A Linear Catchment Model for Real Time Flood Forecasting

D. S. SINCLAIR

Submitted in partial fulfillment of the requirements for the degree of
Master of Science in Engineering
In the
Civil Engineering Programme
University of Natal

May 2001

ABSTRACT

A linear reservoir cell model is presented which is proposed as a good candidate for real time flood forecasting applications. The model is designed to be computationally efficient since it should be able to run on a P.C and must operate online in real time. The model parameters and forecasts can be easily updated in order to allow for a more accurate forecast based on real time observations of streamflow and rainfall.

The final model, once calibrated, should be able to operate effectively without requiring highly skilled and knowledgeable operators. Thus it is hoped to provide a tool which can be incorporated into an early warning system for mitigation of flood damage, giving water resources managers the extra lead-time to implement any contingency plans which may be necessary to ensure the safety of people and prevent damage to property.

The use of linear models for describing hydrological systems is not new, however the model presented in this thesis departs from previous implementations. A particular departure is the novel method used in the conversion of observed to effective rainfall. The physical processes that result in the rainfall to runoff conversion are non-linear in nature. Most of the significant non-linearity results from rainfall losses, which occur largely due to evaporation and human extraction. The remaining rainfall is converted to runoff. These losses are particularly significant in the South African climate and in some regions may be as much as 70-90 % of the total observed rainfall. Loss parameters are an integral part of the model formulation and allow for losses to be dealt with directly. Thus, input to the model is observed rainfall and not the "effective" rainfall normally associated with conceptual catchment models.

The model is formulated in Finite Difference form similar to an Auto Regressive Moving Average (ARMA) model; it is this formulation which provides the required computational efficiency. The ARMA equation is a discretely coincident form of the State-Space equations that govern the response of an arrangement of linear reservoirs. This results in a functional relationship between the reservoir response constants and the ARMA coefficients, which guarantees stationarity of the ARMA model.

PREFACE

I HEREBY DECLARE THAT THE CONTENTS OF THIS DISSERTATION ARE ENTIRELY MY OWN WORK, EXCEPT WHERE SPECIFIED OTHERWISE. THIS WORK HAS NOT BEEN PREVIOUSLY SUBMITTED IN WHOLE OR IN PART TO THIS OR ANY OTHER UNIVERSITY.

Signed: ..... Date: 05 April 2002

AS THE CANDIDATE'S SUPERVISOR I HAVE/HAVE NOT APPROVED THIS DISSERTATION FOR SUBMISSION

Signed : ..... Name : ..... Date : .....

ACKNOWLEDGEMENTS

Thanks must go to a great many people who I have come into contact with during the last two years. In particular various members of staff from the Civil Engineering Programme. Some specific acknowledgements must also be made.

Professor G.G.S. Pegram:

My supervisor whose great knowledge and enthusiasm for Hydrology has kept me on track during a difficult learning curve.

My parents:

For their support and interest and mainly for doing what parents do.

The Water Research Commission:

For providing the funding for this study. Thanks must go to the chairman and members of the steering committee for valuable advice and insight.

The South African Weather Bureau METSYS:

For keeping the weather radar in working order and supplying me with precipitation data.

Umgeni Water:

For postponing my contractual obligations and allowing me two years to do full time research.

The Department of Water Affairs and Forestry:

For the provision of copious streamflow data in a friendly and efficient manner.

Dr Derek Stretch and Tony Clothier:

For interesting lunch time discussions and a limitless interest in how the world works.

The University of Natal:

For providing me with an office from which to work and the use of their computing facilities.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF FIGURES

TABLE OF FIGURES

TABLE OF FIGURES

CHAPTER 1 INTRODUCTION

In this chapter the physical processes involved in the modelling of a catchment are introduced. The aim is to give an overview of the various modelling approaches available, and to explain the selection of the particular approach adopted in this study. A point to be highlighted is the major non-linearity in the loss processes.

1.1 BACKGROUND

The research reported here has been conducted with funding from the Water Research Commission (WRC). The study came about as a result of earlier WRC projects providing a facility for recording rainfall rates at high temporal (five minute intervals) and spatial (1 km^2) resolution, through the use of weather radar (e.g Mather et al., 1997). With the development of the "String of Beads" Rainfall Model (Pegram & Clothier, 1999); the extension of this technology to a real time flood forecasting application is an obvious way to gain new practical benefits from the products arising out of previous projects. A linear reservoir cell model is presented which is proposed as a good candidate for real time flood forecasting applications. The model is designed to be computationally efficient since it should be able to run on a P.C and must operate online in real time. The model parameters and forecasts can be easily updated in order to allow for a more accurate forecast based on real time observations of streamflow and rainfall.

The final model, once calibrated, should be able to operate effectively without requiring highly skilled and knowledgeable operators. Thus it is hoped, by including short term (a few hours ahead) rainfall forecasts, to provide a tool which can be incorporated into an early warning system for mitigation of flood damage; giving water resources managers the extra lead-time to implement any contingency plans which may be necessary to ensure the safety of people and prevent damage to property. This can provide a particular benefit in the South African context due to the significant numbers of people who live in informal settlements often situated in flood prone areas.

Fully distributed physically based and complex conceptual catchment models, using physical analogies, are computationally expensive (for flood forecasting) and have significant data requirements. The approach followed here has been to use a simple linear conceptual framework, to reduce the data requirements and maintain parsimony. The use of linear models for describing hydrological systems is not new, however the model presented

in this thesis departs from previous implementations. A particular departure is the method used in the conversion of observed to effective rainfall. The physical processes that result in the rainfall to runoff conversion are non-linear in nature. Most of the significant non-linearity results from rainfall losses, which occur largely due to evaporation and human extraction. The remaining rainfall is converted to runoff. These losses are particularly significant in the South African climate and in some regions may be as much as 70-90 % of the total observed rainfall. A loss parameter is an integral part of the model formulation and allows for losses to be dealt with directly. Thus, input to the model is observed rainfall and not the "effective" rainfall normally associated with conceptual catchment models.

The model is formulated in Finite Difference form similar to an Auto Regressive Moving Average (ARMA) model; it is this formulation that provides the required computational efficiency. The ARMA equation is a discretely coincident form of the State-Space equations that govern the response of a conceptual arrangement of linear reservoirs. This results in a functional relationship between the reservoir response constants and the ARMA coefficients, which guarantees stationarity of the ARMA model.

The three reservoir feed forward model is applied to the Liebenbergsvlei study catchment as a semi-distributed multi-cell model. The responses from each of the cells are linearly summed at the catchment outlet to produce the total catchment response. Good model fits to selected calibration events are achieved using Monte Carlo and Metropolis based automatic optimization routines.

1.2 Rainfall to Runoff Conversion Processes

A successful catchment model must be able to reasonably duplicate two major hydrological processes occurring on the catchment. The first is the rainfall to runoff conversion process, which defines the proportion of rainfall to reach the channel. Some of the rainfall will be lost due to the effects of evaporation or other withdrawals, the remaining portion will reach the river via overland or subsurface flow, each of which has a different travel time to reach the channel. The second process, that of channel routing, takes place with the runoff being routed down the river, once it has entered the channel. The rainfall to runoff conversion process is highly non-linear in nature and it is this that proves to be the main difficulty in

streamflow modelling. Channel routing can be reasonably approximated as a linear process (Todini & Bossi, 1986), provided backwater effects are negligible.

Several modelling approaches can be followed. The most natural would be to attempt to model the physical processes that are actually occurring on the catchment. To do so requires an understanding of a very complex system in great detail. Since this is an exceptionally difficult task a number of approximations are usually required in order to decrease the complexity of the problem. At the other end of the spectrum of models is a purely mathematical model. Transfer functions like the Instantaneous Unit Hydrograph perform a linear conversion of effective rainfall input to runoff, implicitly assuming that the conversion processes are time invariant for a particular catchment. Because a simple analysis of the physical processes will confirm that this is not actually the case, it is suggested that a compromise between physical and mathematically based modelling approaches could yield useful results while avoiding some complexity. This conceptual type of model attempts to linearize parts of the conversion and routing tasks while using some technique of introducing the required non-linearity into the system, where it is most needed.

1.2.1 Loss Processes

The volume of rainfall which reaches the channel, and when it arrives, depends on a number of things. When the rainfall impacts the surface of the catchment a proportion will infiltrate into the soil while the rest will become direct runoff flowing over the surface to reach the channel. The rate at which infiltration to the soil store can occur is dependent on the hydraulic conductivity of the soil as well as the degree of saturation. The hydraulic conductivity is a function of the soil type and does not vary with time but there is a large spatial (and some temporal) variability due to ground cover providing extra flow paths, through cracks and macro pores, other than those inherent in the soil structure. The catchment moisture condition varies considerably with time depending on the recent history of rainfall and other climatic conditions on the catchment. Therefore to determine the rate of infiltration as input to a physically based module requires a detailed knowledge of the soil types and corresponding moisture states on the catchment at any given time.

Once infiltration into the soil has occurred the water may travel to the channel as subsurface flow or more slowly as groundwater seepage. Water may either be taken up by the roots of plants and lost due to evapotranspiration or be lost due to direct evaporation from the upper layers of the soil and free water surfaces.

When attempting to model a catchment using a physically based modelling approach, each of the above processes must be successfully modelled in order to determine what ends up in the channel. Infiltration models rely on determining the flow rate of water through a soil, this process can be described by Darcy's equation (Darcy, 1856)

$$q = -K\frac{dH}{dz}$$

where q is the flow rate per unit area, K is the hydraulic conductivity and $\frac{dH}{dz}$ represents the difference in head along the direction of flow. This linear equation only applies to a saturated soil that is homogeneous and isotropic.

Richard (1931) presented a general equation for unsteady unsaturated flow in three dimensions. Richard's equation for one dimension is given as

$$\frac{\partial\theta}{\partial t} = \frac{\partial}{\partial z}(D\frac{\partial\theta}{\partial z} + K)$$

where θ is the soil moisture content, t represents time and D is the soil water diffusivity. Two well-known infiltration models presented by Horton (1940) and Philip (1957) are approximate solutions to the one-dimensional form of Richards's equation. The Green-Ampt method (Green & Ampt, 1911) uses an approximation of the physical theory for which a direct analytical solution can be found.

Taking these in turn, Horton's equation (Horton, 1940) is:

$$f(t) = f_c + (f_0 - f_c)e^{-kt}$$

where $f(t)$ is the potential rate of infiltration, f_0 is the initial potential rate, f_c is the constant rate reached at equilibrium conditions, t is time and k is a decay constant. The potential infiltration rate can only be realized if there is sufficient water to pond at the soil surface. If the rate of precipitation is less than the infiltration potential, all rainfall will infiltrate until the potential infiltration rate has sufficiently reduced. When the rate of precipitation is

channel from the soil store (Figure 1.1) has a much slower response time and may often be ignored for the purposes of event based modelling. However, it should be taken into account in a continuous streamflow model.

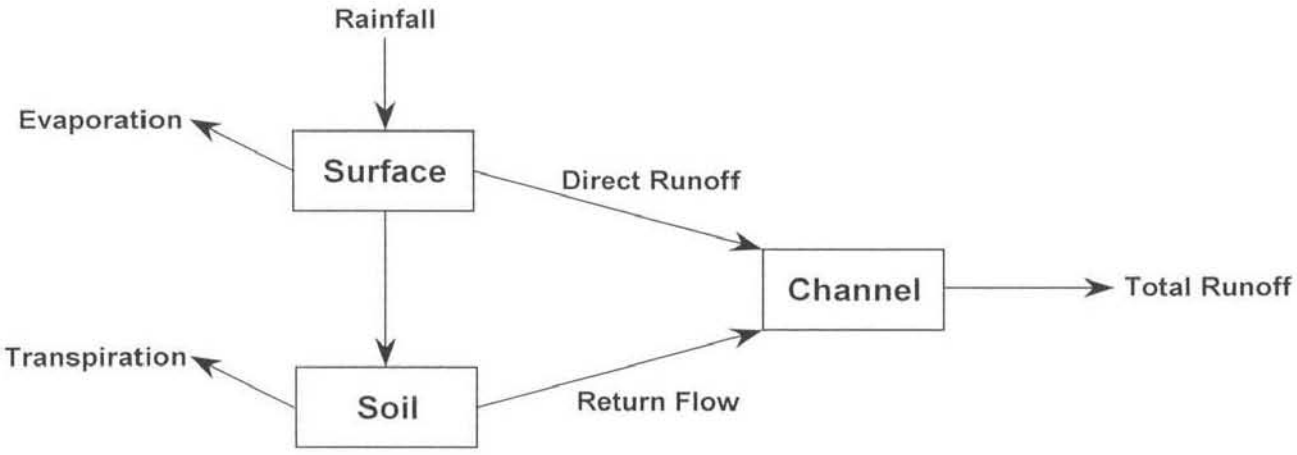


Figure 1.1: How observed rainfall is converted to total runoff

The loss module of a typical catchment model is used to determine what volume of the observed rainfall is converted to effective rainfall; which must then be routed overland and along the channel. The remaining rainfall may be considered as a loss. The manner in which this loss component is determined and then removed from the observed rainfall sequence will have a significant effect on the arrival times and magnitudes of the (discrete) inputs to the routing module. The choice of loss model will therefore have a significant effect on the eventual streamflow outputs.

Various types of loss models have been proposed historically, all of these are approximate, as the physical processes resulting in the losses are not clearly determinable. Following is a discussion of some of the more commonly used methods to determine the effective rainfall.

The losses can be removed in several ways. Assuming that the bulk of the losses occurring are due to infiltration losses which decrease exponentially with time an initial loss can be removed. The total volume of loss can be established by integrating the loss rate over the period of the rainfall event under consideration. In this case, the entire loss volume is removed from the beginning of the storm and the remaining rainfall is then the effective rain. Alternatively, a continuing loss approach can be used where losses are removed

throughout the event. The removal may be made at a constant rate or as a proportion of the recorded rainfall depth during each period. Combination *initial-continuous* or *initial-proportional* loss models are sometimes used (Hill et al., 1998). These methods of loss removal are known as index types.

Bras (1990) has suggested the following model

$$f(t) = f_c + (f_0 - f_c)e^{-B(A_{30}+P)}$$

which is a modified Horton equation with A_{30} a 30 day antecedent precipitation index, f_o and f_c are as for Horton's equation, P is the accumulated rainfall during the event under consideration and B is a parameter to be estimated for the catchment. The exponential decay term ensures that the bulk of the losses occur near the beginning of the storm, this corresponds with our understanding of the physical mechanisms producing the losses.

The SCS method of loss determination may also be used but the parameters would need adjustment for local conditions as it is a semi-empirical method based on observations of catchments in the United States. The method is based on the assumption that the following equation holds

$$\frac{F_a}{S} = \frac{P_e}{P - I_a}$$

where F_a is the depth of precipitation which infiltrates after the initial abstraction I_a has been removed, S is a hypothetical "maximum retention" for the catchment, P is the total precipitation during the storm and P_e is the excess precipitation. Using continuity and the empirical relation $I_a = mS$ it follows that

$$P_e = \frac{(P - mS)^2}{P + (1 - m)S} \qquad (1.1)$$

where the coefficient m is a factor to be decided. The United States department of Agriculture suggests a value of 0.2 (NEH-4, 1972). Schmidt and Schulze (1987) suggest that this value is too high (for South Africa) and recommend the use of 0.1 as a reasonable value.

A dimensionless curve number (for S in inches) is defined as

$$CN = \frac{1000}{10 + S}$$

and tabulated values of these curve numbers for a variety of different soil types and antecedent moisture conditions are available. Once a curve number has been chosen from such a table, the value of S can be calculated. Equation 1.1 is then used to find the effective precipitation from a known value of P. The temporal distribution of the SCS abstractions could be determined using a combination *initial-continuing* loss method, computed at convenient discrete timesteps (Chow et al., 1988).

1.2.2 Overland Flow and Channel Routing

Overland flow refers to that portion of surface runoff which flows down flat sloping surfaces such as hillslopes, it can be considered distinct from flow along the small channels and rills which eventually join the main channel. The water is thought of as flowing across the surface in a shallow sheet, with the flow processes often dominated by viscous effects.

General flow routing may be characterized as being either lumped or distributed. Lumped routing calculates flows at a particular position with respect to time while distributed routing calculates flows at a number of different positions along the flow path with respect to time.

The attractiveness of lumped routing methods lies mainly in their simplicity when compared with distributed methods. Their basis is the equation of the conservation of mass and as such they are governed by the continuity equation

$$I(t) - Q(t) = \frac{dS}{dt}$$

which, in the context of streamflow, states that the rate of change of storage in a river reach is equal to the difference between inflow and outflow from the reach. Three basic types of lumped routing methods are commonly used

- Level pool routing assumes that the water surface is level throughout the reach and is generally used for routing flows through large reservoirs.

- Routing methods applicable to channels where a sloped surface profile is accounted for in the storage flow relationship. For example the Muskingum method (McCarthy, 1938).
- Linear systems described by convolution integrals such as the linear reservoir first introduced by Zoch (1934).

In order to determine accurately the properties of a flood wave at positions of interest along the river reach, distributed routing methods can be employed, but are computationally expensive. The *Saint-Venant* equations for one-dimensional unsteady flow provide the full solution. The conservation of mass equation in differential form is

$$\frac{\partial(AV)}{\partial x}+\frac{\partial A}{\partial t}-q=0$$

with A representing the cross-sectional area, V the velocity, q is the lateral inflow, t time and x the distance along the channel centre-line. The equation of momentum conservation is

$$\frac{\partial V}{\partial t}+V\frac{\partial V}{\partial x}+g\left(\frac{\partial y}{\partial x}-S_0+S_f\right)=0$$

with g the gravitational acceleration, y the depth of the water in the channel, S_0 the bed slope and S_f the friction slope for the channel. No general analytical solutions to these equations have been found but they can be solved numerically.

Approximate methods for routing flow down the channel have been developed, particularly before the advent and common use of computers for numerical computations. Most of these methods ignore or linearize some of the terms in the full equation in order to make it possible to find an analytical solution.

The *Kinematic wave* equation ignores the local acceleration$(\partial V/\partial t)$, convective acceleration$(V\,\partial V/\partial x)$ and pressure$(g\,\partial y/\partial x)$ terms in the momentum equation reducing it to

$$S_0=S_f$$

The Kinematic form therefore makes the assumption that the gravity and friction forces balance each other and continuity is the only condition to be satisfied. This form only accounts for the effect of translation on the wave front.

The *Diffusion wave* model includes the pressure term in the momentum equation accounting for backwater effects as well as translation. The momentum equation for this type of model is given as

$$\frac{\partial y}{\partial x} - S_0 + S_f = 0$$

1.3 PHYSICALLY BASED MODELS

Physically based models attempt to reproduce the physical processes on the catchment, which produce the flows occurring in the channel. They are fully distributed models based on the partial differential equations describing catchment flow processes. Binley et al. (1989) list:

> "Models such as the Systeme Hydrologique Europèen (SHE) ... and the Institute of Hydrology Distributed Model (IHDM) ..."

Such complex models require very detailed information on the current state and characteristics of the catchment. Data such as terrain information, vegetation type, soil type and properties as well as climatic variables are often required. The complexity of this type of model requires skilled operators and, often, significant computing resources to run properly. The advantage of complex physical models is that they result in an improved understanding of the processes involved in the production of streamflow.

These catchment models are effectively equivalent to semi-distributed conceptual models in the sense that they are lumped at the grid scale and as such not spatially continuous. Large increases in the number of grid points will increase the complexity and data requirements of the model but will not necessarily provide significant improvements in the accuracy of the output.

The spatial resolution of the model will also have an effect on model parameterization. Bevan (1989) showed that the expected variance in parameter estimates actually decreases with increasing grid size. This suggests that increasing the spatial resolution could introduce difficulties in calibration of the model, and provides an argument supporting spatially lumped modelling techniques. Also noted by Bevan is the fact that the governing equations for the physical processes being modelled are based on homogeneous systems at

very small scales. It is not clearly known whether these equations are still relevant at grid resolutions with spatially averaged parameters.

1.4 MATHEMATICAL MODELS

Mathematically based streamflow models may be either deterministic, attempting to reproduce the flows actually observed, or Stochastic, reproducing a set of flows having the same statistical properties as historically observed flow records. Each of these may be useful for flood prediction. A stochastically based model can predict the most likely future flows conditioned on the current flows. This simply takes into account what has happened in the past and hence infers what may happen in the future; it does not attempt to reproduce that which actually will happen. A deterministic model uses some kind of functional relationship between rainfall inputs and streamflow outputs to try to predict the future flows which will actually occur. An example of a stochastic model would be a time series model where the model parameters to be estimated are auto-regressive and/or moving average components.

1.5 CONCEPTUAL MODELS

Conceptual catchment models use the concepts of physical models and apply mathematical techniques to reduce the complexity while retaining the characteristics of the physical processes which produce streamflow. An example of a simple conceptual model is the *abc* model (Fiering, 1967). It is a 4 parameter state based annual streamflow model with only one storage element and is represented schematically in Figure 1.2. The model proportions the precipitation input, accumulated in an interval, into three parts. These are Baseflow recharge (given by aP_t) which moves to a groundwater storage, the second component is an evapotranspiration loss which is removed from the system (given by bP_t) and the Quickflow portion which contributes directly to streamflow (given as $(1\text{-}a\text{-}b)P_t$). The groundwater storage contributes a proportion to the total streamflow so that

$$Q_t = (1\text{-}a\text{-}b)P_t + cS_{t-1}$$

where

$$S_t = S_{t-1} + aP_t - cS_{t-1}$$

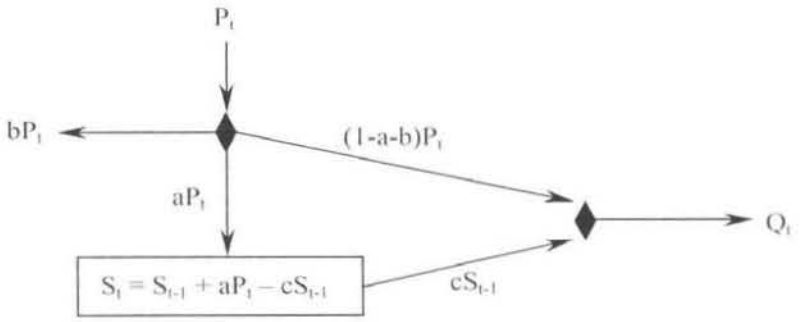


Figure 1.2 : Schematic of the *abc* model

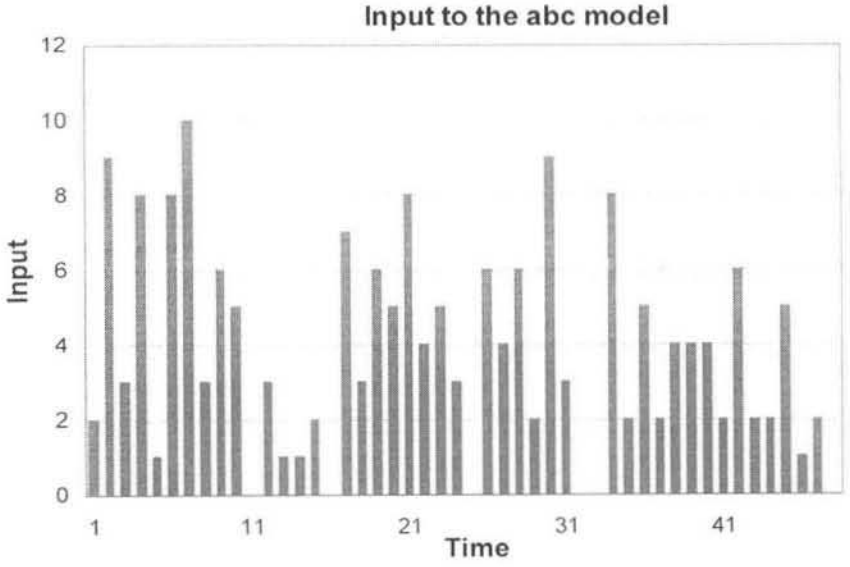


Figure 1.3: Example input sequence to the abc model

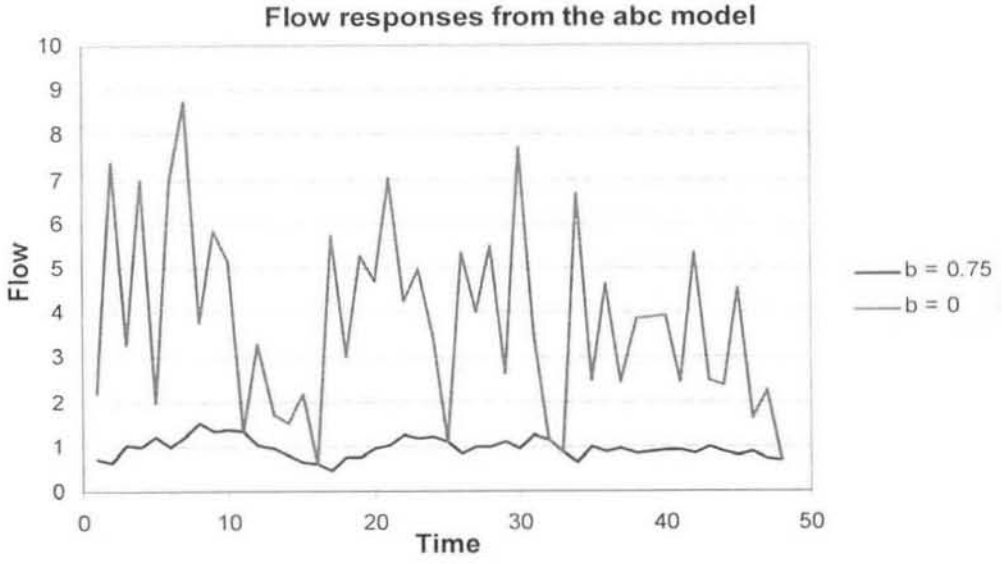


Figure 1.4: Generated flow responses from the abc model

Figure 1.4 illustrates the effect of the loss term in the *abc* model. The response of the model to a randomly generated sequence of pulses, between zero and ten units (Figure 1.3), is plotted for two separate values of the loss parameter *b*. The other two parameters are fixed. There is a clear reduction in the magnitude of the streamflow (as expected) for the case where the loss parameter is used. In addition, there is a slight smoothing effect which is not just an artifact of the difference in scale between the two sequences. It appears to come from the fact that a relatively larger proportion of the rainfall is routed through the storage.

The *abc* model uses simple formulations for each of the major processes occurring to produce a streamflow estimate. Combinations of linear reservoirs (such as the cell models to be introduced in Chapter 2) may also be used to represent certain physical components of the catchment such as groundwater storage and surface runoff. Many conceptual arrangements can be used. Some successful conceptual models are more complex. Examples of such models are TOPMODEL and the ARNO (Todini, 1996) model. A basic description of TOPMODEL is given below.

TOPMODEL can be viewed as a simplified aquifer model (AFORISM, 1996). The soil moisture accounting module keeps a record of the depth of the water table. The depth is expressed in terms of the amount of water needed to bring the water table to the surface. If the water table reaches the surface in a particular grid square, then that area is considered to be temporarily impervious and all rainfall input to the area (other than that lost through evapotranspiration or interception storage) becomes runoff. The soil storage is initialized by assuming that the catchment has experienced a long dry spell. The initial depth of the water table can then be calculated from the streamflow, because there is no direct runoff occurring. The streamflow in the channel is composed of direct runoff from the temporarily impervious areas and lateral inflow from the soil storage. Routing of the flow down the channel is achieved using a Kinematic or Parabolic (Franchini et al., 1996) scheme. TOPMODEL can be parsimoniously described by three parameters and a topological index (Franchini et al., 1996).

CHAPTER 1 INTRODUCTION

Summary:

The major physical processes involved in the conversion of rainfall to runoff have been presented and briefly discussed. A number of different modelling paradigms have been compared and the notion of linear transfer functions (to be expanded on in Chapter 2) has been introduced. It is a conceptual transfer function type of catchment model, which will be focussed upon in the remainder of the text. The loss component, which is part of the abc model discussed in this chapter, becomes an integral part of the catchment model formulated in Chapter 3 and applied in Chapters 4 and 5.

CHAPTER 2 LINEAR TRANSFER FUNCTIONS

The modelling strategy adopted in this study was based on the choice of a linear transfer function approach rather than a physically based catchment model. This choice was made in the interests of parsimony as the larger number of parameters often associated with physical models can be difficult to estimate. Jakeman & Hornberger (1993) provide many examples from hydrological literature supporting this premise and according to Bevan (1989):

> "There is a great danger of overparameterization if it is attempted to simulate all hydrological processes thought to be relevant, and fit those parameters by optimization against an observed discharge record...It appears that three to five parameters should be sufficient to reproduce most of the information in a hydrological record."

Transfer functions also have reduced data requirements, for calibration, when compared with physical models; records of rainfall and subsequent streamflow are sufficient. In this case, where online updating will require improving estimates of the parameters in real time to reduce differences between forecast and observed flows it is important that the model be computationally efficient. A conceptual model based on a linear transfer function approach possesses reduced catchment data collection requirements, has an advantage over physically based models in terms of the number of parameters required and is computationally highly efficient.

These ideas will be extended with the introduction of a particular type of linear model, the Linear Reservoir. Effective conceptual catchment models can be designed from combinations of linear reservoirs arranged in series or parallel. The response functions for individual reservoirs can be defined in terms of exponential IRF's from which outputs can be easily calculated. The responses from various combinations of reservoirs are also well known, and in many cases can be represented exactly by a gamma function or in a difference equation form. The response functions are based on the continuity relation, meaning that the input to these models must be the effective rainfall (from the model's point of view) for the catchment. The losses must therefore be separately determined.

CHAPTER 2 LINEAR TRANSFER FUNCTIONS

2.1 SYSTEM THEORY

A system can be defined as a conceptual or physical object that produces a response to input. The transformation of an input to output does not have to be linear in nature. A catchment can be considered to be a system that receives input (precipitation) and produces an output (streamflow). If the non-linearity's can be considered to integrate out over time then a catchment may be considered as a linear system and modelled using a linear Transfer Function (Chow et al., 1988).

If a system operates on an input $r(t)$ to produce an output $q(t)$, then the system is linear if, for $r_1(t)$ producing $q_1(t)$ and $r_2(t)$ producing $q_2(t)$ it follows that $r_3(t) = r_1(t) + r_2(t)$ results in $q_3(t) = q_1(t) + q_2(t)$; and $cr(t)$ produces $cq(t)$ where c is a constant. The simplest case to deal with is that of a system which accepts a single lumped input and produces a single output. In the case of hydrology these are: the mean rainfall rate over the catchment area being considered and the streamflow response from this area. For this case, the output to an infinitesimally short duration pulse of unit volume is given by the impulse response $u(t)$. The unit impulse is defined as the *Dirac delta* function $\partial(\tau)$:

$$\int_{-\infty}^{\infty} \delta(t - t_0)\, dt = 1$$

$$\delta(t - t_0) = 0 \quad \text{when } t \neq t_0$$

$$\partial(0) = \infty$$

A general input $x(t)$ can be thought of as being made up of an infinite number of weighted delta functions

$$x(t) = \int_{-\infty}^{\infty} \delta(t - \tau) x(\tau)\, d\tau$$

The impulse response $u(t)$ is defined as the system response when the input $x(t)$ is a delta function (Dooge, 1973), i.e when $x(t) = \delta(t)$ then the system output $q(t) = u(t)$. Therefore, for a general continuous input sequence $x(t)$ (a series of impulses) the output $q(t)$ is given by integrating the weighted output from each of the delta functions (since the system is linear)

$$q(t) = \int_{-\infty}^{\infty} x(\tau)u(t-\tau)\,d\tau$$

This operation is a convolution often represented as $q(t) = x(t) * u(t)$.

If we consider the system to be causal (output only begins after the input has occurred) so that the limits of integration can be changed; the response to an isolated input starting at time zero can be represented as

$$q(t) = \int_{0}^{t} x(\tau)u(t-\tau)\,d\tau$$

which is the continuous response for a causal, time invariant, linear system to a general input $x(t)$.

The continuous form of the convolution integral is however not very useful when dealing with discrete data so a discrete version may be developed. A rectangular pulse of length d and unit volume can be defined to replace the Dirac delta function

$$P(t-kd) = \frac{1}{d} \qquad \text{for} \qquad kd < t < (k+1)d$$

$$P(t-kd) = 0 \quad \text{for all other } t$$

As for the continuous case, a general form of input is given by a weighted summation of pulses

$$r(t) = \sum_{k=-\infty}^{\infty} r(kd)P(t-kd)$$

In place of the impulse response $u(t)$ we define $h(t)$ as the system pulse response function. Where $q(t) = h(t)$ if the input to the system is a unit volume pulse at a rate of $1/d$ over a finite period d. In that case, the discrete convolution equation for the system can be written as

$$q(t) = \sum_{k=-\infty}^{\infty} r(kd)h(t-kd)$$

If we again consider a causal system, we can change the limits of the summation to yield

$$q(t) = \sum_{k=0}^{t} r(kd)h(t-kd) \qquad (2.1)$$

where t is the current time and $h(t)$ is the continuous system pulse response function. The output $q(t)$ and the pulse response $h(t)$ are defined continuously and the input $r(t)$ is defined in a discrete manner.

A discrete formulation of equation 2.1 is

$$q_t = \sum_{i=0}^{t} r_i h_{t-i} \qquad (2.2)$$

where t is the time measured in equally spaced intervals, q_t is the system output sampled at an instant, r_i is the input pulse and h_t the ordinates of $h(t)$ sampled at evenly spaced discrete intervals.

The operations involved in predicting the response from a system with a known response function are linear. This ease of calculation (by convolution) is what makes system theory so attractive, despite its limitations in dealing with non-linearity (Dooge, 1973).

2.2 System Response Functions

The response of a linear system to various types of input can be characterized by its response functions. These are the Pulse, Step and Impulse responses. For an input of a unit volume at a rate of $1/d$ over a finite period d the system output is the d period unit pulse response. The system response to any number of pulses can be found by a convolution of the input pulses with the impulse response function or discretely by numerical convolution with the discrete unit pulse response. The input must however be at constant rates over intervals of d. The step response function is simply an accumulation of the ordinates of a constant pulse response of a unit rate and must therefore reach a maximum value of unity. The usefulness of the step response function is in deriving pulse responses for different period pulses. An impulse response function characterizes the response of a system to a unit input occurring in an infinitesimally short space of time. Figure 2.1 (Chow et al., 1988 : p209) illustrates these different response functions for a single linear reservoir with a storage constant of 3 time units and a pulse input length of 2 time units. The equations governing the linear reservoir's response will be developed later in Sections 2.5 and 2.6.

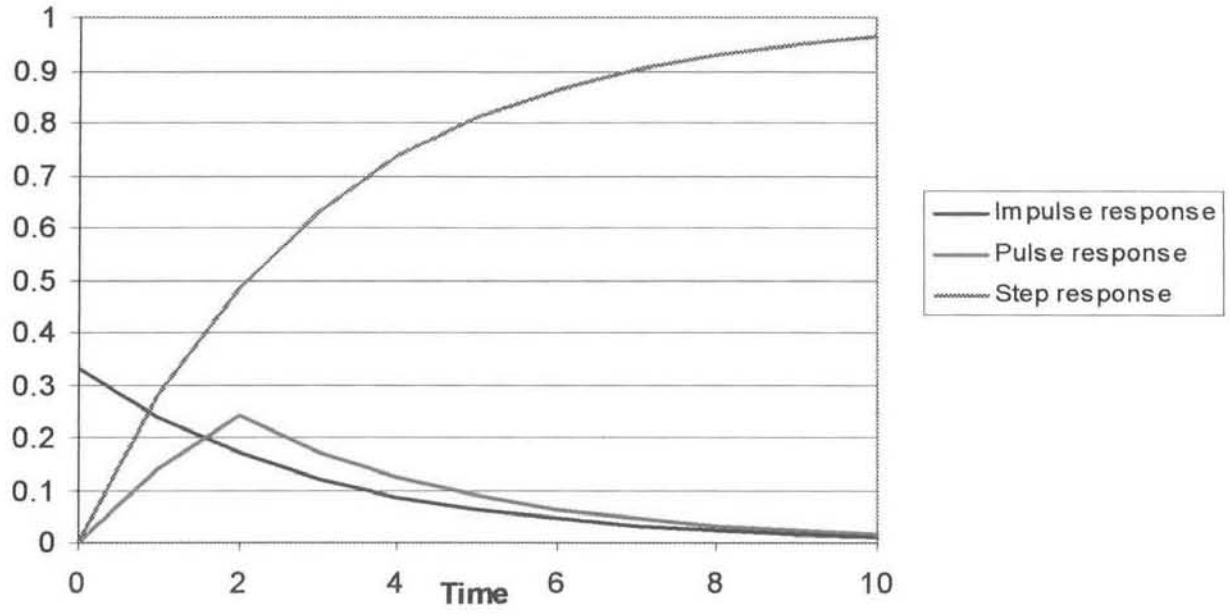


Figure 2.1 : Response functions for a linear reservoir (after Chow et al., 1988)

2.3 IDENTIFICATION OF RESPONSE FUNCTIONS

The calculation of system outputs using the convolution equation relies on having some knowledge of the system's response function. Identification of this response function is therefore an important aspect of the modelling procedure. The classical approach is: given an appropriate input (effective rainfall) and the specification of a linear transfer function model, it is computationally straightforward to obtain the output (Direct Runoff) via convolution. However the difficulty lies in finding a suitable transfer function model for the catchment based on historical observations of rainfall and runoff. The problem of model identification is one of deconvolution. If the system were truly linear, the input and output known, and error free, then the determination of a unique pulse response function is straightforward.

This could be done in several ways. The most obvious is to solve the set of convolution equations simultaneously in reverse to obtain the values of the pulse response function. Let r_t and q_t , $t = 1, 2, \ldots, n$ respectively be the (known) sequence of pulsed input to and the instantaneous records of output from the linear system described in discrete form by h_k ; $k = 0, 1, \ldots, p$, the System Operation Function (Dooge, 1973 : p10) or Transfer Function (Chow et al., 1988 : p203). Then the output is given as for equation 2.2

$$q_t = \sum_{i=0}^{t} r_i h_{t-i} \qquad \text{for} \qquad t \le p$$

$$q_t = \sum_{i=0}^{p} r_i h_{t-i} \qquad \text{for} \qquad t > p$$

For the situation where $t > p$ the upper limit of the summation is set equal to p since the unit pulse response has a finite length, after which all it's ordinates are zero. The idea is then to find the coefficients of h_t. However to reverse this calculation procedure (a procedure often called deconvolution) is inherently numerically unstable. The coefficients of h_t must be non-negative in order to satisfy physical constraints since a positive input cannot result in a negative flow response. The measurement errors encountered in real data, along with the non-linearity inherent in the system, invariably result in negative coefficients being produced in h_t when derived by deconvolution.

2.3.1 Transform Methods

Transform methods (e.g. Fourier or Laplace) can be used to determine the pulse response function. A suitable transform is found for which the deconvolution equivalent in the transformed space is a simple operation. Hence the transform of the pulse response can be easily found and back transformed to find the actual pulse response.

This can be accomplished using (for example) the Fourier Transform as follows :

If $H(s)$ and $R(s)$ are respectively the Transfer Function (in the frequency domain) and the (Fourier) Transform of the input sequence, then $\hat{Q}(s)$, the estimate of the Transformed output (assuming that $H(s)$ is known) is given as

$$\hat{Q}(s) = R(s)H(s)$$

Alternatively if $H(s)$ is not known, we can (theoretically) obtain $Q(s)$ from the observed output and find

$$\hat{H}(s) = Q(s)/R(s)$$

This can then be back-transformed to obtain $\hat{h}(t)$. This is of course all well known (Dooge, 1973).

Investigations of transform methods making use of the Fast Fourier Transform did not produce satisfactory results, because of the instabilities alluded to above. An example of a Transfer Function (TF) found in this way is shown in Figure 2.2, also shown is the input and output. The rainfall and subsequent streamflow response were obtained from observed data on the Liebenbergsvlei. The oscillations clearly shown in the tail, of the derived transfer function, are typically produced from this kind of analysis. This is the numerical instability commonly encountered in numerical deconvolution and differentiation. Note that the area under the transfer function in Figure 2.2 is ≈ 0.3. This is the ratio of the total output to the total input, giving an idea of the proportion of observed rainfall that becomes effective rainfall on the Liebenbergsvlei catchment. Trying to find a transfer function which relates input and output directly, as was done here, complicates the situation since the transfer function is required to do two things; first it must convert observed to effective rainfall and second it must convert this effective rainfall into direct runoff. It may prove beneficial to perform these conversions in two separate steps.

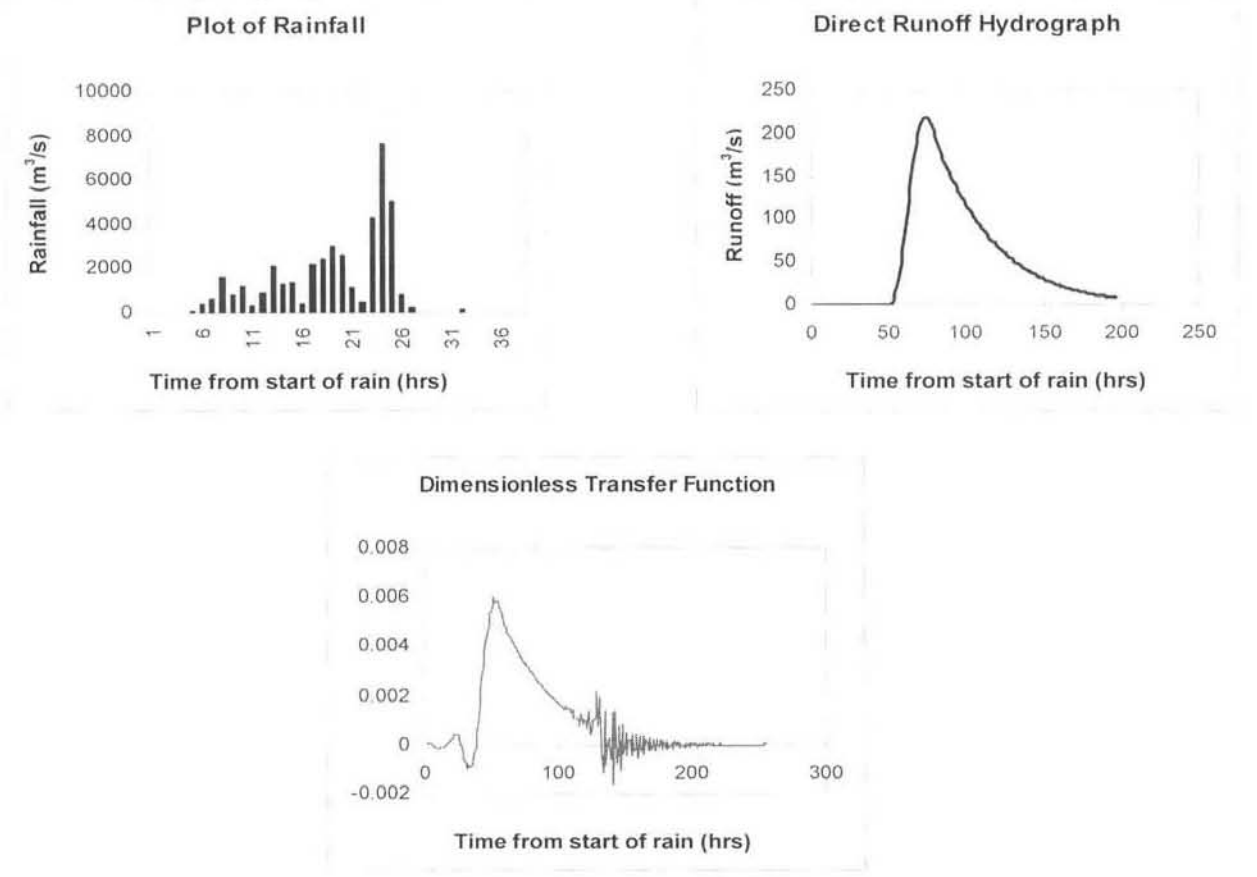


Figure 2.2: Transfer Function found making use of Fast Fourier Transforms

2.4 THE INSTANTANEOUS UNIT HYDROGRAPH (IUH)

The Unit Hydrograph was first presented by Sherman (1932), and is defined as the direct runoff hydrograph resulting from a unit depth of excess precipitation occurring over a finite time period. It is catchment specific and must be defined in terms of the storm duration from which it was derived (Chow et al., 1988). It is fairly easy to see that the unit hydrograph is actually equivalent to a discrete pulse response function for the catchment.

The Unit Hydrograph requires that the following assumptions hold for the catchment of concern

- *The properties of the catchment are time invariant.* This implies that for any identical storms the runoff response of the catchment will be identical. This property does not hold exactly, since the past rainfall history affects the wetness of the catchment and hence the proportion of runoff which will occur from a given storm.
- *The spatial distribution of rainfall over the catchment is uniform.* This requirement will not hold even if the catchment is subdivided into small sub-areas. However, if the number of sub-areas is large enough, each may be treated this way.
- *The rainfall intensity during the storm remains constant during each discrete interval.* This is not true. However, if the time intervals are short enough, compared to the response time of the individual sub-areas, then this could be assumed to hold.
- *The principle of linear superposition holds.*

The IUH is equivalent to the impulse response of a catchment. It is a more general form than the unit hydrograph, since it can be applied to any combination of inputs.

A derivative of the IUH is the Geomorphological Unit Hydrograph (Rodriguez-Iturbe & Valdes, 1979). The GUH theory attempts to make use of physical catchment characteristics such as stream-length and drainage patterns to establish what the IUH should be on previously ungauged catchments. Nash and Shamseldin (1998), in a recent review of the theory, suggest a relationship between the complexity of the drainage pattern and the skewness of the resulting IUH but conclude that this relationship is untested and that the scale of the IUH has not been determined from geomorphological attributes. They went on

to show that a linear combination of Gamma functions could equally well substitute for the GUH. These are convolutions of exponential IRF's which come from trees of cascades of linear reservoirs. They are strictly only applicable to direct runoff. This still leaves the problem of determining the effective rainfall input to the system.

2.5 RESPONSE FUNCTIONS OF LINEAR STORAGE ELEMENTS

Each linear reservoir relates storage *(S)* to outflow *(q)* using the relation $S = kq$ where k is a constant. The impulse, step and pulse response functions can be derived from this relation and the continuity equation:

$$\frac{dS}{dt} = I(t) - q(t)$$

where *I(t)* is the continuous input as a function of time.

Following the development of Chow et. al. (1988) the (unit) Impulse response *u(t)* of a single reservoir is thus found to be

$$u(t) = \frac{1}{k} e^{-t/k}$$

The Step response *g(t)* is the response of the reservoir to a continuous input at a unit rate and is

$$g(t) = 1 - e^{-t/k}$$

The unit Pulse response function *h(t)* gives the response of the reservoir to an input of unit amount over a time period Δt. This implies that the input is occurring at a constant intensity of $1/\Delta t$ during the time period from 0 to Δt and is zero elsewhere. The pulse response function *h(t)* is

$$h(t) = \frac{1}{\Delta t}(1 - e^{-t/k}) \qquad \text{for} \quad 0 < t \le \Delta t$$

$$h(t) = \frac{1}{\Delta t}[e^{\Delta t/k} - 1]e^{-t/k} \qquad \text{for} \quad t > \Delta t$$

$$h(t) = re^{-t/k}$$

where r is a constant. Examples of these for $k = 3$ are given in Figure 2.1.

This treatment can be generalized so that a number of reservoirs can be combined in series or parallel to produce a suitable linear catchment model. The pulse response functions from the individual elemental storages are combined in a catchment model pulse response function, which may consist of a larger number of ordinates than the number of different storage response parameters defining the model. There is thus a hidden structure in this type of model, as well as parsimony, with the distinct advantage that all the ordinates of *h(k)* are guaranteed non-negative.

2.6 LINEAR RESERVOIR MODELS

A number of different conceptual models making use of linear reservoirs have been proposed and used successfully. Formulations for various arrangements of reservoirs have been developed. Nash (1957) suggested that a catchment could be represented by a series of n linear reservoirs with identical storage constants k. The output from the catchment is the convolution of the input to the last reservoir in the chain with its impulse response function. Since the input to this reservoir is given by the impulse response of the previous reservoir in the chain, it is clear that the total impulse response of all the reservoirs is the result of n successive convolutions of the reservoir response function. The resulting general impulse response function for this arrangement is

$$u(t) = \frac{1}{k\Gamma(n)}\left(\frac{t}{k}\right)^{n-1} e^{-t/k} \qquad (2.3)$$

The function *u(t)* is the gamma probability distribution function. $\Gamma(n)$ is the gamma function; its values can be read from standard tables. $\Gamma(n)$ is equivalent to *(n-1)!* for integer values of n.

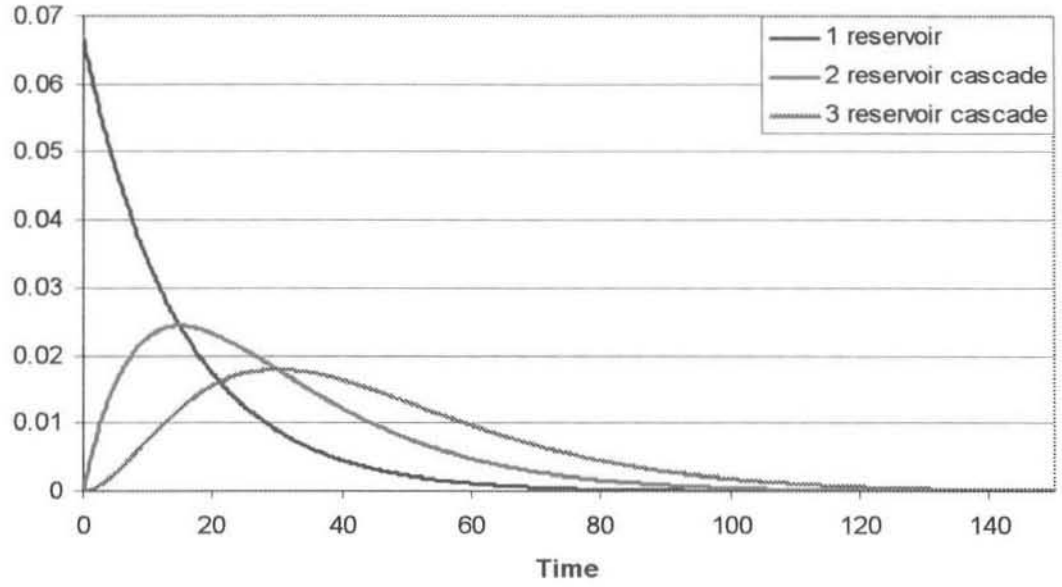


Figure 2.3: Impulse response functions for a single reservoir and cascades of equal reservoirs (with $k = 15$ time periods).

Figure 2.3 shows the effect on the system impulse response function, of routing inputs through a cascade of equal reservoirs. These are impulse responses for the Nash (1957) linear cascade shown in Figure 2.4. The most striking difference between the three IRF's shown is the lag time which is introduced between the start of the response and the peak. For a single reservoir there is no time delay, with the reservoir instantly filled at time zero, the response begins immediately. The peak response occurs at the time when the last reservoir in the chain has its maximum storage, hence the peak response for the case of a single reservoir is at time zero. As more reservoirs are added to the series, the peak occurs later. There is also an attenuation effect as the number of reservoirs is increased. The magnitude of the peak is reduced and the mean length of the response (of significant magnitude, since the response is infinite) is increased with the number of reservoirs. By introducing a lag, and smoothing the input sequence, it is possible for realistic looking hydrographs to be produced when routing "noisy" rainfall inputs through arrangements of linear reservoirs.

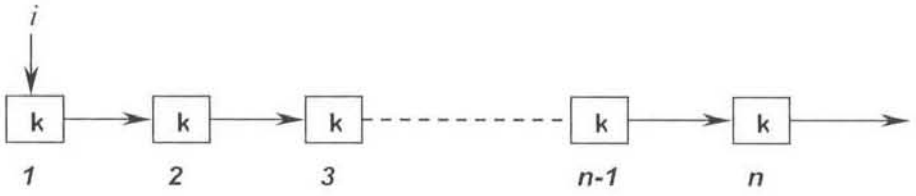


Figure 2.4: The Linear cascade (Nash, 1957)

Dooge (1959) presented a distributed input cascade model. This formulation was different from the Nash (1957) linear cascade in that it divided the total effective input equally between each reservoir in the chain. Figures 2.4 and 2.5 illustrate the difference between the linear cascade and the distributed input cascade.

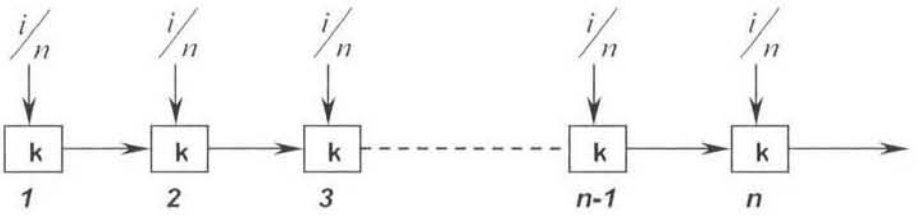


Figure 2.5: The Distributed input cascade (Dooge, 1959)

Linear reservoirs can be used successfully to produce semi-distributed cell models (Pegram & Diskin, 1987a). This type of model treats the catchment to be modeled as a number of cells. Each cell is a sub-catchment of the larger catchment and the divisions are usually made along watershed boundaries. The individual cells are then treated as lumped models each represented by an arrangement of linear reservoirs. The input to each cell is usually the *effective* rainfall for that part of the catchment and the outputs of upstream cells. This implies that some kind of loss function or model is applied to the observed rainfall in order to calculate the *effective* precipitation. It is interesting to note that the distributed input cascade (Dooge, 1959) is in effect a cell model with a single reservoir to represent each cell and an assumption of a uniform rainfall distribution over the catchment (Pegram & Diskin, 1987a). The inputs to each cell are the distributed rainfall input and the output from the upstream reservoir. A similar cell model presented by Diskin & Simpson (1978) consists of a cascade of pairs of unequal reservoirs. In this model, each cell could receive a different

rainfall input, equal to the effective rainfall occurring over its area. Laurenson (1964) presented a cell model with a cascade of non-linear storages, each receiving as input the rainfall excess for the cell and the output from the upstream cell. Pegram & Diskin (1987a) cite a model (Diskin, 1984) in which the channel input to each cell is routed through a single reservoir and the rainfall input through a pair of unequal reservoirs. The combined output from these is then the channel input to the next cell in the series.

The Manifold cell model (Pegram & Diskin, 1987a, b and c) is similar to a distributed input cascade with the precipitation input to each cell first routed through a single reservoir. Each cell can be thought of as having a surface runoff element and a channel routing element. The surface element for each cell receives a portion of the distributed input for the catchment. The resulting output from the surface element is then routed by a channel element, which has a response parameter proportional to the distance of the cell from the catchment outlet. The response parameter for the surface element is assumed constant throughout the catchment, but the input to each cell is scaled by the proportion of its area to the total catchment area.

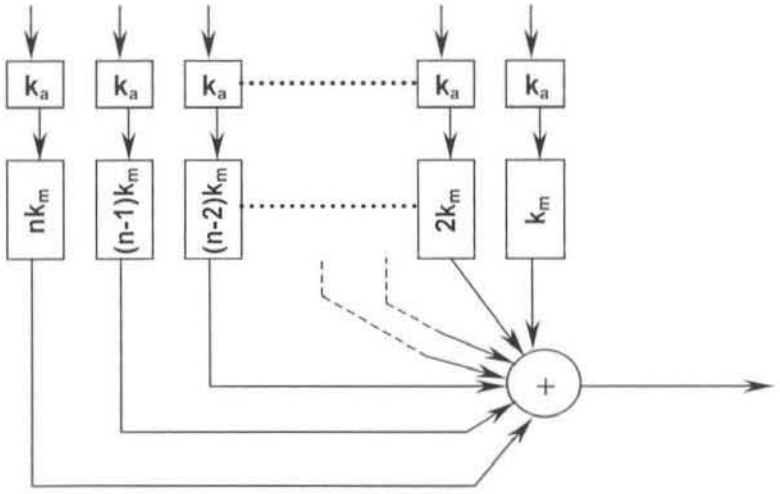


Figure 2.6: The Manifold cell model (Pegram & Diskin, 1987a)

CHAPTER 2 LINEAR TRANSFER FUNCTIONS

Summary:

This chapter introduced the concept of a linear transfer function and it's uses for determining the streamflow from a catchment. The system response function can be used as part of a parsimonious conceptual catchment model, with convolution equations providing a (computationally efficient) procedure for calculating the system output from a given sequence of inputs. These ideas were extended with the introduction of a particular type of linear model, the Linear Reservoir.

Effective conceptual catchment models can be designed from combinations of linear reservoirs arranged in series or parallel. The response functions for individual reservoirs are defined in terms of exponential IRF's from which outputs can be easily calculated. The responses from various combinations of reservoirs are also well known, and in many cases can be represented exactly by a gamma function or in a difference equation form. The response functions are based on the continuity relation, meaning that the input to these models must be the effective rainfall (from the model's point of view) for the catchment. The losses must therefore be separately determined.

Linear reservoirs can be used to develop semi-distributed cell models. The individual cells are treated as lumped areas modeled by an arrangement of linear reservoirs. The parsimonious structure of this kind of model is attractive when parameter fitting processes are considered. In Chapter 3, the ideas presented by the Manifold model will be extended to a 3 reservoir feed-forward model that is semi-distributed in nature.

CHAPTER 3 MODEL DEVELOPMENT AND OPTIMIZATION

Discretely coincident difference equation forms of continuous linear models are a convenient way to handle the discrete nature of hydrological data because they have a small number of terms, and their formulation as a convolution lends itself to high speed computation. The difference equation forms derived here are similar in appearance to the Auto-Regressive Moving Average (ARMA) time series model described by Box and Jenkins (1970). However, there is a functional relationship between the linear reservoir parameters and the ARMA coefficients, which guarantees stationarity. The discretely coincident form for a general 3 reservoir feed forward cell model is derived and shown to collapse to a number of particular forms. The use of this model to represent a large catchment consisting of many cells will be described in Chapter 4.

The model parameter fitting process required the use of an optimization technique suitable for minimization of a function that may exhibit highly non-linear behaviour. Monte Carlo techniques are employed to ensure that a global minimum is found. A modified simplex search routine incorporating the Metropolis algorithm and using Monte Carlo techniques was used as the function minimization tool. A least squares fitting criterion was used as a measure of the "goodness" of fit between the observed and calculated hydrographs.

3.1 Difference Equations and Time Series Analysis

Various models for the analysis of discrete time series were described by Box and Jenkins (1970). These types of model had in fact been applied to Hydrology since the introduction of the unit hydrograph (Sherman, 1932), which is simply a linear filter transforming excess rainfall to direct runoff. The distinguishing feature being that the input to the Box-Jenkins ARMA-models is assumed to be white noise, whereas the input to the hydrological linear models is effective rainfall. Quimpo (1967) used a moving average model to generate daily streamflows and later (1971) showed that the autocorrelation function for white noise routed through a single linear reservoir is a first order autoregressive process. O'Connor (1976) showed that cascades of linear reservoirs could be modelled, in some cases, by autoregressive and ARMA models of the type described by Box and Jenkins. Earlier Spolia and Chander (1974) had introduced a similar ARMA formulation for cascades of reservoirs with the particular case of two reservoirs in series being presented in detail. They had also established that a structural relationship exists between the ARMA coefficients and the

reservoir parameters. Pegram (1980) derived an ARMA representation for a general multi-layered multi-compartmental reservoir model, using a State-Space approach. He further estimated the effective input for the model from observed streamflow records. O'Connor (1982) provides a transfer function approach for determining the discretely coincident difference equation forms of continuous parametric models such as the Nash (1959) cascade. In their introduction of the manifold cell model, Diskin & Pegram (1987c) derived the discretely coincident pseudo-ARMA form for two unequal reservoirs in series.

3.1.1 AUTOREGRESSIVE MODELS

Autoregressive time series models make use of a linear combination of past values of the process, to be modelled, as a means of predicting future values. Following Box and Jenkins (1970) treatment, let z_{t-i} for $0 \leq i \leq n$ be the sampled values of an assumed stationary Gaussian process at equally spaced temporal intervals. Further let y_{t-i} be the deviations of the process from it's (stationary) mean. The process may then be modelled as an Autoregressive process of order p, with the following form

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \ldots + \phi_p y_{t-p} + a_t$$

Where the ϕ_j's are the autoregressive weights and a_t is a random shock term. The shock term is drawn from a Normal distribution assumed to have zero mean and a variance which must be estimated from past observations of the process to be modelled. The introduction of the backward shift operator B, for which $By_t = y_{t-1}$ and $B^i y_t = y_{t-i}$, allows the AR model to be written in a more condensed form as

$$\phi(B) y_t = a_t$$

where the AR operator is

$$\phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \ldots - \phi_p B^p$$

3.1.2 MOVING AVERAGE MODELS

A moving average model defines the current value of a process as a linear combination of white noise shocks a_t. Thus a Moving Average model of order q is

$$y_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \ldots - \theta_q a_{t-q}$$

where the moving average weights θ_i are not constrained to be positive nor sum to unity. As was the case for the Autoregressive model, we can define a Moving Average operator

$$\theta(B) = 1 - \theta_1 B - \theta_2 B^2 - ... - \theta_q B^q$$

such that the Moving Average model may be written in a condensed form as

$$y_t = \theta(B) a_t$$

3.1.3 Combination Autoregressive-Moving Average Models

The specification of a mixed Autoregressive Moving Average (ARMA) model is given as follows

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + ... + \phi_p y_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - ... - \theta_q a_{t-q} \quad (3.1)$$

This is an ARMA (p,q) process. It follows easily that equation 3.1 can also be expressed in terms of the Autoregressive and Moving Average operators, defined earlier, giving the condensed form for a general ARMA model as

$$\phi(B) y_t = \theta(B) a_t$$

Box and Jenkins (1970) also show that this form is equivalent to a transfer function model

$$Y_t = v_0 X_t + v_1 X_{t-1} + v_2 X_{t-2} + ...$$

where Y_t is the response of a system to a sequence of inputs X_t, X_{t-1}, X_{t-2}..., and the weights v_0, v_1, v_2... represent the impulse response function described in Chapter 2.

3.2 Discretely Coincident Difference Equation Forms

The discretely coincident form of a continuous model refers to a discrete model formulation that is exactly coincident with the continuous form, at discrete sample points (O'Connor, 1982). In the particular case of linear cell models, the combinations of (continuous) exponentials, which define the pulse response functions of an arrangement of linear reservoirs, may be exactly represented at discrete timesteps by difference equations (Spolia & Chander, 1974; O'Connor, 1976; Pegram, 1980; O'Connor, 1982; Diskin & Pegram, 1987; Jakeman & Hornberger, 1993). The form of these difference equations is similar in appearance to the ARMA time series models of Box and Jenkins (Equation 3.1). The formulation is as follows

$$y_t = \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{j=0}^{q} \theta_j x_{t-\tau-j}$$

where the x_t's are the pulsed rainfall inputs to the model, τ represents the lag between the precipitation input and the time at which it produces a response from the model and y_t is the resulting streamflow response.

3.3 Formulation of a Model for a Single Catchment Cell

The feed forward cell model comprising three linear reservoirs, presented here differs from the specification of the cell models discussed in Chapter 2. The models previously mentioned are all conservative in nature, meaning that the volume of output must equal the volume of input, as no losses can occur in the system. The specification given here allows losses to occur from any or all of the linear reservoirs in the conceptual arrangement comprising each cell. Essentially this allows a single transfer function (in difference equation form) to represent the entire rainfall to streamflow conversion process. It is suggested that this approach is novel.

The exponential nature of the reservoir pulse response function results in a loss removal which is also exponential. The majority of the loss models described in Chapter 1 have formulations that are based on an exponential decay term; these models attempt to mimic the physical processes of infiltration. It follows that allowing the losses to "drain" from the reservoirs is analogous, in some sense, to the physical processes of infiltration thought to be the major contributing factor to losses. In this approach, as in Fiering's *abc* model (Fiering, 1967), there is no need for a separate loss model to estimate the effective rainfall; input to the model is simply the observed (pulsed) rainfall at an appropriate lag. This extension to the linear modelling approach which has been a core element in this study, makes a useful addition to the modelling armoury.

The discretely coincident ARMA formulation, for the general linear three reservoir feed forward model with losses (Figure 3.1), is determined following Diskin and Pegram's (1987c) treatment for a cascade of two unequal reservoirs, and was developed specifically for this study.

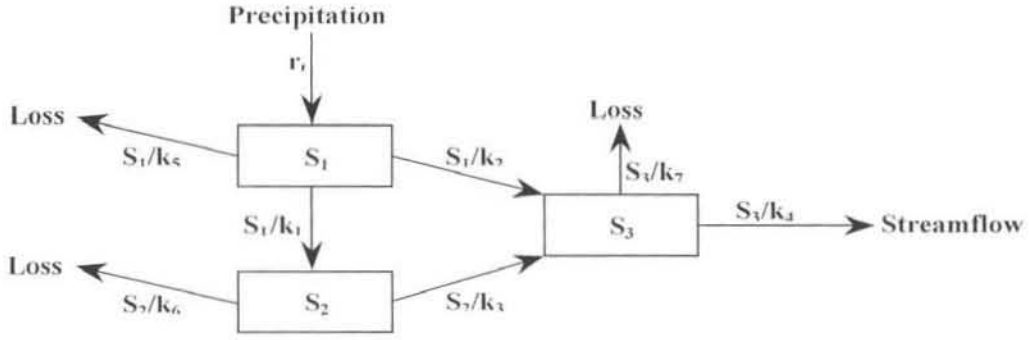


Figure 3.1: A general linear 3 reservoir feed forward model with (possible) losses from each reservoir.

The (continuous time) State-Space representation for the arrangement in Figure 3.1 is given by the following set of differential equations

$$\dot{S}_1(t) = -\left(\frac{1}{k_1} + \frac{1}{k_2} + \frac{1}{k_5}\right) S_1(t) + r(t-\tau)$$

$$\dot{S}_2(t) = \frac{1}{k_1} S_1(t) - \left(\frac{1}{k_3} + \frac{1}{k_6}\right) S_2(t)$$

$$\dot{S}_3(t) = \frac{1}{k_2} S_1(t) + \frac{1}{k_3} S_2(t) - \left(\frac{1}{k_4} + \frac{1}{k_7}\right) S_3(t)$$

where $\dot{S}_i(t)$ is the time derivative of the storage in the i'th reservoir at time t, $S_i(t)$ is the sequence of storage's in the i'th reservoir, the k_i's are the response constants for each of the reservoirs and $r(t-\tau)$ is the lagged (by τ intervals) sequence of precipitation inputs to the system. The resulting streamflow $y(t)$ is

$$y(t) = \frac{1}{k_4} S_3(t)$$

To simplify notation, let $a = 1/k_1$, $b = 1/k_2$, $c = 1/k_3$, $d = 1/k_4$, $e = 1/k_5$, $f = 1/k_6$ and $g = 1/k_7$. The derivation of the extended model follows Diskin and Pegram (1987c) closely.

State Equations :

$$\begin{aligned}
\dot{S}_1(t) &= -(a+b+e)S_1(t) && && +r(t-\tau)\\
\dot{S}_2(t) &= aS_1(t) && -(c+f)S_2(t) &&\\
\dot{S}_3(t) &= bS_1(t) && +cS_2(t) && -(d+g)S_3(t)
\end{aligned}$$

$$y(t) = dS_3(t)$$

These equations can be written in matrix form, omitting the integer lag τ (for simplicity)

$$\dot{S}(t) = AS(t) + \beta r(t) \tag{3.2}$$

$$y(t) = C^T S(t) \tag{3.3}$$

where

$$A = \begin{bmatrix} -(a+b+e) & & \\ a & -(c+f) & \\ b & c & -(d+g) \end{bmatrix}$$

$$\beta = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad \text{and} \qquad C^T = \begin{bmatrix} 0 & 0 & d \end{bmatrix}$$

Solving equation 3.2 for $S(t)$ and substituting into equation 3.3 yields

$$y(t) = C^T \left\{ e^{A(t-t_0)} S(t_0) + \int_{t_0}^{t} e^{A(t-\alpha)} \beta r(\alpha) d\alpha \right\}$$

Now making the assumption of pulsed input r_k and writing in a discrete form (introducing some new notation)

$$S_k = e^{A\Delta t} S_{k-1} + \beta r_k \int_0^{\Delta t} e^{A(\Delta t - \alpha)} d\alpha \tag{3.4}$$

$$S_k = GS_{k-1} + H\beta r_k \tag{3.5}$$

$$y_k = C^T S_k \tag{3.6}$$

Introducing the backward shift operator B where $BS_k = S_{k-1}$ and $B^i S_k = S_{k-i}$ allows equation 3.5 to be rewritten as

$$(I - GB)S_k = H\beta r_k \tag{3.7}$$

Combining equations 3.6 and 3.7

$$y_k = (I - GB)^{-1} H\beta r_k \tag{3.8}$$

Using the fact that

$$(I - GB)^{-1} = \frac{adj(I - GB)}{\det(I - GB)}$$

equation 3.8 becomes

$$\det(I - GB)y_k = C^T adj(I - GB)H\beta r_k \tag{3.9}$$

Equation 3.9 is in the form $\phi(B)y_k = \theta(B)r_k$ i.e. in the form of a standard ARMA equation. We must now solve for the Autoregressive $\phi(B)$ and Moving Average $\theta(B)$ operators.

To evaluate $\phi(B)$ note that $G = e^{A\Delta t}$ can be written as $G = Ue^{D\Delta t}U^{-1}$ where UDU^{-1} is the canonical decomposition of A and D is the diagonal matrix of eigenvalues λ_i of A.

The λ_i are the solutions to $\det[\lambda I - A] = 0$ where

$$\lambda I - A = \begin{bmatrix} \lambda + a + b + e & & \\ -a & \lambda + c + f & \\ -b & -c & \lambda + d + g \end{bmatrix}$$

solutions are λ_1 = -(a+b+e), λ_2 = -(c+f) and λ_3 = -(d+g). (3.10)

It follows that the matrix $Q = e^{D\Delta t}$ has only diagonal elements $q_i = e^{\lambda_i \Delta t}$, so that we can now write

$$G = UQU^{-1}$$

One element in each column of U can be chosen arbitrarily (e.g Fraleigh & Beauregard, 1990)

$$U = \begin{bmatrix} 1 & & \\ u_{21} & 1 & \\ u_{31} & u_{32} & 1 \end{bmatrix} \quad \text{therefore} \quad U^{-1} = \begin{bmatrix} 1 & & \\ -u_{21} & 1 & \\ v & -u_{32} & 1 \end{bmatrix}$$

where $v = (u_{31} - u_{21}u_{32})$. We solve for the unknowns using the fact that $[A - \lambda_i I]u_i = 0$, where u_i is the i'th column of U. By successively substituting the λ_i's we can obtain the following solutions for the unknowns

$$u_{21} = \frac{a}{\lambda_1 - \lambda_2}$$

$$u_{31} = \frac{b + cu_{21}}{\lambda_1 - \lambda_3} \tag{3.11}$$

$$u_{32} = \frac{c}{\lambda_2 - \lambda_3}$$

Solving for G

$$G = \begin{bmatrix} q_1 & & \\ u_{21}(q_1 - q_2) & q_2 & \\ u_{31}q_1 - u_{21}u_{32}q_2 - vq_2 & u_{32}(q_2 - q_3) & q_3 \end{bmatrix}$$

where

$$q_1 = e^{-(a+b+c)\Delta t}$$

$$q_2 = e^{-(c+f)\Delta t} \tag{3.12}$$

$$q_3 = e^{-(d+g)\Delta t}$$

The Autoregressive operator $\phi(B)$ becomes

$$\boxed{\phi(B) = \det(I - GB) = (1 - q_1 B)(1 - q_2 B)(1 - q_3 B)}$$

To find H we note that the definite integral for the exponential of a matrix is given as

$$\int_{a}^{b} e^{A(\Delta t-\alpha)}d\alpha = UD^{-1}\left[e^{A\Delta t-a} - e^{A\Delta t-b}\right]U^{-1} \tag{3.13}$$

where U and U^{-1} are as before and D^{-1} is the inverse of the diagonal matrix of λ_i's. It follows from 3.4, 3.5 and 3.13 that

$$H = UD^{-1}\left[Q - I\right]U^{-1}$$

Omitting the tedious algebra

$$C^{T}adj(I-GB) = d\begin{bmatrix} X \\ u_{32}(q_2-q_3)(1-q_1B)B \\ (1-q_1B)(1-q_2B) \end{bmatrix}$$

where

$$X = u_{21}u_{32}(q_1-q_2)(q_2-q_3)B^2 + (u_{31}q_1 - u_{21}u_{32}q_2 - vq_3)(1-q_2B)B$$

also

$$H\beta = \begin{bmatrix} \dfrac{q_1-1}{\lambda_1} \\ u_{21}\left[\dfrac{(q_1-1)}{\lambda_1} - \dfrac{(q_2-1)}{\lambda_2}\right] \\ \dfrac{u_{31}(q_1-1)}{\lambda_1} - \dfrac{u_{21}u_{32}(q_2-1)}{\lambda_2} - \dfrac{v(q_3-1)}{\lambda_3} \end{bmatrix}$$

Introducing some additional notation for simplification, the Moving Average operator $\theta(B)$ becomes

$$\theta(B) = C^{T}adj(I-GB)H\beta$$

$$\boxed{\theta(B) = d\left\{\left[\alpha-\delta-\gamma\right] - \left[\alpha(q_2+q_3) - \delta(q_1+q_3) - \gamma(q_1+q_2)\right]B + \left[\alpha q_2q_3 - \delta q_1q_3 - \gamma q_1q_2\right]B^2\right\}}$$

where

$$\alpha = \frac{u_{31}(q_1 - 1)}{\lambda_1}$$

$$\delta = \frac{u_{21}u_{32}(q_2 - 1)}{\lambda_2} \qquad (3.14)$$

$$\gamma = \frac{v(q_3 - 1)}{\lambda_3}$$

Finally the difference equation representation of the model is found by expanding equation 3.9 and applying the AR and MA operators, this has similar form to an ARMA (3, 2) model

$$\boxed{y_k = \phi_1 y_{k-1} + \phi_2 y_{k-2} + \phi_3 y_{k-3} + \theta_0 r_{k-\tau} - \theta_1 r_{k-\tau-1} - \theta_2 r_{k-\tau-2}} \qquad (3.15)$$

with

$$\phi_1 = q_1 + q_2 + q_3$$

$$\phi_2 = -(q_1 q_2 + q_2 q_3 + q_1 q_3)$$

$$\phi_3 = q_1 q_2 q_3$$

$$\theta_0 = (\alpha - \delta - \gamma)d$$

$$\theta_1 = [\alpha(q_2 + q_3) - \delta(q_1 + q_3) - \gamma(q_1 + q_2)]d$$

$$\theta_2 = -(\alpha q_2 q_3 - \delta q_1 q_3 - \gamma q_1 q_2)d$$

where q_i for $i = 1, 2, 3$ are (See equation 3.12) given as:

$$q_1 = e^{-(a+b+e)\Delta t}$$

$$q_2 = e^{-(c+f)\Delta t}$$

$$q_3 = e^{-(d+g)\Delta t}$$

and where α, δ and γ are given above in 3.14. Where u_{21}, u_{31}, u_{32} and v are (Equation 3.11) given as

$$u_{21} = \frac{a}{\lambda_1 - \lambda_2}$$

$$u_{31} = \frac{b + cu_{21}}{\lambda_1 - \lambda_3}$$

$$u_{32} = \frac{c}{\lambda_2 - \lambda_3}$$

$$v = \left(u_{31} - u_{21}u_{32}\right)$$

and, finally, λ_i for $i = 1, 2, 3$ are given as in 3.10 :

$$\lambda_1 = -(a+b+e)$$

$$\lambda_2 = -(c+f)$$

$$\lambda_3 = -(d+g)$$

The coefficients ϕ_i and θ_i are determined entirely by the values of the reservoir response parameters ($a, b, \ldots, g$) and automatically satisfy the usual stability conditions required for a deterministic model (that the sign of the eigenvalues is positive) which is akin to the stationarity conditions associated with ARMA models. Although the expressions for the difference equation parameters are quite complicated, they only need to be calculated once and after this do not change, during the computation of a given response. In a true stochastic ARMA model the θ_0 term is equal to one and the inputs r_t are replaced by a white noise process. This difference equation (3.15) provides the response from each cell, with the total catchment response being a linear summation of these outputs; as will be explained in Chapter 4.

3.3.1 Functional Relationship Between Linear Reservoir and ARMA Parameters

The difference equation coefficients are defined in terms of the reservoir response parameters. This functional relationship between the linear reservoir and difference equation parameters ensures that the model produces a sensible pulse response function regardless of the choice of parameters. The reservoir parameters can be chosen entirely independently of each other and the resulting pulse response is unconditionally stable (non-oscillatory) and non-negative.

It is the State-Space formulation that provides these attractive properties, since no outflows can occur from an empty reservoir. In this way, it can be guaranteed that negative flows will not occur. The derived difference equation form (using equation 3.15) of the model corresponds exactly to the State-Space (from equation 3.5 and 3.6) form as shown in Figure 3.2 for an example where $k_1 = 10$, $k_2 = 29$, $k_3 = 13$, $k_4 = 14$, $k_5 = 10^6$, $k_6 = 25$, $k_7 = 10^6$ and the 3 reservoir cell model is fed by a unit volume pulse with a duration of ten time periods.

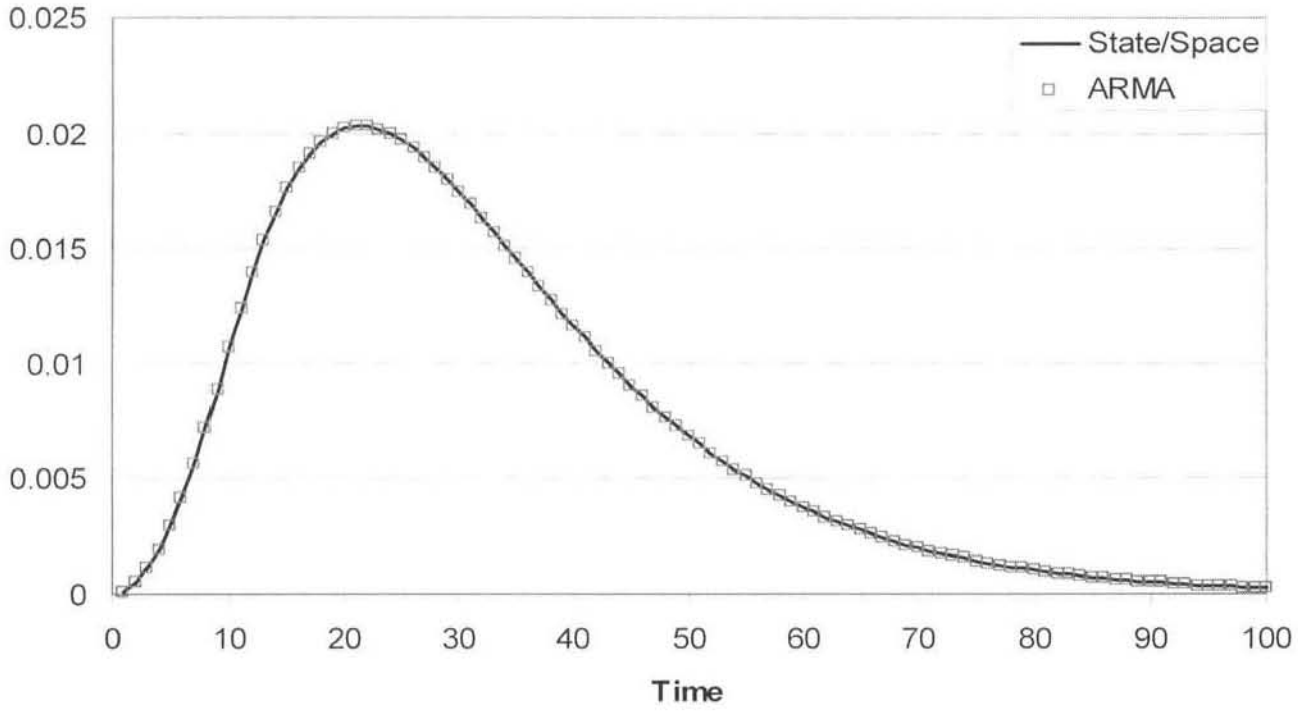


Figure 3.2: Comparisons of a pulse response function for the State-Space and pseudo-ARMA representations of the 3 linear reservoir model

3.4 STABILITY

As discussed earlier the three reservoir model has an unconditionally stable and non-negative pulse response function. However, there are stability problems which occur when trying to fit model parameters using the difference equation form of the model. There are two conditions to gaurantee stability, first the three starting values must be sensible (i.e the model should be able to reproduce them) and second if any of the starting values are zero, then the input at the corresponding timepstep should also be zero.

The condition imposed on the rainfall will be discussed first, as it is easiest to understand. By examining equation 3.15 it is clear that an immediate response to rainfall input occurs when θ_0 is non-zero. It follows that for $y_k = 0$ $r_{k-\tau}$ must also be zero. The starting values should be part of a sequence of outputs produced (or possibly produced) by the model.

The Autoregressive operator requires the three previous values of y_t in order to calculate the current value. In a fitting procedure which attempts to match the model output to an observed flow hydrograph, by adjusting the model parameters, it makes sense to select the initial starting values to match the observed flow response. In so doing we hope to start with the catchment in the correct storage state. The difficulty is that we are representing the catchment with a *model* which by definition is a simplification of the physical catchment processes. The sequence of flows produced by the catchment satisfies the principle of continuity and cannot have resulted from anything but a positive storage in the catchment. The only way for the model to reproduce exactly this sequence of flows may be by allowing a negative starting storage to occur in one, or more, of the reservoirs. A negative storage is a physical impossibility, and the full State-Space equations do not allow such a situation to arise. The pseudo-ARMA formulation of the model is exactly equivalent to the State-Space representation, but the principle of GIGO (Garbage In Garbage Out) applies. If the model is started with a sequence of values which it could not have produced in reality, then the resulting output will very likely be unstable.

Having established that it is the starting values producing the stability problems, the apparent solution is to determine the relationship between successive values produced by the model and simply select starting values which conform to these specifications. Stability can be guaranteed for the difference equation form of a cascade of two unequal reservoirs, such as that derived by O'Connor (1982) or Diskin & Pegram (1987c). Examining the Auto Regressive portion of the difference equation form, and constraining y_t to be non-negative can establish the criterion for stability. The result, given in Appendix A, is that the ratio of successive values must be such that

$$\frac{y_t}{y_{t-1}} > e^{-\frac{1}{k_i}\Delta t}$$

where k_i is the longer of the reservoir response times. By ensuring that the starting values selected from the observed flow record meet this criterion it is possible to guarantee a stable two-reservoir model, regardless of the parameter values chosen.

The three reservoir model needs three starting values for its pseudo-ARMA form. It was not possible to establish a relationship between the three starting values and the values of q_i, as was done for the two reservoir case. Until such a relationship can be determined, the only way to absolutely guarantee stability of the model is by using the State-Space formulation with positive initial storages. This will entail extra computing time during the parameter fitting procedure, as a record of the previous storage in each reservoir must now be maintained. During calibration, therefore, the calculation of the new storage in each reservoir must now take the place of the simple difference equation computations. Nevertheless the pseudo-ARMA formulation will still be used in an operational situation where the parameter set has already been selected. The starting values can be set to zero, the equivalent of a dry catchment. From this starting position the model can be run in forecast mode, accepting the most recently observed rainfall inputs and projecting a possible future sequence of flows.

Figure 3.3 shows the kind of unphysical response that can result from the wrong choice of starting values. The plot in Figure 3.4 shows the response for the same parameter set with a different choice of starting values. Much exploration of the phenomenon reveals that, the negative response appears to occur when the starting values are such that the second derivative, of the output y_t with respect to time, $\frac{d^2y}{dt^2}$ is negative (around a local peak), and the curvature is greater than the model could have produced. If the second derivative is positive (around a local trough), the magnitude of the curvature has not been found to have an effect on the model's stability. This apparently anomalous situation does make physical sense. The relative values of the model response parameters determine the magnitude of the curvature produced once input to the model has stopped. Any additional input occurring at or near the peak response time can only result in a reduction in the curvature. It is therefore essential that starting values chosen near the peak response do not have a curvature greater than the minimum allowable. Inputs occurring on the recession limb of the response will tend to produce a sharp upturn (see the pulse responses in Figure 3.8) since the model has

an immediate response to input, as described earlier. Thus, choosing starting values which are on a steep upturn is less likely to produce a negative response. A simple way to avoid the problem of instability is to select the starting values of model flow (not storages) from a relatively flat portion of the hydrograph, or at the start of the rising limb. In most cases, this criterion has been sufficient to ensure a stable response from the model. However, a more precise definition of this criterion would be preferable. This form is still convenient for event modelling and makes for easy implementation.

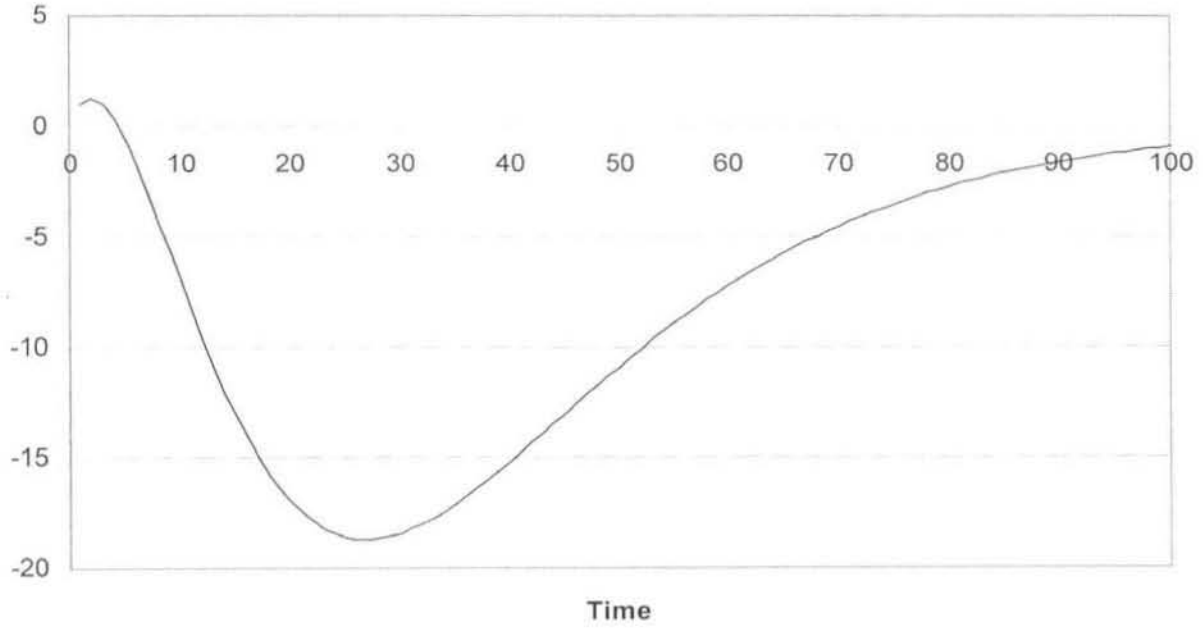


Figure 3.3: A negative response from a three reservoir arrangement

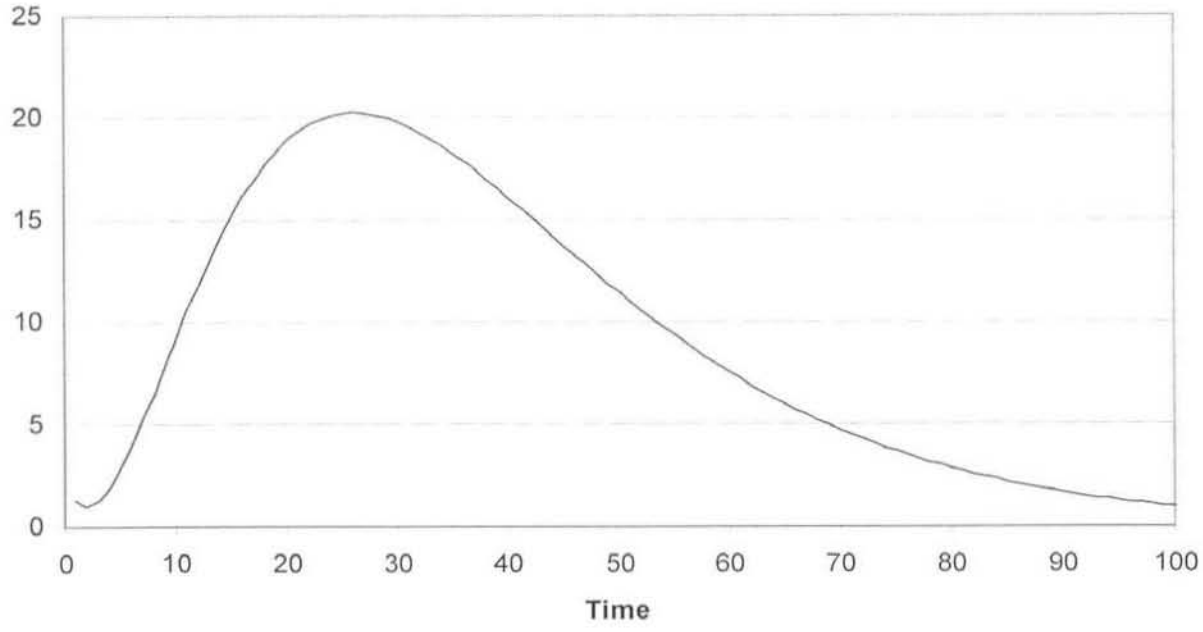


Figure 3.4: A non-negative response from a three reservoir arrangement

3.5 PARTICULAR FORMS

The general form of the three-reservoir model has great flexibility in its application to catchment modelling. A careful choice of parameter values allows specialization of the model to represent an entire family of linear reservoir models. This can be done by selecting extreme (physically unlikely) values for any of the reservoir response parameters k_i. The selection of a very long ($> 10^6$ time periods) response time results in a slow draining of the reservoir, effectively blocking that particular flow path and removing the connection from the formulation. In a similar way the selection of a very quick (< 1 time period) response time can "hide", or completely open, an inter-reservoir connection thus removing its influence from the model. Using suitable combinations of extreme response values, the general three linear reservoir model with loss terms can be "collapsed" to a number of particular forms. These may be a single reservoir, a cascade of two or three reservoirs (equal or unequal), a pair of parallel reservoirs or the three linear reservoir arrangement of Figure 3.1. All of these variations may include a loss term, or combination of loss terms, to suit the modelling requirements. The effect of including loss terms is illustrated in Figure 3.5.

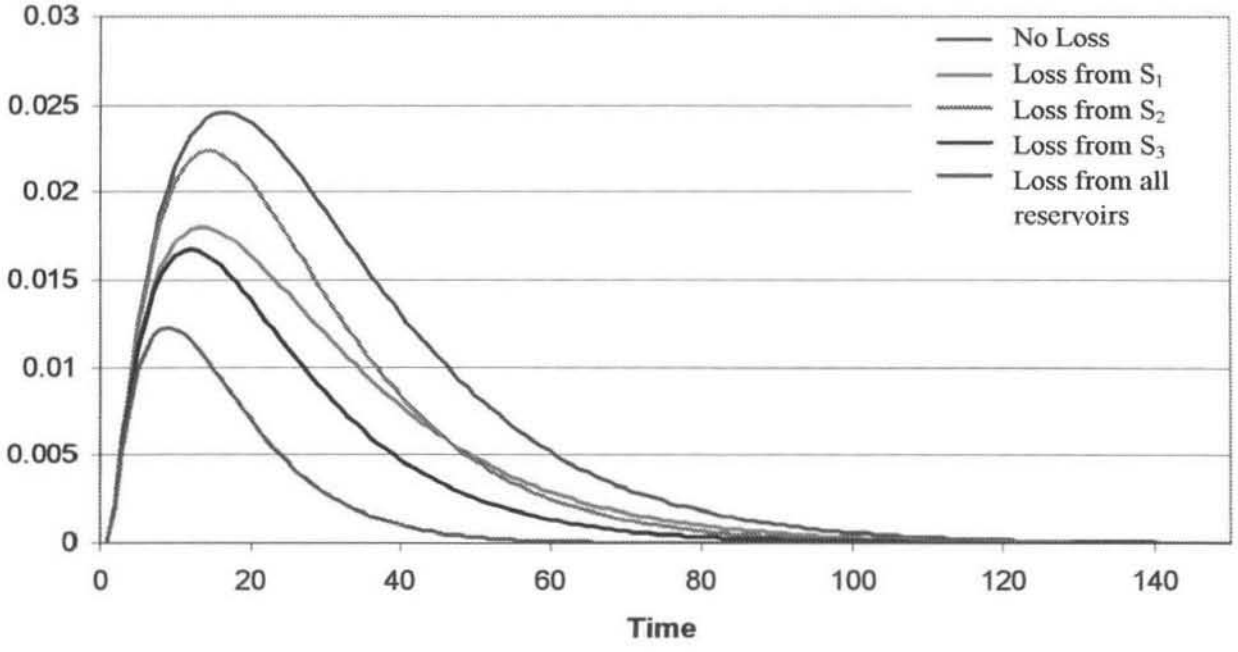


Figure 3.5: Comparison of some pulse response functions for the 3 reservoir model

It is a simple matter to have a separate conceptual arrangement for each of the cells by fixing some parameter values at the cell level. There may be cases where this kind of flexibility is useful. The specification of a single difference equation formulation will simplify the modelling procedure from a computational standpoint. Figures 3.6 and 3.7 show a comparison of the pulse response functions for a single reservoir, a cascade of two reservoirs and the equivalent collapsed forms of the general three reservoir model. The pulse is of a unit volume and has a duration of 10 time periods. Figure 3.8 shows the pseudo-Impulse response functions for the three linear reservoir model collapsed to a single reservoir, two and three reservoir cascade, with equal response times. The reduction of the model to these forms is made by allowing k_2, k_5, k_6 and k_7 to approach infinity, creating a three reservoir cascade. The two reservoir cascade is produced by further allowing k_1 to approach zero. The single reservoir form is reached by further allowing both k_1 and k_3 to approach zero. The pseudo-IRF is the response function for a unit pulse input, with a duration of one time period. Figure 3.8 should be compared to Figure 2.3, which shows the true IRF's calculated using the continuous formulation (Equation 2.3) and the same reservoir response time (k = 15 time periods).

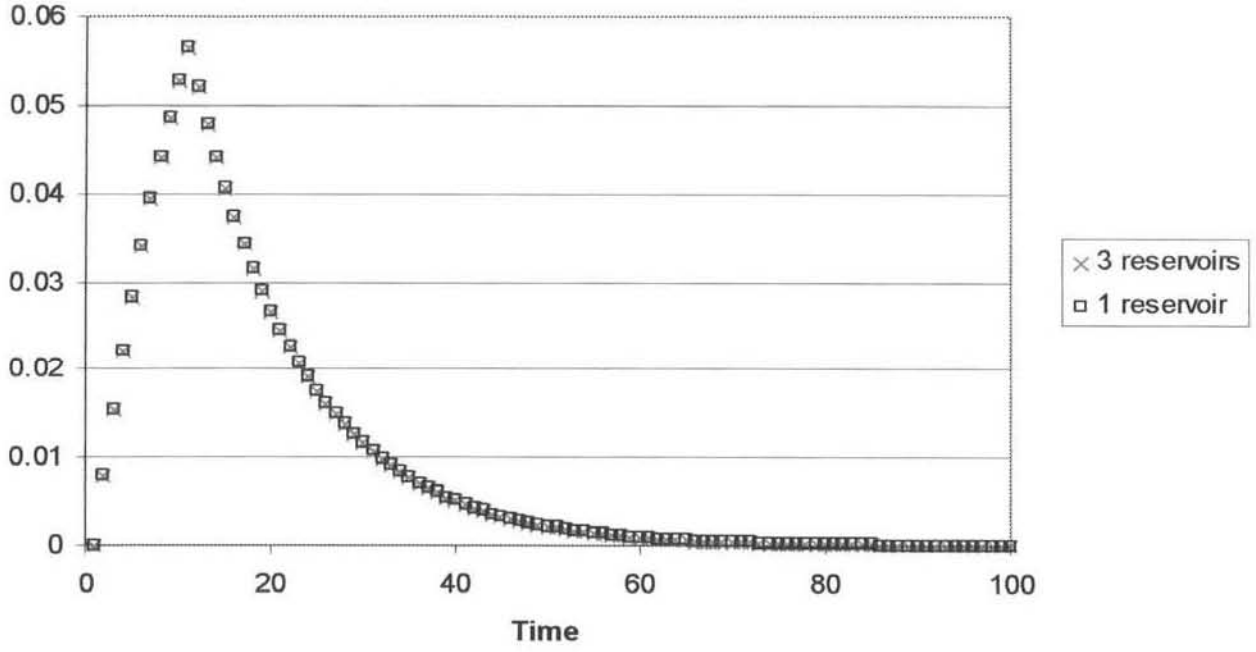


Figure 3.6: Comparison of the pulse response function for a single reservoir and the equivalent collapsed form of the 3-linear reservoir model

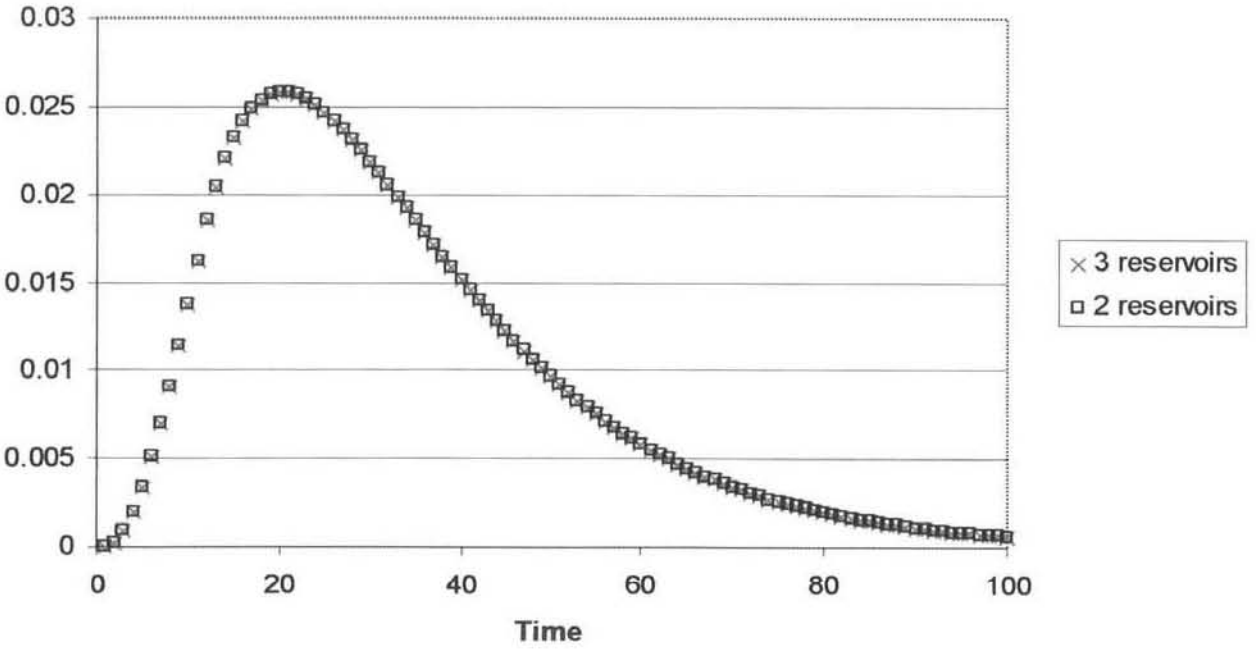


Figure 3.7: Comparison of the pulse response function for a cascade of two unequal reservoirs and the equivalent collapsed form of the 3-linear reservoir model

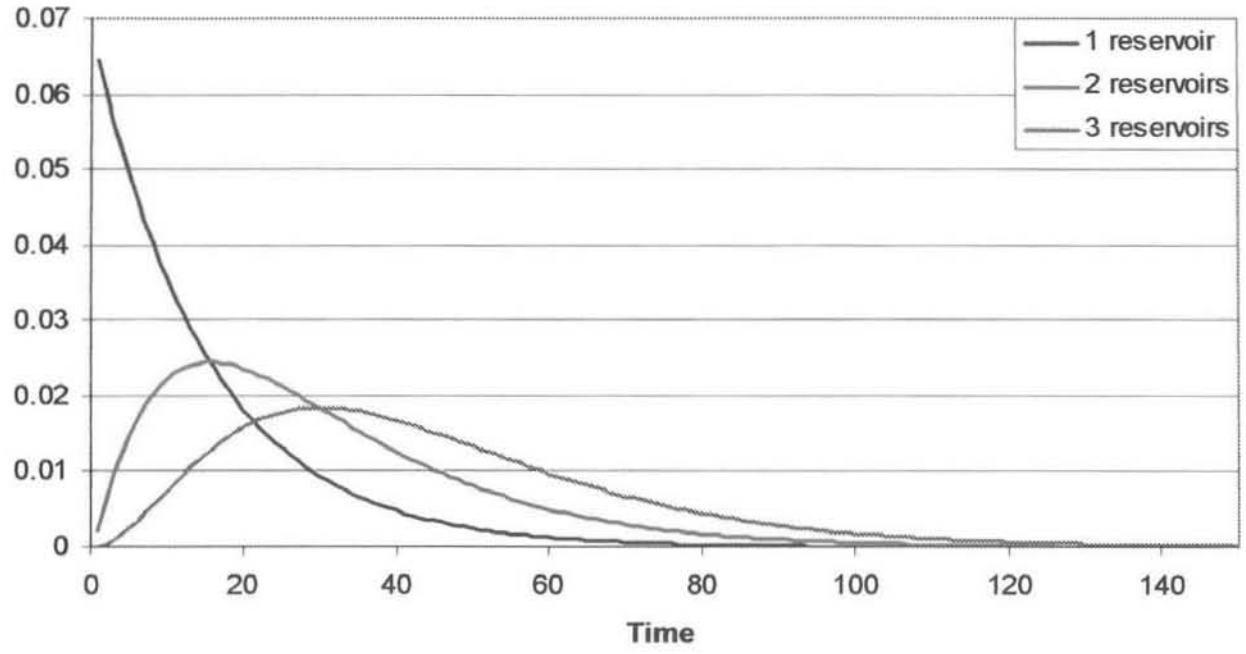


Figure 3.8: Impulse response functions for the 3 linear reservoir model reduced to 1, 2 & 3 reservoir cascades (compare with Figure 2.3, Chapter 2)

3.6 RANDOM NUMBER GENERATORS-AN APPRAISAL

The optimization method chosen relies heavily on random number generation. The generation of a truly random set of numbers with a computer algorithm is a difficult task. A computer is a totally deterministic predictable machine and although it is possible to generate a *random* sequence of numbers using the internal clock, this sequence is not repeatable. However, if a sequence of numbers can be shown to be statistically independent of one another then they may be considered as a random sequence. The optimization techniques employed here require the generation of long sequences of random numbers to be effective. It is therefore essential that a good generator be selected for this task. A short appraisal of their relative merits therefore seems in order.

The usual way to provide random numbers is to generate a sequence of uniform random deviates lying between 0 and m . A linear congruential generator has the following form

$$I_{j+1} = (aI_j + c) \text{mod}(m)$$

Where the I_j's are integer numbers. I_{j+1} is the uniformly distributed random deviate. The multiplier a and incrementer c are positive valued integers chosen (presumably) to produce the longest possible sequence of uniform deviates. By taking the modulus of $(aI_j + c)$ it is

guaranteed that the I_{j+1}'s will be in the range *0* to *m-1* (inclusive), since the modulus operation has only m possible outcomes. The sequence of deviates will therefore eventually repeat itself. The maximum sequence length (period of the generator) is defined by the chosen value of m. The choice of initial I_j (random seed) will determine where in the sequence the generator will start.

Linear congruential generators tend to exhibit serial correlations between successive calls. This manifests itself in banding phenomena when sets of n successive random numbers from a sequence are used to plot points in n dimensional space. Thus the points are not uniformly distributed throughout the parameter space but tend to lie on a number of distinct planes (Press et al., 1992). This is especially problematic when the random deviates are being used in Monte Carlo based applications, which rely on a complete investigation of the entire parameter space.

Park and Miller (1988) suggest that a multiplicative generator of the form

$$I_{j+1} = (aI_j) \bmod (m)$$

is capable of generating random sequences which are as good as those produced by any of the linear congruential types which have values of c greater than zero. The *minimal standard* generator that they suggest has the following values

$$a = 16807 \qquad m = 2147483647$$

The random number generator used in this study (Press et al., 1992) is based on this minimal standard, but uses an additional shuffling algorithm (Knuth, 1981) to get rid of the low order serial correlations which may be associated with the generator. The period of this generator, approximately 2×10^9, is considered more than adequate for the use which is to be made of it, since the number of successive calls to the generator is seldom greater than about 10^6. It should also be noted that the parameter fitting algorithm actually relies on shorter portions of the random sequence in the order of 10^5, or less. The generator produces floating point precision random deviates lying between zero and one. These deviates can then be used to select random numbers within the appropriate bounds by a linear scaling.

3.7 Optimization Techniques

Optimization of the model parameter set refers to the process by which a parameter set is selected to minimize the value of a chosen objective function. In this study, the objective function to be minimized is the sum of squared differences between the observed streamflow hydrograph being analyzed and the model output using observed rainfall as input. The optimization routine chosen to perform this minimization should be capable of making a thorough search of the relevant region of the parameter space, because the complex non-linear rainfall to runoff conversion relationship is likely to have a parameter space containing a number of sub-optimal local minima.

3.7.1 The Simplex Search Routine

The basis of the optimization method used is the downhill simplex search routine (Nelder & Mead, 1965). This routine is a multi-dimensional function minimization algorithm. The search is not directional in the sense of function derivatives, but moves towards the optimal point in the parameter space by finding successively lower values of the objective function. For a function depending on n parameters, an initial n-dimensional simplex is constructed as a starting point for the algorithm. This choice of initial simplex is usually made to coincide with the region of parameter space where a minimum is expected. The objective function is then evaluated at each of the $n+1$ vertices of the simplex. The vertex with the highest value of the objective function is reflected through the opposite face of the simplex and the value of the objective function evaluated at that point. If the value has decreased then a further move is made in the same direction otherwise the initial point is retained. A contraction in one or many dimensions towards the lowest point is also possible. All of the expansions and contractions are made in such a way that the simplex remains non-degenerate (it must enclose a finite volume in n dimensional space). Any move to a point with a lower value of the objective function is accepted while a move resulting in a larger value of the objective function will always be rejected. In this way the simplex tumbles and contracts toward a region where a minimum for the objective function may be found.

3.7.2 Monte Carlo Techniques and The Metropolis Algorithm

The downhill simplex method is most likely to converge to the nearest minimum since it will always accept a downhill step. In situations with many dimensions it is probable that many local minima (at least relative to the machine precision) may occur in the parameter space, making location of the global minimum a difficult task. This problem may be solved through the application of Monte Carlo techniques. A sequence of random starting simplexes is selected which fall within the parameter space. The downhill simplex algorithm is applied to each of these and their respective minima and parameter sets recorded, after a reasonable number of iterations. Provided a sufficient number of starting simplexes are chosen and their distribution through the parameter space is uniform, the global minimum can be taken as occurring at the best position overall.

Simulated annealing using the Metropolis algorithm (Metropolis et al., 1953) makes use of a random walk through the parameter space in an attempt to discover the global minimum. This technique has been used successfully to solve minimization problems involving large numbers of independent variables, such as in the design of electronic systems (Kirkpatrick et al., 1983). The algorithm is based on the analogy of the physical process of annealing where a solidifying substance will arrange its molecules in such a way that the lowest energy state is achieved if the cooling process occurs sufficiently slowly. The state can be thought of as the value of the objective function (E_i) at a given point in the parameter space. The algorithm will move from a point with state E_1 to a point having state E_2 with a probability p.

$$p = e^{\left[-(E_2-E_1)/kT\right]}$$

Where k is the Boltzmann constant and T is the current "temperature". If E_2 is less than E_1 then p will be greater than 1. In this case p is set to 1 and the step is accepted. In this way a favourable (downhill) step is always accepted with an uphill step sometimes being accepted. Different choices of probability distribution may be made for different optimization problems.

The variable T is initially at the maximum expected scale of the problem. The value of this variable is reduced, after a number of steps, using an appropriate scheme. As it is reduced

the likelihood of an uphill step being accepted is reduced. At large values of T the major minima are explored and the algorithm provides an opportunity to escape from large local minima and investigate a greater portion of the parameter space. As the variable is reduced in magnitude the algorithm allows for more focussed investigation of minima. A further advantage of the algorithm is that it may be adapted to allow for the assessment of uncertainty in parameter estimates (Kuczera & Parent, 1998).

3.7.3 The Modified Simplex Search

The minimization routine chosen (AMEBSA) is a modified version of the downhill simplex method that incorporates the Metropolis algorithm (Press et. al., 1992). The value of the objective function associated with each of the simplex vertices is increased by a logarithmically distributed uniform random deviate, proportional in magnitude to the variable T. A similarly proportioned deviate is subtracted from the function value associated with each new point tried by the search process. The Metropolis algorithm is applied indirectly in that a downhill step is always accepted but there is also a possibility of uphill steps being accepted. If the function value at the new simplex vertex is lower than at the old point then the *randomly disturbed* function value at the new point will always be lower than the disturbed function value at the old position and the move accepted. There are two possibilities in the case where the function value at a new simplex point is higher than at the previous position. The disturbed value associated with the new point has the possibility of being either higher or lower than the disturbed value at the old position resulting in the possible acceptance of an uphill move.

The advantage of using this method as an optimization tool is its great flexibility. The routine can be used in either of the following ways depending on the function being minimized. If the value of T is initialized as a large value and the maximum number of iterations allowed is also set to be a large number; then the algorithm can operate in a purely "simulated annealing" mode. The parameter space will be completely investigated with the best local minimum of scale T being selected at each step. After a reduction in the magnitude of T the region within the minimum can then be searched for further minima within a confined region of the parameter space. The process continues in this way until T reaches zero (and the algorithm reduces, exactly, to the standard downhill simplex search)

or the maximum number of iterations allowed is reached. Alternatively if the initial T is set to zero and the maximum number of iterations made small, a large number of restarts can be made from randomly distributed starting simplexes within the parameter space. Each of the restarts performs a "cheap" minimization not requiring much computing effort, however if a sufficient number of restarts are made to provide a complete investigation of the parameter space then the global minimum is likely to be found.

3.8 Description Of The Fitting Algorithm

The parameter fitting was carried out by minimizing the sum of squared differences between an observed streamflow hydrograph and the calculated hydrograph produced by the model. The algorithm adopted was to fit sub-sections of the flow event, one at a time, and to calculate the total sum of squared differences for the entire event. A computer program was written in the C language to implement this procedure. An outline of the main points follows.

The parameter fitting process of minimization varies slightly depending on whether the catchment is being treated as a single cell or as a number of cells. The case where the catchment has been divided into a number of cells will be discussed here; the particular case of a single cell is an obvious specialization of the treatment.

The hourly flow data for the period of investigation are read into memory from a text file. The first *sub-event* is selected from the entire record and stored in a separate memory array. A sub-event is defined as that portion of the hydrograph which lies between two troughs or "local minima" in the hydrograph (Figure 3.9). The ratio between successive flow values is used to determine whether a data point is on the rising limb or the recession. If consecutive values of the ratio change from being greater than one (a recession) to being less than one (a rising limb), a trough is defined. These local minima on the hydrograph mark the beginning and end of each sub-event, except for the first and last sub-events. The start of the first sub-event is determined by the start of the data to be analysed and it's end point by the first trough. The end of the data, similarly, determines the end point of the last sub-event. The sub-event selection algorithm used here is very simple, only deciding if a change in the hydrograph slope is occuring. The algorithm does not distinguish between sub-events of differing magnitudes, this was not deemed necessary for the particular flow sequences

selected for the analysis (see Chapter 5). The inclusion of a more sophisticated algorithm would likely improve the fitting procedure as relatively small (in terms of the peak value) fluctuations in flow would not be unnecessarily analysed as separate sub-events.

The contributing rainfall for each of the cells, over the time period of the sub-event, is selected at the minimum integer lag. The function minimization routine is then implemented for the required range of integer lags. Once a best fit parameter vector has been determined for a sub-event at each of the rainfall lags within the range, a new sub-event is selected and the process repeated.

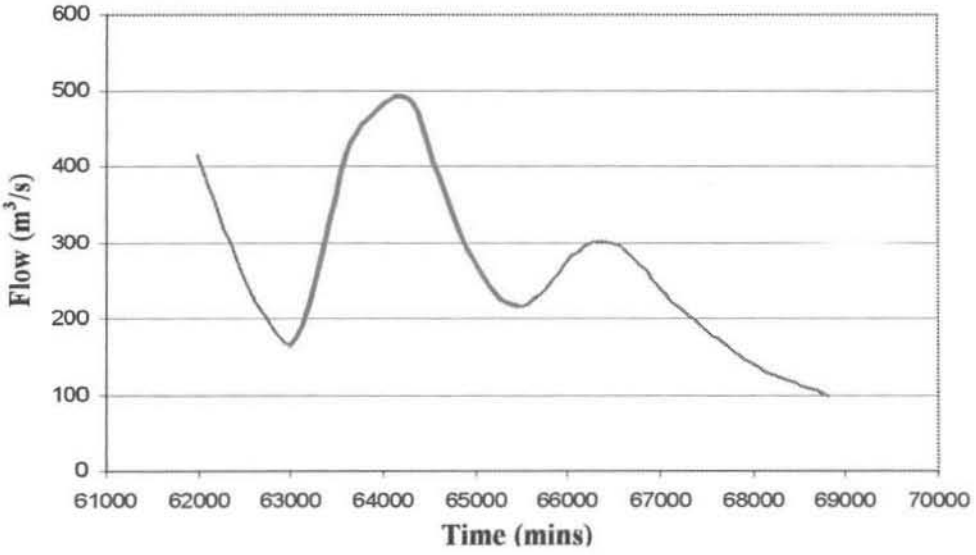


Figure 3.9: A selected sub-event (shown in red)

At each rainfall lag a random starting simplex within the bounded parameter space is selected. This simplex is transformed into an unbounded domain using the logit transform.

For $a \le x \le b$ y is unbounded if $y = \ln\left[\frac{x-a}{b-x}\right]$ (logit transform)

The Modified Simplex Search (MSS) is then carried out in the unbounded domain (in log space), with the parameter values (in normal space) being determined by the reverse Logit transformation at each function evaluation. Each parameter is chosen independently of the others, so there is no need to condition the value of one parameter on another.

If y is unbounded $a \le x \le b$ if $x = \left[\frac{be^{y}+a}{1+e^{y}}\right]$ (reverse logit transform)

The minimum function value and corresponding parameter set is recorded. A number of Monte Carlo restarts are made using randomly selected initial simplexes and the overall minimum function value and parameter set recorded. The evaluations of the objective function proceed by applying the three-reservoir model to each cell and summing their contributions. This will be fully described in Figure 4.10 and the surrounding text (Chapter 4). Two random sequences are used during this process, the first is used to generate the starting simplexes and the second is used by the MSS algorithm to produce the disturbed objective function values. The flowchart in Figure 3.10 shows schematically how the fitting procedure was designed.

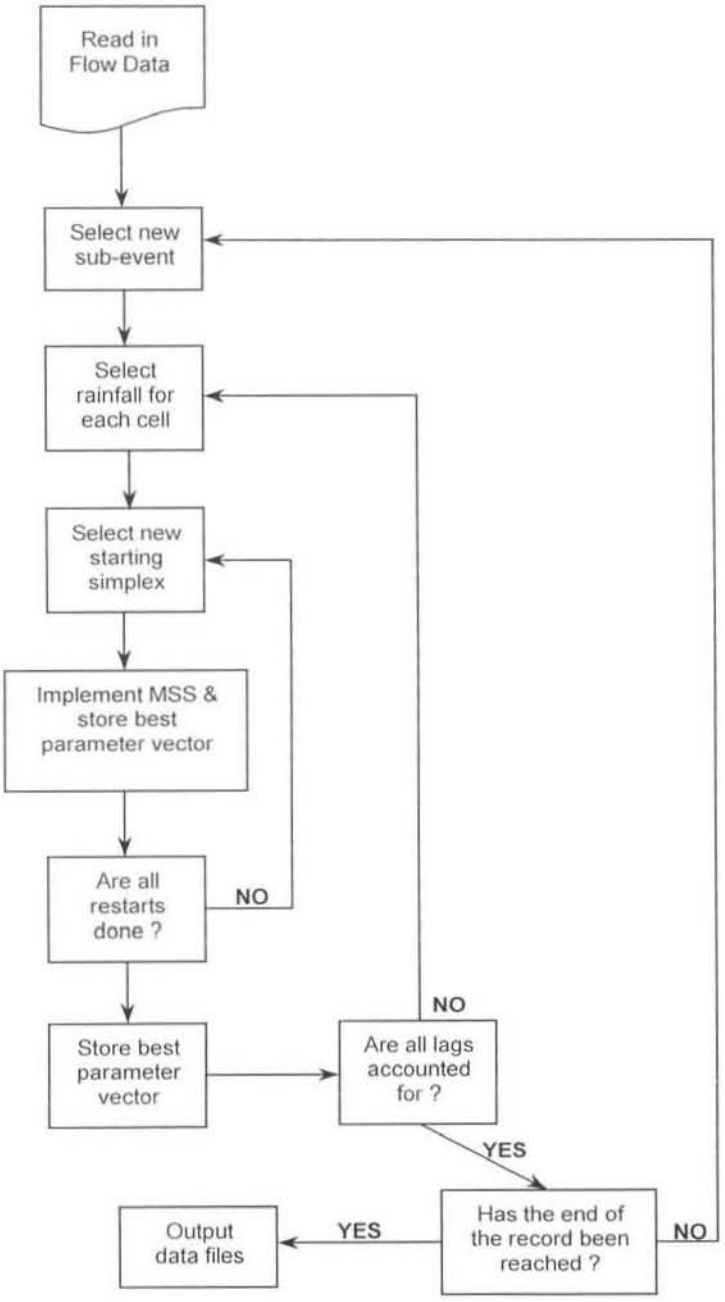


Figure 3.10: Flow chart for the parameter fitting procedure

CHAPTER 3 MODEL DEVELOPMENT AND OPTIMIZATION

Summary:

A general three reservoir feed forward model, with loss terms, has been presented. Discretely co-incident pseudo-ARMA and State-Space representations of the model have been developed following Diskin & Pegram's (1987c) treatment of a cascade of two unequal reservoirs. It has been shown that the general model can be collapsed to a number of particular forms, with or without losses. The great flexibility of this family of sub-models could be of great use in the development of cell models such as those described in Chapter 2. The application of this model to a test catchment is described in Chapter 4.

This Chapter also provided a description of the Monte Carlo and simulated annealing methods used in the model fitting process. The fitting procedure makes use of a modified simplex search routine incorporating the Metropolis algorithm and Monte Carlo restarts to minimize the objective function. The algorithm used to fit the model output to a number of observed rainfall-runoff events has been explained. The results of the fits that were achieved are presented in Chapter 5.

CHAPTER 4 APPLICATION OF THE RESERVOIR MODEL

The practical application of the linear reservoir model introduced in Chapter 2 and developed in Chapter 3 is applied to a well-instrumented catchment. The study was carried out using rainfall and streamflow data for the Liebenbergsvlei test catchment near Bethlehem. Precipitation data were obtained from the South African Weather Bureau's (SAWB) Meteorological Systems and Technology (METSYS) team based at Bethlehem, while flow data from the two streamgauges on the catchment were obtained from the Department of Water Affairs and Forestry (DWAF). A description of the study catchment and the model's application is provided. The issue of spatial distribution in rainfall is also discussed. The catchment was subdivided into a number of cells (sub-catchments) each of which received a separate rainfall input, found using the Image masking technique. Outputs from each cell are linearly summed to produce the total catchment output. A comparison is made between this semi-distributed cell model and the Manifold model of Pegram & Diskin (1987a, b and c)

4.1 LIEBENBERGSVLEI

The Liebenbergsvlei catchment is a subcatchment of the Vaal, situated near Bethlehem in the Free State Province. The portion used in the study covers an area of approximately 3600 km^2 and has a relatively gentle slope. The vegetation is predominantly dry thornveld with high levels of evapotranspiration. The catchment with its quaternary sub-catchment boundaries is shown in Figure 4.1.

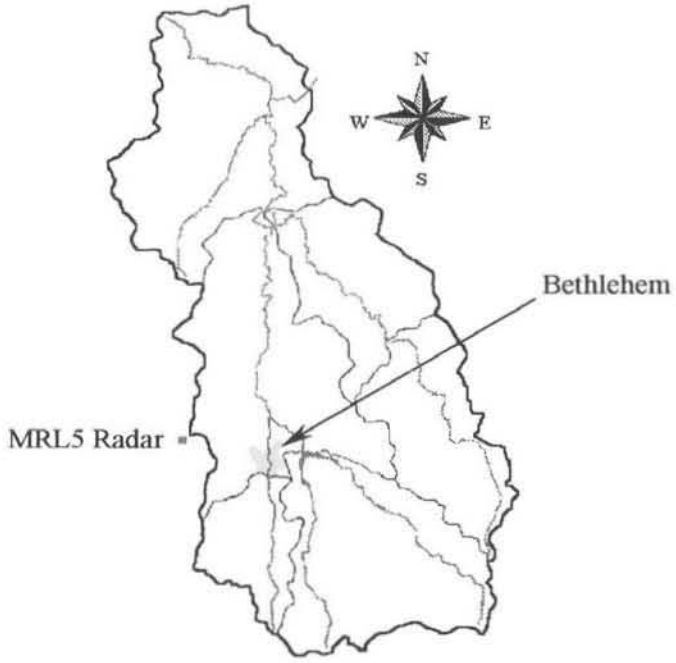


Figure 4.1: The portion of the Liebenbergsvlei catchment used in this study, showing the approximate position of the MRL5 radar.

This catchment is in a fairly dry region of the country with an average annual rainfall total of 650 mm. It receives most of this precipitation during the summer season, which ranges from October to February. The mean annual runoff volume from this portion of the catchment for the twenty-one year period from 1978 up and including 1998 was 126×10^6 m^3. The equivalent average precipitation volume is 2412×10^6 m^3. This suggests that on average there is a loss of more than 95 percent in the conversion from rainfall to runoff.

4.2 RADAR COVERAGE

A mosaic of eleven weather radars provide partial coverage of South Africa. The wetter eastern portion of the country is relatively well covered by the radar mosaic. The data stream from each radar is connected to the METSYS offices at Witbanksfontein and the

data are collected there in real time and archived in MDV (Meteorological Data Volume) format. The radars have an effective radius of operation of approximately 100 km. Outside of this range the accuracy of the data reduces significantly. Rain rates are calculated from the returned power by the METSYS software, using the Marshall-Palmer relationship (Marshall & Palmer, 1948)

$$Z = 200R^{1.6}$$

where R is the rain rate in mm/hr and Z is the returned power in DBz.

Figure 4.2 shows the position and range of the radars, with the study catchment indicated. There is particularly good coverage of the Free State province, Kwa-Zulu Natal, Gauteng, Mpumalanga and the Northern province. These eastern and central regions are where the bulk of the country's population lives.

The SAWB's MRL5 weather radar (Figure 4.3) to the Northwest of Bethlehem provides full coverage of the Liebenbergsvlei catchment, estimating real-time rain rates at a spatial resolution of 1 km^2. The temporal resolution is at, approximately, five-minute intervals. The data from the radar's full volume scans are stored in floating point precision by SAWB in the MDV format. The MDV files may then be converted to Windows bitmap images which provide both a visual and quantitative representation of the data field. In this way current and accurate estimates of precipitation are available which is of prime importance in flash flood prediction (Georgakakos, 1987).

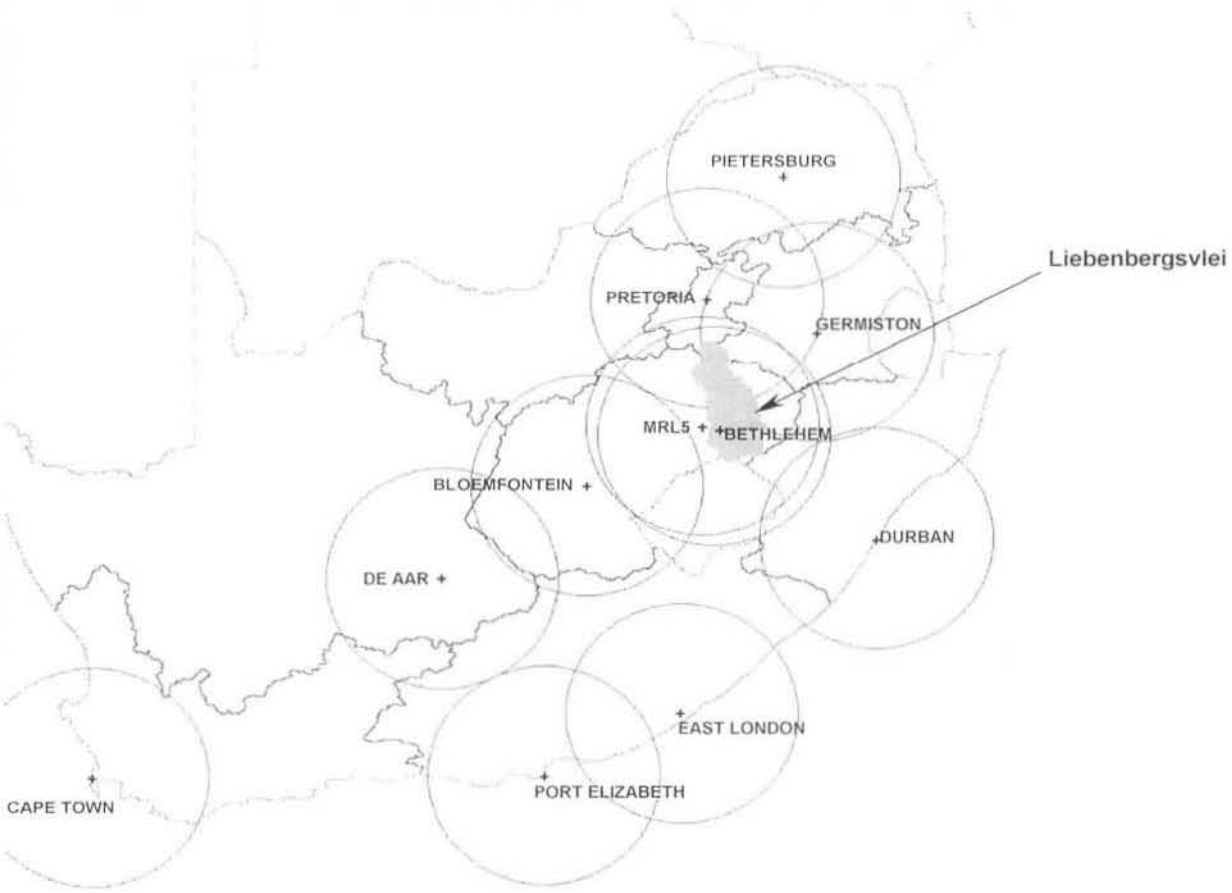


Figure 4.2: Radar coverage of South Africa

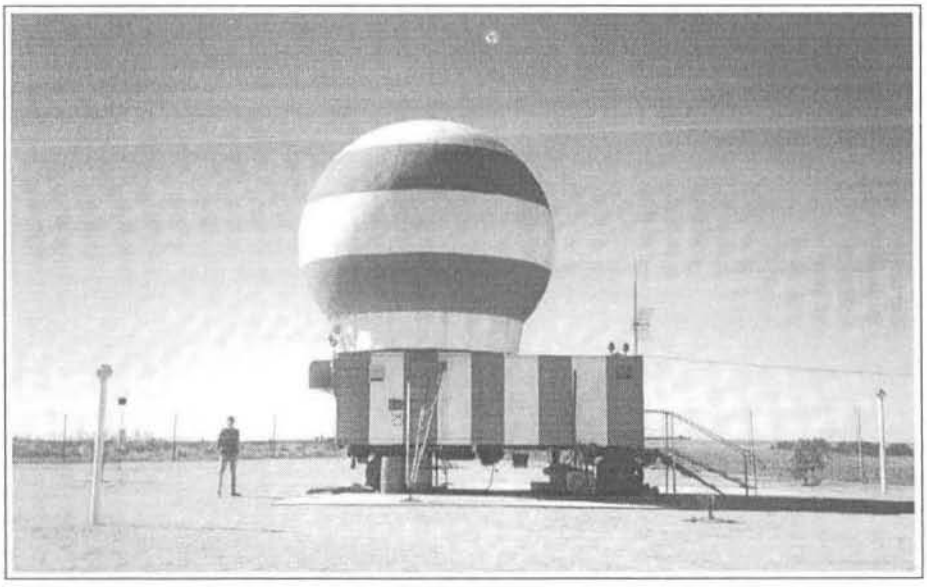


Figure 4.3: The MRL5 radar at Bethlehem

4.2.1 The Bitmap Image Format

The Bitmap image format is designed to store raster image data in a way that is independent of the manner in which specific hardware devices specify colour schemes. The file is formatted in the following way: *Bitmap header->Bitmap information->Image Data*. The file header contains an indicator that the file is in fact a bitmap as well as the size of the file and the size of the image data. The information section of the file contains details of the image size (in pixels), the image resolution (pixels/metre), the number of bits per pixel (used to determine the size of the colour palette) and the colour palette information; for more detail refer to Pegram & Clothier (1999). The image data is stored as a two dimensional array. Pixel values are stored row-wise from left to right and the rows are stored from bottom to top. This means that the origin of the data is at the bottom left hand corner of the image. When the image is viewed the colour represented by the value at each pixel is displayed in the appropriate position.

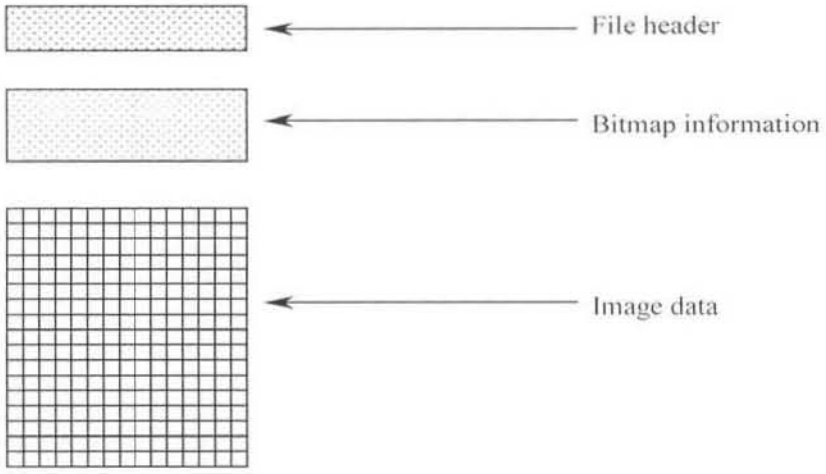


Figure 4.4: Schematic representation of the bitmap file format

4.2.2 Data Representation

The MDV format radar data were viewed and processed in the form of bitmap images. The instantaneous rain rate at a height of two kilometres above the level of the radar is represented by the colour on each pixel of the image. Each CAPPI (Constant Altitude Plan Position Indicator) bitmap has a resolution at which 1 pixel represents 1 km^2. The chosen 256-colour palette uses a logarithmically distributed colour sequence to display an integer

range of rainfall rates between 0 and 100 mm/hr. Figure 4.5 shows an example of a CAPPI displayed using this colour palette and the rain rates indicated by the pixel values. This image shows many convective rainfall clusters with a number of very intense storm cells (indicated by the red and black areas). The ground clutter in the lower right hand corner of the image is caused by the Maluti mountains. This image was recorded at 14:22 on the 5th of October 1998.

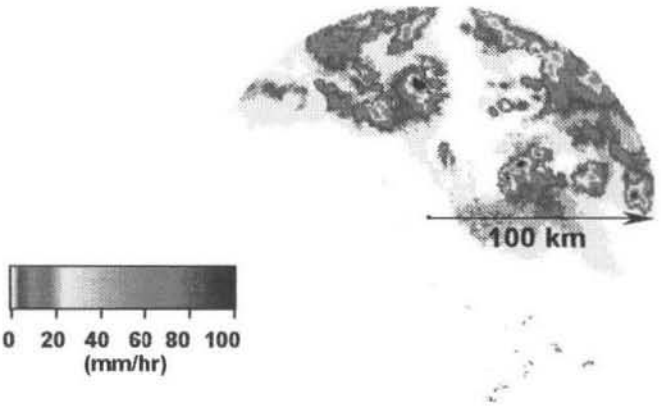


Figure 4.5: A CAPPI bitmap showing the colour palette and associated rain rates

4.3 RAINGAUGE NETWORK

The Liebenbergsvlei catchment is covered by a nearly rectangular network of 45 tipping bucket raingauges, of which 36 are situated on the portion of the catchment under study (Figure 4.6). The gauges are on a grid spacing of roughly 10 km by 10 km. A data logger is attached to each gauge and records the date, time, event count and various flags for each tip event. The gauges record 0.2 mm of rain per tip which means that the loggers memory can record 510 mm of rain since there is enough storage for 2550 events (Terblanche et. al, 1997).

The data obtained from METSYS contained a record of the tips, as recorded by the data loggers, for each of the gauges on the catchment. A computer program was written which searches through these records and accumulates the hourly rainfall depth for each gauge. Hours not experiencing any tips were filled with zero records.

The raingauge network is considered suitably regular in nature to allow for a simple mean of the gauges to be used as the average rainfall field; rather than using an areally weighted average, such as that obtained from Thiessen polygons, or more sophisticated interpolation algorithms making use of techniques such as multiquadrics to fit the optimal surface to the data. This is because these methods would have given nearly equal weight to each gauge with much more effort. The average rain depth was therefore taken to be the mean of the depths recorded by each gauge for any given time period.

Figure 4.6: Approximate positioning of the Liebenbergsvlei rain-gauge network on the six Quaternary sub-catchments.

4.4 Streamflow Measurement and Losses

Streamflow data for the catchment under study were obtained from DWAF. A streamgauging weir is situated at the catchment outlet and the records are available as breakpoint streamflow data. The data were not recorded at regular discrete intervals, making it difficult to use. A computer program was written to calculate the streamflow at regular intervals. A linear interpolation was used between the data points and the entire streamflow record was treated in this way.

For the cell model developed in Chapter 3 losses are defined as that proportion of observed rainfall which is not converted to streamflow at the catchment outlet. The magnitude of these losses from this catchment were calculated so that some idea of the loss proportion could be obtained. Losses occurring on an event were calculated for a number of events by comparing, volumetrically, the rainfall input to and the runoff from the catchment. The loss

proportions calculated in this way were in the range of 70 – 95 %, this means that most of the precipitation occurring on the catchment is lost before reaching the gauging station. For this reason, it was decided that a rainfall to runoff conversion model for this catchment would need a good method of removing losses to cope with this situation.

4.5 Sub-Catchment Division

The Liebenbergsvlei catchment was treated in one of two ways during the model fitting process, it was either lumped and treated as a single cell or it was divided into six cells using the quaternary sub-catchment boundaries (Midgley et al., 1994) shown in Figure 4.7. This sub-division was chosen in order to account for the spatial variability in rainfall and to provide versatility in the catchment model constrained by the topography and topology. The quaternary subcatchments were used since they are defined by their watershed boundaries and part of the precipitation occurring within each area must eventually contribute to streamflow at its channel outlet. Thus, the contributions from each quaternary catchment at the catchment outlet could be linearly summed to produce the total catchment response to a complex rainfield.

4.5.1 Spatial Distribution of Rainfall

From preliminary investigations and observation of the radar image sequences, it is clear that variation in the spatial distribution of rainfall could, in some cases, be an important factor on this catchment due to its long narrow shape. The cusum (double mass) plots in Figure 4.8 illustrate this point. What these plots show is the hourly-accumulated rainfall totals over a month for a chosen gauge on the catchment plotted against the same accumulation for the rest. Gauges 6 and 30 were selected; their locations are shown in Figure 4.7. If rainfall were to occur uniformly across the entire catchment there would be a direct linear relationship between the total and individual accumulations. The plot for gauge 6 initially shows no increase in the rainfall collected by the gauge while the total depth for the catchment is increasing significantly. The plot for gauge 30 does not exhibit this trend at this time, but shows a linear increase in depth in concert with the increase in total rainfall over the catchment. Later in the month this behaviour is reversed with gauge 30 showing no increase while the accumulations for gauge 6 and the catchment as a whole (from 2300mm

to 3000mm) are seen to increase. The double mass curve for each of these gauges is typical of other gauges in its vicinity. These observations clearly indicate the spatial variation in rainfall over the catchment, since the two gauges are far apart and nearby gauges show similar double mass plots. The reason for choosing a semi-distributed approach (by compartmentalizing the catchment) is to capture the essence of this spatial variation.

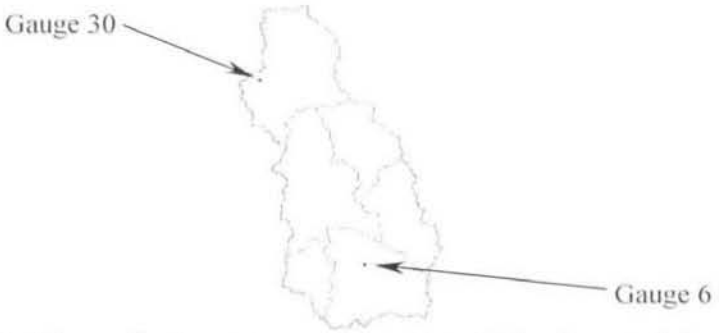


Figure 4.7: Position of relevant gauges on the Liebenbergsvlei catchment

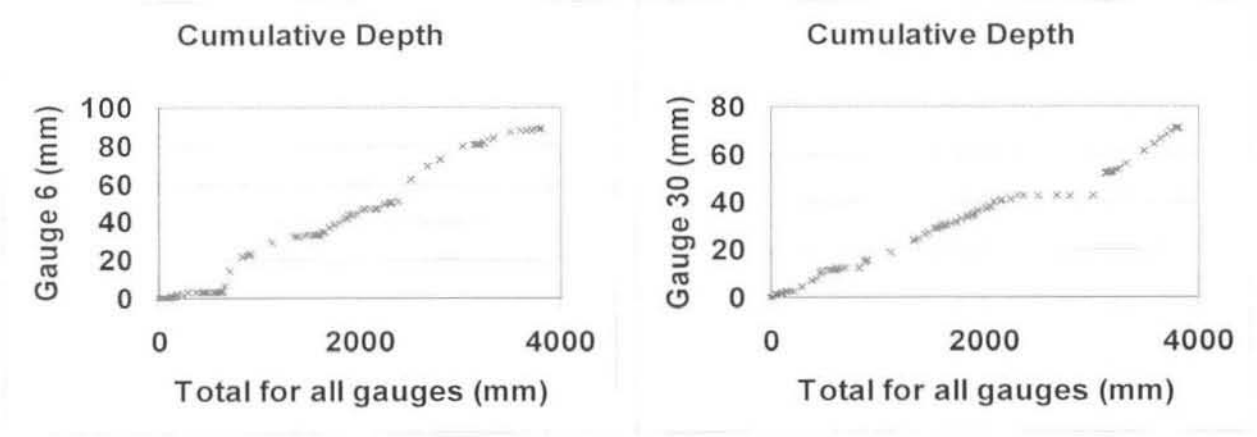


Figure 4.8: Gauge accumulations for the month of October 1995

4.5.2 IMAGE MASKING

In order to determine the average rainfall occurring over each of the six quaternary subcatchment areas a computer program was written which could extract the relevant data from either the raingauge or radar data sets. The method followed was to use an appropriate template to "mask" the CAPPI images. Hourly accumulations of rainfall based on the rain rates occurring only over a particular sub-catchment area were then computed, allowing the spatial distribution of rainfall occurring over the catchment to be accounted for. The average hourly depth occurring over that area was calculated and output to a suitably

formatted text file. A more detailed description of the programming logic is contained in Appendix B of this document.

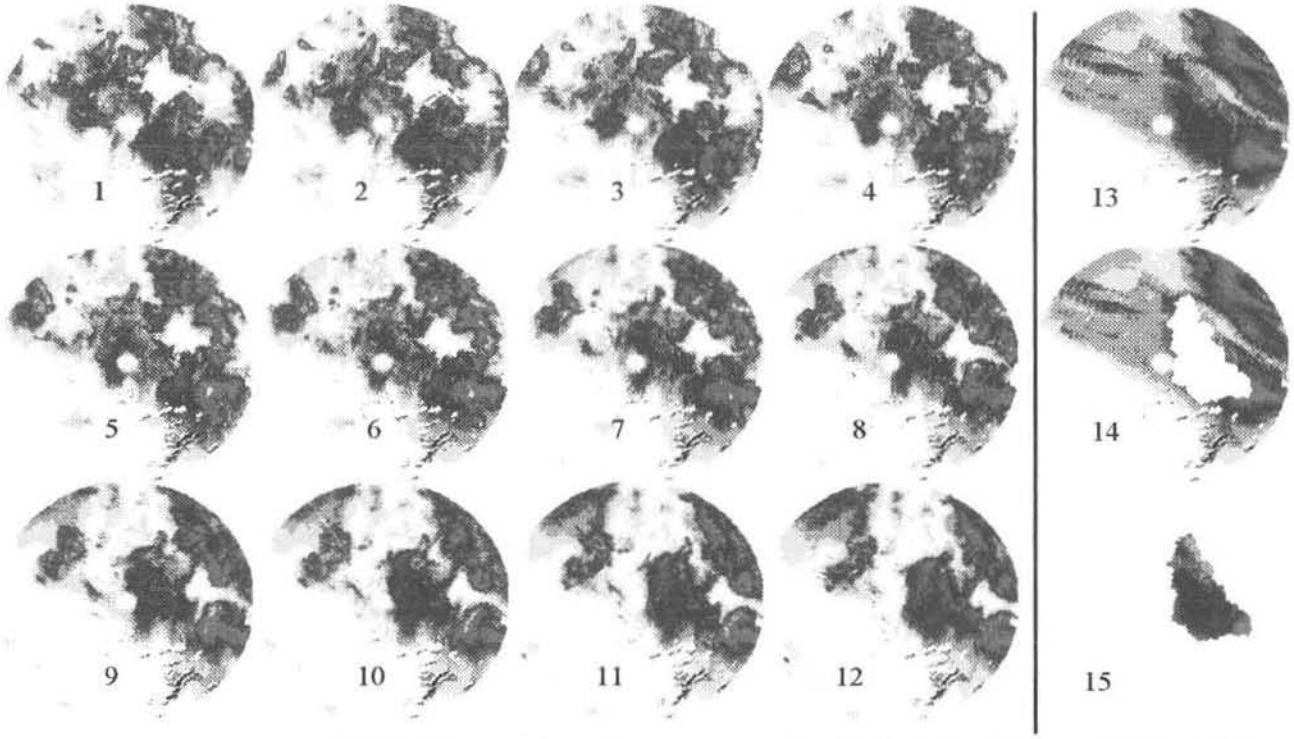


Figure 4.8: Example of CAPPI masking, the sequence 1 – 12 in intensities and 13 – 15 in depths (all scaled in mm/hr)

Figure 4.8 shows a numbered sequence of twelve CAPPI images, which span a period of one hour. Image 13 shows the accumulated depth of rainfall during the one hour period in millimetres. All the colours on the images are scaled in millimetres per hour. The remaining two images (14 & 15) are, respectively, the accumulated image with a masked region removed and the rainfall depths occurring over the masked area only. It is important to distinguish between the CAPPI's on the left, which indicate the instantaneous rain rate (in mm/h), and the images on the right showing the integration of the CAPPI data over an hour, the average rain rate from each of the pixels in the 12 instantaneous images accumulated as a depth.

All the CAPPI data were treated in this way. In cases where there were less than twelve images in an hour the best estimate of the accumulated depth was made with whatever data were available. This entire process was automated so that long runs of images could be accumulated and masked over each of the six quaternary sub-catchment masks.

The masks themselves were defined as bitmap images. The quaternary catchment boundaries were obtained from GIS data available as part of a WRC report (Midgley et al., 1994). A high-resolution bitmap image was produced from the vector based GIS data using the *Arcview* software package. The various sub-catchments were then isolated and saved as separate images, each of which was scaled to the same (pixel: distance) resolution as the CAPPI's. Each mask image was finally processed to position it correctly on a 200x200 pixel resolution image. This was done to facilitate the masking procedure because the CAPPI's were centered on an image of this size at a resolution of 1 pixel: 1 km^2.

4.6 SUPERPOSITION OF CELL OUTPUTS

Each quaternary subcatchment is treated as a single cell modelled by a three reservoir model as specified in Chapter 3. Each of the cells receives separate rainfall input sequences as determined by the masking procedure. The output from each of the cells is calculated using a single set of general catchment parameters which are scaled for each cell according to its physical characteristics (Diskin & Pegram, 1987c). These characteristics are an areal parameter and a channel parameter. The areal parameter is simply the area of the cell as a proportion of the total catchment area. The channel parameter is calculated as the ratio CL/TL, where CL is the distance along the longest collector of a particular cell to the stream-gauging station and TL is the longest of the distances calculated in this way. Physically the areal parameter allows scaling of the cells contribution to total streamflow since the sub-catchment area will be significant in determining the volume of rainfall input received and hence the runoff volume. The channel parameter allows a representation of the travel time associated with each cell. The cell with the longest travel distance along the channel will obviously have the ratio (CL/TL) equal to unity and the other cells will have ratios (and relative travel times) lower than this maximum travel time.

As an example of how this parameter scaling was used; consider a form of the model where a single loss term is included and the removal made from the sub-surface linear reservoir (Figure 4.9). The delay parameters (maintaining the notation of Figure 3.1, Chapter 3) associated with each path are shown; these are scaled using the cell scaling parameters described above. The parameter k_1 is scaled by the areal parameter; this is because it is considered to represent the groundwater recharge, which is largely proportional to the

surface area through which this infiltration can occur. The parameter k_2 is scaled by the square root of the areal parameter, which can be thought to represent the average distance of overland flow to the channel. Similarly, k_3 is scaled by the square root of the areal parameter since it can be conceptualized as representing the distance to the channel via groundwater seepage. The channel scaling parameter scales the channel flow parameter k_4, as a longer channel should result in a longer response time. The loss parameter k_6 is scaled by the areal parameter because the evapotranspiration losses can be considered to be proportional to the area of the cell under consideration. These scalings help to match the model to the gross physical features of the catchment, maintaining relative parsimony.

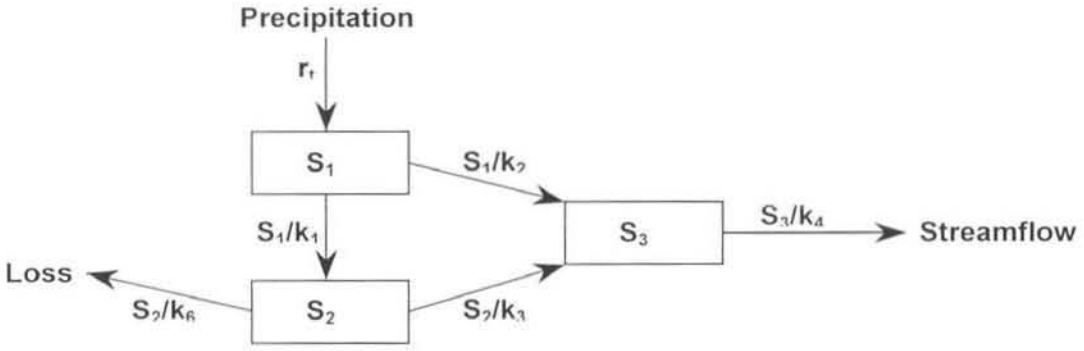


Figure 4.9: Schematic of a 3 linear reservoir feed forward linear catchment model

The streamflow contribution of each cell, *at the catchment outlet*, is linearly combined with the others to produce the total streamflow from the catchment. Figure 4.10 shows an example of this process. Each cell's position is shown on the CAPPI and the contributions from cells C83A – C83F (each modelled as three linear reservoirs with a loss term as in Figure 4.9) are also shown. The combined streamflow at any time is simply the sum of the contributions from each cell.

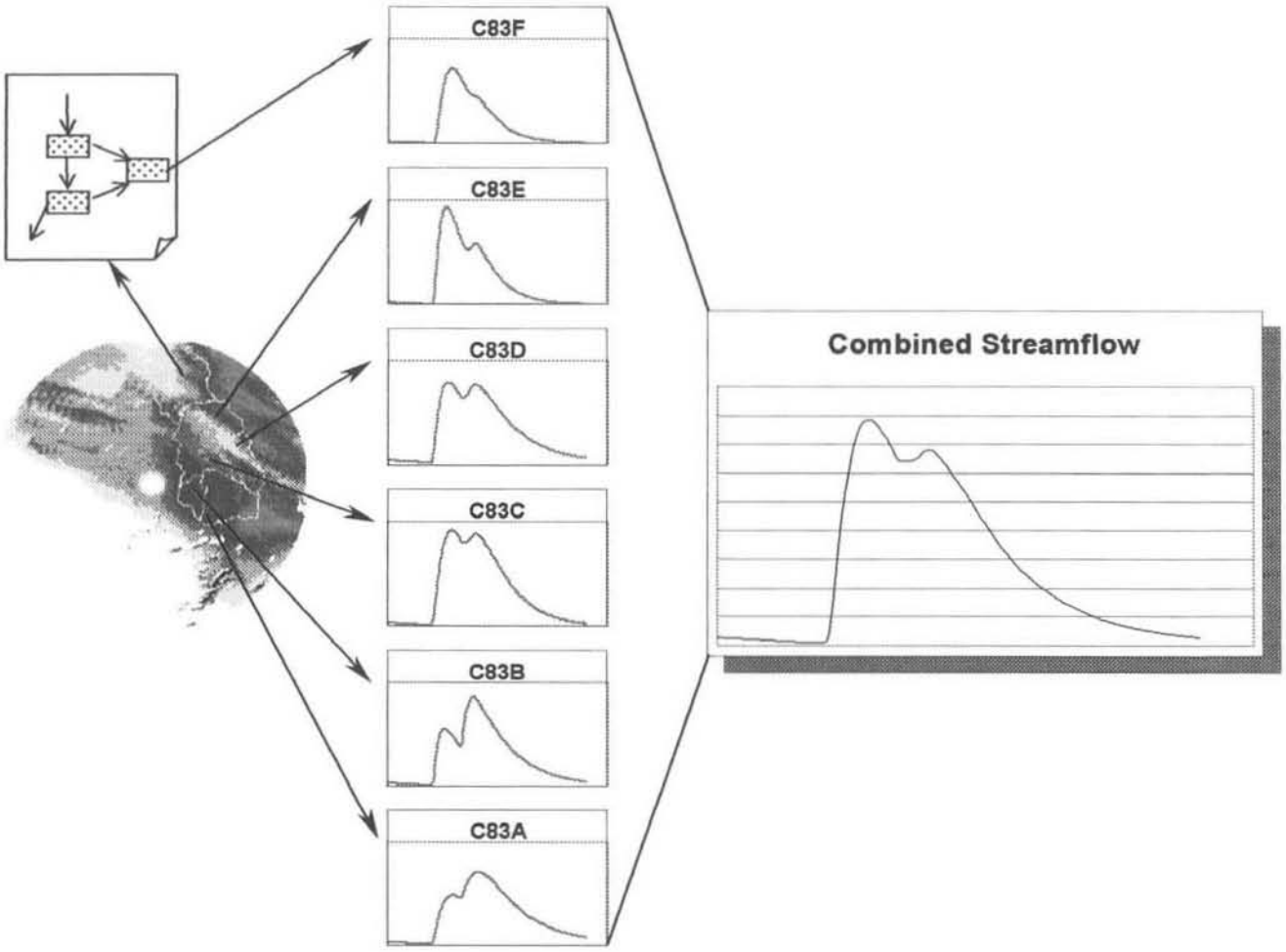


Figure 4.10: Linear combination of cell outputs

The similarities between the structure of this model and the Manifold cell model presented by Pegram & Diskin (1987a and c) should be noted. The Manifold model routes the distributed inputs to each cell through two reservoirs in series; the first has a response time which is assumed constant throughout the catchment irrespective of the size of the cell while the second has a response time which is proportional to the distance of the cell from the catchment outlet. These reservoirs can be conceived of as a surface runoff component and a channel element respectively. The model presented here is similar in the respect that it is also a feed forward model in which the input is routed by a channel element before being summed at the catchment outlet. The response time of the channel element is proportioned according to the distance along the channel from the catchment outlet. The differences are the third reservoir, representing a sub-surface flow response, and the loss components, which allow for the observed precipitation (rather than the computed effective precipitation) to be used as input to the model.

Further differences occur in the manner in which the various reservoir response components are treated for each cell. The manifold model treats the surface runoff parameter as a constant for each cell through the catchment, and scales the input to the cell according to its relative area. This is not the case for the model presented here, as discussed earlier.

Summary:

The Liebenbergsvlei catchment has been described and has been used as a study catchment for the application of the linear reservoir model introduced in Chapter 3. The application of the linear reservoir cell model to the Liebenbergsvlei catchment has been discussed in detail. This Chapter also provided a description of the spatial masking technique applied to the CAPPI radar data. By masking the images an average rain rate over each cell is computed, allowing the model to account for the spatial distribution of rainfall discussed in Section 4.5.1. The catchment was sub-divided into six cells, each of which was modelled as an arrangement of three linear reservoirs (Chapter 3). A single set of catchment parameters is used. These parameters are scaled according to the gross physical characteristics of a particular cell, and the response from the cell (at the catchment outlet) calculated. The flows from each cell are summed to produce the total flow from the catchment (Figure 4.10 and surrounding text). The semi-distributed structure of the model presented here is similar to the Manifold cell model described in Chapter 2 (Pegram & Diskin, 1987a).

This chapter presents the results of the parameter fitting algorithm described in Chapter 3, when used to fit the three reservoir model to both synthetically derived and historically observed input/output sequences. The manufactured sequences were used as a check on the usefulness and accuracy of the fitting procedure. The model was then fitted to three observed flood events on the Liebenbergsvlei catchment (see Chapter 4 for a description of the catchment). Some conditional forecasts are also presented. A discussion on validation of the fitted model parameters concludes the Chapter.

5.1 TESTING THE PARAMETER FITTING ALGORITHM

Before embarking on an attempt to fit the general model to historical data from the study catchment, it was necessary to test the parameter fitting algorithm. This was done by generating artificial streamflow responses from the model in various configurations with known parameter values. The optimization routine described in Chapter 3 was then applied to the manufactured output sequences to recover the "unseen" parameters. A high degree of success is to be expected in recovering the parameter values, since the model is able to exactly match the response of the system which produced the output (this is not the case for rainfall and runoff observations on a real catchment).

The artificial rainfall input sequence shown in Figure 5.2 was used to produce flow sequences (100 hours in length) from the model with fixed parameter values. There was no lag applied to the input sequence. The first model configuration tested was the collapsed form representing a single linear reservoir. The parameter values used are given in Table 5.1. The fit was carried out allowing k_4 to be free and fixing the remaining parameter values at the values indicated in Table 5.1. Figure 5.1 shows the model configuration schematically.

Parameter	k_1	k_2	k_3	k_4	k_5	k_6	k_7
Value (hrs)	10^6	10^{-3}	10^6	10	10^6	10^6	10^6

Table 5.1: Parameter values for single reservoir representation

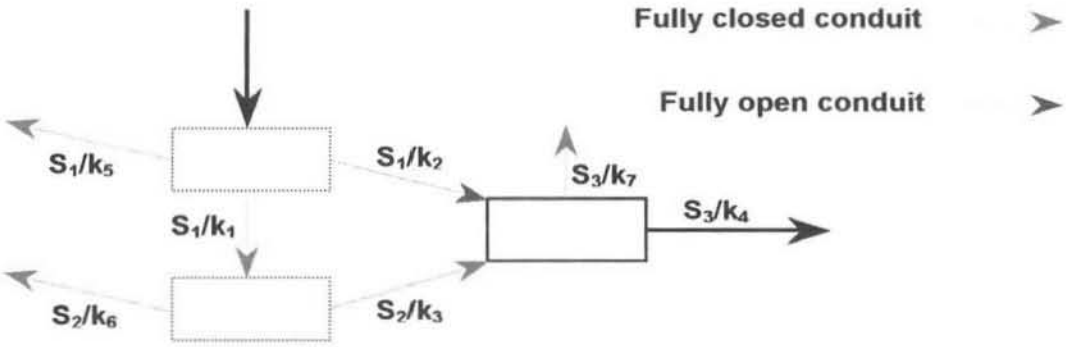


Figure 5.1: Schematic of "single reservoir" model configuration

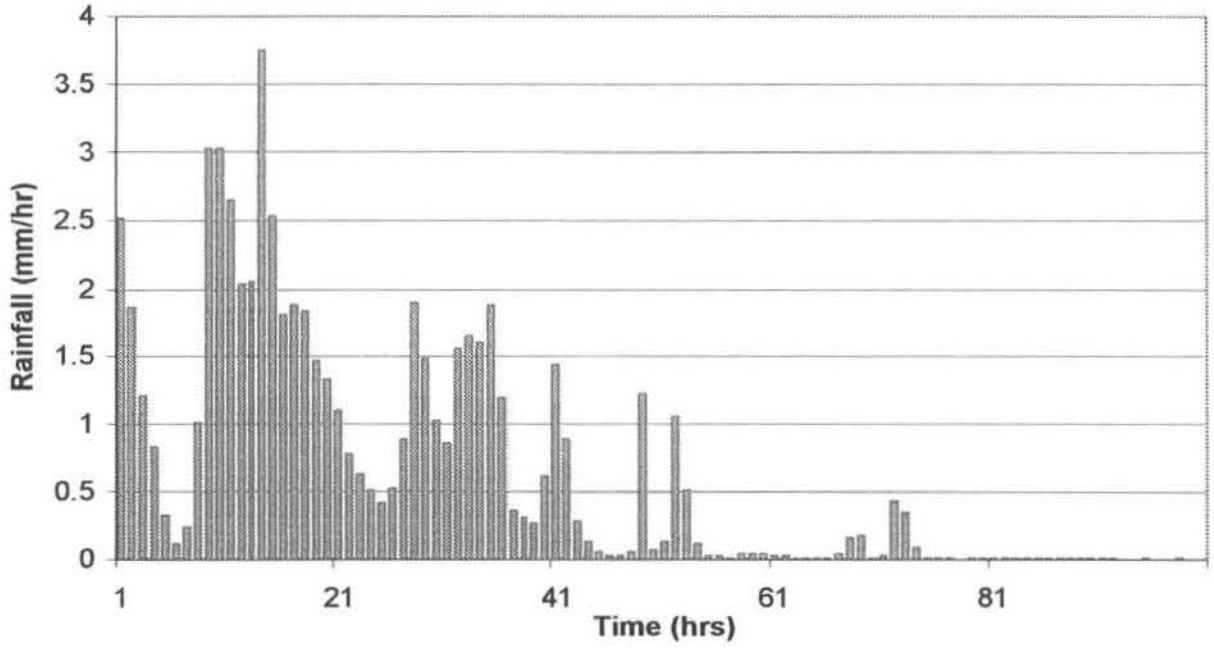


Figure 5.2: Artificial rainfall input sequence

The resulting streamflow sequence and the best fitted estimate thereof are shown in Figure 5.3. The scatter plot (Figure 5.4) indicates the areas of the parameter space where the minimum objective function value was found at each Monte Carlo iteration. Five thousand Monte Carlo iterations were carried out from random starting points in the parameter space (each parameter in the range 0.1 – 10000 hours) with 500 iterations of the AMEBSA algorithm (Press et al., 1992) from each starting point. The processing time was approximately 25 seconds on a Pentium 350 MHz machine.

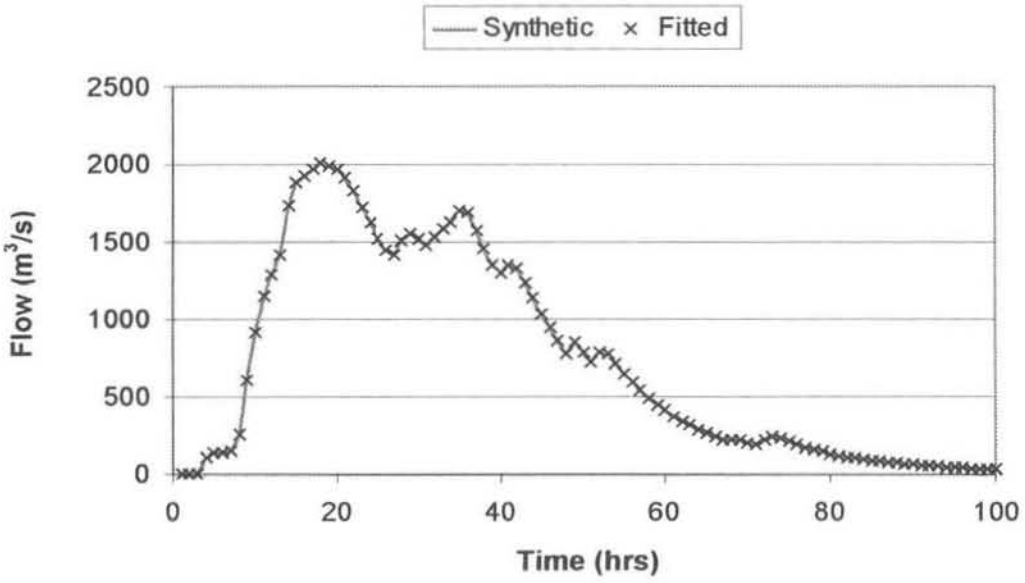


Figure 5.3: Streamflow responses for a single reservoir representation

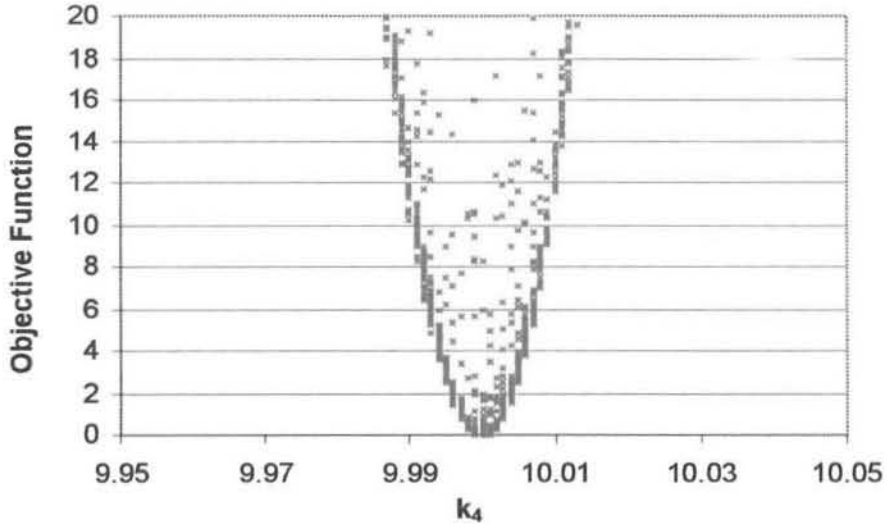


Figure 5.4: Plot showing the optimum parameter value of k_4 at each Monte Carlo restart

Figure 5.4 clearly shows the global minimum to be in the near vicinity of 10 hours, the value used to generate the sequence of flows.

The artificial rainfall sequence shown in Figure 5.2 was used to generate flow sequences from representations of the model equivalent to cascades of two (Figure 5.5) and three (Figure 5.6) reservoirs, respectively. Table 5.2 gives the chosen values of each parameter.

Parameter	k_1	k_2	k_3	k_4	k_5	k_6	k_7
2 reservoir cascade	10^6	15	10^6	10	10^6	10^6	10^6
3 reservoir cascade	14	10^6	12	10	10^6	10^6	10^6

Table 5.2: Parameter values for 2 and 3 reservoir cascades

For each of these representations 5000 Monte Carlo iterations were carried out from random starting points in the parameter space with 500 iterations of the amebsa algorithm (Press et al., 1992) from each starting point. The processing time for the 2 reservoir cascade was approximately one minute on a Pentium 350 MHz machine. The time to fit the 3 reservoir cascade was around five minutes.

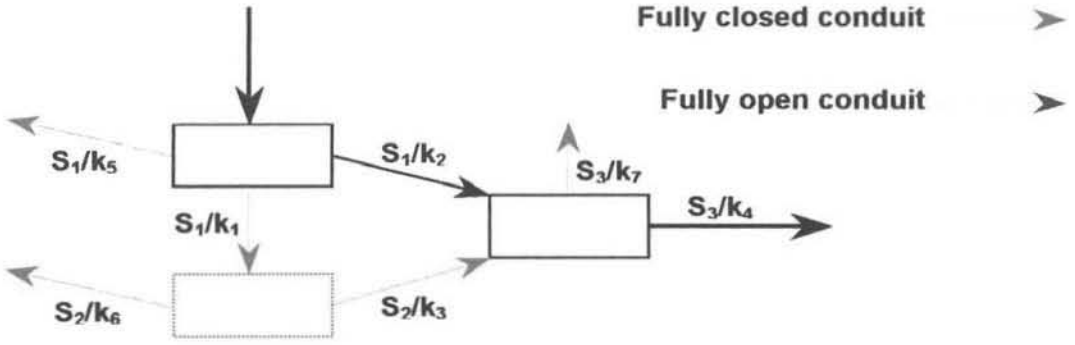


Figure 5.5: A "two reservoir cascade" configuration of the model

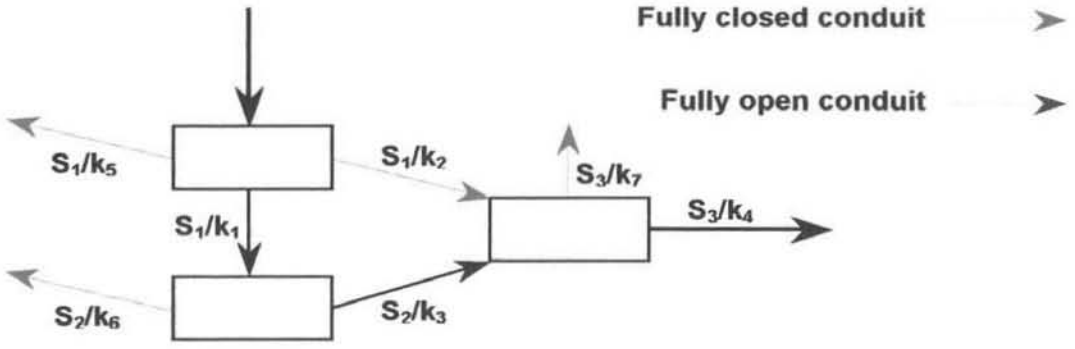


Figure 5.6: A "three reservoir cascade" configuration of the model

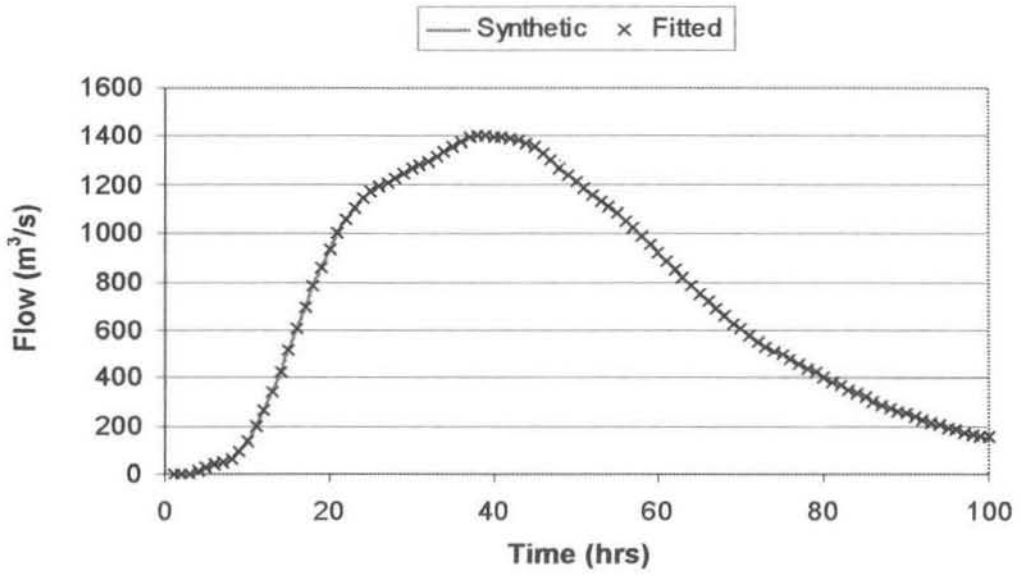


Figure 5.7: Streamflow responses for a "2 reservoir cascade"

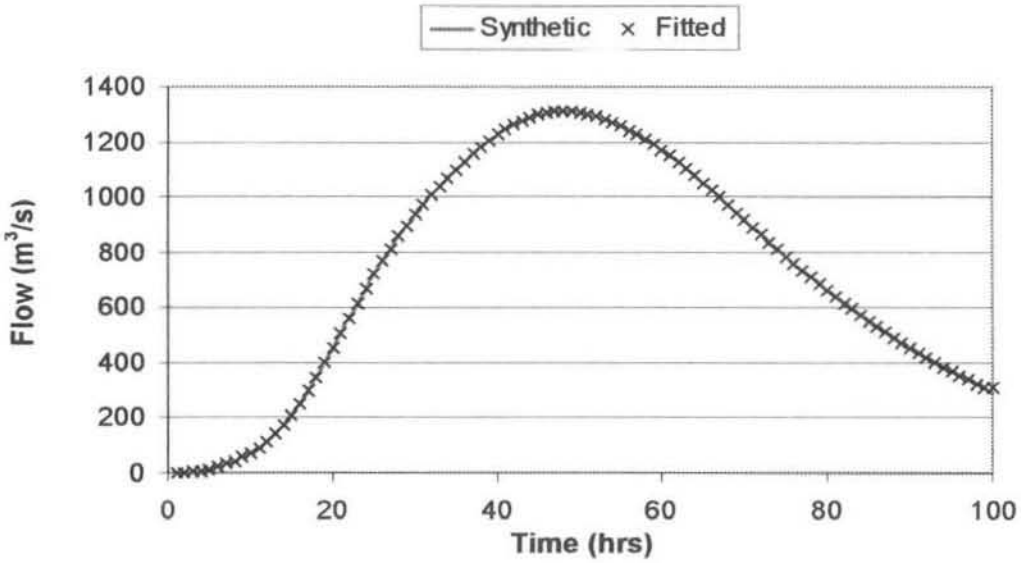


Figure 5.8: Streamflow responses for a "3 reservoir cascade"

Figure 5.9 shows the scatter plots generated from the fitting process for the model collapsed to a 2 reservoir cascade. The parameters k_2 and k_4 were fitted, with the other parameters fixed at the values indicated in Table 5.2. There are two obvious regions where distinct minima are found. An explanation for this phenomenon is the fact that the response from the cascade of two reservoirs (in series) for the case where $k_2 = 10$ and $k_4 = 15$ is the same as for the case when $k_2 = 15$ and $k_4 = 10$ (the order in which the reservoirs appear doesn't affect the response).

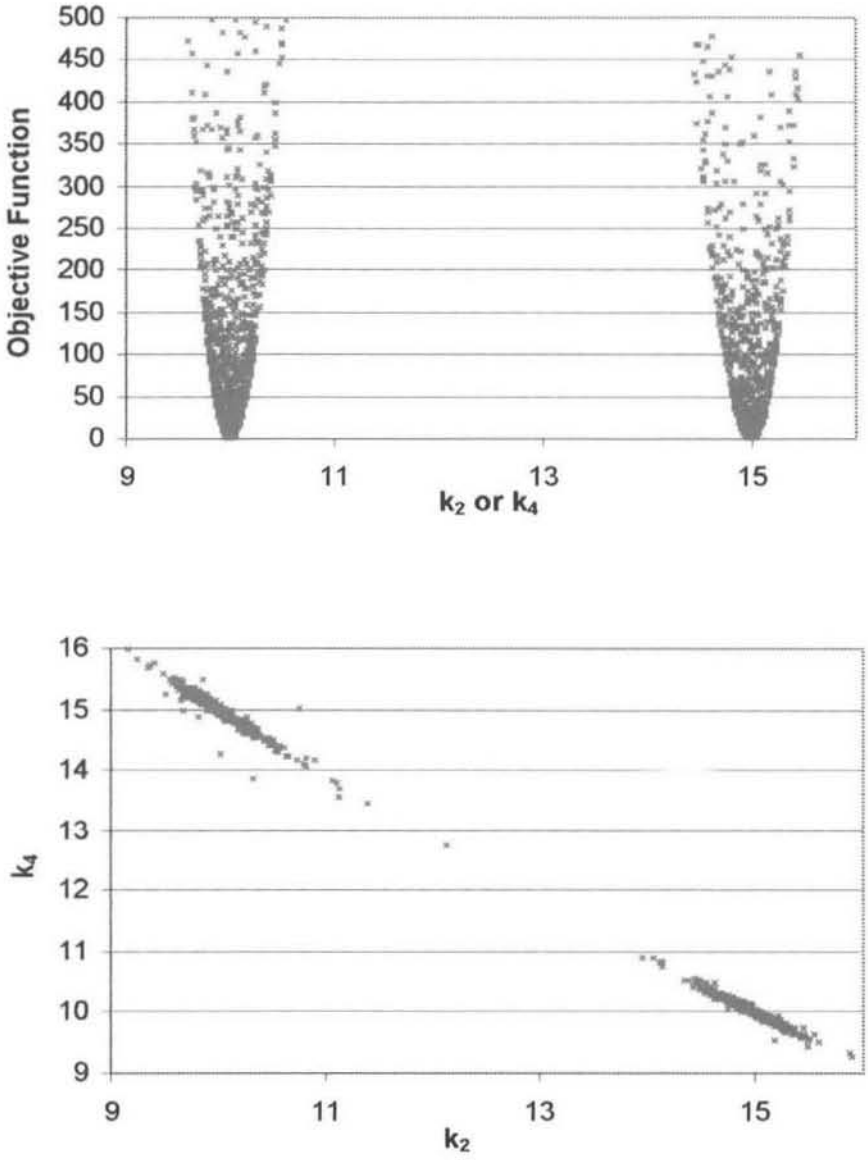


Figure 5.9: Scatter plots for "2 reservoir cascade"

The plots for the three reservoir cascade (Figure 5.10) show a similar situation to that experienced for the case of the two reservoir cascade. Each of the three parameters (k_1, k_3, k_4) can take on one of three "optimum" values (10, 12 or 14 hours). This is because the order in which the three reservoirs appear is not important, provided the input is routed through all three in succession. The circular collection of points (evident in the plot of k_1 against k_3 as well as the plot of k_1 against k_4) indicates the region where the global optimum is to be found. The other dense cluster of points on the plots is due to a local minimum, which may have trapped a standard gradient search routine.

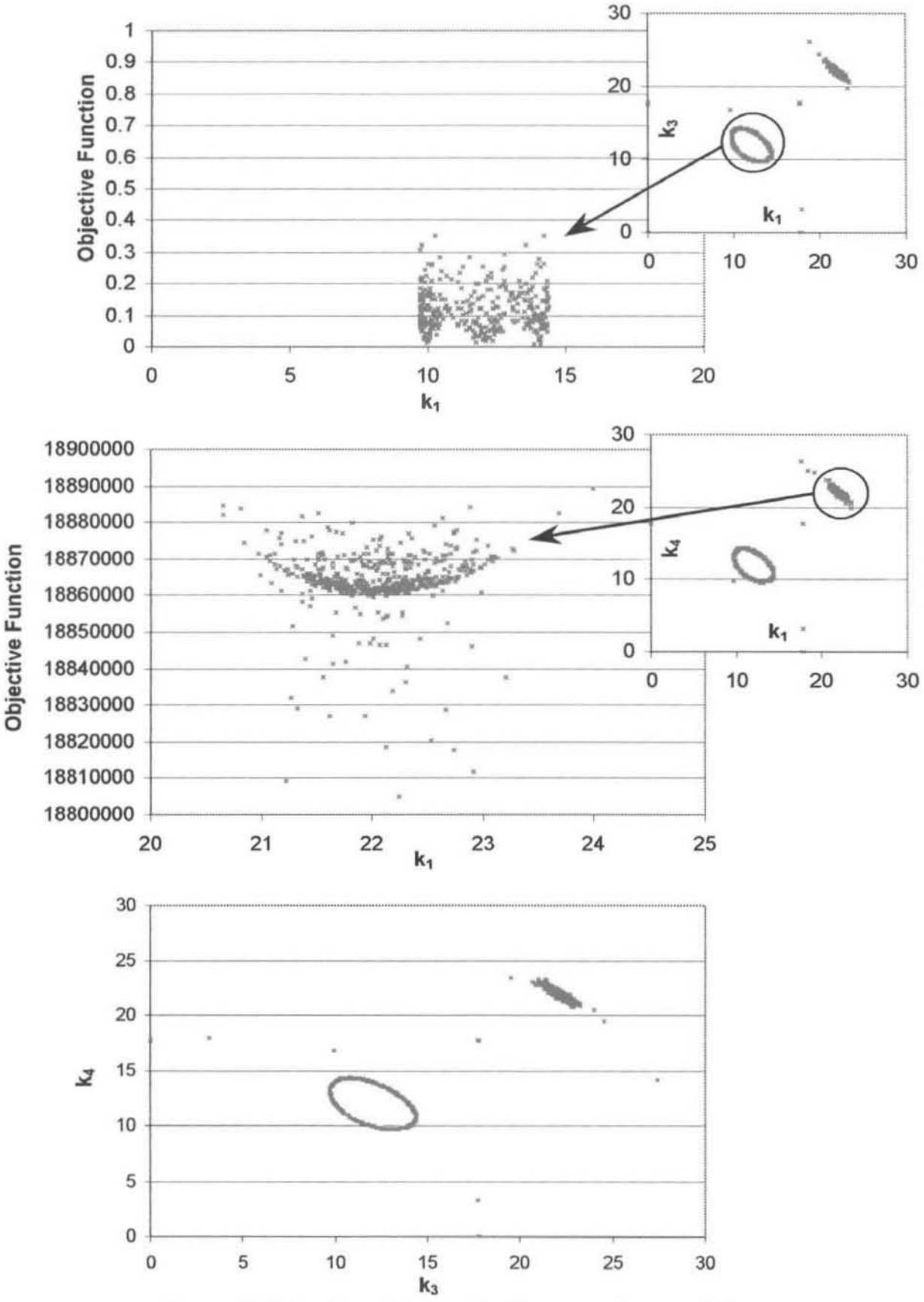


Figure 5.10: Scatter plots for the "3 reservoir cascade"

A streamflow sequence using the full three reservoir feed forward model was also generated with selected parameter values (Table 5.3). All seven parameters were fitted with the global optimum giving a difference between generated and fitted output of 0.03 m^3/s on average over 100 data points. The generated flows and fitted flows are shown in Figure 5.11 below.

Parameter	k_1	k_2	k_3	k_4	k_5	k_6	k_7
Value (hrs)	14	19	12	10	23	14	26

Table 5.3: Chosen parameter values for the full three reservoir model

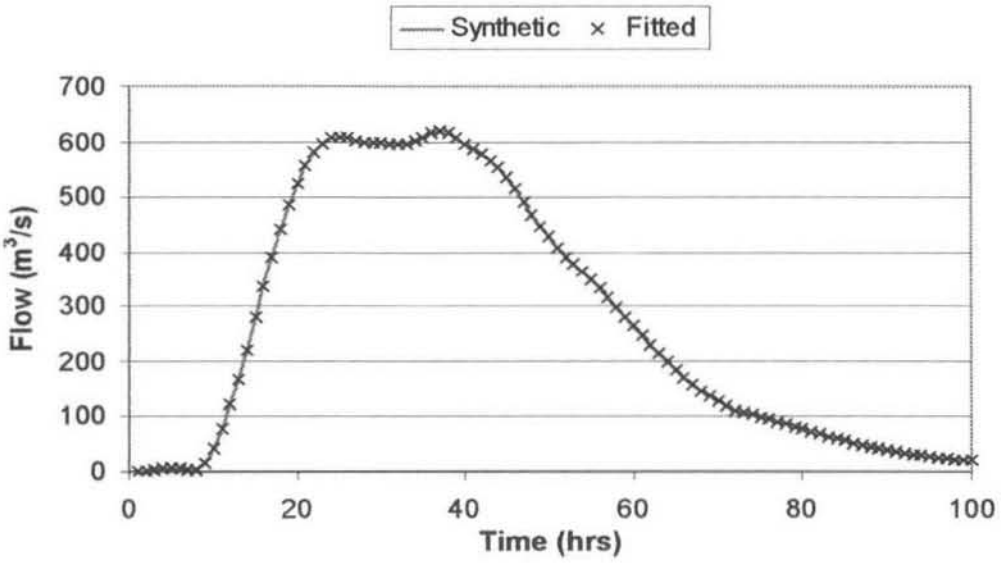


Figure 5.11: Streamflow responses for the full 3 reservoir model

It should be noted that the parameter interchangeability evident in Figures 5.9 and 5.10 is limited to the situation where the reservoirs are arranged in series. In this case there is redundancy in the parameter estimation as the order of the reservoirs does not affect the final outflows. This is overcome in a practical situation by fixing the relative values of the parameters so that for example we ensure $k_1 > k_2 > k_3$. As soon as any parallel links are introduced the reservoir configuration is effectively frozen. An example of this is the example given in Table 5.3 and Figure 5.11.

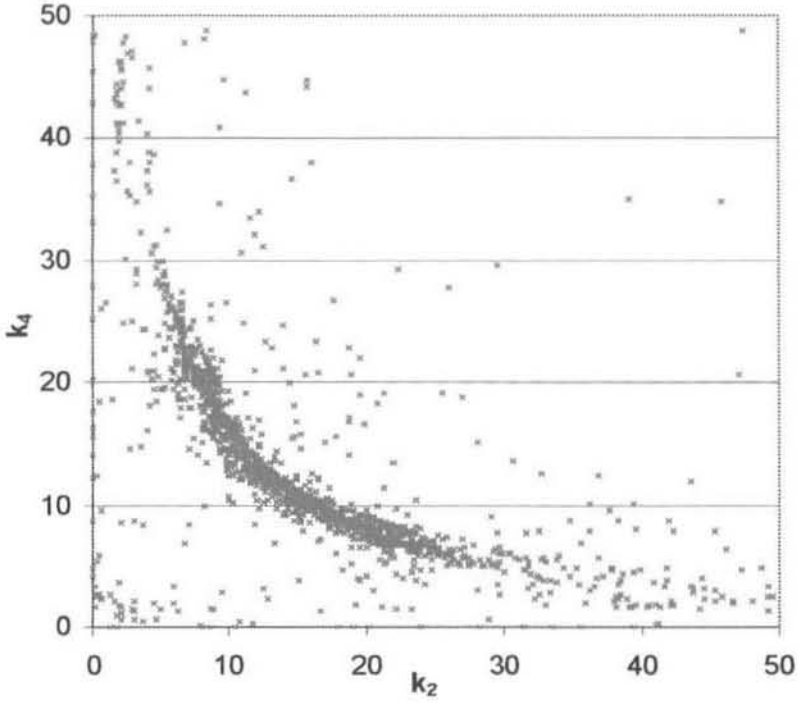


Figure 5.12: Example scatter plot showing a clustering of points near the optimum

The ability of the fitting process to accurately recover the parameter values for synthetically derived sequences (with the model in a number of different configurations) gives confidence for the fitting of parameters to observed sequences. The use of Monte Carlo techniques allows for a good investigation of the parameter space and provides many insights into the parameter relationships and the shape of the various surfaces. Situations such as that shown in Figure 5.10 and the surrounding text justify the application of Monte Carlo fitting techniques.

5.2 FITTING THE GENERAL THREE RESERVOIR FEED FORWARD MODEL

Three historical flooding events were selected from the available rainfall and runoff data sets. The general form of the three reservoir model (all 7 response parameters free) was then fitted to each of these events in turn. Three different fitting combinations were used. First a single set of parameters (treating the catchment as a single cell) was fitted to minimize the objective function, over the entire event. Subsequently, the fitting procedure described in Chapter 3 was used and the event automatically divided into a number of sub-events. The second fitting combination used the technique of fitting for each sub-event, but treated the catchment as a single lumped cell. The third option was to fit the model to the

catchment, where the catchment was represented as six separate cells in a semi-distributed manner (Chapter 4). The results of the fits for each of these three events follow in chronological order.

5.2.1 EVENT ONE –DECEMBER 1995

Event one is 360 hours in length, starting on the 16th of December and ending on the 31st of December 1995. The maximum discharge of 242.6 m^3/s was reached during the second of three main peaks. Figure 5.13 shows the precipitation input for this event.

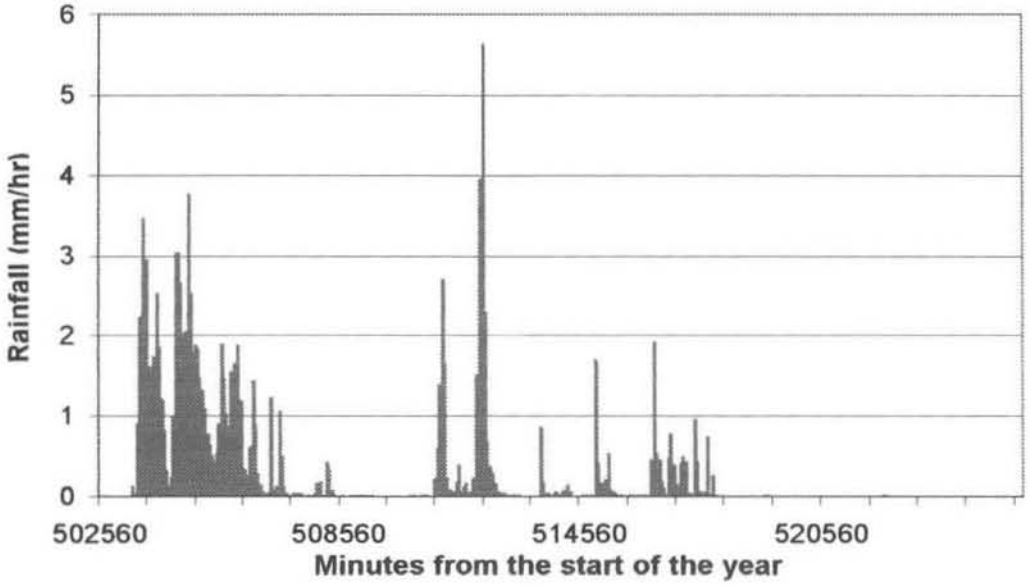


Figure 5.13: Rainfall series for December 1995

For the case where a single set of parameters was fitted to the entire event, the lowest value of the objective function occurred with the rainfall lagged by three hours. The sum of squared differences between the observed and fitted hydrograph was 226 221.4 m^6/s^2. This means that the average difference between the observed and fitted values is 25 m^3/s. A plot of these flows is given in Figure 5.14.

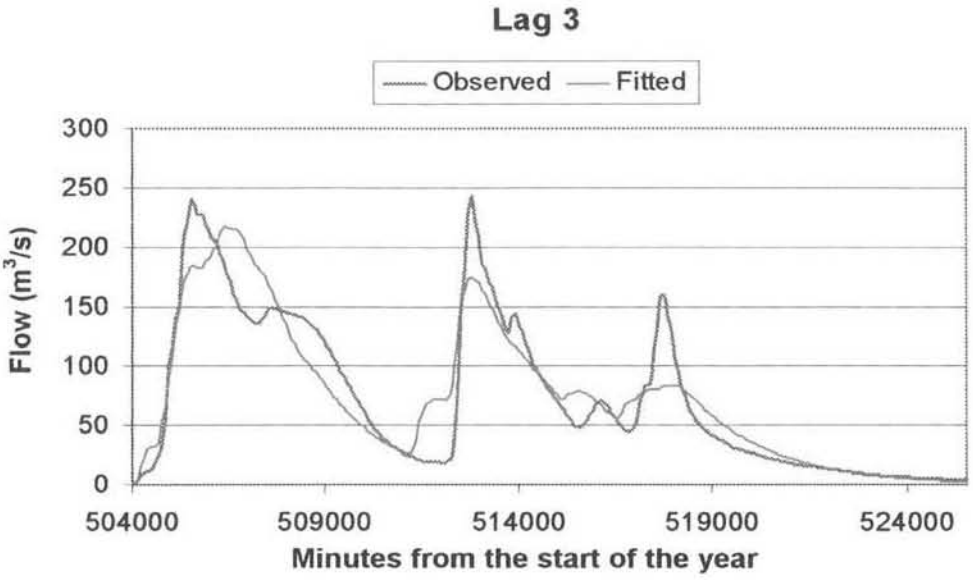


Figure 5.14: Observed and fitted streamflow for December 1995
(single cell, single parameter set for the entire event)

Figure 5.15 shows similar plots for a range of precipitation lags between zero and nine hours. For lag times greater than this, the fit is much poorer. Although the peaks are not well matched, it is encouraging that the model is able to mimic the general behaviour of the observed hydrograph.

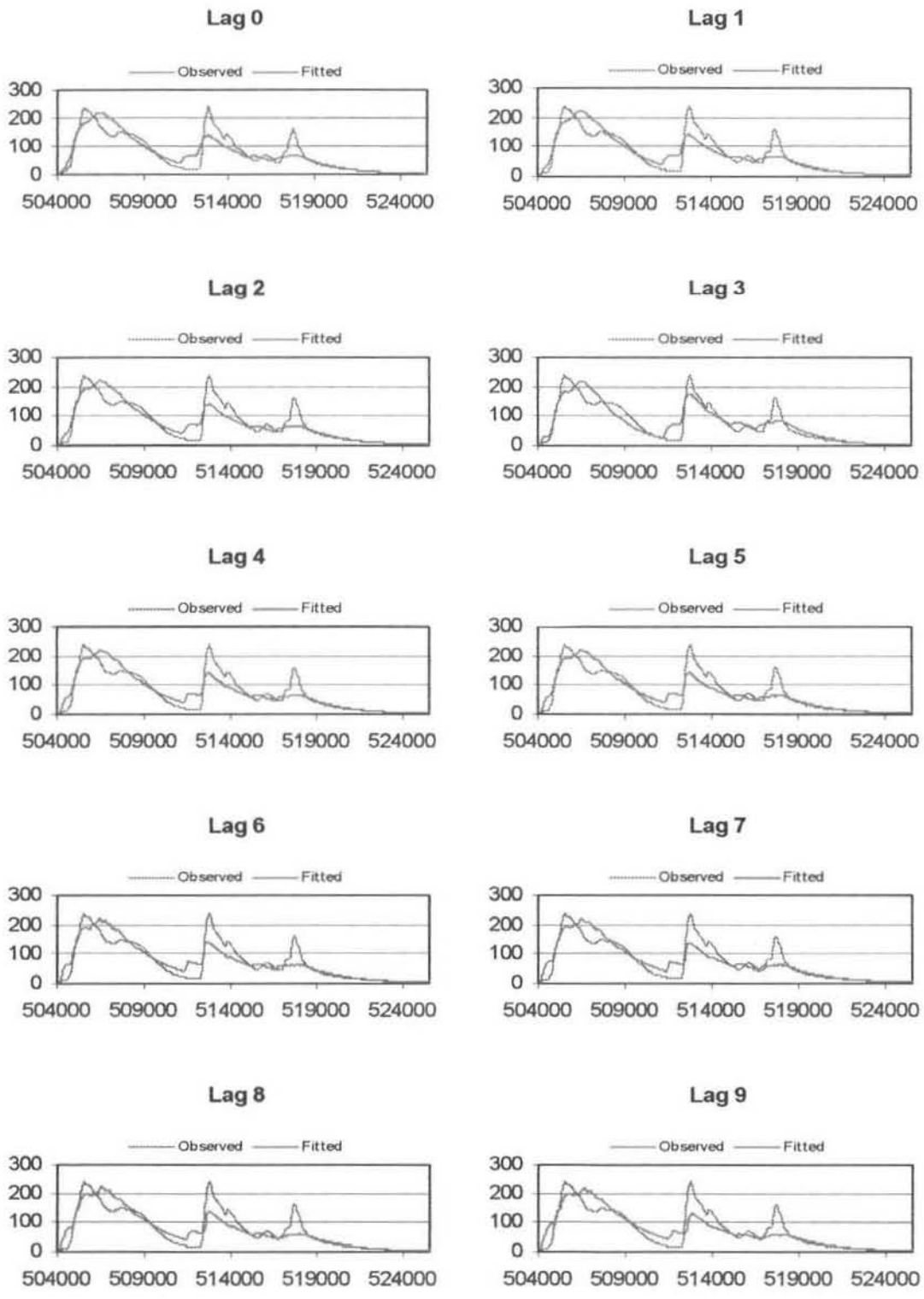


Figure 5.15: Observed and fitted streamflow for December 1995 (single cell, single parameter set for the entire event)

The fits are greatly improved by using the algorithm described in Chapter 3. For the case where the catchment was treated as a single lumped cell (with separate parameters for each sub-event), the sum of squared differences between the observed and fitted flows sequences was 31 138 m^6/s^2 for a rainfall lag of five hours. This equates to an average difference between observed and fitted flows of 9.3 m^3/s at each timestep, which is less than half the difference obtained by fitting for the entire event as a whole. The observed and fitted streamflows are plotted in Figure 5.16. Figure 5.17 gives plots for a range of lags. The best fit for event one when the catchment was sub-divided into six cells occurred for a rainfall lag of five hours. In this case the sum of squared differences (28 782.4 m^6/s^2) was slightly lower than for the case of a single cell. The average difference at each timestep was therefore 8.9 m^3/s. Figures 5.18 and 5.19 show plots of the fitted and observed flows.

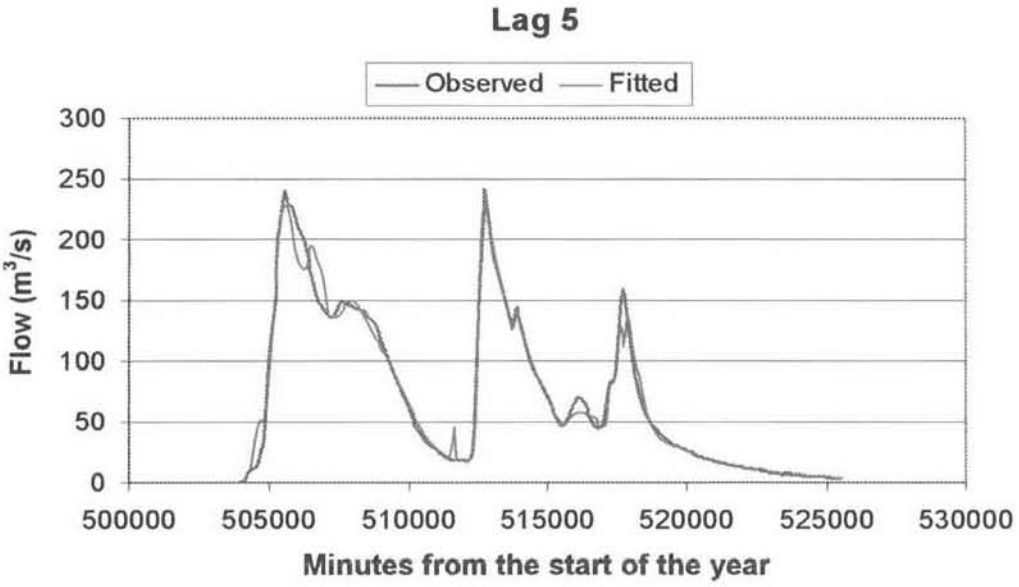


Figure 5.16: Observed and fitted streamflow for December 1995 (single cell, separate parameter set for each sub-event)

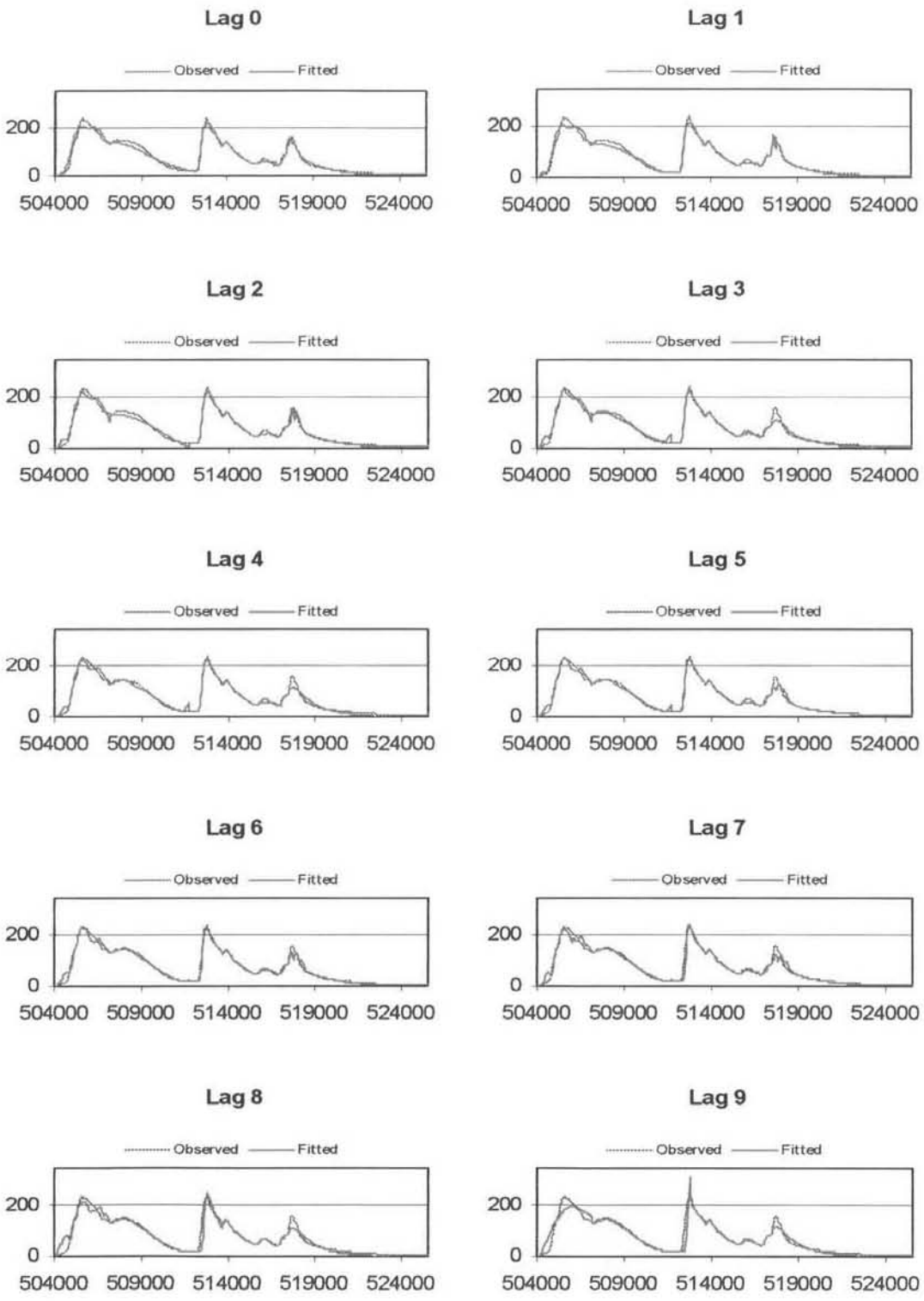


Figure 5.17: Observed and fitted streamflow for December 1995 (single cell, separate parameter set for each sub-event)

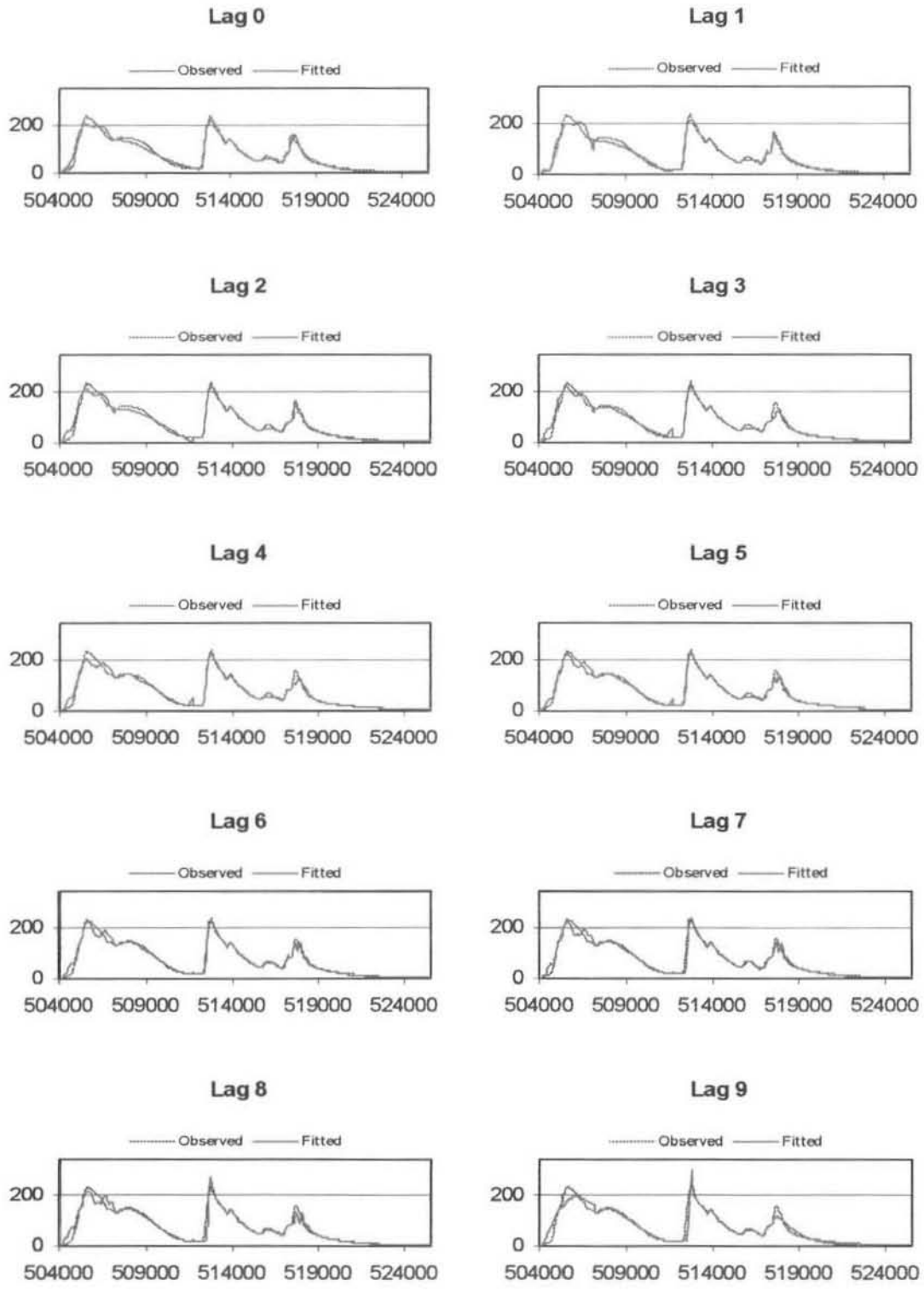


Figure 5.18: Observed and fitted streamflow for December 1995 (six cells, separate parameter set for each sub-event)

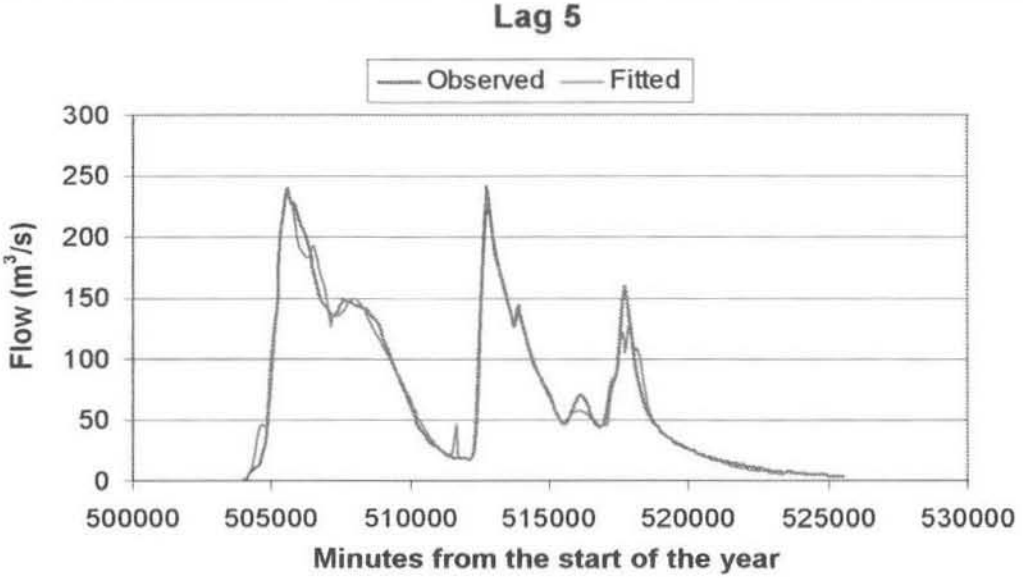


Figure 5.19: Observed and fitted streamflow for December 1995
(six cells, separate parameter set for each sub-event)

5.2.2 EVENT TWO – FEBRUARY 1996

The second event studied was a major flood event occurring between the 10th and 20th of February 1996. The peak discharge recorded at the gauge on the Liebenbergsvlei was 562 m^3/s. Close co-operation between the Department of Water Affairs and the South African Weather Bureau ensured that the consequences of this flood were mitigated. The Liebenbergsvlei is a sub-catchment of the Vaal catchment. The peak flood discharge entering the Vaal dam was 4700 m^3/s. Maximum outflow from the dam was reduced to 2300 m^3/s. This was achieved using rainfall information obtained from the MRL5 weather radar at Bethlehem. A flood routing model at the Department of Water Affairs was used to route the input and make decisions on how much water to release from the already full Vaal dam in order to prevent overtopping of the wall and subsequent downstream flooding. The rainfall sequence generating this event is shown in Figure 5.20.

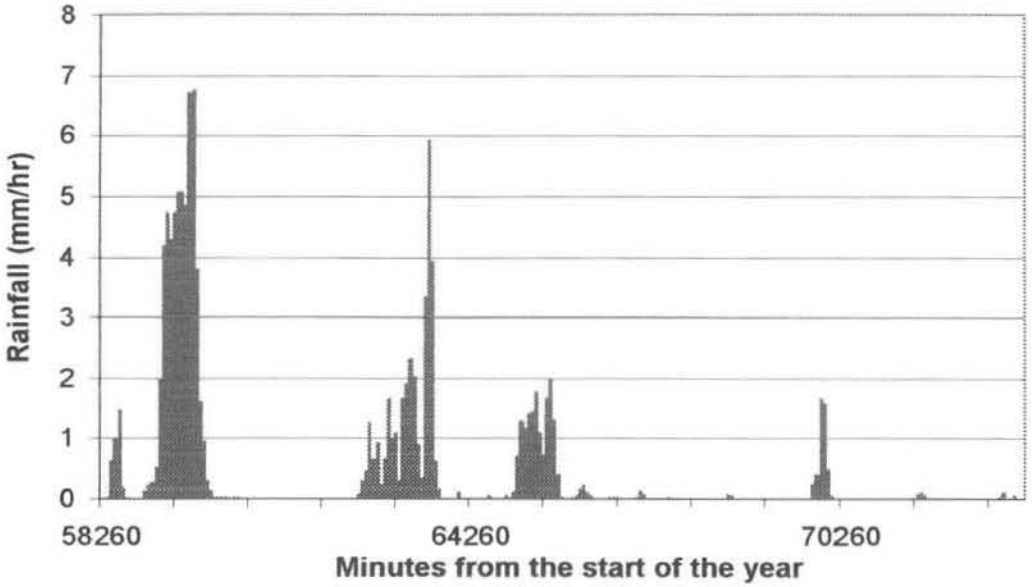


Figure 5.20: Rainfall sequence for February 1996

The fit achieved when fitting a single set of parameters for the entire event is shown in Figure 5.21. The lowest value of the objective function (914 847 m^6/s^2) was found for a rainfall lag of nine hours. The average difference between observed and fitted flows over 226 hours was 63.6 m^3/s. The fits for a full range of rainfall lags are shown in Figure 5.22.

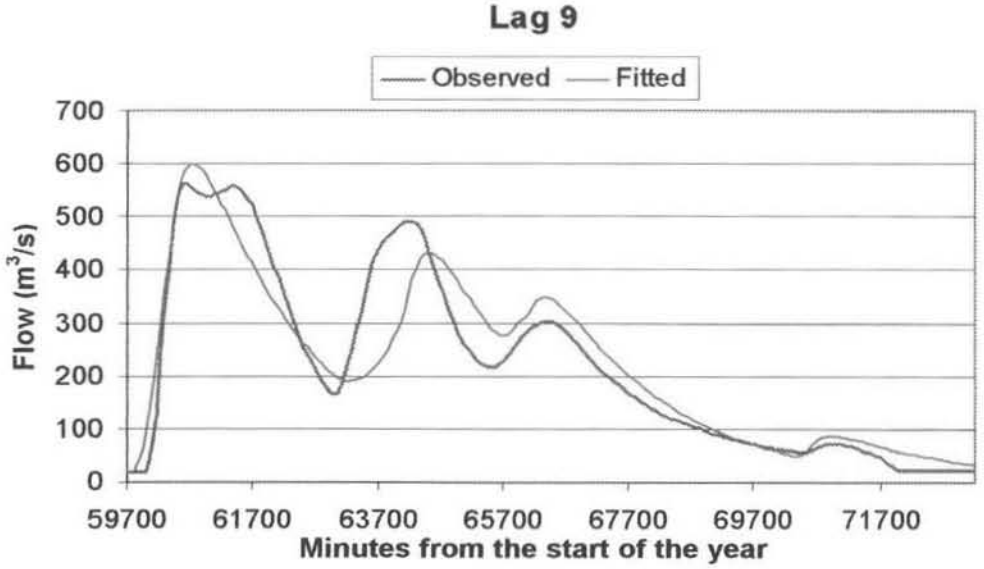


Figure 5.21: Observed and fitted streamflow for February 1996
(single cell, single parameter set for the entire event)

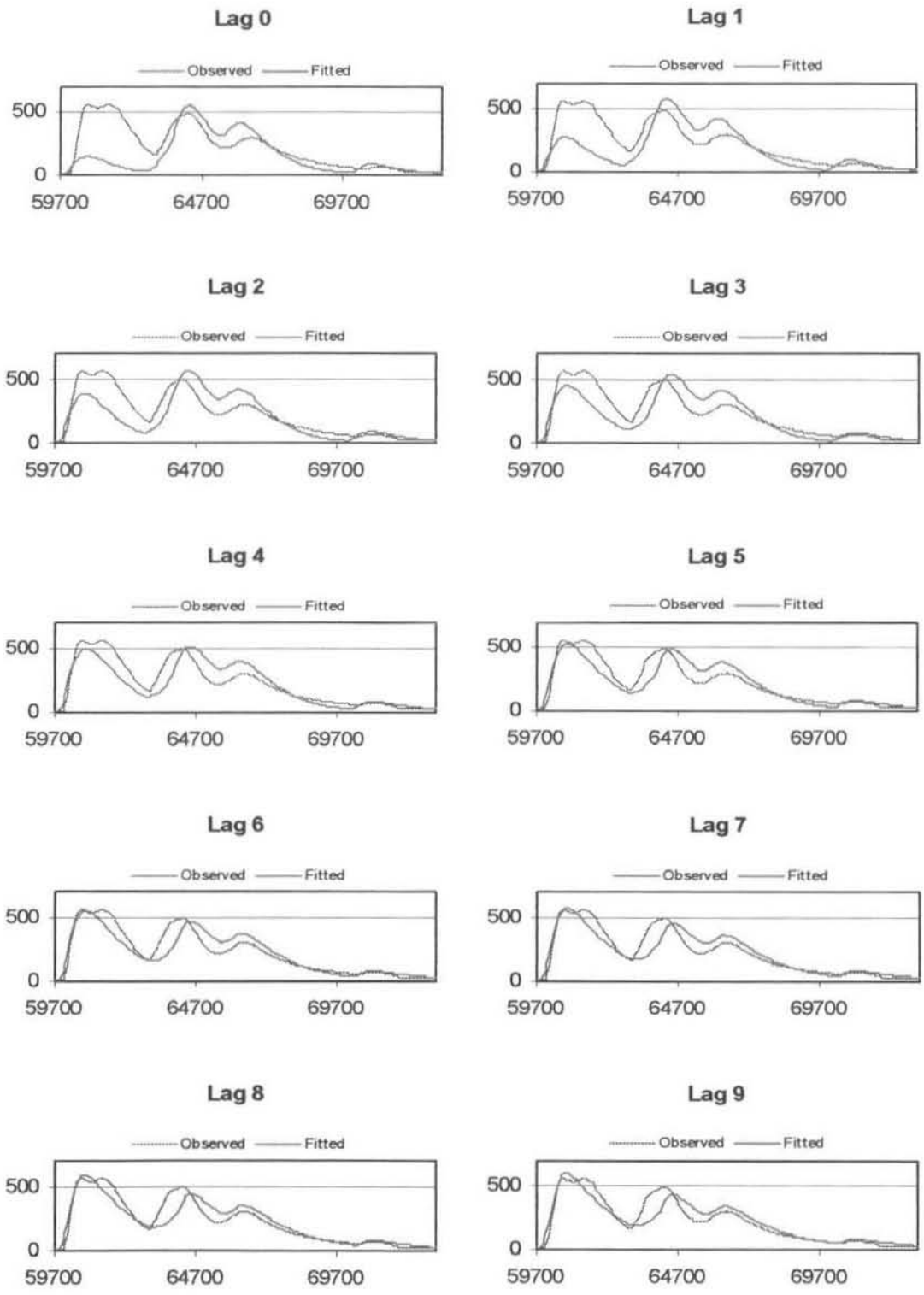


Figure 5.22: Observed and fitted streamflow for February 1996 (single cell, single parameter set for the entire event)

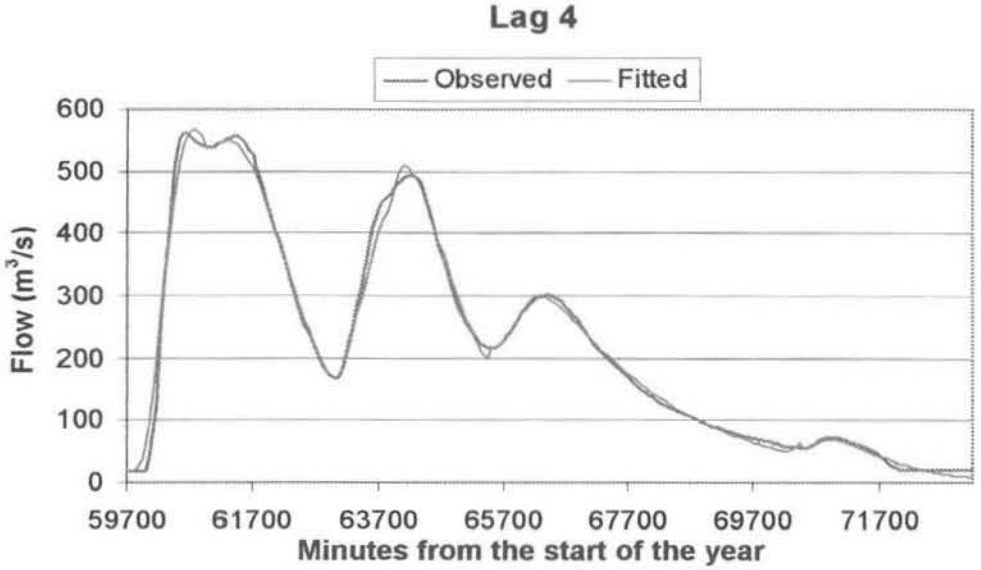


Figure 5.23: Observed and fitted streamflow for December 1995
(single cell, separate parameter set for each sub-event)

When the sub-event fitting procedure was used the degree of fit was greatly improved, this visual improvement can be seen in Figures 5.23 and 5.25. When the catchment was treated as a single lumped cell, the value of the objective function was at a minimum (42 265 m^6/s^2) for a rainfall lag of four hours. The average difference between observed and fitted flows was 13.8 m^3/s. The objective function value with the catchment divided into six cells was 51 510 m^6/s^2, at a rainfall lag of three hours. The average difference between observed and fitted streamflows is therefore 15.1 m^3/s. The fits over a range of lags are indicated in Figures 5.24 and 5.26.

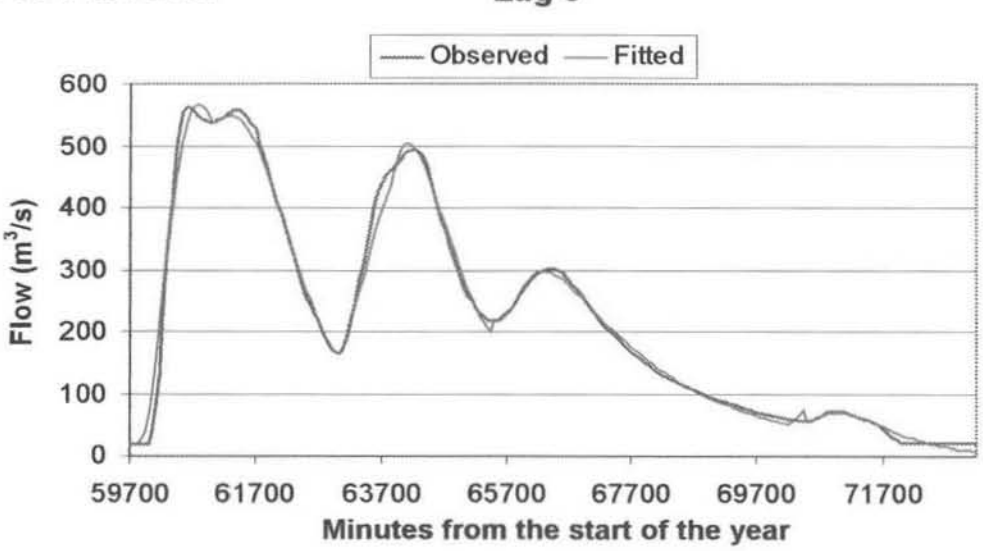


Figure 5.24: Observed and fitted streamflow for February 1996
(six cells, separate parameter set for each sub-event)

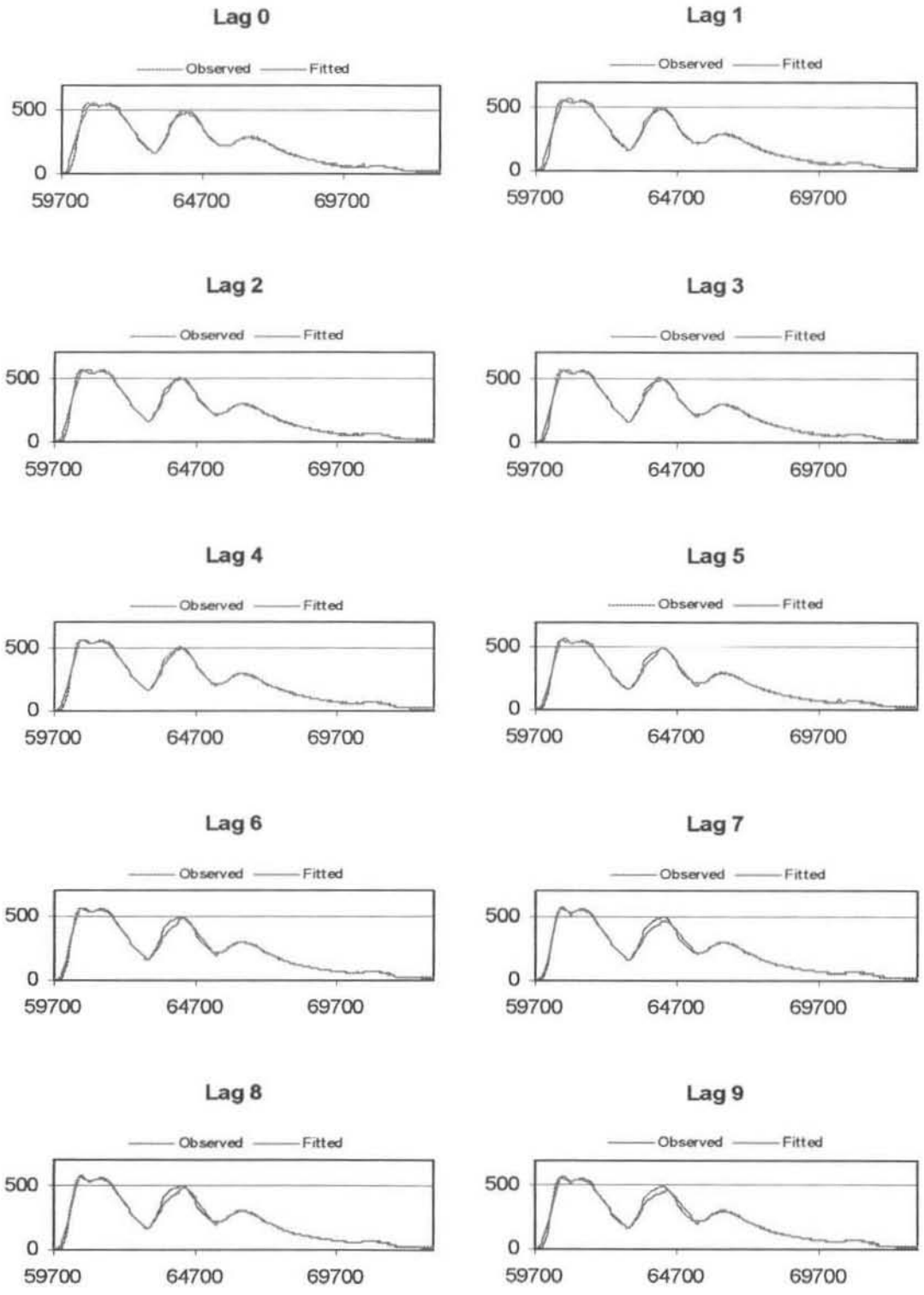


Figure 5.25: Observed and fitted streamflow for February 1996
(single cell, separate parameter set for each sub-event)

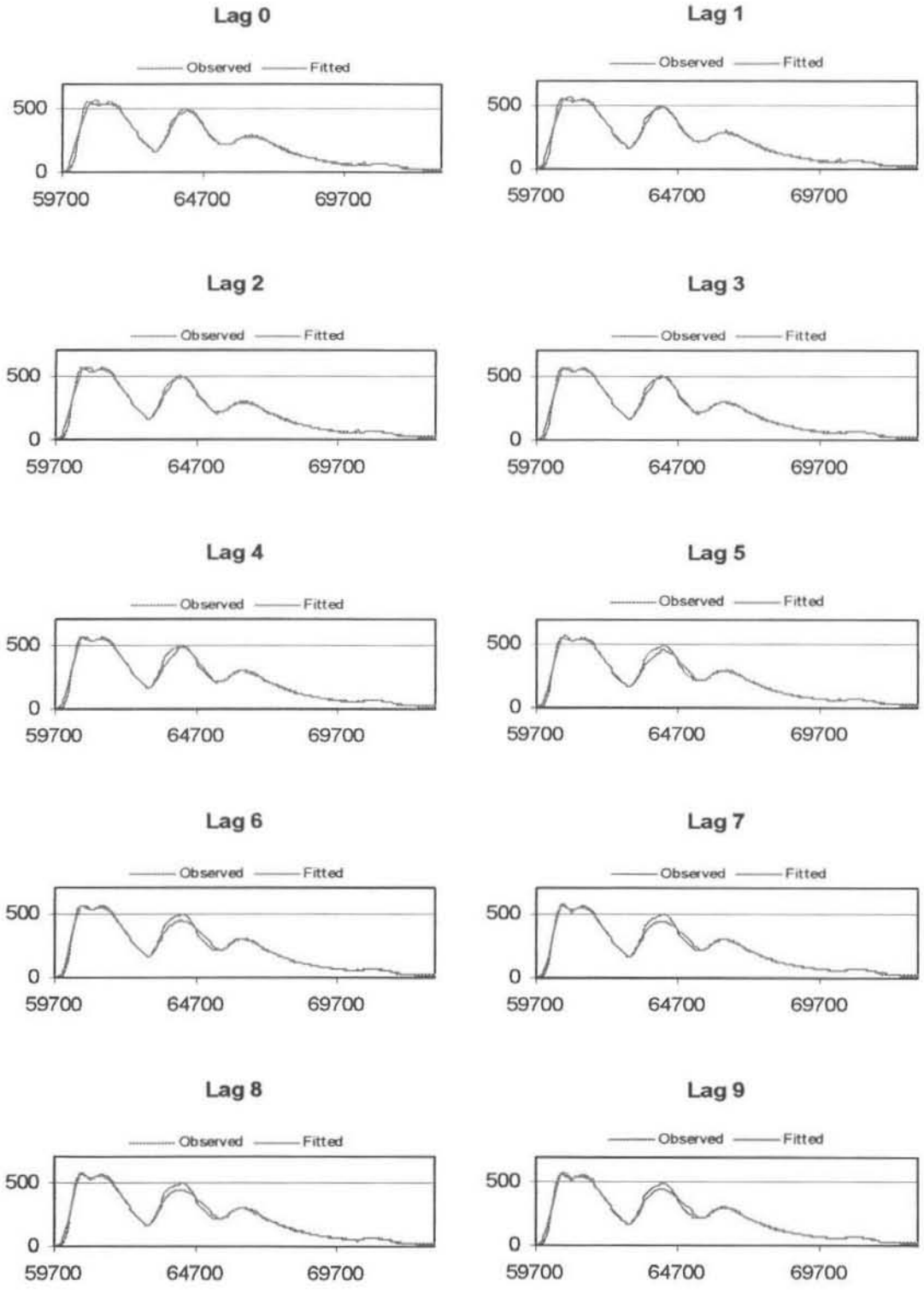


Figure 5.26: Observed and fitted streamflow for February 1996
(six cells, separate parameter set for each sub-event)

5.2.3 EVENT THREE – NOVEMBER/DECEMBER 1998

The third event for occurred between the 16th of November and the 7th of December 1998. The maximum discharge was 129 m^3/s. Figure 5.27 shows the rainfall sequence that contributed to the streamflows.

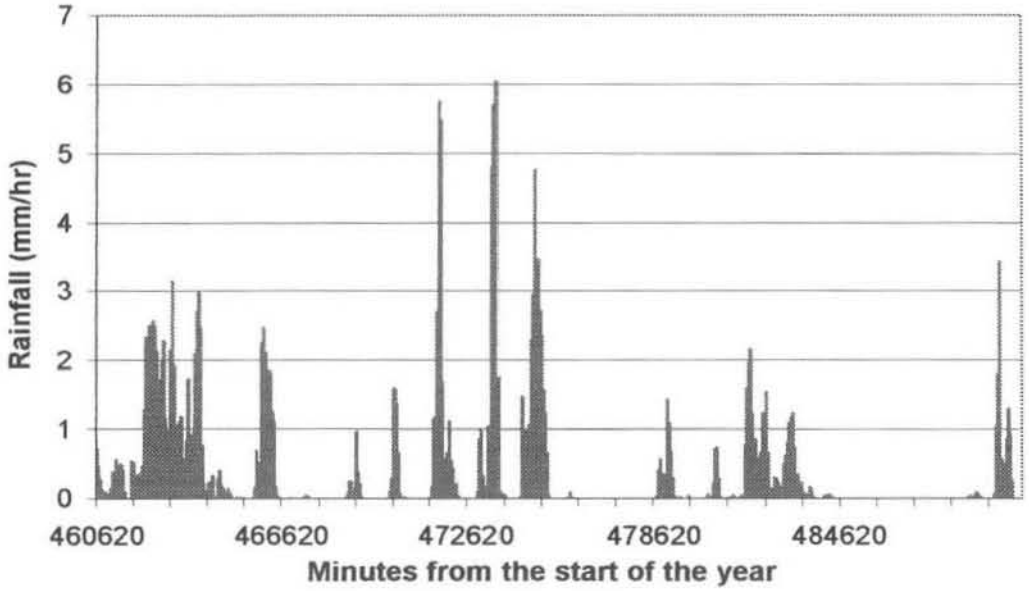


Figure 5.27: Rainfall sequence for November/December 1998

The fit achieved when using a single set of parameters for the entire event was not very good at all. The minimum value of the objective function, at a lag of three hours, was 319385 m^6/s^2. Over the 474 hour long event, this produces an average difference between observed and fitted streamflows of 26 m^3/s. This average difference is twenty percent of the peak discharge. Figures 5.28 and 5.29 show plots of the fits achieved.

A significant improvement in the fit is made by using the sub-event fitting algorithm. With the catchment lumped as a single cell the optimum value of the objective function drops to 6464 m^6/s^2. Reducing the average difference between observed and fitted values to 3.7 m^3/s, now less than three percent of the peak discharge. The plots are given in Figures 5.30 and 5.32.

With the catchment sub-divided into six cells the optimum fit occurs at a rainfall lag of 7 hours, the objective function value is 5445 m^6/s^2. The corresponding average difference between fitted and observed discharges is now 3.4 m^3/s, 2.6 percent of the maximum discharge. The results are shown in Figures 5.31 and 5.33.

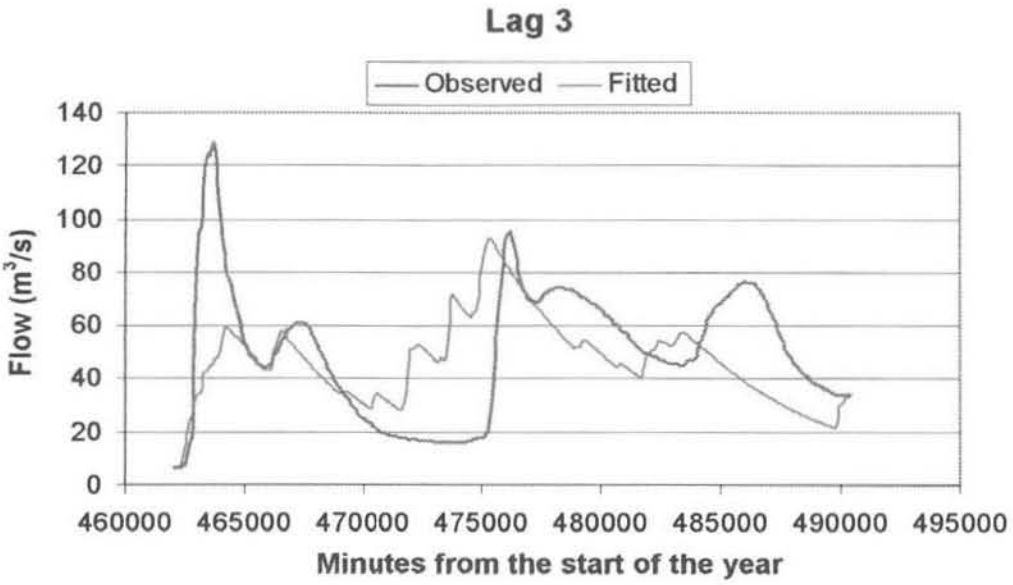


Figure 5.28: Observed and fitted streamflow for November/December 1998 (single cell, single parameter set for the entire event)

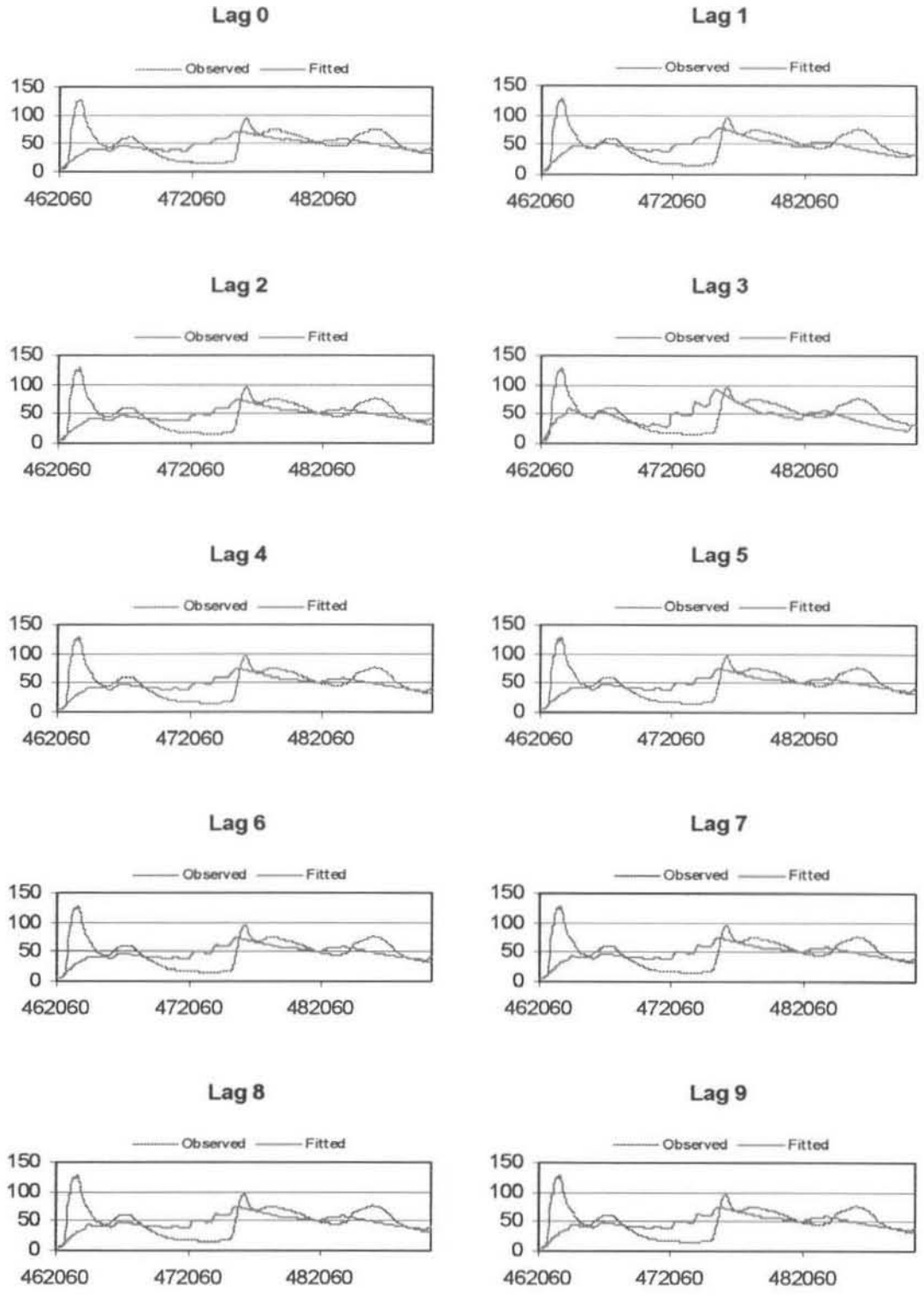


Figure 5.29: Observed and fitted streamflow for November/December 1998 (single cell, single parameter set for the entire event)

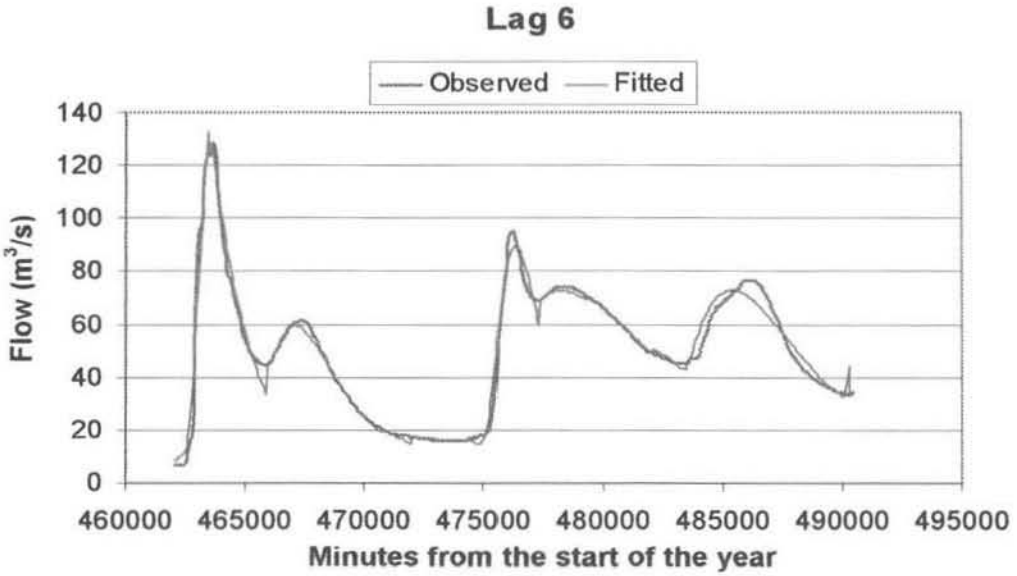


Figure 5.30: Observed and fitted streamflow for November/December 1998 (single cell, separate parameter set for each sub-event)

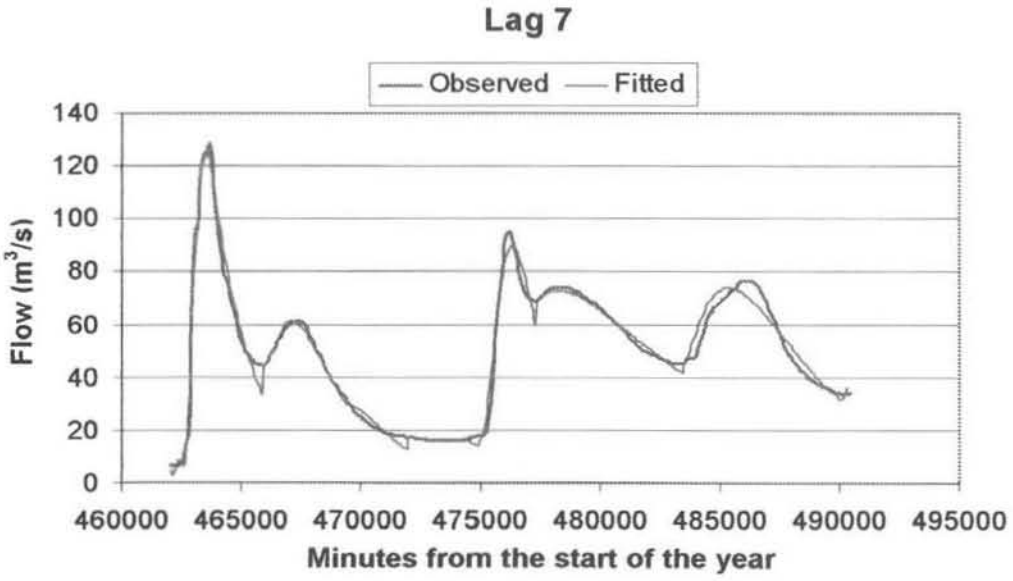


Figure 5.31: Observed and fitted streamflow for November/December 1998 (six cells, separate parameter set for each sub-event)

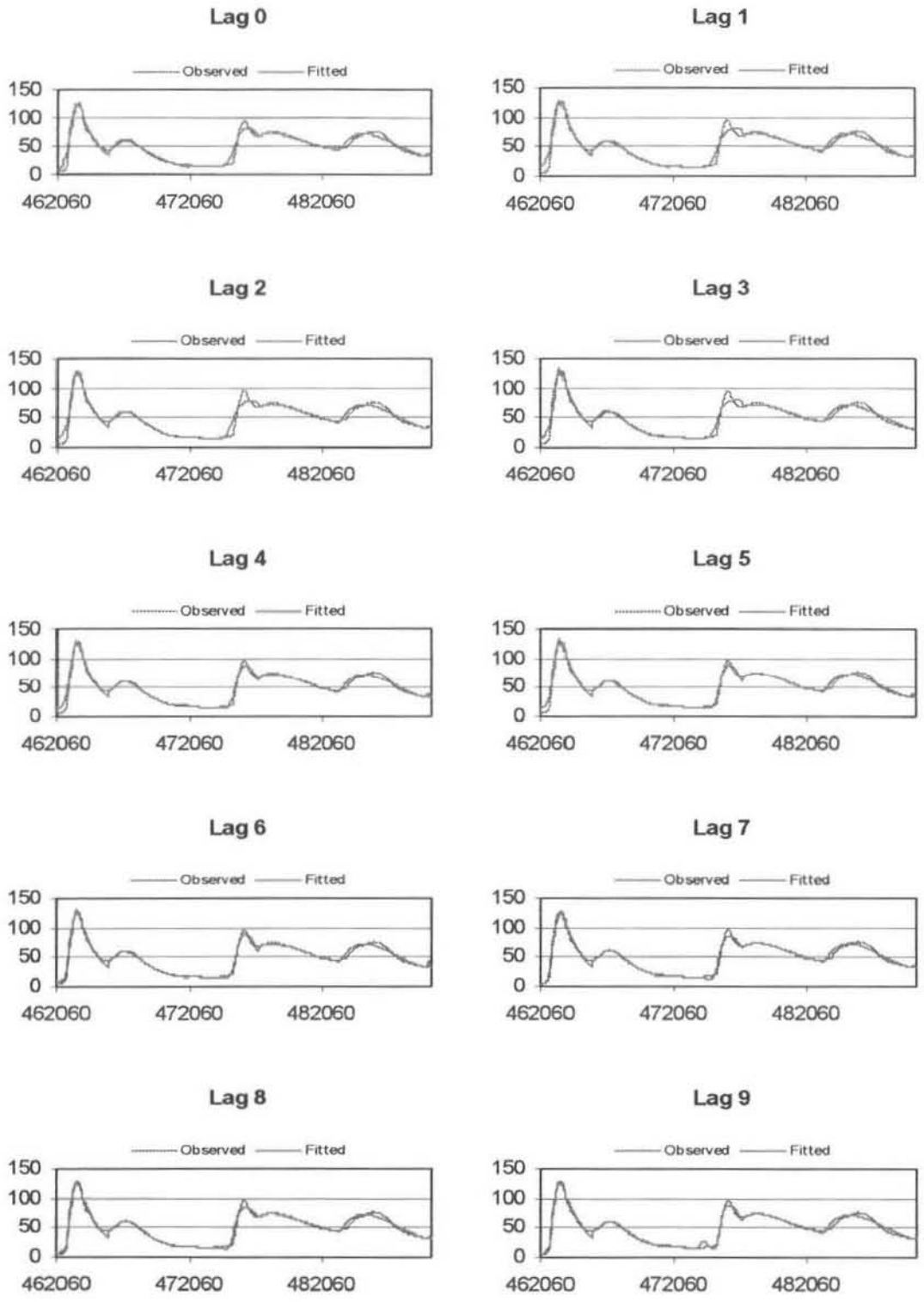


Figure 5.32: Observed and fitted streamflow for November/December 1998 (single cell, separate parameter set for each sub-event)

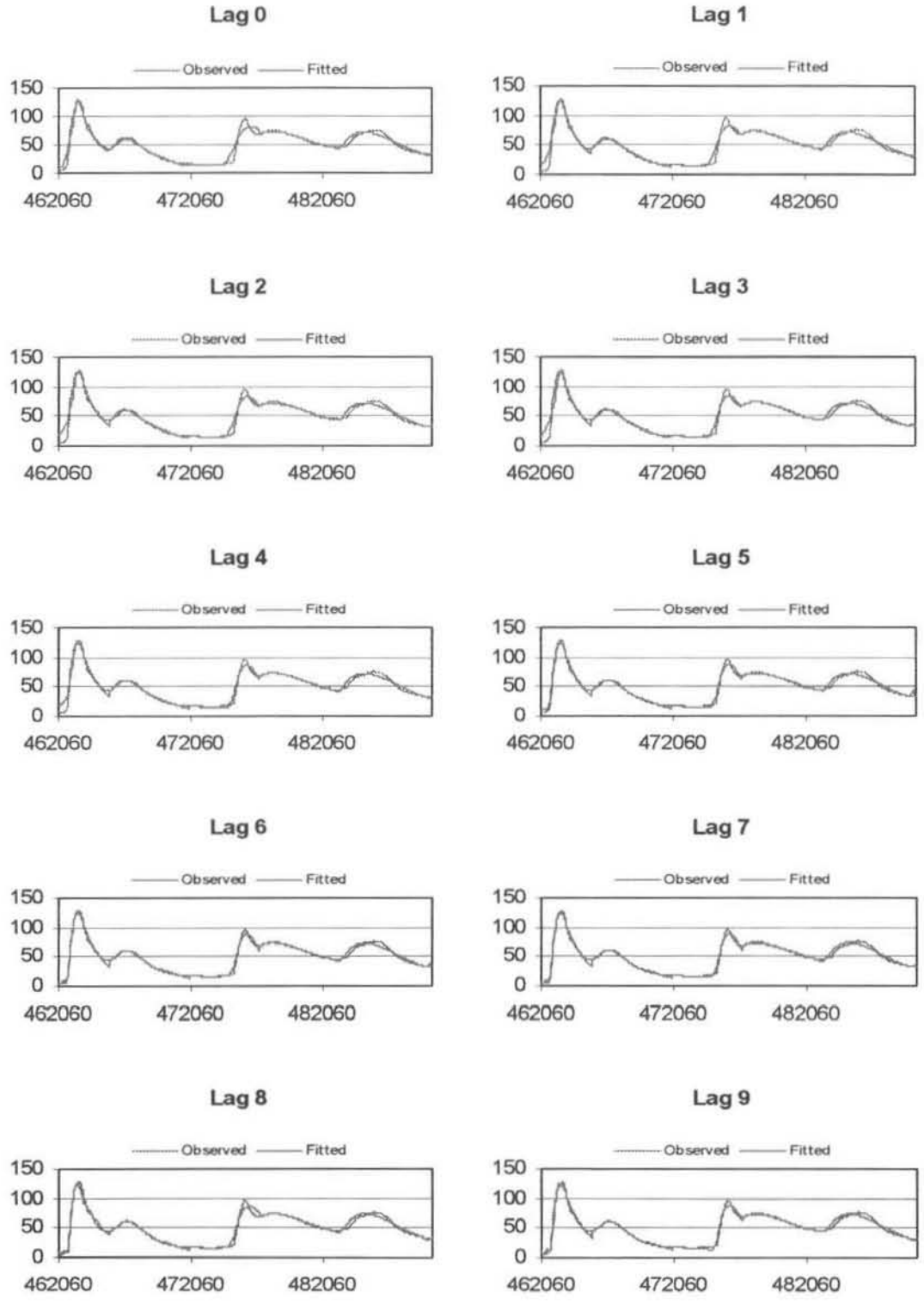


Figure 5.33: Observed and fitted streamflow for November/December 1998 (six cells, separate parameter set for each sub-event)

5.3 CONDITIONAL FORECASTS

Forecasts of future streamflow can be made using parameters conditioned on current observations. Using a conditional forecasting technique such as this ensures that the parameters being used to produce estimates of the streamflow reflect the current moisture state of the catchment. The moisture condition of the catchment varies with time and has a significant effect on the systems response to a given precipitation input. Figures 5.35 and 5.36 show an example of streamflow estimation using conditional parameter sets for the February 1996 event. The seven reservoir response parameters were adjusted to produce the best fit (in a sum of squared differences sense) to various portions of the observed streamflow hydrograph. The resulting "forecasts" are the models response to the subsequently observed rainfall using the conditional parameter set. The plots show what flows the model would have predicted as the rainfall observations were received in real time. The fitted parameters are indicated in figure 5.37, with k_5 being omitted as it maintains a constant value of 10^6 hours. The mean value of each parameter over this event is given in Table 5.4.

Parameter	k_1	k_2	k_3	k_4	k_5	k_6	k_7
Value (hrs)	5.78	238556.63	8.04	23.14	10^6	10645.20	4.42

Table 5.4: Mean parameter values for the full three reservoir model

Looking at the mean parameter values one can see that the flow paths associated with k_2, k_5 and k_7 are effectively blocked, since the corresponding response times are very long. The resulting model configuration is a 3 reservoir cascade with a loss from the second reservoir (Figure 5.34).

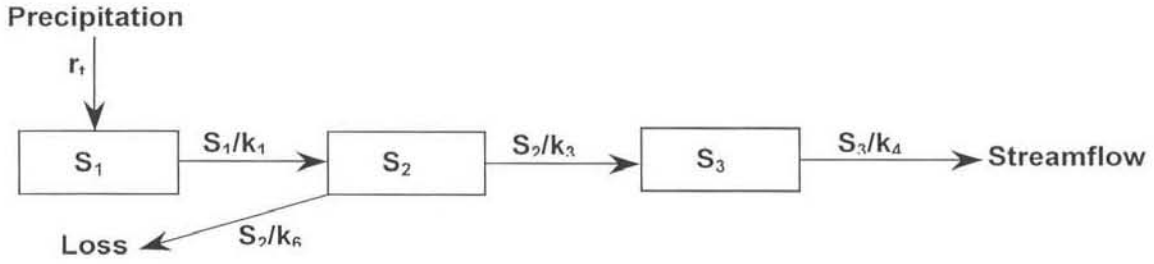


Figure 5.34: A 3 reservoir cascade with loss from the second reservoir.

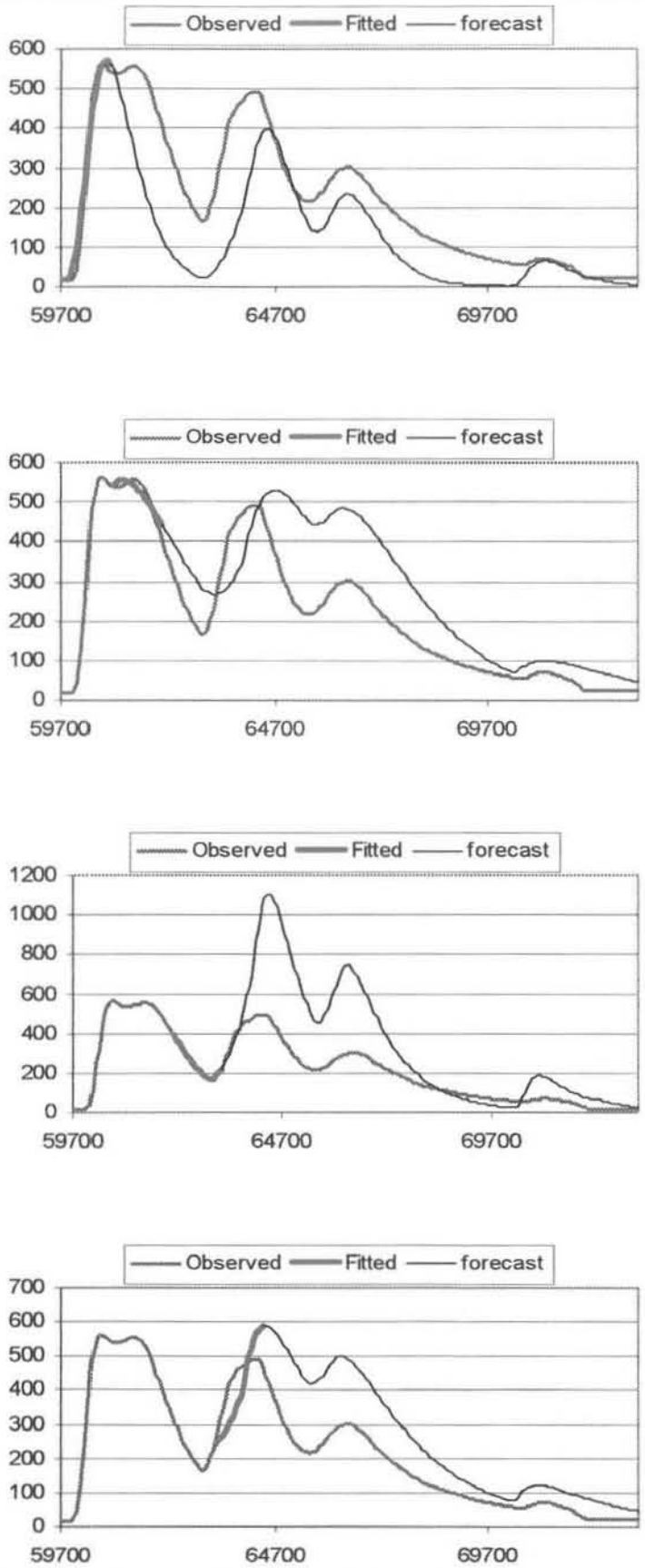


Figure 5.35: Conditional streamflow simulation with parameters fitted by minimizing the sum of squared differences on different portions of the flow record

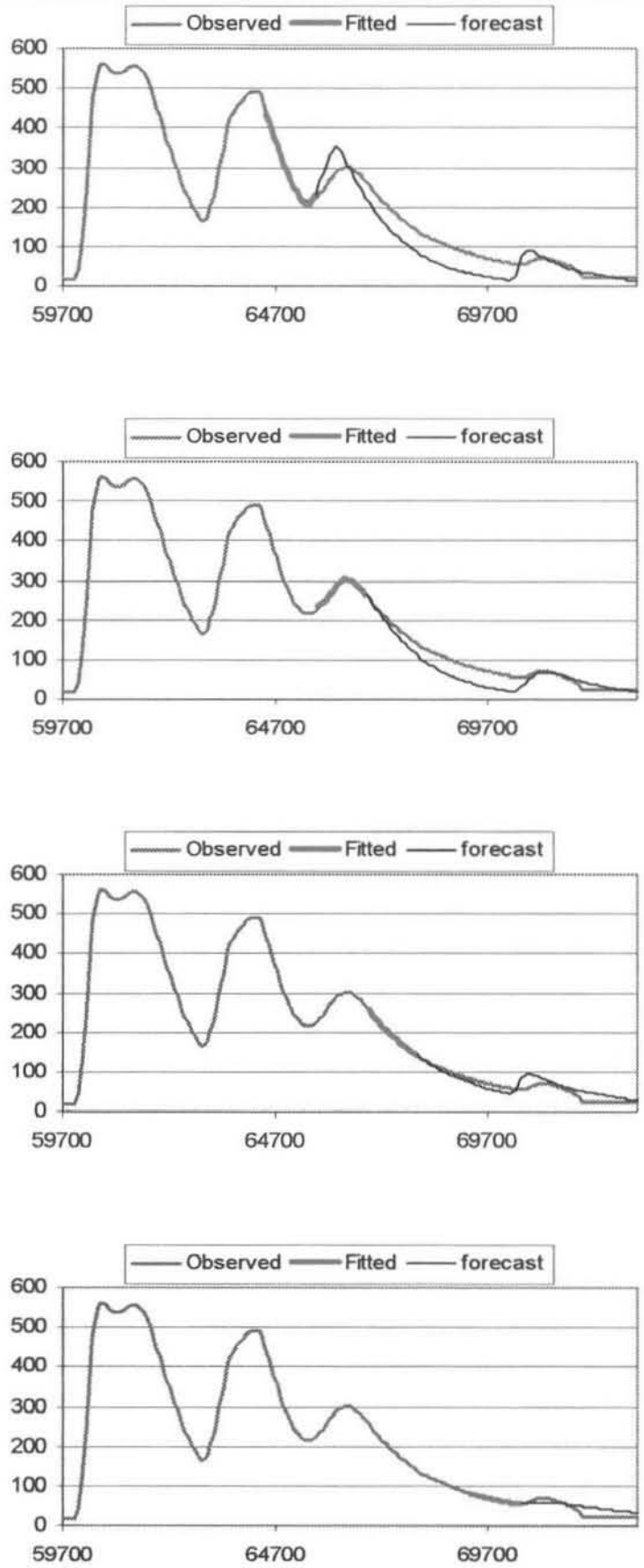


Figure 5.36: Conditional streamflow simulation with parameters fitted by minimizing the sum of squared differences on different portions of the flow record

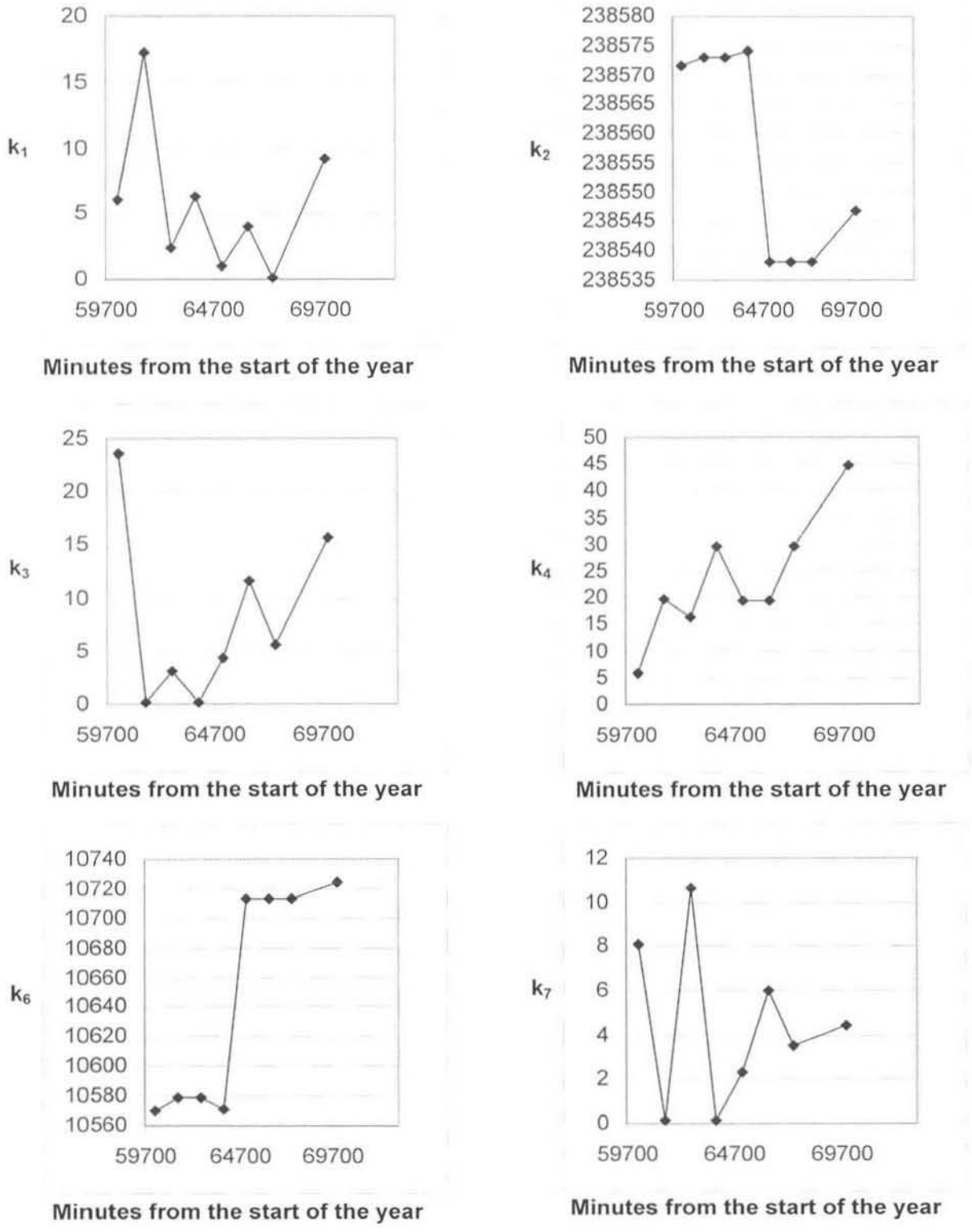


Figure 5.37: Parameters fitted by minimizing the sum of squared differences on different portions of the flow record

Forecasting in the true sense of the word, implies that there is no prior knowledge of future inputs. Figures 5.40 and 5.41 indicate what forecasts of future flows are produced when there is no future information available. As would be expected; with no input, the reservoirs produce a streamflow recession draining according to the values of the response parameters and reservoir storages. Figure 5.38 is a "close up" view and shows that the forecast with no information into the future is actually quite good up to 2 hours ahead, after this the forecast is rather poor.

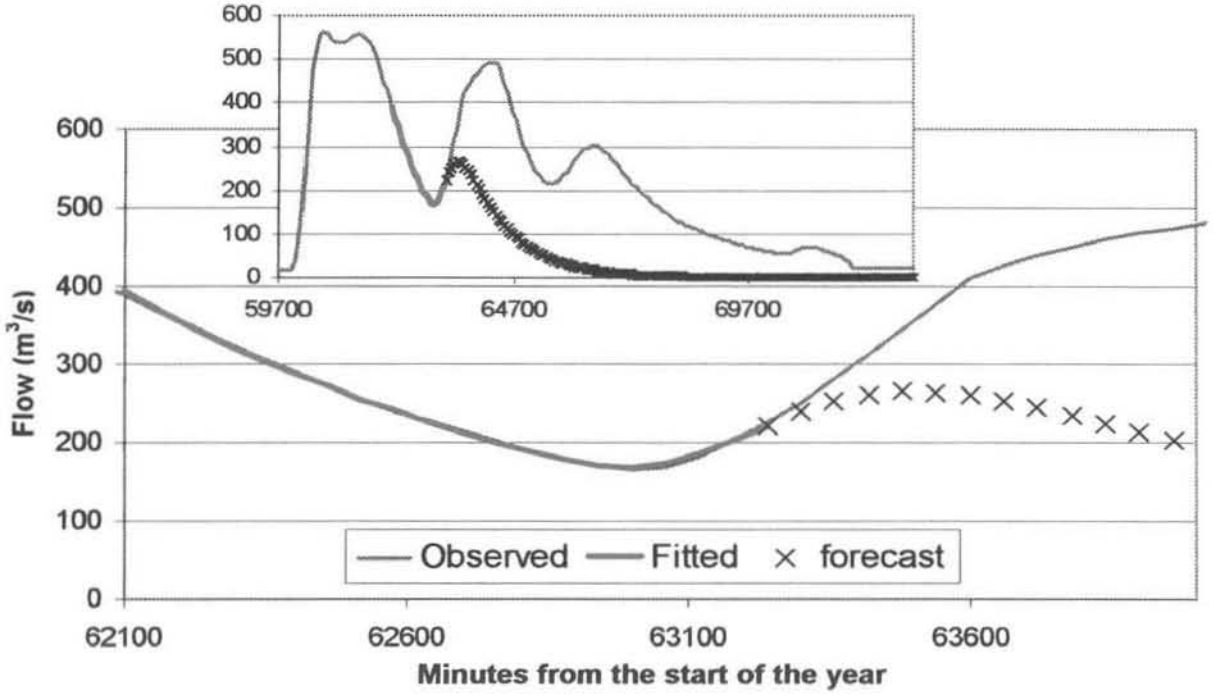


Figure 5.38: Forecast deviation from observed streamflow values

Figure 5.39 shows a similar "close up" and indicates that the model is able to produce a far better forecast when the parameters have been fitted on a recession. This is because the catchment behaves reasonably linearly on the recession as most of the non-linearity's occur in the observed to effective rainfall conversion process (see Chapter 2).

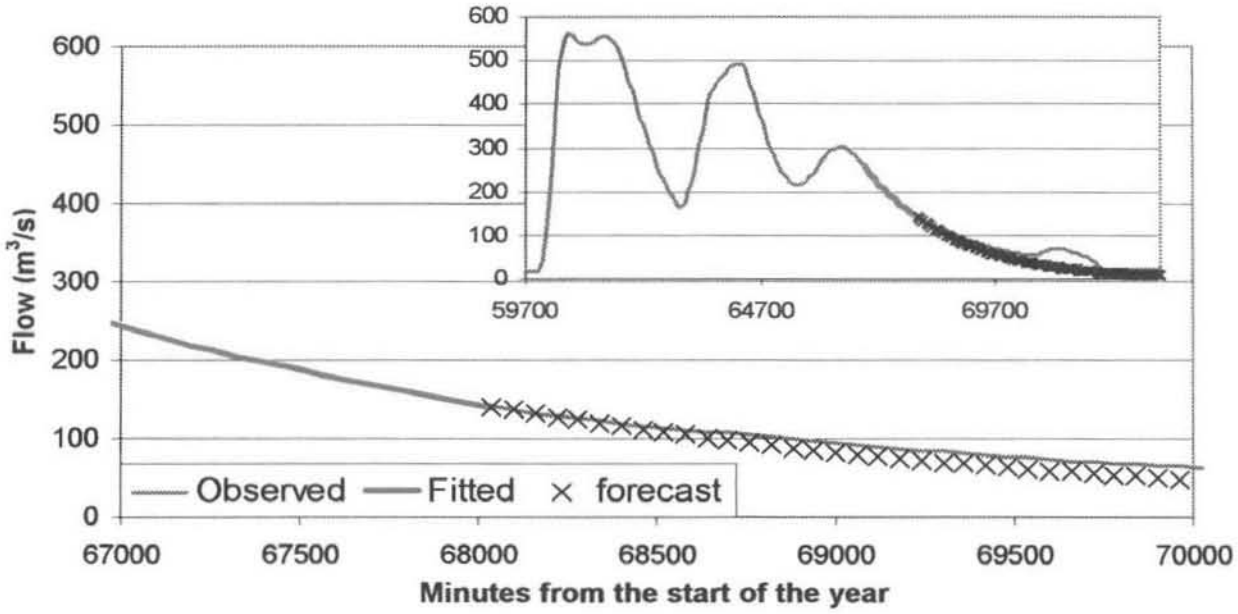


Figure 5.39: Forecast deviation from observed streamflow values

It is clear from figure 5.37 that there is significant temporal variation in the parameter values, for this event. The lead time for which we can expect a fairly accurate forecast also varies (Figures 5.38 and 5.39). When we are not expecting any more rain a confident forecast of the (linear) recession can be made for several hours ahead. However, if more rainfall does occur the forecast based on current knowledge cannot be made with great confidence.

A number of options are available to improve the lead time and confidence in forecasts. Firstly conditional forecasts of the expected rainfall can be made using a rainfall simulation model of some kind. The lead time during which we can be confident in the rainfall forecasts will allow an increase in the lead time for forecasting of streamflow using the current parameter set. Another option is incremental updating of the forecast, using recursive estimation techniques (such as the Kalman filter) to improve the forecast at each timestep. This requires obtaining the most current observations of streamflow and comparing them to the equivalent forecast values, the model parameters or states can then be updated to produce the optimal forecast.

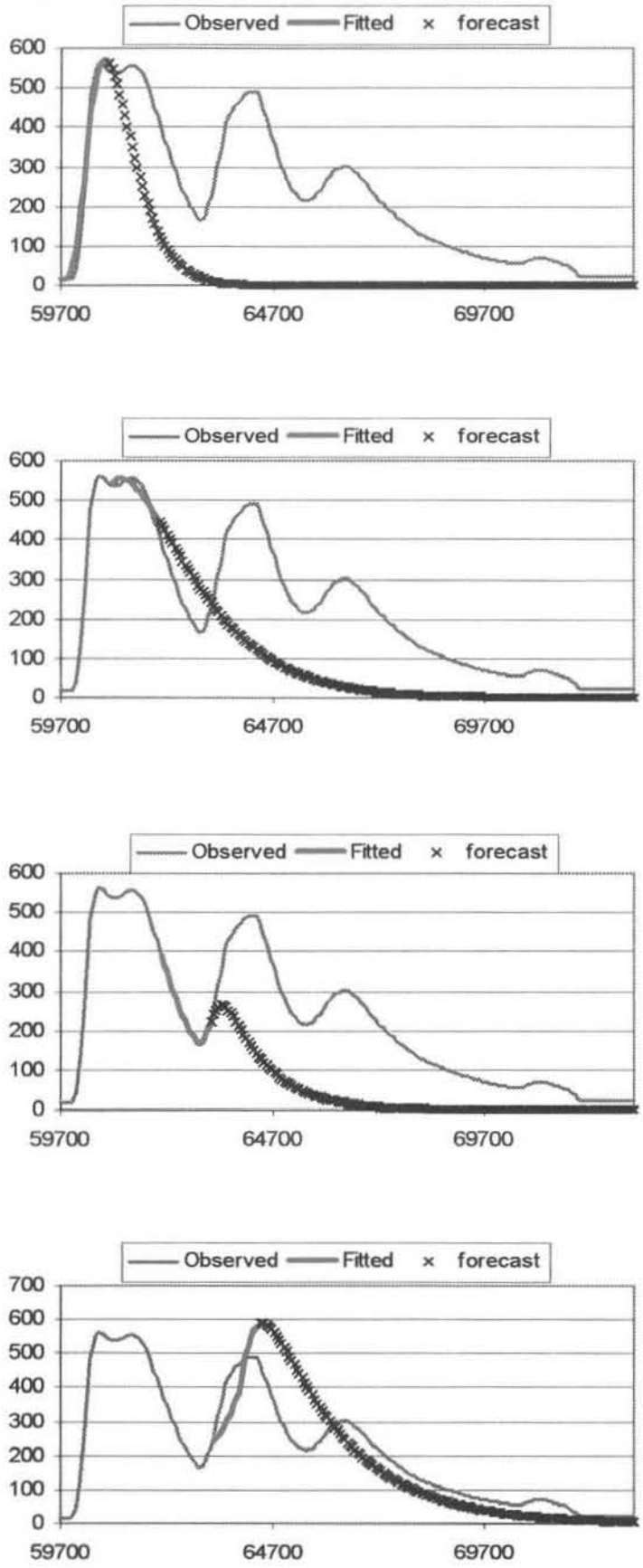


Figure 5.40: Conditional forecasts made with no future knowledge

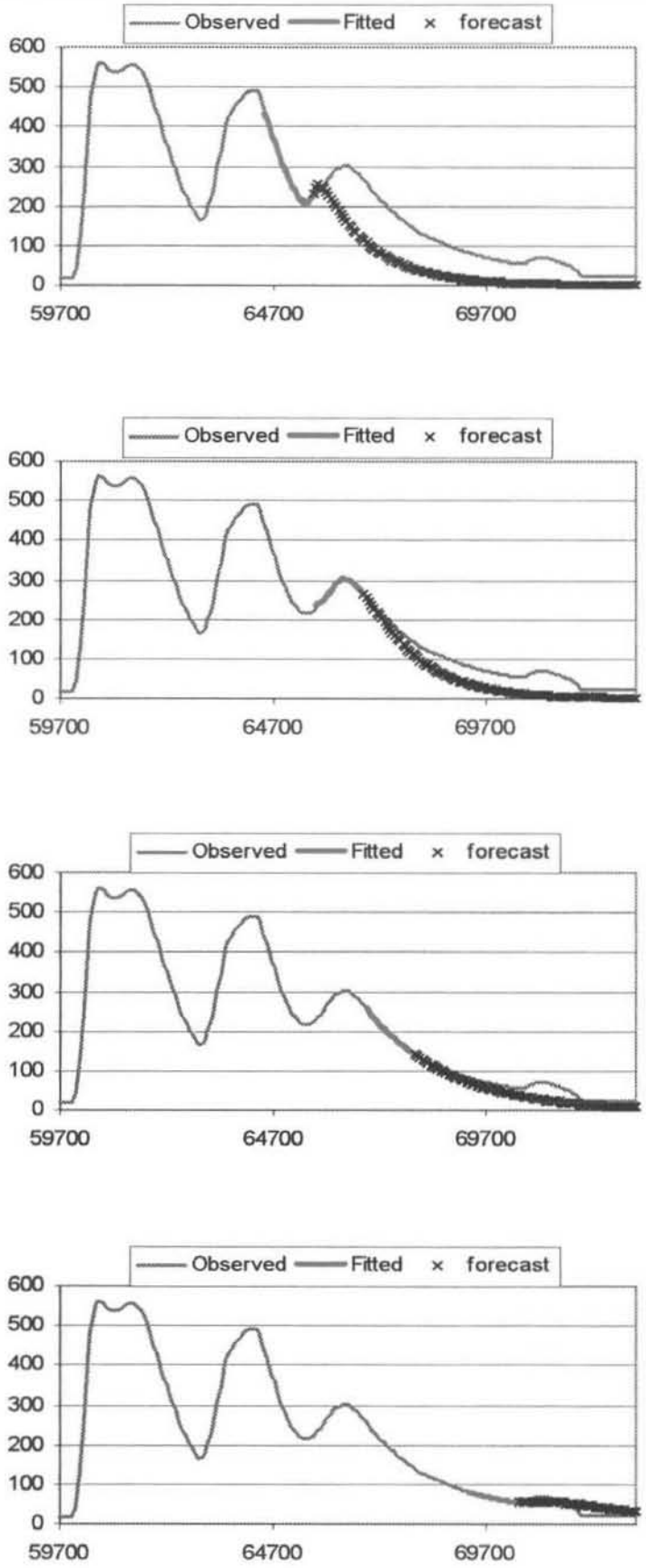


Figure 5.41: Conditional forecasts made with no future knowledge

5.4 Model Validation

Validation of the fitted model parameters, on data that wasn't used in the calibration procedure, could not be carried out. The reason for this is that a single optimal parameter set could not be identified from the events analysed. Identification of such parameter set was made difficult by the unavailability of consistent rainfall data. For example, there is no radar data available for the later part of the event in February 1996, as the radar went down. Rainfall data from the logging raingauges were used in fitting this event. Conversely, for the event occurring during November and December 1998, only radar data were obtained. The use of two different input data sources cannot produce a sensible single parameter set. Another difficulty encountered was the conversion of the radar data storage format to the current MDV format, from late 1998 onwards. Rain rates stored using the old storage format do not seem to match those found using the MDV format, another source of difficulty in calibrating the model. The current consistent use of a single archiving format will hopefully ensure that calibration using later data sets will produce a more sensible result. The fact that the model fits are reasonably close to the observed streamflow record is encouraging. This indicates that the model has great flexibility. A further point to note is that since there is evidence of temporal variation in the parameters (which is expected due to the continually varying moisture state of the catchment) it is more beneficial to make use of recursive techniques for producing the optimal forecast. This allows for updating of the forecasts based on previous observations and current inputs.

CHAPTER 5 RESULTS

Summary:

The fitting procedure described in Chapter 3 has been used to recover the parameters from synthetically derived streamflow sequences. These sequences were produced with the model in a number of configurations. The fitted parameter sets have been shown to match the chosen parameters almost exactly, as expected. The Monte Carlo approach to parameter fitting has allowed insight into the behaviour of the parameters within the bounded parameter space. Three historical events have been used to calibrate the model parameters. Good fits, in terms of the least squares criterion, have been achieved. However, discrepancies in the data record have not allowed a single optimum parameter set to be identified. Validation of the model has not been adequately carried out.

The application of the model to a new catchment will naturally require a process of calibration. A calibration procedure for a gauged catchment is suggested. The factors to be considered include a decision on whether to sub-divide the catchment or treat it as a single lumped cell. The availability of radar data is also an important factor. It's accuracy in reflecting the actual rain rates occurring is less important provided there is a defined linear relationship (not necessarily known) between the rain rates occurring and the rates reflected in the CAPPI data. The model parameters should still be able to convert this pseudo-rain rate into the observed streamflow response, as long as the radar is recording when it's raining or not and if the relationship is reasonably linear.

6.1 DATA REQUIREMENTS

The presented model is only applicable to gauged catchments, since the calibration procedure to fit the model parameters requires some historical data. The advantage of using radar to establish the precipitation input is that the only gauging requirement is a streamgauge. If there are existing raingauges on the catchment then they can be used to supplement the rainfall measurements obtained from the radar. The great advantage of using radar is the spatial and temporal resolution that can be achieved (Chapter 6). This allows the spatial distribution of rainfall discussed in Chapter 7 to be incorporated into the model through catchment sub-division.

The requirements for application of the model to a new catchment are

- *A telemetering streamgauge is available.* This is necessary so that a data set for calibration of the model is provided. Real time forecasting utilizes online updating of streamflow estimates and will require up to date measurements of streamflow.
- *There is coverage of the catchment by weather radar.* To provide the spatial and temporal rainfall input to the catchment.
- *Telemetering raingauges would be advantageous.* This is not a strict requirement but additional input data is always helpful, especially if the radar fails for some reason. Some raingauges are needed to validate the radar accumulations, but data logging gauges are suitable for this.

6.2 A Suggested Calibration Procedure

Having established that the catchment meets the necessary requirements for application of the model to flood forecasting, the following calibration procedure is suggested

- *Extract a representative historical streamflow sequence at regular discrete timesteps.* The timestep will be determined by the temporal resolution of available precipitation data and the expected time-scale of the catchment response. Depending on how the flow rate is measured and subsequently archived; some interpolation of streamflow values may be necessary to get flow data in the required format.
- *Determine the contributing precipitation inputs producing the streamflow.* The catchment and sub-catchment boundaries will have to be established and used to mask the radar data. Accumulated rainfall over discrete intervals will then need to be calculated. If rain gauge data is available for the calibration data set, and the subsequent forecasting operation, then a suitable merged rainfield may be determined.
- *Examine the input and output data sets for consistency.* Before embarking on the model calibration procedure, it is essential that some quality control be carried out on the data sets to minimize the effect of inconsistencies. Good calibration and verification of the model cannot be achieved with poor data sets.
- *Fit model parameters to observed data sets.* Using the parameter fitting procedure described in Chapter 3 (or a suitable alternative) find the model configuration and corresponding parameter set which produces the best fit between the model output and observed streamflow. Treat the catchment as a single cell initially.
- *Verification of the parameter set.* Use the fitted model configuration and parameter set to produce flow estimates from historical data not used in the determination of the parameters. Establish the "goodness" of the model fit to the observed data. If a satisfactory fit is achieved then the parameter set may be adopted for forecasting. If a satisfactory fit cannot be achieved, try fitting the parameters using more than one cell.

6.3 Operational Application of the Model

The purpose in developing this model has been to apply it in a flood forecasting application. Although the operational use of the model has not been fully tested, the following is envisaged.

- *Calibration of the model to the particular catchment.* This involves carrying out the calibration procedure described in section 6.2 above, to fit representative parameters for the model.
- *Choose a starting point on the recently observed streamflow record.* Select stable starting values for the model to ensure that the forecasting process starts suitably close to some observed streamflows.
- *Forecast future flows using the current rainfall inputs.* Use the parameters found during the calibration procedure to forecast future flows using the difference equation form of the linear reservoir cell model. The latest rainfall inputs are obtained online from radar or telemetering raingauges.
- *Apply an online correction procedure.* To ensure that there is a maintained correspondence between the forecast flows and the observed streamflow readings, as these become available, some kind of fast correction procedure will need to be applied. This may be as simple as replacing the forecast flows with the actually observed values at each timestep, ensuring that the forecast is proceeding from a feasible position. An online parameter updating procedure may also be employed. Such a parameter updating procedure would make use of a computationally efficient optimization algorithm to recompute an optimal parameter vector at each timestep. The parameter vector can be computed based on a fixed number of previous data. Each new optimization would be carried out using virtually the same information and good starting values, ensuring that computational load is not too great. Conditional forecasts based on the current catchment state can then be made.

- *Use the forecast flow values as a decision making tool.* The forecasts will provide an indication as to whether or not flooding can be expected. Projected flows above a certain critical magnitude can be automatically reported to the relevant disaster management authorities.

6.4 CONCLUSIONS

This study has presented a generalized linear reservoir model module. The model can be conceptualized in terms of a surface, sub-surface and channel storage, each of which is represented by a linear storage element incorporating a linear loss term. The discrete observed rainfall depths, during a given time period, are inputs to the model. The inputs enter the first reservoir and feed forward into the remaining reservoirs, the output is the outflow from the final reservoir.

Selecting extreme parameter values can create "open" (instantaneous transfer) or "closed" (no transfer) links between the various reservoirs. By varying which links are open or closed, the model can be collapsed to a number of reduced forms. This flexibility allows the model to operate in a number of permutations ranging from a single linear reservoir through cascades, or parallel arrangements, to the general model form. All of these may include various combinations of loss terms.

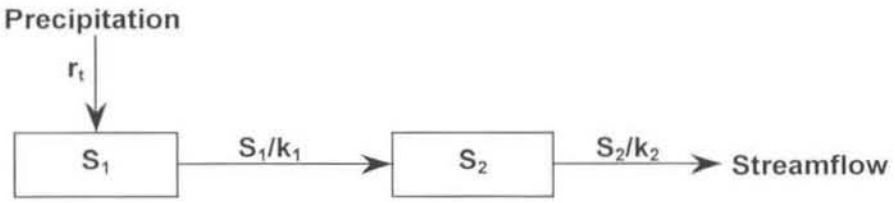
The model is a semi-distributed multi-cell type, with the study catchment being divided into a number of cells and the model outputs from each cell summed at the catchment outlet to give the total streamflow from the catchment (Chapter 4). Parameter fitting using an automatic optimization algorithm (Chapter 3), produced encouraging correspondence between the observed and fitted streamflow records for three rainfall-runoff events of up to nineteen days in length (Chapter 5).

The model has been formulated as a difference equation in ARMA form. It has a defined relationship between the reservoir response parameters and the pseudo-ARMA coefficients, which guarantees stationarity (given non-negative inputs). In this form, the model is highly computationally efficient. The input to the model comes from high-resolution space-time radar data, further enhancing its potential as a real time flood forecasting model.

The model formulation is unconditionally stable, allowing the parameter values to be chosen independently of each other. However, instability of the model, in pseudo-ARMA form, may result from a poor choice of initial flow values (Chapter 3). This issue will need to be fully addressed before operational use can be made of the model, however, a sensible starting point for each sub-event is the upturn of the hydrograph. The full state-space representation of the model does not produce any instability, provided the initial storages are positive. The ability of the model to represent many conceptual reservoir arrangements makes it a useful tool.

APPENDIX A

STABLE STARTING VALUES FOR THE TWO RESERVOIR CASCADE



For a zero input situation, i.e. $r_t = 0$ for all t, the pseudo-ARMA form reduces to

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} \tag{A1}$$

$$y_{t+1} = \phi_1 y_t + \phi_2 y_{t-1} \tag{A2}$$

where

$$\phi_1 = q_1 + q_2$$

$$\phi_2 = -q_1 q_2$$

Equations A1 and A2 can be written in the following matrix form

$$\begin{bmatrix} y_t \\ y_{t+1} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \phi_2 & \phi_1 \end{bmatrix} \begin{bmatrix} y_{t-1} \\ y_t \end{bmatrix} \tag{A3}$$

equivalently

$$\bar{y}^{t+1} = A\bar{y}^t$$

with

$$\bar{y}^t = \begin{bmatrix} y_{t-1} \\ y_t \end{bmatrix}$$

and A a 2x2 square matrix. Considering a sequence of equation A3 it follows that we can write

$$\bar{y}^{t+n} = A^n \bar{y}^t \tag{A4}$$

The canonical decomposition of A is

$$A = UDU^{-1}$$

where D is the diagonal matrix of eigenvalues and U contains the corresponding eigenvectors.

Equation A4 becomes

$$\bar{y}^{t+n} = UD^n U^{-1}\bar{y}^t \tag{A5}$$

The eigenvalues λ_i are the solutions to $\det(A-\lambda I)=0$ and are $q_i = e^{-\frac{1}{k_i}\Delta t}$

Expanding equation A5

$$\begin{bmatrix} y_{t+n-1} \\ y_{t+n} \end{bmatrix} = \frac{1}{q_1 - q_2}\begin{bmatrix} 1 & 1 \\ q_2 & q_1 \end{bmatrix}\begin{bmatrix} {q_2}^n & \\ & {q_1}^n \end{bmatrix}\begin{bmatrix} q_1 & -1 \\ -q_2 & 1 \end{bmatrix}\begin{bmatrix} y_{t-1} \\ y_t \end{bmatrix}$$

$$\begin{bmatrix} y_{t+n-1} \\ y_{t+n} \end{bmatrix} = \frac{1}{q_1 - q_2}\begin{bmatrix} {q_2}^n & {q_1}^n \\ {q_2}^{n+1} & {q_1}^{n+1} \end{bmatrix}\begin{bmatrix} q_1 y_{t-1} - y_t \\ -q_2 y_{t-1} + y_t \end{bmatrix} \tag{A6}$$

From equation A6, we find that

$$y_{t+n} = \frac{1}{q_1 - q_2}\left(q_1{q_2}^{n+1} y_{t-1} - {q_2}^{n+1} y_t - {q_1}^{n+1} q_2 y_{t-1} + {q_1}^{n+1} y_t\right)$$

$$\frac{y_{t+n}}{y_{t-1}} = \frac{1}{q_1 - q_2}\left[q_1 q_2\left({q_2}^n - {q_1}^n\right) + \left({q_1}^{n+1} - {q_2}^{n+1}\right)\frac{y_t}{y_{t-1}}\right] \tag{A7}$$

The conditions to guarantee A7 non-negative are

$$\frac{y_t}{y_{t-1}} > \frac{q_1 q_2\left({q_1}^n - {q_2}^n\right)}{{q_1}^{n+1} - {q_2}^{n+1}} \tag{A8}$$

and

$$q_1 > q_2$$

Noting that

$$\frac{a^{n+1} - b^{n+1}}{a-b} = a^n + a^{n-1}b + \ldots + ab^{n-1} + b^n$$

equation A8 becomes

$$\frac{y_t}{y_{t-1}} > \frac{q_1 q_2 (q_1 - q_2)\left({q_1}^{n-1} + {q_1}^{n-2} q_2 + \ldots + q_1 {q_2}^{n-2} - {q_2}^{n-1}\right)}{(q_1 - q_2)\left({q_1}^n + {q_1}^{n-1} q_2 + \ldots + q_1 {q_2}^{n-1} - {q_2}^n\right)}$$

$$\frac{y_t}{y_{t-1}} > \frac{q_1\left({q_1}^{n-1} q_2 + \ldots + q_1 {q_2}^{n-1} - {q_2}^n\right)}{\left({q_1}^n + {q_1}^{n-1} q_2 + \ldots + q_1 {q_2}^{n-1} - {q_2}^n\right)}$$

$$\frac{y_t}{y_{t-1}} > \frac{q_1}{\left(1+\frac{1}{\alpha}{q_1}^n\right)}$$

where

$$\alpha = {q_1}^{n-1}q_2 + \ldots + q_1{q_2}^{n-1} - {q_2}^n$$

For large n, $\alpha >> {q_1}^n$ therefore ensure $\boxed{\frac{y_t}{y_{t-1}} > q_1}$

APPENDIX B

Much of the data analysis and model fitting for this study required specific software to be written. The remaining tasks were performed using a commercially available spreadsheet package and image editing software. This appendix provides a more detailed description of the various C programs mentioned throughout this text.

B.1 RAINGAUGE ACCUMULATION

The tipping bucket raingauge data, provided by METSYS, were in the format illustrated in Figure B1. The first column of data contains the gauge identifier. The next six columns of two digit integer numbers represent, the year, month, day, hour, minute and second at which the reading was taken. The next column (accurate to two decimal places) is the rainfall depth represented by each tip; always 0.2 mm. Column 9 shows the number of days so far in the current year. The final column is a tip count.

stn	yy	mm	dd	hh	mm	ss	r.rr	yd	count
L001	96	03	01	14	16	13	0.00	060	00000
L001	96	03	01	23	27	45	0.20	060	00002
L001	96	03	01	23	32	51	0.20	060	00003
L001	96	03	02	00	47	54	0.20	061	00004
L001	96	03	02	00	53	10	0.20	061	00005

Figure B1: Example of tipping bucket gauge data

Each data file stores an entire month's data, with the data for each of the 45 gauges following consecutively. Every gauge makes at least one reading per day to confirm that the gauge was operational each day of the month. This format is not useful as an input sequence to the model. A computer program was written to correctly format the data so that it could be used as input to the model. The average rainfall over the catchment, or sub-catchment, was output in the required format, shown in Figure B2.

Year	Time	Depth
1996	36420	1.75
1996	36480	0.40
1996	36540	0.00
1996	36600	0.05
1996	36660	0.15

Figure B2: Example of correctly formatted output from the raingauge accumulation program

APPENDIX B

Each record in the formatted output file contains, the year, the time in minutes from the start of that year and the accumulated depth (mm) of rainfall during the one hour period starting at the time indicated.

The algorithm used to format the data is

- Read first line of data.
- Determine the time from the start of the year (in minutes) to the end of the first and last hours in the current month.
- Allocate a memory array, large enough to hold the hourly rainfall accumulations for each gauge, every hour of the month.
- While the end of the file has not been reached
    - While the gauge identifier is the same as the current one
        - Read the next line of data.
        - Calculate the time from the start of the year.
        - If the time is less than, or equal to, the end of the current hour; add the tip depth to the current hours running total.
        - Else If the time is greater than the end of the current hour; write the running total to the appropriate place in the memory array, initialize the running total to zero and increment the current hour.
    - If the end of the month has been reached; initialize the current hour to the start of the month.
    - Else; While the end of the month has not been reached.
        - Infill the memory blocks with zeros.
- Calculate the average precipitation depth for the relevant gauges for each hour of the month and write the results to a formatted text file.

B.2 STREAMFLOW INTERPOLATION

The breakpoint streamflow data obtained from DWAF are not formatted for easy comparison with the model output. It was necessary to write a conversion program which would extract the relevant flow rates and interpolate between them where required. A linear interpolation was used between points which did not coincide with the correct discrete intervals. This was done for simplicity, and because the breakpoints are (by definition) the points where the continuous flow record deviates from a straight line.

The algorithm for this program is as follows

- While the end of file has not been reached
- Read a line of data.
- Calculate the time (in minutes) from the start of the year.
- Write the year, time and flow rate to an intermediate file.
- Close the primary data file.
- Read the first line of the intermediate file.
- Determine the time to the start of the hour after the first record.
- While it is not the end of the intermediate file
- Read the next line of data.
- If the time is after the current hour.
    - Linearly interpolate between the previous and current flows to calculate the flow for this hour.
    - Write the flow and current hour to file.
    - Replace the previous flow with the current one.
    - Increment the current hour
- Else If the time is prior to the current hour; update previous flow.

B.3 CAPPI MASKING

The CAPPI masking program accepts as input the file path to the first CAPPI and the number of consecutive CAPPI's to be processed. In addition, input is the path to the first mask image and the number of consecutive images with which the CAPPI data must be masked. The output is a number of text files, containing hourly rainfall accumulations, formatted as shown in Figure B2. The original bitmap images remain unchanged.

The algorithm used is

- Open the first CAPPI bitmap.
- Determine the time (in minutes) from the start of the year to the end of the hour during which the CAPPI data was recorded.
- Allocate a memory array large enough to hold the rain rate for each pixel on the CAPPI.
- Open the mask image.
- Define a "mask" for the memory array.
- Write the rain rates, for pixels falling within the masked region, to the appropriate place in the memory array.
- While there are still more CAPPI's to process.
- Open the next CAPPI.
- If the time is during the current hour
    - Increment the relevant values in the memory array by the new rain rates.
    - Increment the CAPPI count
- Else If the time is after the current hour
    - Calculate the rainfall depth during the current hour by dividing the values in the memory array by the number of CAPPI's recorded during the hour.
    - Initialize the memory array.
    - Initialize the CAPPI count.

- Increment the current hour.
- Increment the relevant values in the memory array by the new rain rates.
- Increment the CAPPI count

- Repeat the above for each mask region.

B.4 Parameter Fitting

The flow chart in Chapter 8 (Figure 8.2) provides an adequate description of the algorithm used in the parameter fitting process. The following pages contain an example of the information file used by the fitting program. All lines beginning with the hash (#) symbol are comments and ignored by the program. A separate copy of this file was used for fits of the various model configurations, with the appropriate changes made.

```
# A typical information file for the parameter fitting process.
# Nov/Dec-1998   Model 4     Six cells
# Above are comments on the event date and number of cells in the
# catchment, as well as the model configuration used.


# Channel parameters for each cell.
# These are calculated as the ratio CL/TL.
# Where CL is the distance along the longest collector of the cell
# to the gauging station.
# TL is the longest of the distances calculated in this way.

channelC83A             1.0000
channelC83B             0.9123
channelC83C             0.6491
channelC83D             0.9386
channelC83E             0.5351
channelC83F             0.4035

# Areal parameters for the cells.
# The parameters are the area of the cell taken as a proportion of
# the total catchment area.

areaC83A                0.2054
areaC83B                0.0713
areaC83C                0.2290
areaC83D                0.1294
areaC83E                0.118
areaC83F                0.2464

# Conversion factors from mm/hr to m^3/s.
# The rainfall totals are in mm/hr but streamflow is in m^3/s.

conversionC83A          338.6111
conversionC83B          117.5
conversionC83C          377.5
conversionC83D          213.3333
conversionC83E          195.2777
conversionC83F          406.1111

# Path to text file of observed flows.

OBSERVED_FLOW           \\FLOWMASTER\process\flowsC8H020.txt

# Name of working directory to create.

WORKING_DIR             \\FLOWMASTER\process\median40

# Paths to save calculated flows and parameters.

CALCULATED_PARAMETERS \\FLOWMASTER\process\median40\parameters.txt
CALCULATED_FLOWS      \\FLOWMASTER\process\median40\streamflow.txt
```

```
# Paths to save intermediate flows and parameters.
PARAMETER_ESTIMATE    \\FLOWMASTER\process\median40\pA.txt
STREAMFLOW_ESTIMATE   \\FLOWMASTER\process\median40\sA.txt

# Year in which to start fitting.
START_YEAR            1998
# Minutes from the start of the year to the first record.
START_MINUTES         452340

# Length of record to analyse (Hours).
SIZE                  636

# Path to first text file of observed rainfall.
# The files must be named in the following format
# C83A.txt,C83B.txt....C83F.txt
RECORDED_RAIN
\\FLOWMASTER\Accumulated_Rain_Data\11_12_1998_C83A.txt

# The number of cell's in the catchment.
CELL_NUMBER           1

# Determines where the losses are taken from in the model.
# A "comment" parameter to keep track of the model configuration.
# 1->Loss from tank 1
# 2->Loss from tank 2
# 3->Loss from tank 3
# 4->Loss from tanks 1&2
# 5->Loss from tanks 1&3
# 6->Loss from tanks 2&3
# 7->Losses from all tanks
MODEL_NUMBER          4
```

```
# The tank response parameters (Hours)
# Zero or negative values indicate that the parameter is to be
# fitted.
# Positive values indicate the value at which the parameter is
# fixed. K = 10' if the reservoir is "closed".
K1                      -1
K2                      -1
K3                      -1
K4                      -1
K5                      -1
K6                      -1
K7                      1000000

# The minimum lag (Hours).
START_LAG               5

# The maximum lag (Hours).
END_LAG                 8

# Initial temperature for the Metropolis annealing schedule.
# Defines a scale for the random fluctuations, which is reduced
# after a given number of iterations.
INITIAL_TEMP            10000

# The number of iterations before the Metropolis temperature is
reduced.
ITERATIONS              2000

# The maximum number of iterations allowed in total
# (for each restart).
MAXIMUM_ITERATIONS      20000

# The number of restarts from random simplexes.
RESTARTS                25

# The choice of annealing schedule to reduce the Metropolis
# temperature.
# Schedule 1 reduces temp. exponentially.
# Schedule 2 reduces temp. linearly.
# Schedule 3 reduces temp. based on the difference between the
# lowest objective value on the current simplex and the lowest
# found at previous temperatures.
ANNEALING_SCHEDULE      1
```

```
# Value of exponent for schedule 1.
ALPHA                   2.43

# Linear temperature reduction factor for schedule 2.
EPSILON                 1.0E-3

# Factor of order one, to scale the difference in values at
# simplex vertices, Schedule 3.
BETA                    2

# Biggest factor by which the temperature may be reduced during
# one step.Schedule 3.
GAMMA                   7.5E-1

# Stopping tolerance for the
# amebsa algorithm (Press et al., 1992).
FUNCTION_TOLERANCE      1.0e-8

# The upper bound for the reservoir response parameters.
# This value allows the response time to be essentially infinite.
KMAX                    10000

# The lower bound for the reservoir response parameters.
# This value allows the response time to be negligible.
KMIN                    0.001
```

AFORISM (1996), A Comprehensive Forecasting System for Flood Risk Mitigation and Control, *Final Report to the Commission of European Communities*, Research Contract No. EPOC-CT90-0023

BEVEN K. (1989), Changing ideas in hydrology – The case of Physically-based models, *J. Hydrology*, **105**, 157 - 172

BINLEY A., ELGY J. and BEVEN K. (1989), A Physically Based Model of Heterogeneous Hillslopes : 1. Runoff Production, *Water Resources Research*, **25**(6), 1219 - 1226

BOX G. E. P. and JENKINS G. M. (1970), Time series analysis forecasting and control, *Holden-Day*, San Fransisco, California

BRAS (1990), Hydrology: An Introduction to Hydrologic Science, *Addison-Wesley*

CHOW V. T. (1964), Handbook of Applied Hydrology, *McGraw-Hill*

CHOW V. T., MAIDMENT D. R. and MAYS L. W. (1988), Applied Hydrology, *McGraw-Hill*

DARCY H. (1856), Les fontaines publiques de la ville de Dijon

DISKIN M. H. and PEGRAM G. G. S. (1987), A Study of Cell Models: 3. A Pilot Study on the Calibration of Manifold Cell Models in the Time Domain and the Laplace Domain, *Water Resources Research*, **23**(4), 663 - 673

DOOGE J. C. I. (1959), A general theory of the unit hydrograph, *Journal of Geophysical Research*, **64**(2), 241-256

DOOGE J. C. I. (1973), Linear Theory of Hydrologic Systems, *Agricultural Research service.*, US Dept. of Agriculture, Technical Bulletin No. 1468

FIERING MYRON B. (1967), Streamflow Synthesis, *Macmillan*

FRALEIGH J. B. and BEAUREGARD R. A. (1990), Linear Algebra, *Addison-Wesley*

FRANCHINI M., WENDLING J., OBLED C. H. and TODINI E. (1996), Physical interpretation and sensitivity analysis of the TOPMODEL, *Journal of Hydrology*, **175**

GEOGAKAKOS K. P. (1987), Real-time flash flood prediction, *Journal of Geophysical Research*, **92**, 9615 – 9629

GREEN W. H. and AMPT G. A. (1911), Studies on soil physics: 1. Flow of air and water through soils, *Journal of Agricultural Science*, **4**, 1 – 24

HILL P., MEIN R. and SIRIWARDENA L. (1998), How much rainfall becomes runoff ? Loss modelling for flood estimation, *Cooperative Research Centre for Catchment Hydrology*, **Industry report** 98/5

HORTON R. E. (1940), An approach toward a physical interpretation of infiltration capacity, *Soil Sci. Soc. Am. J.*, **5**, 399 – 417

JAKEMAN A. J. and HORNBERGER G. M. (1993), How much complexity is warranted in a Rainfall-Runoff model ?, *Water Resources Research*, **29**(8), 2637 – 2649

KIRKPATRICK S., GELATT C. D. and VECCHI M. P. (1983), Optimization by Simulated Annealing, *Science*, **220**(4598), 671-679

LAURENSON E. M. (1964), A catchment storage model for runoff routing, *Journal of Hydrology*, **2**(2), 141-163

KNUTH D. E. (1981), The Art of Computer Programming, *vol. 2 Semi-Numerical Algorithms*, Addison-Wesley

KUCZERA G. and PARENT E. (1998), Monte Carlo assessment of parameter uncertainty in conceptual models: the Metropolis algorithm, *Journal of Hydrology*, **211**, 69-85

MARSHALL J. S. and PALMER W. M. (1948), The distribution of raindrops with size, *Journal Meteorology*, **5**, 165-166

MATHER G. K., TERBLANCHE D. E. and STEFFENS F. E. (1997), National Precipitation Research Programme Final Report for the period 1993-1996, *WRC Report 726/1/97*

McCARTHY G. T. (1938), The unit hydrograph and flood routing, *Conf. North Atlantic Div.*, U.S Corps of Engineers, New London, Conn.

METROPOLIS N., ROSENBLUTH A. W., ROSENBLUTH M. N. and TELLER A. H. (1953), Equation of state calculation by fast computing machines, *Journal of Chemical Physics*, **21**(6), 1087-1092

MIDGLEY D. C., PITMAN W. V. and MIDDLETON B. J. (1994), Surface Water Resources of South Africa 1990, *WRC report 298/94*

NASH J. E. (1957), The form of the instantaneous unit hydrograph, *IASH publication no. 45*, **vol. 3-4**, 114-121

NEH-4: NATIONAL ENGINEERING HANDBOOK, (1972), Section 4, Hydrology, *United States Department of Agriculture*, Soil Conservation Service, Washington D.C.

NELDER J. A. and MEAD R. (1965), A Simplex method for function minimization, *Computer Journal*, **7**, 308-313

O'CONNOR K. M. (1976), A discrete linear cascade model for hydrology, *Journal of Hydrology*, **29**, 203-242

O'CONNOR K. M. (1982), Derivation of discretely coincident forms of continuous linear time-invariant models using the transfer function approach, *Journal of Hydrology*, **59**, 1-48

PARK S. K. and MILLER K. W. (1988), Random Number Generators: Good ones are hard to find, *Communications of the ACM*, **31**, 1192-1201

PEGRAM G. G. S. (1980), A continuous streamflow model, *Journal of Hydrology*, **47**, 65-89

PEGRAM G. G. S. and DISKIN M. H. (1987), A Study of Cell Models : 1. A Manifold Cell Model for Distributed Surface Runoff Systems, *Water Resources Research*, **23**(4), 646 - 654

PEGRAM G. G. S. and DISKIN M. H. (1987), A Study of Cell Models : 2. The Effect of Time Delay on the Limiting Forms of Cascade and Manifold Cell Model Response Functions, *Water Resources Research*, **23**(4), 655 – 662

PEGRAM G. and CLOTHIER A. (1999), High Resolution Space-Time Modelling of Rainfall: The "String of Beads" Model, *WRC report 752/1/99*, 62-64, Water Research Commission, Pretoria

PHILIP J. R. (1957), The theory of infiltration:1. The infiltration equation and it's solution, *Soil Science*, **83**, 345 – 357

PRESS W. H., TEUKOLSKY S. A., VETTERLING W. T. and FLANNERY B. P. (1992), Numerical Recipes in C – The art of scientific computing, *Second Edition*, Cambridge University Press

QUIMPO R. G. (1967), Stochastic model of daily river flow sequences, *Colorado State University*, Fort Collins Colorado, Hydrology Paper 18

QUIMPO R. G. (1971), Structural relation between parametric and stochastic hydrology models, *Proc. Int. Symp. Math models Hydrol.*, Warsaw, 301-312

RICHARD L. A. (1931), Capillary conduction of liquids in porous mediums, *Physics*, **1**, 318 – 333

RODRIGUEZ–ITURBE I. and VALDES (1979), The geomorphologic structure of the hydrologic response, *Water Resources Research*, **15**, 1409 - 1420

SHAMSELDIN A. Y. and NASH J. E. (1998), The geomorphological unit hydrograph – a critical review, *Hydrology and Earth System Sciences*, **2**(1), 1 – 8

SHERMAN L. K. (1932), Streamflow from rainfall by the unit-graph method, *Eng. News rec.*, 108, 501 – 505

SCHMIDT E. J. and SCHULZE R. E. (1987), Flood volume and peak discharge from small catchments in Southern Africa, based on the SCS technique, *WRC Project No. 155*, TT 31/87

SPOLIA S. K. and CHANDER S. (1974), Modelling of surface runoff systems by an ARMA model, *Journal of Hydrology*, **22**, 317-332

TODINI E. and BOSSI A. (1986), PAB (Parabolic and Backwater) an unconditionally stable flood routing scheme particularly suited for real time forecasting and control, *Journal of Hydraulic Research*, **24**, 405 - 424

TODINI E. (1996), The ARNO Rainfall-Runoff model, *Journal of Hydrology*, **175**

ZOCH R. T. (1934), On the relation between rainfall and runoff, *Monthly Weather Review*, **62**, 315 - 322