NRA Anglian 157

DEPARTMENT OF CIVIL ENGINEERING

Water Resources Research Group

Software Profile

A User Manual for TFFOR

UNIVERSITY OF SALFORD

Department of Civil Engineering, University of Salford, Salford, M5 4WT, U.K. Telephone: 061-736 5843. Telex: 68741 (Salcel). Fax: 061-745 7027.

NRA Anglian 157

~~~~~~

TFFOR
(Transfer Function FORecasting)

A program for the validation and testing of real-time transfer function models using detailed error analysis and incorporating the delta updating technique.

~~~~~~

Software Profile

A User Manual for TFFOR

November 1989

Water Resources Research Group
Department of Civil Engineering
University of Salford
Salford
M5 4WT

Report Prepared by

K.A. Tilford
November 1989

Contents

List of Figures

1. Introduction

2. Typography and Flow Chart Symbols

3. Software Specification and System Requirements

4. The Transfer Function Model
   - 4.1. Introduction
   - 4.2. Stage or Flow?
   - 4.3. Baseflow
   - 4.4. Catchment Lag
   - 4.5 Real-Time Updating
   - 4.6. Steady State Gain/Unit Step Response
   - 4.7. Conclusions

5. Program Structure and Data Requirements
   - 5.1. Structure
   - 5.2. Input Datafiles
   - 5.3. Presentation of Results

6. Running the Program

7. Conclusions

Bibliography

Appendices
- Appendix 1 TFFOR Source Code Listing
- Appendix 2 Example Rainfall Datafile
- Appendix 3 Example River Datafile
- Appendix 4 Example Model/Catchment Datafile
- Appendix 5 Example Ouput File
- Appendix 6 Runtime Listing of TFFOR

List of Figures

Figure 1.	Block Diagram Representation of the Transfer Function Model
Figure 2a.	Flowchart for TFFOR Main Body
Figure 2b.	Flowchart for TFFOR Main Body
Figure 3.	Example Model/Catchment Datafile
Figure 4.	Example Rainfall Datafile

1. Introduction

This manual is a report in a series of Technical Reports produced by the Water Resources Research Group at the Department of Civil Engineering, University of Salford.

The manual is a reference to the software package known as TFFOR (Transfer Function FORecasting), a program for the assessment of the forecasting performance of a transfer function rainfall runoff model in simulated real-time and off-line environments. The report begins by stating the software specification and goes on to discuss the input files required. After a brief conceptual introduction to the transfer function model, the structure of TFFOR is described and illustrated with a flowchart and an example run-time session described. Annotated samples of the input datafiles are included.

The Appendices provide a source listing of the program together with a hard copy listing of example input and output datafiles. The datafiles accompany the program on the distribution disk and may be used to replicate the run-time example in the report main body. User input datafiles should exactly replicate the format of the example datafiles.

This manual is not a definitive guide to transfer function models. Further information can be found in the references listed in the bibliography.

The Water Resources Research Group would welcome any comments on this Software Profile. Please contact Professor Ian Cluckie at the address at the front of the report.

2. Typography and Flow Chart Symbols

The body of this manual is printed in a normal (Times font) typeface; other typfaces have special meanings.

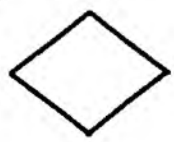
`Courier` is used for the listings of the program, datafiles and screen output. **`Bolded courier`** represents interactive user keyboard input whilst annotated comments of source code and datafile listings are made in **bolded times**.

The program structure is illustrated by a flowchart and described (summarised) textually. Algorithms are described in terms of steps such as input, output and computations. Decisions are made by testing Boolean expressions that are evaluated to be true or false. The flowchart symbols for these processes, along with a symbol to indicate beginning and end are:

Assignments or computations

Input or output

Boolean expressions



Start or stop

3. Software Specification and System Requirements

TFFOR (Transfer Function FORecasting) is an interactive, graphically based FORTRAN program for the verification of the forecasting performance of a transfer function model in a simulated real-time environment.

The software is coded in ANSI FORTRAN 77 and has been developed on a Digital Electronic Company (DEC) MicroVAX II minicomputer using VMS V5.0 and FORTRAN 77 V5.0. The code does not use any non-standard VAX FORTAN 77 implementations (extensions) and is easily ported to a wide range of FORTRAN environments.

Graphics play an integral role in TFFOR and are facilitated by UNIRAS Graphics Software[1] package (Version 5.01, RASPAK module). UNIRAS graphics modules are upwardly compatible with subsequent releases of UNIRAS. UNIRAS graphics modules are machine independent and can be implemented on a wide range of machines. A menu of devices for which the graphics elements of the software have already been implemented prompt the user to indicate the device on which the software is running enabling the correct device driver to be software selected. Implementation for new devices is straightforward if a UNIRAS driver for the device is available.

[1] UNIRAS A.S.,376 Gladsaxevej, DK-2860 Søborg, DENMARK
UNIRAS Ltd, Ambassador House, 181 Farnham Road, Slough, SL1 4XP, UNITED KINGDOM.

4. The Transfer Function Model

This section summarises the lumped transfer function rainfall-runoff model developed by the Water Resources Research Group. The model is parametrically efficient, structurally compact and robust to data loss or error. It is well suited to real-time operational environments and has been widely implemented in the U.K by the Water Authorities.

4.1. Introduction

The lumped transfer function rainfall runoff model uses present and past observed rainfall and flow data to forecast future river flow. Model updating allows the model to update the percentage runoff it represents thereby facilitating an input of total rainfall. These implicitly introduce robustness into the model; the self correction buffering inaccurate forecasts. The feedback of recently observed rainfall/flow data ensures maximum utilisation of telemetry data.

The transfer function model comprises essentially of two components: the flow part (a parameters), and a rainfall component (b parameters) and has a memory for past rainfall and flow values. The structure of the transfer function model is shown in eq. 1:

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \ldots + a_p y_{t-p} + b_1 u_{t-1} + b_2 u_{t-2} + \ldots + b_q u_{t-q} \qquad \text{(eq. 1)}$$

where:

a_i , b_i = model parameters.

y_t = runoff forecast for time t (y_{t-n} = instantaneous observed runoff for time t-n).

u_t = total observed rainfall between time t-1 and time t.

yt = instantaneous observed runoff at time t.

The block diagram representation of the transfer function model highlights the structure of the model and is shown in figure 1 (where z is a backward difference operator such that $u_t z^{-n} = u_{t-n}$).

$$\text{Rainfall Input } u_t \longrightarrow \boxed{\frac{\Delta\left[b_1 z^{-1} + b_2 z^{-2} + , \ldots , + b_q z^{-q}\right]}{1 - a_1 z^{-1} - a_2 z^{-2} - , \ldots , - a_p z^{-p}}} \longrightarrow \text{Runoff Output } y_t$$

FIGURE 1: Block Diagram Representation of the Transfer Function Model

4.2. Stage or Flow?

The model utilises a rainfall-runoff relationship to forecast riverflow. If the river data are in the form of stage measurements, these are converted to flow using the rating equation for the gauging station (often using the equation shown below):

$$Q = a\,(H - h)^{b} \qquad \text{(eq. 2)}$$

where:
Q = river discharge
H = river stage
h = gauging station correction factor for stations where Q is not zero when the stage H=0.
a,b = rating parameters.

4.3. Baseflow

The model forecasts runoff. Baseflow (assumed as riverflow before the event) is subtracted from the river flow data before forecasts are made and is added for graphic presentation and results output.

4.4. Catchment Lag

In some catchment there may be some delay between rainfall and resultant flow and the incorporation of a pure time delay (τ) in the model may be appropriate (eq. 3).

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \ldots + a_p y_{t-p} + b_1 u_{t-1-\tau} + b_2 u_{t-2-\tau} + \ldots b_q u_{t-q-\tau} \qquad \text{(eq. 3)}$$

4.5. Real-Time Updating

The model utilises on-line parameter scaling to change the state of the model through an event. The procedure adapts the magnitude of the model unit input response by scaling the rainfall component of the model (using a factor delta, Δ). One-step ahead forecast errors are used to update delta in an attempt to adapt the percentage runoff of the model in order to minimise the one-step ahead forecast error. The form of transfer function model including on-line updating and a time delay is:

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \ldots + a_p y_{t-p} + \Delta \left[b_1 u_{t-\tau-1} + b_2 u_{t-\tau-2} + \ldots + b_q u_{t-\tau-q} \right] \quad \text{(eq. 4)}$$

The following constraints are applied to Δ in the updating algorithm:

- if river runoff is less than a threshold (i.e. less than 1.2 times river baseflow) delta is not updated (to simulate filling of catchment storage).
- if rainfall in the model memory is below a threshold delta is not updated.
- updated delta can be at most 1.5 times greater or less than the previous delta (an oversensitivity limit to prevent large fluctuations in delta).

The updating of delta (eq. 5) also incorporates a smoothing factor μ, (0≤ μ≥1, usually μ=0.5).

$$\Delta_t = \mu\, \Delta_{t-1} + (1-\mu) \frac{y_t - \left[a_1 y_{t-1} + a_2 y_{t-2} + \ldots + a_p y_{t-p} \right]}{b_1 u_{t-1} + b_2 u_{t-2} + \ldots + b_q u_{t-q}} \quad \text{(eq. 5)}$$

An advantage of on-line updating is that it facilitates an input of total rainfall simultaneously providing a means of coping with events which have a percentage runoff which differ from the (initial) percentage runoff represented by the model.

4.6. Steady State Gain/Unit Step Response

The steady-state gain of a transfer function model is the ratio of (steady) output to a constant input of unit magnitude i.e. a measure of model amplification. In this context steady-state gain is directly analogous to the percentage runoff and it is this which is adjusted through the event by the updating procedure. The steady state gain may be determined directly from the model parameters as shown in eq. 6.

$$SSG = CF \cdot \left\{ \frac{(b_1 + b_2 + \ldots + b_q)}{1 - (a_1 + a_2 + \ldots + a_p)} \right\} \qquad \text{(eq. 6)}$$

where CF is a unit conversion factor to ensure that the SSG is dimensionless.

4.7. Stability

A transfer function model should be implicitly stable i.e. a finite input produces a finite output (the so called BIBO rule - bounded-input bounded output) and the model output should decay with time when there is no rainfall. Stability of transfer function model is a complex issue and the conditions which must be satisfied to guarantee stability are beyond the scope of this report (instead see Box and Jenkins, 1976). However in the vast majority of cases if the condition shown in eq. 10 is satisfied , the model will be stable.

$$\left| \sum_{i=1}^{p} a_i \right| < 1.0 \qquad \text{(eq. 7)}$$

4.8. Conclusions

This section has briefly described the lumped transfer function rainfall-runoff model. A great deal of more detailed documentation exists and is listed in the bibiography.

5. Program Structure and Data Requirements

The determination of a good rainfall-runoff forecasting model is dependent on evolutionary calibration and periodical performance verification. TFFOR enables a transfer function model to be assessed from forecasting performance in a simulated real-time environment and/or simulation in an off-line analytical environment. This chapter describes the theoretical basis of the program, its structure, and the input and output files it uses.

5.1. Structure

The program consists of an initialisation segment which executes sequentially, an options block executed according to the choices seleceted by the user, an output writing segment, and a number of subroutines. All graphics code is contained in subroutines and are called from the main program body. A full source code listing of TFFOR is provided in Appendix 1. The program flowchart in figure 2 illustrates the program structure which is also outlined below:

- Character and array initialisation
- Welcome message
- Determination of current device type (for graphics modules)
- Establishment of input and output datafile names
- Read input data
- Selection of forecasting options
- Model Forecasting/Simulation
- Graphical presentation of results
- Writing of results to output file.

The hydrometric and model/catchment data are read from the three input datafiles. The model produces flow forecasts from a selected point in time for a specified duration or until the forecasted flow has fallen to a pre-defined threshold. The user enters a forecasting time interval (the choice being limited by the interval of the model and of the rain and river data) and selects a future rainfall scenarios (from the unattainable, perfect foresight) to the more realistic (e.g. average of past data). Updating can be deselected if it is not required.

5.2. Input Data Files

The program uses three input data files, example listings of which are given in Appendices 2, 3 and 4. An annotated diagram of an example rainfall/river data file showing the header block and the first few lines of data is shown in figure 3 and for the model datafile in figure 4.

The primary requirements are rainfall and river data for a given period. Care should be taken to ensure the validity of the data and that the datafiles are the same length, in phase, and spaced at a constant time interval.

The rainfall and river data files have an identical format, essentially consisting of a five line header block followed by the data. Essential information i.e. identification title, data type, date and time coding, event length and data time interval (minutes) is held in a five line header block. The data is read in 'free format' (though a format mirroring the example is recommended for user files). The river data can be stage or flow, the program recognising data data type by virtue of a flag in the header block; for the former stage data are converted to flow using stage-discharge relationships stored in the third input datafile.

A model data file contains details of the model, the gauging station and the catchment. The file includes a title line, details of model structure, the model parameters, the model interval, the catchment area (in sq km), an alarm level option used during graphical presentation of forecasts, and the gauging station rating equation.

5.3. Presentation of Results

Results can be optionally presented graphically and/or in a tabular output file.

The graphic output is in three parts:

- interactive forecasting/simulation production
- retrospective forecasted hydrographs for set forecast lead-times and respective errors.
- display of updating parameter Δ and one step-ahead forecasting residuals.

Forecasting results can be optionally written to an textual output results file. A listing of an example ouput file is provided in Appendix 5. In summary, the following are written to the file.

- title
- structure and parameters of the model used for forecasting
- storm event rainfall and river flow data
- storm event characteristics: total depth of rain; mean hourly rainfall; maximum hourly rainfall; maximum river discharge; approximate percentage runoff.
- forecasts for 1,2,4,6, and 8 model interval lead-times are listed at forecasting interval intervals; the value of delta at each time step is also listed.
- the root mean square errors of the 1 to 8 model interval lead-time forecasts.

6. Running the Program

The following is summary of the options that confront a user when the program runs. A run-time listing of the program user-interface during program execution is provided in Appendix 6. A series of prompts sequentially leads the user through a program initialisation phase establishing filenames, reading input data, and opening a results output file. Referral to the program flow chart (figure 2) may aid the reader.

Device Type:
At present about five devices are supported by TFFOR. If you wish to run the software on another device, contact the Water Resources Research Group indicating the UNIRAS driver name for the device. This will then be incorporated in the code for a subsequent release version.

Name of Model/Rain/River Datafile:
The approriate names should be entered.

Forecast interval:
The time intervals of the data and the model are read by the program and form the basis of the suggested forecasted intervals. The smallest forecasting interval allowable is that of the data (rainfall or river) with the smallest time interval.

Output file:
If an output file is required the name is entered.

Program Menu:
The program menu is the main control point of the program.

- 'Event Simulation' applies the model in a pure simulation mode, i.e. no feedback of past rainfall or flow values and 'perfect foresight of rainfall' (see below). Useful for assessing model representativeness.
- 'Forecasting with a specified lead-time' produces forecasts from selected moments in time for a selected lead-time (see below). Feedback of information is used in this forecasting mode. Corresponds to real-time forecasting.
- 'Error/Delta Plots' are post event analysis routines which plot: the forecasting errors for set forecast lead-times (2, 4, 6 and 8 model interval lead-times) and the observed and forecasted hydrographs; the value of the updating parameter Δ throught the event and the one-step ahead error.
- 'STOP Program' terminates the program, writing forecasting performance statistics to the output datafile if this option was selected.

Future Rainfall Scenario:

The quality of the rainfall forecast is of high importance when multiple step-ahead forecasts are being produced. The program facilitates several options from 'no more rain' through 'past rain averaged', and 'perfect foresight to a 'user defined input'. Until quantitative precipitation forecasts are routinely available from the Meteorological Office FRONTIERS system, the last option may be of most use taking the form of x mm of rainfall in y hours.

Delta:

The only initialisation the model requires is an initial value of delta. The program specifies maximum and minimum values which correspond to 0% and 100% runoff, and the user can select any value between these limits. Model updating is not mandatory and can be overidden by using a constant delta.

The program is invoked by entering `Run TFFOR`. After each response the return (enter) key is pressed, the display scrolls and the next prompt is displayed. The reader is referred to Appendix 5 for the runtime listing. When the initialisation is completed control passes to the graphics subroutines and then back to the Program Menu. If the stop option is selected from the main option menu forecasting results are written to an output file before the program terminates.

To summarise the options selected during the example run:

Device driver selected=VT Emulator (i.e. Regis graphics). Input model datafile=fot3.mod. Input rainfall datafile= cs3ver.rai. Input river datafile=cs3ver.sta. Interval of rainfall data=15 minutes. Interval of river data=15 minutes, Model interval=60 minute. Selected forecasting interval=60 minutes. An output file (test.out) is required. Forecasting with a lead-time of ten model intervals (hours) using perfect foresight of future rainfall. Updating is required using an inital delta value 2.0.

7. Conclusions

This report is a users guide to the FORTRAN software package TFFOR, a program for off-line verification of the forecasting/simulation performance of a lumped transfer function rainfall-runoff model. The report contains listings of full source code and input and output datafiles; all of which are contained on the software distribution disk.

The transfer function model has been developed over a number of years by Professor Ian Cluckie and his research team, and is currently being implemented for a number of Water Authorities in the U.K. The model is well documented and the explanatory notes in the document are kept to a minimum. A bibliography lists references where further information can be sourced.

TFFOR is an interactive, user-friendly program featuring user selected options and a powerful graphics front-end. The structure of the program is described and illustrated with a flowchart. Data requirements are explictly described. Results may be presented graphically and/or written to a results output file.

A runtime listing is provided and described in the text, and the user options are described.

A bibliography is provided if further information on related topics is required.

Bibliography

Box. G.E.P. and Jenkins. G.M. (1976). Time Series Analysis: Forecasting and Control, Holden-Day Inc.

Cluckie, I.D. and Han, D. (1989). "Radar Data Quantisation, Sampling and Preliminary Model Assessment using Upavon Data", Wessex Radar Information Project, Report no. 4.

Cluckie, I.D. and Tilford, K.A. (1988). "An Evaluation of the Influence of Radar Rainfall Intensity Resolution for Real-Time Operational Flood Forecasting", Anglian Radar Information Project, Report no. 2.

Cluckie, I.D. and Tilford, K.A. (1989). "Transfer Function Models for Flood Forecasting in Anglian Water Authority", Anglian Radar Information Project, Report no. 3.

Cluckie, I.D., Yu, P.S. and Tilford, K.A. (1989). "Real-Time Forecasting: Model Structure and Data Resolution", Proc. Seminar on Weather Radar Networking, Brussels, Belgium, 1989.

Cluckie, I.D., Tilford, K.A., and Shepherd, G.W. (1989). "Radar Rainfall Quantisation and its Influence on Rainfall Runoff Models", Proc. Int. Symp. on Hyd. App. of Weather Radar, University of Salford, U.K, August.

Cluckie, I.D., Tilford, K.A., and Han, D. (1989). "The Influence of Radar Rainfall Quantisation on Flood Forecasting", to be presented at Int. Workshop on Precipitation Measurement, St. Moritz, Switzerland, December.

Cluckie, I.D. and Yu, P.S. (1988) "Stochastic Models for Real-Time Riverflow Forecasting Utilising Radar Data", 5th IAHR International Symposium on Stochastic Hydraulics, University of Birmingham, U.K. August.

Owens. M.D. (1986). "Real-Time Flood Forecasting Using Weather Radar Data", Ph.D Thesis, University of Birmingham, Department of Civil Engineering.

Tilford. K.A. (1989). "Software Profile - A User Manual for TFCAL", University of Salford, Department of Civil Engineering.

Tilford. K.A. (1989). "Software Profile - A User Manual for TFUH", University of Salford, Department of Civil Engineering.

Tilford. K.A. (1989). "Software Profile - A User Manual for AWSTAGE", University of Salford, Department of Civil Engineering.

Tilford. K.A. (1989). "Software Profile - A User Manual for RADRAIN", University of Salford, Department of Civil Engineering.

Yu. P.S. (1989). "Real-Time Grid-Based Distributed Rainfall-Runoff Model for Flood Forecasting with Weather Radar", Ph.D Thesis, University of Birmingham, Department of Civil Engineering.

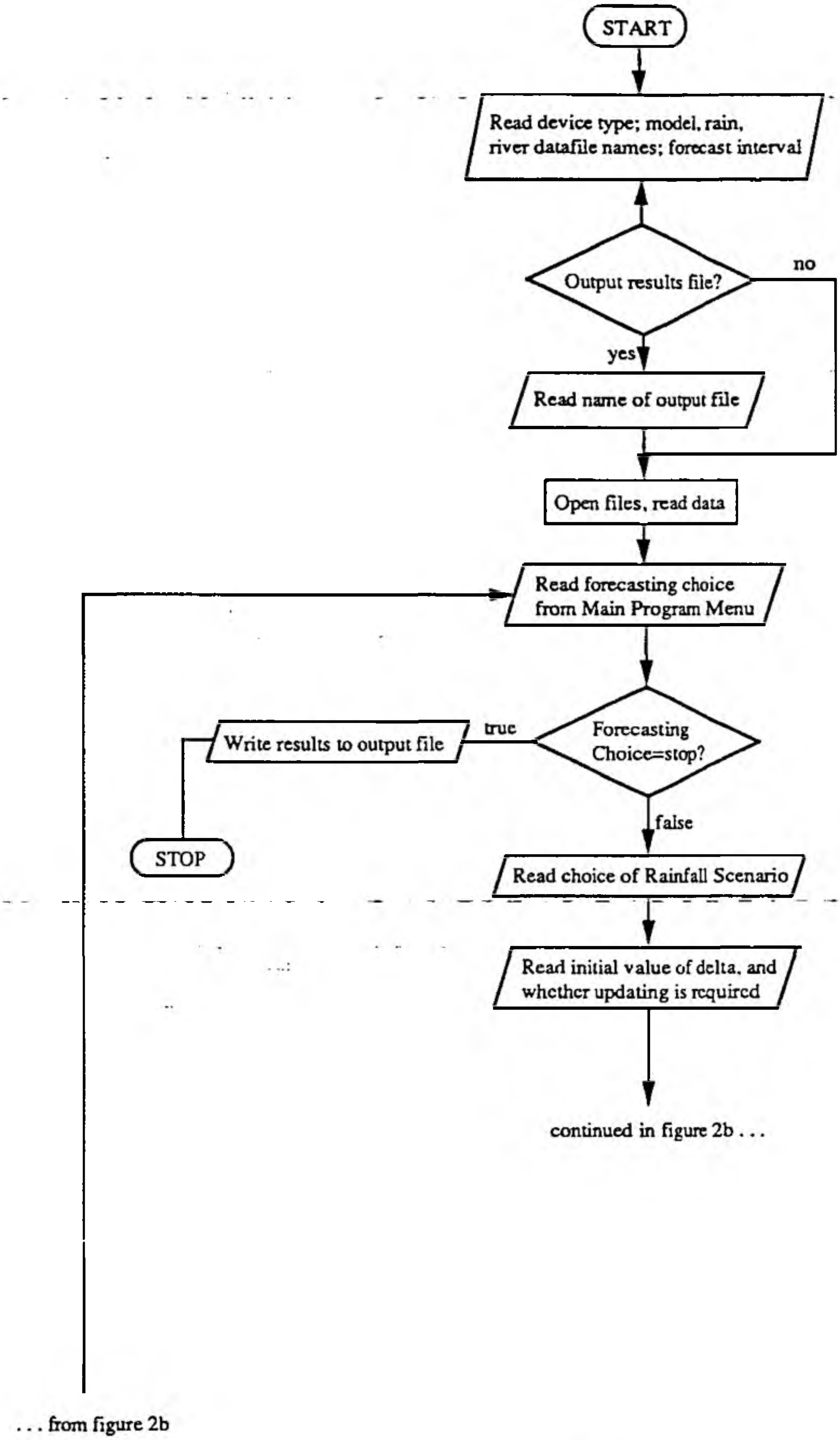


FIGURE 2a: Flow Chart for TFFOR Main Body

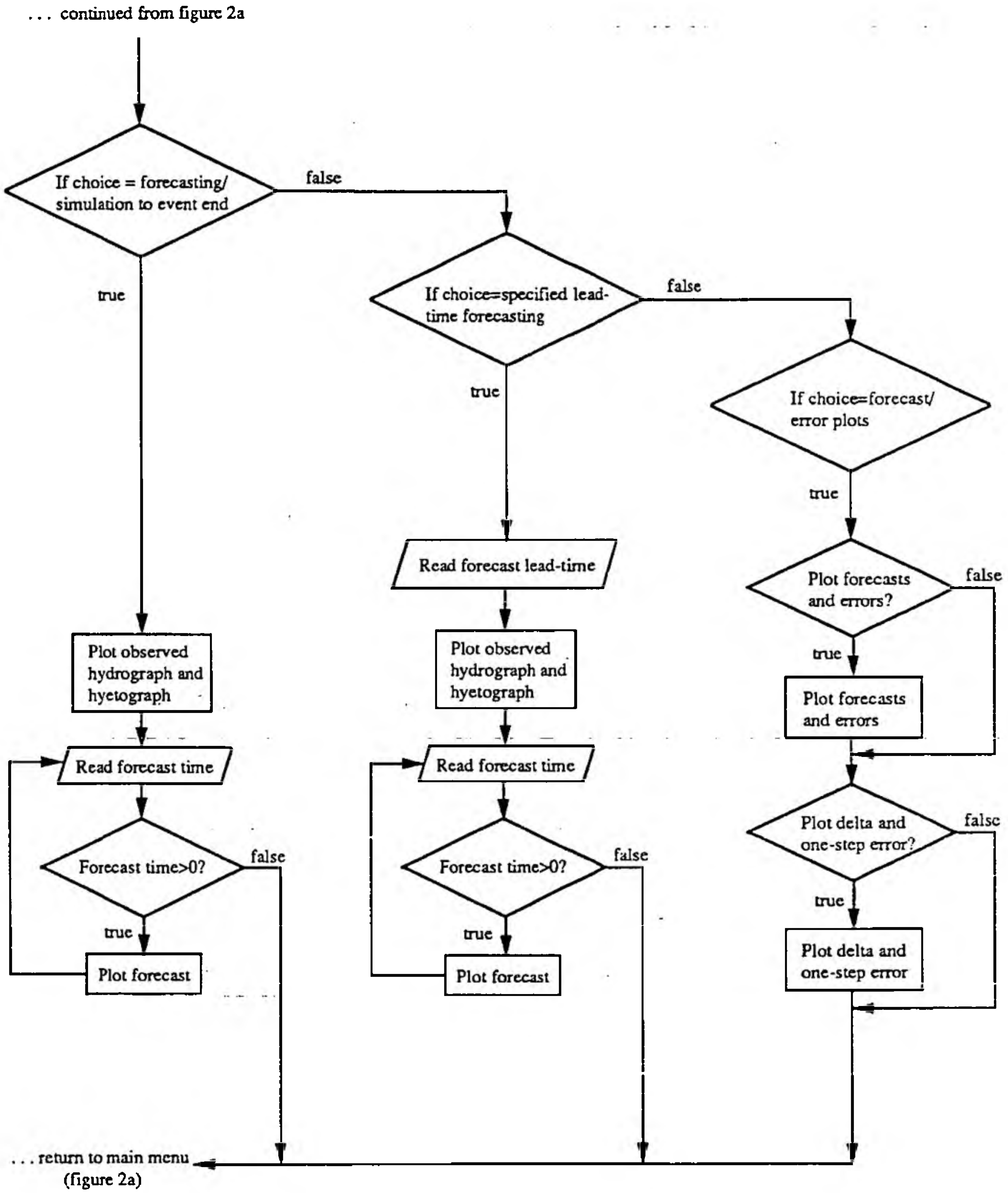


FIGURE 2b: Flow Chart for TFFOR Graphics Subroutines

Title block (A60): any useful information

```
Fotheringhay (Willow Brook) Model Data File (use cal_test3)
3,5,0
1.9566 -1.2581  0.2884  0.0113 -0.0067  0.0108  0.0074  0.0211
60
85.25
1
0
4
0.1370,  3.2142,  0.0012,  1.5413
0.3770,  2.9942,  0.0003,  1.5002
1.2940,  3.0029, -0.0009,  1.4984
5.0000,  2.9858,  0.0031,  1.5030
```

Model structure a,b,τ → 3,5,0

Model parameters ← 1.9566 -1.2581 0.2884 0.0113 -0.0067 0.0108 0.0074 0.0211

Model interval (minutes) → 60

85.25 ← Catchment area (sq km)

Number of alarm levels (if none, =1) → 1

Alarm level (in metres stage) → 0

Number of segments in rating equation → 4

Rating equation: Maximum stage (m), a,h and b → the four rows of the rating table

FIGURE 3: Example Model/Catchment Datafile

Data Type Flag
(RAIN, STAGE or DISCHARGE)
First 2 characters are read and are preceeded by 2 spaces i.e. 2X, A4

Title block (A60): any useful information

```
Willow Brook, FOTHERINGHAY (for verifying cal3 model)

RAIN

9999,99,99,99

1, 448

15
   0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
   0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
   0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
   0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
   0.000     0.000     0.000     0.000     0.000     0.000     0.000     0.000
   0.000     0.020     0.062     0.011     0.003     0.000     0.000     0.000
   0.000     0.026     0.071     0.106     0.000     0.001     0.005     0.028
```

Time and Data Code
(hhmm, DD, MM, YY)

Number of Storms
(Redundant so =1)

Number of Data in File

Data interval (minutes)

First 7 Lines of Data
(read in free format)

FIGURE 4: Example Rainfall Datafile

Appendices

Appendix 1 TFFOR Source Code Listing
Appendix 2 Example Rainfall Datafile
Appendix 3 Example River Datafile
Appendix 4 Example Model/Catchment Datafile
Appendix 5 Example Ouput File
Appendix 6 Runtime Listing of TFFOR

Appendix 1: TFFOR Source Code Listing

```
c
c  ----------------------------------------------------------------------------------
c  ----------------------------------------------------------------------------------
c
c                                        PROGRAM TFFOR
c
c  UNIRAS RASPAK Version 2.0b
c  v1.00  5th August 1989
c  v1.01 22nd August 1989
c  v2.00b 25th October 1989
c
c  ----------------------------------------------------------------------------------
c
c  A program for verification of the forecasting performance of an lumped
c  transfer function rainfall-runoff model.
c
c Water Resources Research Group
c Department of Civil Engineering
c University of Salford
c SALFORD
c M5 4WT
c
c For further information contact:
c          Prof. Ian Cluckie
c
c  ----------------------------------------------------------------------------------
c  ----------------------------------------------------------------------------------
c
c
c
      dimension rain_temp(2000),rain(2000),f_rain(2000)
      dimension river_temp(2000),runoff(2000),river_flow(2000)
      dimension delta(2000),f_flow(2000),temp(2000)
      dimension fore(8,2000),error(8,2000),p(40)
      dimension rmse(8)
      dimension stage_max(10),alpha(10),beta(10),gamma(10)
      dimension decision_level(10)
c
      real mean_intensity,numerator
      integer model_order, future_rain, choice
      integer rain_time, rain_day, rain_month, rain_year, rain_storm
      integer river_time, river_day, river_month, river_year, river_storm
c
      character*4 sod
      character*3 karacter,kar
      character*1 out_file,mono,ansd,anss,ansdel,ansfe
      character*60 rain_title,river_title,model_title
      character*60 model_file,rain_file,river_file,output file
c
      logical exist1
c
  10 format (a60)
  15 format (2x,a4)
  30       format (i4)
  40 format (a1)
c
c
c -------------------------------------------------------
c
c
c Establish device name for graphics output
c
```

```
      write (*,*)
      write (*,*)' -----------------------------------------------
     #-------------------'
      write (*,*)'  The UNIRAS graphics routines in this program are
     # device independent.'
      write (*,*)' -----------------------------------------------
     #-------------------'
  696      write (*,*)'  Please type in the integer corresponding to the
     #device you wish'
      write (*,*)'  graphics to be directed to;'
      write (*,*)
      write (*,*)'      (1) VAXstation'
      write (*,*)'      (2) VT Emulator (eg VT340)'
      write (*,*)'      (3) Pen Plotter'
      write (*,*)'      (4) Ink Jet Printer'
      write (*,*)'      (5) IBM PS2'
      write (*,*)'  Please type in choice [1,2,3,4, or 5]'
      read (*,30,err=696)idevice
      if (idevice.gt.5) goto 696
c
c  call logo(idevice)
c
  111      write (*,*)
      write (*,*)'  Type in name of model datafile '
      read (*,10)model_file
      open (unit=3,status='old',file=model_file,err=111)
  112      write (*,*)
      write (*,*) '  Type in name of rain datafile '
      read (*,10)rain_file
      open (unit=1,status='old',file=rain_file,err=112)
  113      write (*,*)
      write (*,*) '  Type in name of river datafile '
      read (*,10)river_file
      open(unit=2,status='old',file=river_file,err=113)
c
c   Read rain/river
c
      read(1,10)rain_title
      read(1,*)
      read(1,*)rain_time,rain_day,rain_month,rain_year
      read(1,*)rain_storm,num_rain
      read(1,*)int_rain
c
      read(2,10)river_title
      read(2,15)sod
      read(2,*)river_time,river_day,river_month,river_year
      read(2,*)river_storm,num_river
      read(2,*)int_river
c
      read(3,10)model_title
      read(3,*)num_p,num_q,lag
      model_order=num_p+num_q
      read(3,*)(p(i),i=1,model_order)
      read(3,*)int_model
      read(3,*)catchment_area
      read(3,*)num_decisions
      read(3,*)(decision_level(i),i=1,num_decisions)
      if(sod.eq.'stag'.or.sod.eq.'STAG') then
         read(3,*)num_segments
         do 201 i=1,num_segments
            read(3,*)stage_max(i),beta(i),alpha(i),gamma(i)
  201          continue
      end if
c
c  Read rainfall and river data and check model parameters
c
      read(1,*)(rain_temp(i),i=1,num_rain)
      read(2,*)(river_temp(i),i=1,num_river)
c
      total_rainp=0.0
      do 202 i=1,num_p
         total_flowp=total_flowp+p(i)
```

```
 202     continue
   if (total_flowp.gt.1.0) stop '-- ar gt 1.0'
   total_rainp=0.0
   do 203 i=num_p+1,model_order
      total_rainp=total_rainp+p(i)
 203     continue
   if (total_rainp.lt.0.0) stop '--ma -ive effect'
c
   close (unit=1)
   close (unit=2)
   close (unit=3)
c
   write(*,41)int_rain
 41 format(3x,'Rainfall data interval ',i3,' minutes')
   write(*,42)int_river
 42 format(3x,'River data interval     ',i3,' minutes')
   write(*,43)int_model
 43 format(3x,'Model interval          ',i3,' minutes')
   int_min=int_rain
   if (int_river.lt.int_rain) int_min=int_river
   if (int_model.lt.int_min) int_min=int_model
   write(*,*)
   write(*,*)'  Select a forecast interval:'
   write(*,44)int_min
 44 format(5x,i3,' minutes')
   write(*,46)int_min*2
 46 format(5x,i3,' minutes')
   write(*,47)int_min*3
 47 format(5x,i3,' minutes')
   write(*,48)int_min*4
 48 format(5x,'Other [enter integer, multiple of ',i3,']')
   read(*,*)int_fore
c
   num=num_river/(int_fore/int_river)
c
c  Convert stage to flow, manipulate data, remove baseflow from riverflow
c
   if (sod.eq.'STAG'.or.sod.eq.'stag') then
      do 204 i=1,num_river
         do 49 j=1,num_segments
            if (river_temp(i).lt.stage_max(j)) goto 49
 49      continue
         j=j-1
            river_temp(i)=((river_temp(i)+alpha(j))**gamma(j))*beta(j)
 204        continue
   end if
c
c  Remove baseflow
c
   base=river_temp(1)
   do 206 i=1,num_river
      runoff(i)=river_temp(i)-base
 206     continue
c
c  Manipulate data
c
   k=1
   store=0.0
   ifactor=int_fore/int_rain
   do 207 i=1,num_rain,ifactor
      do 208 j=1,ifactor
         store=store+rain_temp((j-1)+i)
 208        continue
      rain(k)=store
      k=k+1
      store=0.0
 207     continue
c
   k=1
   ifactor=int_fore/int_river
   do 209 i=1,num_river,ifactor
```

```
          river_flow(k)=river_temp(i+ifactor-1)
          runoff(k)=runoff(i+ifactor-1)
          k=k+1
 209      continue
c
c  Establish initial delta and delta throughout event
c
          gain=(total_rainp/(1-(total_flowp)))*
     #                        (0.06*float(int_fore)/catchment_area)
          delta_max=1.0/gain
          delta_min=0.05/gain
c
c  Determine if a results output file is necessary
c
 59   write (*,*)
      write (*,*)'  Is an output file required? [Y or N]'
      read (*,40,err=59)out_file
      if (out_file.ne.'Y'.and.out_file.ne.'y'.and.
     #              out_file.ne.'n'.and.out_file.ne.'N') goto 59
      if (out_file.eq.'y'.or.out_file.eq.'Y') then
 674        write (*,*)'  Type in name of output file '
         read (*,10)output_file
         inquire(file=output_file,exist=exist1)
         if (.not.exist1) goto 673
              write(*,*)'File already exists . . .'
            goto 674
 673        open (unit=4,status='new',file=output_file)
      end if
c
c  Determine maximum flow and time of occurence, and maximum rainfall
c
      ipeak=0
      flowmax=0.0
      rainmax=0.0
      total_rain=0.0
      total_runoff=0.0
      max_intensity=0.0
      do 210 i=1,num
         if (river_flow(i).gt.flowmax) then
            flowmax=river_flow(i)
            ipeak=i
         end if
         if (rain(i).gt.rainmax) rainmax=rain(i)
         total_rain=total_rain+rain(i)
         total_runoff=total_runoff+runoff(i)
 210      continue
      mean_intensity=total_rain/float(num)
      rain_volume=total_rain*catchment_area*1000.0
      total_runoff=total_runoff*float(60*int_fore)
      percentage_runoff=(total_runoff/rain_volume)*100.0
c
c
c  --------------------------------------------------------------------
c
 61   write(*,*)
      write(*,*)
      write(*,*)'  Program Menu'
      write(*,*)'  ------------'
      write(*,*)
      write(*,*)'  (1) Event Simulation'
      write(*,*)'  (2) Forecasting with Specified Lead-Times'
      write(*,*)'  (3) Error/Delta Plots'
      write(*,*)'  (0) STOP Program'
 212      write(*,*)
      write(*,*)'  Enter integer corresponding to choice [1,2,3 or 0]'
      write(*,*)
      read(*,*,err=212)choice
      if (choice.gt.3) goto 61
      if (choice.eq.0) goto 888
      if (choice.eq.1) goto 767
c
 27   write(*,*)
```

```
      write(*,*)'  Please select future rainfall scenario:'
      write(*,*)'    (1) No more rain'
      write(*,*)'    (2) Past rain averaged'
      write(*,*)'    (3) FRONTIERS rainfall forecast [not
     # implemented]'
      write(*,*)'    (4) Perfect foresight of rainfall'
      write(*,*)'    (5) User defined [not implemented]'
      write(*,*)'  Enter integer corresponding to choice [1,2,
     # 3,4,or 5]'
      read(*,30,err=27)future_rain
      if (future_rain.gt.5) goto 27
         write(*,*)
 52      write(*,51)delta_min,delta_max
 51      format(3x,'Enter an initial value of delta (real) between
     # ',f4.2,' and ',f4.2)
         read(*,*,err=52)first_delta
         if (first_delta.lt.delta_min.or.first_delta.gt.delta_max)
     #           goto 52
         delta(1)=first_delta
 53      write(*,*)'  Is delta to be kept constant? [Y or N]'
         read(*,40,err=53)ansd
         if (ansd.ne.'Y'.and.ansd.ne.'y'.and.ansd.ne.'n'.
     #                                 and.ansd.ne.'N') goto 53
c        if (ansd.eq.'n'.or.ansd.eq.'N') then
c 34           write(*,*)'  Is a monotonic delta required?  [Y or N]'
c           read(*,40)mono
c           if (mono.ne.'Y'.and.mono.ne.'y'.and.mono.ne.'n'.
c     #                              and.mono.ne.'N') goto 34
c        end if
         if (ansd.eq.'Y'.or.ansd.eq.'y') then
            do 211 i=1,num
               delta(i)=first_delta
 211              continue
          else
            smooth=0.5
            do 237 i=2,num
               if (river_flow(i).lt.(2.0*base)) then
                  delta(i)=delta(i-1)
                  goto 54
               end if
               suma=0.0
               do 213 j=1,num_p
                  suma=suma+(p(j)*runoff(i-j))
 213                 continue
               sumb=0.0
               sum_rain=0.0
               do 214 j=num_p+1,model_order
                  sumb=sumb+(p(j)*rain(i-j+num_p-lag))
                  sum_rain=sum_rain+rain(i-j+num_p-lag)
 214                 continue
               if (sum_rain.lt.1.0) then
                  delta(i)=delta(i-1)
                  goto 54
               end if
               numerator=runoff(i)-suma
               if (numerator.lt.0.1) numerator=0.1
               delta(i)=(smooth*delta(i-1))+
     #                     ((1.0-smooth)*(numerator/sumb))
                  if (delta(i)/delta(i-1).gt.1.5)
     #                              delta(i)=delta(i-1)*1.5
                  if (delta(i-1)/delta(i).gt.1.5)
     #                              delta(i)=delta(i-1)/1.5
               if (delta(i).gt.delta_max)
     #                              delta(i)=delta_max
               if (delta(i).lt.delta_min)
     #                              delta(i)=delta_min
 54         continue
c
c              if (mono.eq.'y'.or.mono.eq.'Y') then
c                 if (delta(i).lt.delta(i-1)) delta(i)=delta(i-1)
c              end if
 237              continue
```

```
      end if
c
c    -   - - -        - - - - - - -
      if (choice.eq.1) then
c
c     ----------------------------------------------------------------
c
c     !! SIMULATION TO EVENT END (i.e. no feedback, perfect foresight)
c
c
 767      call dataplot(idevice,river_flow,num,rain,flowmax,rainmax)
c
 74   call gdiags(0,0,60,1)
      call gdiagw(52,'Enter value for delta [integer],
     # (zero [0] to stop)',-1)
      call gdiagr(k1,karacter,k2)
      call toint(karacter,idelta)
      if (idelta.eq.0) goto 61
      karacter=' '
      if (idelta.lt.delta_min.or.idelta.gt.delta_max) goto 74
c
      do 215 i=1,num
         f_flow(i)=0.0
 215     continue
      do 217 i=1,num
         f_flow(i)=0.0
         do 218 j=1,num_p
           f_flow(i)=f_flow(i)+(p(j)*f_flow(i-j))
 218        continue
         do 219 j=num_p+1,model_order
                f_flow(i)=f_flow(i)+(p(j)*rain(i-j+num_p-lag)*idelta)
 219        continue
 217     continue
      open (unit=11,name='simula.dat',status='new')
      do 220 i=1,num
         f_flow(i)=f_flow(i)+base
            write(11,*)f_flow(i)
 220     continue
c
      call plot1(idevice,num,f_flow,flowmax)
      goto 74
c
c
      else if (choice.eq.2) then
c
c
c     ----------------------------------------------------------------
c
c     !! FORECASTING WITH FINITE TIME FRAME, feedingback flow forecasts !!
c
c
 70   write(*,*)
      write(*,*)' Enter required forecast lead-time [in hours]'
      read(*,30,err=70)lead_time
      if (lead_time.gt.num*int_fore) then
         write(*,*)' Lead-time is too large'
         goto 70
      end if
      lead_time=(lead_time*60)/int_fore
c
      call dataplot(idevice,river_flow,num,rain,flowmax,rainmax)
c
 71   call gdiags(0,0,60,1)
      call gdiagw(43,'Enter time for forecast, (zero [0] to stop)',-1)
      call gdiagr(k3,karacter,k4)
      call toint(karacter,istart)
      karacter=' '
      if (istart.gt.num.or.istart.eq.0) goto 61
c
c     Calculate forecasted flows
c
      call rainscen(num,future_rain,istart,rain,f_rain)
```

```
      do 221 i=1,num
         f_flow(i)=0.0
221      continue
      do 222 i=1,istart
         f_flow(i)=runoff(i)
222      continue
      do 223 i=(istart+1),(istart+lead_time)
         f_flow(i)=0.0
         do 224 j=1,num_p
            f_flow(i)=f_flow(i)+(p(j)*f_flow(i-j))
224         continue
         do 225 j=num_p+1,model_order
        f_flow(i)=f_flow(i)+(p(j)*f_rain(i-j+num_p-lag)*
     #                                    delta(istart))
225         continue
         if (f_flow(i).lt.0.0) f_flow(i)=0.0
223      continue
      do 226 i=istart,istart+lead_time
         f_flow(i)=f_flow(i)+base
226      continue
c
         call plot2(idevice,num,istart,lead_time,f_flow,flowmax)
c
      goto 71
c
c
c
      else if (choice.eq.3) then
c
c ------------------------------------------------------------------
c
c  Calculate 1-8 step-ahead forecast residuals and root mean square errors
c
888      do 227 i=1,num
         temp(i)=0.0
227      continue
      do 228 k=1,num
         do 229 i=1,k
            temp(i)=runoff(i)
229         continue
         call rainscen(num,future_rain,k,rain,f_rain)
         do 230 i=2+(k-1),9+(k-1)
            temp(i)=0.0
            do 231 j=1,num_p
               temp(i)=temp(i)+(p(j)*temp(i-j))
231            continue
            do 232 j=num_p+1,model_order
               temp(i)=temp(i)+(p(j)*f_rain(i-j+num_p)*delta(k))
232            continue
            if (temp(i).lt.0.0) temp(i)=0.0
            fore(i-k,k)=temp(i)+base
            error(i-k,k)=river_flow(i)-fore(i-k,k)
230         continue
228      continue
      do 233 j=1,8
         sum=0.0
         do 234 i=1,num-j
            sum=sum+(error(j,i)*error(j,i))
234         continue
         rmse(j)=sqrt(sum/float(i))
233      continue
      if (choice.eq.0) goto 889
c
c  Plot errors if required
c
      write(*,*)
      write(*,*)'  Do you want to plot forecasts and errors? [Y or N]'
      read(*,40)ansfe
      if (ansfe.eq.'y'.or.ansfe.eq.'Y') then
            call plotferr(idevice,num,river_flow,
```

```
     #                                              fore,error,flowmax)
      end if
c
c   Plot delta if required
c
 88 write(*,*)
      write(*,*)'  Is a plot of delta required? [Y or N]'
      read(*,40,err=88)ansdel
      if (ansdel.ne.'Y'.and.ansdel.ne.'y'.and.
     #                    ansdel.ne.'n'.and.ansdel.ne.'N') goto 88
      if (ansdel.eq.'y'.or.ansdel.eq.'Y')
     #    call plotdelt(idevice,num,delta,delta_max,error)
c
c
       else
         write(*,*)' Problem, Control should never reach here'
      end if
c
c
c   Write results to output file
c
 889      if (out_file.eq.'y'.or.out_file.eq.'Y') then
c
      write(4,*)
      write(4,*)'   Model Forecast Verification Results'
      write(4,*)'   ----------------------------------'
      write(4,*)
      write(4,*)
      write(4,447)num_p,num_q,lag
 447      format (3x,'Forecasting with Model Order',(i3,',',i3,',',i3))
      write(4,*)
      write(4,*)'  Model Parameters '
      write(4,467)(i,p(i),i=1,num_p)
      write(4,468)(i-num_p,p(i),i=num_p+1,model_order)
 467      format(4x,'a',i2,')',1x,f7.4)
 468      format(4x,'b',i2,')',1x,f7.4)
      write(4,*)
      write(4,*)
      write(4,*)'   Actual Subcatchment Averaged Rainfall (mm)'
      write(4,437)(rain(i),i=1,num)
      write(4,*)
      write(4,*)'   River Discharge (cumecs)'
      write(4,437)(river_flow(i),i=1,num)
 437      format (8(1x,f7.3))
      write(4,*)
      write(4,*)
      write(4,*)
      write(4,438)total_rain
      write(4,439)mean_intensity
      write(4,440)rainmax
      write(4,441)flowmax
      write(4,442)base
      write(4,443)percentage_runoff
 438      format(3x,' Total depth of rain     :',f8.2)
 439      format(3x,' Mean hourly rainfall    :',f8.2)
 440      format(3x,' Maximum hourly rainfall :',f8.2)
 441      format(3x,' Maximum river discharge :',f8.2)
 442      format(3x,' River baseflow          :',f8.2)
 443      format(3x,' Percentage runoff       :',f8.2)
c
      write(4,*)
      write(4,*)
      if (future_rain.eq.1) write(4,*)'  Forecasts: No More Rainfall'
      if (future_rain.eq.2) write(4,*)'  Forecasts: Average of Past
     # Rainfall'
      if (future_rain.eq.3) write(4,*)'  Forecasts: FRONTIERS Forecast'
      if (future_rain.eq.4) write(4,*)'  Forecasts: Perfect Foresight'
      if (future_rain.eq.5) write(4,*)'  Forecasts: Other Forecast'
      write(4,*)
      write(4,*)'  Time  Delta     Flow       1      2       4      6
     #      8'
      do 235 i=1,num
```

```
         write (4,469)i,delta(i),river_flow(i),fore(1,i),fore(2,i),
     #                             fore(4,i),fore(6,i),fore(8,i)
 235     continue
 469     format(4x,i3,3x,f4.2,3x,(f7.1,1x,5(2x,f5.1)))
c
      write (4,*)
      write (4,*)
      write (4,*)'   Forecast Root Mean Square Errors (RMSE)'
         write(4,4017)(k,rmse(k),k=1,8)
 4017    format (8(/3x,i2,' Step-Ahead ',f10.3))
c
      write (4,*)
      write (4,*)
      write (4,*)'   Forecast Errors of 2,4,6 and 8 Step-Ahead Forecasts
     # (cumecs)'
      write (4,*)
      write (4,*)'  Time        2          4          6          8 '
      do 236 i=1,num
         write (4,4018)i,error(2,i),error(4,i),error(6,i),error(8,i)
 236     continue
 4018    format(4x,i3,4(5x,f6.2))
      close (unit=4)
      end if
c
c
      if (choice.gt.0) goto 61
      call gclose
c
      stop
      end
```

```
c
c
c
c
c
c    ---------------------------------------------------------------------
c         Subroutine to compute future rainfall scenario
c    ---------------------------------------------------------------------
c
      subroutine rainscen(num,future_rain,istart,rain,f_rain)
c
      dimension rain(2000),f_rain(2000)
      integer future_rain
c
      if (future_rain.eq.4) then
c Perfect foresight
         do 10 k=1,num
            f_rain(k)=rain(k)
 10      continue
       else if (future_rain.eq.3) then
c FRONTIERS
         write(*,*)' Sorry, no FRONTIERS yet'
       else if (future_rain.eq.2) then
c PAST RAIN AVERAGED
         sum_rain=0.0
         do 11 k=1,istart
            sum_rain=sum_rain+rain(k)
            f_rain(k)=rain(k)
 11      continue
         apr=sum_rain/istart
         do 12 k=istart+1,num
            f_rain(k)=apr
 12      continue
       else
         do 13 k=1,istart
c NO MORE RAIN
            f_rain(k)=rain(k)
 13      continue
         do 14 k=istart+1,num
            f_rain(k)=0.0
 14      continue
      end if
c
      return
c
      end
```

```
c
c
c
c ---------------------------------------------------------------------
c       Subroutine to convert character to integer
c ---------------------------------------------------------------------
c
      subroutine toint (karac,kint)
c
      character*3 karac
c
      kint=0
      ires1=ichar(karac(1:1))
      ires2=ichar(karac(2:2))
      ires3=ichar(karac(3:3))
c
      if (ires1.eq.48) goto 6
      if (ires3.gt.32.and.ires2.gt.32.and.ires1.gt.32) then
         kint=(ires3-48)+((ires2-48)*10)+((ires1-48)*100)
c 100-999
        else if ((ires3.eq.32.or.ires3.eq.0).and.(ires2.gt.32)) then
         kint=(ires2-48)+((ires1-48)*10)
c 10-99
        else if ((ires3.eq.32.or.ires3.eq.0).and.
     #                      (ires2.eq.32.or.ires2.eq.0)) then
         kint=ires1-48
c 1-9
        else
         write(*,*)' Problem!, control should never reach here!'
      end if
c
         karac=' '
c
    6 return
c
      end
```

```
c
c
c
c ---------------------------------------------------------------------
      subroutine dataplot (idevice,river_flow,num,rain,fmax,rmax)
c     -----------------------------------------------------------------
c
      dimension river_flow(2000),rain(2000)
c
      if (idevice.eq.1) call groute('select mgpx;exit')
      if (idevice.eq.2) call groute('select mregis;exit')
      if (idevice.eq.3) call groute('select mhpgl;exit')
      if (idevice.eq.4) call groute('select glj250;exit')
      if (idevice.eq.5) call groute('select lvga;exit')
      call gopen
c
      call grpsiz(xdim,ydim)
c xdim,ydim are device window sizes
c
c   River flow axes (code=1)
c
c  origin of x-axis
      xor1=xdim*0.125
c  origin of y-axis
      yor1=ydim*0.16
c  end of x-axis
      xend1=xdim*0.83
c  end of y-axis
      yend1=ydim*0.59
c  length of x-axis
      xlen1=xend1-xor1
c  length of y-axis
      ylen1=yend1-yor1
c
c   Rainfall axes (code=2)
c
      xor2=xdim*0.125
      yor2=ydim*0.66
      xend2=xdim*0.83
      yend2=ydim*0.84
      xlen2=xend2-xor2
      ylen2=yend2-yor2
c
      xinc=xlen1/float(num)
      yinc1=ylen1/(1.05*fmax)
      yinc2=ylen2/(1.05*rmax)
c
      call gcharc(3)
c
c    Label hydrograph axes
c
      do 10 i=0,5
              call gnumb(num/5.*i,xor1+(xlen1*(i/5.))-2.5,(yor1-7.),3.,-1)
   10 continue
      do 11 i=0,4
         call gnumb((1.05*fmax)*(i/4.0),(xor1-11.0),
     #                              yor1+(ylen1*(i/4.0))-3.0.3.0,1)
   11 continue
c
c    Label rainfall y axis
c
      do 12 i=0,4
         call gnumb((1.05*rmax)*(i/4.0),(xor2-9.0),
     #                               yor2+(ylen2*(i/4.0))-2.0,3.0,1)
   12 continue
c
c    Draw event hydrograph
c
      call gvect(xor1+xinc,yor1+(river_flow(1)*yinc1),0)
      call gwicol(0.2,5)
      do 13 i=2,num
         call gvect(xor1+(i*xinc),yor1+(river_flow(i)*yinc1),1)
```

```
   13 continue
c
c   Draw rainfall hyetograph
c
      call gvect(xor2,yor2,0)
      call gwicol(0.2,5)
      do 14 i=1,num
         call gvect(xor2+(i*xinc),yor2,0)
         call gvect(xor2+(i*xinc),yor2+(rain(i)*yinc2),1)
   14 continue
c
      call gwicol(0.2,3)
c
c   Draw riverflow axes
c
      call gvect(xorl,yorl,0)
      call gvect(xendl,yorl,1)
      call gvect(xorl,yorl,0)
      call gvect(xorl,yendl,1)
c
c   Draw rainfall hyetograph axes
c
      call gvect(xor2,yor2,0)
      call gvect(xend2,yor2,1)
      call gvect(xor2,yor2,0)
      call gvect(xor2,yend2,1)
c
c   Label axes and add title
c
      call gwicol(0.2,3)
      call gchara(90)
      call gchar('Discharge (cumecs)$',xdim*0.07,ydim*0.21,3.5)
      call gchara(90)
      call gchar('Rain (mm)$',xdim*0.07,ydim*0.66,3.5)
      call gchar('Time (hours)$',xdim*0.4,ydim*0.05,3.5)
c
c   if (idevice.eq.4) then
c      call qlj250
c   end if
c
      return
      end
```

```
c
c
c
c    ------------------------------------------------------------------
      subroutine plot1 (idevice,num,f_flow,fmax)
c    ------------------------------------------------------------------
c
      dimension f_flow(2000)
c
      if (idevice.eq.1) call groute('select mgpx;exit')
c     if (idevice.eq.2) call groute('select mregis;exit')
      if (idevice.eq.3) call groute('select mhpgl;exit')
      if (idevice.eq.4) call groute('select glj250;exit')
      if (idevice.eq.5) call groute('select lvga;exit')
c
      call grpsiz(xdim,ydim)
c    xdim,ydim are device window sizes
c
c
      xor1=xdim*0.125
c    origin of x-axis
      yor1=ydim*0.16
c    origin of y-axis
      xend1=xdim*0.83
c    end of x-axis
      yend1=ydim*0.59
c    end of y-axis
      xlen1=xend1-xor1
c    length of x-axis
      ylen1=yend1-yor1
c    length of y-axis
c
      xinc=xlen1/float(num)
      yinc1=ylen1/(1.05*fmax)
c
      call gvect (xor1+(xinc),yor1+(yinc1*f_flow(1)),0)
      call gwicol(0.2,2)
      do 10 k=istart+1,num
         call gdash(5)
         call gvect(xor1+(xinc*k),yor1+(f_flow(k)*yinc1),1)
   10 continue
c
c     if (idevice.eq.4) then
c        call qlj250
c     end if
c
      return
      end
```

```
c
c
c ---------------------------------------------------------------------
      subroutine plot2(idevice,num,istart,lead_time,f_flow,fmax)
c ---------------------------------------------------------------------
c
      dimension f_flow(2000)
c
      if (idevice.eq.1) call groute('select mgpx;exit')
      if (idevice.eq.2) call groute('select mregis;exit')
      if (idevice.eq.3) call groute('select mhpgl;exit')
      if (idevice.eq.4) call groute('select glj250;exit')
      if (idevice.eq.5) call groute('select lvga;exit')
      call grpsiz(xdim,ydim)
c xdim,ydim are device window sizes
c
      xorl=xdim*0.125
c origin of x-axis
      yorl=ydim*0.16
c origin of y-axis
      xendl=xdim*0.83
c end of x-axis
      yendl=ydim*0.59
c end of y-axis
      xlenl=xendl-xorl
c length of x-axis
      ylenl=yendl-yorl
c length of y-axis
      xinc=xlenl/float(num)
      yincl=ylenl/(1.05*fmax)
c
      call gvect (xorl+(xinc*istart),yorl+(yincl*f_flow(istart)),0)
      if (istart+lead_time.gt.num) then
         lead_time=num-istart
      end if
      lead_time_temp=lead_time
      call gwicol(0.2,2)
      do 10 k=(istart+1),(istart+lead_time)
         call gdash(5)
         call gvect(xorl+(xinc*k),yorl+(f_flow(k)*yincl),1)
 10   continue
      lead_time=lead_time_temp
c
c     if (idevice.eq.4) then
c        call qlj250
c     end if
c
      return
      end
```

```
c
c
c
c  -----------------------------------------------------------------
   subroutine plotdelt(idevice,num,delta,delta_max,error)
c  -----------------------------------------------------------------
c
   dimension delta(2000),error(8,2000)
c
   if (idevice.eq.1) call groute('select mgpx;exit')
   if (idevice.eq.2) call groute('select mregis;exit')
   if (idevice.eq.3) call groute('select mhpgl;exit')
   if (idevice.eq.4) call groute('select glj250;exit')
   if (idevice.eq.5) call groute('select lvga;exit')
   call gopen
c
   call grpsiz(xdim,ydim)
c  xdim,ydim are device window sizes
c
   errmax=0.0
   do 10 i=1,num-1
      if (abs(error(1,i)).gt.errmax) errmax=abs(error(1,i))
 10 continue
c
c   Delta axes
c
   xor=xdim*0.125
c  origin of x-axis
   yor=ydim*0.36
c  origin of y-axis
   xend=xdim*0.83
c  end of x-axis
   yend=ydim*0.9
c  end of y-axis
   xlen=xend-xor
c  length of x-axis
   ylen=yend-yor
c  length of y-axis
   xinc=xlen/float(num)
   yinc=ylen/delta_max
   yor2=ydim*0.2
   ylen2=ydim*0.2
   yinc2=ylen2/(2.0*errmax)
c
c   Draw delta
c
   call gwicol(0.2,5)
   call gvect(xor,yor+(delta(1)*yinc),0)
   do 11 i=2,num
      call gvect(xor+(i*xinc),yor+(delta(i)*yinc),1)
 11 continue
c
c   Label delta axes
c
   call gcharc(3)
   do 12 i=0,5
      call gnumb((num/5.0)*i,
     #                       xor+(xlen*(i/5.0))-3.0,(yor-7.0),3.0,1)
 12 continue
   do 13 i=0,4
      call gnumb((1.05*delta_max)*(i/4.0),(xor-11.0),
     #                         yor+(ylen*(i/4.0))-3.0,3.0,1)
 13      continue
   do 14 i=0,4
      call gnumb(100.0*(i/4.0),(xend+5.5),
     #                         yor+(ylen*(i/4.0))-2.0,3.0,-1)
 14 continue
c
c   Draw and label delta axes
c
   call gwicol(0.2,3)
   call gvect(xor,yor,0)
```

```
      call gvect(xend,yor,1)
      call gvect(xend,yend,1)
      call gvect(xor,yor,0)
      call gvect(xor,yend,1)
      call gchara(90)
      call gchar('Delta$',xdim*0.07,ydim*0.6,3.5)
      call gchara(90)
      call gchar('Percentage Runoff$',xdim*0.88,ydim*0.53,3.5)
      call gchar('Time (hours)$',xdim*0.4,ydim*0.07,3.5)
c
c   Draw and label error axes
c
      call gwicol(0.2,3)
      call gvect(xor,yor2,0)
      call gdash(5)
      call gvect(xend,yor2,1)
      call gvect(xor,0.1*ydim,0)
      call gvect(xor,0.3*ydim,1)
      call gcharc(3)
      call gnumb(1.05*errmax,(xor-11.0),yor2+(ylen2*0.5)-2.0,3.0,1)
      call gnumb(errmax*0,xor-11.0,yor2,3.0,-1)
          call gnumb(-1.05*errmax,(xor-11.),yor2+(-ylen2*0.5)-2.,3.0,1)
      call gchara(90)
      call gchar('Error (cumecs)$',xdim*0.07,ydim*0.12,3.5)
      call gchar('One step-ahead errors$',xdim*0.68,ydim*0.26,3.5)
c
c   Draw one step-ahead error
c
      call gvect(xor+xinc,yor2,0)
      call gwicol(0.2,2)
      do 15 i=1,num-1
        call gvect(xor+((i+1)*xinc),yor2+(error(1,i)*yinc2),1)
   15 continue
c
c  if (idevice.eq.4) then
c     call qlj250
c  end if
c
      call gchar(' Return to continue$',3.0,0.95*ydim,4.5)
      read(*,*)
      call gclose
      return
      end
```

```
c
c
c
c     ------------------------------------------------------------------
      subroutine plotferr(idevice,num,river_flow,
     #                                          fore,error,flowmax)
c     ------------------------------------------------------------------
c
      dimension river_flow(2000),error(8,2000),fore(8,2000)
c
      if (idevice.eq.1) call groute('select mgpx;exit')
      if (idevice.eq.2) call groute('select mregis;exit')
      if (idevice.eq.3) call groute('select mhpgl;exit')
      if (idevice.eq.4) call groute('select glj250;exit')
      if (idevice.eq.5) call groute('select lvga;exit')
      call gopen
      call grpsiz(xdim,ydim)
c
c      Calculate error and flow maxima
c
      errmax=0.0
      fmax=flowmax
      do 10 j=2,8,2
         do 11 i=1,num-j
            if (abs(error(j,i)).gt.errmax) errmax=abs(error(j,i))
            if (fore(j,i).gt.fmax) fmax=fore(j,i)
  11     continue
  10 continue
c
c      Coordinates for axes
c
      xor1=xdim*0.125
      xor2=xdim*0.6
      xend1=xdim*0.5
      xend2=xdim*0.975
      xlen=xend1-xor1
      yor1=ydim*0.1
      yor2=ydim*0.175
      yor3=ydim*0.555
      yor4=ydim*0.63
      yend2=ydim*0.475
      yend4=ydim*0.93
      ylen=yend2-yor2
      xinc=xlen/float(num)
      yinc1=(0.15*ydim)/(2.2*errmax)
      yinc2=ylen/(1.1*fmax)
c
c      Draw axes
c
      call gwicol(0.2,3)
      call gvect(xor1,ydim*0.025,0)
      call gvect(xor1,yend2,1)
      call gvect(xor1,ydim*0.48,0)
      call gvect(xor1,yend4,1)
      call gvect(xor2,ydim*0.025,0)
      call gvect(xor2,yend2,1)
      call gvect(xor2,ydim*0.48,0)
      call gvect(xor2,yend4,1)
      call gvect(xor1,ydim*0.1,0)
      call gdash(5)
      call gvect(xend1,ydim*0.1,1)
      call gvect(xor2,ydim*0.1,0)
      call gdash(5)
      call gvect(xend2,ydim*0.1,1)
      call gvect(xor1,ydim*0.555,0)
      call gdash(5)
      call gvect(xend1,ydim*0.555,1)
      call gvect(xor2,ydim*0.555,0)
      call gdash(5)
      call gvect(xend2,ydim*0.555,1)
      call gvect(xor1,ydim*0.175,0)
      call gvect(xend1,ydim*0.175,1)
```

```
      call gvect(xor2,ydim*0.175,0)
      call gvect(xend2,ydim*0.175,1)
      call gvect(xor1,ydim*0.63,0)
      call gvect(xend1,ydim*0.63,1)
      call gvect(xor2,ydim*0.63,0)
      call gvect(xend2,ydim*0.63,1)
c
c   Draw 8,6,4 and 2 step-ahead errors
c
      call gwicol(0.2,5)
      call gvect(xor2+(8.0*xinc),yor1,0)
      do 12 i=1,num-8
         call gvect(xor2+((i+8)*xinc),yor1+(error(8,i)*yinc1),1)
   12 continue
c
      call gvect(xor1+(6*xinc),yor1,0)
      do 13 i=1,num-6
         call gvect(xor1+((i+6)*xinc),yor1+(error(6,i)*yinc1),1)
   13 continue
c
      call gvect(xor2+(4*xinc),yor3,0)
      do 14 i=1,num-4
         call gvect(xor2+((i+4)*xinc),yor3+(error(4,i)*yinc1),1)
   14 continue
c
      call gvect(xor1+(2*xinc),yor3,0)
      do 15 i=1,num-2
         call gvect(xor1+((i+2)*xinc),yor3+(error(2,i)*yinc1),1)
   15 continue
c
c   Draw observed hydrograph (4 times)
c
      call gwicol(0.2,5)
      call gvect(xor2,yor2+(river_flow(1)*yinc2),0)
      do 16 i=2,num
         call gvect(xor2+(xinc*(i-1)),yor2+(river_flow(i)*yinc2),1)
   16 continue
c
      call gvect(xor1,yor2+(river_flow(1)*yinc2),0)
      do 17 i=2,num
         call gvect(xor1+(xinc*(i-1)),yor2+(river_flow(i)*yinc2),1)
   17 continue
c
      call gvect(xor2,yor4+(river_flow(1)*yinc2),0)
      do 18 i=2,num
         call gvect(xor2+(xinc*(i-1)),yor4+(river_flow(i)*yinc2),1)
   18 continue
c
      call gvect(xor1,yor4+(river_flow(1)*yinc2),0)
      do 19 i=2,num
         call gvect(xor1+(xinc*(i-1)),yor4+(river_flow(i)*yinc2),1)
   19 continue
c
c   Draw 8,6,4 and 2 step-ahead forecasted hydrographs
c
      call gwicol(0.2,2)
c
      call gvect(xor1+(2*xinc),yor4+(fore(2,1)*yinc2),0)
      do 20 i=2,num-2
         call gvect(xor1+((i+1)*xinc),yor4+(fore(2,i)*yinc2),1)
   20 continue
c
      call gvect(xor2+(4*xinc),yor4+(fore(4,1)*yinc2),0)
      do 21 i=2,num-4
         call gvect(xor2+((i+3)*xinc),yor4+(fore(4,i)*yinc2),1)
   21 continue
c
      call gvect(xor1+(6*xinc),yor2+(fore(6,1)*yinc2),0)
      do 22 i=2,num-6
         call gvect(xor1+((i+5)*xinc),yor2+(fore(6,i)*yinc2),1)
   22 continue
c
```

```
      call gvect(xor2+(8*xinc),yor2+(fore(8,1)*yinc2),0)
      do 23 i=2,num-8
         call gvect(xor2+((i+7)*xinc),yor2+(fore(8,i)*yinc2),1)
 23   continue
c
c     Label axes
c
      do 24 i=0,2
         call gnumb(1.05*fmax*i/2,xor1-9.0,
     #                           yor2+(ylen*i/2)-3.0,3.0,1)
         call gnumb(1.05*fmax*i/2,xor1-9.0,
     #                           yor4+(ylen*i/2)-3.0,3.0,1)
         call gnumb(1.05*fmax*i/2,xor2-9.0,
     #                           yor2+(ylen*i/2)-3.0,3.0,1)
         call gnumb(1.05*fmax*i/2,xor2-9.0,
     #                           yor4+(ylen*i/2)-3.0,3.0,1)
 24   continue
c
      call gnumb(errmax*0,xor1-9.0,yor1-3.0,3.0,1)
      call gnumb(errmax*0,xor1-9.0,yor3-3.0,3.0,1)
      call gnumb(errmax*0,xor2-9.0,yor1-3.0,3.0,1)
      call gnumb(errmax*0,xor2-9.0,yor3-3.0,3.0,1)
c
      call gchara(90)
      call gchar('Error$',xdim*0.07,ydim*0.07,3.5)
      call gchara(90)
      call gchar('Error$',xdim*0.07,ydim*0.52,3.5)
      call gchara(90)
      call gchar('Discharge$',xdim*0.07,ydim*0.27,3.5)
      call gchara(90)
      call gchar('Discharge$',xdim*0.07,ydim*0.74,3.5)
      call gchar('Time (hours)$',xdim*0.27,ydim*0.03,3.5)
      call gchar('Time (hours)$',xdim*0.74,ydim*0.03,3.5)
      call gchar('8 step-ahead$',xdim*0.86,ydim*0.44,3.5)
      call gchar('6 step-ahead$',xdim*0.38,ydim*0.44,3.5)
      call gchar('4 step-ahead$',xdim*0.86,ydim*0.9,3.5)
      call gchar('2 step-ahead$',xdim*0.38,ydim*0.9,3.5)
c
c     Put time on hydrograph axes
c
      do 25 i=1,2
         call gnumb((num/2.0)*i,
     #               xor1+(xlen*(i/2.0))-3.0,(yor2-7.0),3.0,-1)
         call gnumb((num/2.0)*i,
     #               xor2+(xlen*(i/2.0))-3.0,(yor2-7.0),3.0,-1)
         call gnumb((num/2.0)*i,
     #               xor1+(xlen*(i/2.0))-3.0,(yor4-7.0),3.0,-1)
         call gnumb((num/2.0)*i,
     #               xor2+(xlen*(i/2.0))-3.0,(yor4-7.0),3.0,-1)
 25   continue
c
c     if (idevice.eq.4) then
c        call qlj250
c     end if
c
      call  gchar(' Return to continue$',3.0,0.95*ydim,4.5)
      read(*,*)
      call gclose
      return
      end
```

```
c
c
c
c ---------------------------------------------------------------
      subroutine logo (idevice)
c     -----------------------------------------------------------
c
      if (idevice.eq.1) call groute('select mgpx;exit')
      if (idevice.eq.2) call groute('select mregis;exit')
      if (idevice.eq.3) call groute('select mhpgl;exit')
      if (idevice.eq.4) call groute('select glj250;exit')
      if (idevice.eq.5) call groute('select lvga;exit')
      call gopen
c
      call gwicol(.5,1)
      call gvect(80.,147.,0)
      call gvect(122.,147.,1)
      call gwicol(1.,1)
      call gvect(80.,145.,0)
      call gvect(122.,145.,1)
c
      call gcharc(1)
      call gchar('UNIVERSITY$',80.,138.,5.)
      call gcharc(2)
      call gchar('SALFORD$',88.,130.,5.)
c
      call gchara(90)
      call gcharc(1)
      call gchar('OF$',85.,130.,3.4)
c
      call gvect(122.,128.,0)
      call gvect(80.,128.,1)
C
      call gwicol(.5,1)
      call gvect(80.,126.,0)
      call gvect(122.,126.,1)
c
      call gwicol(2.0,4)
      call gvect(20.,120.,0)
      call gvect(20.,120.,1)
      call gvect(200.,120.,1)
      call gvect(200.,115.,0)
      call gvect(20.,115.,1)
C
      call gvect(20.,58.,0)
      call gvect(200.,58.,1)
      call gvect(200.,53.,0)
      call gvect(20.,53.,1)
      call gcharc(15)
C
      call gchar('WATER RESOURCES RESEARCH GROUP$',60.,35.0,3.)
      call gchar('DEPARTMENT OF CIVIL ENGINEERING$',61.,30.,3.)
      call gchar('UNIVERSITY OF SALFORD$',75.,25.,3.)
      call gchar('SALFORD M5 4WT$',80.,20.,3.)
c
      call gcharc(2)
      call gchar('T F F O R$',40.,90.,15.)
c
      call gcharc(15)
            call gchar('A PROGRAM FOR VERIFYING THE FORECASTING
        1 PERFORMANCE
        1  $',25.0,75.0,3.0)
      call gchar('OF A SIMPLE TRANSFER FUNCTION MODEL IN AN
        1 OFF-LINE,
        1  $',25.0,70.0,3.0)
      call gchar('SIMULATED REAL-TIME ENVIRONMENT$',
        1  25.0,65.0,3.0)
c
c     if (idevice.eq.4) then
c        call glj250
c     end if
c
```

```
      call  gchar(' Return to continue$',3.0,0.95*ydim,5.0)
      read(*,*)
      call gclose
      return
      end
```

Appendix 2: Example Rainfall Datafile

```
Willow Brook, FOTHERINGHAY (for verifying cal3 model)
RAIN
9999,99,99,99
1, 448
15
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.020    0.062    0.011    0.003    0.000    0.000    0.000
  0.000    0.026    0.071    0.106    0.000    0.001    0.005    0.028
  0.028    0.026    0.050    0.043    0.056    0.039    0.053    0.051
  0.151    0.371    0.368    0.302    0.186    0.091    0.002    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.002    0.016
  0.015    0.065    0.064    0.125    0.207    0.320    0.273    0.194
  0.651    0.342    0.781    1.639    1.562    0.151    0.024    0.003
  0.000    0.000    0.002    0.020    0.011    0.145    0.158    0.252
  1.000    1.471    0.728    0.359    0.002    0.140    0.090    0.004
  0.000    0.121    0.308    0.055    0.000    0.000    0.001    0.138
  0.157    0.295    0.723    0.018    0.000    0.000    0.011    0.471
  1.207    0.420    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.000    0.000    0.013    0.021    0.024    0.000    0.000    0.000
  0.000    0.000    0.000    0.000    0.000    0.000    0.000    0.000
  0.003    0.028    0.025    0.000    0.000    0.001    0.045    0.212
```

Appendix 3: <u>Example River Datafile</u>

```
Willow Brook, FOTHERINGHAY (for verifying cal3 model)
STAGE
9999,99,99,99
1, 448
15
  0.432    0.431    0.431    0.430    0.429    0.429    0.428    0.427
  0.426    0.425    0.424    0.423    0.422    0.422    0.421    0.420
  0.420    0.419    0.419    0.419    0.418    0.418    0.418    0.418
  0.418    0.418    0.417    0.417    0.417    0.418    0.418    0.418
  0.418    0.419    0.420    0.420    0.420    0.420    0.420    0.420
  0.420    0.420    0.420    0.419    0.420    0.419    0.419    0.420
  0.420    0.420    0.421    0.421    0.421    0.421    0.419    0.420
  0.420    0.420    0.421    0.422    0.424    0.425    0.426    0.426
  0.426    0.427    0.427    0.427    0.427    0.429    0.429    0.430
  0.430    0.433    0.434    0.436    0.438    0.439    0.442    0.444
  0.446    0.447    0.449    0.450    0.453    0.454    0.457    0.459
  0.461    0.462    0.465    0.466    0.468    0.470    0.472    0.475
  0.478    0.480    0.483    0.485    0.486    0.489    0.490    0.492
  0.494    0.497    0.499    0.502    0.505    0.507    0.511    0.515
  0.518    0.521    0.525    0.530    0.536    0.542    0.549    0.557
  0.564    0.571    0.580    0.589    0.599    0.611    0.625    0.638
  0.651    0.663    0.676    0.692    0.709    0.726    0.743    0.760
  0.777    0.796    0.811    0.826    0.842    0.858    0.878    0.897
  0.922    0.949    0.978    1.006    1.032    1.059    1.083    1.112
  1.139    1.166    1.188    1.213    1.236    1.257    1.279    1.301
  1.319    1.340    1.355    1.374    1.390    1.404    1.417    1.434
  1.449    1.462    1.474    1.485    1.494    1.504    1.513    1.519
  1.523    1.533    1.537    1.545    1.550    1.556    1.562    1.571
  1.578    1.586    1.593    1.600    1.605    1.612    1.620    1.629
  1.635    1.642    1.646    1.651    1.654    1.657    1.660    1.664
  1.666    1.667    1.669    1.670    1.671    1.673    1.674    1.675
  1.675    1.675    1.675    1.675    1.675    1.674    1.672    1.670
  1.667    1.664    1.659    1.653    1.645    1.633    1.620    1.601
  1.583    1.568    1.550    1.529    1.510    1.490    1.469    1.449
  1.430    1.412    1.394    1.375    1.357    1.338    1.322    1.306
  1.290    1.275    1.258    1.244    1.230    1.216    1.202    1.189
  1.177    1.166    1.155    1.143    1.133    1.123    1.113    1.103
  1.093    1.082    1.074    1.065    1.056    1.049    1.042    1.034
  1.028    1.021    1.015    1.007    1.001    0.994    0.987    0.981
  0.973    0.968    0.962    0.956    0.951    0.944    0.938    0.932
  0.927    0.918    0.911    0.904    0.897    0.890    0.885    0.879
  0.874    0.868    0.862    0.857    0.851    0.846    0.840    0.834
  0.830    0.825    0.821    0.816    0.811    0.807    0.803    0.799
  0.794    0.790    0.786    0.782    0.778    0.773    0.770    0.766
  0.763    0.759    0.755    0.752    0.748    0.744    0.741    0.737
  0.734    0.730    0.727    0.724    0.720    0.717    0.714    0.710
  0.708    0.705    0.702    0.699    0.697    0.694    0.691    0.689
  0.686    0.683    0.680    0.678    0.674    0.672    0.670    0.667
  0.665    0.662    0.659    0.657    0.654    0.652    0.650    0.648
  0.646    0.644    0.642    0.640    0.638    0.635    0.632    0.630
  0.629    0.627    0.624    0.622    0.619    0.617    0.615    0.614
  0.611    0.610    0.608    0.605    0.603    0.600    0.598    0.596
  0.594    0.591    0.590    0.588    0.586    0.585    0.584    0.583
  0.581    0.579    0.578    0.575    0.574    0.573    0.571    0.569
  0.566    0.566    0.565    0.562    0.561    0.558    0.557    0.556
  0.554    0.553    0.551    0.550    0.549    0.548    0.547    0.546
  0.545    0.544    0.543    0.542    0.542    0.540    0.540    0.538
  0.538    0.537    0.536    0.535    0.535    0.534    0.533    0.533
  0.532    0.531    0.530    0.529    0.529    0.528    0.527    0.526
  0.526    0.525    0.524    0.524    0.524    0.523    0.522    0.522
  0.522    0.521    0.521    0.520    0.520    0.519    0.519    0.518
```

Appendix 4: Example Model/Catchment Datafile

```
Fotheringhay (Willow Brook) Model Data File (use cal_test3)
3,5,0
1.9566 -1.2581  0.2884  0.0113 -0.0067  0.0108  0.0074  0.0211
60
85.25
1
0
4
0.1370,  3.2142,  0.0012,  1.5413
0.3770,  2.9942,  0.0003,  1.5002
1.2940,  3.0029, -0.0009,  1.4984
5.0000,  2.9858,  0.0031,  1.5030
```

Appendix 5: Example Output File

Model Forecast Verification Results

Forecasting with Model Order 3, 5, 0

Model Parameters

a 1)	1.9566
a 2)	-1.2581
a 3)	0.2884
b 1)	0.0113
b 2)	-0.0067
b 3)	0.0108
b 4)	0.0074
b 5)	0.0211

Actual Subcatchment Averaged Rainfall (mm)

0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.093	0.003	0.203	0.034	0.147	0.199
1.192	0.279	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.018	0.269	0.994
3.413	1.740	0.022	0.566	3.558	0.236	0.484	0.139
1.193	0.482	1.627	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.034	0.024	0.000	0.000	0.056	0.258

River Discharge (cumecs)

0.849	0.840	0.828	0.820	0.817	0.814	0.811	0.814
0.820	0.820	0.817	0.820	0.823	0.820	0.825	0.837
0.840	0.849	0.867	0.890	0.908	0.936	0.957	0.985
1.016	1.038	1.070	1.111	1.160	1.249	1.358	1.531
1.728	1.989	2.253	2.549	3.027	3.517	4.007	4.450
4.830	5.149	5.427	5.614	5.759	5.905	6.069	6.235
6.361	6.437	6.472	6.501	6.501	6.472	6.373	6.075
5.670	5.230	4.835	4.476	4.161	3.888	3.665	3.474
3.297	3.154	3.031	2.915	2.804	2.699	2.579	2.473
2.381	2.286	2.212	2.143	2.076	2.012	1.957	1.899
1.849	1.796	1.755	1.717	1.676	1.636	1.599	1.567
1.538	1.502	1.474	1.445	1.414	1.382	1.355	1.338
1.310	1.290	1.266	1.246	1.226	1.213	1.199	1.186
1.176	1.170	1.157	1.147	1.140	1.134	1.127	1.121

Total depth of rain	:	17.26
Mean hourly rainfall	:	0.15
Maximum hourly rainfall	:	3.56
Maximum river discharge	:	6.50
River baseflow	:	0.81
Percentage runoff	:	43.61

Forecasts: Perfect Foresight

Time	Delta	Flow	1	2	4	6	8
1	2.00	0.8	0.9	0.9	0.9	0.9	0.9
2	2.00	0.8	0.8	0.8	0.8	0.8	0.8
3	2.00	0.8	0.8	0.8	0.8	0.8	0.8
4	2.00	0.8	0.8	0.8	0.8	0.8	0.8
5	2.00	0.8	0.8	0.8	0.8	0.8	0.8
6	2.00	0.8	0.8	0.8	0.8	0.8	0.8

7	2.00	0.8	0.8	0.8	0.8	0.8	0.8
8	2.00	0.8	0.8	0.8	0.8	0.8	0.9
9	2.00	0.8	0.8	0.8	0.8	0.8	0.9
10	2.00	0.8	0.8	0.8	0.8	0.8	0.9
11	2.00	0.8	0.8	0.8	0.8	0.9	0.9
12	2.00	0.8	0.8	0.8	0.9	0.9	1.0
13	2.00	0.8	0.8	0.8	0.9	1.0	1.1
14	2.00	0.8	0.8	0.8	0.9	1.0	1.2
15	2.00	0.8	0.8	0.9	1.0	1.1	1.3
16	2.00	0.8	0.9	0.9	1.0	1.2	1.3
17	2.00	0.8	0.9	0.9	1.0	1.2	1.3
18	2.00	0.8	0.9	0.9	1.1	1.2	1.2
19	2.00	0.9	0.9	1.0	1.2	1.3	1.2
20	2.00	0.9	0.9	1.0	1.2	1.2	1.2
21	2.00	0.9	1.0	1.0	1.1	1.1	1.1
22	2.00	0.9	1.0	1.0	1.0	1.0	1.0
23	2.00	1.0	1.0	1.0	1.0	0.9	0.9
24	2.00	1.0	1.0	1.0	1.0	1.0	1.0
25	2.00	1.0	1.0	1.0	1.0	1.0	1.0
26	2.00	1.0	1.0	1.0	1.0	1.0	1.1
27	2.00	1.1	1.1	1.1	1.1	1.1	1.3
28	2.00	1.1	1.1	1.1	1.1	1.2	1.5
29	2.00	1.2	1.2	1.2	1.2	1.4	1.7
30	2.00	1.2	1.3	1.4	1.5	1.7	2.3
31	2.00	1.4	1.4	1.5	1.7	2.0	2.8
32	2.00	1.5	1.7	1.8	2.1	2.7	3.3
33	3.00	1.7	2.0	2.1	2.7	3.8	4.6
34	3.36	2.0	2.2	2.4	3.5	4.6	5.5
35	5.04	2.3	2.6	3.2	5.2	6.6	7.9
36	4.26	2.5	3.0	3.9	5.3	6.5	7.2
37	4.16	3.0	3.9	4.6	5.7	6.8	7.2
38	2.78	3.5	3.9	4.2	4.8	5.3	5.4
39	4.16	4.0	4.5	4.9	6.0	6.5	6.9
40	3.70	4.5	4.8	5.4	6.1	6.3	6.6
41	3.62	4.8	5.4	5.7	6.1	6.4	6.5
42	2.51	5.1	5.3	5.4	5.5	5.5	5.4
43	3.76	5.4	5.7	5.8	6.1	6.3	6.0
44	3.15	5.6	5.7	5.8	5.9	5.8	5.5
45	4.73	5.8	6.0	6.2	6.6	6.4	5.9
46	3.71	5.9	6.0	6.1	6.1	5.7	5.3
47	4.81	6.1	6.3	6.4	6.2	5.8	5.3
48	4.11	6.2	6.3	6.2	5.8	5.4	5.0
49	4.11	6.4	6.4	6.3	5.9	5.4	5.0
50	4.11	6.4	6.4	6.3	5.9	5.4	5.0
51	4.11	6.5	6.4	6.3	5.9	5.4	5.0
52	4.11	6.5	6.4	6.3	5.9	5.5	5.1
53	4.11	6.5	6.4	6.3	5.9	5.4	5.0
54	4.11	6.5	6.4	6.2	5.8	5.4	5.0
55	4.11	6.4	6.2	6.0	5.6	5.2	4.8
56	4.11	6.1	5.7	5.4	5.0	4.6	4.3
57	4.11	5.7	5.3	5.0	4.6	4.2	3.9
58	4.11	5.2	4.9	4.6	4.2	3.9	3.6
59	4.11	4.8	4.5	4.3	4.0	3.7	3.4
60	4.11	4.5	4.2	4.0	3.7	3.4	3.2
61	4.11	4.2	3.9	3.7	3.5	3.2	3.0
62	4.11	3.9	3.7	3.5	3.3	3.1	2.9
63	4.11	3.7	3.5	3.3	3.1	2.9	2.8
64	4.11	3.5	3.3	3.2	3.0	2.8	2.6
65	4.11	3.3	3.1	3.0	2.8	2.6	2.5
66	4.11	3.2	3.0	2.9	2.8	2.6	2.4
67	4.11	3.0	2.9	2.8	2.7	2.5	2.4
68	4.11	2.9	2.8	2.7	2.5	2.4	2.3
69	4.11	2.8	2.7	2.6	2.5	2.3	2.2
70	4.11	2.7	2.6	2.5	2.4	2.2	2.1
71	4.11	2.6	2.5	2.4	2.2	2.1	2.0
72	4.11	2.5	2.4	2.3	2.2	2.1	2.0
73	4.11	2.4	2.3	2.2	2.1	2.0	1.9
74	4.11	2.3	2.2	2.1	2.0	1.9	1.8
75	4.11	2.2	2.1	2.1	2.0	1.9	1.8
76	4.11	2.1	2.1	2.0	1.9	1.8	1.7
77	4.11	2.1	2.0	2.0	1.9	1.8	1.7
78	4.11	2.0	2.0	1.9	1.8	1.7	1.6

79	4.11	2.0	1.9	1.9	1.8	1.7	1.6
80	4.11	1.9	1.8	1.8	1.7	1.6	1.6
81	4.11	1.8	1.8	1.8	1.7	1.6	1.5
82	4.11	1.8	1.7	1.7	1.6	1.6	1.5
83	4.11	1.8	1.7	1.7	1.6	1.5	1.5
84	4.11	1.7	1.7	1.6	1.6	1.5	1.5
85	4.11	1.7	1.6	1.6	1.5	1.5	1.4
86	4.11	1.6	1.6	1.6	1.5	1.4	1.4
87	4.11	1.6	1.6	1.5	1.5	1.4	1.4
88	4.11	1.6	1.5	1.5	1.4	1.4	1.3
89	4.11	1.5	1.5	1.5	1.4	1.4	1.3
90	4.11	1.5	1.5	1.4	1.4	1.3	1.3
91	4.11	1.5	1.4	1.4	1.4	1.3	1.3
92	4.11	1.4	1.4	1.4	1.3	1.3	1.3
93	4.11	1.4	1.4	1.4	1.3	1.3	1.2
94	4.11	1.4	1.4	1.3	1.3	1.2	1.2
95	4.11	1.4	1.3	1.3	1.3	1.2	1.2
96	4.11	1.3	1.3	1.3	1.3	1.2	1.2
97	4.11	1.3	1.3	1.3	1.2	1.2	1.1
98	4.11	1.3	1.3	1.3	1.2	1.2	1.2
99	4.11	1.3	1.2	1.2	1.2	1.2	1.1
100	4.11	1.2	1.2	1.2	1.2	1.1	1.1
101	4.11	1.2	1.2	1.2	1.2	1.1	1.1
102	4.11	1.2	1.2	1.2	1.2	1.1	1.1
103	4.11	1.2	1.2	1.2	1.1	1.1	1.1
104	4.11	1.2	1.2	1.2	1.1	1.1	1.1
105	4.11	1.2	1.2	1.2	1.1	1.1	1.1
106	4.11	1.2	1.2	1.2	1.1	1.1	1.1
107	4.11	1.2	1.1	1.1	1.1	1.1	1.1
108	4.11	1.1	1.1	1.1	1.1	1.1	1.1
109	4.11	1.1	1.1	1.1	1.1	1.1	1.2
110	4.11	1.1	1.1	1.1	1.1	1.2	1.2
111	4.11	1.1	1.1	1.1	1.1	1.2	1.2
112	4.11	1.1	1.1	1.1	1.1	1.2	1.2

Forecast Root Mean Square Errors (RMSE)

1 Step-Ahead	0.055
2 Step-Ahead	0.121
3 Step-Ahead	0.210
4 Step-Ahead	0.290
5 Step-Ahead	0.370
6 Step-Ahead	0.442
7 Step-Ahead	0.515
8 Step-Ahead	0.572

Forecast Errors of 2,4,6 and 8 Step-Ahead Forecasts (cumecs)

Time	2	4	6	8
1	-0.08	-0.11	-0.10	-0.09
2	0.01	0.00	0.00	0.00
3	0.00	0.00	0.01	0.00
4	0.00	0.00	0.01	0.01
5	-0.01	0.00	0.00	0.00
6	0.00	0.01	0.01	0.00
7	0.01	0.00	0.01	0.00
8	0.00	0.00	-0.01	-0.01
9	-0.01	-0.01	-0.02	-0.03
10	0.00	0.00	-0.01	-0.06
11	0.01	0.00	-0.02	-0.08
12	-0.01	-0.01	-0.07	-0.12
13	-0.01	-0.03	-0.09	-0.17
14	0.01	-0.04	-0.09	-0.22
15	-0.02	-0.09	-0.17	-0.31
16	-0.05	-0.10	-0.23	-0.32
17	-0.04	-0.12	-0.26	-0.27
18	0.00	-0.12	-0.21	-0.18
19	-0.07	-0.22	-0.24	-0.17
20	-0.10	-0.19	-0.15	-0.05
21	-0.08	-0.08	-0.03	0.09

22	0.00	0.04	0.13	0.28
23	0.05	0.11	0.21	0.42
24	0.03	0.11	0.26	0.55
25	0.03	0.13	0.34	0.70
26	0.06	0.22	0.51	0.87
27	0.06	0.27	0.63	0.99
28	0.10	0.39	0.76	1.10
29	0.16	0.51	0.87	1.28
30	0.18	0.52	0.88	1.21
31	0.24	0.58	1.02	1.23
32	0.16	0.49	0.84	1.15
33	0.10	0.32	0.16	0.21
34	0.10	-0.01	-0.17	-0.35
35	-0.19	-1.20	-1.75	-2.45
36	-0.40	-0.87	-1.32	-1.62
37	-0.62	-0.90	-1.36	-1.47
38	0.27	0.35	0.36	0.49
39	-0.11	-0.60	-0.78	-0.78
40	-0.23	-0.45	-0.43	-0.33
41	-0.30	-0.38	-0.31	-0.14
42	0.20	0.44	0.71	1.04
43	-0.07	-0.05	0.07	0.42
44	0.15	0.32	0.62	1.03
45	-0.16	-0.21	0.11	0.57
46	0.09	0.35	0.77	1.16
47	-0.01	0.30	0.74	1.04
48	0.24	0.66	1.06	1.06
49	0.21	0.63	0.93	0.63
50	0.23	0.60	0.64	0.20
51	0.23	0.51	0.24	-0.19
52	0.16	0.17	-0.24	-0.59
53	0.11	-0.18	-0.59	-0.86
54	-0.13	-0.56	-0.89	-1.09
55	-0.35	-0.75	-1.01	-1.14
56	-0.22	-0.50	-0.72	-0.81
57	-0.16	-0.40	-0.57	-0.65
58	-0.11	-0.30	-0.42	-0.49
59	-0.13	-0.29	-0.39	-0.42
60	-0.09	-0.20	-0.27	-0.30
61	-0.06	-0.15	-0.20	-0.22
62	-0.03	-0.10	-0.14	-0.16
63	-0.05	-0.09	-0.13	-0.17
64	-0.04	-0.06	-0.10	-0.16
65	0.01	-0.02	-0.07	-0.11
66	-0.02	-0.05	-0.12	-0.15
67	-0.02	-0.08	-0.12	-0.15
68	-0.02	-0.08	-0.12	-0.12
69	-0.04	-0.08	-0.10	-0.11
70	-0.05	-0.08	-0.09	-0.10
71	0.01	-0.01	-0.03	-0.04
72	-0.02	-0.03	-0.05	-0.05
73	-0.02	-0.03	-0.04	-0.05
74	0.02	0.01	-0.01	-0.02
75	-0.02	-0.03	-0.04	-0.04
76	-0.01	-0.02	-0.03	-0.02
77	0.00	-0.01	-0.01	-0.01
78	0.00	-0.01	-0.01	-0.01
79	-0.01	-0.02	-0.01	-0.02
80	0.00	0.01	0.01	0.01
81	-0.01	-0.01	-0.01	0.00
82	0.02	0.01	0.01	0.01
83	-0.01	-0.01	-0.01	-0.01
84	-0.01	-0.01	-0.01	-0.01
85	0.00	0.01	0.01	0.00
86	0.00	0.01	0.01	0.00
87	0.00	0.00	0.00	-0.01
88	0.00	0.00	-0.01	-0.01
89	-0.01	-0.01	-0.02	-0.02
90	0.01	0.01	0.01	0.01
91	-0.01	-0.02	-0.02	-0.02
92	-0.01	0.00	-0.01	-0.01
93	0.00	0.00	0.00	0.00

94	0.01	0.01	0.00	0.01
95	0.00	0.00	0.00	0.01
96	-0.02	-0.02	-0.02	-0.01
97	0.01	0.01	0.02	0.03
98	-0.01	-0.01	0.00	0.02
99	0.00	0.01	0.02	0.03
100	0.00	0.01	0.02	0.03
101	0.01	0.02	0.03	0.04
102	0.00	0.01	0.02	0.03
103	0.01	0.02	0.02	0.03
104	0.01	0.02	0.03	0.03
105	0.00	0.01	0.02	-1.11
106	-0.01	0.00	0.00	-1.12
107	0.01	0.02	-1.11	-1.12
108	0.01	0.01	-1.12	-1.14
109	0.00	-1.13	-1.14	-1.18
110	0.00	-1.13	-1.15	-1.20
111	-1.13	-1.14	-1.18	-1.21
112	-1.12	-1.14	-1.19	-1.19

Appendix 6: Runtime Listing of TFFOR

The following is a runtime listing of TFFOR as described in section 5.3 of the main report.

```
--------------------------------------------------------------------
 The UNIRAS graphics routines in this program are device independent.
--------------------------------------------------------------------
 Please type in the integer corresponding to the device you wish
 graphics to be directed to;

    (1) VAXstation
    (2) VT Emulator (eg VT340)
    (3) Pen Plotter
    (4) Ink Jet Printer
    (5) IBM PS2
 Please type in choice (1,2,3,4, or 5)
2

 Type in name of model datafile
fot3.mod

 Type in name of rain datafile
cs3ver.rai

 Type in name of river datafile
cs3ver.sta
 Rainfall data interval  15 minutes
 River data interval     15 minutes
 Model interval          60 minutes

 Select a forecast interval:
    15 minutes
    30 minutes
    45 minutes
    Other [enter integer, multiple of  60]
60

 Is an output file required? [Y or N]
Y
 Type in name of output file
test.out



 Program Menu
 ------------


 (1) Event Simulation
 (2) Forecasting with specified lead-times
 (3) Error/Delta Plots
```

```
 (0) STOP Program

 Enter integer corresponding to choice [1,2,3 or 0]

2

 Please select future rainfall scenario:
   (1) No more rain
   (2) Past rain averaged
   (3) FRONTIERS rainfall forecast
   (4) Perfect foresight of rainfall
   (5) Other (entered by user)
 Enter integer corresponding to choice [1,2, 3, 4, or 5]
4

 Enter an initial value of delta [real] between    0.28 and 5.64
2.0
 Is delta to be kept constant? [Y or N]
n

 Enter required forecast lead-time [in hours]
10
```