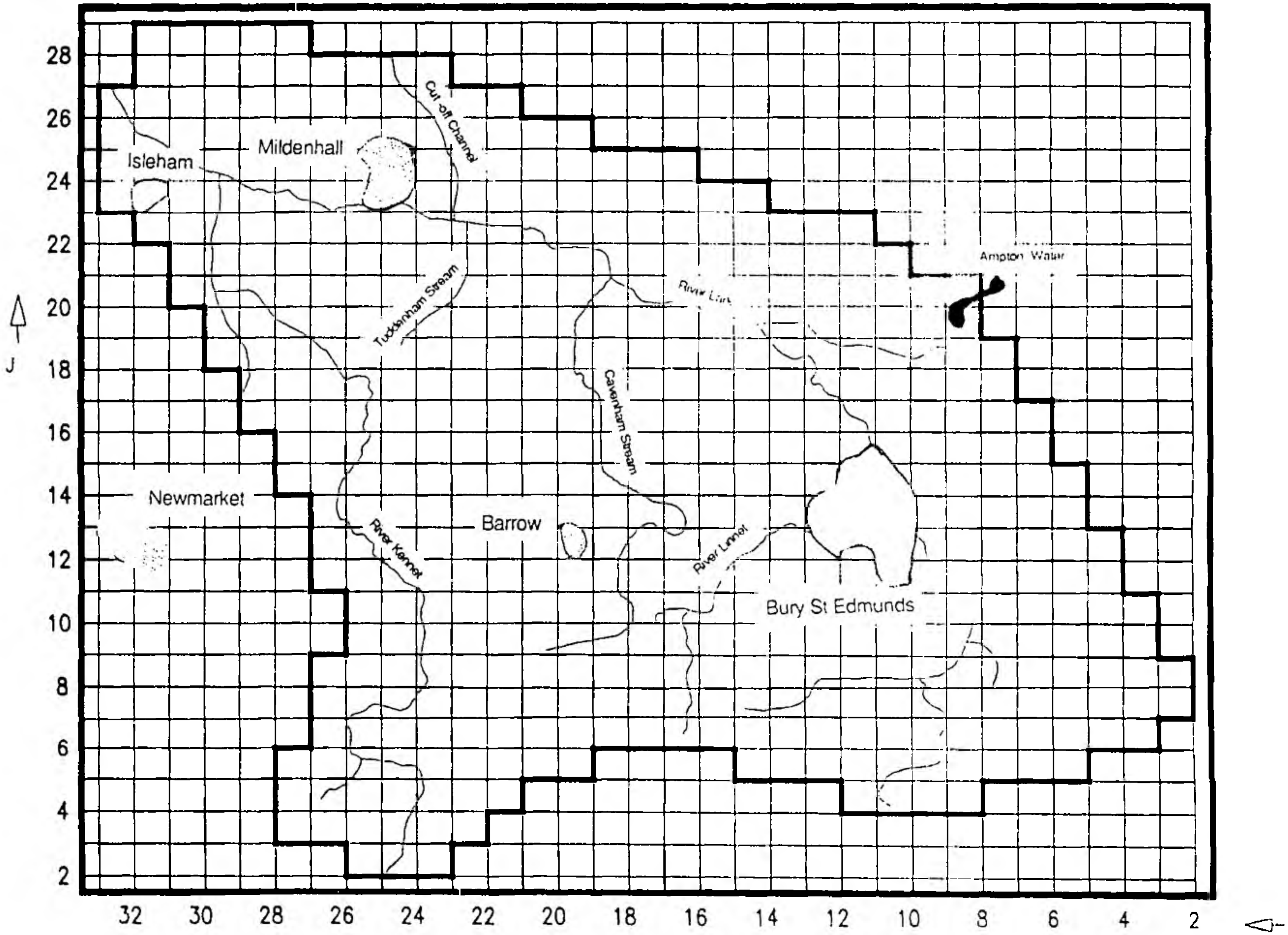

28
26
24
22
20
18
16
14
12
10
8
6
4
2
J
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
Isleham
Mildenhall
Cut-off Channel
Tuddenham Stream
Ampton Water
Cavenham Stream
Newmarket
Barrow
River Kennet
River Linnet
Bury St Edmunds

Fig 1.1

research, observation and specific application of his basic theory to a variety of different field problems. Grindley (1967) related vegetation and season to evapotranspiration and improved upon the method suggested by Penman. For particular situations, Rushton and Ward (1979) give a more accurate estimate of the recharge by allowing a fraction of the effective precipitation - i.e. the actual precipitation minus the evaporation - to enter the aquifer directly. The balance is determined using Penman-Grindley theory.

The proportion of recharge due to each node is determined by superimposing the area over which the recharge is uniformly distributed on the finite difference grid. The recharge is then entered as a total inflow which is distributed by the model to each of the affected nodes.

Main Program

The program, RECH, is used to determine the recharge for the River Lark Chalk aquifer from the potential evaporation, precipitation and soil moisture content data.

The model comprises a main program and 9 subroutines. The main program contains a routine to determine the recharge on a monthly basis from daily rainfall and evapotranspiration data. the monthly crop and root constants, also required for the model, have been determined from local data and remain constant for each year of recharge calculated.

Subroutines

Most of the subroutines are related to data preparation and the interactive operation of the program. The full list is as follows:

ACTDIR: ⎤ Routines to calculate recharge components
EFDIR: | from proportion of actual and effective
SURDIR: ⎦ precipitation that infiltrate directly.

COLOR: ⎤
CLEAR: | Routines related to
CURSOR: | the screen display.
NORMAL: ⎦

MOREC: Prepares potential evaporation data available from the MORECS system into the required format for entry into the recharge program.

RAIN: Converts rainfall data for the station at Bury St Edmunds into monthly values of daily average flows.

2.2 Operation of the Recharge Model

Updating Input Files

In order to update recharge records for the model, it is necessary to enter daily rainfall data into the file named "buryrain.dat" and the potential evaporation data into "morec.dat" for the entire period to be simulated. These data are entered according to the formats specified in Appendix A.

The first stage of running RECH is to prepare field data for the recharge calculation. This is achieved from the interactive questions in the subroutines MOREC and RAIN, which are addressed from the main program. It is important to note that the exact time period for the required recharge record to be used in the Lark model must be specified when using MOREC and RAIN to prepare the necessary input for RECH.

Executing the Recharge Model

The aquifer is divided for purposes of the RECH program into two zones, the Chalk and the Boulder Clay. Once the data has been prepared in the RAIN and MOREC routines, the program continues execution to calculate the recharge and runoff for both the Chalk and the Boulder Clay. The format of the output file "recharge.dat", which is also the input file for the Lark model is given in Table (B3) in Appendix B. The resulting data set is output for the entire period as monthly averages of daily flows, expressed in Ml/day.

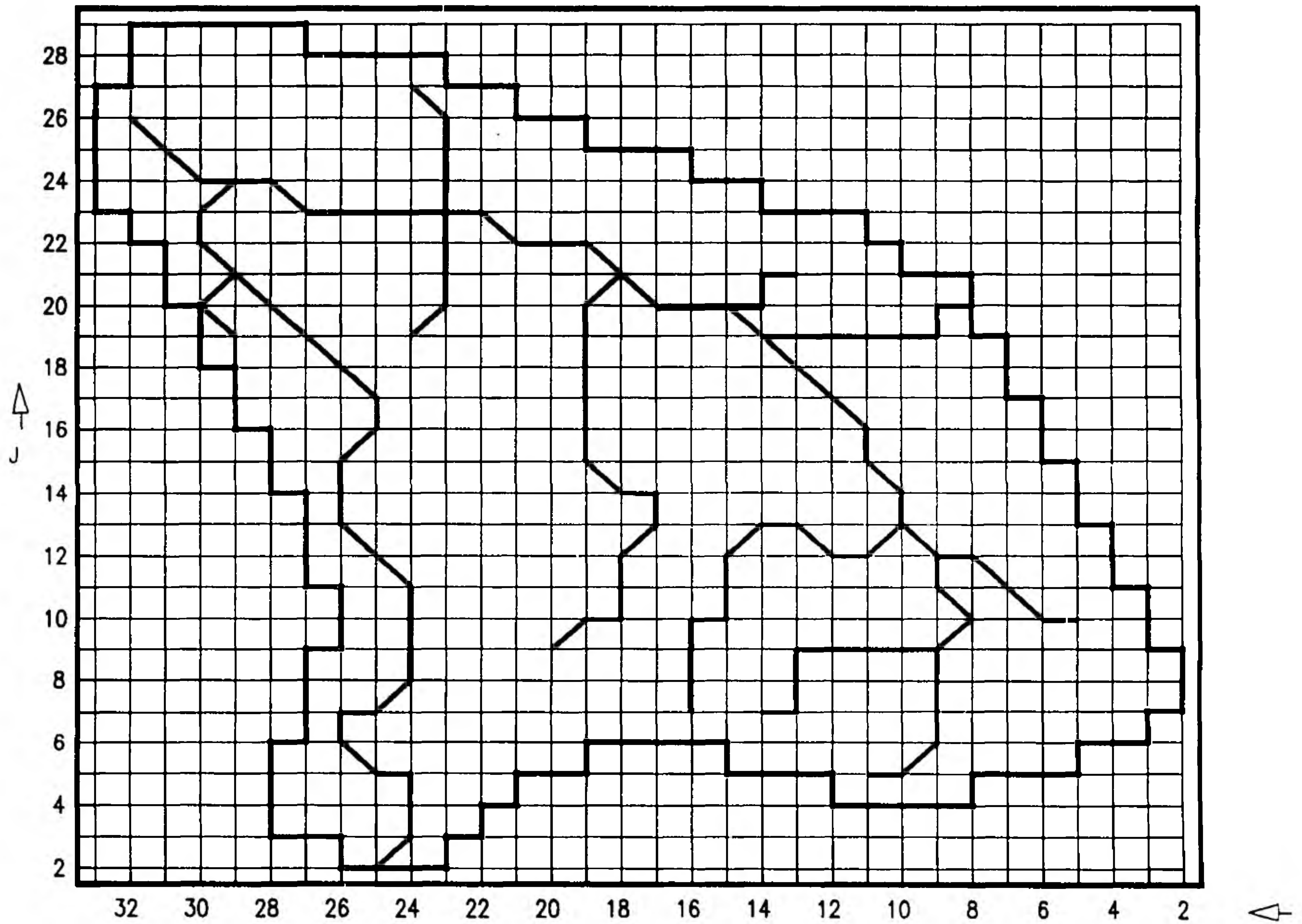
3 Regional Groundwater Flow Model

3.1 Overview of the Groundwater Model.

The model program is coded in FORTRAN77, and is approximately 1800 lines in length. It has a main segment with five associated subroutines. Three of these routines are concerned with the variable transmissivity aspect of the groundwater flow model and govern the revision of the transmissivities at each time step. There are also two routines that determine the monthly streamflows and baseflows along the watercourses defined by the selected nodes shown in Fig.(3.1). The remaining routines are concerned with presentation of the model output.

Description of Main Program

The main program controls progress of the simulation and



28
26
24
22
20
18
16
14
12
10
8
6
4
2
J
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
i

Fig 3.1

contains the following functions of the model:

> Entry and preparation of the various data sets for simulation.

> The central equations of the finite difference technique.

> The successive over relaxation (SOR) routine used for solving the simultaneous equations formulated in the finite difference method at each time step.

> The control of input and output to dump files, used in stopping and restarting the model at desired month of the 21 year record span.

> The output of simulated parameters for comparison with recorded field data, including streamflows, baseflows and groundwater heads for the end of each month of simulation.

Subroutines called from the main program prepare output and also perform tasks related to the variable transmissivity and streamflow simulation features of the model.

Description of Subroutines

There are three subroutines to the main program that are concerned mainly with the simulation of variable transmissivity. The remaining subroutines are used to output the results of the simulation either for printing or plotting.

> **MOSOUT:** This routine is called from the main program at the end of each month of simulation and writes the relevant data to the output files. The presentation of simulation output from the model is determined by the ultimate use of the results.

PRIN: Calls to this routine can be made at selected points in the main program, for printing out any of the major data sets. Either the full arrays, or segments of these arrays storing the following parameters may be specified for output at any stage of simulation:

Groundwater heads
Saturated depths
Permeabilities (in x or y directions)
Storage coefficients
Transmissivities (in x or y directions)
Base levels
Fixed heads

FLOWR & BAL: The leakage flows between the aquifer and the river are determined in FLOWR and are added to the effluent discharge and runoff data entered for each node at the start of the simulation. In BAL, a nodal flow balance is then carried out at each node in succession along the relevant watercourses to determine the net streamflows. The streamflows at the selected sites such as Temple Weir, Fornham St Martin and Isleham.

CORD: Generates nodal coordinates from given grid spacings

FACT & FD: Boundaries are defined in terms of the coefficients A,B,C and D which apply to the finite difference equations of the SOR routine.

TRAN: Calculates recharge and storage factors for the whole area at the start of simulation. In addition, this routine reads in the nodal storage coefficients, transmissivities and base levels.

CALZ: Calculation of the saturated depths across the aquifer.

PERMS: Permeability calculation, carried out only once at the start of simulation. The permeability is determined from the entered transmissivity at each node. This value is modified by two factors to produce a primary and secondary permeability used to establish transmissivity in TRAN.

FILE_OPEN: This routine opens all the necessary input and output files at the start of simulation.

RIVER_DATA: Routine for entry of river parameters, which include the nodal coordinates of the watercourses, leakage factors, river levels and runoff coefficients.

RECABSENT: Entry of recharge and abstraction data.

IRIGINDENT: Entry of irrigation and industrial users data.

CLEAR:
BLINK:
VIDEO_MODE:
CURSOR:
NORMAL:

Routines for controlling screen display during simulation.

Data Requirements

Aquifer Geometry

The geometry of the aquifer in the model is represented on a rectangular grid, with the coordinate convention as shown in Fig.(3.2). They define the horizontal extent of the aquifer and remain constant throughout simulation.

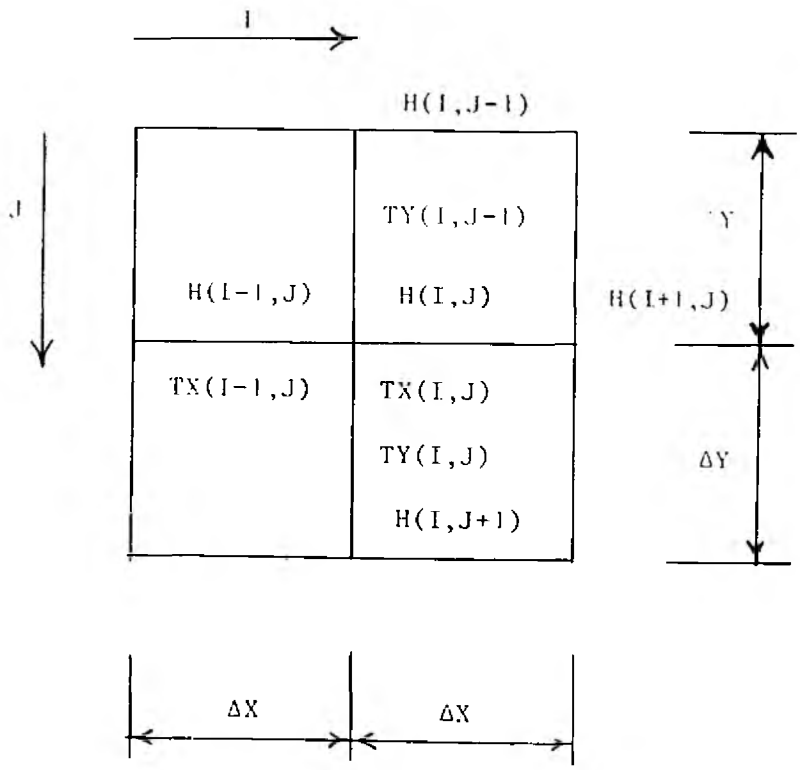


Fig 3.2

A lower limit of the aquifer in the vertical direction is defined by base levels of the aquifer entered for each node. These levels are used in determining saturated depths during simulation.

Aquifer properties

The model allows simulation of anisotropic, heterogeneous conditions. The physical parameters of the aquifer medium required for each node, are transmissivities in both x and y directions and the storage coefficients.

Permeability is assumed to be vary with depth according to the simple nonlinear relationship shown in Fig.(3.3) which is based on a primary and a secondary permeability.

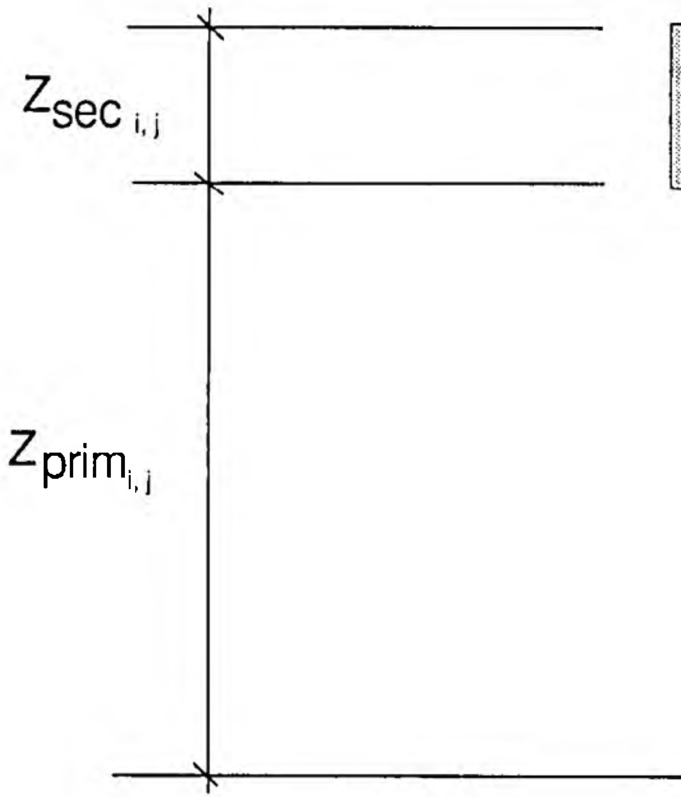
Storage coefficients are assumed to remain constant with saturated depth.

Monthly Flow Data

The categories of flow data relevant to simulating the regional groundwater movement in the Lark Chalk are

> Recharge
> Public water supply abstraction
> Industrial abstraction
> Irrigation abstraction

In addition, the flowslisted below are also entered for streamflow calculation. However, these flows do not influence groundwater simulation as they are only included in a post-processing routine which uses model output to calculate



$z_{sec\ i,j}$
$z_{prim_{i,j}}$

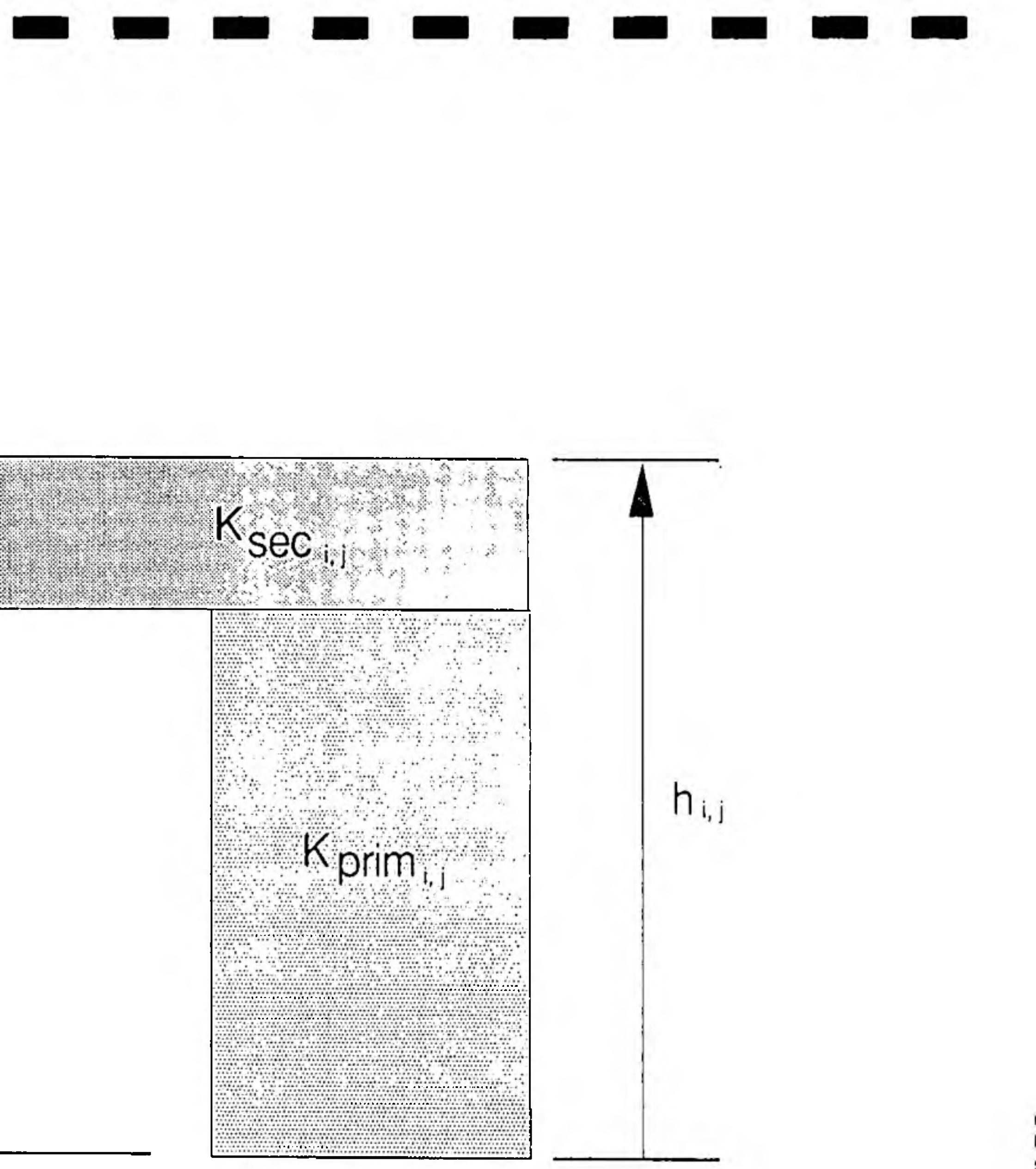


Fig 3.3

streamflow:

> Effluent
> Boulder Clay runoff
> Chalk runoff

Abstraction data are included in the model for a pumped borehole that is located at the nearest nodal point to the field position of the well.

All of the above categories of flow data are entered as flow rates that remain constant over each calendar month. By treating the flow data in this way, it is possible to divide time steps into periods with durations of less than a month. Because of the relatively slow response of an aquifer, this is an acceptable approximation.

Boundary Conditions

Three possible boundary conditions can be applied in finite difference groundwater modelling: fixed head boundaries, known inflow boundaries and impermeable boundaries.

Fixed head boundaries are encountered when the aquifer is in direct contact with a large source of water. They are modelled by setting the groundwater head at the affected nodes to a value which remains constant throughout the simulation. In the Lark model, fixed heads are applied on the boundary to three nodes [(33,24),(33,25) and (33,26)] at the downstream section where the River Lark leaves the area included in the model. All three nodes are fixed to a 0.00 mASL datum level throughout simulation.

Impermeable boundaries are defined by setting transmissivities immediately outside the boundary to zero and by reducing the plan area at the affected nodes accordingly. These conditions are applied to each node on the boundary to the area of the

Lark catchment considered in the model with the exception of the three fixed head nodes described previously.

Known inflow boundary conditions which are modelled by introducing the inflows as an equivalent recharge to the affected nodes, are not required in the Lark model.

Data Files Used in the Model

In this section, the contents of each input file are described. The files are divided approximately according to the data types.

river.dat

Storage coefficients and transmissivity values;
aquifer base levels;
Grid spacings;
Nodal coordinates of aquifer boundary;
Fixed heads - locations and magnitude;
Time step lengths;
Over-relaxtion factor;
Error criteria;
Limits to number of iterations per time step.

riverst.dat

Leakage coefficients and river stages for nodes along the watercourse of the River Lark and its tributaries.

lark1.dmp

The file name convention is optional for these unformatted dump files listing contents of all variable registers at the end of a simulation for use in restarting the model. These files are controlled by the model and are therefore never edited by the user. They are in binary format and are thus inaccessible to most editing programs.

abst.dat

Abstraction data - nodal coordinates of each site; average monthly abstraction for individual wells.

recharge.dat

Average monthly runoff and recharge from the Chalk and from the Boulder Clay.

nodes.dat

Nodal recharge factors;
Nodal coordinates of the Boulder Clay and Chalk recharge areas that correspond to the separate flow blocks in "recharge.dat".

3.2 Start-stop facility

An important feature has been added to the Lark model to allow simulations to be carried out between any two months within the period January 1970 to December 1989.

This is achieved using unformatted data files into which are written the model arrays that contain all of the relevant data that describe the state of the aquifer at any given month end.

This facility allows the model to be started from any given time provided that a dump file has been created with the necessary starting conditions. To create such a file, the model is run from steady state to the selected date and then the relevant data is output to a dumpfile. This file can then be used to initialize the model to carry out any number of simulations from this date without having to repeat simulation from steady state.

Use of the start-stop option is demonstrated in section 3.5.

3.3 Initial Conditions

The flow records available for simulation, span the period 1970 to 1989. To determine suitable initial conditions for simulation starting in 1970, typical inflows and outflows based on historical data are entered into the model. A repetitive calculation is then carried out until a steady state solution is reached. For this calculation, storage coefficients across the aquifer are all set to a very small value, in order to render the time component of the governing equation obsolete. This is then followed by a further set of repetitive calculations based on historical monthly recharge for 1969, in which storage coefficients are set to normal values and where no abstraction takes place. Simulation of historical flow behaviour commences from January 1970.

Because the task of calculating initial conditions is demanding in terms of processing time, provision is made for saving the initial conditions between simulations. The solution to the steady state calculation can be stored in a dump file for use in subsequent model trials. This same approach can be applied to store the conditions at any time to allow repeated trials of a particular year during operational investigations.

3.4 Variable Transmissivity

The transmissivities for any given time step are determined from saturated depths at end of the previous time step. The transmissivities in the unconfined regions of the Lark are based on an assumption that there is a primary and a secondary flow zone. The nature of the fissure system in the area suggests a sharp increase in transmissivity with water table elevation. To allow for the presence of a more permeable layer towards the top of the Chalk in the Lark area, the secondary zone in the model, which varies with the saturated depth, has a higher permeability than the primary zone below it.

Groundwater head can never fall below the base level of the primary zone. The relationship is defined in Fig.(3.3).

The secondary saturated depth at each node, $Z_s(I,J)$ is found from $Z_p(I,J)$, the entered primary saturated depth and the groundwater head:

$$Z_s(I,J) = H(I,J) - Z_p(I,J) \qquad (3.1)$$

where H(I,J) = groundwater head

The nodal permeability is found from dividing the entered transmissivity by the primary depth:

$$K(I,J) = TSS(I,J)/Z_p(I,J) \qquad (3.2)$$

where TSS(I,J) = steady state transmissivity at node entered at the start of simulation,
K(I,J) = nodal permeability

The nodal permeability is then factored to determine the primary and secondary permeabilities:

$$K_s(I,J) = (2-\alpha)K(I,J) \qquad (3.3)$$

$$K_p(I,J) = \alpha K(I,J) \qquad (3.4)$$

where $\alpha < 1.0$.

This relationship ensures that $K_p(I,J) < K_s(I,J)$ to accommodate the nonlinear variation of transmissivity with depth in weathered Chalk. For the Lark model $\alpha = 0.90$ and applies throughout the model. However, as understandings of the aquifer system improve, it may be appropriate to alter this parameter at different nodes to accommodate regional variations in the relationship between transmissivity and depth.

The saturated depths are used to calculate the nodal transmissivity from the following relationship based on conditions at the end of the previous time step:

$$T(I,J,N) = Z_p(I,J,N-1)K_p(I,J) + Z_s(I,J,N-1)K_s(I,J) \quad (3.5)$$

where N = time step number,

T(I,J) = nodal transmissivity considered to be the same in the x and y directions in the Lark model. However, the model can be altered to allow for anisotropic conditions if there is justification for this.

3.5 Streamflow Simulation

Outflows are calculated by means of a head-dependent relationship at selected nodes in the model that correspond to the river node sites. The extent of the river system as represented on the model grid, is shown in Fig.(3.1).

The streamflow routine in the Lark model combines the runoff which is determined in the recharge model in a flow balance with the river-aquifer leakage flow calculated at each time step. The runoff component is balanced at the end of each time step and does not influence local groundwater heads, i.e. the river flow balance calculation is a post-processing routine independent of the simulation of groundwater movement. In other words the only natural outflow that affects the aquifer in the model, is the baseflow component of the streamflow.

The leakage flow is calculated from the following relationship during solution of the finite difference equations:

$$QRA(I,J) = R(I,J)[H(I,J,N)-G(I,J)] \qquad (3.6)$$

where

QRA(I,J) = the flow between river and aquifer at node (I,J)
R(I,J) = leakage factor
H(I,J,N) = groundwater head at time step N
G(I,J) = river stage at river node (I,J)

The routine using the above relationship to determine the river-aquifer leakage flows is capable of simulating sections of the Lark that are are both influent and effluent along their course. The model can be used to calculate the baseflow at given sites on the River Lark, i.e. it is the sum of the effluent leakage flows along a specified reach of the river. Both the baseflow and the full streamflow can be produced as output from a simulation run.

The full streamflow is calculated from the following algorithm:

$$R(I,R+1) = R(I,R) + QNODE(I,R+1) + QRA(I,R+1) \qquad (3.7)$$

where

R(I,R+1) = river flow for the current river node, R+1

R(I,R) = river flow for the preceding river node, R

QNODE(I,R+1) = the runoff component for the current river node determined from the recharge model and entered at the start of simulation

QRA(I,R+1) = the river-aquifer leakage flow for the current node

If R(I,R) < 0, only the entered runoff for the current river node constitutes the river flow before the water balance takes place. In the flow balance, the amount of water gained by the aquifer is controlled by the amount available in the river. The river is considered to be dry when the leakage into the aquifer is greater than or equal to the river flow.

Flow in the cut-off channel is accounted for by the diversion of a constant proportion of about 38% of the River Lark streamflow simulated at a node corresponding to a site adjacent to Barrow Mills immediately upstream of Isleham.

3.6 Operation of the Groundwater Model

In this section, it is assumed that the user is familiar with the relevant background theory and that the model has been successfully implemented on the user's own computer system. It is also assumed that the user is able to make limited modifications to the FORTRAN77 coding of the model, e.g. for the purposes of altering the format of output data.

Instructions on model operation are based on an example in which PWS abstractions are increased to their full licence limit and examined for their impact on the selected criteria of the streamflow at Temple Weir and the groundwater head variation at Barrow, Ingham and Herringswell over the period July 1975 to December 1976. The example is run using the dump file "larkex.dmp" created from a previous simulation to provide the initial conditions for July 1975.

How is the model used to carry out a simulation trial?

The essential parameters that affect a simulation trial are all contained in the file "grdwo.r" shown in Appendix B. The parameters shown in Table (B1) would result in a simulation of the historical behaviour of the Lark Chalk aquifer for the period January 1970 to December 1989. It must be remembered that the data for 1969 contained in the model are not the historical values and are only used to produce suitable conditions for simulating the state of the aquifer at the beginning of 1970.

The example demonstrated in this section requires that the simulated historical conditions for the period July 1975 to December 1976 are compared to those that would have resulted from a policy in which the abstractions at the 10 public water supply sites were all increased to the licence limits, which expressed in Ml/day are:

Beck Row	7.471
Eriswell (1&2)	7.721
Twelve Acre Wood	1.868
Isleham	3.112
Moulton	5.603
Tuddenham	0.748
Risby	4.090
Barrow Heath	9.090
Bury St Edmunds	6.849
Rushbrooke	4.910

To achieve this, it is necessary to carry out three simulations: one to produce a dump file containing the simulated historical conditions for the end of June 1975, one to simulate the historical behaviour for the period July 1975 to December 1976 and one to simulate the impact the increases above the historical PWS abstraction for the period July 1975 to December 1976.

To produce a dump file, "larkex.dmp" containing the data to represent aquifer state at the end of June 1975, the file "grdwo.r" shown in Appendix B has to b-

(Note: The bold characters in upper case are only shown for information concerning the data types and any editing of the data necessary for the example is shown underlined and bold).

```
1969 1975   1   6 YEAR START,YEAR END,MONTH START,MONTH END
    0   1       READ FROM DUMP FILE? WRITE TO DUMP FILE? (1=YES)
null            INPUT DUMP FILE NAME
larkex.dmp      OUTPUT DUMP FILE NAME
river.dat       MODEL GEOMETRY AND MATERIAL PARAMETER FILE NAME
out.dat         OUTPUT FILE NAME
riverst.dat     RIVER PARAMETER INPUT FILE NAME
bcflow.dat      OUTPUT OF FLOWS
nodes.dat       INPUT PARAMETER DATA FILE NAME
abst.dat        PWS ABSTRACTION INPUT
recharge.dat    RECHARGE INPUT
indust.dat      INDUSTRIAL AND IRRIGATION ABSTRACTION INPUT
1969 1989       FIRST AND LAST YEARS OF RECORDED FLOW DATA
```

The model program is executed and will run until June 1975 after which time the relevant data for representing the aquifer state for purposes of restarting the program are output to "larkex.dmp". The file "grdwo.r" is then edited to carry out the simulation of historical behaviour for the period July 1975 to December 1976:

```
1975 1976    7  12 YEAR START,YEAR END,MONTH START,MONTH END
    1    0     READ FROM DUMP FILE? WRITE TO DUMP FILE? (1=YES)
larkex.dmp     INPUT DUMP FILE NAME
null           OUTPUT DUMP FILE NAME
river.dat      MODEL GEOMETRY AND MATERIAL PARAMETER FILE NAME
out.dat        OUTPUT FILE NAME
riverst.dat    RIVER PARAMETER INPUT FILE NAME
bcflow.dat     OUTPUT OF FLOWS
nodes.dat      INPUT PARAMETER DATA FILE NAME
abst.dat       PWS ABSTRACTION INPUT
recharge.dat   RECHARGE INPUT
indust.dat     INDUSTRIAL AND IRRIGATION ABSTRACTION INPUT
1969 1989      FIRST AND LAST YEARS OF RECORDED FLOW DATA
```

The final run contains the modified PWS abstraction data in file "abstlic.dat", which was created by editing the historical abstraction data file, "abst.dat". This file was altered by increasing the PWS abstractions for the period July 1975 to December 1976 to the licence limits. Thus the file "grwdo.r" for the final trial differs from the second trial as follows:

```
1975 1976    7  12 YEAR START,YEAR END,MONTH START,MONTH END
    1    0     READ FROM DUMP FILE? WRITE TO DUMP FILE? (1=YES)
larkex.dmp     INPUT DUMP FILE NAME
null           OUTPUT DUMP FILE NAME
river.dat      MODEL GEOMETRY AND MATERIAL PARAMETER FILE NAME
out.dat        OUTPUT FILE NAME
riverst.dat    RIVER PARAMETER INPUT FILE NAME
bcflow.dat     OUTPUT OF FLOWS
nodes.dat      INPUT PARAMETER DATA FILE NAME
abstlic.dat    PWS ABSTRACTION INPUT
```

recharge.dat RECHARGE INPUT
indust.dat INDUSTRIAL AND IRRIGATION ABSTRACTION INPUT
1969 1989 FIRST AND LAST YEARS OF RECORDED FLOW DATA

The required data, i.e. the streamflow at Temple Weir and the groundwater head variations at Barrow, Ingham and Herringswell are saved following each trial simulation of the period July 1975 to December 1976. These data are contained in the following files:

Feltons, Barrow	"head.dat"	(column 3)
Rushbrooke	"head1.dat"	(column 3)
Temple Weir Baseflow	"temple.dat"	(column 2)
Isleham Baseflow	"isleham.dat"	(column 2)

Full details of the output files that contain simulated streamflows and groundwater head variations are given in sections 8.1 and 8.2.

The results of the above example are plotted in Fig.(3.4).

<u>What sort of simulations can be run using the model?</u>

Data sets prepared for use in simulating hydrogeological conditions in the Lark Chalk are available for the period 1970 to 1989. Simulations spanning the entire record or any part of it may be carried out with the model.

Alterations are possible to any aspect of the recharge record or water supply system in order to test specific management policies.

By altering the recharge record it is possible to reproduce the effects of a drought on the aquifer. Drought periods may be inserted into any part of the historical data.

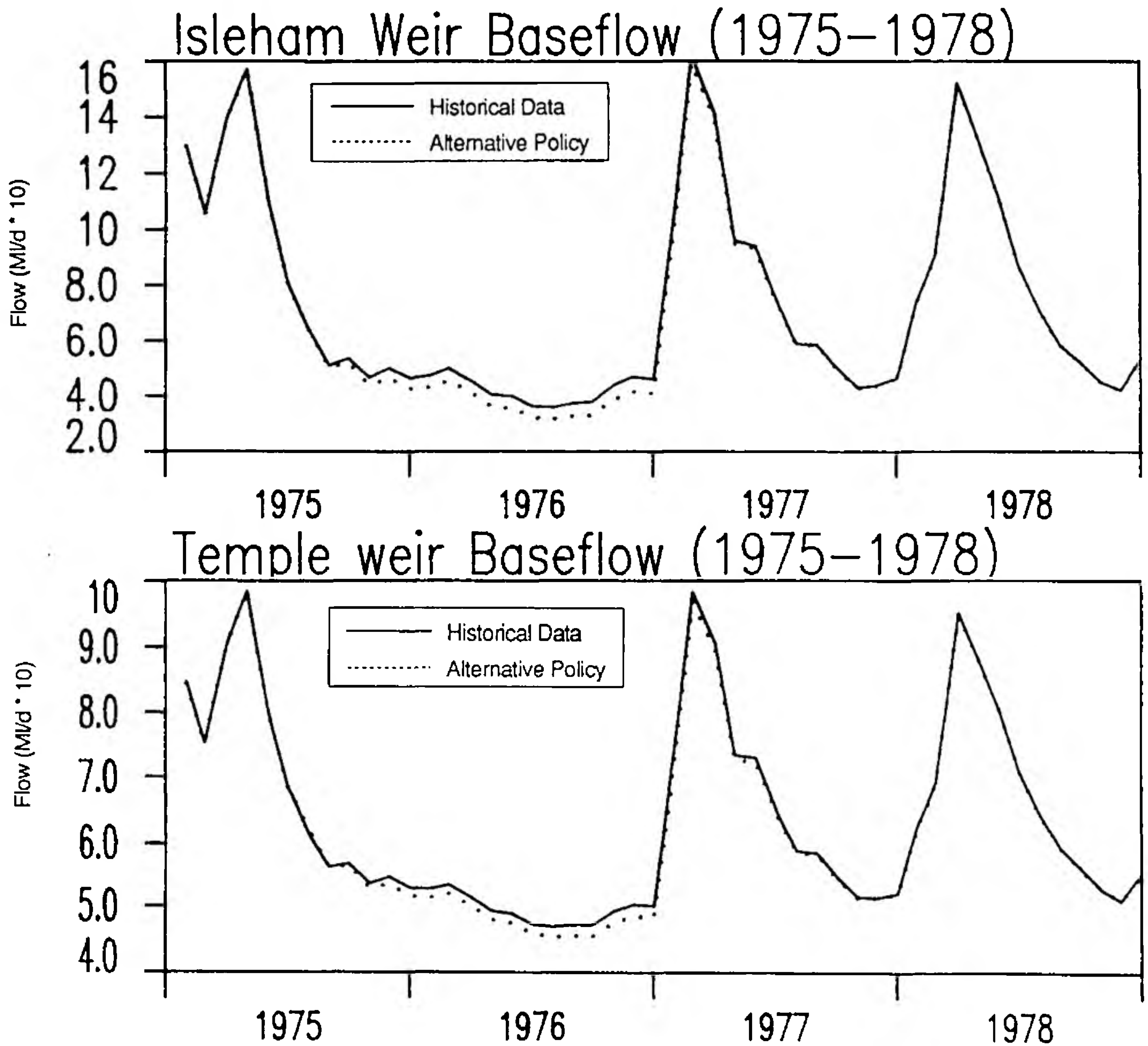

Isleham Weir Baseflow (1975-1978)
Historical Data
Alternative Policy
Flow (Ml/d * 10)
16
14
12
10
8.0
6.0
4.0
2.0
1975
1976
1977
1978
Temple weir Baseflow (1975-1978)
Historical Data
Alternative Policy
Flow (Ml/d * 10)
10
9.0
8.0
7.0
6.0
5.0
4.0
1975
1976
1977
1978

Fig 3.4(a)

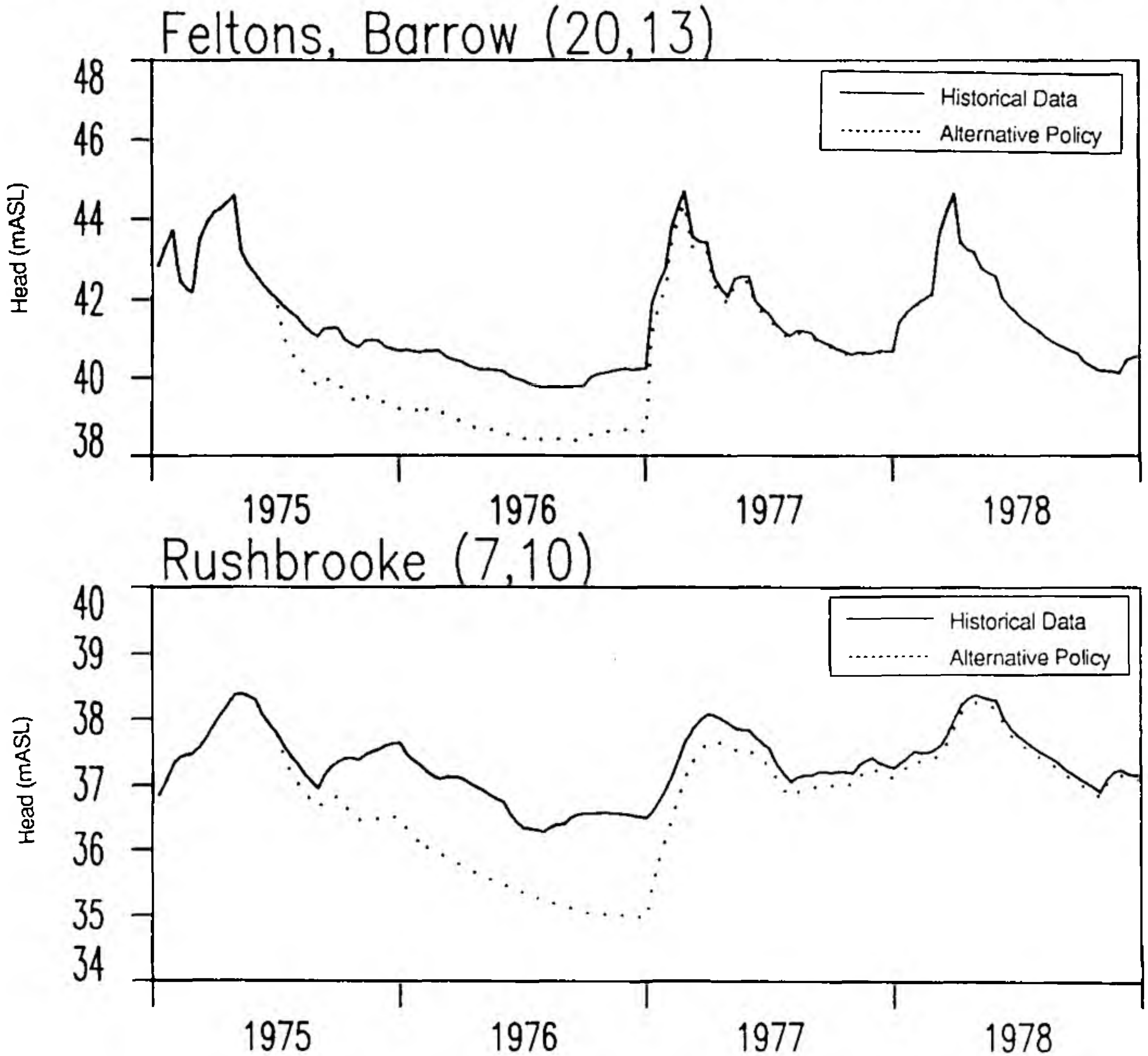

Feltons, Barrow (20,13)
Head (mASL)
48
46
44
42
40
38
Historical Data
Alternative Policy
1975
1976
1977
1978
Rushbrooke (7,10)
Head (mASL)
40
39
38
37
36
35
34
Historical Data
Alternative Policy
1975
1976
1977
1978

Fig 3.4(b)

In a similar manner it is possible to change abstraction details, either in the historical record or in data sets that include a synthetic recharge sequence, such as an extreme drought. The monthly quantities at individual abstraction sites may be modified, and new sites may be added in simulations that test the effect of increased development.

In taking advantage of the flexibility of the model, it is possible to experiment with a wide variety of alternative strategies based on the existing supply system. These may be aimed at increasing operational efficiency or at enhancing the total aquifer yield. It is also possible to use the model to indicate the potential of new sites for future development.

The model can also be used in developing control rules. If the rule is responsive and is based on the state of the resource during the progress of the simulation, it will be necessary to alter sections of the FORTRAN77 coding of the model program.

<u>How is the water supply system represented in the model?</u>

Data referring to the Lark Chalk aquifer water supply system for the duration of the simulation are contained in files "abst.dat" and "indust.dat". The pumping stations are placed at the nearest model grid node to their actual location. New pumping stations can be introduced into the model at any nodal point.

The PWS abstraction sites:

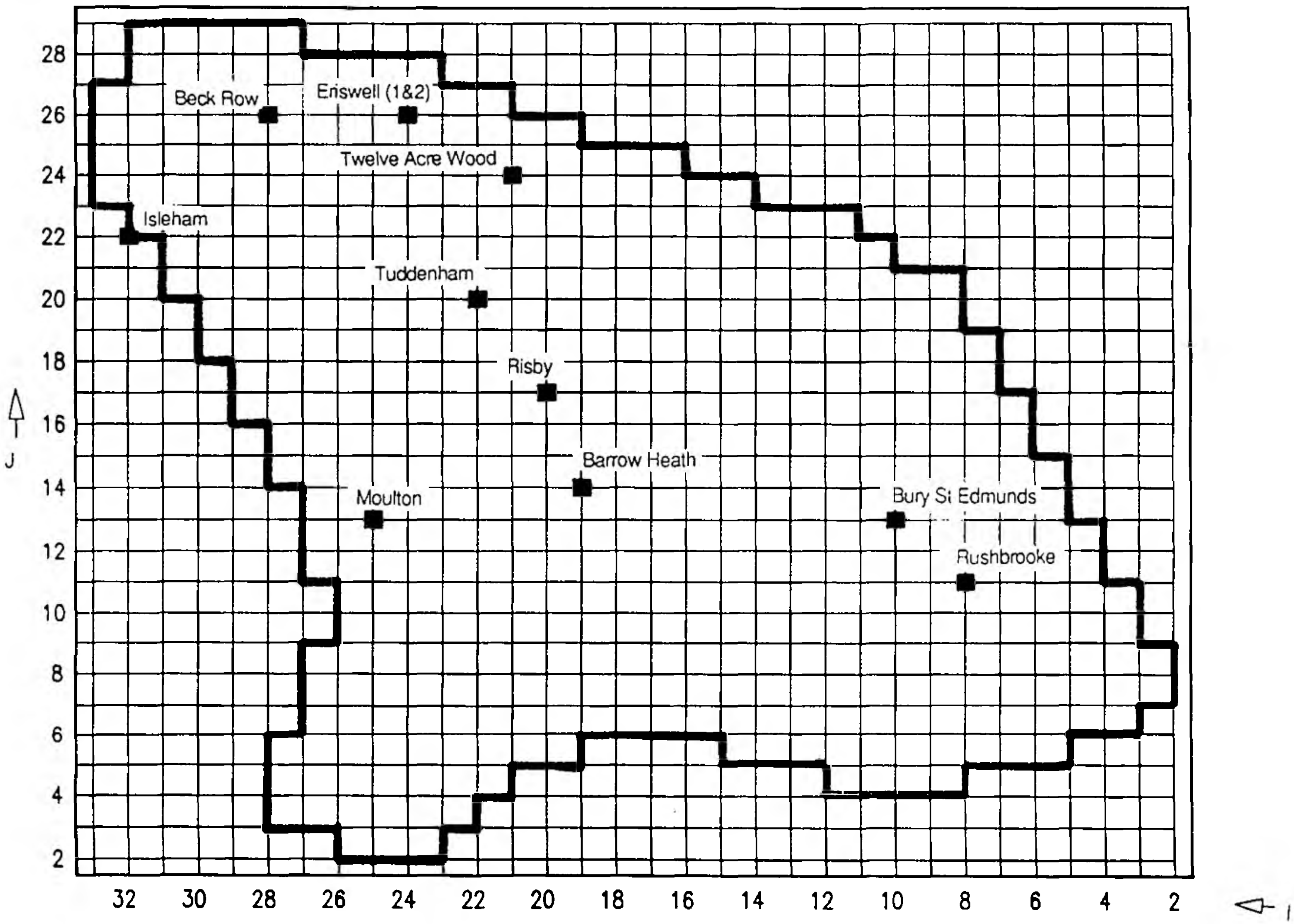
Name	Model Node	Grid Ref.
Beck Row	(28,26)	TL 680773
Eriswell (1&2)	(24,26)	TL 730759
Twelve Acre Wood	(21,24)	TL 748755
Isleham	(32,22)	TL 641729
Moulton	(25,13)	TL 700646
Tuddenham	(22,20)	TL 751710
Risby	(20,17)	TL 773681
Barrow Heath	(19,14)	TL 779654
Bury St Edmunds	(10,13)	TL 850642
Rushbrooke	(8,11)	TL 874624

The location of the above sites as represented on the finite-difference mesh, are shown in Fig.(3.5).

3.7 Updating Model Data Records

The data sets that require updating are summarised below in tabular form. Some updates are essential if the time span is to be extended, e.g. monthly recharge to the aquifer; whereas other updates are only possible if and when further development of the resource takes place, e.g. the transmissivity distribution may be refined as the result of new pumping test information.

Data type:	Source:	Data file/s:
Recharge	Output of RECH described in section 2.2	recharge.dat



Beck Row
Eriswell (1&2)
Twelve Acre Wood
Isleham
Tuddenham
Risby
Barrow Heath
Moulton
Bury St Edmunds
Rushbrooke
28
26
24
22
20
18
16
14
12
10
8
6
4
2
J
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2
I

Fig 3.5

Abstraction	Averaged monthly from NRA licence returns.	indust.dat, abst.dat
Chalk Base Levels, Storage Coefficient, Transmissivities	Field data collected from further development of the resource.	river.dat

4 Graphical presentation of model results

Many options are available for graphical presentation of simulation results. They vary according to the nature of the specific management study, as well as the availability of computer hardware and software. There are a variety of graphical routines which provide an efficient means of testing the performance of the model in representing field conditions, by comparing output against recorded data. The output of different simulations performed to compare management alternatives, are best assessed by comparison to simulated historical conditions.

4.1 Groundwater head

The groundwater head variation is simulated for 22 sites across the River Lark area within each model run. For purposes of model development, six sites were selected where a greater emphasis was placed on simulation of the response of the aquifer to historical flow data. The sites were chosen to provide a broad coverage across the aquifer, with preference given to those locations where hydrograph records of reasonable length are available.

Associated data files

head.dat:

Details of the model output contained in this file are:

Name	Model Node	Output File Column No.	NRA No.
Chevington	(17, 9)	1	TL 75/ 9
Front St, Ousden	(22, 9)	2	TL 75/19
Feltons, Barrow	(20,13)	3	TL 76/19
Higham Heath	(20,18)	4	TL 76/29
Rougham	(6,14)	5	TL 86/23
Great Barton	(8,14)	6	TL 86/22

head1.dat:

Details of the model output contained in this file are:

Name	Model Node	Output File Column No.	NRA No.
Bernersfield Fm.	(17,24)	1	TL 77/ 4
Cavenham Mill	(19,21)	2	TL 77/46
Rushbrooke Park	(7,10)	3	TL 86/ 5
Long Meadow	(9,18)	4	TL 86/169
Park Lane	(11,10)	5	TL 86/170
Ingham School	(11,20)	6	TL 86/10

head2.dat:

Details of the model output contained in this file are:

Name	Model Node	Output File Column No.	NRA No.
Butchers, Ashley	(26,11)	1	TL 66/ 4
Freckenham	(29,22)	2	TL 67/77
Park Fm, Ousden	(23, 9)	3	TL 75/68
Herringswell	(25,19)	4	TL 76/ 2
Higham, Gazely	(24,16)	5	TL 76/59
Tank Hall,Higham	(23,14)	6	TL 76/110

head3.dat:

Details of the model output contained in this file are:

Name	Model Node	Output File Column No.	NRA No.
Twelve Acrewood	(22,25)	1	TL 77/53
Gt. Horringer Hall	(13,12)	2	TL 86/ 1
Waterhall Junction	(29,16)	3	TL 66/88
nr. Wordwell Hall	(13,22)	4	TL 86/114

Examples of comparisons between observed data and model output are shown in Appendix E.

4.2 Streamflow and baseflows

As stated earlier, the streamflow routine in the Lark model combines the runoff which is determined in the recharge model in a flow balance with the river-aquifer leakage flow calculated at each time step. The only natural outflow that affects the aquifer in the model therefore, is the leakage flow component of the streamflow.

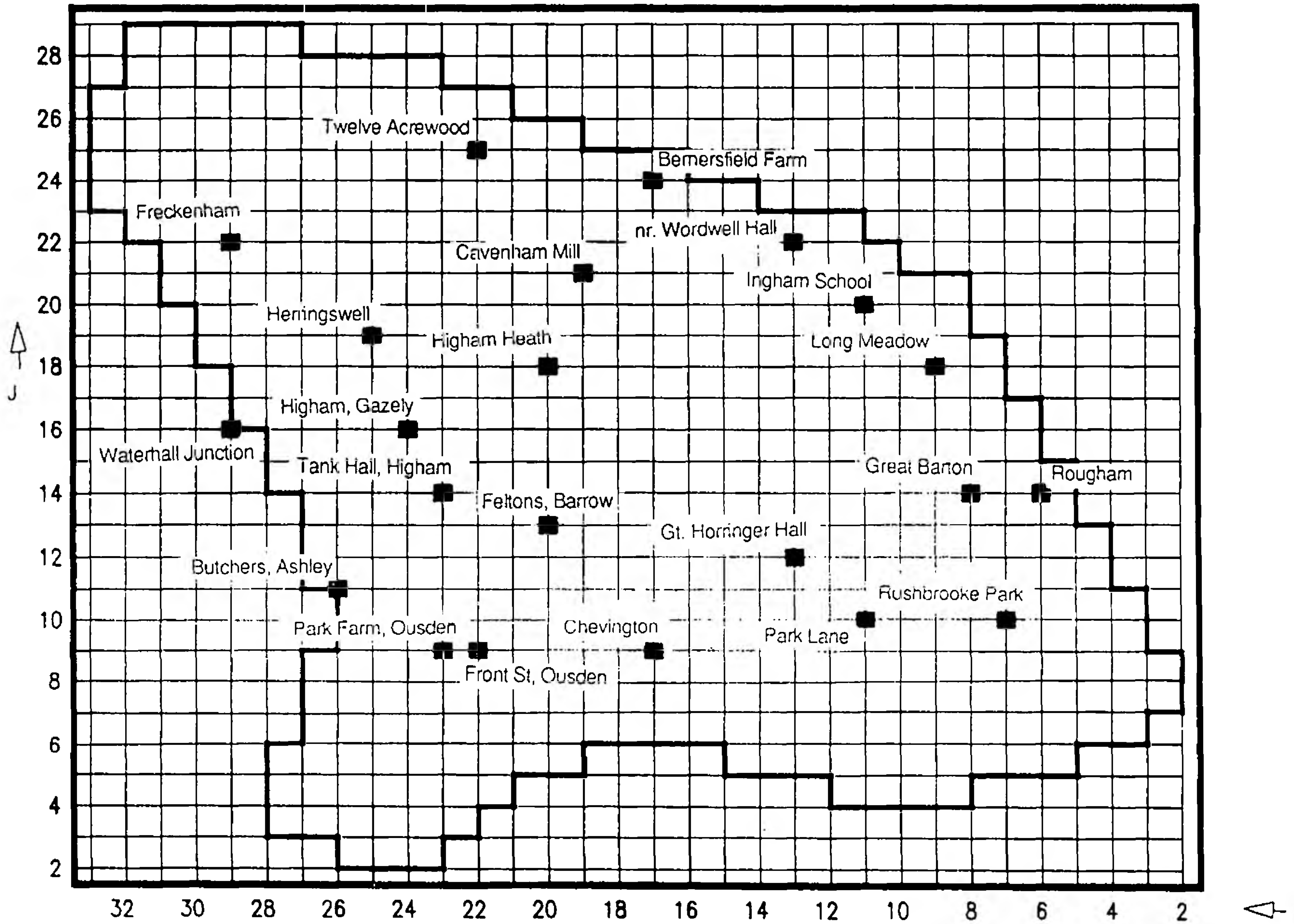

Twelve Acrewood
Bernersfield Farm
Freckenham
nr. Wordwell Hall
Cavenham Mill
Ingham School
Herringswell
Higham Heath
Long Meadow
Higham, Gazely
Waterhall Junction
Tank Hall, Higham
Great Barton
Rougham
Feltons, Barrow
Gt. Horringer Hall
Butchers, Ashley
Rushbrooke Park
Park Farm, Ousden
Chevington
Park Lane
Front St, Ousden
J
i
28
26
24
22
20
18
16
14
12
10
8
6
4
2
32
30
28
26
24
22
20
18
16
14
12
10
8
6
4
2

Fig 3.6

The leakage flow is calculated at each river node during solution of the finite difference equations. These nodal leakage flows can be used to determine the net river-aquifer leakage flows at specified sections of the River Lark, thus providing an indication of the local baseflow - i.e. where these net flows are effluent - at the given sites. Both the baseflow and the full streamflow are produced as output by the model at Beck Bridge, Temple Weir, Fornham St Martin and Isleham. The nodes used to represent these sites are:

Beck Bridge	(30,22)
Temple Weir	(20,22)
Fornham St Martin	(11,16)
Isleham	(31,25)

<u>Associated data files</u>

<u>temple.dat</u>

This file contains the simulated streamflow as well as the simulated baseflow data for Temple Weir and Beck Bridge.

<u>isleham.dat</u>

This output file contains the simulated streamflow data for Isleham and Fornham St Martin.

Typical output plots from both "temple.dat" and "isleham.dat" are shown in Appendix E.

Table of References.

Grindley, J. (1967). The estimate of soil moisture deficits. The Met. Mag., 96, no. 1137, pp. 97-108.

Penman, H.L. (1949). The dependence of transpiration on weather and soil conditions. J. Soil. Sci., 1, pp. 74-89.

Rushton, K.R. and Redshaw, S.C. (1979). Seepage and Groundwater Flow. Wiley, Chichester, England.

Rushton, K.R. and Ward, C. (1979). The estimation of groundwater recharge. J. Hydrol. (41). pp. 345-361.

Appendix A
Data Files - Recharge Model

There are two files that must be updated before the recharge file of the model, "recharge.dat", can be extended. These files are: "morec.dat" and "buryrain.dat". The format of these files is given in this appendix by listing a typical section of each file.

MOREC.DAT

3	1	89	1.6	1.6	2.4	0.0	1.4	1.4	3.9	0.1	0.1	8.1	5.9	8.3	149.
10	1	89	2.2	2.2	1.0	7.5	2.0	2.0	2.6	7.8	11.1	10.1	6.9	9.1	246.
17	1	89	4.3	4.3	0.3	7.1	3.9	3.9	1.6	7.1	12.0	18.3	6.3	8.0	310.
24	1	89	1.3	1.3	0.5	7.7	1.0	1.0	1.2	7.5	8.8	27.2	3.5	7.2	176.
31	1	89	2.3	2.3	0.7	2.6	2.1	2.1	1.3	2.7	4.7	16.0	4.5	7.5	205.
7	2	89	3.9	3.9	2.7	0.0	3.6	3.6	3.1	0.1	1.9	23.0	6.1	8.2	298.
14	2	89	4.6	4.6	2.9	0.0	4.3	4.3	3.1	0.0	4.4	23.6	5.5	7.6	309.
21	2	89	6.4	6.4	2.8	5.6	5.9	5.9	3.1	6.2	12.1	23.4	6.5	7.9	340.
28	2	89	5.5	5.5	1.6	15.2	5.0	5.0	1.5	15.2	21.9	26.4	3.7	6.6	355.
7	3	89	5.5	5.5	2.8	3.5	4.8	4.8	2.4	3.9	7.8	10.5	7.3	9.0	278.
14	3	89	9.3	9.3	0.0	1.3	7.9	7.9	0.0	3.1	13.4	17.1	7.3	8.1	332.
21	3	89	9.2	9.2	0.0	21.2	7.7	7.7	0.0	22.7	30.4	21.9	5.6	7.4	308.
28	3	89	12.2	12.1	9.3	0.0	11.1	11.0	8.2	0.0	2.8	33.8	8.8	8.7	380.
4	4	89	11.2	11.0	5.0	0.0	10.1	9.4	2.8	0.8	15.3	42.3	7.0	8.1	262.
11	4	89	10.7	10.7	0.0	6.5	9.5	9.5	0.0	9.9	22.2	27.2	6.5	8.0	278.
18	4	89	11.4	11.4	4.7	0.0	10.9	10.9	4.2	0.1	6.7	31.5	7.7	8.3	217.
25	4	89	9.3	9.3	1.1	9.2	8.6	8.6	1.1	10.4	22.1	12.0	5.4	7.7	251.
2	5	89	14.3	14.3	12.7	0.0	14.3	14.2	12.5	0.0	2.7	46.0	8.3	8.4	207.
9	5	89	22.8	22.6	35.3	0.0	23.0	21.0	33.5	0.0	0.0	81.6	12.5	10.1	156.
16	5	89	22.3	21.7	53.4	0.0	22.3	18.7	48.7	0.0	3.6	60.4	11.2	9.4	237.
23	5	89	30.5	24.8	78.2	0.0	31.4	21.3	69.9	0.0	0.0	84.5	16.2	12.3	201.
30	5	89	27.1	15.9	93.8	0.0	27.9	14.4	84.0	0.0	0.3	76.5	13.2	10.0	243.
6	6	89	16.9	13.2	83.7	0.0	18.8	14.7	75.6	0.2	23.3	40.7	9.2	8.5	175.
13	6	89	21.4	20.9	88.3	0.0	24.0	23.4	82.9	0.2	16.3	53.7	14.4	12.4	178.
20	6	89	30.1	16.8	105.1	0.0	33.9	18.1	101.0	0.0	0.0	92.9	18.1	13.2	167.
27	6	89	26.1	17.4	106.6	0.0	28.4	18.4	103.6	0.1	15.9	63.7	15.4	12.2	251.
4	7	89	23.7	21.6	109.0	0.0	24.5	20.6	105.2	0.2	19.2	61.3	14.4	12.1	250.

11	7	89	21.8	17.5	106.2	0.0	22.1	18.0	103.0	0.2	20.3	45.9	18.0	16.3	198.
18	7	89	22.9	7.9	114.1	0.0	20.8	6.9	110.0	0.0	0.0	57.5	16.6	13.4	204.
25	7	89	31.2	7.6	119.5	0.0	28.3	6.1	113.9	0.0	2.2	76.1	20.5	14.7	171.
1	8	89	26.7	11.2	121.3	0.0	23.2	10.4	115.0	0.1	9.4	54.4	17.2	12.7	251.
8	8	89	26.2	2.2	123.5	0.0	22.4	1.8	116.8	0.0	0.0	73.4	18.0	13.1	180.
15	8	89	27.0	17.1	124.3	0.0	23.3	15.2	115.7	0.1	16.3	44.3	18.2	14.0	301.
22	8	89	26.6	1.2	125.0	0.0	23.5	2.7	117.9	0.0	0.5	68.8	18.1	12.7	232.
29	8	89	18.5	10.2	121.4	0.0	17.3	9.7	114.0	0.1	13.5	41.9	15.3	12.6	260.
5	9	89	17.8	3.9	125.0	0.0	17.0	4.3	118.1	0.0	0.2	47.2	15.0	11.5	205.
12	9	89	13.8	1.6	125.0	0.0	13.1	1.8	118.5	0.0	1.3	23.6	17.0	14.4	241.
19	9	89	13.9	5.8	125.0	0.0	12.6	5.8	118.6	0.0	5.7	23.0	16.4	15.0	256.
26	9	89	15.1	0.6	125.0	0.0	14.3	0.7	118.8	0.0	0.4	43.0	16.3	13.2	203.
3	10	89	7.2	0.7	125.0	0.0	6.8	0.7	118.8	0.0	0.6	6.9	12.4	11.9	204.
10	10	89	10.1	7.0	122.6	0.0	9.1	6.9	116.4	0.1	9.3	22.5	11.2	10.6	264.
17	10	89	9.1	3.1	125.0	0.0	8.5	3.2	118.9	0.0	0.7	27.0	11.6	10.7	255.
24	10	89	10.2	7.8	119.5	0.0	9.4	8.2	114.0	0.1	13.3	29.2	13.5	12.5	321.
31	10	89	9.5	8.2	114.0	0.0	8.1	7.7	108.0	0.1	13.7	18.0	12.0	11.4	329.
7	11	89	4.5	4.1	101.5	0.0	3.8	3.7	95.4	0.1	16.6	24.6	7.2	9.1	192.
14	11	89	4.4	4.1	94.4	0.0	3.9	3.8	88.0	0.1	11.2	19.7	8.5	10.0	244.
21	11	89	3.6	3.2	97.4	0.0	3.3	3.2	91.0	0.0	0.2	25.6	7.7	8.9	213.
28	11	89	2.2	1.9	99.2	0.0	1.8	1.7	92.6	0.0	0.1	32.4	2.4	6.0	171.
5	12	89	0.9	0.9	99.0	0.0	0.6	0.6	92.1	0.0	1.1	23.5	2.1	6.5	130.
12	12	89	0.9	0.8	93.5	0.0	0.7	0.6	86.5	0.1	6.3	7.8	3.1	7.3	142.
19	12	89	3.9	3.9	32.7	0.0	3.3	3.3	28.4	3.3	64.7	2.7	7.1	9.0	294.
26	12	89	5.0	5.0	11.5	0.0	4.5	4.5	10.9	4.2	26.2	10.8	8.2	9.2	350.

BURYRAIN.DAT

```
1986
31
     103    19    -2   -46     8     0    73    40    43   -11
      -0    16     3    -0    19     0   -68   -26    19    32
      16    22     7     0     0     0    23    -0    52     1    -10 659
28
     -36    31     2    -0    16    11    -3   -11     0     0
      -0     0     0    -5    -0     0     3    -0    -0     0
      -0    -0    -0    -0    -0    -0    -0   -28           146
31
     -10     0    28    62     2     5    -0    -0     9     5
      -0     0     0    -0    -0     5    -0    70    17     2
     -12   -47    74    20     8    60   -39    -5   -17   -30     34 561
30
      37    26     1    -3   -25    20    41     3     0    -0
     -10   -13    28    28    64    47    51    -4   -59    30
      41    10     5    -0    -1   -71     3     3     0     0 624
31
       0    -7    -0    -0    27    50     5    31   -28    -0
      17     6     4    40    42    -0   -10     0     1    27
      10    32    -0    -0    -0    -0     0    -3    -0     0    -99 439
30
      -5     4    66    37     7   -12    -0     0     0    -0
      38     0    -0    -0     0     0     0     0     0    -0
      -0    53    18     0     0     0    -0    -0     0     0 240
31
       0    12     0  -154  -275     1     0    30     6     0
      -4   -45     0     0     0     0     0    -0    -2     4
       4     0     0    37   -30    -0     0    54     0    20      0 678
31
      -2    -0   223    13     0     0     0    -0    -0    89
       0     0     0     0     0     0     0    16     3     0
      34   -86    -0    -0   367    38     3    15    -0    -5      1 895
30
      -0    70    20     0    -0    -0     0     0     0     0
       0    -0  -170     0     9     0     0     0    -0    -0
       0     0     0     0     0    -0    -0     0     0     0 269
```

31

0	0	-0	-0	-0	0	0	0	0	-2	
-0	1	0	185	-0	-0	-9	-19	227	8	
151	7	0	-12	-8	3	61	26	-0	0	-100 819

30

-19	0	41	-0	2	-0	-62	-5	2	23	
8	0	46	-63	-0	5	58	68	1	93	
-5	-40	13	10	37	9	-0	-2	-1	0	613

31

-0	0	0	0	-70	-0	0	10	12	0	
12	-24	-1	2	36	1	63	14	-0	-0	
12	3	-1	-74	-23	-0	-0	0	94	24	-120 596

Appendix B

Data Files - Lark Chalk Model.

The following data files listed here to demonstrate the required formatting only contain the information necessary to simulate historical groundwater conditions in the Lark Chalk aquifer from 1970 to 1971. However, the full data sets of the model have been updated to allow simulations for the entire period of 1970 to 1989. Changes that are required for different simulations are described by example in Section 3.6.

Note: Any text shown bold and in uppercase in the tables that follow in this appendix is for information purposes only; it is not required in simulation and does not appear in the actual model data files.

Table B1 - grdwo.r
Table B2 - abst.dat
Table B3 - recharge.dat
Table B4 - indust.dat
Table B5 - nodes.dat
Table B6 - river.dat
Table B7 - riverst.dat

```
 1969 1989     1    12   YEAR START, YEAR END, MONTH START, MONTH END
    0    0               READ FROM DUMP FILE? WRITE TO DUMP FILE?  (1=YES)
null                     INPUT DUMP FILE NAME
null                     OUTPUT DUMP FILE NAME
river.dat                MODEL GEOMETRY AND MATERIAL PARAMETER FILE NAME
out.dat                  OUTPUT FILE NAME
riverst.dat              RIVER PARAMETER INPUT FILE NAME
bcflow.dat               OUTPUT OF FLOWS
nodes.dat                INPUT PARAMETER DATA FILE NAME
abst.dat                 PWS ABSTRACTION INPUT
recharge.dat             RECHARGE INPUT
indust.dat               INDUSTRIAL AND IRRIGATION ABSTRACTION INPUT
1969 1989 FIRST AND LAST YEARS OF RECORDED FLOW DATA
```

Table (B1): Input file "grdwo.r"

10 - NUMBER OF PWS SITES

28 26 - NODAL COORDINATES OF BECK ROW,ETC.

Beck Row

24 26

Eriswell(1&2)

21 24

Acrewood

32 22

Isleham

25 13

Moulton

22 20

Tuddenham

20 17

Risby

19 14

Barrow Heath

10 13

Bury St Edmunds

8 11

Rushbrooke

DETAILS OF 10 PWS SITES THAT EACH CORRESPOND TO A COLUMN OF FLOW DATA BELOW

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1969
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

0.000	0.000	0.000	0.000	0.000	0.000	1.968	0.000	6.065	2.645	
0.000	0.000	0.000	0.000	0.000	0.000	2.143	0.000	5.786	2.679	
0.000	0.000	0.000	0.000	0.000	0.000	2.194	0.000	5.774	2.645	
0.000	0.000	0.000	0.000	0.000	0.000	2.133	0.000	5.967	2.267	
0.000	0.000	0.000	0.000	0.000	0.000	2.419	0.000	6.677	2.645	
0.000	0.000	0.000	0.000	0.000	0.000	3.433	0.000	8.000	2.667	1970
0.000	0.000	0.000	0.000	0.000	0.000	2.742	0.000	6.645	2.484	
0.000	0.000	0.000	0.000	0.000	0.000	2.903	0.000	6.581	2.581	
0.000	0.000	0.000	0.000	0.000	0.000	2.400	0.000	6.433	2.533	
0.000	0.000	0.000	0.000	0.000	0.000	2.387	0.000	6.065	2.484	
0.000	0.000	0.000	0.000	0.000	0.000	2.433	0.000	5.867	2.367	
0.000	0.000	0.000	0.000	0.000	0.000	2.161	0.000	5.903	2.355	
7.097	4.065	0.000	1.258	0.839	0.206	2.097	0.000	6.194	2.387	
6.643	3.857	0.000	1.250	0.750	0.195	2.250	0.000	6.214	2.464	
6.097	4.032	0.000	1.177	0.677	0.205	2.419	0.000	6.226	2.548	
6.667	4.233	0.000	0.977	0.667	0.215	2.700	0.000	6.400	2.600	
7.194	4.710	0.000	0.968	0.710	0.228	2.806	0.000	7.000	2.677	
6.467	4.233	0.000	0.843	0.600	0.215	2.867	0.000	6.600	2.533	1971
6.742	5.419	0.000	1.458	0.710	0.276	3.323	0.000	7.355	2.548	
6.226	4.226	0.000	0.806	0.581	0.233	2.419	0.000	6.323	2.516	
7.033	5.300	0.000	0.843	0.667	0.250	2.967	0.000	6.867	2.567	
6.677	4.387	0.000	0.787	0.613	0.218	2.387	0.000	6.484	2.387	
6.900	4.300	0.000	0.757	0.533	0.203	2.100	0.000	6.333	2.400	
6.452	4.323	0.000	0.855	0.548	0.207	2.258	0.000	6.484	2.452	

Table (B2): Input file "abst.dat"

	RECHARGE		RUNOFF		
	B.C.	CHALK	B.C.	CHALK	
31	0.000	417.278	358.298	7.156	
28	0.000	372.101	338.912	7.104	
31	0.000	476.798	431.412	9.053	
30	0.000	129.181	137.625	4.392	
31	0.000	307.748	383.853	12.604	
30	0.000	36.389	36.975	6.853	
31	0.000	69.066	56.177	10.412	1969
31	0.000	38.581	49.863	9.241	
30	0.000	0.000	0.975	0.181	
31	0.000	0.000	4.282	0.794	
30	1.500	70.749	61.500	11.398	
31	2.177	118.285	96.895	9.416	
31	2.177	451.934	396.318	8.609	
28	0.000	549.101	467.931	9.308	
31	0.000	189.411	184.259	5.125	
30	0.000	400.655	389.825	10.147	
31	0.000	17.008	13.500	2.502	
30	0.000	1.372	8.625	1.598	1970
31	0.000	6.134	27.073	5.017	
31	0.000	15.447	21.774	4.035	
30	0.750	39.502	41.700	7.728	
31	1.452	15.049	22.863	4.237	
30	3.750	144.096	113.625	21.059	
31	2.903	245.920	283.139	7.156	

31	0.000	519.613	451.948	9.483	
28	0.000	59.393	62.943	2.040	
31	0.000	201.654	207.852	5.179	
30	0.000	24.744	21.675	4.017	
31	0.000	31.586	28.452	5.273	
30	0.000	161.286	218.125	12.454	1971
31	0.000	29.497	32.008	5.932	
31	0.726	53.989	55.306	10.250	
30	1.500	10.451	14.100	2.613	
31	0.000	47.150	39.992	7.412	
30	3.750	71.842	63.450	11.759	
31	0.000	12.066	16.258	3.013	

Table (B3) Recharge and runoff data in "recharge.dat".

```
1971   - FINAL YEAR OF FLOW RECORD TO BE ENTERED
   7   - NO. OF EFFLUENT FLOW SITES
  12   17 --┐
  27   23   |
  15   20   |
  17   13   ├─ NODAL POSITIONS OF EFFLUENT FLOW SITES.
   9   12   |  EACH CORRESPONDS TO A COLUMN OF
  16    8   |  EFFLUENT FLOW DATA BELOW.
  23   20 --┘
   7.23     1.92     3.67     0.46     0.34     0.24     0.62 - 1969 ┐
   7.49     1.96     3.67     0.46     0.34     0.24     0.62 - 1970 ├─ EFFLUENT
   7.76     2.00     3.67     0.46     0.34     0.24     0.62 - 1971 ┘  FLOW DATA
   2 - NO. OF INDUSTRIAL ABSTRACTION SITES
  10   15 ┐ INDUSTRIAL ABSTRACTION
  10   14 ┘ SITES
   0.00     0.00  - 1969 ┐
 504.00    57.70  - 1970 ├─ INDUSTRIAL ABSTRACTION
 307.00    60.30  - 1971 ┘
  16 - NO. OF IRRIGATION ABSTRACTION SITES
  14   14 ┐
  25   20 |
  13   12 |
  24   20 |
  26   22 |
  26   23 |
  21   15 |
  18   16 ├─ COORDINATES OF IRRIGATION
  24   17 |  ABSTRACTION SITES
   7   11 |
  16   15 |
  24   21 |
  10   16 |
  23   14 |
  17   17 |
  26   21 ┘
```

0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	- 1969
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	- 1970
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	- 1971

1.00 - OPTIONAL FACTOR FOR ADJUSTING ABSTRACTION

IRRIGATION ABSTRACTION

Table (B4) Input file "indust.dat"

2			NO. OF DIFFERENT RECHARGE ZONES
301			NO. OF BOULDER CLAY NODES
2	7	1.0	BOULDER CLAY NODAL COORDINATES, NODAL RECHARGE FRACTION.
2	8	1.0	
2	9	1.0	
3	6	1.0	
3	7	1.0	
3	8	1.0	
3	9	1.0	
3	10	1.0	
3	11	1.0	
4	6	1.0	
4	7	1.0	
4	8	1.0	
4	9	1.0	
4	10	1.0	
4	11	1.0	
4	12	1.0	
5	5	1.0	
5	6	1.0	
5	7	1.0	
5	8	1.0	
5	9	1.0	
5	10	1.0	
5	11	1.0	
5	12	1.0	
6	5	1.0	
6	6	1.0	
6	7	1.0	
6	8	1.0	
6	9	1.0	
6	10	1.0	
6	11	1.0	
6	12	1.0	
6	15	1.0	
6	16	1.0	
6	17	1.0	
7	5	1.0	

7	6	1.0
7	7	1.0
7	8	1.0
7	9	1.0
7	10	1.0
7	11	1.0
7	12	1.0
7	15	1.0
7	16	1.0
7	17	1.0
7	18	1.0
7	19	1.0
8	4	1.0
8	5	1.0
8	6	1.0
8	7	1.0
8	8	1.0
8	9	1.0
8	10	1.0
8	11	1.0
8	15	1.0
8	16	1.0
8	17	1.0
8	18	1.0
8	19	1.0
8	20	1.0
8	21	1.0
9	4	1.0
9	5	1.0
9	6	1.0
9	7	1.0
9	8	1.0
9	9	1.0
9	10	1.0
9	11	1.0
9	15	1.0
9	16	1.0
9	17	1.0

9	18	1.0
9	19	1.0
9	20	1.0
9	21	1.0
10	4	1.0
10	5	1.0
10	6	1.0
10	7	1.0
10	8	1.0
10	9	1.0
10	10	1.0
10	11	1.0
10	15	1.0
10	16	1.0
10	17	1.0
10	18	1.0
10	19	1.0
10	20	1.0
10	21	1.0
10	22	1.0
11	4	1.0
11	5	1.0
11	6	1.0
11	7	1.0
11	8	1.0
11	9	1.0
11	10	1.0
11	11	1.0
11	16	1.0
11	17	1.0
11	18	1.0
11	19	1.0
11	20	1.0
11	21	1.0
11	22	1.0
12	4	1.0
12	5	1.0
12	6	1.0

12	7	1.0
12	8	1.0
12	9	1.0
12	10	1.0
12	11	1.0
12	15	1.0
12	16	1.0
12	17	1.0
12	18	1.0
12	19	1.0
12	20	1.0
12	21	1.0
13	5	1.0
13	6	1.0
13	7	1.0
13	8	1.0
13	9	1.0
13	10	1.0
13	11	1.0
13	15	1.0
13	16	1.0
13	17	1.0
13	18	1.0
13	19	1.0
13	20	1.0
14	5	1.0
14	6	1.0
14	7	1.0
14	8	1.0
14	9	1.0
14	10	1.0
14	11	1.0
14	12	1.0
14	15	1.0
14	16	1.0
14	17	1.0
14	19	1.0
14	20	1.0

15	5	1.0
15	6	1.0
15	7	1.0
15	8	1.0
15	9	1.0
15	10	1.0
15	11	1.0
15	12	1.0
15	16	1.0
15	17	1.0
15	18	1.0
15	19	1.0
16	6	1.0
16	7	1.0
16	8	1.0
16	9	1.0
16	10	1.0
16	11	1.0
16	12	1.0
16	13	1.0
16	16	1.0
16	17	1.0
16	18	1.0
16	19	1.0
17	6	1.0
17	7	1.0
17	8	1.0
17	9	1.0
17	10	1.0
17	11	1.0
17	12	1.0
17	13	1.0
17	14	1.0
17	15	1.0
17	16	1.0
17	17	1.0
17	18	1.0
18	6	1.0

18	7	1.0
18	8	1.0
18	9	1.0
18	10	1.0
18	11	1.0
18	12	1.0
18	13	1.0
18	14	1.0
18	17	1.0
18	18	1.0
18	19	1.0
18	20	1.0
19	5	1.0
19	6	1.0
19	7	1.0
19	8	1.0
19	9	1.0
19	10	1.0
19	11	1.0
19	12	1.0
19	13	1.0
19	14	1.0
19	19	1.0
19	20	1.0
20	5	1.0
20	6	1.0
20	7	1.0
20	8	1.0
20	9	1.0
20	10	1.0
20	11	1.0
20	12	1.0
21	4	1.0
21	5	1.0
21	6	1.0
21	7	1.0
21	8	1.0
21	9	1.0

21	10	1.0
21	11	1.0
21	12	1.0
21	13	1.0
21	14	1.0
21	15	1.0
22	3	1.0
22	4	1.0
22	5	1.0
22	6	1.0
22	7	1.0
22	8	1.0
22	9	1.0
22	10	1.0
22	11	1.0
22	12	1.0
22	13	1.0
22	14	1.0
22	15	1.0
23	2	1.0
23	3	1.0
23	4	1.0
23	5	1.0
23	6	1.0
23	7	1.0
23	8	1.0
23	9	1.0
23	10	1.0
23	11	1.0
23	12	1.0
23	13	1.0
23	14	1.0
23	15	1.0
24	2	1.0
24	3	1.0
24	4	1.0
24	5	1.0
24	6	1.0

24	7	1.0
24	8	1.0
24	9	1.0
24	11	1.0
24	12	1.0
24	13	1.0
24	14	1.0
24	15	1.0
25	2	1.0
25	3	1.0
25	4	1.0
25	5	1.0
25	6	1.0
25	7	1.0
25	8	1.0
25	9	1.0
25	13	1.0
25	14	1.0
26	2	1.0
26	3	1.0
26	4	1.0
26	5	1.0
26	6	1.0
26	7	1.0
26	8	1.0
26	9	1.0
27	3	1.0
27	4	1.0
27	5	1.0
27	6	1.0
27	7	1.0
27	8	1.0
27	9	1.0
28	3	1.0
28	4	1.0
28	5	1.0
28	6	1.0

265 NO. OF CHALK NODES

```
 4 13  1.0 CHALK NODAL COORDINATES, NODAL RECHARGE FRACTION.
 5 13  1.0
 5 14  1.0
 5 15  1.0
 6 13  1.0
 6 14  1.0
 7 13  1.0
 7 14  1.0
 8 12  1.0
 8 13  1.0
 8 14  1.0
 9 12  1.0
 9 13  1.0
 9 14  1.0
10 12  1.0
10 13  1.0
10 14  1.0
11 12  1.0
11 13  1.0
11 14  1.0
11 15  1.0
11 23  1.0
12 12  1.0
12 13  1.0
12 14  1.0
12 22  1.0
12 23  1.0
13 12  1.0
13 13  1.0
13 14  1.0
13 21  1.0
13 22  1.0
13 23  1.0
14 13  1.0
14 14  1.0
14 18  1.0
14 21  1.0
14 22  1.0
```

14	23	1.0
14	24	1.0
15	13	1.0
15	14	1.0
15	15	1.0
15	20	1.0
15	21	1.0
15	22	1.0
15	23	1.0
15	24	1.0
16	14	1.0
16	15	1.0
16	20	1.0
16	21	1.0
16	22	1.0
16	23	1.0
16	24	1.5
16	25	1.5
17	19	1.0
17	20	1.0
17	21	1.0
17	22	1.0
17	23	1.0
17	24	1.0
17	25	1.0
18	15	1.0
18	16	1.0
18	21	1.0
18	22	1.0
18	23	1.0
18	24	1.0
18	25	1.0
19	15	1.0
19	16	1.0
19	17	1.0
19	18	1.0
19	21	1.0
19	22	1.0

19	23	1.0
19	24	1.0
19	25	1.0
19	26	1.0
20	13	1.0
20	14	1.0
20	15	1.0
20	16	1.0
20	17	1.0
20	18	1.0
20	19	1.0
20	20	1.0
20	21	1.0
20	22	1.0
20	23	1.0
20	24	1.0
20	25	1.0
20	26	1.0
21	16	1.0
21	17	1.0
21	18	1.0
21	19	1.0
21	20	1.0
21	21	1.0
21	22	1.0
21	23	1.0
21	24	1.0
21	25	1.0
21	26	1.0
21	27	1.0
22	16	1.0
22	17	1.0
22	18	1.0
22	19	1.0
22	20	1.0
22	21	1.0
22	22	1.0
22	23	1.0

22	24	1.0
22	25	1.0
22	26	1.0
22	27	1.0
23	16	1.0
23	17	1.0
23	18	1.0
23	19	1.0
23	20	1.0
23	21	1.0
23	22	1.0
23	23	1.0
23	24	1.0
23	25	1.0
23	26	1.0
23	27	1.0
23	28	1.0
24	10	1.0
24	16	1.0
24	17	1.0
24	18	1.0
24	19	1.0
24	20	1.0
24	21	1.0
24	22	1.0
24	23	1.0
24	24	1.0
24	25	1.0
24	26	1.0
24	27	1.0
24	28	1.0
25	10	1.0
25	11	1.0
25	12	1.0
25	15	1.0
25	16	1.0
25	17	1.0
25	18	1.0

25	19	1.0
25	20	1.0
25	21	1.0
25	22	1.0
25	23	1.0
25	24	1.0
25	25	1.0
25	26	1.0
25	27	1.0
25	28	1.0
26	10	1.0
26	11	1.0
26	12	1.0
26	13	1.0
26	14	1.0
26	15	1.0
26	16	1.0
26	17	1.0
26	18	1.0
26	19	1.0
26	20	1.0
26	21	1.0
26	22	1.0
26	23	1.0
26	24	1.0
26	25	1.0
26	26	1.0
26	27	1.0
26	28	1.0
27	11	1.0
27	12	1.0
27	13	1.0
27	14	1.0
27	15	1.0
27	16	1.0
27	17	1.0
27	18	1.0
27	19	1.0

27	20	1.0
27	21	1.0
27	22	1.0
27	23	1.0
27	24	1.0
27	25	1.0
27	26	1.0
27	27	1.0
27	28	1.0
27	29	1.0
28	14	1.0
28	15	1.0
28	16	1.0
28	17	1.0
28	18	1.0
28	19	1.0
28	20	1.0
28	21	1.0
28	22	1.0
28	23	1.0
28	24	1.0
28	25	1.0
28	26	1.0
28	27	1.0
28	28	1.0
28	29	1.0
29	16	1.0
29	17	1.0
29	18	1.0
29	19	1.0
29	20	1.0
29	21	1.0
29	22	1.0
29	23	1.0
29	24	1.0
29	25	1.0
29	26	1.0
29	27	1.0

29	28	1.0
29	29	1.0
30	18	1.0
30	19	1.0
30	20	1.0
30	21	1.0
30	22	1.0
30	23	1.0
30	24	1.0
30	25	1.0
30	26	1.0
30	27	1.0
30	28	1.0
30	29	1.0
31	20	1.0
31	21	1.0
31	22	1.0
31	23	1.0
31	24	1.0
31	25	1.0
31	26	1.0
31	27	1.0
31	28	1.0
31	29	1.0
32	22	1.0
32	23	1.0
32	24	1.0
32	25	1.0
32	26	1.0
32	27	1.0
32	28	1.0
32	29	1.0
33	23	1.0
33	24	1.0
33	25	1.0
33	26	1.0
33	27	1.0

Table (B5) Input file "nodes.dat"

```
 0.00 1.0  1.0
 1.70 0.0000001
2 7
3 6
3 7
4 6
5 5
5 6
6 5
7 5
8 4
8 5
9 4
10 4
11 4
12 4
12 5
13 5
14 5
15 5
15 6
16 6
17 6
18 6
19 5
19 6
20 5
21 4
21 5
22 3
22 4
23 2
23 3
24 2
25 2
26 2
26 3
27 3
```

28	3
28	4
28	5
28	6
27	6
27	7
27	8
27	9
26	9
26	10
26	11
27	11
27	12
27	13
27	14
28	14
28	15
28	16
29	16
29	17
29	18
30	18
30	19
30	20
31	20
31	21
31	22
32	22
32	23
33	23
33	24
33	25
33	26
33	27
32	27
32	28
32	29
31	29

30	29
29	29
28	29
27	29
27	28
26	28
25	28
24	28
23	28
23	27
22	27
21	27
21	26
20	26
19	26
19	25
18	25
17	25
16	25
16	24
15	24
14	24
14	23
13	23
12	23
11	23
11	22
10	22
10	21
9	21
8	21
8	20
8	19
7	19
7	18
7	17
6	17
6	16

6 15
5 15
5 14
5 13
4 13
4 12
4 11
3 11
3 10
3 9
2 9
2 8
1 1

	33	32	31	30	29	28	27	26	25	24	23	22	21
29	.	750.	750.	750.	750.	750.	750.	.	.	.	.	.	.
28	.	750.	750.	750.	750.	750.	750.	2500.	2500.	2500.	2500.	.	.
27	750.	750.	750.	2500.	2500.	2500.	2500.	2500.	2500.	2500.	2500.	2500.	2500.
26	750.	750.	750.	2500.	2500.	2500.	2500.	2500.	2500.	2500.	2500.	2500.	2500.
25	750.	750.	2600.	2600.	2600.	2600.	2600.	2600.	2600.	2600.	2500.	2500.	2500.
24	750.	750.	2300.	2300.	2300.	2300.	2300.	2800.	2800.	2800.	2500.	2500.	2500.
23	750.	750.	2300.	2300.	2300.	2300.	2300.	2800.	2800.	2800.	2300.	2500.	2800.
22	750.	750.	1250.	2300.	2300.	2300.	2300.	2800.	2800.	2800.	2800.	2800.	2800.
21	.	.	1000.	1500.	1750.	1750.	2300.	2800.	2800.	2800.	2800.	2800.	2800.
20	.	.	1000.	1250.	1250.	1250.	1250.	2300.	2300.	2300.	2300.	2800.	2800.
19	.	.	750.	750.	750.	750.	1250.	2250.	2300.	2300.	2300.	2300.	2500.
18	.	.	750.	750.	750.	750.	1250.	1250.	1250.	1250.	1750.	2300.	2300.
17	.	.	750.	750.	750.	750.	1250.	1000.	1000.	1000.	1250.	1750.	1750.
16	.	.	750.	750.	750.	750.	1000.	1000.	1000.	900.	900.	1250.	1250.
15	.	.	650.	650.	650.	650.	900.	900.	900.	900.	900.	750.	900.
14	.	.	650.	650.	650.	650.	350.	500.	750.	750.	750.	750.	750.
13	.	.	350.	350.	350.	350.	350.	500.	500.	750.	750.	750.	500.
12	.	.	350.	350.	350.	350.	350.	500.	500.	500.	500.	500.	250.
11	.	.	.	.	.	.	500.	500.	750.	750.	500.	250.	250.
10	.	.	.	.	.	.	500.	500.	750.	750.	250.	250.	250.
9	.	.	.	.	.	.	500.	500.	750.	750.	250.	250.	450.
8	.	.	.	.	.	.	250.	500.	750.	750.	250.	150.	250.
7	.	.	.	.	.	50.	50.	250.	250.	250.	250.	50.	50.
6	.	.	.	.	.	50.	50.	50.	50.	50.	50.	50.	50.

5	.	.	.	.	.	50.	50.	50.	50.	50.	50.	50.	50.
4	.	.	.	.	.	50.	50.	50.	50.	50.	50.	50.	50.
3	.	.	.	.	.	50.	50.	50.	50.	50.	50.	50.	50.
2	.	.	.	.	.	50.	50.	50.	50.	50.	50.	.	.

	20	19	18	17	16	15	14	13	12	11	10	9	8
29	.	.	.	.	.	.	.	.	.	.	.	.	.
28	.	.	.	.	.	.	.	.	.	.	.	.	.
27	.	.	.	.	.	.	.	.	.	.	.	.	.
26	250.	250.	.	.	.	.	.	.	.	.	.	.	.
25	250.	150.	150.	200.	200.	.	.	.	.	.	.	.	.
24	250.	150.	150.	200.	200.	150.	500.	500.	.	.	.	.	.
23	2700.	150.	150.	150.	150.	200.	500.	1000.	650.	650.	650.	.	.
22	2800.	150.	150.	150.	150.	1500.	1500.	1500.	650.	650.	650.	200.	200.
21	2800.	250.	250.	250.	250.	1500.	450.	450.	650.	650.	650.	200.	200.
20	2800.	2800.	2700.	2700.	1500.	2500.	450.	450.	650.	650.	650.	200.	200.
19	2700.	2700.	2700.	2700.	2700.	2500.	2500.	1500.	1000.	750.	750.	350.	750.
18	2300.	2700.	2700.	2700.	2700.	2500.	2500.	1500.	1000.	1250.	1250.	350.	750.
17	2000.	2500.	2700.	2700.	2700.	2500.	2500.	1500.	1000.	1500.	1250.	350.	250.
16	1750.	1750.	2500.	1250.	750.	750.	750.	750.	750.	1500.	1250.	350.	250.
15	750.	750.	1250.	750.	750.	750.	750.	750.	750.	2500.	1250.	350.	250.
14	250.	250.	750.	250.	250.	750.	750.	750.	750.	2500.	1250.	350.	250.
13	250.	250.	250.	250.	250.	750.	500.	750.	1650.	1650.	1250.	350.	250.
12	250.	150.	250.	250.	250.	750.	450.	750.	1650.	1650.	1000.	350.	250.
11	250.	150.	150.	150.	150.	600.	450.	750.	750.	750.	750.	350.	250.
10	250.	150.	150.	150.	150.	350.	450.	50.	50.	550.	350.	350.	250.
9	250.	150.	150.	150.	150.	250.	350.	50.	50.	550.	250.	250.	250.
8	150.	50.	50.	50.	50.	150.	250.	50.	50.	50.	50.	50.	50.
7	150.	50.	50.	50.	50.	150.	250.	50.	50.	50.	50.	50.	50.
6	50.	50.	50.	50.	50.	50.	150.	50.	50.	50.	50.	50.	50.
5	50.	50.	.	.	.	50.	50.	50.	50.	50.	50.	50.	50.
4	.	.	.	.	.	50.	50.	50.	50.	50.	50.	50.	50.
3	.	.	.	.	.	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.	.	.	.	.	.

	7	6	5	4	3	2
29	.	.	.	.	.	.
28	.	.	.	.	.	.
27	.	.	.	.	.	.
26	.	.	.	.	.	.

25	.	.	.	.	.	.
24	.	.	.	.	.	.
23	.	.	.	.	.	.
22	.	.	.	.	.	.
21	.	.	.	.	.	.
20	.	.	.	.	.	.
19	200.	200.	.	.	.	.
18	250.	200.	.	.	.	.
17	250.	200.	.	.	.	.
16	250.	150.	200.	200.	200.	.
15	250.	150.	250.	200.	200.	.
14	150.	150.	250.	200.	200.	.
13	150.	150.	250.	200.	200.	.
12	150.	150.	250.	200.	200.	.
11	150.	100.	200.	200.	200.	.
10	150.	50.	50.	50.	50.	50.
9	150.	50.	50.	50.	50.	50.
8	50.	50.	50.	50.	50.	50.
7	50.	50.	50.	50.	50.	50.
6	50.	50.	50.	50.	50.	50.
5	50.	50.	50.	50.	50.	50.
4	50.	.	.	.	.	.
3	.	.	.	.	.	.
2	.	.	.	.	.	.

	33	32	31	30	29	28	27	26	25	24	23	22
29	.	.01000	.01000	.01000	.01000	.01000	.01000	.	.	.	.	.
28	.	.01000	.01000	.01000	.01000	.01000	.01000	.01700	.01700	.01700	.01700	.
27	.01000	.01000	.01000	.02000	.02000	.02000	.02000	.01700	.01700	.01700	.01700	.02000
26	.01000	.01000	.01000	.02000	.02000	.02000	.02000	.01700	.01700	.01700	.01700	.02000
25	.01000	.01000	.02000	.02000	.02000	.02000	.02000	.01700	.01700	.01700	.01700	.02000
24	.01000	.01000	.02000	.02000	.02000	.02000	.02000	.01700	.01700	.01700	.01700	.02000
23	.01000	.01000	.02000	.02000	.02000	.02000	.02000	.01700	.01700	.01700	.01700	.02000
22	.01000	.01000	.02000	.02000	.02000	.02000	.02000	.02000	.02000	.02000	.02000	.02000
21	.	.	.02000	.02000	.02000	.02000	.02000	.02000	.02000	.02000	.02000	.02000
20	.	.	.02000	.02000	.02000	.02000	.01500	.01500	.01500	.02000	.02000	.02000
19	.	.	.02000	.02000	.02000	.02000	.01500	.01500	.01500	.02000	.01500	.02000
18	.	.	.02000	.02000	.02000	.02000	.02000	.02000	.02000	.01500	.01500	.02000
17	.	.	.02000	.02000	.02000	.02000	.02000	.02000	.02000	.01500	.01500	.01500

	20	19	18	17	16	15	14	13	12	11	10	9
16	.	.	.02000	.02000	.02000	.02000	.02000	.02000	.02000	.01500	.01500	.01500
15	.	.	.01000	.01000	.01000	.01000	.02000	.02000	.02000	.01500	.01500	.01500
14	.	.	.01000	.01000	.01000	.01000	.02000	.02000	.02000	.00200	.00150	.00200
13	.	.	.01000	.01000	.01000	.01000	.02000	.02000	.02000	.00200	.00150	.00200
12	.	.	.01000	.01000	.01000	.01000	.02000	.02000	.02000	.00200	.00150	.00100
11	.	.	.	.	.	.	.02000	.02000	.00200	.00200	.00150	.00100
10	.	.	.	.	.	.	.02000	.02000	.00200	.00200	.00150	.00100
9	.	.	.	.	.	.	.01000	.01000	.00200	.00200	.00150	.00100
8	.	.	.	.	.	.	.01000	.01000	.00100	.00100	.00010	.00050
7	.	.	.	.	.	.00010	.00010	.00010	.00100	.00100	.00010	.00010
6	.	.	.	.	.	.00010	.00010	.00010	.00010	.00010	.00010	.00010
5	.	.	.	.	.	.00010	.00010	.00010	.00010	.00010	.00010	.00010
4	.	.	.	.	.	.00010	.00010	.00010	.00010	.00010	.00010	.00010
3	.	.	.	.	.	.00010	.00010	.00010	.00010	.00010	.00010	.00010
2	.	.	.	.	.	.00010	.00010	.00010	.00010	.00010	.00010	.

	20	19	18	17	16	15	14	13	12	11	10	9
29	.	.	.	.	.	.	.	.	.	.	.	.
28	.	.	.	.	.	.	.	.	.	.	.	.
27	.	.	.	.	.	.	.	.	.	.	.	.
26	.02000	.02000	.	.	.	.	.	.	.	.	.	.
25	.02000	.02000	.02000	.02000	.02000	.	.	.	.	.	.	.
24	.02000	.02000	.02000	.02000	.02000	.02000	.01500	.01500	.	.	.	.
23	.02000	.02000	.02000	.02000	.02000	.02000	.01500	.01500	.01400	.00550	.00550	.
22	.02000	.02000	.02000	.02000	.02000	.02000	.01500	.01500	.01400	.00550	.00550	.00200
21	.02000	.02000	.02000	.02000	.02000	.02000	.01500	.01400	.01400	.01000	.00550	.00200
20	.02000	.02000	.02000	.02000	.02000	.02000	.01500	.01400	.01400	.01000	.00550	.00200
19	.02000	.02000	.02000	.02000	.02000	.02000	.02000	.01400	.01400	.01400	.01400	.00200
18	.02000	.02000	.01500	.01500	.02000	.02000	.02000	.01400	.01400	.01400	.01400	.00200
17	.02000	.02000	.01500	.01500	.02000	.02000	.02000	.01400	.01400	.01400	.01400	.00200
16	.01500	.01500	.01500	.02000	.00700	.00700	.00700	.00700	.00700	.02000	.02000	.01000
15	.01500	.01500	.01500	.02000	.00700	.00700	.00700	.00700	.00700	.02000	.02100	.02100
14	.00150	.00150	.00500	.02000	.00700	.00700	.00700	.00700	.00700	.02000	.02100	.02100
13	.00150	.00150	.00150	.02000	.00700	.01000	.01000	.01000	.01000	.01000	.02100	.02100
12	.00150	.00150	.02000	.00700	.00700	.01000	.01000	.01000	.01000	.01000	.00300	.00300
11	.00150	.00150	.02000	.00050	.00050	.01000	.01000	.01000	.01000	.01000	.00300	.00300
10	.00150	.02000	.02000	.00050	.00050	.00010	.00010	.00010	.00010	.00300	.00300	.00300
9	.02000	.00050	.00010	.00050	.00050	.00010	.00010	.00010	.00010	.00300	.00300	.00300
8	.00050	.00050	.00010	.00050	.00050	.00010	.00010	.00010	.00010	.00010	.00010	.00010

7	.00010	.00010	.00010	.00050	.00050	.00010	.00010	.00010	.00010	.00010	.00010	.00010
6	.00010	.00010	.00010	.00050	.00050	.00010	.00010	.00010	.00010	.00010	.00010	.00010
5	.00010	.00010	.	.	.	.00010	.00010	.00010	.00010	.00010	.00010	.00010
4	.	.	.	.	.	.00010	.00010	.00010	.00010	.00010	.00010	.00010
3	.	.	.	.	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.	.	.	.	.

	7	6	5	4	3	2
29	.	.	.	.	.	.
28	.	.	.	.	.	.
27	.	.	.	.	.	.
26	.	.	.	.	.	.
25	.	.	.	.	.	.
24	.	.	.	.	.	.
23	.	.	.	.	.	.
22	.	.	.	.	.	.
21	.	.	.	.	.	.
20	.	.	.	.	.	.
19	.00200	.00200	.	.	.	.
18	.00200	.00200	.	.	.	.
17	.00200	.00200	.	.	.	.
16	.02000	.00400	.00400	.00400	.00400	.
15	.02000	.02000	.02000	.00400	.00400	.
14	.02000	.02000	.02000	.00400	.00400	.
13	.02000	.02000	.02000	.00400	.00400	.
12	.02000	.02000	.01500	.00400	.00400	.
11	.00150	.01500	.00500	.00400	.00400	.
10	.00050	.00500	.00010	.00010	.00010	.00010
9	.00050	.00010	.00010	.00010	.00010	.00010
8	.00010	.00010	.00010	.00010	.00010	.00010
7	.00010	.00010	.00010	.00010	.00010	.00010
6	.00010	.00010	.00010	.00010	.00010	.00010
5	.00010	.00010	.00010	.00010	.00010	.00010
4	.00010	.	.	.	.	.
3	.	.	.	.	.	.
2	.	.	.	.	.	.

AQUIFER BASE LEVELS

	33	32	31	30	29	28	27	26	25	24	23	22	21
29	0.0	6.0	6.0	6.0	6.0	6.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0
28	0.0	6.0	6.0	6.0	6.0	6.0	10.0	10.0	10.0	10.0	10.0	0.0	0.0

27	6.0	6.0	6.0	6.0	6.0	10.0	10.0	10.0	10.0	10.0	10.0	15.0	15.0
26	6.0	6.0	6.0	6.0	10.0	10.0	10.0	10.0	10.0	10.0	15.0	15.0	15.0
25	6.0	6.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	15.0	15.0	15.0
24	6.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	15.0	15.0	15.0	15.0
23	6.0	10.0	10.0	10.0	10.0	10.0	10.0	15.0	15.0	15.0	15.0	15.0	15.0
22	0.0	10.0	10.0	10.0	10.0	10.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
21	0.0	0.0	10.0	10.0	10.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
20	0.0	0.0	10.0	10.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0
19	0.0	0.0	0.0	10.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	18.0
18	0.0	0.0	0.0	10.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	18.0
17	0.0	0.0	0.0	0.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	15.0	18.0
16	0.0	0.0	0.0	0.0	15.0	15.0	15.0	15.0	15.0	15.0	18.0	18.0	18.0
15	0.0	0.0	0.0	0.0	0.0	15.0	15.0	15.0	15.0	18.0	18.0	18.0	18.0
14	0.0	0.0	0.0	0.0	0.0	15.0	15.0	15.0	18.0	18.0	18.0	18.0	18.0
13	0.0	0.0	0.0	0.0	0.0	0.0	15.0	15.0	18.0	18.0	18.0	18.0	18.0
12	0.0	0.0	0.0	0.0	0.0	0.0	15.0	15.0	18.0	18.0	18.0	18.0	18.0
11	0.0	0.0	0.0	0.0	0.0	0.0	15.0	15.0	18.0	18.0	18.0	18.0	18.0
10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	18.0	18.0	18.0	18.0	18.0	18.0
9	0.0	0.0	0.0	0.0	0.0	0.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0
8	0.0	0.0	0.0	0.0	0.0	0.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0
7	0.0	0.0	0.0	0.0	0.0	0.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0
6	0.0	0.0	0.0	0.0	0.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0
5	0.0	0.0	0.0	0.0	0.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0
4	0.0	0.0	0.0	0.0	0.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0
3	0.0	0.0	0.0	0.0	0.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	18.0	18.0	18.0	18.0	0.0	0.0
	20	19	18	17	16	15	14	13	12	11	10	9	8
29	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
26	18.0	18.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
25	18.0	18.0	18.0	18.0	18.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
24	18.0	18.0	18.0	18.0	18.0	18.0	18.0	0.0	0.0	0.0	0.0	0.0	0.0
23	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	30.0	30.0	0.0	0.0	0.0
22	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	30.0	30.0	30.0	0.0	0.0
21	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0
20	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0
19	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0

18	18.0	19.0	18.0	18.0	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0
17	18.0	18.0	18.0	18.0	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0
16	18.0	18.0	18.0	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0	30.0
15	18.0	18.0	18.0	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0	30.0
14	18.0	18.0	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
13	18.0	18.0	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
12	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
11	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
10	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
9	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
8	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
7	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
6	18.0	18.0	18.0	18.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
5	18.0	18.0	0.0	0.0	0.0	30.0	18.0	30.0	30.0	30.0	30.0	30.0	30.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	30.0	30.0	30.0	30.0	30.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

	7	6	5	4	3	2
29	0.0	0.0	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0	0.0
19	30.0	0.0	0.0	0.0	0.0	0.0
18	30.0	0.0	0.0	0.0	0.0	0.0
17	30.0	30.0	0.0	0.0	0.0	0.0
16	30.0	30.0	0.0	0.0	0.0	0.0
15	30.0	30.0	30.0	0.0	0.0	0.0
14	30.0	30.0	30.0	0.0	0.0	0.0
13	30.0	30.0	30.0	30.0	0.0	0.0
12	30.0	30.0	30.0	30.0	0.0	0.0
11	30.0	30.0	30.0	30.0	30.0	0.0
10	30.0	30.0	30.0	30.0	30.0	0.0

```
  9    30.0  30.0  30.0  30.0  30.0  30.0
  8    30.0  30.0  30.0  30.0  30.0  30.0
  7    30.0  30.0  30.0  30.0  30.0  30.0
  6    30.0  30.0  30.0  30.0  30.0   0.0
  5    30.0  30.0  30.0   0.0   0.0   0.0
  4     0.0   0.0   0.0   0.0   0.0   0.0
  3     0.0   0.0   0.0   0.0   0.0   0.0
  2     0.0   0.0   0.0   0.0   0.0   0.0
33 24 0.0       NODAL COORDS, FIXED HEAD LEVEL
33 25 0.0       NODAL COORDS, FIXED HEAD LEVEL
33 26 0.0       NODAL COORDS, FIXED HEAD LEVEL
1 1 0.0         TERMINATION OF FIXED HEAD ENTRY
2               NO. OF TIME STEPS-1 PER MONTH
10              DAY FIRST TIME STEP ENDS
20              DAY SECOND TIME STEP ENDS
400,150         MAX. NO. OF CYCLES FOR STEADY STATE, TRANSIENT S.O.R.
1                 ┐
1                 ├ OBSOLETE (USED FOR FIXED HEAD DATA BLOCK)
33 24 33 26 0     ┘
var             PARAMETER TO SPECIFY VARIABLE TRANSMISSIVITY, OTHERWISE "cst"
2               NO. OF YEARS FOR DETAILED FLOW OUTPUT
1970            FIRST YEAR OF DETAILED FLOW OUTPUT
1971            SECOND YEAR OF DETAILED FLOW OUTPUT, ETC.
0.7 0.3         RUNCURR, RUNPREV - FACTORS TO PROPORTION RUNOFF BETWEEN TIME STEPS
0.62            RATE - FACTOR USED TO DETERMINE CUTOFF CHANNEL FLOW RATE
5.0 0.9  1      DETAILS OF SATURATED DEPTH VS. TRANSMISSIVITY RELATIONSHIP
```

Table (B6) Input file "river.dat"

```
11              NO OF WATERCOURSES
Tributary 1     NAME OF WATERCOURSE #1
0.08 0.0        FRACTION OF B.C. RUNOFF, CHALK RUNOFF
7               NO. OF B.C. NODES
1               NO. OF CHALK NODES
Tributary 2     ETC.
0.07 0.0
4
1
Linnet
.18 .05
10
2
Tributary 3
0.0  .10
1
7
Tributary 4
0.0  .06
1
3
Cavenham
.18  .16
8
7
Tuddenham
0.0   .10
1
4
Tributary k
0.0   .06
1
4
Kennet
.10   .15
12
13
Lark
```

```
.19    .35
15
18
Cutoff channel
0.0    .06
1
3                                                         EACH LINE:
14  7     90      .000005   .051   0 -T  I,J,RIVER STAGE,
13  7     80      .000005   .092   0  |    LEAKAGE COEFF,
13  8     76      .000005   .107   0  |      NODAL FRACTION OF
13  9     70      .000005   .133   0  |        STREAM RUNOFF: B.C.,CHALK
12  9     60      .000005   .158   0  |
11  9     60      .000005   .214   0  |
10  9     50      .000005   .245   0  |
9   9     48       0         0     0  V

5  10     72      .000005   .083   0
6  10     66      .000005   .207   0
7  11     56      .000005   .298   0
8  12     41      .000005   .413   0
9  12     31      0          0     0

16  7     107     .000060   .01    0
16  8     91      .000060   .01    0
16  9     83      .000060   .02    0
16  10    73      .000060   .02    0
15  10    68      .000025   .04    0
15  11    60      .000010   .08    0
15  12    52      .000010   .12    0
14  13    42      .000008   .15    0
13  13    32      .000008   .20    0
12  12    30      .000008   .35    0
11  12    28      .000008     0    .5
10  13    27      0             0    .5
```

8 20	36.0	.0002	0	.060
9 20	26.0	.0002	0	.065
9 19	26.0	.0004	0	.085
10 19	24.0	.0005	0	.105
11 19	23.0	.0008	0	.115
12 19	21.5	.00085	0	.120
13 19	19.0	.0010	0	.205
14 19	17.0	0	0	.245
13 21	25	0	0	.20
14 21	20	.0025	0	.25
14 20	19	.0025	0	.28
15 20	15	0	0	.37
20 9	105	.00010	.02	0
19 10	85	.00100	.05	0
18 10	73	.00700	.07	0
18 11	70	.01500	.10	0
18 12	61	.01500	.13	0
17 13	53	.01500	.17	0
17 14	52	.01500	.21	0
18 14	51	.01500	.25	0
19 15	45	.00500	0	.08
19 16	34	.00100	0	.09
19 17	21	.00020	0	.11
19 18	16	.00020	0	.13
19 19	15	.00020	0	.16
19 20	14	.00020	0	.18
18 21	11	.0	0	.25
24 19	15.0	.01500	0	.10
23 20	12.0	.01500	0	.16
23 21	10.0	.01500	0	.20
23 22	9.0	.01500	0	.24
23 23	8.5	.0	0	.30

29	17	14	.0015	0	.10
29	18	11	.0015	0	.12
29	19	9.5	.0015	0	.20
30	20	9.5	.0015	0	.28
29	21	6	0	0	.30
25	2	87	.00003	.05	0
24	3	85	.00003	.06	0
24	4	80	.00030	.07	0
24	5	79	.00030	.08	0
25	5	75	.00030	.10	0
26	6	73	.00030	.10	0
26	7	65	.00030	.10	0
25	7	64	.00030	.10	0
24	8	61	.00030	.10	0
24	9	54	.00030	.10	0
24	10	50	.00030	0	.077
24	11	49	.00060	.14	0
25	12	48	.00060	0	.077
26	13	44	.00060	0	.077
26	14	41	.00040	0	.077
26	15	25	.00009	0	.077
25	16	25	.00009	0	.077
25	17	25	.00009	0	.077
26	18	15	.00009	0	.077
27	19	12	.00009	0	.077
28	20	7	.00052	0	.077
29	21	6	.0005	0	.077
30	22	5	.0005	0	.077
30	23	5	.0025	0	.077
29	24	4	0	0	.077
11	5	90	.000100	.02	0
10	5	90	.000100	.03	0
9	6	70	.000100	.042	0
9	7	55	.000110	.056	0
9	9	48	.00010	.070	0
8	10	38	.00010	.078	0
9	11	32	.00020	.088	0

9	12	31	.00045	.098	0
10	13	27.0	.0060	0	.056
10	14	26.5	.0060	0	.056
11	15	24.5	.0060	0	.056
11	16	24.0	.0040	.108	0
12	17	24.0	.0020	.122	0
13	18	23.5	.0020	.136	0
14	19	17.0	.0011	.148	0
15	20	15.0	.005	0	.056
16	20	12.5	.005	0	.056
17	20	12.0	.0350	0	.056
18	21	11.0	.0350	0	.056
19	22	10.5	.0250	0	.056
20	22	10.0	.0250	0	.056
21	22	9.5	.0200	0	.056
22	23	8.5	.0200	0	.056
23	23	8.5	.0162	0	.056
24	23	8.5	.0162	0	.056
25	23	8.3	.0162	0	.056
26	23	8.0	.0165	0	.056
27	23	7.6	.0165	0	.056
28	24	6.4	.0165	0	.056
29	24	4.0	.0165	0	.056
30	24	3.0	.0165	0	.056
31	25	2.0	.0165	0	.056
32	26	0.1	.0165	0	.056
23	24	5	.00005	0	.25
23	25	4.5	.00005	0	.25
23	26	4	.00005	0	.25
24	27	3.2	.00005	0	.25

Table (87) Input file "riverst.dat"

Appendix C

Listing of Source Code for Model

The computer program for the digital model listed below is written in FORTRAN77. Minor changes - related mainly to file handling considerations - are necessary to implement the program on other computer systems that support FORTRAN77.

The program has been developed in five parts, each in a separate file:

COMMON.INC: This section contains the common block, array and variable declarations for all of the routines addressed from the main program of the Lark model

CHAR.F77: subprogram to initialize character variables

LARK.F77: The main program of the Lark Chalk model.

SUB.F77: A series of subroutines that are used mainly in the data preparation and the input/output stages of a simulation run.

UTILITY.F77: The subroutines contained in this file are used in controlling the monitor display during the course of execution. They are not essential for the model and can be omitted if the necessary changes are made to the main program coding, at the lines where calls are made to these subroutines.

```
c----------------------------------------------------------------------
c                            COMMON.INC
c----------------------------------------------------------------------
c this file contains the common block, array and variable declarations
c for all of the routines addressed from the main program
c----------------------------------------------------------------------
      implicit double precision(a-h,o-z)
c----------------------------------------------------------------------
c dimensions and declaration of variable/array types
c character variables and arrays are initialized in 'CHAR.F77'
c----------------------------------------------------------------------
      character*12 indump,outdump
      character*20 format,input,output,rivdat,fldat,industdat,
     1 nodesdat,absdat,rechdat,rivd,well(30),river(15),monthp(3)
      character blk(20)*10,month(12)*12
      character cond_tran*3,iyrcha*4,cond*1,pass*5,passc*5
      character*82 adummy(3)
      integer yrstart,yrstop
c----------------------------------------------------------------------
c common blocks for character variables/arrays
c----------------------------------------------------------------------
      common/c1/cond_tran,iyrcha,cond,pass,passc
      common/c2/format,well,river,rechdat
      common/c3/blk,month,monthp,absdat,indump,outdump
      common/c4/input,output,rivdat,fldat,rivd,nodesdat
c----------------------------------------------------------------------
c common blocks for real and integer arrays
c----------------------------------------------------------------------
      common/area1/id(15),jd(15),ia(5),ja(5),iar(20),jar(20)
      common/area2/disc(1965:2000,5),efflt(1965:2000,10)
      common/area3/abir(1965:2000,20),listseq(100)
      common/area4/iw(30),jw(30),nday(1965:2000,12),x(40),y(40)
      common/area5/qfrech(3,1965:2000,12),qabs(15,1965:2000,12)
      common/area6/runoff(1965:2000,12),runofc(1965:2000,12)
      common/block1/ib(40,40),jb(40,40),ibb(40,40),jbb(40,40)
      common/block2/afact(40,40),bfact(40,40),st(40,40)
      common/block3/cfact(40,40),dfact(40,40),ssfac(40,40)
      common/block4/hhold(40,40),hold(40,40),hold1(40,40),h(40,40)
```

```
      common/block5/a(40,40),b(40,40),c(40,40),d(40,40),s(40,40)
      common/block6/rs(40,40),qav(20),tty(40,40),ttx(40,40)
      common/block7/tday(15),catrbc(15),catrc(15),mc(12),one(16)
      common/block8/ihf(40,40),rvc(40,40),qd(40,40),qbb(40,40)
      common/block9/qabsv(15),nnode(3),nkkk(2),icon(3,6),iyprt(25)
      common/block10/hx(40,40),hy(40,40),g(40,40),nb(6),ne(6)
      common/block11/rcoefbc(15,35,30),rcoefc(15,35,30)
      common/block12/ie(3,6),je(3,6),ist(3,6),jst(3,6),nr(2,15)
      common/block13/tx(40,40),ty(40,40),hfix(40,40),rchg(40,40)
      common/flblk/qr(40,40),qra(40,40),rconst(40,40),qbase(40,40)
      common/rnode/ir(2,80,15),jr(2,80,15),qnode(15,35,30)
      common/perm/px(40,40),py(40,40),base(40,40),z(40,40)
      common/node/ic(2,1100),jc(2,1100),fac1(2,1100)
c-------------------------------------------------------------
c common block declarations for real and integer variables
c-------------------------------------------------------------
      common/v1/m,n,min,nin,mbound,nbound,mfict,nfict,nab,qa,idate
      common/v2/imin,jmin,imax,jmax,nrech,nwell,ndis,iyear,iss,kday
      common/v3/nriver,imonth,iblock,nfirst,nfinal,outputchk,nc,imx,jmy
      common/v4/iyold,iyst,bass,date,rate,nyr,neflt,inputchk,tri,sfa c
      common/v5/ofac,ncyss,ncyts,yrstart,yrstop,mstart,mstop

c-------------------------------------------------------------
                            CHAR.F77
c-------------------------------------------------------------


c-------------------------------------------------------------
c subprogram to initialize character variables
c-------------------------------------------------------------
      BLOCK DATA CHAR
      include 'common.inc'
      data  blk/'one','two','three','four','five','six','seven',
     1 'eight','nine','ten','eleven','twelve','thirteen',
     2 'fourteen','fifteen','sixteen','seventeen',
     3 'eighteen','nineteen','twenty'/
      data month/'January','February','March','April','May','June',
     1  'July','August','September','October','November',
     2  'December'/
```

```
data monthp/'February','May','August'/
data one/1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1./
data mc/31,30,31,30,31,30,31,30,31,30,31,30/
end
```

```
c-----------------------------------------------------------------
                              PROGRAM LARK
c-----------------------------------------------------------------
c      ___________________________________________________________
c     |   _____________________________________________________   |
c     |  |                                                     |  |
c     |  |            A.W. - LARK CATCHMENT STUDY              |  |
c     |  |                GROUNDWATER MODEL                    |  |
c     |  |                                                     |  |
c     |  |                                                     |  |
c     |  |         DEPARTMENT OF CIVIL ENGINEERING             |  |
c     |  |           UNIVERSITY OF BIRMINGHAM                  |  |
c     |  |_____________________________________________________|  |
c     |___________________________________________________________|
c
c
c-----------------------------------------------------------------
c Note: all program segments enclosed by lines with > "GGGGGGGGGGG"
c contain coding for screen display
c-----------------------------------------------------------------
      include 'common.inc'
      include 'timer.inc'
      include 'dos.inc'
      integer dosblock(f77_dos_block_size)
      dimension factx(40,40),facty(40,40)
c-----------------------------------------------------------------
c top left hand corner numbered (2,2)
c m is number of mesh intervals in horizontal direction
c n is number of mesh intervals in vertical direction
c input number of mesh intervals
c-----------------------------------------------------------------
      call clear
      call normal
      call file_open
c-----------------------------------------------------------------
c dump file input routine
c-----------------------------------------------------------------
      if(inputchk.eq.1)then
cGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
          call color(32,40)
          call clear
```

```
      call cursor(0,0)
      print*,' '
      print*,' '
      print*,' '
      print*,' '
      print*,' '
      print*,' '
      print*,' '
      print*,' '
      print*,' '
      print*,(char(0),i=1,15),char(219),(char(223),i=1,46),
     *      char(219)
      print*,(char(0),i=1,15),char(219),
     *      '    Reading input dump file : ',
     *      indump,(char(0),i=1,4),char(219)
      print*,(char(0),i=1,15),char(219),(char(220),i=1,46),
     *      char(219)
      call color(30,40)
      print*,' '
      print*,' '
      print*,' '
      print*,' '
      print*,' '
      print*,' '
      print*,' '
      print*,' '
cGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
      close(10)
      open(10,file=indump,form='unformatted',status='unknown')
      rewind(10)
      read(10)cond_tran,cond,pass,passc,format,iyold,
     *      blk,month,monthp,input,output,rivdat,fldat,rivd,
     *      x,y,imx,jmy,ncyss,ncyts,h,a,b,c,d,s,qav,ihf,
     *      ib,jb,ibb,jbb,afact,bfact,qbb,qabsv,iyprt,
     *      st,cfact,dfact,ssfac,rcoefbc,rcoefc,sfac,g,
     *      tty,ttx,tday,catrbc,catrc,mc,one,ihf,rvc,qd,
     *      je,ist,jst,tx,ty,hfix,rchg,qr,qra,rconst,hx,hy,
     *      nkkk,qnode,px,py,z,m,n,min,nin,mbound,nbound,
```

```
     *     mfict,nfict,imin,jmin,imax,jmax,tri,ofac,error,
     *     imonth,iss,bass,date,nyr,tyinput,sinput,ie,
     *     hstart,sriv,hfixa,kday,iibeg,iiend,jjbeg,jjend,
     *     flownow,flowb4,rate,alpk,ntran,txinput,nc,icon
           close(10)
           call river_data(catrbc,catrc,rcoefbc,rcoefc)
           call irigindent
           call recabsent
           iseq=0
cGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
           call color(33,40)
           call cursor(0,0)
           print*,' '
           print*,' '
           print*,' '
           print*,' '
           print*,' '
           print*,' '
           print*,' '
           print*,' '
           print*,' '
           print*,(char(0),i=1,15),char(219),(char(223),i=1,46),
     *     char(219)
           print*,(char(0),i=1,15),char(219),
     *     '          Input dump file entered',
     *     (char(0),i=1,9),char(219)
           print*,(char(0),i=1,15),char(219),(char(220),i=1,46),
     *     char(219)
cGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
           write(28,*)'2'
           write(39,*)'1'
           write(29,*)'6'
           write(69,*)'6'
           write(51,*)'6'
           write(52,*)'4'
           goto 9000
        endif
c------------------------------------------------------------------
```

```
c generation of coordinates for the system
c-----------------------------------------------------------------------
      call cord
      call normal
c-----------------------------------------------------------------------
c initialise variables and arrays
c-----------------------------------------------------------------------
      do 61 i = 1,40
      do 61 j = 1,40
      hx(i,j)  =  0.0d0
      hy(i,j)  =  0.0d0
      base(i,j)   =  0.0d0
      z(i,j)  =  0.0d0
      g(i,j)  =  0.0d0
      rconst(i,j) =  0.0d0
      qr(i,j)  =  0.0d0
      qra(i,j)  =  0.0d0
      qd(i,j)  =  0.0d0
      tx(i,j)  =  0.0d0
      ty(i,j)  =  0.0d0
      ttx(i,j)  =  0.0d0
      tty(i,j)  =  0.0d0
      st(i,j)  =  0.0d0
      s(i,j)  =  0.0d0
      h(i,j)  =  0.0d0
      hfix(i,j)   =  0.0d0
      ihf(i,j)  =  0.0d0
      rchg(i,j)   =  0.0d0
      rs(i,j)  =  0.0d0
      a(i,j)  =  0.0d0
      b(i,j)  =  0.0d0
      c(i,j)  =  0.0d0
      d(i,j)  =  0.0d0
61    continue
c-----------------------------------------------------------------------
c ifbd = flag to control initial display of flow balance difference which
c will be incorrect for first time step of simulation, as there is no
c change in aquifer storage at that stage (ifbd=1 thereafter)
```

```
c------------------------------------------------------------------------
      ifbd=0
c------------------------------------------------------------------------
c input river parameters:
c nriver= no of rivers
c river(flag)= rivername
c catrbc(iriver)=catchment ratio
c nr(in,iriver)=1 or 2 for boulder clay and chalk respectively
c ir(), jr() are river nodes
c rcoefc()=coefficient for Chalk runoff
c rcoefbc()=coefficient for Boulder Clay runoff
c rconst()=leakage factor
c g()= river stage
c------------------------------------------------------------------------
      call river_data(catrbc,catrc,rcoefbc,rcoefc)
c------------------------------------------------------------------------
c numbering of boundary and fictitious nodes
c------------------------------------------------------------------------
      min=m+1
      nin=n+1
      mbound=m+2
      nbound=n+2
      mfict=m+3
      nfict=n+3
c------------------------------------------------------------------------
c nodes are read in into arrays ic( ) and jc( )
c nrech= no of recharge blocks
c nodes with nrech=1 are on boulder clay otherwise are on the chalk
c------------------------------------------------------------------------
      call irigindent
c------------------------------------------------------------------------
c reading of public water abstraction PWS
c nwell=no of wells
c well()= wellname
c iw() and jw() are well nodes
c qabs()=abstraction
c------------------------------------------------------------------------
      call recabsent
```

```
c------------------------------------------------------------------------
c reading overall aquifer parameters
c sriv=factor for storage coeff. for the river nodes
c tri=factor for river nodes transmissivity
c ofac=sor weighting factor
c error=convergence criterion
c------------------------------------------------------------------------
      read(1,*)hstart,sriv,tri
      do 121 i=1,40
      do 121 j=1,40
      hold(i,j)=hstart
121   hhold(i,j)=hstart
      read(1,*) ofac,error
      write(2,130)
130   format(4x,5hxmesh,2x,5hymesh,2x,
     1 13hinitial head,3x,6hfactor,4x,5herror)
      write(2,140) m,n,hstart,ofac,error
140   format(1x,i7,2x,i5,3x,2f12.4,f16.10)
      write(2,160)
160   format(' ')
c------------------------------------------------------------------------
      time=0.0
c------------------------------------------------------------------------
c call routine to determine boundary coefficients
c------------------------------------------------------------------------
      call fact
c------------------------------------------------------------------------
c call to calculate the recharge distribution factors,
c transmissivity and storativity
c------------------------------------------------------------------------
      call tran
c------------------------------------------------------------------------
c all fixed heads are set initially to -999999.0
c------------------------------------------------------------------------
      do 991 i=1,mfict
      do 992 j=1,nfict
      if(ibb(i,j).gt.0)hfix(i,j)=-999999.0d0
992   continue
```

```
991   continue
c-----------------------------------------------------------------------
c calculation of finite difference coefficients and modification
c of boundary node coefficients, no flow nodes are set to a flag
c-----------------------------------------------------------------------
      call fd(tx,ty,s)
c-----------------------------------------------------------------------
c input fixed heads
c-----------------------------------------------------------------------
230   continue
      read(1,*)i,j,hfixa
      write(2,34456)i,j,hfixa
34456 format(2i4,f7.2)
      if((i.eq.1).and.(j.eq.1))goto 250
      h(i,j)=hfixa
      hfix(i,j)=hfixa
      hold(i,j)=hfixa
      hhold(i,j)=hfixa
      ihf(i,j)=i
      go to 230
250   continue
c-----------------------------------------------------------------------
c set initial heads
c-----------------------------------------------------------------------
540   continue
      do 530 i=1,mfict
      do 530 j=1,nfict
      h(i,j)=hold(i,j)
530   if(hfix(i,j).eq.-99.9) h(i,j)=-99.9
c-----------------------------------------------------------------------
c print out initial conditions
c-----------------------------------------------------------------------
      imx=imax+1
      jmy=jmax+1
      call prin(tx,1,2,mbound,2,nbound)
      call prin(s,3,2,mbound,2,nbound)
      do 70 m=1,3
      write(2,'(a)')adummy(m)
```

```
      do 70 j=29,2,-1
70    write(2,5)j,(base(i,j),i=nb(m),ne(m),-1)
5     format(i3,1x,13(f6.1))
c-------------------------------------------------------------------
      do 1030 i=1,mfict
      do 1030 j=1,nfict
      st(i,j) =s(i,j)
      ttx(i,j)=tx(i,j)
      tty(i,j)=ty(i,j)
1030  continue
c---------------------------------------------------------------------
c calc average monthly flows for steady state calculation
c flow data for 1969 are used for the lark catchment
c---------------------------------------------------------------------
      do 516 k=1,nrech
516   qav(k)=0.0d0
      do 1555 nw=1,nwell
1555  qabsv(nw)=0.0d0
      runav=0.0
      do 517 imonth=1,12
517   runav=runav+runoff(1969,imonth)+runofc(1969,imonth)
      runav=runav/12.0d0
      do 590 k=1,nrech
      do 590 imonth=1,12
590   qav(k) = qav(k)+qfrech(k,1969,imonth)
      do 1556 nw=1,nwell
      do 1556 imonth=1,12
1556  qabsv(nw)=qabsv(nw)+qabs(nw,1969,imonth)*qa
      do 600 k=1,nrech
      qav(k)=qav(k)/12.0d0
600   continue
      do 1557 nw=1,nwell
      qabsv(nw)=qabsv(nw)/12.0d0
1557  continue
c---------------------------------------------------------------------
c set iss (= steady state test flag) negative for initial steady state
c---------------------------------------------------------------------
      iss= -100
```

```
c-------------------------------------------------------------
c input of times in days when calcn is performed
c-------------------------------------------------------------
      read(1,*)kday
      do 620 k=1,kday
620   read(1,*)tday(k)
c-------------------------------------------------------------
c ncy = no of cycles
c nc = no of fixed head nodes segment
c ist(), jst() = start of fixed nodes
c ie(), je() = end of fixed nodes
c-------------------------------------------------------------
      read(1,*)ncyss,ncyts
      read(1,*)nc
      do 621 l=1,nc
      read(1,*)nkkk(l)
      do 621 nll=1,nkkk(l)
621   read(1,*)ist(l,nll),jst(l,nll),ie(l,nll),je(l,nll),icon(l,nll)
c-------------------------------------------------------------
c flownow and flowb4 = runoff factors, current and previous time step
c rate = allowance made for cut-off channel water outflow
c-------------------------------------------------------------
      read(1,'(a3)')cond_tran
      read(1,*)nyr
      do 1312 iy=1,nyr
1312  read(1,*)iyprt(iy)
      read(1,*)flownow,flowb4
      read(1,*)rate
      read(1,*)bass,alpk
      iyold=yrstart
9000  continue
cGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
      call color(34,46)
      call clear
cGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
      write(2,160)
c-------------------------------------------------------------
c time increased  :  change in year
```

```
c-----------------------------------------------------------------------
      qaby=0.0
      qrey=0.0
c-----------------------------------------------------------------------
c iyear = current year
c-----------------------------------------------------------------------
      do 700 iyear=yrstart,yrstop
      m1=1
      m2=12
      if(iyear.eq.yrstart)m1=mstart
      if(iyear.eq.yrstop)m2=mstop
      if(iss.lt.0)then
         m1=1
         m2=1
         goto 750
      end if
      qabstv=0.0
      qrechv=0.0
      write(2,720)iyear
720   format(10x,9hYEAR NO.:,i5)
750   continue
c-----------------------------------------------------------------------
c set flag for output to 'mos.dat'
c-----------------------------------------------------------------------
      moscheck=0
c-----------------------------------------------------------------------
c change in month
c-----------------------------------------------------------------------
      do 730 imonth=m1,m2
      if(iss.lt.0)goto 751
      close(23)
      open(23,file='tem.del',status='unknown')
      rewind(23)
      write(23,111)iyear
111   format(i4)
      rewind(23)
      read(23,1121)iyrcha
1121  format(a4)
```

```
c-----------------------------------------------------------------------
c rivtw = modelled river flow at Temple Weir gauging station [Ml*10]
c rivih = modelled river flow at Isleham gauging station [Ml*10]
c-----------------------------------------------------------------------
       rivtw=0.0
       rivfm=0.0
       rivih=0.0
       bastw=0.0
       basfm=0.0
       basih=0.0
c-----------------------------------------------------------------------
c combine all flows per node in rs(i,j) [m**3/d]
c special calculation for initial heads
c-----------------------------------------------------------------------
      write(2,740)imonth,nday(iyear,imonth),(qfrech(k,iyear,imonth
     1),k = 1,nrech)
740   format(1x,7h month=  ,i4,10h no. days=  ,i4,7h flows= ,5f12.1)
751   continue
      do 5000 k=1,nrech
      do 800 nod=1,nnode(k)
      fa=fac1(k,nod)
      i=ic(k,nod)
      j=jc(k,nod)
      if(iss.lt.0)goto 810
      rs(i,j)=rchg(i,j)*qfrech(k,iyear,imonth)*fa
      go to 800
810   rs(i,j)=rchg(i,j)*qav(k)*fa
800   continue
5000  continue
      do 811 nd=1,ndis
      i=ia(nd)
      j=ja(nd)
      if(iss.gt.0)then
          rs(i,j)=rs(i,j)-disc(iyear,nd)
      else
          rs(i,j)=rs(i,j)-disc(1969,nd)
      endif
811   continue
```

```
      do 822 nd=1,nab
      i=iar(nd)
      j=jar(nd)
      if(iss.gt.0)then
          rs(i,j)=rs(i,j)-abir(iyear,nd)
      else
          rs(i,j)=rs(i,j)-abir(1969,nd)
      endif
822   continue
      do 820  nw=1,nwell
      i=iw(nw)
      j=jw(nw)
      if(iss.lt.0)goto 830
      rs(i,j)=rs(i,j)-qabs(nw,iyear,imonth)*qa
      go to 820
830   rs(i,j)=rs(i,j)-qabsv(nw)
820   continue
c-----------------------------------------------------------------------
c divide nodal flow by area to give m/d
c-----------------------------------------------------------------------
      do 840 i=2,imax+1
      do 840 j=2,jmax+1
840   rs(i,j)=4.0*rs(i,j)/((x(i+1)-x(i-1))*(y(j+1)-y(j-1)))
c-----------------------------------------------------------------------
c kday = 1 means monthly time step
c-----------------------------------------------------------------------
      if(kday.eq.1)lday=kday
      if(kday.gt.1)lday=kday+1
c-----------------------------------------------------------------------
c increase time ; calculate delt
c-----------------------------------------------------------------------
      do 900 iday=1,lday
      if(lday.eq.1)then
          if(iss.lt.0)then
              delt=float(nday(1969,imonth))
              goto 930
          else
              goto 920
```

```
          endif
      else
          if(iday.ne.1)goto 910
      endif
      delt=tday(1)
      date=tday(1)
      idate=int(date)
      go to 930
910   if(iday.eq.lday)goto 920
      delt=tday(iday)-tday(iday-1)
      date=tday(iday)
      idate=int(date)
      go to 930
920   date=float(nday(iyear,imonth))
      idate=int(date)
      delt=date-tday(kday)
c-----------------------------------------------------------------
c test for end of month
c-----------------------------------------------------------------
930   if(delt.le.0.001)goto 900
c-----------------------------------------------------------------
c start of s.o.r. calculation
c multiplying factor for storage for steady state analysis
c-----------------------------------------------------------------
      sfac=1.0
      if(iss.lt.0)sfac=0.00000000001d0
c-----------------------------------------------------------------
c multiplier and previous time step factors; use arrays tx(),ty()
c-----------------------------------------------------------------
      rdelt=1.0d0/delt
c-----------------------------------------------------------------
c factoring of river node transmissivity and injection of runoff
c into the nodes on the boulder clay
c-----------------------------------------------------------------
      do 6 iriver=1,nriver
      nnn=1
      nk=0
      ni=0
```

```
      do 30 in=1,2
      nk=nk+nr(in,iriver)
      do 31 l=nnn,nk
      ni=ni+1
      i=ir(in,l,iriver)
      j=jr(in,l,iriver)
      factx(i,j)=factx(i,j)*tri
      facty(i,j)=facty(i,j)*tri
      if(iss.lt.0)goto 45
      if(imonth.ne.1)then
          totroc=flownow*runofc(iyear,imonth)+flowb4*
     *    runofc(iyear,imonth-1)
          totrobc=flownow*runoff(iyear,imonth)+flowb4*
     *    runoff(iyear,imonth-1)
      else
          totroc=runofc(iyear,imonth)
          totrobc=runoff(iyear,imonth)
      endif
      qnode(iriver,i,j)=totrobc*catrbc(iriver)*rcoefbc(iriver,i,j)
     * +totroc*catrc(iriver)*rcoefc(iriver,i,j)
      goto 31
45    qnode(iriver,i,j)=runav*catrbc(iriver)
31    continue
      nnn=nk+1
30    continue
6     continue
c-----------------------------------------------------------------
c inclusion of industrial discharge into the river flow
c-----------------------------------------------------------------
      do 819 l=1,neflt
      i=id(l)
      j=jd(l)
      if(iss.lt.0)then
          qnode(iriver,i,j)=qnode(iriver,i,j)+efflt(1969,l)*1000.0d0
      else
          qnode(iriver,i,j)=qnode(iriver,i,j)+efflt(iyear,l)*1000.0d0
      endif
819   continue
```

```
c-------------------------------------------------------------------
c calculation of alpha and beta for finite difference solution
c-------------------------------------------------------------------
      do 940 i=2,imax+1
      do 940 j=2,jmax+1
      hhold(i,j)=h(i,j)
      hold(i,j)=h(i,j)
      factx(i,j)=(sfac*st(i,j)*rdelt+a(i,j)+b(i,j)+c(i,j)+d(i,j))
      facty(i,j)=sfac*st(i,j)*h(i,j)*rdelt
940   continue
c-------------------------------------------------------------------
c inclusion of leakage component into the finite difference equation
c-------------------------------------------------------------------
      do 3 iriver=1,nriver
      nnn=1
      nk=0
      do 35 in=1,2
      nk=nk+nr(in,iriver)
      if(in.eq.2)then
          ii=ir(2,nk,iriver)
          jj=jr(2,nk,iriver)
      endif
      do 36 l=nnn,nk
      i=ir(in,l,iriver)
      j=jr(in,l,iriver)
      rcnx=rconst(i,j)
      rcny=rconst(i,j)*g(i,j)
      factx(i,j)=factx(i,j)+rcnx
      facty(i,j)=facty(i,j)+rcny
36    continue
      nnn=nk+1
35    continue
3     continue
c-------------------------------------------------------------------
c iteration loop; max no of iterations for SOR
c-------------------------------------------------------------------
      if(iss.lt.0)ncycle=ncyss
      if(iss.gt.0)ncycle=ncyts
```

```
      issflag=0
      do 950 icycle=1,ncycle
      ind=0
      do 960 i=2,imax+1
      do 960 j=2,jmax+1
      hprev=h(i,j)
      if(hfix(i,j).ge.-1000.0d0.or.ibb(i,j).eq.0)goto 970
      ab=a(i,j)*h(i+1,j)+b(i,j)*h(i,j-1)+c(i,j)*h(i-1,j)+
     *   d(i,j)*h(i,j+1)+facty(i,j)+rs(i,j)
      abrr=abs(ab-factx(i,j)*hprev)
      if(abrr.lt.error)goto 980
      ind=100
980   h(i,j)=(1.0d0-ofac)*hprev+ofac*ab/factx(i,j)
      go to 960
970   h(i,j)=hfix(i,j)
960   continue
c-----------------------------------------------------------------------
c no of cycles to reach steady state is recorded issflag ensures that
c it is recorded only once after first convergence of S.O.R. routine.
c ind = 0 following statement 960 indicates convergence
c-----------------------------------------------------------------------
      if(iss.lt.0)then
         if(ind.eq.0.and.issflag.eq.0)then
            icyout=icycle
            issflag=1
         end if
         goto 950
      end if
      if(icycle.lt.2)goto 950
c-----------------------------------------------------------------------
c no of cycles to convergence for time variant conditions is recorded
c-----------------------------------------------------------------------
      if(ind.eq.0)then
        icyout=icycle
        goto 990
      end if
950   continue
      if(iss.gt.0)goto 1000
```

```
c-----------------------------------------------------------------
c output section for initial steady heads
c-----------------------------------------------------------------
      iss=100
      write(2,1010)
1010  format(1x,27hINITIAL STEADY STATE HEADS )
      call prin(h,7,2,mbound,2,nbound)
      if(cond_tran.eq.'cst')goto 1222
      call calz
      call perms
1222  continue
c-----------------------------------------------------------------
c calculation of flow to fixed heads for steady state
c-----------------------------------------------------------------
      qra2  = 0.0d0
      qff1  = 0.0d0
      qff2  = 0.0d0
      qfflx = 0.0d0
      qffly = 0.0d0
      reco  = 0.0d0
      reci  = 0.0d0
      delx=x(33)-x(32)
      do 4111 j=24,26
      dely=(y(j+1)-y(j-1))/2.0d0
      qff1=qff1+h(32,j)*ttx(32,j)*dely/delx
4111  continue
      delx=delx/2.0d0
      dely=y(23)-y(22)
      qff2=h(33,23)*tty(33,23)*delx/dely
      dely=y(27)-y(26)
      qff2=qff2+h(33,27)*tty(33,27)*delx/dely
c-----------------------------------------------------------------
c balancing river flow for steady state
c-----------------------------------------------------------------
      call flowr
      nkk=0
      do 911 iriver=1,nriver
      nnn=1
```

```
      nk=0
      do 317 in=1,2
      nk=nk+nr(in,iriver)
      do 318 l=nnn,nk
      i=ir(in,l,iriver)
      j=jr(in,l,iriver)
      qra2=qra2+qra(i,j)
318   continue
      nnn=nk+1
317   continue
911   continue
      do 14 k=1,nrech
      do 15 l=1,nnode(k)
      fa=fac1(k,l)
      i=ic(k,l)
      j=jc(k,l)
      if(i.eq.ihf(i,j).and.jb(i,j).ne.0)goto 15
      if(jbb(i,j).ne.0)then
      reci=reci+rchg(i,j)*qav(k)*fa
      endif
15    continue
14    continue
      qavb=0.0d0
      do 446 nw=1,nwell
446   qavb=qavb+qabsv(nw)
      qffl=qff1+qff2
      dif=reci-qffl-qra2
cGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
      call color(34,46)
      call cursor(0,0)
      call clear
      do 1462 line=1,8
1462  print*,' '
      print*,(char(0),i=1,21),char(219),(char(223),i=1,35),char(219)
      print*,(char(0),i=1,21),char(219),
     * '  Steady State Analysis Completed  ',char(219)
      print*,(char(0),i=1,21),char(219),(char(0),i=1,35),char(219)
      print733,(char(0),i=1,21),char(219),icyout,char(219)
```

```
733   format(' ',22a1,'          No of iterations:',i3,'        ',a1)
      print*,(char(0),i=1,21),char(219),(char(220),i=1,35),char(219)
      call f77_timer_delay(50000)
cGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
      write(28,*)'4'
      write(39,*)'2'
      write(29,*)'6'
      write(69,*)'6'
      write(51,*)'6'
      write(52,*)'4'
      go to 9000
1000  write(2,1020)ncycle
1020  format(' convergence not achieved in',i4,' iterations ')
      icyout=ncycle
c-----------------------------------------------------------------------
c end of sor routine
c-----------------------------------------------------------------------
990   continue
      if(cond_tran.eq.'cst')goto 938
      call calz
      do 791 k=1,nrech
      do 791 l=1,nnode(k)
      i=ic(k,l)
      j=jc(k,l)
      if(k.ne.1)then
          ttx(i,j)=alpk*px(i,j)*(base(i,j)+bass)
     *    +(2.-alpk)*px(i,j)*hx(i,j)
          tty(i,j)=alpk*py(i,j)*(base(i,j)+bass)
     *    +(2.-alpk)*py(i,j)*hy(i,j)
      endif
      st(i,j)=s(i,j)
791   continue
c-----------------------------------------------------------------------
c calculation of finite difference coefficients and modification
c of boundary node coefficients, no flow nodes are set to a flag
c-----------------------------------------------------------------------
      call fd(ttx,tty,st)
c-----------------------------------------------------------------------
```

```
c river flow balancing for transient state
c-----------------------------------------------------------------
938   continue
      call flowr
c-----------------------------------------------------------------
c calculating outflow to the fixed head nodes
c-----------------------------------------------------------------
      hfixfx = 0.0d0
      hfixfy = 0.0d0
      rreci = 0.0d0
      rreco = 0.0d0
      dinout = 0.0d0
      qrivaq = 0.0d0
c-----------------------------------------------------------------
c calculation of flow to fixed heads
c-----------------------------------------------------------------
      delx=x(33)-x(32)
      do 4112 j=24,26
      dely=(y(j+1)-y(j-1))/2.0d0
      hfixfx=hfixfx+h(32,j)*ttx(32,j)*dely/delx
4112  continue
      delx=delx/2.0d0
      dely=y(23)-y(22)
      hfixfy=h(33,23)*tty(33,23)*delx/dely
      dely=y(27)-y(26)
      hfixfy=hfixfy+h(33,27)*tty(33,27)*delx/dely
      hfixfw=hfixfx+hfixfy
c-----------------------------------------------------------------
c calculating the recharge into the system
c-----------------------------------------------------------------
      do 17 k=1,nrech
      do 18 l=1,nnode(k)
      fa=fac1(k,l)
      i=ic(k,l)
      j=jc(k,l)
      if(i.eq.ihf(i,j).and.jb(i,j).ne.0)goto 18
      if(jbb(i,j).ne.0)then
          rreci=rreci+rchg(i,j)*qfrech(k,iyear,imonth)*fa
```

```
      endif
18    continue
17    continue
c-----------------------------------------------------------------------
c calculating the total abstraction from the system
c-----------------------------------------------------------------------
      qab=0.0
      do nw=1,nwell
      qab=qab+qabs(nw,iyear,imonth)*qa
      end do
      qdis=0.0
      qabr=0.0
      do nd=1,ndis
      qdis=qdis+disc(iyear,nd)
      end do
      do nd=1,nab
      qabr=qabr+abir(iyear,nd)
      end do
      qab=qab+qdis+qabr
c-----------------------------------------------------------------------
c calculation of the total recharge and abstraction for fractioning
c-----------------------------------------------------------------------
      qabstv=qabstv+qab*delt
      qrechv=qrechv+rreci*delt
c-----------------------------------------------------------------------
c calculating net river-aquifer flow
c-----------------------------------------------------------------------
      do 91 iriver=1,nriver
      nnn=1
      nk=0
      do 37 in=1,2
      nk=nk+nr(in,iriver)
      do 38 l=nnn,nk
      i=ir(in,l,iriver)
      j=jr(in,l,iriver)
      qrivaq=qrivaq+qra(i,j)
38    continue
      nnn=nk+1
```

```
37     continue
91     continue
       dinout=rreci-qab-hfixfw-qrivaq
       write(2,3754)iday,imonth,dinout,rreci,qab,hfixfw,qrivaq
3754   format(/,'iday,imonth,dinout,rreci,qab,hfixfw,qrivaq:',/,2i3,
     * 1x,5f15.5)
cGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
       if(iss.gt.0)then
           call color(30,46)
           call cursor(0,0)
           print*,(char(0),i=1,25),char(219),(char(223),i=1,25),
     *     char(219)
       print*,(char(0),i=1,25),char(219),'    ',month(imonth),
     *    iyrcha,(char(0),i=1,5),char(219)
       print*,(char(0),i=1,25),char(219),(char(220),i=1,25),
     *    char(219)
       endif
       call color(31,46)
       if(iday.eq.1)then
               print*,(char(0),i=1,25),(char(219),i=1,9),
     *         (char(178),i=1,18)
           elseif(iday.eq.2)then
               print*,(char(0),i=1,25),(char(219),i=1,18),
     *         (char(178),i=1,9)
           else
               print*,(char(0),i=1,25),(char(219),i=1,27)
       end if
       call color(34,46)
       print*,(char(0),i=1,25),char(219),(char(223),i=1,25),char(219)
       print224,(char(0),i=1,25),char(219),qab,char(219)
       print223,(char(0),i=1,25),char(219),rreci,char(219)
       print225,(char(0),i=1,25),char(219),hfixfw,char(219)
       print*,(char(0),i=1,25),char(219),(char(220),i=1,25),char(219)
       call normal
       call color(34,46)
       print*,(char(0),i=1,23),char(219),(char(223),i=1,30),char(219)
       print222,(char(0),i=1,23),char(219),qrivaq,char(219)
223    format(' ',26a1,'    rech =',f10.1,'     ',a1)
```

```
224   format(' ',26a1,'    abst =',f10.1,'     ',a1)
225   format(' ',26a1,'    outf =',f10.1,'     ',a1)
cGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
c---------------------------------------------------------------------
c calculate change in the storage
c storch = storage change for time step
c dinout = inflow - outflow
c fbdisc = flow balance discrepancy
c---------------------------------------------------------------------
      storch=0.0d0
      do 21 i=imin,imax
      dddx=(x(i+1)-x(i-1))/2.0d0
      do 22 j=jmin,jmax
      dddy=(y(j+1)-y(j-1))/2.0d0
      if(jbb(i,j).eq.0)goto 22
      delh=h(i,j)-hhold(i,j)
      storch=storch+st(i,j)*delh*dddx*dddy/delt
 22   continue
 21   continue
      if(ifbd.eq.1)then
          fbdisc=dinout-storch
      else
          fbdisc=0.0
          ifbd=1
      endif
cGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
      print211,(char(0),i=1,23),char(219),dinout,char(219)
      print444,(char(0),i=1,23),char(219),storch,char(219)
      print445,(char(0),i=1,23),char(219),fbdisc,char(219)
      print448,(char(0),i=1,23),char(219),icyout,char(219)
      print*,(char(0),i=1,23),char(219),(char(220),i=1,30),char(219)
211   format(' ',24a1,'  In/Outflow    =',f10.1,'   ',a1)
222   format(' ',24a1,'  River/Aquifer =',f10.1,'   ',a1)
444   format(' ',24a1,'  Storage Chg.  =',f10.1,'   ',a1)
445   format(' ',24a1,'  Difference    =',f10.4,'   ',a1)
448   format(' ',24a1,'  No of Cycles  =',i6,'       ',a1)
cGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
c
```

```
c-----------------------------------------------------------------
c print out the result
c-----------------------------------------------------------------
        rivtw=rivtw+delt*qr(20,22)
        rivfm=rivfm+delt*qr(11,16)
        rivih=rivih+delt*qr(31,25)
        bastw=bastw+delt*qbase(20,22)
        basfm=basfm+delt*qbase(11,16)
        basih=basih+delt*qbase(31,25)
       write(29,119)idate,imonth,iyear-1900,h(17,9),h(22,9),
     *      h(20,13),h(20,18),h(6,14),h(8,14)
       write(69,119)idate,imonth,iyear-1900,h(17,24),h(19,21),
     *      h(7,10),h(9,18),h(11,10),h(11,20)
       write(51,119)idate,imonth,iyear-1900,h(26,11),h(29,22),
     *      h(23,9),h(25,19),h(24,16),h(23,14)
       write(52,119)idate,imonth,iyear-1900,h(22,25),h(13,12),
     *      h(29,16),h(13,22)
118    format(i2,'/',i2,'/',i2,2(1x,f7.2))
119    format(i2,'/',i2,'/',i2,6(1x,f7.2))
900    continue
c-----------------------------------------------------------------
c change units to 10 x Ml/d for monthly average flows
c-----------------------------------------------------------------
       rivtw=rivtw/(date*10000.0)
       rivfm=rivfm/(date*10000.0)
       rivih=rivih/(date*10000.0)
       bastw=bastw/(date*10000.0)
       basfm=basfm/(date*10000.0)
       basih=basih/(date*10000.0)
       if(bastw.lt.0)bastw=0.0
       if(basfm.lt.0)basfm=0.0
       if(basih.lt.0)basih=0.0
       write(28,119)idate,imonth,iyear-1900,rivtw,bastw,rivfm,basfm
       write(39,118)idate,imonth,iyear-1900,rivih,basih
       if(moscheck.eq.1)goto 730
c-----------------------------------------------------------------
c call to subroutine mosout to write S, T and head data to file for
c post-processing in a contouring routine
```

```
c-----------------------------------------------------------------------
      call mosout
730   continue
c-----------------------------------------------------------------------
c output of heads at end of each year
c-----------------------------------------------------------------------
      call prin(h,7,2,mbound,2,nbound)
      qaby=qaby+qabstv
      qrey=qrey+qrechv
700   continue
      iyold=iyear-1
c-----------------------------------------------------------------------
c output of terminator lines for time series data files
c-----------------------------------------------------------------------
      write(28,117)(one(lg),lg=1,2)
      write(39,117)(one(lg),lg=1,1)
      write(29,117)(one(lg),lg=1,6)
      write(69,117)(one(lg),lg=1,6)
      write(51,117)(one(lg),lg=1,6)
      write(52,117)(one(lg),lg=1,4)
      write(99,117)(one(lg),lg=1,8)
117   format('-1/-1/-1',16(1x,f7.2))
c-----------------------------------------------------------------------
c output dump routine
c-----------------------------------------------------------------------
      if(outputchk.eq.1)then
cGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
          call color(31,40)
          call clear
          call cursor(0,0)
          print*,' '
          print*,' '
          print*,' '
          print*,' '
          print*,' '
          print*,' '
          print*,' '
          print*,' '
```

```
          print*,' '
          print*,(char(0),i=1,15),char(219),(char(223),i=1,44),
     *    char(219)
          print*,(char(0),i=1,15),char(219),
     *    '      Output to dump file : ',
     *    outdump,(char(0),i=1,5),char(219)
          print*,(char(0),i=1,15),char(219),(char(220),i=1,44),
     *    char(219)
          print*,' '
          print*,' '
          print*,' '
          print*,' '
          print*,' '
          print*,' '
          print*,' '
cGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
          close(10)
          open(10,file=outdump,form='unformatted',status='unknown')
          rewind(10)
          write(10)cond_tran,cond,pass,passc,format,iyold,
     *    blk,month,monthp,input,output,rivdat,fldat,rivd,
     *    x,y,imx,jmy,ncyss,ncyts,h,a,b,c,d,s,qav,ihf,
     *    ib,jb,ibb,jbb,afact,bfact,qbb,qabsv,iyprt,
     *    st,cfact,dfact,ssfac,rcoefbc,rcoefc,sfac,g,
     *    tty,ttx,tday,catrbc,catrc,mc,one,ihf,rvc,qd,
     *    je,ist,jst,tx,ty,hfix,rchg,qr,qra,rconst,hx,hy,
     *    nkkk,qnode,px,py,z,m,n,min,nin,mbound,nbound,
     *    mfict,nfict,imin,jmin,imax,jmax,tri,ofac,error,
     *    imonth,iss,bass,date,nyr,tyinput,sinput,ie,
     *    hstart,sriv,hfixa,kday,iibeg,iiend,jjbeg,jjend,
     *    flownow,flowb4,rate,alpk,ntran,txinput,nc,icon
          close(10)
      endif
      call ring
      call clear
      call normal
      stop
      end
```

```
c----------------------------------------------------------------------
                                                SUB.F77
c----------------------------------------------------------------------
c---------------------------------------------------------------------
c Subroutine for printing out results
c---------------------------------------------------------------------
      SUBROUTINE PRIN(func,no,ibeg,iend,jbeg,jend)
c---------------------------------------------------------------------
c              no:                 parameter printed:
c               1                         tx
c               2                         ty
c               3                          s
c               4                    initial heads
c               5                        rchg
c               6                        hfix
c               7                          h
c---------------------------------------------------------------------
      implicit double precision(a-h,o-z)
      dimension func(40,40)
  100 format(10x,' X Transmissivities')
  101 format(10x,' Y Transmissivities')
  102 format(10x,' Storage factors')
  103 format(10x,' Initial values of head')
  104 format(10x,' Recharge values')
  105 format(10x,' Fixed heads')
  106 format(10x,' Head distribution')
  107 format(1x,i3,2x,1p13e9.2)
  153 format(1x,i3,2x,13f9.0)
  154 format(1x,i3,2x,13f9.5)
  108 format(1h )
  110 format(1x,i11,12(7x,i2))
  111 format(1x,i3,13f8.3)
  112 format(3x,13i8)
  115 format(1x,29h-1.00e+06 signifies free head,5x,34h-9.99e 1  is node
     1 outside boundary)
      if(no.eq.7) goto 6
      if(no.ne.1) goto 1
      write(2,100)
```

```
      goto 7
    1 if(no.ne.2) goto 2
      write(2,101)
      goto 7
    2 if(no.ne.3) goto 3
      write(2,102)
      goto 7
    3 if(no.ne.4) goto 4
      write(2,103)
      goto 7
    4 if(no.ne.5) goto 5
      write(2,104)
      goto 7
    5 if(no.ne.6) goto 6
      write(2,105)
      write(2,115)
      goto 7
6     write(2,106)
      ipbeg=ibeg
9     ipend=ipbeg+12
      if(ipend.gt.iend) then
      ipend=iend
      else if(ipend.lt.iend)then
      ipend=ipend
      endif
      write(2,112) (i,i=ipbeg,ipend)
      do 11 j=jbeg,jend
11    write(2,111)j,(func(i,j),i=ipbeg,ipend)
      write(2,108)
      ipbeg=ipend+1
      if(ipend.lt.iend) goto 9
      goto 10
7     ipbeg=ibeg
12    ipend=ipbeg+12
      if(ipend .gt. iend) then
      ipend=iend
      else if(ipend.lt.iend)then
      ipend=ipend
```

```
      endif
      write(2,110) (i,i=ipbeg,ipend)
      do 8 j=jbeg,jend
      if(no.le.2)then
           write(2,153)j,(func(i,j),i=ipbeg,ipend)
      elseif(no.eq.3)then
           write(2,154)j,(func(i,j),i=ipbeg,ipend)
      else
           write(2,107)j,(func(i,j),i=ipbeg,ipend)
      end if
8     continue
      write(2,108)
      ipbeg=ipend+1
      if(ipend .lt. iend) goto 12
10    write(2,108)
      return
      end
c-----------------------------------------------------------------------
c Subroutine for generating the coordinates of the system
c-----------------------------------------------------------------------
      SUBROUTINE CORD
      include 'common.inc'
      dimension dx(10),dy(10),mx(10),my(10)
c-----------------------------------------------------------------------
c x-length = xlen (34000)
c y-length = ylen (30000)
c-----------------------------------------------------------------------
      xlen=34000.0d0
      ylen=30000.0d0
      x(1)=-1000.0d0
      y(1)=-1000.0d0
      x(2)=0.0d0
      y(2)=0.0d0
300   kx=3
      ky=3
      ix=1
c-----------------------------------------------------------------------
c x-axis interval = dx()
```

```
c-----------------------------------------------------------------
20    dx(ix)=1000.0d0
c-----------------------------------------------------------------
c no of intervals in x direction = mx()
c-----------------------------------------------------------------
      mx(ix)=34
      do 11 ixx=1,mx(ix)
      x(kx)=x(kx-1)+dx(ix)
11    kx=kx+1
      if(x(kx-1).ge.xlen)goto 30
      ix=ix+1
      goto 20
30    iy=1
c-----------------------------------------------------------------
c y-axis interval = dy()
c-----------------------------------------------------------------
40    dy(iy)=1000.0d0
c-----------------------------------------------------------------
c no of intervals in the y direction = my()
c-----------------------------------------------------------------
      my(iy)=30
      do 50 iyy=1,my(iy)
      y(ky)=y(ky-1)+dy(iy)
50    ky=ky+1
      if(y(ky-1).ge.ylen)goto 60
      iy=iy+1
      goto 40
60    continue
      write(2,*)'x-axis coordinate'
      do 80 i=1,kx-1
80    write(2,70)x(i)
      write(2,*)'y-axis coordinate'
      do 90 j=1,ky-1
90    write(2,70)y(j)
70    format(f8.1)
      m=kx-4
      n=ky-4
      write(2,351)m-2,n-2
```

```
351   format(//,'No. of intervals in the x direction:',i2,
     * /,'No. of intervals in the y direction:',i2,//)
      return
      end
c-------------------------------------------------------------------
c Subroutine for calculating the factors for boundary nodes
c coefficients for finite difference equation
c-------------------------------------------------------------------
      SUBROUTINE FACT
      include 'common.inc'
      do 30 i=1,mfict
      do 40 j=1,nfict
      jb(i,j)    = 0
      ib(i,j)    = 0
      jbb(i,j)    = 0
      ibb(i,j)    = 0
      afact(i,j)  = 0.
      bfact(i,j)  = 0.
      cfact(i,j)  = 0.
      dfact(i,j)  = 0.
      ssfac(i,j)  = 0.
40    continue
30    continue
      imax=0
      jmax=0
      imin=99999
      jmin=99999
51    read(1,*)i,j
      if(i.eq.1.and.j.eq.1)goto 50
      ib(i,j)=i
      jb(i,j)=j
      if(ib(i,j).gt.imax)imax=ib(i,j)
      if(jb(i,j).gt.jmax)jmax=jb(i,j)
      if(ib(i,j).lt.imin)imin=ib(i,j)
      if(jb(i,j).lt.jmin)jmin=jb(i,j)
      goto 51
50    continue
      jmean=(jmin+jmax)/2
```

```
      imean=(imin+imax)/2
      do 70 i=imin,imean
      do 80 j=jmin,jmean
      if(jb(i,j).gt.0)then
      jbb(i,j)=j
      ibb(i,j)=i
      elseif(jbb(i-1,j).gt.0.and.jbb(i,j-1).gt.0)then
      ibb(i,j)=i
      jbb(i,j)=j
      endif
80    continue
70    continue
      do 71 i=imax,imean+1,-1
      do 81 j=jmin,jmean
      if(jb(i,j).gt.0)then
          jbb(i,j)=j
          ibb(i,j)=i
      elseif(ibb(i+1,j).gt.0.and.ibb(i,j-1).gt.0)then
          ibb(i,j)=i
          jbb(i,j)=j
      endif
81    continue
71    continue
      do 72 i=imin,imean
      do 82 j=jmax,jmean+1,-1
      if(jb(i,j).gt.0)then
         jbb(i,j)=j
         ibb(i,j)=i
      elseif(jbb(i-1,j).gt.0.and.jbb(i,j+1).gt.0)then
         ibb(i,j)=i
         jbb(i,j)=j
      endif
82    continue
72    continue
      do 73 i=imax,imean+1,-1
      do 83 j=jmax,jmean+1,-1
      if(jb(i,j).gt.0)then
            jbb(i,j)=j
```

```
            ibb(i,j)=i
      elseif(ibb(i+1,j).gt.0.and.ibb(i,j+1).gt.0)then
            ibb(i,j)=i
            jbb(i,j)=j
      endif
83    continue
73    continue
      do 10 j=jmin,jmax+1
      do 20 i=imin,imax+1
      if((i.eq.ib(i,j)).and.(j.eq.jb(i,j)))then
            if((i+1).eq.ib(i+1,j))then
                aa=0.5d0
            elseif((i+1).eq.ibb(i+1,j))then
                aa=1.0d0
            elseif(ibb(i+1,j).eq.0)then
                aa=0.0d0
      endif
      afact(i,j)=aa
      if((i-1).eq.ib(i-1,j))then
      cc=0.5d0
      elseif((i-1).eq.ibb(i-1,j))then
      cc=1.0d0
      elseif(ibb(i-1,j).eq.0)then
      cc=0.0d0
      endif
      cfact(i,j)=cc
      if((j+1).eq.jb(i,j+1))then
      dd=0.5d0
      elseif((j+1).eq.jbb(i,j+1))then
      dd=1.0d0
      elseif(jbb(i,j+1).eq.0)then
      dd=0.0d0
      endif
      dfact(i,j)=dd
      if((j-1).eq.jb(i,j-1))then
      bb=0.5d0
      elseif((j-1).eq.jbb(i,j-1))then
      bb=1.0d0
```

```
      elseif(jbb(i,j-1).eq.0)then
      bb=0.0d0
      endif
      bfact(i,j)=bb
      if( (aa.eq..5d0.and.bb.eq..5d0.and.cc.eq..0d0.and.dd.eq..0d0)
     *.or.(bb.eq..5d0.and.cc.eq..5d0.and.aa.eq..0d0.and.dd.eq..0d0)
     *.or.(cc.eq..5d0.and.dd.eq..5d0.and.aa.eq..0d0.and.bb.eq..0d0)
     *.or.(dd.eq..5d0.and.aa.eq..5d0.and.cc.eq..0d0.and.bb.eq..0d0)
     *       )then
      ss=0.25d0
      elseif( (aa.eq.0.5d0.and.cc.eq.0.5d0.and.dd.eq.1.0d0)
     *.or.    (bb.eq.0.5d0.and.dd.eq.0.5d0.and.aa.eq.1.0d0)
     *.or.    (bb.eq.0.5d0.and.dd.eq.0.5d0.and.cc.eq.1.0d0)
     *.or.    (aa.eq.0.5d0.and.cc.eq.0.5d0.and.bb.eq.1.0d0))then
      ss=0.5d0
      elseif( (aa.eq.1.0d0.and.bb.eq.1.0d0)
     *.or.    (bb.eq.1.0d0.and.cc.eq.1.0d0)
     *.or.    (cc.eq.1.0d0.and.dd.eq.1.0d0)
     *.or.    (dd.eq.1.0d0.and.aa.eq.1.0d0)) then
      ss=0.75d0
      endif
      ssfac(i,j)=ss
      endif
20    continue
10    continue
      return
      end
c---------------------------------------------------------------------
c Subroutine for calculating the recharge and storage coefficients
c for the whole system
c---------------------------------------------------------------------
      SUBROUTINE TRAN
c     implicit integer (i - n)
      include 'common.inc'
      dimension iibeg(60),iiend(60),jjbeg(60),jjend(60)
      do 1000 k=1,nrech
      ks=0
      ssum=0.0d0
```

```
      do 10 ii=1,nnode(k)
      i=ic(k,ii)
      j=jc(k,ii)
      if(ibb(i,j).eq.ib(i,j))then
          ssum=ssum+ssfac(i,j)
      elseif(ibb(i,j).ne.ib(i,j).and.ibb(i,j).ne.0)then
          ks=ks+1
      endif
10    continue
      ssum=ssum+real(ks)
      rssum=1.0d0/ssum
      do 30 ii=1,nnode(k)
      i=ic(k,ii)
      j=jc(k,ii)
      if(ibb(i,j).eq.ib(i,j).and.ssfac(i,j).eq.0.25d0)then
          rchg(i,j)=ssfac(i,j)*rssum
      elseif(ibb(i,j).eq.ib(i,j).and.ssfac(i,j).eq.0.5d0)then
          rchg(i,j)=ssfac(i,j)*rssum
      elseif(ibb(i,j).eq.ib(i,j).and.ssfac(i,j).eq.0.75d0)then
          rchg(i,j)=ssfac(i,j)*rssum
      elseif(ibb(i,j).ne.ib(i,j).and.ibb(i,j).gt.0)then
          rchg(i,j)=rssum
      endif
30    continue
1000  continue
c-------------------------------------------------------------------
c routine to read in t + s values in tabular form
c-------------------------------------------------------------------
        ntran=0
        nb(1)=33
        nb(2)=20
        nb(3)=7
        nb(4)=33
        nb(5)=20
        nb(6)=7
        ne(1)=21
        ne(2)=8
        ne(3)=2
```

```
          ne(4)=21
          ne(5)=8
          ne(6)=2
          do 60 m=1,6
          read(1,'(a)')adummy(m)
          write(2,'(a)')adummy(m)
          do 60 j=29,2,-1
          if(m.le.3)then
          read(1,*)kdummy,(tx(i,j),i=nb(m),ne(m),-1)
          do 55 i=nb(m),ne(m),-1
55        ty(i,j)=tx(i,j)
          write(2,3)j,(tx(i,j),i=nb(m),ne(m),-1)
          else
          read(1,*)kdummy,(s(i,j),i=nb(m),ne(m),-1)
          write(2,4)j,(s(i,j),i=nb(m),ne(m),-1)
          end if
60        continue
          do 70 m=1,3
          read(1,'(a)')adummy(m)
          do 70 j=29,2,-1
70        read(1,*)j,(base(i,j),i=nb(m),ne(m),-1)
3         format(i3,1x,13(f6.0))
4         format(i3,1x,13(1x,f6.5))
      return
      end
c-------------------------------------------------------------------
c Subroutine to calculate factors a,b,c,d, and boundary constants
c-------------------------------------------------------------------
      SUBROUTINE FD(txsub,tysub,ssub)
      include 'common.inc'
      dimension txsub(40,40),tysub(40,40),ssub(40,40)
      do 500 i= 2,imax+1
      do 500 j= 2,jmax+1
      a(i,j)=2.0d0*txsub(i,j)/((x(i+1)-x(i-1))*(x(i+1)-x(i)))
      c(i,j)=2.0d0*txsub(i-1,j)/((x(i+1)-x(i-1))*(x(i)-x(i-1)))
      b(i,j)=2.0d0*tysub(i,j-1)/((y(j+1)-y(j-1))*(y(j)-y(j-1)))
      d(i,j)=2.0d0*tysub(i,j)/((y(j+1)-y(j-1))*(y(j+1)-y(j)))
500   continue
```

```
      do 510 in=imin,imax
      do 511 jn=jmin,jmax
      if(ib(in,jn).eq.0)goto 511
      aa=afact(in,jn)
      bb=bfact(in,jn)
      cc=cfact(in,jn)
      dd=dfact(in,jn)
      ss=ssfac(in,jn)
      a(in,jn)=aa*a(in,jn)
      b(in,jn)=bb*b(in,jn)
      c(in,jn)=cc*c(in,jn)
      d(in,jn)=dd*d(in,jn)
c----------------------------------------------------------------------
c   -99.9 outside no flow boundary
c----------------------------------------------------------------------
      if(aa.le.0.000001d0) hfix(in+1,jn)=-99.9d0
      if(bb.le.0.000001d0) hfix(in,jn-1)=-99.9d0
      if(cc.le.0.000001d0) hfix(in-1,jn)=-99.9d0
      if(dd.le.0.000001d0) hfix(in,jn+1)=-99.9d0
      if(aa.le.0.000001d0) c(in+1,jn) = 0.0d0
      if(bb.le.0.000001d0) d(in,jn-1) = 0.0d0
      if(cc.le.0.000001d0) a(in-1,jn) = 0.0d0
      if(dd.le.0.000001d0) b(in,jn+1) = 0.0d0
      ssub(in,jn)=ss*ssub(in,jn)
511   continue
510   continue
      return
      end
c----------------------------------------------------------------------
c Subroutine to calculate z()
c----------------------------------------------------------------------
      SUBROUTINE CALZ
      include 'common.inc'
      do 10 nh=1,nrech
c     print*,nnode(nh)
      do 20 nd=1,nnode(nh)
      i=ic(nh,nd)
      j=jc(nh,nd)
```

```
      delh=h(i,j)-base(i,j)-bass
      if(delh.le.0.0d0)then
      z(i,j)=0.0d0
      else
      z(i,j)=delh
      endif
20    continue
10    continue
      do 30 nh=1,nrech
      do 30 nd=1,nnode(nh)
      i=ic(nh,nd)
      j=jc(nh,nd)
      hx(i,j)=(z(i+1,j)+z(i,j))*0.5d0
30    hy(i,j)=(z(i,j+1)+z(i,j))*0.5d0
      return
      end
c-----------------------------------------------------------------
c Subroutine to calculate permeability
c-----------------------------------------------------------------
      SUBROUTINE PERMS
      include 'common.inc'
      do 10 nor=1,nrech
      do 10 nd=1,nnode(nor)
      i=ic(nor,nd)
      j=jc(nor,nd)
      px(i,j)=0.0d0
      py(i,j)=0.0d0
      px(i,j)=ttx(i,j)/(base(i,j)+bass)
      py(i,j)=tty(i,j)/(base(i,j)+bass)
10    continue
      return
      end
c-------------------------------------------------------
c Subroutine for the river-aquifer flow calculation
c-------------------------------------------------------
      SUBROUTINE FLOWR
      include 'common.inc'
      dimension dummy(40,40)
```

```
c.............................................................
      nkk=0
c-------------------------------------------------------------
c output to files 'bcflow.dat' and 'river.flo' (99)
c-------------------------------------------------------------
      iyold=iyear
      do 1212 i=1,40
      do 1212 j=1,40
      qbase(i,j)=0.0
1212  continue
      if(irivno.gt.1.and.irivno.lt.nriver)then
          ii=ir(2,nk,irivno-1)
          jj=jr(2,nk,irivno-1)
          dummy(ii,jj)=qr(ii,jj)
      endif
      n1=0
      do 10 ichbc=1,2
10    n1=n1+nr(ichbc,irivno)
      nodetot=0
      nodebeg=1
c-------------------------------------------------------------
c calculating the river-aquifer flow exchanges
c qr() = total river flow at node
c qnode() = runoff component for the current node (input data)
c qra() = river-aquifer leakage flow
c qbaseb4 = baseflow at node immediately upsteam of node i,j
c-------------------------------------------------------------
      qbaseb4=0.0
      do 35 ichbc=1,2
      nodetot=nodetot+nr(ichbc,irivno)
      do 45 node=nodebeg,nodetot
      i=ir(ichbc,node,irivno)
      j=jr(ichbc,node,irivno)
      hh=0.5*(hhold(i,j)+h(i,j))
      qra(i,j)=rconst(i,j)*(h(i,j)-g(i,j))
     *    *(x(i+1)-x(i-1))*(y(j+1)-y(j-1))/4.0d0
25    if((ichbc.eq.1).and.(node.eq.1))goto 51
      ik=ir(ichbc,node-1,irivno)
```

```
      if(node.eq.(nr(1,irivno)+1))ik=ir(ichbc-1,node-1,irivno)
      jk=jr(ichbc,node-1,irivno)
      if(node.eq.(nr(1,irivno)+1))jk=jr(ichbc-1,node-1,irivno)
      goto 52
51    ik=ir(ichbc,node,irivno)
      jk=jr(ichbc,node,irivno)
52    continue
c-----------------------------------------------------------------
c total baseflow calculation for each node
c-----------------------------------------------------------------
      qbase(i,j)=qbaseb4+qra(i,j)
      if(qbase(i,j).lt.0)qbase(i,j)=0.0
      qbaseb4=qbase(i,j)
c-----------------------------------------------------------------
c conditional call to routine for balancing the flow
c
c 'if then': irivno is not the no. of cut-off channel
c-----------------------------------------------------------------
      if(irivno.lt.nriver)then
          if((ichbc.eq.1).and.(node.eq.1))then
              qr(i,j)=qnode(irivno,i,j)
              call bal(i,j,irivno,node,n1)
      else
              qr(i,j)=qr(ik,jk)+qnode(irivno,i,j)+dummy(i,j)
              call bal(i,j,irivno,node,n1)
              dummy(i,j)=0.0d0
      endif
c-----------------------------------------------------------------
c 'else': irivno is no. for cut-off channel (irivno=11)
c-----------------------------------------------------------------
      else
          if((ichbc.eq.1).and.(node.eq.1))then
              qr(i,j)=qnode(irivno,i,j)+(1.0-rate)*qr(23,23)/rate
              call bal(i,j,irivno,node,n1)
          else
              qr(i,j)=qr(ik,jk)+qnode(irivno,i,j)
              call bal(i,j,irivno,node,n1)
          endif
```

```
      endif
      if(imonth.ge.2.and.iyear.ne.1983)goto 45
1439  format(19x,2f15.2,/)
45    continue
      nodebeg=nodetot+1
35    continue
5     continue
      return
      end
c-----------------------------------------------------------------
c Subroutine for balancing the flow
c-----------------------------------------------------------------
      SUBROUTINE BAL(ibal,jbal,irivno,node,n1)
      include 'common.inc'
      if(irivno.lt.(nriver-1).and.node.eq.n1)then
          qra(ibal,jbal)=0.0d0
          goto 15
      endif
      qleak=qra(ibal,jbal)
      if(qleak.lt.0.and.abs(qleak).ge.qr(ibal,jbal))then
         qr(ibal,jbal)=0.0d0
         nkk=nkk+1
      else
         qr(ibal,jbal)=qr(ibal,jbal)+qleak
      endif
15    continue
      if(irivno.eq.nriver-1)qr(23,23)=rate*qr(23,23)
      return
      end
c--------------------------------------------------------------------------
c Subroutine for opening files
c--------------------------------------------------------------------------
      SUBROUTINE FILE_OPEN
      include 'common.inc'
      call color(8,40)
      close(89)
      close(1)
      close(2)
```

```
      close(3)
      close(7)
      close(8)
      close(9)
      close(28)
      close(29)
      close(39)
      close(51)
      close(52)
      close(69)
      close(99)
      open(89,file='grdwo.r',status='unknown',iostat=ios)
      rewind 89
      read(89,*)yrstart,yrstop,mstart,mstop
      read(89,*)inputchk,outputchk
      read(89,'(a)')indump
      read(89,'(a)')outdump
      read(89,'(a)')input
      read(89,'(a)')output
      read(89,'(a)')rivdat
      read(89,'(a)')fldat
      read(89,'(a)')nodesdat
      read(89,'(a)')absdat
      read(89,'(a)')rechdat
      read(89,'(a)')industdat
c---------------------------------------------------------------------
c file opening statements
c---------------------------------------------------------------------
      open(1,file=input,status='unknown',iostat=ios)
      open(2,file=output,status='unknown',form='formatted')
      open(3,file=rivdat,status='unknown',iostat=ios)
      open(7,file=nodesdat,status='unknown',iostat=ios)
      open(8,file=absdat,status='unknown',iostat=ios)
      open(9,file=rechdat,status='unknown',iostat=ios)
      open(28,file='temple.dat',status='unknown',form='formatted')
      open(39,file='isleham.dat',status='unknown',form='formatted')
      open(29,file='head.dat',status='unknown',form='formatted')
      open(69,file='head1.dat',status='unknown',form='formatted')
```

```
      open(51,file='head2.dat',status='unknown',form='formatted')
      open(52,file='head3.dat',status='unknown',form='formatted')
      open(99,file=industdat,status='unknown',iostat=ios)
      rewind(1)
      rewind(2)
      rewind(3)
      rewind(7)
      rewind(8)
      rewind(9)
      rewind(28)
      rewind(29)
      rewind(39)
      rewind(51)
      rewind(52)
      rewind(69)
      rewind(99)
      return
      end
c-------------------------------------------------------------------
c Subroutine to read river_data
c-------------------------------------------------------------------
      SUBROUTINE RIVER_DATA(cat,cat1,runct,runct1)
      include 'common.inc'
      dimension runct(15,35,30),runct1(15,35,30),cat1(15),cat(15)
      read(3,*)nriver
      do 8 irivno=1,nriver
      read(3,'(a)')river(irivno)
      read(3,*)cat(irivno),cat1(irivno)
      do 10 ichbc=1,2
c-------------------------------------------------------------------
c nr() = no. of nodes in boulder clay [nr(1,?)], chalk [nr(2,?)]
c-------------------------------------------------------------------
      read(3,*)nr(ichbc,irivno)
10    continue
8     continue
      do 7 irivno=1,nriver
      nodebeg=1
      nodetot=0
```

```
      write(2,1561)river(irivno)
1561  format(2x,a)
      do 11 ichbc=1,2
      nodetot=nodetot+nr(ichbc,irivno)
      if(ichbc.eq.1)write(2,*)' Boulder Clay river nodes'
      if(ichbc.eq.2)write(2,*)' Chalk river nodes'
      do 9 node=nodebeg,nodetot
      read(3,*)ir(ichbc,node,irivno),jr(ichbc,node,irivno),
     *stage,rivc,frbc,frc
      i=ir(ichbc,node,irivno)
      j=jr(ichbc,node,irivno)
      write(2,20)i,j,stage,rivc,frbc,frc
20    format(2i4,6f12.6)
      runct1(irivno,i,j)=frc
      runct(irivno,i,j)=frbc
      rconst(i,j)=rivc
9     g(i,j)=stage
      nodebeg=nodetot+1
11    continue
7     continue
      return
      end
c-----------------------------------------------------------------
c Subroutine for reading recharge and abstraction and runoff data
c-----------------------------------------------------------------
      SUBROUTINE RECABSENT
      include 'common.inc'
c-----------------------------------------------------------------
c qfrech() = recharge
c qabs() = PWS abstraction
c runoff() and runofc() = runoff data.
c nrech = number of recharge blocks
c-----------------------------------------------------------------
      write(2,214)
214   format(//,10x,' Abstraction data',//)
      write(2,270)
270   format(34h Abstraction wells and their nodes,//)
      write(2,275)
```

```
275   format(1x,'  I    J  Pumped wells')
      read(8,*)nwell
      do 280 nw = 1,nwell
      read(8,fmt=*)iw(nw),jw(nw)
      read(8,fmt='(a)')well(nw)
280   write(2,290)iw(nw),jw(nw),well(nw)
290   format(2i4,2x,a)
      do 581 iblock=nfirst,nfinal
      write(2,561) iblock
561   format(1x,'  YEAR:',i5)
      do 581 imonth = 1,12
      do 588 nw=1,nwell
588   qabs(nw,iblock,imonth)=0.0
      read(8,fmt=*,end=1667)(qabs(nw,iblock,imonth),nw=1,nwell)
      write(2,571)(qabs(nw,iblock,imonth),nw=1,nwell)
571   format(15f7.1)
      do 589 nw=1,nwell
589   qabs(nw,iblock,imonth)=qabs(nw,iblock,imonth)*1000.0
581   continue
1667  continue
      close(8)
c-------------------------------------------------------------------
c reading of recharge data for each month
c nday() = days in the month
c qfrech() = monthly recharge rate [Ml/d] for each recharge block
c-------------------------------------------------------------------
      write(2,215)
 215  format(//,' Recharge data      Boulder runoff',/)
      do 580 iblock = nfirst,nfinal
      write(2,560) iblock
560   format(1x,'  YEAR:',i5)
      do 580 imonth = 1,12
      runofc(iblock,imonth)=0.0d0
      runoff(iblock,imonth)=0.0d0
      do 765 nnr = 1,nrech
765   qfrech(nnr,iblock,imonth) = 0.0d0
      read(9,*,end=1669) nday(iblock,imonth),(qfrech(nnr,iblock,imonth)
     *        ,nnr=1,nrech),runoff(iblock,imonth)
```

```
     *       ,runofc(iblock,imonth)
      write(2,570)nday(iblock,imonth),(qfrech(nnr,iblock,imonth),
     *    nnr=1,nrech),runoff(iblock,imonth)
570   format(i5,10f7.1)
c-----------------------------------------------------------------------
c convert input values of Ml/d to m**3/d
c-----------------------------------------------------------------------
      runofc(iblock,imonth)=runofc(iblock,imonth)*1000.0d0
      runoff(iblock,imonth)=runoff(iblock,imonth)*1000.0d0
      do 580 nnr=1,nrech
580   qfrech(nnr,iblock,imonth) = qfrech(nnr,iblock,imonth)*1000.0d0
1669  continue
      close(9)
      return
      end
c-----------------------------------------------------------------------
c Subroutine for entering abstraction data
c-----------------------------------------------------------------------
      SUBROUTINE IRIGINDENT
      include 'common.inc'
c-----------------------------------------------------------------------
c nfirst = first year of available flow data to be entered
c nfinal = final year of available flow data to be entered
c-----------------------------------------------------------------------
      read(7,*) nfirst,nfinal,nrech
      do 1558 nnr=1,nrech
      read(7,*)nnode(nnr)
      do 1559 nod=1,nnode(nnr)
      read(7,*)ic(nnr,nod),jc(nnr,nod),fac1(nnr,nod)
1559  continue
1558  continue
c-----------------------------------------------------------------------
c input of discharge into the river.
c neflt = no of discharge nodes
c id(),jd() = discharge nodes
c efflt() = discharge.
c ndummy = not used in model (only used in data preparation program)
c-----------------------------------------------------------------------
```

```
      read(99,*)ndummy
      read(99,*)neflt
      do 4441 nef=1,neflt
      read(99,*)id(nef),jd(nef)
4441  continue
      write(2,fmt=*)nfirst,nfinal
      do 444 iblock=nfirst,nfinal
      do 445 i=1,neflt
445   efflt(iblock,i)=0.0d0
      read(99,fmt=*)(efflt(iblock,j),j=1,neflt)
444   write(2,2644)iblock,(efflt(iblock,j),j=1,neflt)
2644  format(i5,10f6.2)
c-----------------------------------------------------------------
c input of industrial abstraction using industrial discharge format
c-----------------------------------------------------------------
      read(99,*)ndis
      do 4471 nd=1,ndis
4471  read(99,*)ia(nd),ja(nd)
      do 4461 iblock=nfirst,nfinal
      do 448 nd=1,ndis
448   disc(iblock,nd)=0.0d0
4461  read(99,*)(disc(iblock,j),j=1,ndis)
c-----------------------------------------------------------------
c input of irrigation abstraction using industrial discharge format
c-----------------------------------------------------------------
      read(99,*)nab
      do 4472 nd=1,nab
4472  read(99,*)iar(nd),jar(nd)
      do 4462 iblock=nfirst,nfinal
      do 449 nd=1,nab
449   abir(iblock,nd)=0.0d0
      read(99,*)(abir(iblock,j),j=1,nab)
4462  continue
      read(99,*)qa
c-----------------------------------------------------------------
c following the input of industrial abstraction data, unit 99 is used
c for output of river flow data in subroutine flowr
c-----------------------------------------------------------------
```

```
      close(99)
      open(99,file='river.flo',status='unknown',form='formatted')
      rewind 99
      return
      end
c------------------------------------------------------------------------
c output of parameters for contouring routines
c------------------------------------------------------------------------
      SUBROUTINE MOSOUT
      include 'common.inc'
      close(10)
      open(10,file='mos.dat',status='unknown',iostat=ios)
      rewind(10)
      write(10,*)mbound,nbound
      do 143 j=2,nbound
143   write(10,342)j,(base(i,j),i=2,mbound)
      write(10,*)mbound,nbound
      do 144 j=2,nbound
144   write(10,342)j,(h(i,j),i=2,mbound)
      write(10,*)mbound,nbound
      do 145 j=2,nbound
145   write(10,343)j,(ttx(i,j),i=2,mbound)
      write(10,*)mbound,nbound
      do 146 j=2,nbound
146   write(10,344)j,(st(i,j),i=2,mbound)
      moscheck=1
c------------------------------------------------------------------------
c format statements: 342 - base [base(i,j)], heads [h(i,j)]
c       343 - transmissivity [ttx/tty(i,j)]
c       344 - storage coefficients [st(i,j)]
c------------------------------------------------------------------------
342   format(i3,10f7.3/3x,10f7.3/3x,10f7.3/3x,10f7.3)
343   format(i3,10f7.1/3x,10f7.1/3x,10f7.1/3x,10f7.1)
344   format(i3,10f7.5/3x,10f7.5/3x,10f7.5/3x,10f7.5)
      return
      end
```

```
c-------------------------------------------------------------------
                                   UTILITY.F77
c-------------------------------------------------------------------


c-------------------------------------------------------------------
c Escape sequence ANSI keyboard system used. Reference DOS manual
c subroutine for clearing the screen
c-------------------------------------------------------------------
      SUBROUTINE CLEAR
      implicit double precision(a-h,o-z)
      character list*4
      list(1:1)=char(27)
      list(2:4)='[2J'
      print*,list
      return
      end
c-------------------------------------------------------------------
c Subroutine for colour display of output on the screen.
c The same ANSI Escape sequence keyboard is used. The number
c for both the foreground and background have to be chosen. The
c format is ESC,[31,40m =red foreground and black background
c-------------------------------------------------------------------
      SUBROUTINE COLOR(ifrt,ibac)
      implicit double precision(a-h,o-z)
      logical iflt10, iblt10
      iflt10=ifrt.lt.10
      iblt10=ibac.lt.10
      if(iflt10 .and. iblt10)then
      assign 100 to nform
      else if(iblt10)then
      assign 110 to nform
      else if (iflt10) then
      assign 120 to nform
      else
      assign 130 to nform
      endif
      printnform,char(27),ifrt,ibac
```

```
      return
100   format(a1,'[',i1,';',i1,'m')
110   format(a1,'[',i2,';',i1,'m')
120   format(a1,'[',i1,';',i2,'m')
130   format(a1,'[',i2,';',i2,'m')
      end
c-----------------------------------------------------------------
c This subroutine is for CGA graphic card for resolution 640 by 200
c pixels which is equivalent to 16K RAM. The bufer is cleared and the
c BIOS(basic input and output system video driver is 10h or 16 decimal.
c The call to the PC processor interrupt 21h or 33 decimal with segment
c set .false. neglects the use of dosblock vector and thus the video
c mode can be chosen. Consult Parallel Fortran for the DOS.INC to understand
c the method of accessing the registers of the PC processor.
c-----------------------------------------------------------------
      SUBROUTINE Video_Mode (Mode)
      Integer Mode
      Include 'Dos.Inc'
      Integer CGABuf (0:4095)
      Common /CGA_Common/ CGABuf
      Integer I, DOSBlock (F77_DOS_Block_Size), Dont_Clear_Display
      Integer Video_Set_Mode, BIOS_Video_Driver
      Parameter (Video_Set_Mode=0, BIOS_Video_Driver=16,
     *   Dont_Clear_Display=128)
      If (IAnd (Mode, Dont_Clear_Display) .eq. 0) then
         Do I = 0, 4095
            CGABuf (I) = 0
         End Do
      End If
      DosBlock (F77_DOS_AX) = IOr (IShft(Video_Set_Mode,8), Mode)
      Call F77_Host_Interrupt (BIOS_Video_Driver, .false., DOSBlock)
      return
      End
c-----------------------------------------------------------------
c  Subroutine cursor to position the cursor at the required point
c  on the screen. ANSI escape sequence keyboard used.
c-----------------------------------------------------------------
      SUBROUTINE CURSOR(irow,icol)
```

```
      implicit double precision(a-h,o-z)
      character vary*22
      common/word/vary
      logical iclt10, irlt10
      iclt10=icol.lt.10
      irlt10=irow.lt.10
      if(iclt10 .and. irlt10)then
      assign 100 to nform
      nl=7
      else if(iclt10)then
      assign 110 to nform
      nl=8
      else if (irlt10) then
      assign 120 to nform
      nl=8
      else
      assign 130 to nform
      nl=9
      endif
      printnform,char(27),irow,icol
      return
100   format(a1,'[',i1,';',i1,'H',a)
110   format(a1,'[',i2,';',i1,'H',a)
120   format(a1,'[',i1,';',i2,'H',a)
130   format(a1,'[',i2,';',i2,'H',a)
      end
c---------------------------------------------------------------
c  Subroutine to normalise the screen
c  Same as for SUBROUTINE CURSOR.
c---------------------------------------------------------------
      SUBROUTINE NORMAL
      character list*4
      list(1:1)=char(27)
      list(2:4)='['//char(48)//'m'
      print*,list
      return
      end
c---------------------------------------------------------------
```

```
c This SUBROUTINE produces a tone and is called to alert the
c programmer at the end of simulation.
c---------------------------------------------------------------
      SUBROUTINE RING
      include 'dos.inc'
      include 'timer.inc'
      integer dosblock(f77_dos_block_size),ticks
      data ticks/220/
      do 10 l=1,2
      call clear
      do 10 i=1,5
10    print*,char(7)
      call clear
      return
      end
```

Appendix D

Lark Chalk Model Variable List

The variables used in the Lark model program are based on the metre-day system. The more important arrays and variables used in the model are defined in this appendix:

VARIABLES:

ABSDAT:	Abstraction data file name.
ALPK,BASS:	Factor used in transmissivity saturated depth relationship
BASBB:	Modelled baseflow at Beck Bridge gauging station [Ml*10]
BASFM:	Modelled baseflow at Fornham gauging station [Ml*10]
BASIH:	Modelled baseflow at Isleham gauging station [Ml*10]
BASTW:	Modelled baseflow at Temple Weir gauging station [Ml*10]
COND_TRAN:	Flag used to determine type of simulation - COND_TRAN = cst for constant transmissivity COND_TRAN = vtm for variable transmissivity
DATE:	Day number in month used in finding time step lengths. [d]
DELT:	Length of time steps in days. [d]
DELH:	Primary depth.
DINOUT:	Difference between inflows and outflows over time step. [cu. m/d]
ERROR:	The error criterion applied to heads at each node.
FBDISC:	Flow balance discrepancy

FLOW12B:
FLOW3B:
FLOW4B:
FLOW5B:
FLOW67B:
FLOW8B: — Spot gauging sites where flows are simulated.

FLOWNOW,FLOWB4: Runoff factors - current, previous time step

HSTART: An arbitrary value used only once to initialize the groundwater head throughout the aquifer before first SOR calculation of simulation. [m]

HFIXFW:
HFIXFX:
HFIXFY: — Flow to fixed heads.

I,J: Counters for coordinates in x and y directions respectively.

IBLOCK, ICYCLE, IDAY, IMONTH, IYEAR:
Counters used in program for flow data block number; cycle number in SOR; and day, month and year of simulation respectively.

IND: If IND= 100, then the head given at a node for a particular iteration does not satisfy the error criterion and SOR must be repeated.If IND = 0, SOR has completed successfully.

ISSFLAG: ISSFLAG = -100 before steady-state has been reached. Thereafter, ISSFLAG = 100.

ICHBC: Flag = 1 for Boulder Clay, = 2 for Chalk.

ICYOUT: No. of cycles to convergence for time step.

INDUMP: Input dump file.

INDUSTDAT: Industrial data file.

IRIVNO: Tributary number (from 1 to 11)

IMAX,IMEAN,IMIN,JMAX,JMEAN,JMIN:
Variables used in describing the extent of the aquifer in the I and J directions.

LDAY:	Total number of time steps in a month.
MIN:	= M+1
MBOUND:	= M+2
MFICT:	= M+3
MSTART:	First month of simulation in year YRSTART.
MSTOP:	Last month of simulation in year YRSTOP.
M,N:	Number of intervals on the finite difference in the I (x) and J (y) directions respectively.
NCYSS:	Maximum number of iterations permissable during steady-state calculation.
NCYCLE:	Maximum number of cycles allowed in a single run of the SOR routine.
NCYTS:	Maximum number of cycles allowed in a time step for nonsteady conditions.
NAB:	No. of abstraction nodes.
NIN:	= N+1
NBOUND:	= N+2
NFICT:	= N+3
NDIS:	No. of industrial abstraction nodes.
NEFLT:	No of industrial discharge sites
NFINAL:	Final year of available flow data to be entered
NFIRST:	First year of available flow data to be entered
NRECH:	Number of recharge blocks
NRIVER:	No of rivers
NWELL:	No. of PWS sites.
OUTDUMP:	Name of output dump file.
OUTPUT:	Name of output data file.
OFAC:	Over-relaxation factor.

QA:	Optional abstraction flow factor.
QAB:	Total average abstraction flow factor.
QBASEB4:	Baseflow at node immediately upsteam of node I,J
QFF1:	Variables used in calculating flows to fixed heads.
QFF2:	
QFFL:	
QFFLX:	
QFFLY:	
QLEAK:	
QRIVAQ:	Total river/aquifer flow for month.
RATE:	Allowance made for cut-off channel water outflow
RIVDAT:	River data file.
RIVFM:	Modelled river flow at Fornham gauging station [Ml*10]
RIVIH:	Modelled river flow at Isleham gauging station [Ml*10]
RIVTW:	Modelled river flow at Temple Weir gauging station [Ml*10]
RIVBB:	Modelled river flow at Beck Bridge gauging station [Ml*10]
RRECI:	Total average recharge for time step.
SFAC:	Factor used to set storage coefficients at each node to small value for steady-state calculation. After steady-state has been reached, SFAC = 1.0.
SRIV:	Uniform factor applied to all river node storage coefficient used for testing purposes.
STORCH:	Storage change for time step
TOTROBC:	Total Boulder Clay runoff
TOTROC:	Total Chalk runoff
TRI:	Uniform factor applied to all river node permeabilities used for testing purposes.
XLEN:	Full extent of grid in x direction (34000m)
YLEN:	Full extent of grid in y direction (30000m)

YRSTART: First year of simulation.
YRSTOP: Last year of simulation.

ARRAYS:

A(I,J), B(I,J), C(I,J), D(I,J):
Factors used to set no-flow boundaries. [d^{-1}]

ABIR(): Irrigation abstraction

AFACT(),BFACT(),CFACT(),DFACT():
Nodal boundary factor.

BASE(I,J): Aquifer base level. [m]

CAT(I,J): Fraction of catchment runoff for Chalk node (I,J).

CAT1(I,J): Fraction of catchment runoff for Boulder Clay node (I,J).

CATRBC(): Proportion of Boulder Clay runoff for catchment.

CATRC(): Proportion of Chalk catchment runoff for catchment.

DISC(): Industrial abstraction.

DX(): Mesh interval in X/I direction.

DY(): Mesh interval in Y/J direction.

EFFLT(): Industrial discharge to river

FACTX():] Arrays used in finite
FACTY():] difference calculations.

FUNC(): Dummy array used in the "prin" subroutine

G(I,J): River stage at streamflow node.

H(I,J):	Current groundwater elevation. [m]
HFIX(I,J):	Fixed head. [m]
HOLD(I,J):	Head used in previous iteration of SOR routine. [m]
HPREV(I,J):	Head at the end of last time step. [m]
HX():	Saturated depth for calculating TX().
HY():	Saturated depth for calculating TY().
ID(),JD():	Discharge nodes
IR(), JR()	River node coordinates
IST(), JST():	Start of fixed node block
IE(), JE():	End of fixed nodes
IW(),JW():	PWS abstraction node coordinates.
MONTH():	Month (character format)
MONTHP():	Month for print out (request)
MY():	No of intervals in the y direction
MX():	No of intervals in x direction
NR():	No. of nodes in boulder clay [nr(1,)], chalk [nr(2,)]
NDAY(IBLOCK,IMONTH):	Number of days in month. [d]
NFLOW(NF):	Number of nodes to which an inflow or outflow data block is distributed.
PX(I,J), PY(I,J):	Permeabilities in I and J directions at a node [m/d]
QAV(NF):	Average monthly flow for component NF, calculated from first block. These flows are used in the steady-state calculations. [cu. m/d]
QABS():	PWS abstraction [cu. m/d]
QFRECH():	Monthly recharge rate [Ml/d] for each recharge block
QNODE():	Runoff component for the current node (input data)
QR():	Total river flow at node
QRA():	River-aquifer leakage flow
QBASE():	Nodal baseflow.

RCHG(I,J):	Recharge fraction at node.
RS(I,J):	Total monthly flows at node from entered flow data block. This excludes flows at node determined by the model, such as leakages. [m/d]
RCOEFBC():	Proportion of Boulder Clay runoff for catchment
RCOEFC():	Proportion of Chalk catchment runoff for catchment
RCONST():	Leakage factor
RIVER(FLAG):	Rivername
RUNOFF(),RUNOFC():	Boulder Clay, Chalk runoff input data.
RUNCT():	Fraction of catchment runoff for Chalk node (I,J).
RUNCT1():	Fraction of catchment runoff for Boulder Clay node (I,J).
S(I,J):	Factor applied to nodal area used in defining aquifer boundaries. This factor affects both storage and recharge and is determined during steady-state calculations. It remains constant thereafter.
ST(I,J):	Storage coefficient used in time step.
TDAY(K):	Day number in month at which calculation is performed. [d]
TTX(): TTY():	Entered transmissivities in the x and y directions.
TX(I,J), TY(I,J):	Current transmissivities in I and J direction at node. [sq. m/d]
TX1(I,J), TY1(I,J):	Transmissivities in I and J direction at node before most recent SOR calculation. [sq. m/d]

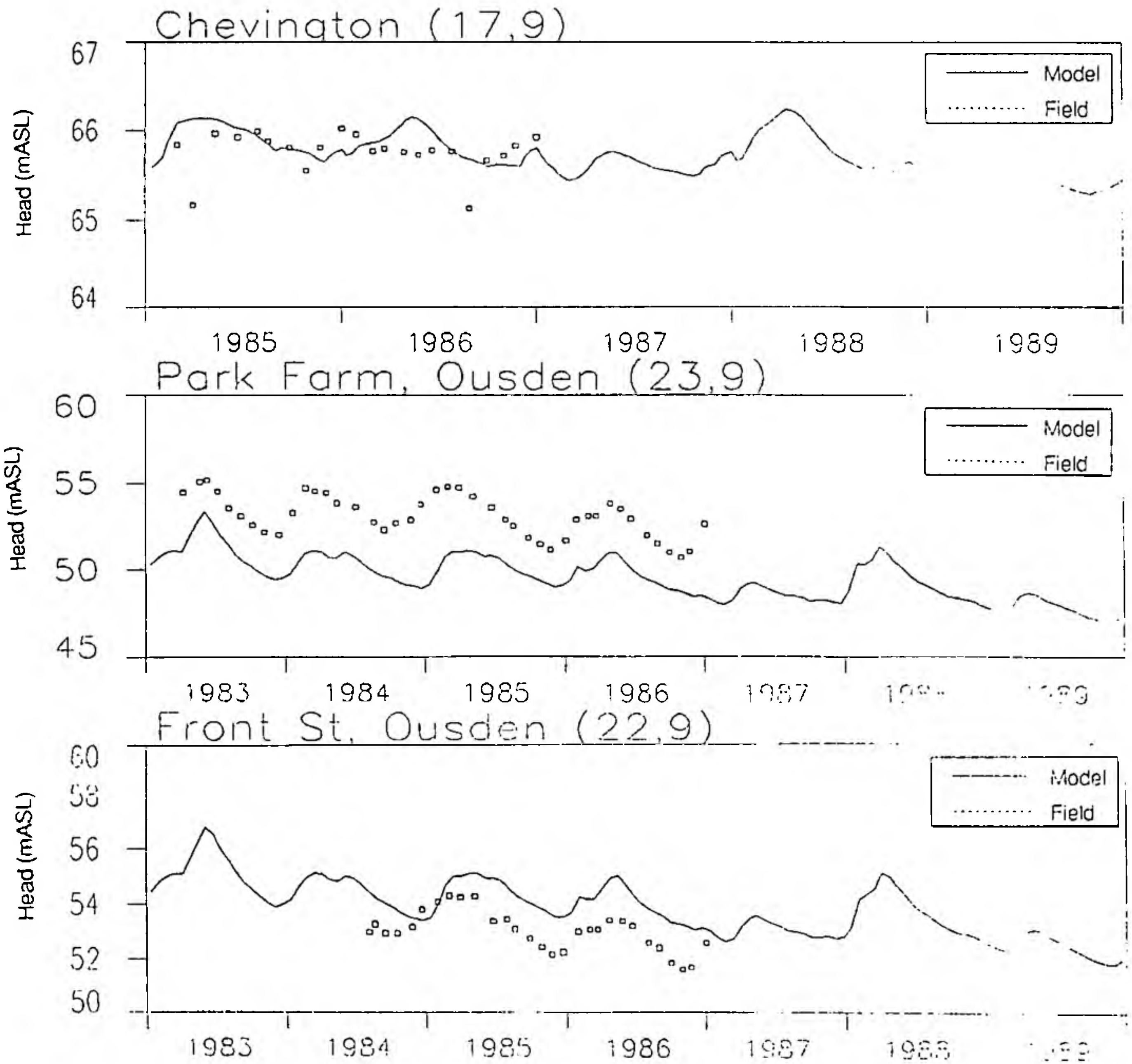
X(I), Y(J): Distances from origin of mesh coordinates in x and y directions. [m]

Z(I,J): Saturated depth at node. [m]

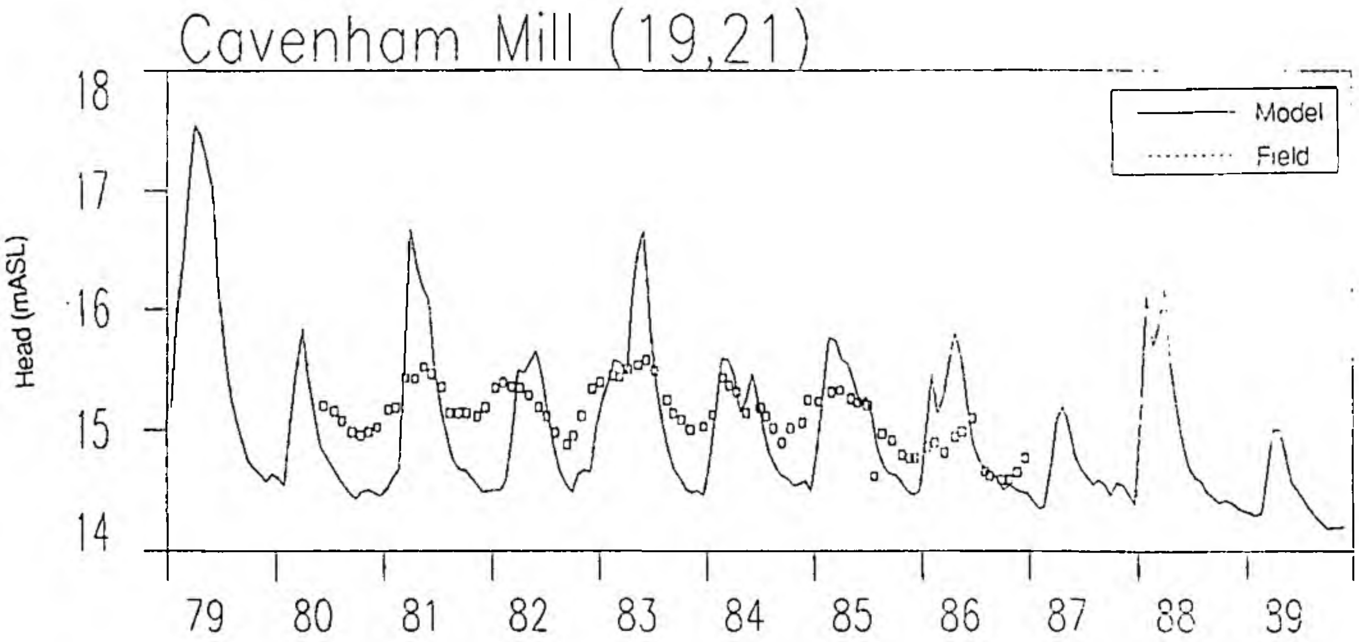
Appendix E

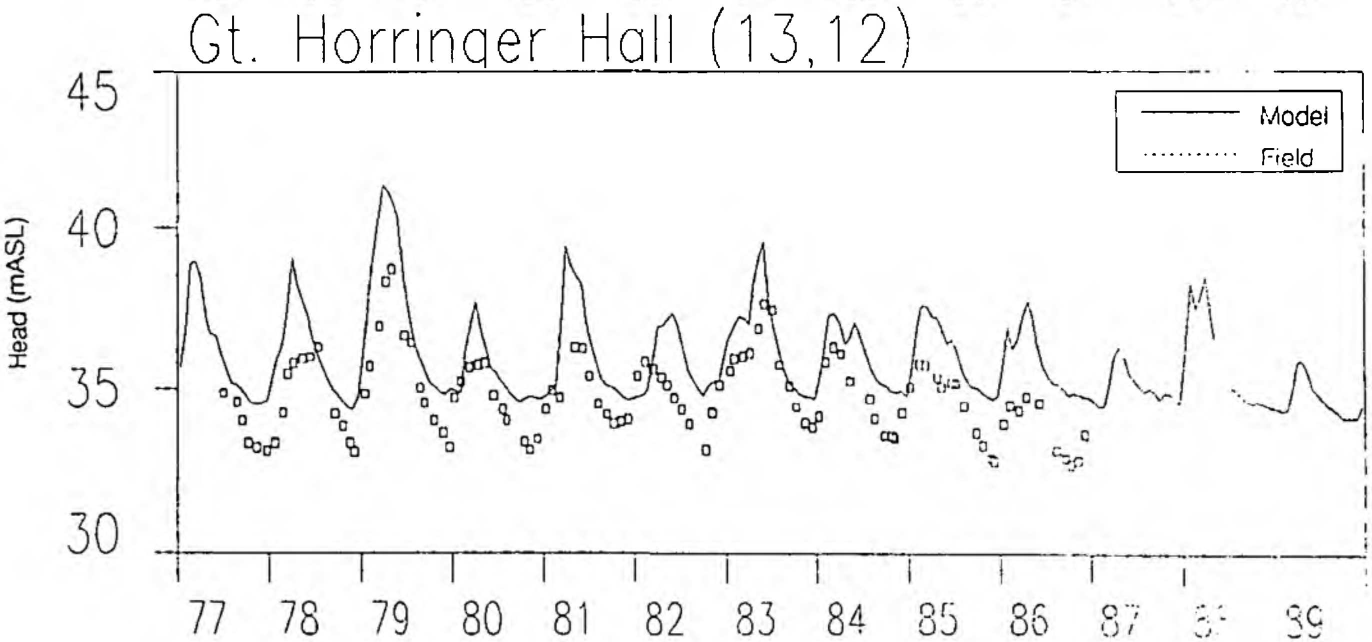
Comparative Plots:

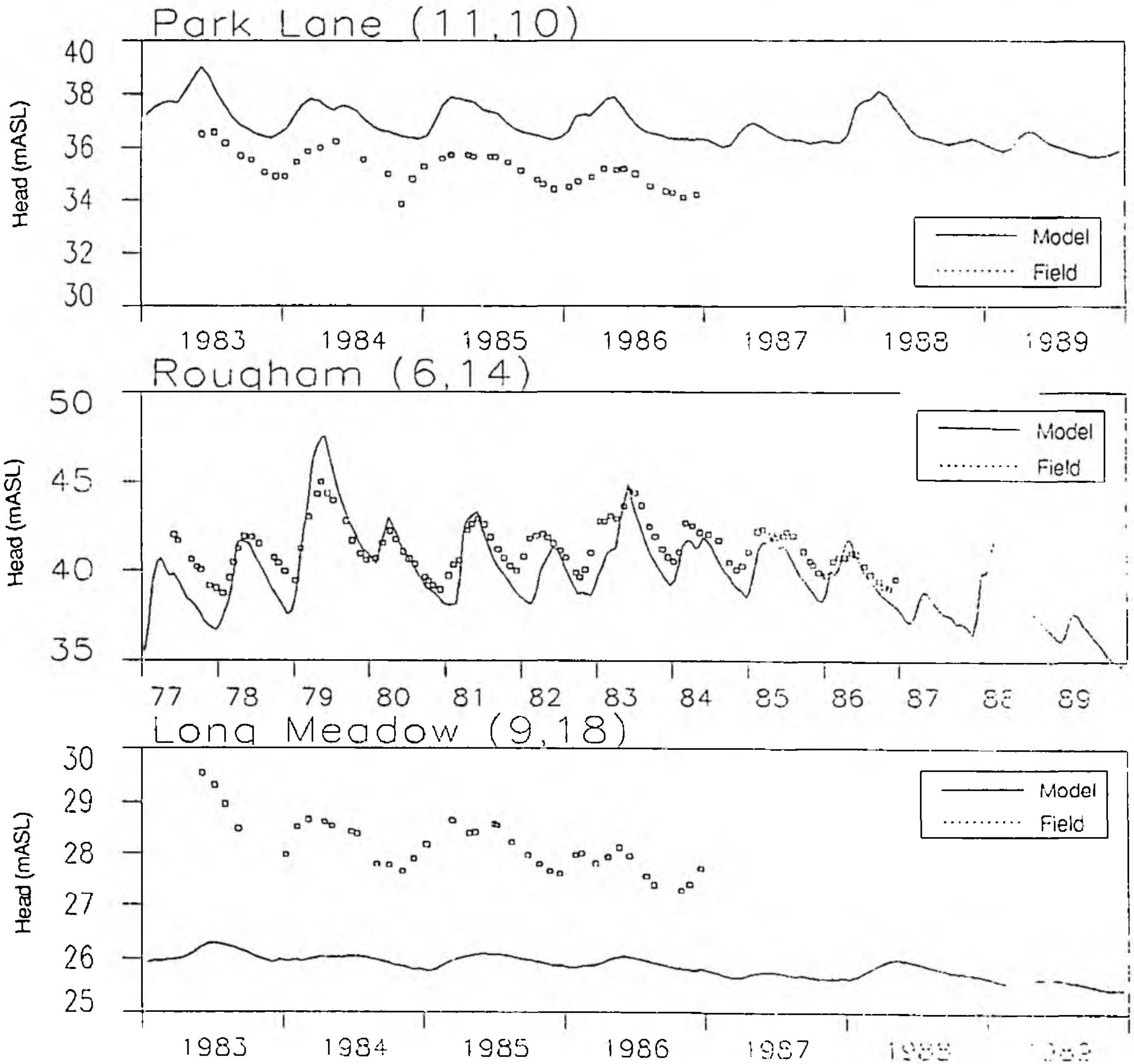
Field vs. Model Data.

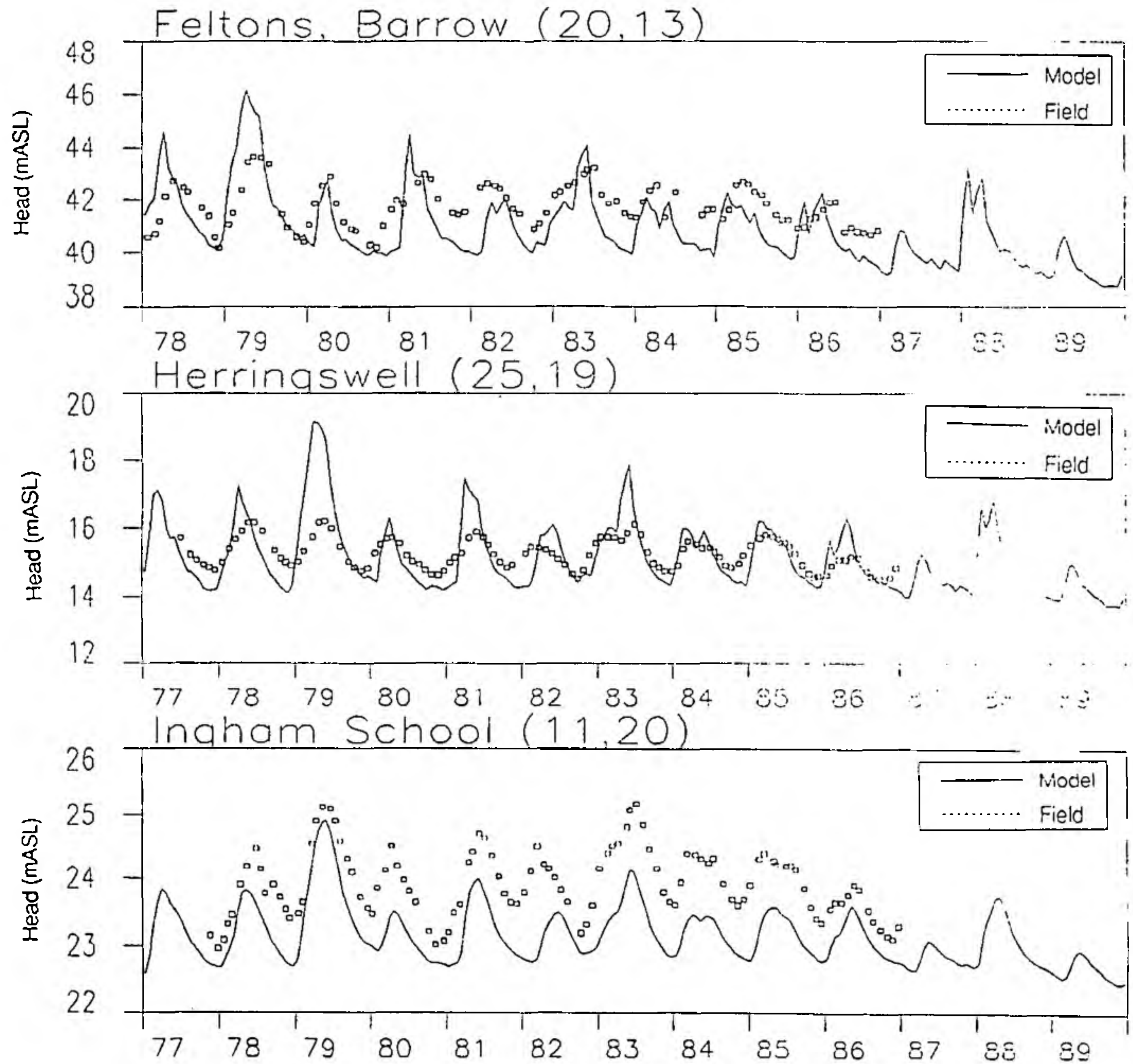


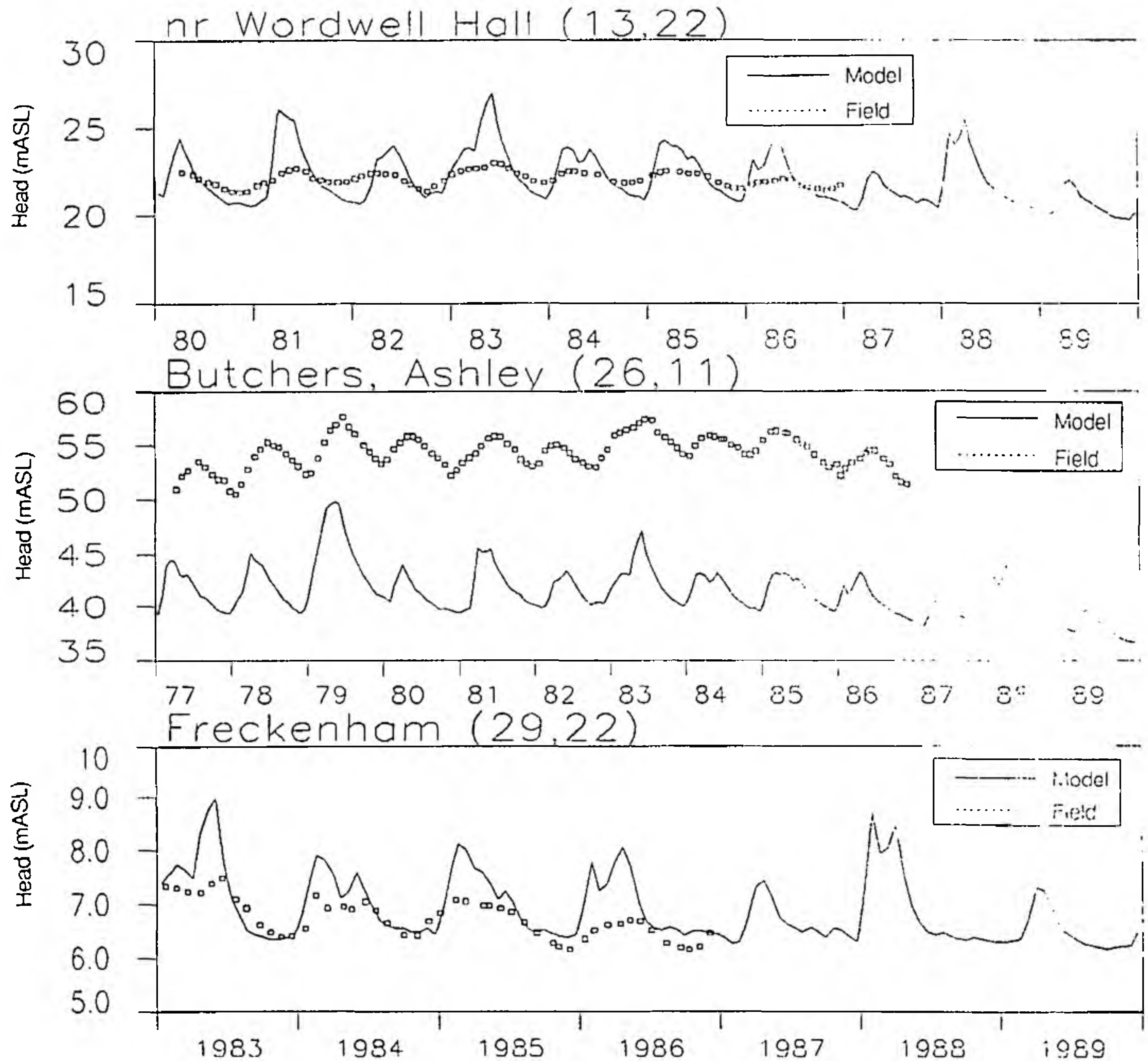
Chevington (17,9)
Head (mASL)
67
66
65
64
Model
Field
1985
1986
1987
1988
1989
Park Farm, Ousden (23,9)
Head (mASL)
60
55
50
45
Model
Field
1983
1984
1985
1986
Front St, Ousden (22,9)
Head (mASL)
60
58
56
54
52
50
Model
Field
1983
1984
1985
1986


Cavenham Mill (19,21)
Head (mASL)
18
17
16
15
14
Model
Field
79
80
81
82
83
84
85
86
87
88

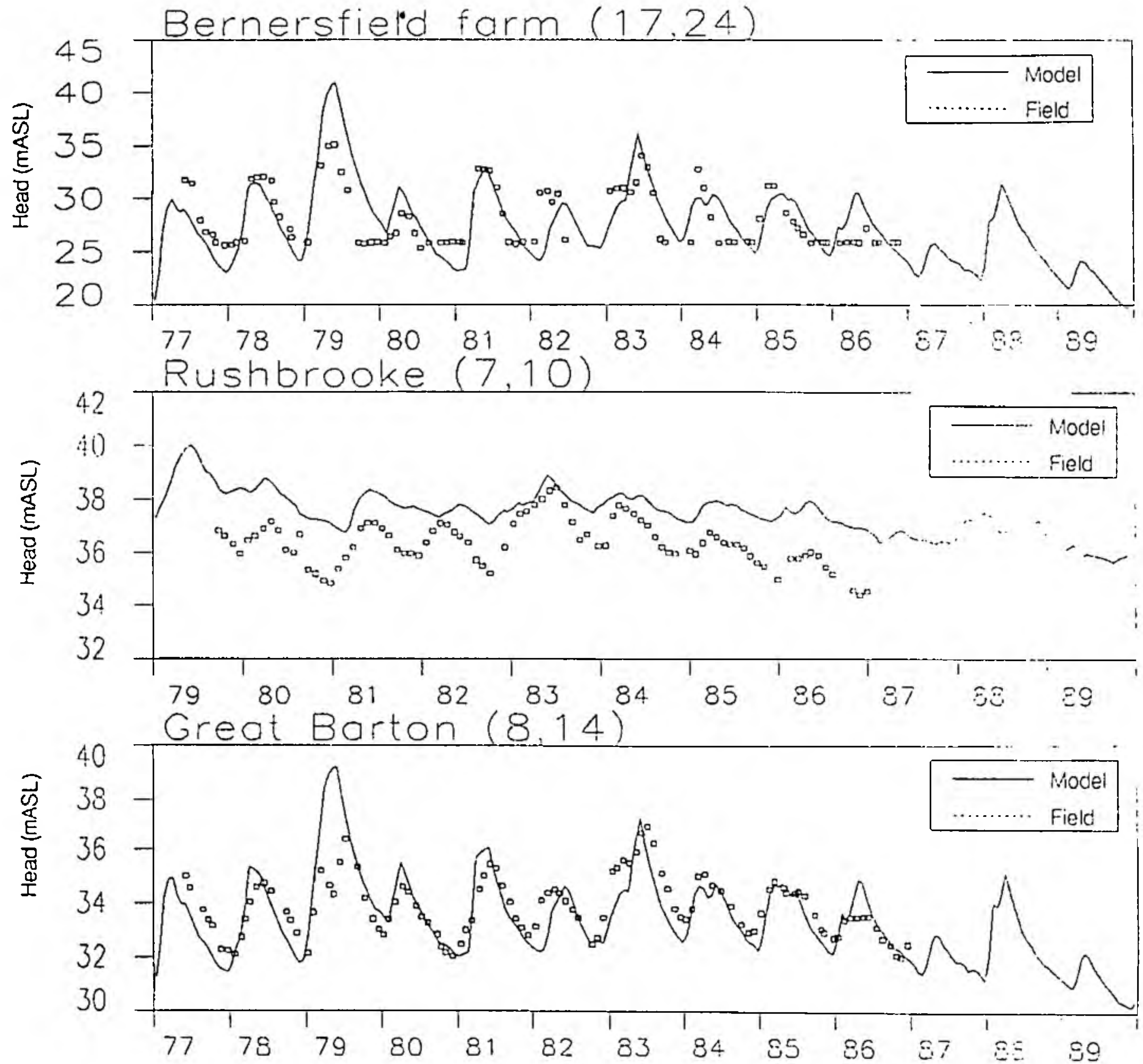

Gt. Horringer Hall (13,12)
Head (mASL)
45
40
35
30
Model
Field
77
78
79
80
81
82
83
84
85
86
87


Park Lane (11,10)
Head (mASL)
40
38
36
34
32
30
Model
Field
1983
1984
1985
1986
1987
1988
1989
Rougham (6,14)
Head (mASL)
50
45
40
35
Model
Field
77
78
79
80
81
82
83
84
85
86
87
89
Long Meadow (9,18)
Head (mASL)
30
29
28
27
26
25
Model
Field
1983
1984
1985
1986
1987

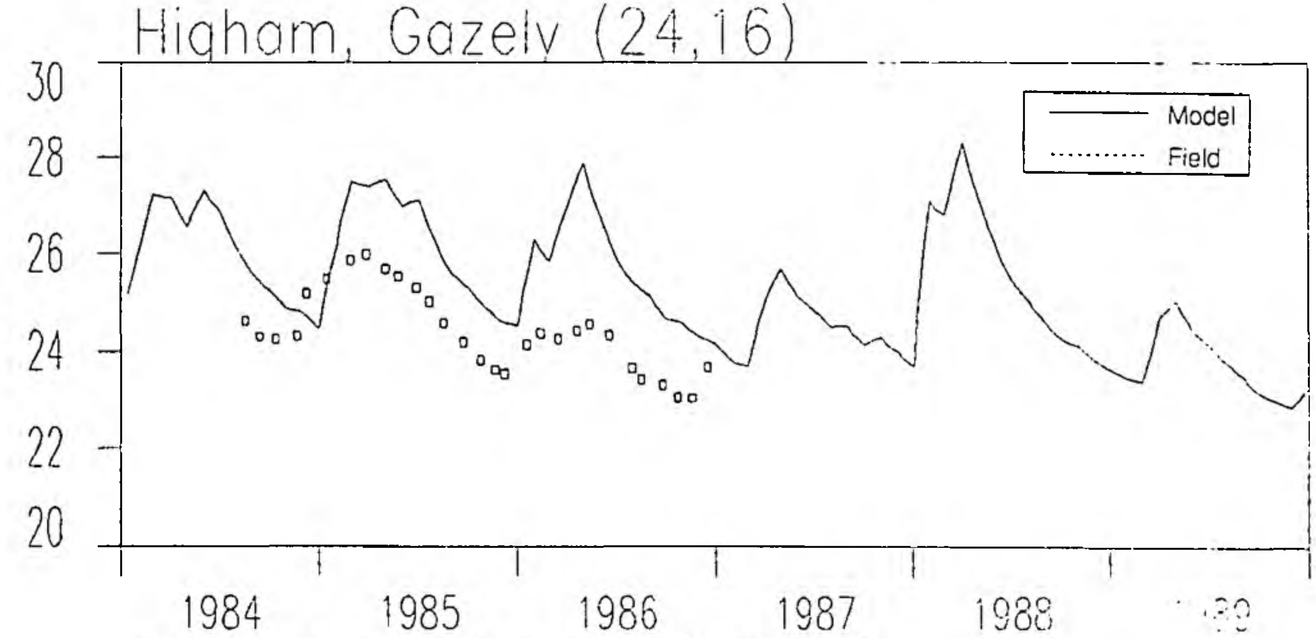

Feltons, Barrow (20,13)
Head (mASL)
48
46
44
42
40
38
Model
Field
78
79
80
81
82
83
84
85
86
87
89
Herringswell (25,19)
Head (mASL)
20
18
16
14
12
Model
Field
77
78
79
80
81
82
83
84
85
86
Ingham School (11,20)
Head (mASL)
26
25
24
23
22
Model
Field
77
78
79
80
81
82
83
84
85
86
87
88
89

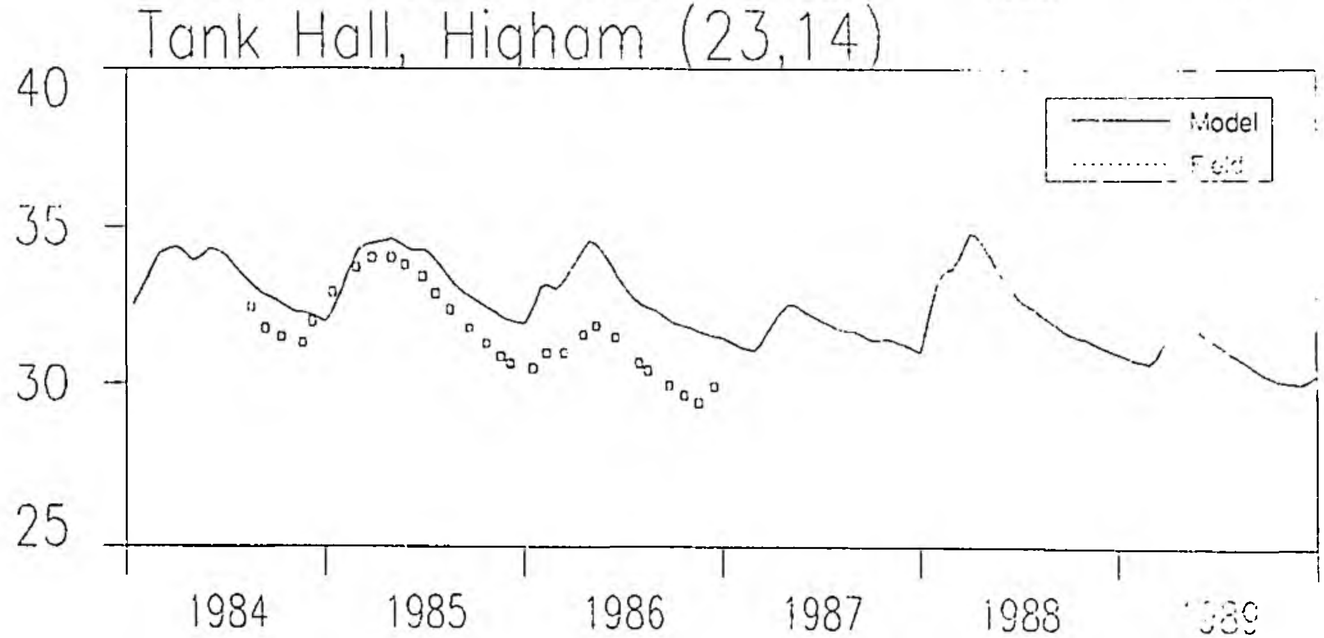


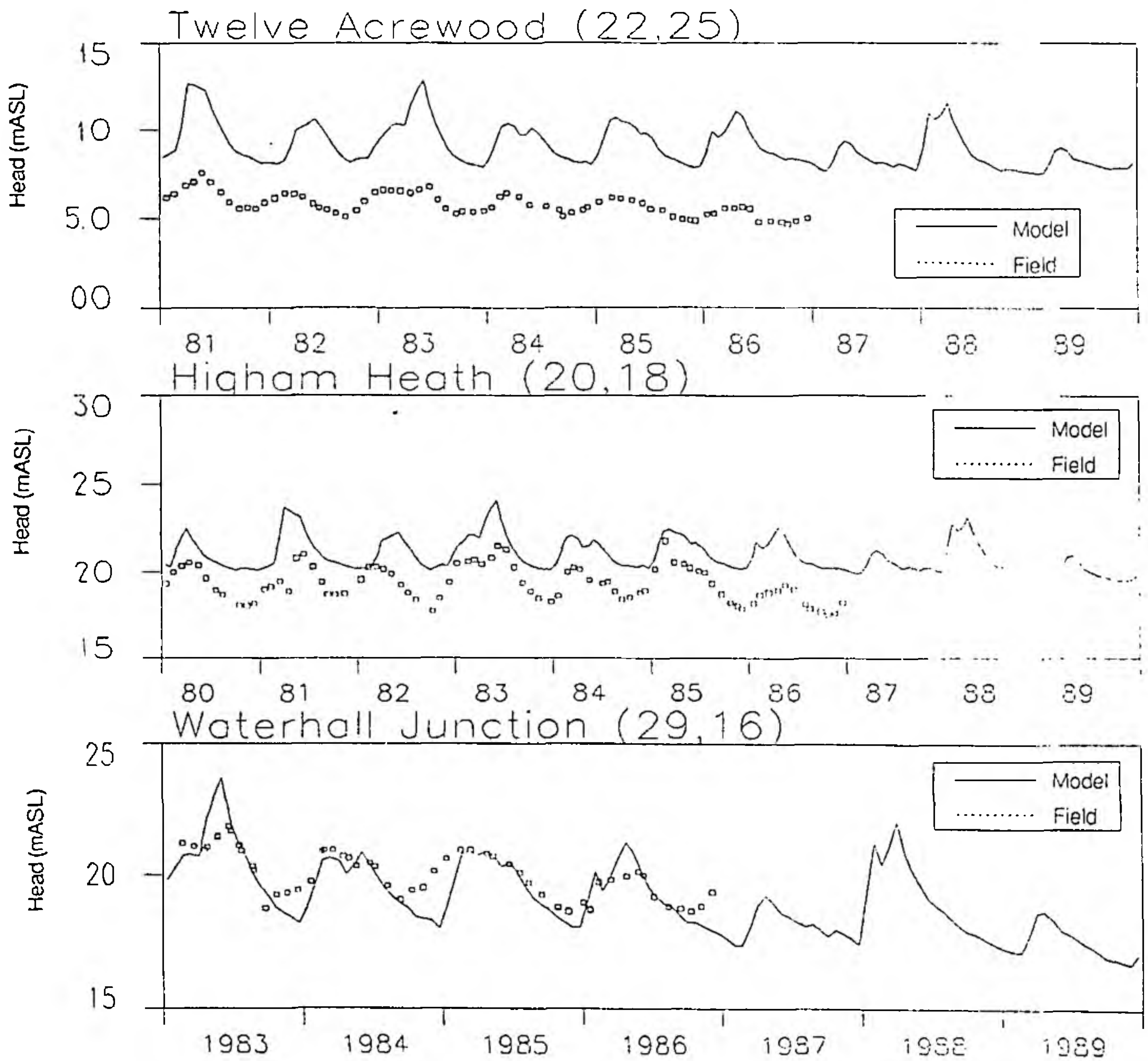
nr Wordwell Hall (13,22)
Head (mASL)
30
25
20
15
80
81
82
83
84
85
86
87
88
89
Model
Field
Butchers, Ashley (26,11)
Head (mASL)
60
55
50
45
40
35
77
78
79
80
81
82
83
84
85
86
87
89
Model
Field
Freckenham (29,22)
Head (mASL)
10
9.0
8.0
7.0
6.0
5.0
1983
1984
1985
1986
1987
1988
1989
Field



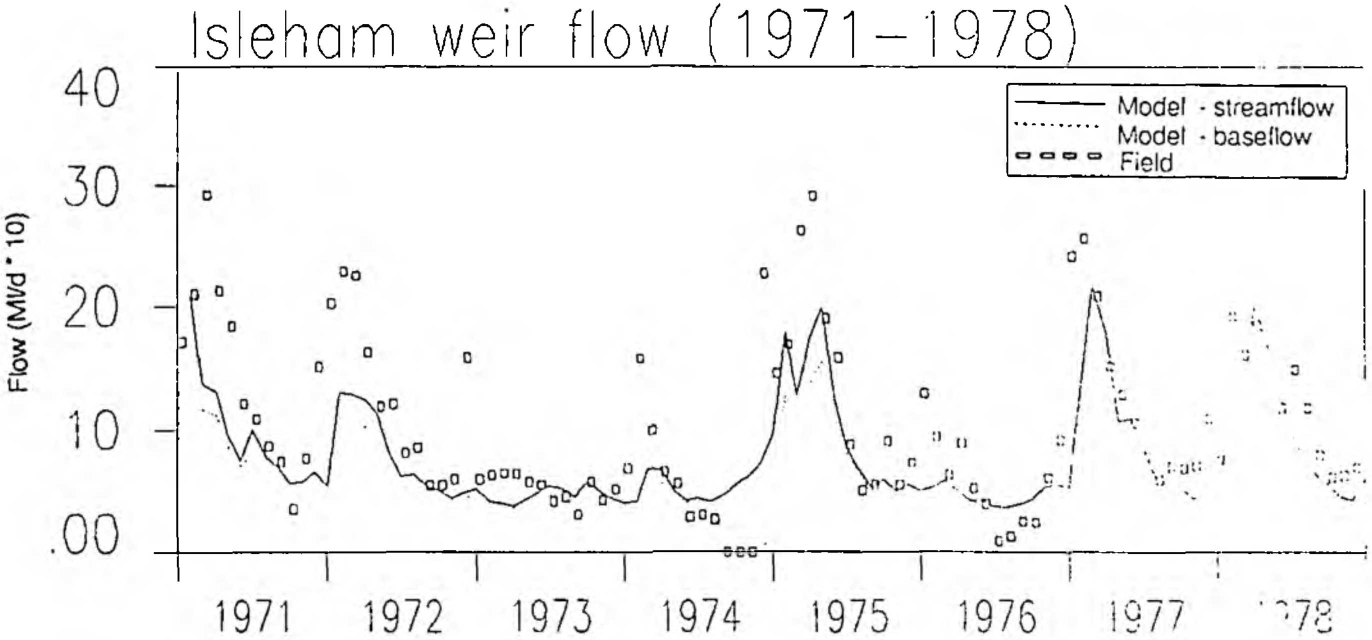
Bernersfield farm (17,24)
Head (mASL)
45
40
35
30
25
20
Model
Field
77
78
79
80
81
82
83
84
85
86
87
89
Rushbrooke (7,10)
Head (mASL)
42
40
38
36
34
32
Model
Field
79
80
81
82
83
84
85
86
87
88
89
Great Barton (8,14)
Head (mASL)
40
38
36
34
32
30
Model
Field
77
78
79
80
81
82
83
84
85
86
87
89


Higham, Gazely (24,16)
30
28
26
24
22
20
Model
Field
1984
1985
1986
1987
1988

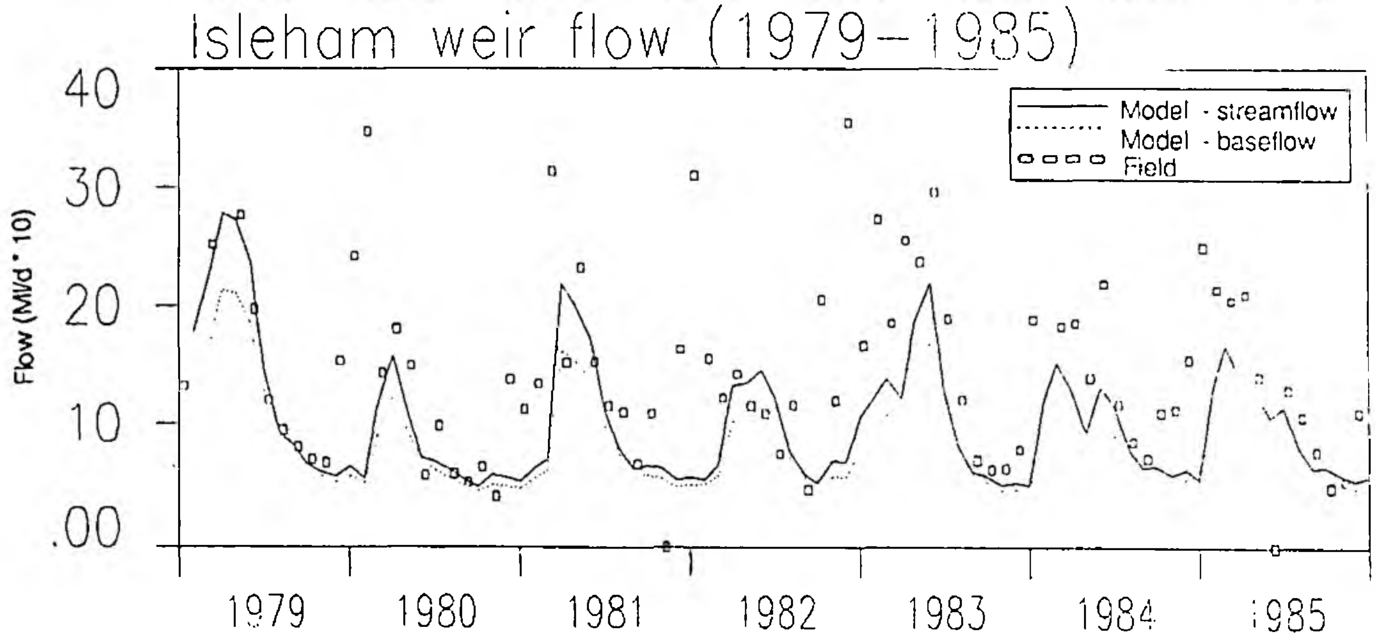

Tank Hall, Higham (23,14)
40
35
30
25
Model
1984
1985
1986
1987
1988



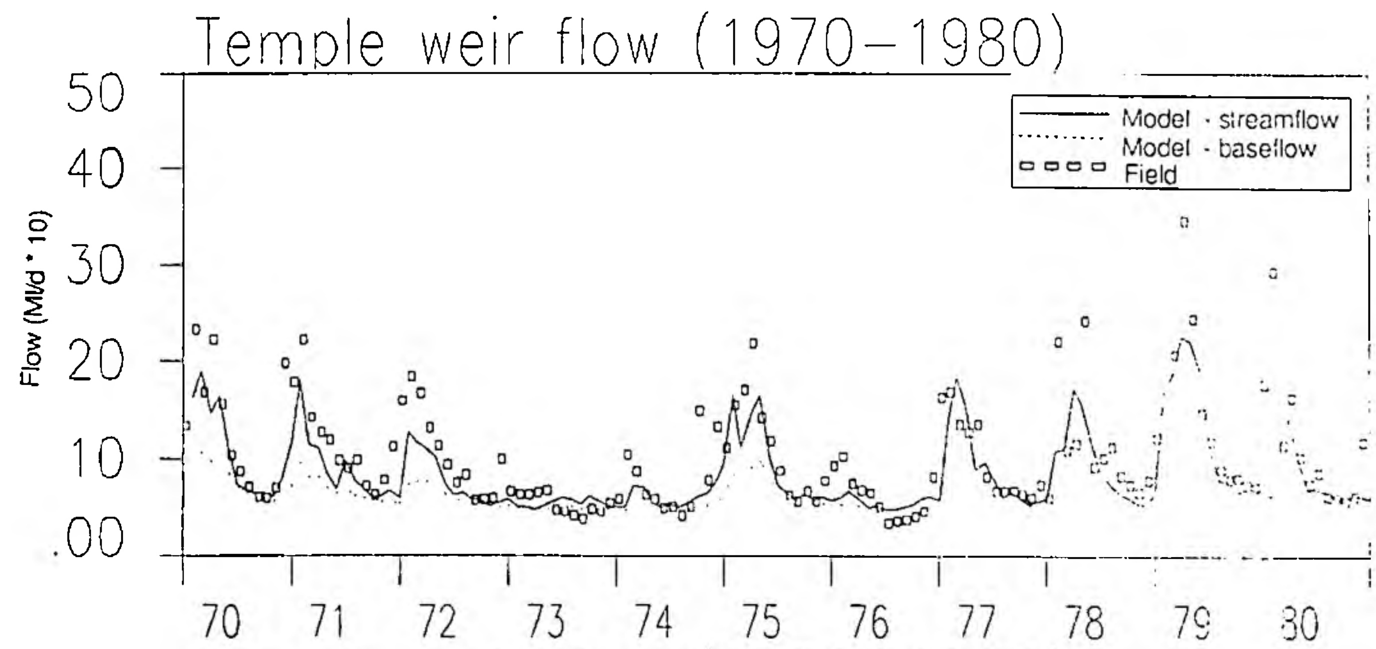
Twelve Acrewood (22,25)
Head (mASL)
15
10
5.0
00
81
82
83
84
85
86
87
88
89
Model
Field
Higham Heath (20,18)
Head (mASL)
30
25
20
15
80
81
82
83
84
85
86
87
88
89
Model
Field
Waterhall Junction (29,16)
Head (mASL)
25
20
15
1983
1984
1985
1986
1987
1988
1989
Model
Field



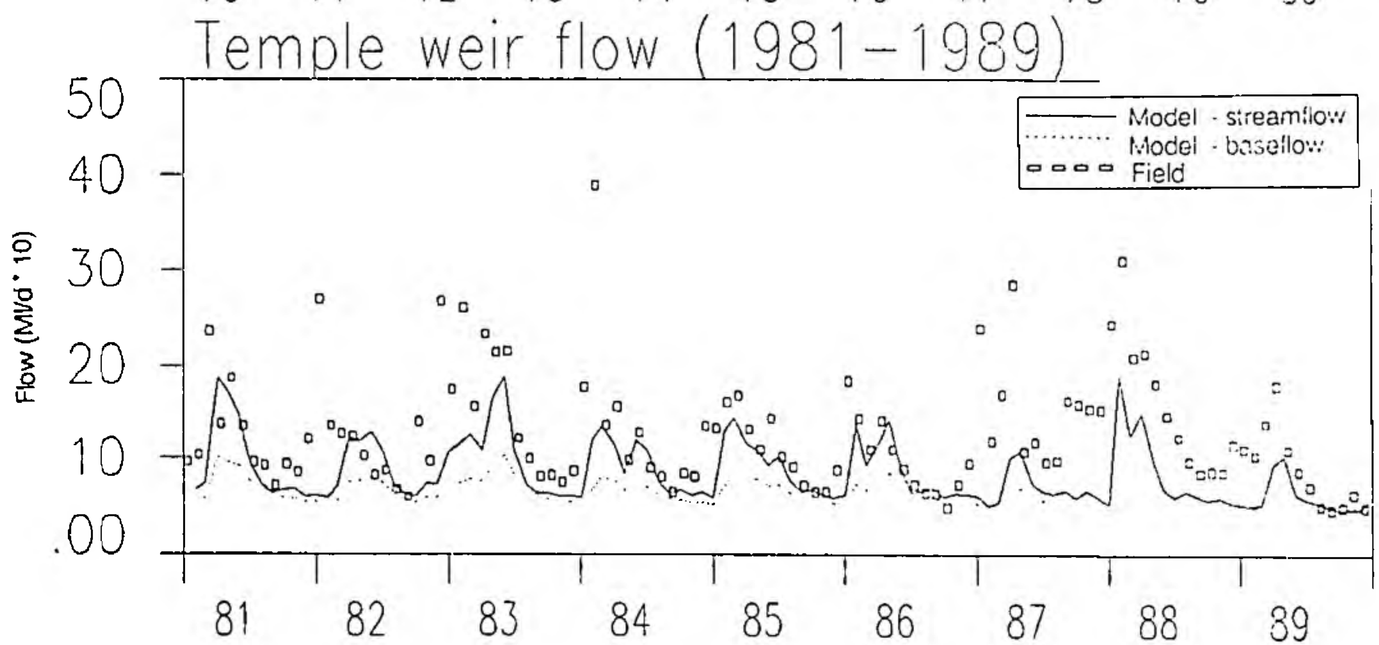
Isleham weir flow (1971-1978)
Flow (Ml/d * 10)
40
30
20
10
.00
Model - streamflow
Model - baseflow
Field
1971
1972
1973
1974
1975
1976
1977



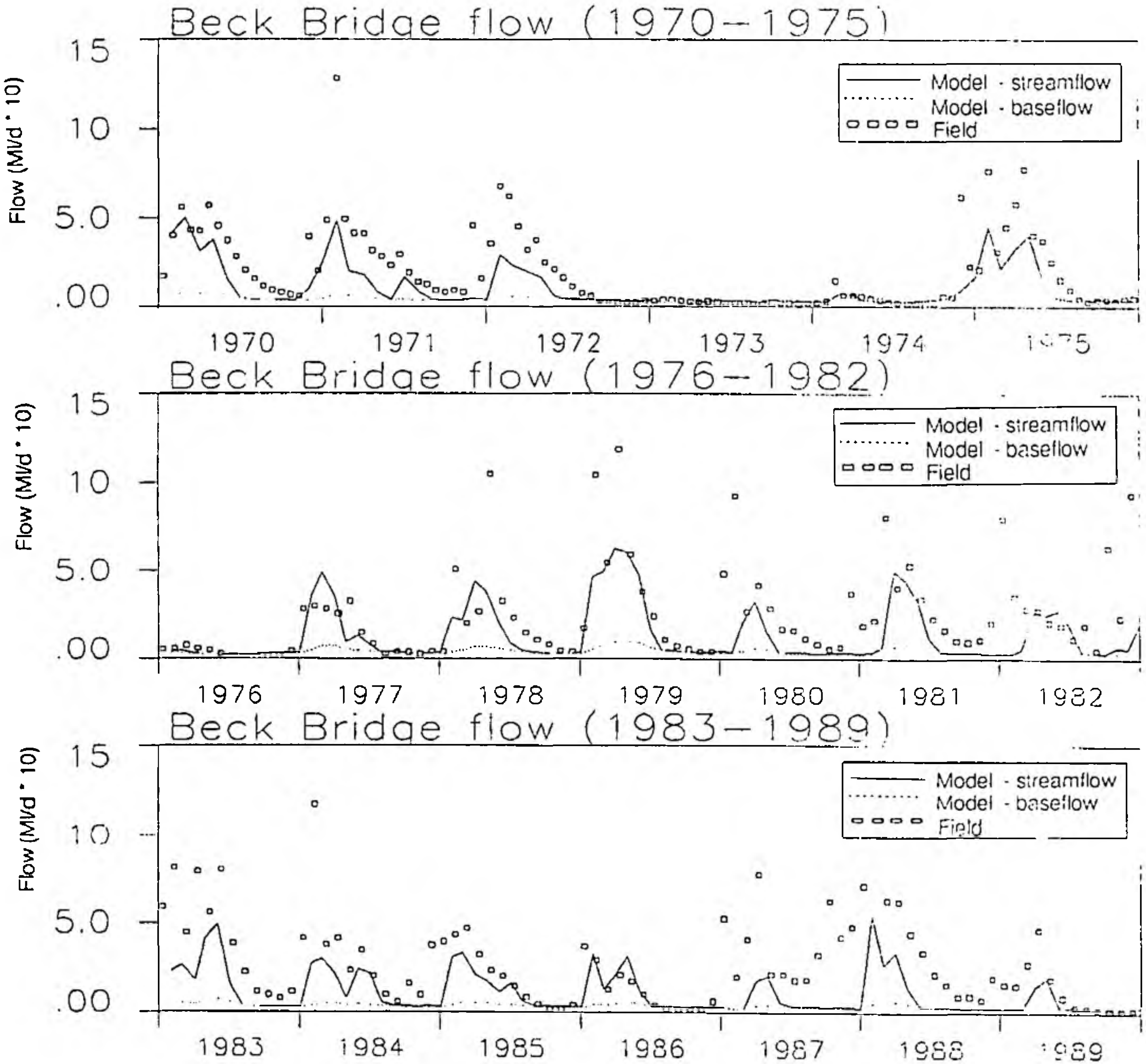
Isleham weir flow (1979-1985)
Flow (Ml/d * 10)
40
30
20
10
.00
Model - streamflow
Model - baseflow
Field
1979
1980
1981
1982
1983
1984
1985



Temple weir flow (1970–1980)
Flow (Ml/d * 10)
.00
10
20
30
40
50
70
71
72
73
74
75
76
77
78
79
80
Model - streamflow
Model - baseflow
Field

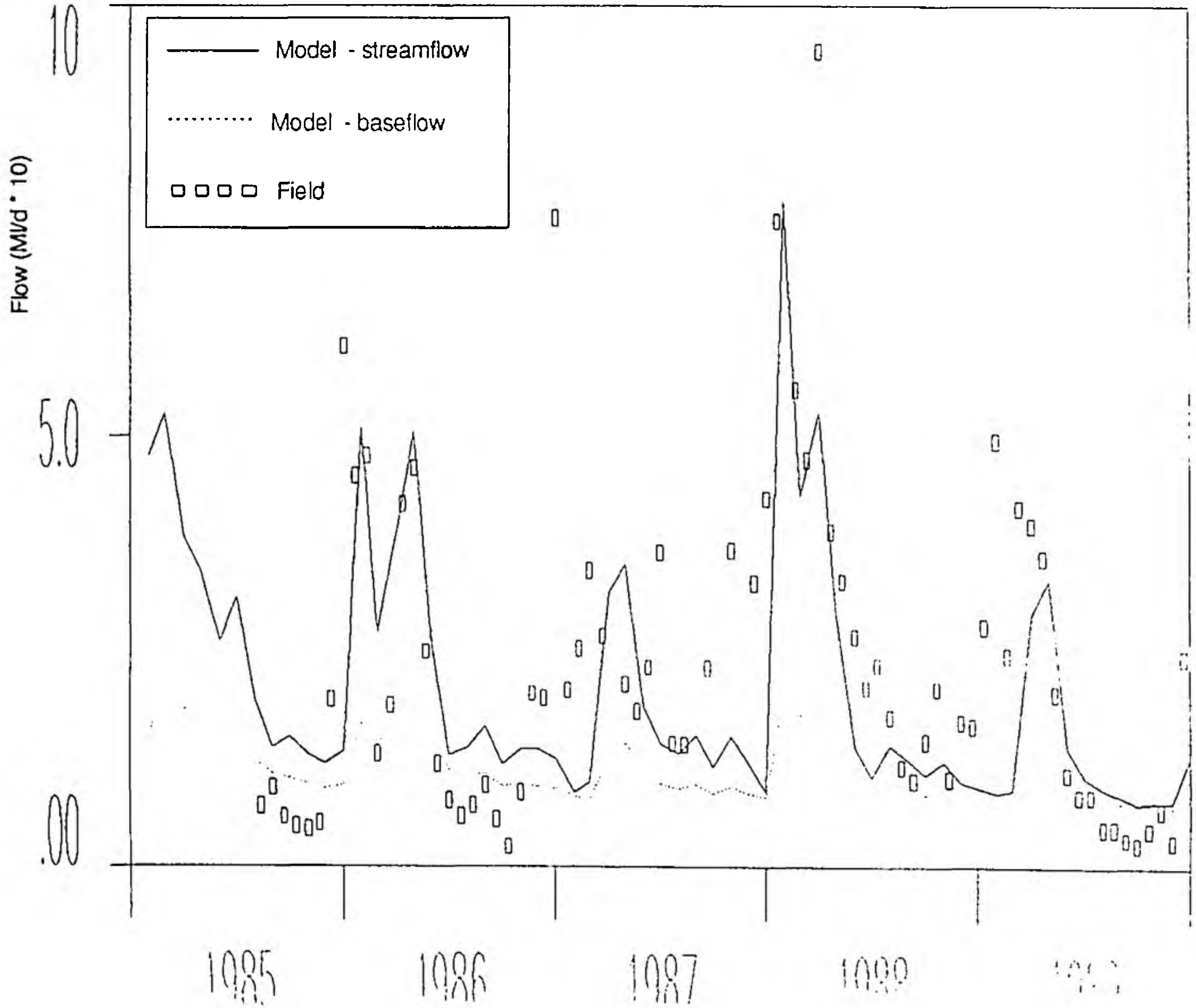


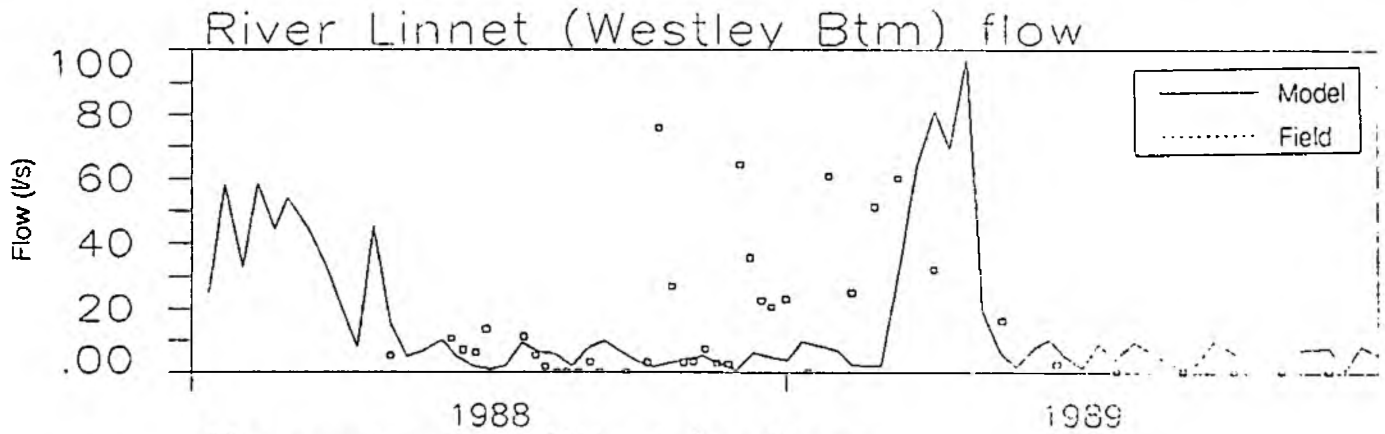
Temple weir flow (1981–1989)
Flow (Ml/d * 10)
.00
10
20
30
40
50
81
82
83
84
85
86
87
88
89
Model - streamflow
Model - baseflow
Field

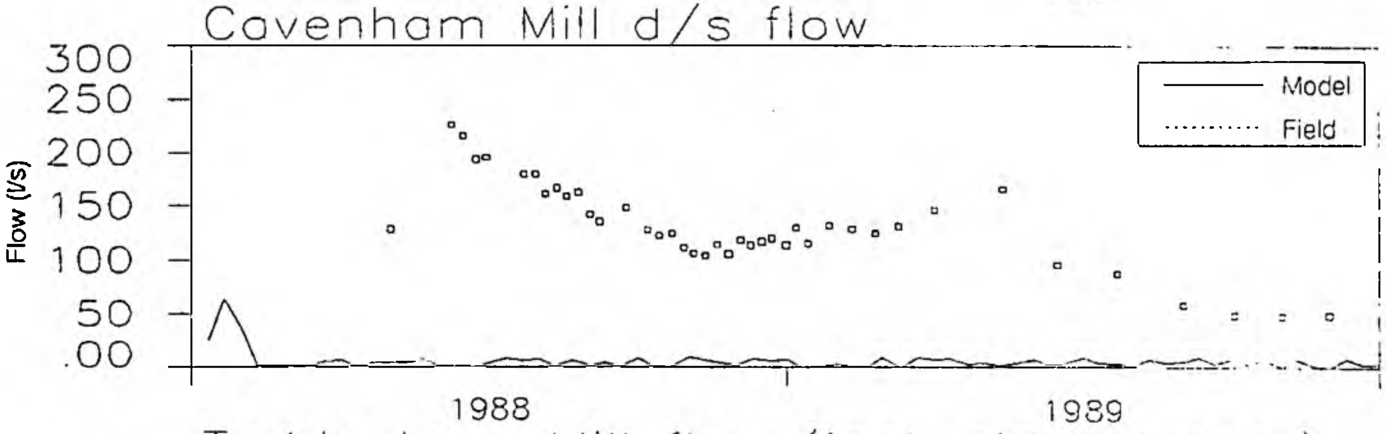


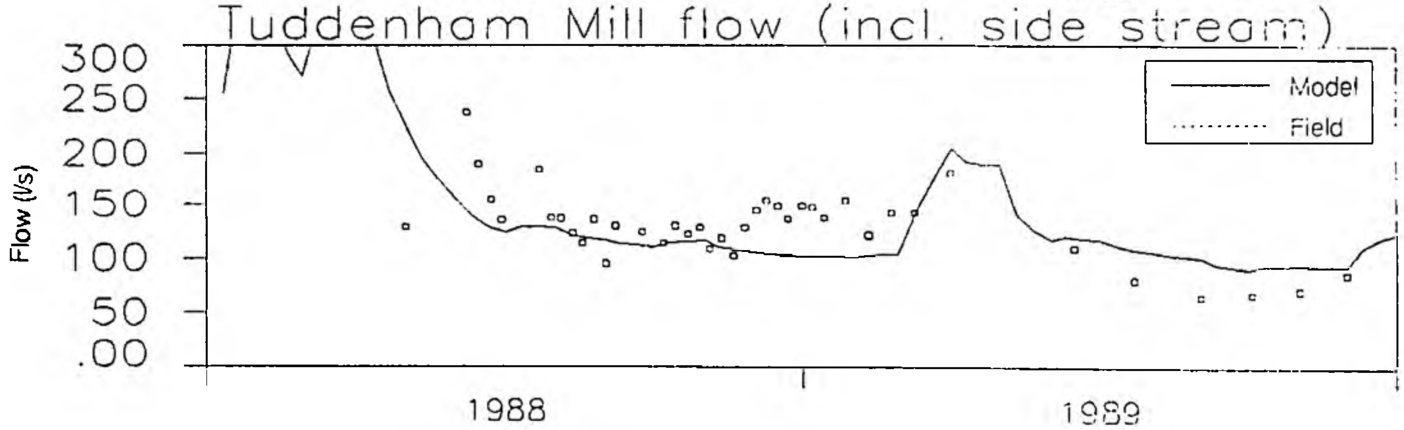
Beck Bridge flow (1970–1975)
Flow (Ml/d * 10)
15
10
5.0
.00
1970
1971
1972
1973
1974
1975
Model - streamflow
Model - baseflow
Field
Beck Bridge flow (1976–1982)
Flow (Ml/d * 10)
15
10
5.0
.00
1976
1977
1978
1979
1980
1981
1982
Model - streamflow
Model - baseflow
Field
Beck Bridge flow (1983–1989)
Flow (Ml/d * 10)
15
10
5.0
.00
1983
1984
1985
1986
1987
1988
1989
Model - streamflow
Model - baseflow
Field

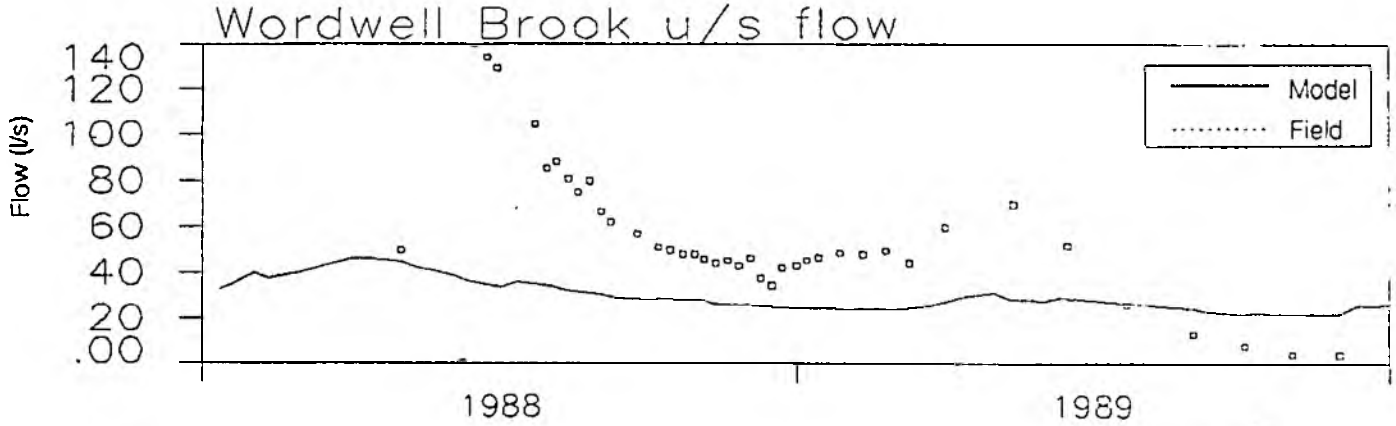
Fornham St Martins flow

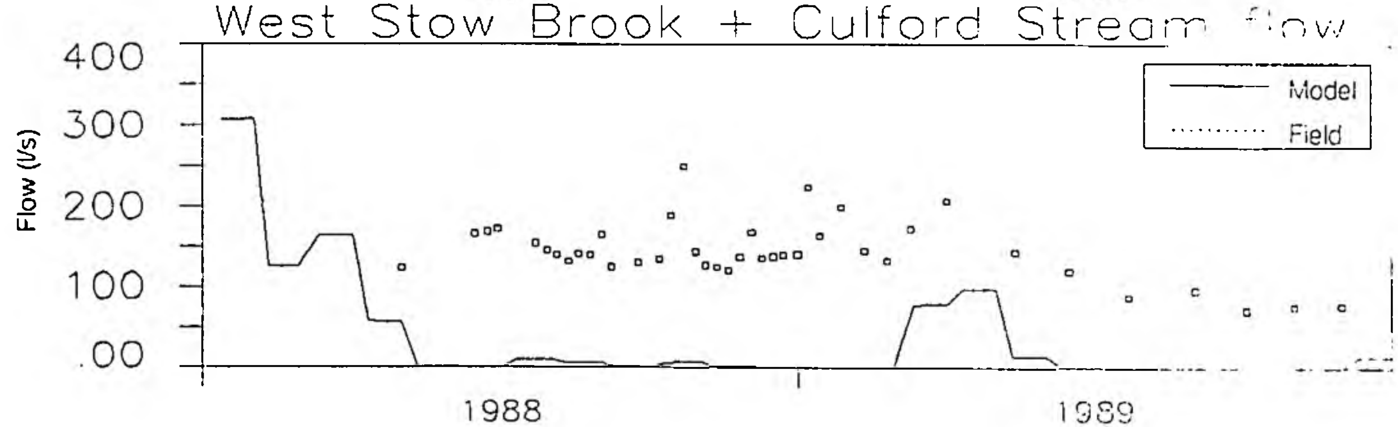


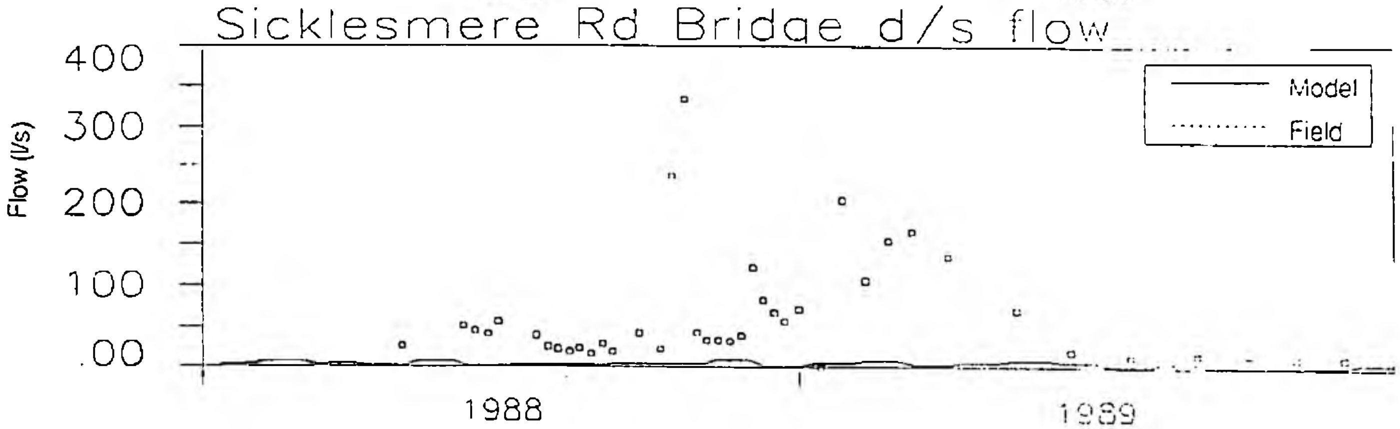

River Linnet (Westley Btm) flow
Flow (l/s)
100
80
60
40
20
.00
Model
Field
1988
1989

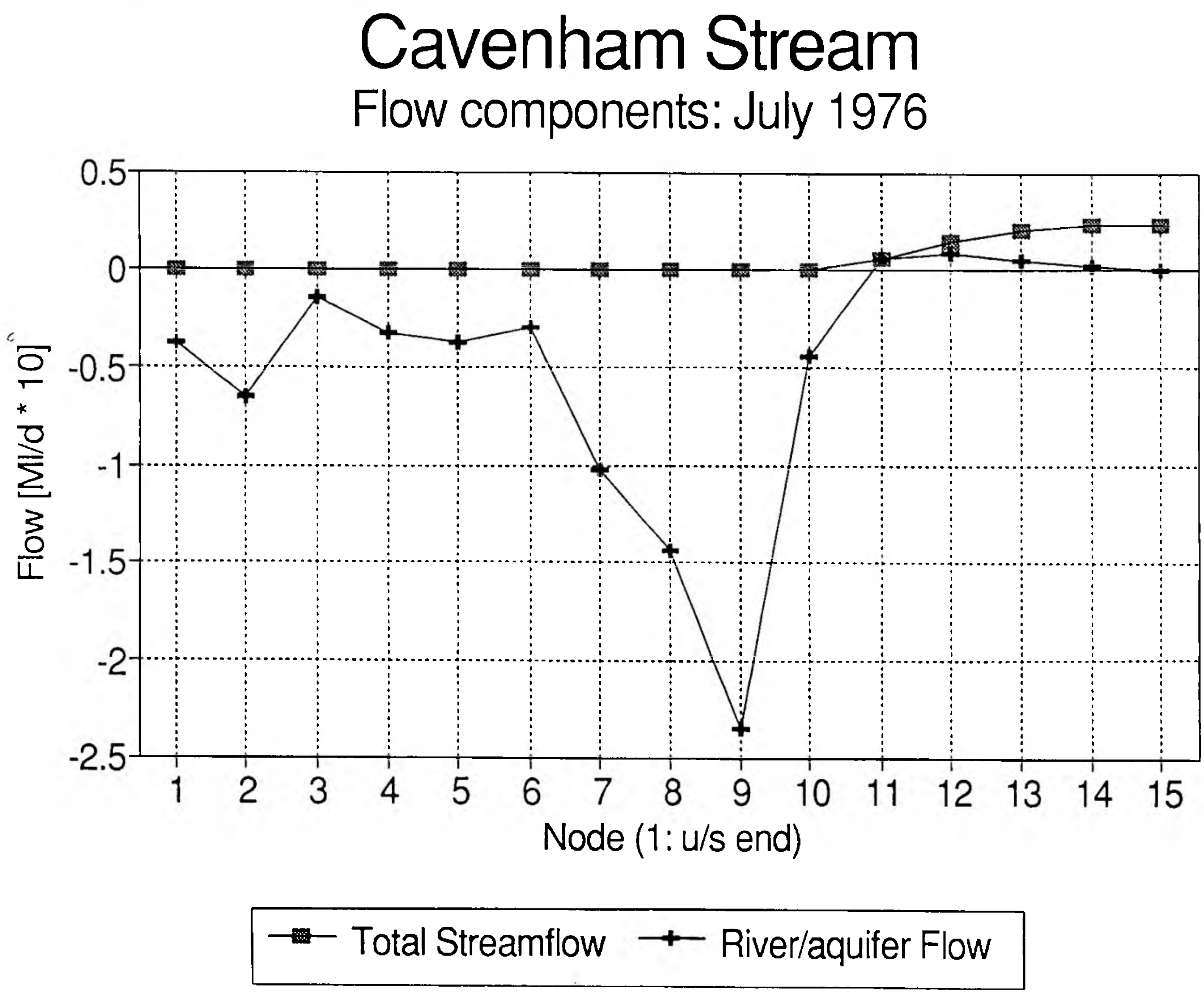

Cavenham Mill d/s flow
Flow (l/s)
300
250
200
150
100
50
.00
Model
Field
1988
1989

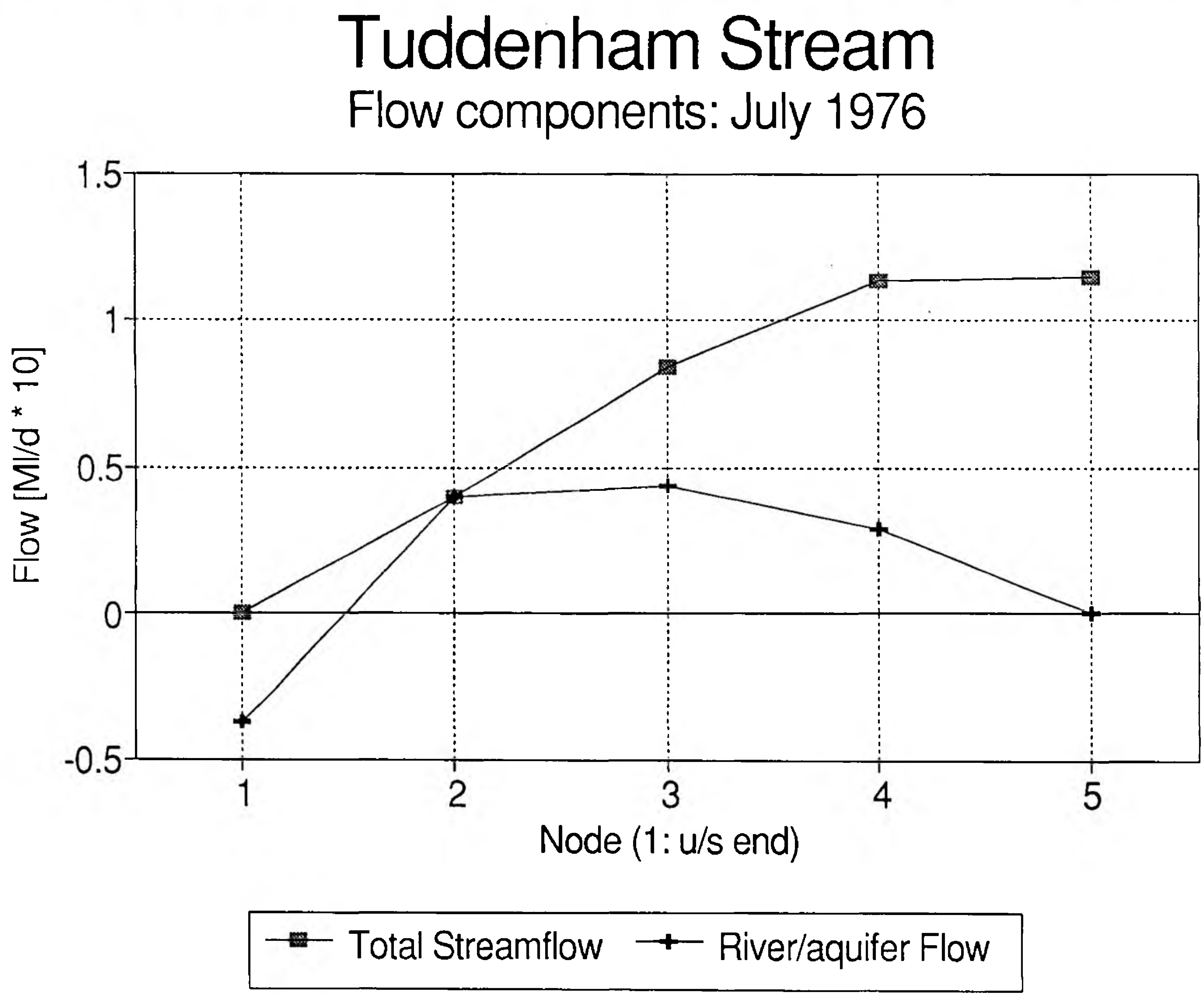

Tuddenham Mill flow (incl. side stream)
Flow (l/s)
300
250
200
150
100
50
.00
Model
Field
1988
1989

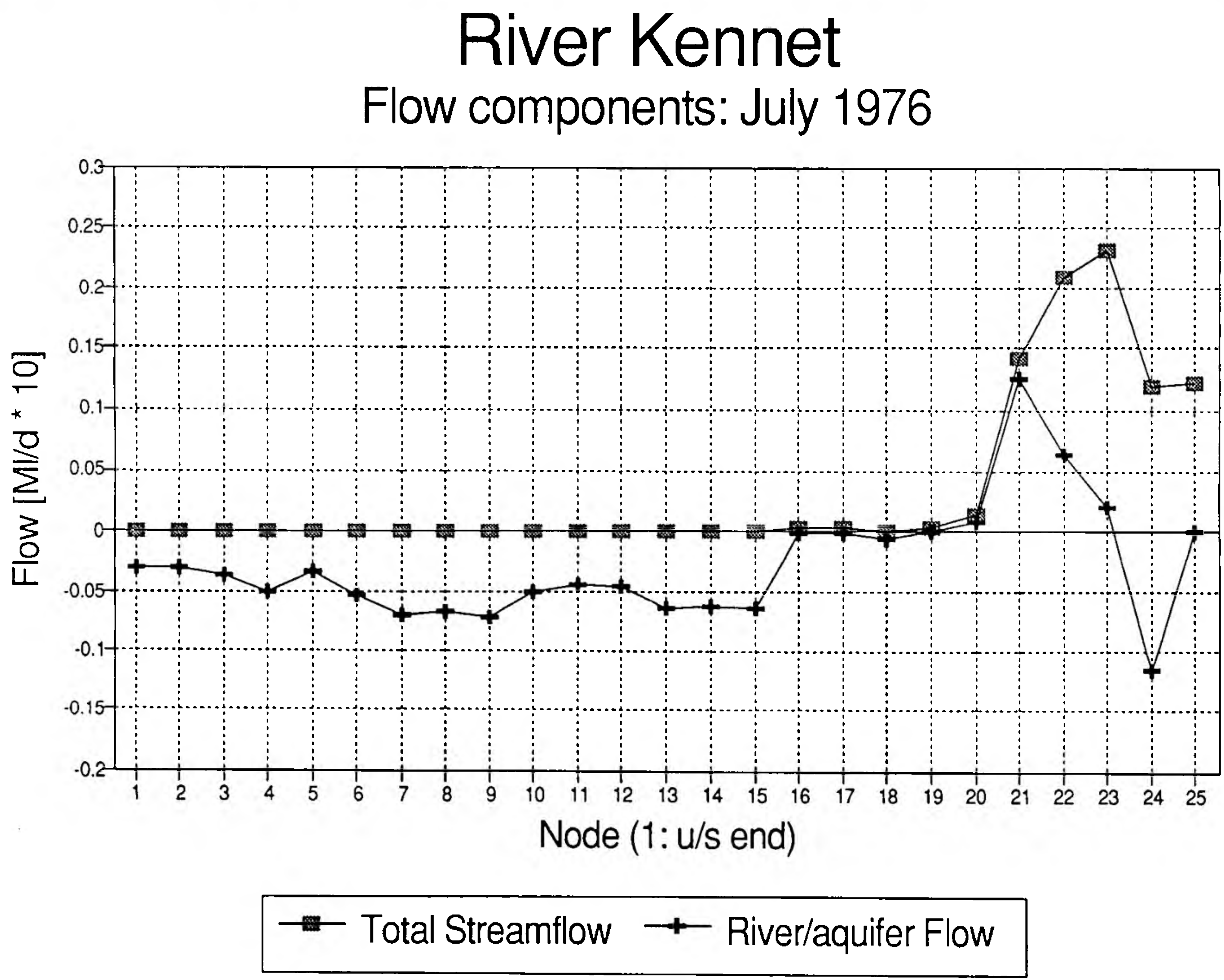

Wordwell Brook u/s flow
Flow (l/s)
140
120
100
80
60
40
20
.00
Model
Field
1988
1989

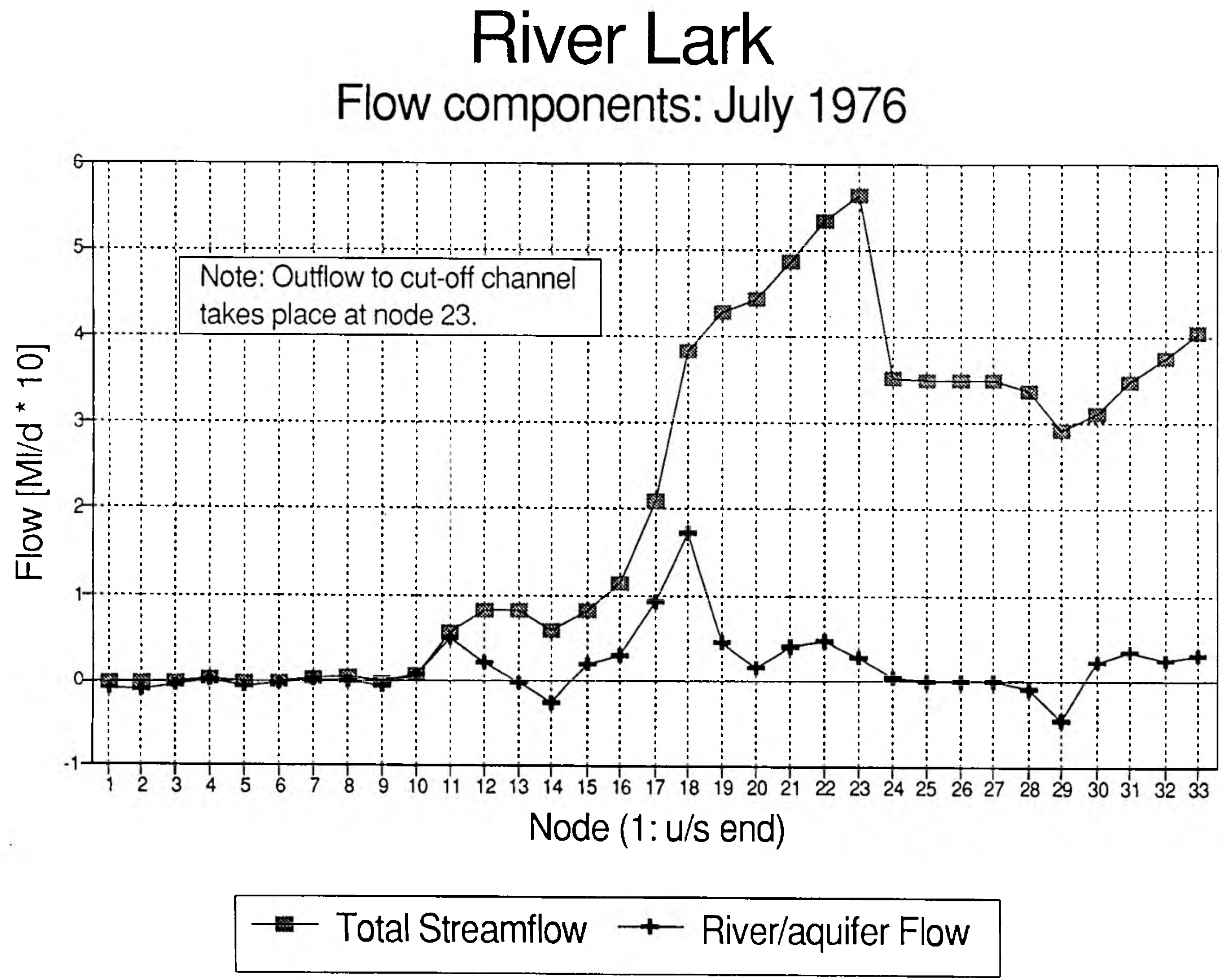

West Stow Brook + Culford Stream flow
Flow (l/s)
400
300
200
100
.00
Model
Field
1988
1989

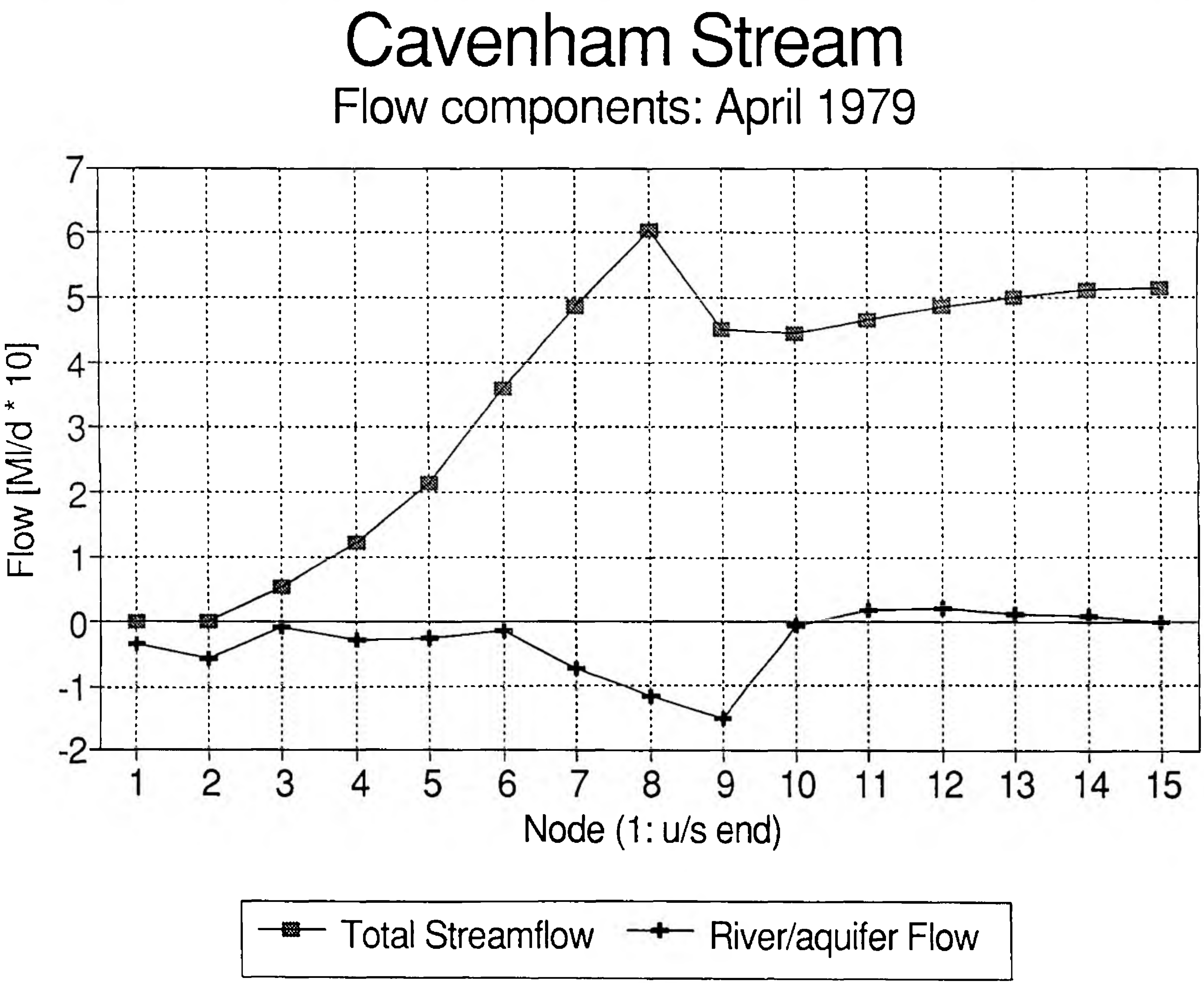

Sicklesmere Rd Bridge d/s flow
Flow (l/s)
400
300
200
100
.00
Model
Field
1988
1989


Cavenham Stream
Flow components: July 1976
0.5
0
-0.5
-1
-1.5
-2
-2.5
Flow [Ml/d * 10]
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Node (1: u/s end)
Total Streamflow
River/aquifer Flow


Tuddenham Stream
Flow components: July 1976
Flow [Ml/d * 10]
1.5
1
0.5
0
-0.5
1
2
3
4
5
Node (1: u/s end)
Total Streamflow
River/aquifer Flow

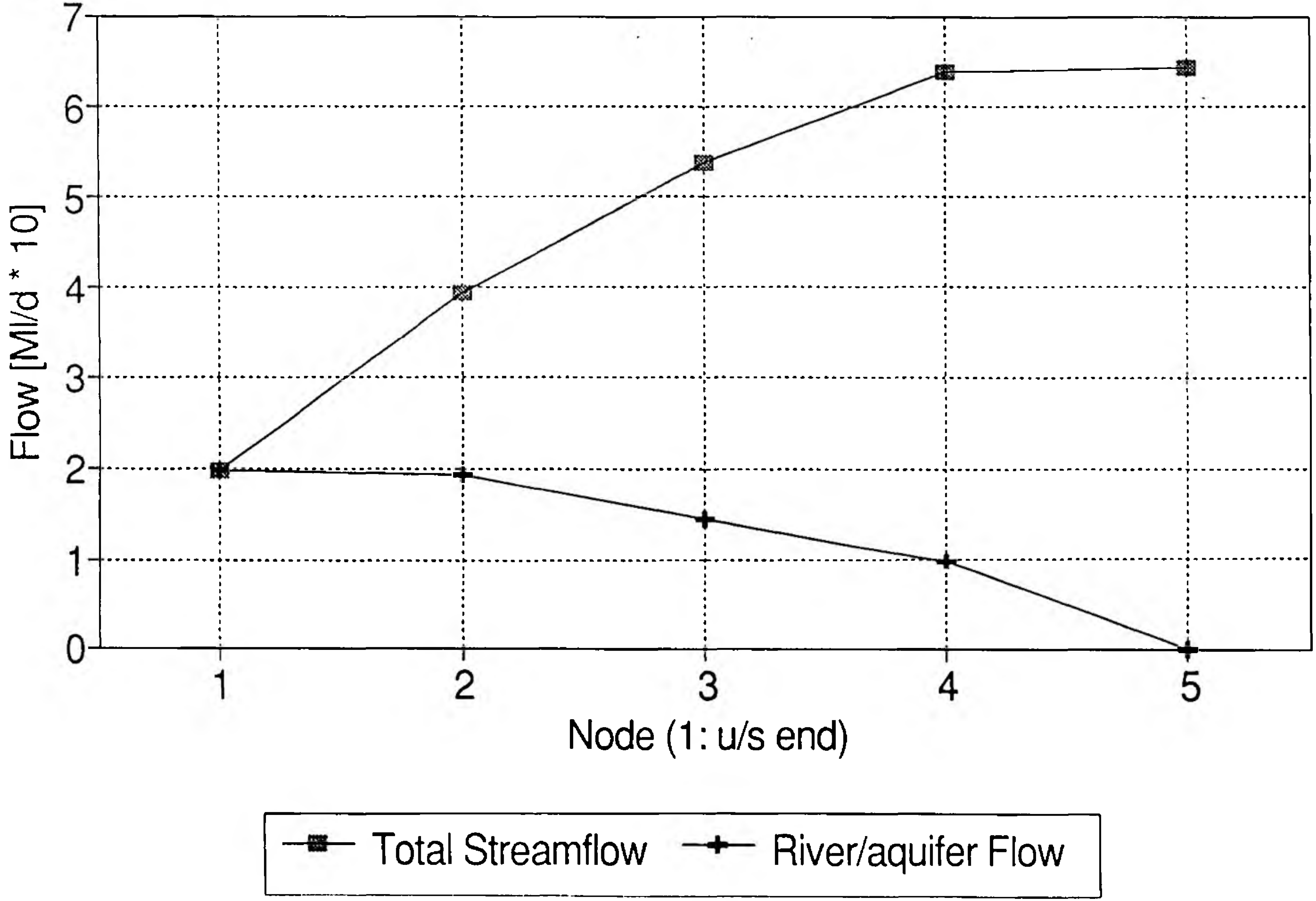

River Kennet
Flow components: July 1976
Flow [Ml/d * 10]
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
-0.15
-0.2
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
Node (1: u/s end)
Total Streamflow
River/aquifer Flow


River Lark
Flow components: July 1976
Note: Outflow to cut-off channel takes place at node 23.
Flow [Ml/d * 10]
Node (1: u/s end)
Total Streamflow
River/aquifer Flow
6
5
4
3
2
1
0
-1
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33


Cavenham Stream
Flow components: April 1979
Flow [Ml/d * 10]
7
6
5
4
3
2
1
0
-1
-2
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Node (1: u/s end)
Total Streamflow
River/aquifer Flow

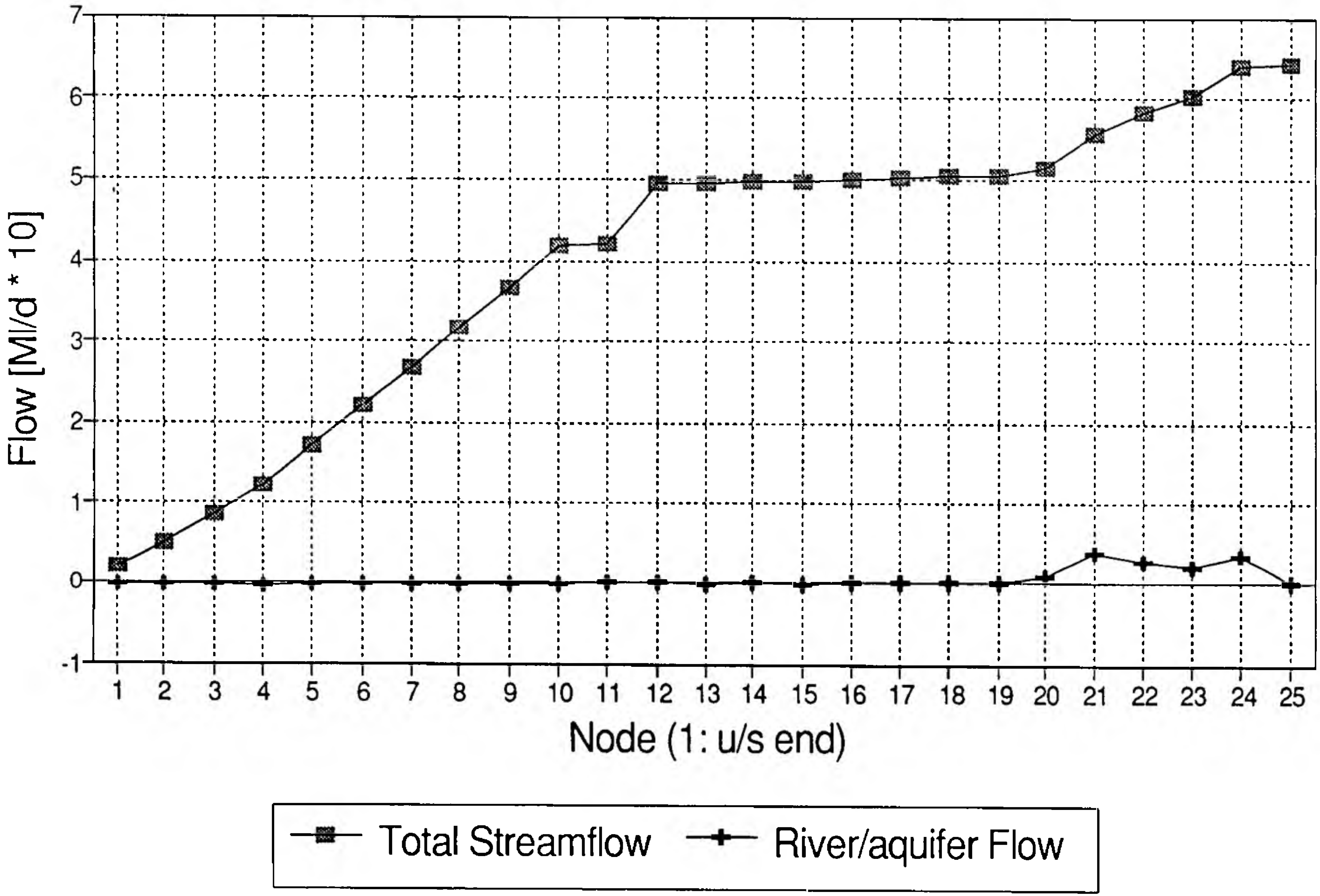
Tuddenham Stream

Flow components: April 1979



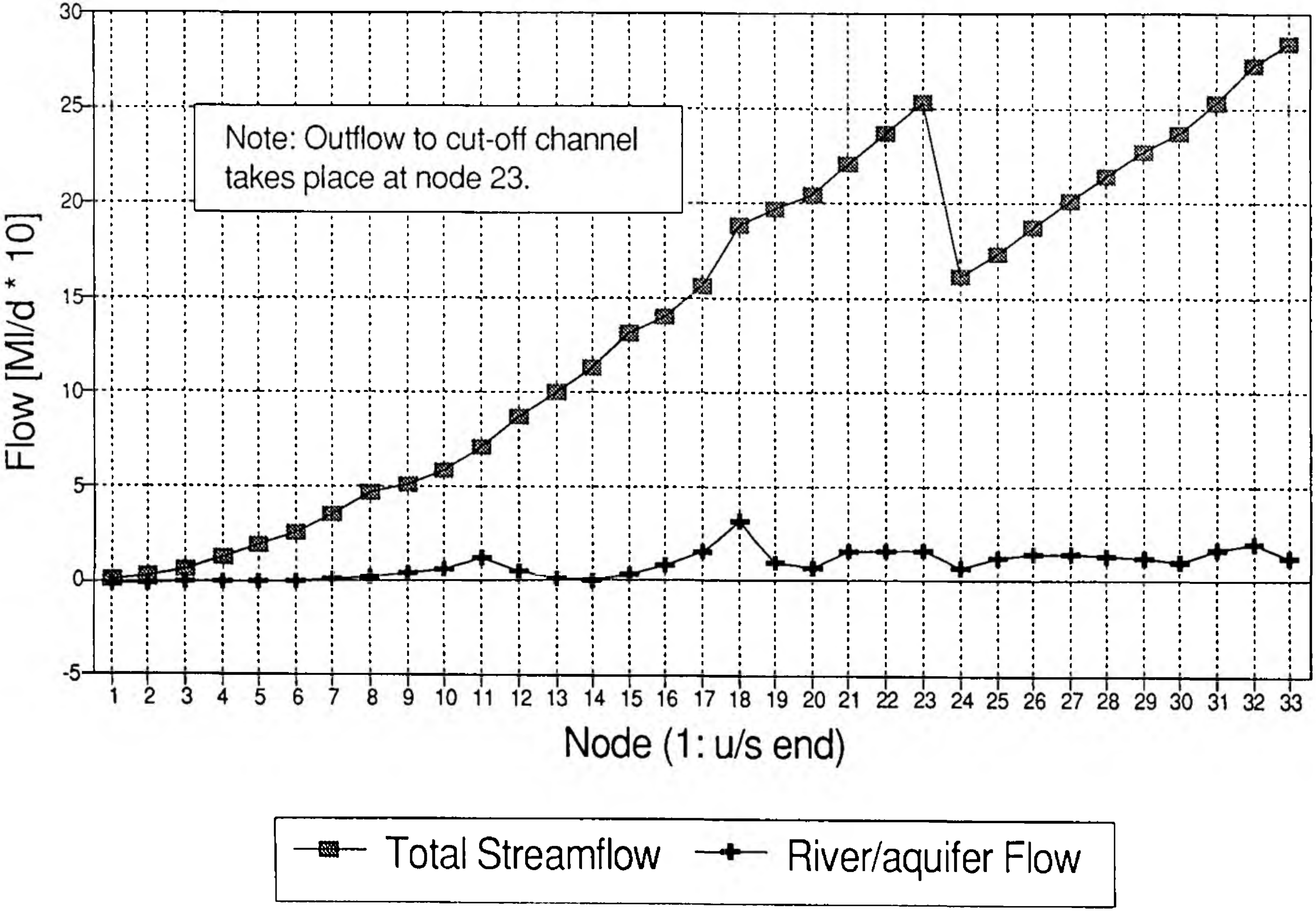
River Kennet

Flow components: April 1979



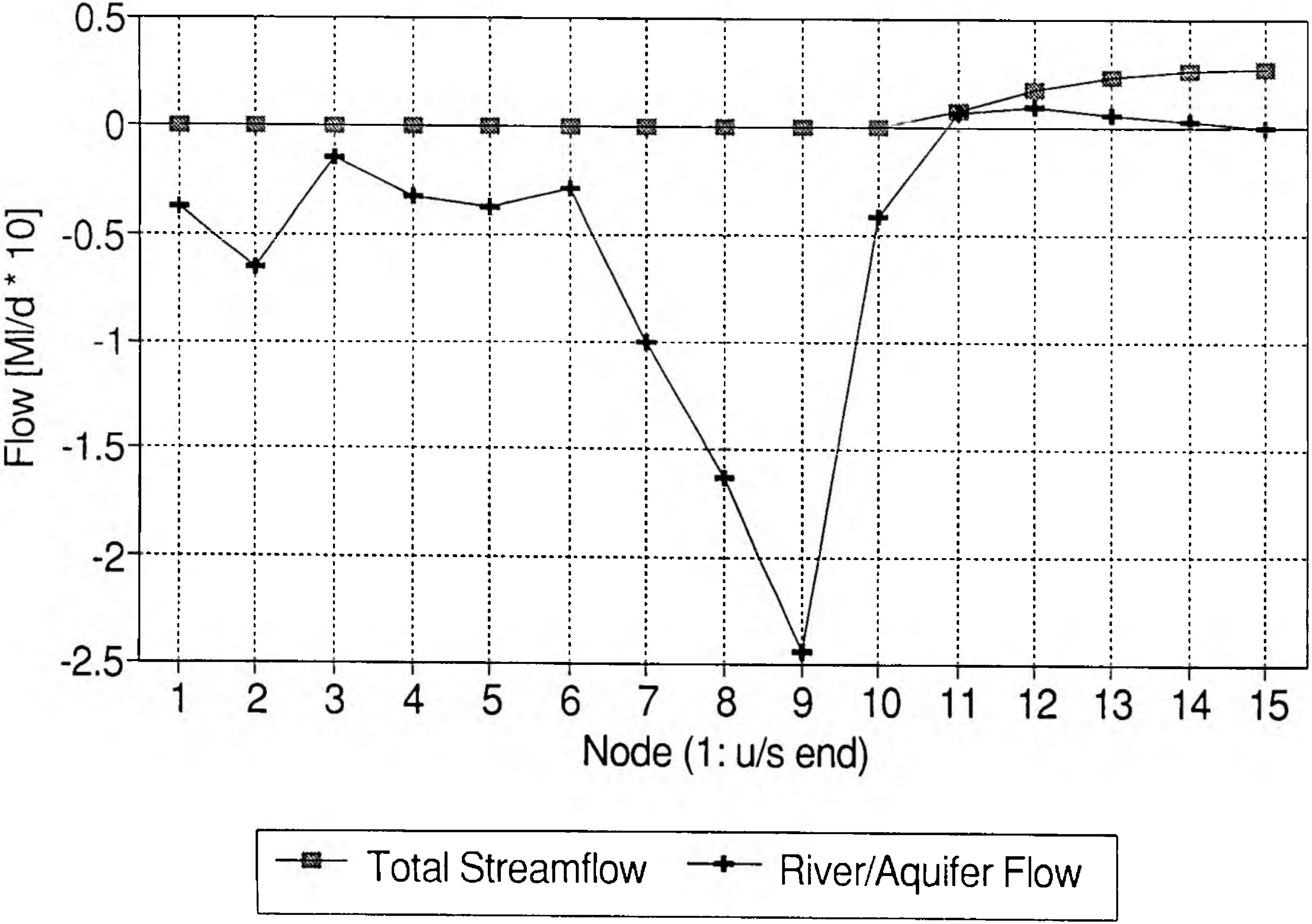
River Lark

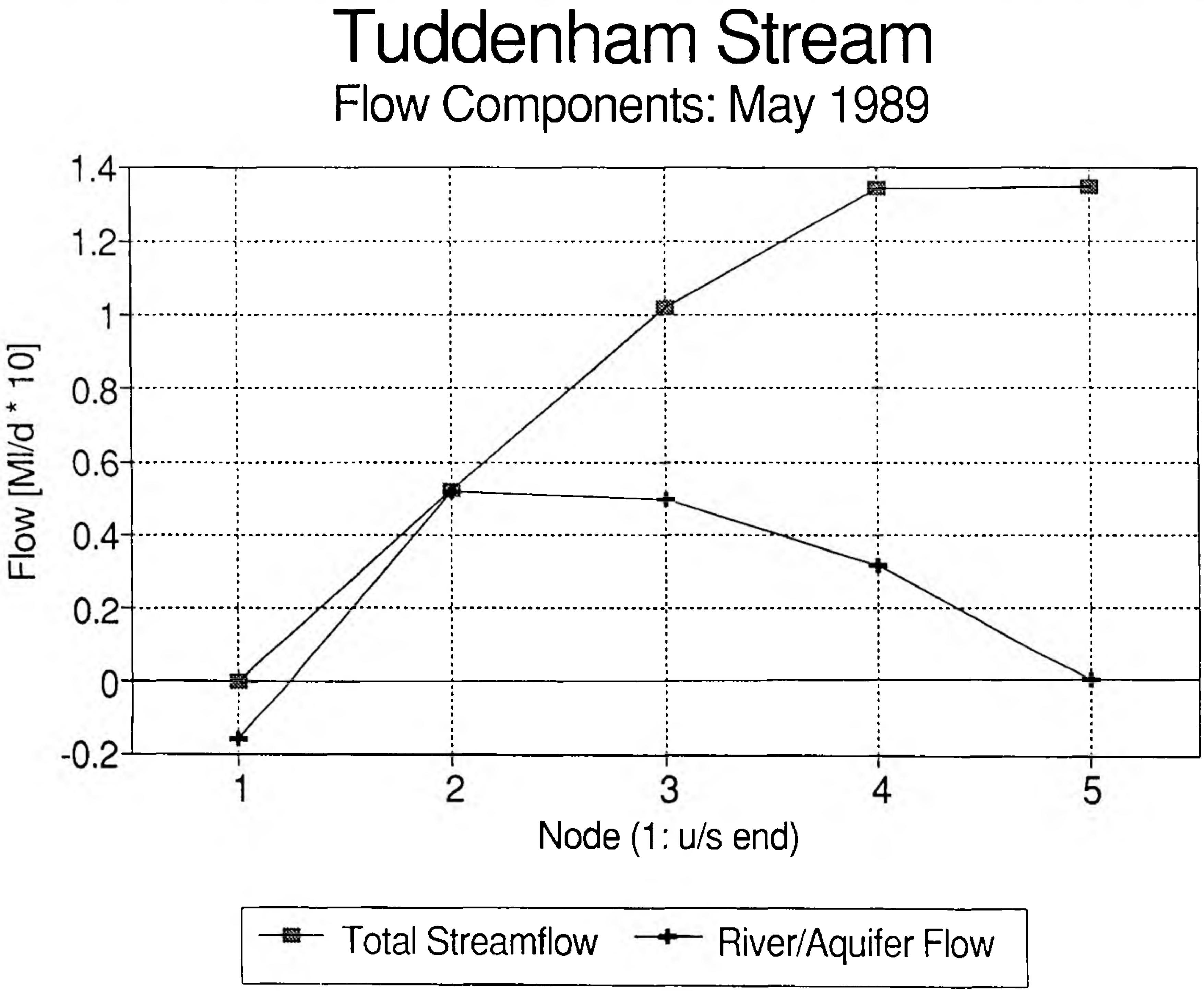
Flow components: April 1979



Cavenham Stream

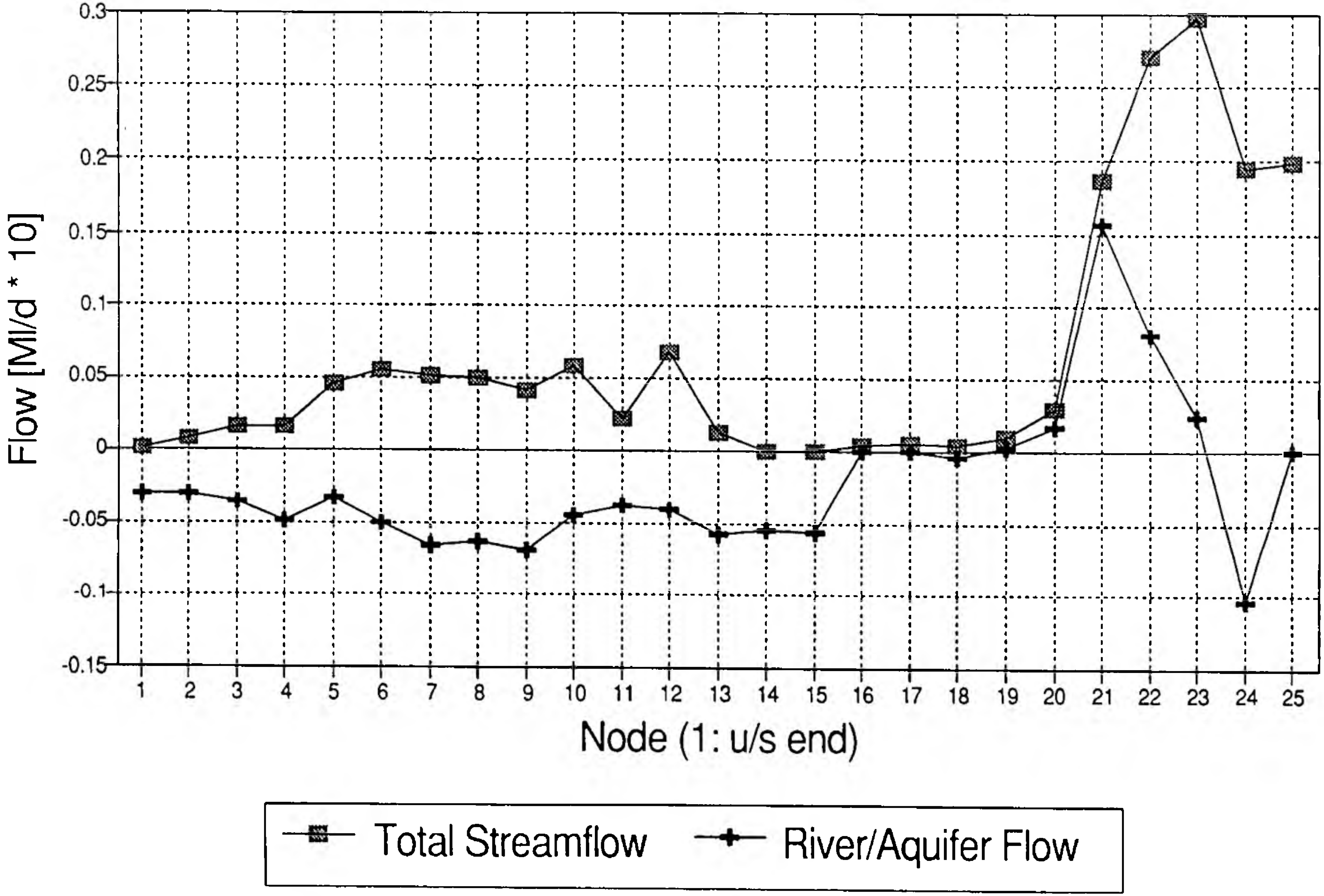
Flow Components: May 1989




Tuddenham Stream
Flow Components: May 1989
Flow [Ml/d * 10]
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
1
2
3
4
5
Node (1: u/s end)
Total Streamflow
River/Aquifer Flow

River Kennet

Flow Components: May 1989



River Lark

Flow Components: May 1989