User Manual for SIBERIA
(Version 8.30)

Prof. Garry Willgoose
Telluric Research,
100 Barton Street,
Scone, NSW, 2337
AUSTRALIA.
Email: g.r.willgoose@leeds.ac.uk

1 Introduction

This manual describes how to use the catchment evolution model SIBERIA developed by the author, beginning in 1986. SIBERIA is a computer model for simulating the evolution of landscapes under the action of runoff and erosion over long times scales (typically more than a few years). SIBERIA is both a very simple model and a very sophisticated one. The hydrology and erosion models are based on ones that are simple and widely accepted in the hydrology and agricultural communities since the 1960's. These models are based on widely accepted physics and have been successfully calibrated in a range of environments. The sophistication of SIBERIA lies in (1) its use of digital terrain maps for the determination of drainage areas and geomorphology and (2) its ability to efficiently adjust the landform with time in response to the erosion that occurs on it.

The basic theory underlying the model and the approximations required for computer solution are described in this manual. The parameters that the models uses are described together with the process involved in running SIBERIA. The standard file formats used for input to the model and for transfer to other data analysis and visualisation packages are described and sample code is provided for input of these files. Finally, details are provided on the standard procedures that are available in the model to extend various components of the model if the standard ones should fail to meet the purposes of the user.

For further information about the details of the theory underlying SIBERIA or results from the model the publications in the references should be consulted.

1.1 A Warning to US Users on Model Units

When the physical units option (ModeErode≥20 and ModeRunoff≥20) is selected for SIBERIA the units assumed are SI. Briefly:

- Vertical elevation and horizontal distance units: metres
- Sediment discharge units: cubic metres of sediment/metre width/timestep (note that cubic metres here is the number of cubic metres of sediment that would consumed by deposited sediment – equivalent to bulk density - NOT the volume consumed by the sediment particles alone – equivalent particle density)
- Timestep units: Normally they are considered as years in EAMS but all flux parameters are defined per timestep
- Bulk Density units: metric tonnes/cubic metre
- Grid Spacing: metres

There are no options in SIBERIA to work in imperial units so that all problems need to be converted to metric units.

2 Background.

2.1 SIBERIA

What follows is a description of the philosophy and methodology used by SIBERIA. Greater detail can be found in Willgoose, et al. (1989, 1991a-d, 1994), Willgoose (1993, 1994a,b) and Willgoose and Riley (1993, 1998a,b).

The flood response of a catchment to rainfall is dependent on the geomorphic form of the catchment. But the catchment runoff not only responds to catchment form, it also shapes it through the erosion processes that act during runoff events. Over geologic time the catchment form, shaped by the range of erosion events, reflects the runoff processes that occur within it. The channel network form and extent reflect the characteristics of both the hillslope and channel processes. Hydrologists have long parameterised the influence of the geomorphology on flood response (e.g., Rodriguez-Iturbe and Valdes, 1979). Geomorphologists have largely fitted statistics to the landscape ignoring the historic processes that created the landscape (Strahler, 1964; Shreve, 1966) though there have been some notable exceptions to this generalisation (Gilbert, 1909; Horton, 1945). The difficulty of the problem is such that the number of researchers that have attempted to unify the geomorphology and the hydrology is small (Kirkby, 1971; Dunne, 1989; Huggett, 1988), even though the importance of both specialisations has long been recognised by geomorphologists:

> "to look upon the landscape ... without any recognition of the labor expended in producing it, or of the extraordinary adjustments of streams to structures and of waste to weather, is like visiting Rome in the ignorant belief that the Romans of today have no ancestors." (page 268, Davis, 1954)

The main stumbling blocks to the fulfillment of the promise of this scientific paradigm have been the range of temporal scales (geologic versus flood event timescales) and spatial scales (catchment and channel length scales) that are important in the problem; the heterogeneity in both space and time of the dominant processes; and the problem of the unification and observation of the processes acting at these disparate scales. Physically based computer models of catchment development (e.g., Ahnert, 1976; Kirkby, 1987) are important tools in the understanding of the interactions between hydrologic process and response, primarily because of their ability to explore the sensitivity of the system to changes in the physical conditions, without many of the difficulties of identification and generalisation associated with the heterogeneity encountered in field studies.

The ultimate goal is to develop a quantitative understanding of how channel networks and hillslopes evolve with time using a computer model of landscape evolution. Catchment form and hydrologic response will then be seen in the context of the complete history of erosion development of the catchment.

A large scale model of catchment evolution (SIBERIA) involving channel network growth and elevation evolution is documented below. This model integrates a model of erosion processes, theoretically and experimentally verified at small scales, with a physically based conceptualisation of the channel growth process. Neither the properties of the channel network nor the properties of the hillslopes can be viewed in isolation. They must be viewed as components of a complicated large scale non linear system: the drainage basin. The basic tenet of this work is that it is necessary to understand the physics of the catchment processes to be able to fully understand the catchment form and that it is necessary to " ... identify linked process equations and so define geomorphic systems in such a way that an analytical, predictive approach can be used ..." (p. 48, Huggett, 1988). It is not claimed, nor is it intended, that the model presented below account for all the processes occurring in the catchment. Rather a general model framework is presented which is both physically realistic and incorporates the dominant physical processes and which provides a useful tool for the study of the important interactions within the catchment. It is, however, believed to model the dominant processes occurring in fluvial landforms.

A crucial component of this model is that it explicitly incorporates the interaction between the hillslopes and the growing channel network based on physically observable mechanisms. An important, and explicit, differentiation between the processes that act on the hillslopes and in the channels is made. A point is defined to be a channel when selected flow and transport processes exceed a threshold value. If a function (called the channel initiation function) is greater than some predetermined threshold at a point, then the channel head advances to that point. The channel initiation function is primarily dependent on the discharge and the slope at that point, and the channel initiation threshold is dependent on the resistance of the catchment to channelisation. Channel growth is thus governed by the hillslope form and processes that occur upstream of the channel head. The channel initiation function is independent of Smith and Bretherton's (1972) definition of channels as points of instability in the flow equations. Nevertheless, these concepts are not necessarily contradictory. This is particularly true given the recent realisation that Smith and Bretherton's analysis would lead to a system of rills spaced at an "infinitesimal distance apart" unless a basic scale is built into the equations (Loewenherz-Lawrence, 1990). Introducing this scale of separation is conceptually consistent with the threshold analogy. The elevations on the hillslopes and the growing channels interact through the different transport processes in each regime and the preferred drainage to the channels that results. The interaction of these processes produces the long–term form of catchments. The preferential erosion in the channels results in the familiar pattern of hills and valleys with hillslope flow being towards the channel network in the bottoms of the valleys.

The model below simulates the growth and evolution of the channel networks and the contributing hillslopes. Two variables are solved for in the plane: the catchment elevation and an indicator variable that identifies where channels exist in space. In the computer implementation (computer code SIBERIA) a drainage direction is assigned to each node in the discretized space on the basis of the direction of steepest slope from node to node. These drainage directions are used to determine the area contributing to (i.e., flowing through) each node. From these areas, and thus discharge, and the steepest slopes at the nodes, continuity equations for flow and sediment transport are written. These areas and steepest slopes are also used to evaluate the channel initiation function (which may be, for example, overland flow velocity) which is then used in the channelisation function to determine regions of active channel network extension.

The governing differential equations for elevation and channel indicator functions are:

$$\frac{\partial z}{\partial t} = U + \frac{1}{\rho_b}\left(\frac{\partial q_{sx}}{\partial x} + \frac{\partial q_{sy}}{\partial y}\right) + D\left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2}\right) \qquad (2.1.1)$$

$$\frac{\partial Y}{\partial t} = f\left(d_t,\ Y,\ \frac{a}{a_t}\right) \qquad (2.1.2)$$

where in Equation (2.1.1) z is the elevation (positive upwards); x and y the horizontal directions; t time; U the tectonic uplift per unit time; ρ_b the bulk density of the soil (where the sediment flux is in units of mass/time); q_{sx} and q_{sy} the sediment transport in x and y directions; and D is the diffusivity. Equation (2.1.1) is simply a continuity equation for sediment transport. In Equation (2.1.2) Y is the variable describing whether that point in the catchment is a channel ($Y=1$) or a hillslope ($Y=0$), d_t is the rate of channel growth at a point, and a and a_t are the channel initiation function and its threshold respectively. Equation (2.1.2) describes the transition of a point in the catchment from hillslope to channel on the basis of a threshold in the channel initiation function a. The constitutive equations that are used for channel initiation function, a, and sediment transport, q_s, are those that apply in the region being modelled. Here we use (the details of these equations may be slightly different from the implementation depending on the input parameters for the computer simulations, see later sections for these details)

$$a = \beta_5 q^{m_5} S^{n_5} \qquad (2.1.3)$$

$$q_s = K q^{m_1} S^{n_1} \qquad (2.1.4)$$

$$K = G \cdot f(Y) \qquad (2.1.5)$$

$$f(Y) = \begin{cases} \beta_1 & Y = 1\ (channel) \\ \beta_1 O_t & Y = 0\ \ (hillslope) \end{cases} \qquad (2.1.6)$$

where q is the discharge per unit width, $\square_1$ and $\square_5$ are rate constants that may be variable on space and m_1, n_1, m_5, n_5 are typically constants with respect to space and time (though that restriction can be loosened in various ways within the model). K is the erodibility. The coefficient O_t represents a reduction factor, in a reduction exists, in sediment transport rate in the hillslopes compared to that in channels. The factor G is related to the runoff processes to be modelled and assumes particular importance when the subsurface saturation runoff mechanism is simulated. For Hortonian runoff G=1. For the subsurface saturation mechanism it needs to be determined, as described in Section 2.5. Methods for determining the other coefficients from the governing physics will discussed later in this chapter. The discharge in a channel, Q_c, and the discharge per unit width, q, again may be parameterised in any way suitable. We choose to represent them here as

$$Q_c = \square_3 A^{m_3} \qquad (2.1.7)$$

$$q = \frac{Q_c}{w} \qquad (2.1.8)$$

$$w = \square_4 Q_c^{m_4} \qquad (2.1.9)$$

where A is the catchment area draining to that point in the channel, w is the width of the channel at that point, $\square_3$ and $\square_4$ are constants that may be variable on space and m_3 and m_4 are constants with respect to time and space. The equations are solved on a spatial domain with boundary conditions

$$\frac{\partial z}{\partial p} = 0 \qquad (2.1.10)$$

where p is the direction perpendicular to the catchment boundary.

The governing equations are non linear partial differential equations of two states; these two states are elevation, z, and an indicator variable for channelisation, Y. The most important qualitative characteristic of a catchment, the branched network of channels that form the backbone of the drainage system of a basin, is thus explicitly modelled. There are five important variables distributed in space, that are derived from these two states. They are the steepest downhill slope, the contributing area, the discharge, and the distribution of channel initiation function and sediment transport in space. The channel initiation function and sediment transport feed back into, as inputs, the two state equations for elevation and channelisation. Thus there is a non linear interaction between the elevation and channelisation, and the channel initiation function and sediment transport in space. This interaction is the central feature of the model that drives the drainage network growth.

2.2 Tectonic uplift

As previously noted Equation (2.1.1) is a continuity equation in space for sediment transport. The first term in the elevation equation (2.1.1) is the rate of tectonic uplift (positive upwards). This term may be quite general with variability both in space and time. For instance, a spatially uniform uplift event, such as that resulting from an earthquake can be described by

$$c_0(x,t) = \bar{c}_0 \, \delta(t-t_0) \tag{2.2.1}$$

where c_0 is the uplift resulting from the tectonic event occurring at time t_0 and $\delta(t)$ is the dirac delta function

Likewise, the tectonic uplift could be one that occurs continuously with time, but variable in space, such as that resulting from continuous bulging of the continental crest

$$c_0(x,t) = \bar{c}_0(x) \tag{2.2.2}$$

where $\bar{c}_0(x)$ is the spatially variable uplift rate.

The model implements 3 standard tectonic uplift models which may be combined to produce many complex time and space varying tectonic uplifts.

1. Continuous spatially uniform uplift over a specified time period from the start of a simulation

$$c_0(x,y,t) = C \qquad t \in [0,t_0] \tag{2.2.3}$$

2. Continuous tilting uplift over a fixed time period from the start of the simulation

$$c_0(x,y,t) = C\,x \qquad t \in [0,t_0] \tag{2.2.4}$$

3. Spatially uniform, cyclic uplift with either a sinusoidal uplift, square wave uplift, or pulse uplift over the entire period of the simulation

$$c_0(x,y,t) \;=\; T_{amp}\sin\left(\frac{2\pi t}{T_{period}} + T_{phase}\right)$$

$$c_0(x,y,t) \;=\; \begin{cases} T_{amp} & for\ \sin\left(\frac{2\pi t}{T_{period}} + T_{phase}\right) > 0 \\ -T_{amp} & for\ \sin\left(\frac{2\pi t}{T_{period}} + T_{phase}\right) < 0 \end{cases}$$

$$c_0(x,y,t) \;=\; T_{amp}\cdot \square\left(\sin\left(\frac{2\pi t}{T_{period}} + T_{phase}\right)\right) \qquad (2.2.4)$$

Note that elevations are defined relative to the elevation datum at the outlet of the catchment (e.g. the elevation of the outlet notch). Thus the tectonic uplift rate, c_0, is defined as the uplift relative to the elevation of the outlet. To illustrate this consider a small catchment with an outlet on the floodplain of a very large river. The outlet elevation of the small catchment is dominated by elevation changes in the floodplain in the large river; i.e. from the viewpoint of the small catchment the elevation at the outlet is externally imposed and variable in time. In this case c_0 for the small catchment is the tectonic uplift <u>relative the floodplain</u> of the large river (i.e. the catchment outlet elevation) not relative to sea level.

In addition to these standard models of tectonic uplift SIBERIA implements a capability that allows the user to create any type of uplift through a user defined uplift module where the user can specify the uplift in their own FORTRAN subroutine. This capability is discussed in Section 4.

2.3 Erosion processes

Two physically based transport processes are modelled in Equation (2.1.1). It is convenient to differentiate between fluvial and diffusive transport processes. The fluvial transport processes (the 2nd term) are dependent on discharge and slope, while the diffusive transport processes (the 3rd term) are dependent on slope alone.

<u>2.3.1 Fluvial Sediment Transport Processes</u>

The first process, fluvial sediment transport, is dependent on discharge and the slope in the steepest downhill direction. Moreover, from Equation (2.1.6), the rate of the fluvial transport is also assumed to be a function of whether that point in space is channelised or not. In the model, sediment transport on the hillslopes can be less than that in the channel. This effect is parameterised by the constant O_t in Equation (2.1.6), where O_t is less than 1. The β_1 can potentially also parameterise the effect of ground cover, landuse factors, etc on the erosion rate using tabulated values for a range of erosion models (e.g. Knisel, 1980)

In addition to this standard models for β_1 SIBERIA implements a capability that allows the user to create any type of β_1 through a user defined uplift module where the user can specify the β_1 in their own FORTRAN subroutine. This capability is discussed in Section 4.

SIBERIA calculates all processes by a flux based approach. That is, the flux into a node and fluxes out and calculated and the imbalance used to determine the rate of elevation change at the node. Thus the actual process used in the calculations is

$$q_s = \begin{cases} \beta_1 q^{m_1} S^{n_1} - q_{st} & \beta_1 q^{m_1} S^{n_1} > q_{st} \\ 0 & \beta_1 q^{m_1} S^{n_1} \le q_{st} \end{cases} \qquad (2.3.1)$$

where q_{st} is a threshold on the transport process. See below for more explanation.

SIBERIA is capable of modelling a spatially uniform sediment transport (i.e. β_1 constant), or as a random field. In both erosion processes a threshold can be applied on the erosion process. In addition SIBERIA allows this β_1 definition to be extended through a user-defined erodibility module. This extension is discussed in Section 4

2.3.2 Diffusive Transport Processes

The second process in the elevation evolution equation is the diffusive transport term. The long-term average of a number of hillslope transport processes can be modelled by use of a spatially constant Fickian diffusion term; the processes include hillslope soil creep, rainsplash, and rock-slide. It is also incorporates a slope stability threshold. As for the fluvial sediment transport term it is implemented in a flux based within the model

$$q_s = \begin{cases} \dfrac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} - D_{zt} & \dfrac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} > D_{zt} \\ 0 & \dfrac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} \le D_{zt} \end{cases} \qquad (2.3.2)$$

Other mass transport mechanisms could also be modelled but are not implemented in SIBERIA. In particular mechanisms that require knowledge of the regolith depth are not modelled (e.g. plastic transport). Any transport mechanism that is dependent on regolith or soil depth is much more difficult to model than those considered here. This is because the soil depth is variable in space and time and is a function of the geomorphology (i.e. deeper in valleys, shallower on hilltops) which, in turn, is part of the solution of the model. The modelling of soil production and depth, dependent on the complex geochemistry and biology of the catchment and climatic variation, is considered beyond the scope of reliable modelling at this stage and consequently has not been considered.

2.3.3 Extensions to these Processes

A capability to have two fluvial erosion processes occurring in the same catchment at the same time is provided. This capability comes in two forms; the additive model and the switching model.

The additive erosion model is as follows

$$q_s = \left(\beta_1 q^{m_1} + \beta_{12} q^{m_{12}}\right) S^{n_1} \tag{2.3.3}$$

where the main limitation of the additive model is that the exponent on the slope must be the same for both processes. This equation acts everywhere in the catchment.

The other model, the switching model, allows the two fluvial processes but one acts in the channels and the second acts in the channels so that

$$q_s = \begin{cases} \beta_1 q^{m_1} S^{n_1} & hillslope \\ \beta_{12} q^{m_{12}} S^{n_1} & channel \end{cases} \tag{2.3.4}$$

where now only one processes operates at any one place in the catchment. The exponent on the slope term is still limited to being the same in both cases.

The limitation of the slope exponent is because the fast solver within SIBERIA requires the slope exponent be the same. In another type of erosion model allowed where different erosion models are allowed for regions within the catchment this limitation is lifted and the slope exponent may vary in a limited fashion between the different node points.

In all cases a threshold is allowed on the erosion and diffusive transport processes. These are implemented as follows

$$\begin{aligned} q_s &= \begin{cases} \beta_1 q^{m_1} S^{n_1} - Q_{st} & \beta_1 q^{m_1} S^{n_1} > Q_{st} \\ 0 & \beta_1 q^{m_1} S^{n_1} \le Q_{st} \end{cases} \\ q_s &= \begin{cases} \dfrac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} - D_{zt} & \dfrac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} > D_{zt} \\ 0 & \dfrac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} \le D_{zt} \end{cases} \\ q_s &= \begin{cases} \left(\beta_1 q^{m_1} + \beta_{12} q^{m_{12}}\right) S^{n_1} - Q_{st} & \left(\beta_1 q^{m_1} + \beta_{12} q^{m_{12}}\right) S^{n_1} > Q_{st} \\ 0 & \left(\beta_1 q^{m_1} + \beta_{12} q^{m_{12}}\right) S^{n_1} \le Q_{st} \end{cases} \end{aligned} \tag{2.3.5}$$

where the threshold is applied separately to the fluvial and diffusive components. It is thus possible to have the diffusive component above the threshold and the fluvial transport below the threshold simultaneously.

2.4 Runoff models

Runoff is not modelled directly in SIBERIA. Thus there is no continuity of water nor runoff-routing within the model. Rather a conceptually (and computationally) simpler approach (using what is called a sub-grid effective parameterisation) is adopted with a generally accepted conceptualisation of runoff being used. This conceptualisation relates the discharge to area draining through that point.

$$Q_c = \square_3 A^{m_3} \tag{2.4.1}$$

This runoff then feeds into the fluvial erosion model

$$q_s = K q^{m_1} S^{n_1} \square Q_t \tag{2.4.2}$$

The key interest here is the *G* factor in the erosion model. Other runoff models can be implemented through a user-defined module. This capability is discussed in Section 4

<u>Hortonian runoff</u>

For the Hortonian runoff mechanism runoff will occur in all points of the catchment for all storms. Thus for all storms erosion will occur at all places within the catchment. We thus set *G*=1 for the Hortonian mechanism so that all runoff events create erosion at all pints within the catchment.

<u>Subsurface saturation runoff</u>

For the saturation from below (or subsurface saturation) mechanism only those points within the catchment that are saturated produce surface runoff. Thus for any particular storm only part of the catchment will be saturated and only that part will have surface runoff and thus fluvial erosion. That part of the catchment that is not saturated will not suffer fluvial erosion during that storm. The larger the storm the greater the proportion of the catchment that will undergo fluvial erosion. Thus the average erosion at any point in the catchment will be determined by how frequently that point is saturated. The more frequently it is saturated then the more erosion will occur. The *G* factor reflects this behaviour; *G*=1 indicates that point is always saturated, *G*=0 that the point is never saturated, and so on. The G factor varies within the catchment and is a function of the subsurface hydrology of the catchment.

This runoff mode is not fully implemented in V8 of SIBERIA and should not used.

2.5 Channel models

SIBERIA implements two types of channel models. The first is the so called deterministic model; channels once created are fixed forever. The second is called the stochastic channels model; channels are modelled as the average of a stochastic process where channels head advance and retreat in response to climatic fluctuations and are not fixed in position once created.

SIBERIA can model the CIF as spatially uniform (i.e. $\square_5$ fixed) or a spatially random field (uncorrelated in space).

The deterministic channel model

The channelisation equation (2.1.2) is the equation governing the development of channels and the extension of the networks. In the deterministic model this function takes the form

$$f\left(d_t, Y, \frac{a}{a_t}\right) = d_t\left[0.0025\,\frac{a}{a_t} + \left(-0.1Y + \frac{Y^2}{1 + 9Y^2}\right)\right] \quad (2.5.1)$$

This is based on one developed by Meinhardt (1982) to differentiate the leaf vein cells in a leaf for a biological model of leaf reticulation. It is a convenient equation based on the phenomenology of channel head extension rather than fundamentally derived from the controlling transport physics at the channel head which is very complicated and poorly understood at the current time. The form of Equation (2.5.1) causes Y to have two stable attractors, 0 and 1. Typically the modelling process starts with Y=0 everywhere, representing a catchment with no channels, only hillslopes. If desired, a pre-existing channel network can be applied (i.e. Y=1 along the channels). When the value of the channel initiation function, a, exceeds the channel initiation threshold, a_t , the value of Y=0 becomes unstable and Y goes into transition, where it is increasing to Y=1, i.e., that spot in the catchment goes into transition from hillslope to channel. When Y reaches a value of 1 it remains there since the value of Y=1 is stable irrespective of the value of the channel initiation function; channel formation is modelled as a one-way process from hillslope to channel. The role of the channel initiation function is to trigger the channelisation process when the threshold is exceeded. The rate at which a point is channelised once the channel initiation threshold is exceeded is governed by the parameter d_t; a large value of d_t results in the channel forming quickly. Equation (2.5.1) is a convenient, but not exclusive, way of parameterising the abrupt switch from hillslopes to channels in terms of the channel initiation function. Any formulation leading to two stable binary solutions (Y=0; Y=1) and which incorporates the threshold behaviour should work similarly. This formulation is believed to adequately simulate the mean position of the channel head, averaging out any stochastic advance and retreat of the channel head that may occur over short time scales.

The stochastic channel model

The stochastic channel model is based on the tenet that at any point in time the channel head will be in balance with some effective climate forcing. The channel is still assumed to an abrupt change from channel to hillslope at the channel head at any point in time. This climatic forcing may be the runoff for the last 10 years, 100 years or maybe last years runoff. As this climate forcing varies so does the position of the channel head advancing during wet periods, retreating during dry periods. In this mode the exact position can, in principle, be predicted exactly if the forcing is known but only the average oscillations backwards and forwards and the mean position can be predicted if we only know the statistical properties of the climate. Since

SIBERIA predicts the average elevation with time then the appropriate model to use is the latter statistical model where we only use the average climate properties. The sediment transport properties at a point then reflects what proportion of the time that point is a channel and what proportion it is a hillslope, this in turn being driven by the effective climatic fluctuations.

SIBERIA models the stochastic properties by applying a long term random forcing on the runoff parameter $\square_3$. The second order effect of this perturbation on fluvial sediment transport are ignored; the conjunction of high runoff rate and channel advance are ignored by assuming that the timescale of climate perturbation that influence the channel advance (years, decades) are much longer than the timescales of runoff events (minutes, hours).

2.6 Internal numerics

2.6.1 Numerical Solver

The details of the numerical solver for the change in elevation at any point used in SIBERIA can be found in Willgoose, et al, (1989, 1991b). The equations (2.1.1) and (2.1.2) are discretised on the square digital terrain map representing the land surface and the equations evaluated at each point in space over the temporal evolution of the landform. The equations are reevaluated at each time step so that the mass balance and channel network respond to the time evolving landform.

Briefly, the differential equation for elevation (Equation 2.1.1) is solved using an explicit numerical scheme. The sediment mass balance is determined at the current time from analysis of the drainage paths and areas draining through a point and the rate of change of elevation predicted from this mass balance. The solver in time is a two point predictor-corrector algorithm. The algorithm varies from the traditional predictor-corrector with the prediction and correction steps being carried out using an approximate analytical solution to the mass balance at that point, rather than the normal linear extrapolation in time in the traditional solver. This nonlinear extrapolation is required by severe problems of stiffness in the mass balance equation with time scale ranges of 10^4 and more being normal. This nonlinear extrapolation results in much improved stability and mass balance compliance than the traditional linear solver.

Under normal circumstances the user will not need to worry about the details of the solution strategy.

The prime consequence of this solution strategy is for the user who will be using the user-defined extensions to the model. The programmer should note that the user-defined subroutine will be called more than once at every. With a two-step predictor-corrector the subroutine will be called twice at each timestep (i.e. at the same simulation time). In particular, the states passed to the user defined subroutines (e.g. slopes, areas, elevations) will change for each call even though the timestep will not.

2.6.2 Adaptive Time stepping

The other component of the solution technique that may be of importance to the user is the adaptive time-stepping mode. If the timesteps specified for parameter `time` are **positive** then the code will divide the timestep within the code into `1/time` sub-timesteps. In this way the user is able to explicitly control the numerical stability of the mass balance algorithm within the code. If however, the value specified for parameter `time` is **negative** the code will determine the optimal sub-timestep based on a number of criteria that have been found to be predictors of good performance in SIBERIA. This sub-timestep will adaptively change throughout the run adjusting to current conditions. Generally speaking there is a small computational overhead in starting the adaptive timestepping at the beginning of the run but during the run it is considerably faster and more accurate than can be achieved by setting `time` manually.

At the current time the adaptive time stepping will only work correctly when the parameter β_1 is constant in space and time (`ModeErode`=0) as it ignores spatial and temporal variation in β_1.

2.6.3 Automated Completion Criteria

For convenience a range of automated completion criteria are provided.

- **Dynamic Equilibrium (Steady State)**: This criteria will run the model until the catchment is at dynamic equilibrium (Willgoose, et al, 1991d). The process adopted here is that at each time step the change in mass of the landform is compared and an uplift is applied that just balances that change. When the change in the uplift from time step to time step and hypsometric integral are small then the model declares convergence and the simulation stops.
- **Declining Equilibrium**: This criteria will run the model until the catchment is at declining equilibrium (Willgoose, et al, 1994). The process adopted here is that at each time step the change in hypsometric integral of the landform is compared. When the change in the hypsometric integral is small then the model declares convergence and the simulation stops.

3 Running SIBERIA.

3.1 Running the program

3.1.1 Starting the program

The program is designed to run interactively. For batch usage on UNIX computers the best way is to use input/output redirection. When the user has started the program by typing `SIBERIA` he/she is asked a number of questions. In order of appearance they are

Question	Answer
(1)`START from a file ? (filename)`	• Input the name of a restart file. If none is specified (i.e. <return> without inputting any text) then it is assumed that a new run is to be started. The file name cannot have spaces in it. • If you input #### you will be given a list of the versions of the all the key computational components within SIBERIA and the implementation limits of the code. The first # must be in the first column. • If you input @<filename> then SIBERIA will read all of the answers to the following questions from the file called <filename>, exactly as if you were inputting the answers yourself at the keyboard. <Filename> cannot have any spaces within it. The @ must be in the first column.
(2)`BOUNDARIES file ? (filename)`	Input the name of a boundary file. If none is specified then it is assumed that a rectangular domain is to be used. The file name cannot have spaces in it.

(3)`No of times with RESTART output`	Input the number of times at which you want restart data files to be output. If you answer 0 or press <return> with no answer (i.e. no output files are required) then skip to question 7. If you input a negative number then SIBERIA will read the numbers one per line until a blank line is input at which stage input of times is terminated.
If no restart output files requested then skip to question 7	
(4)`Input of each time from run start`	Input the number of timesteps from the start of the run at which the restart files are to output. There are as many lines to be input here as times specified in the previous question. When all required times are input a message `END of times input` is output. In V8.30 the capabilities have been extended slightly. If a single negative number is input then SIBERIA will output files at that timestep spacing. That is if –5 is input then files will be output at times 5, 10, 15, 20, 25, 30… etc, without having to input each of the times separately. This is useful in conjunction with EAMS so that you don't have to manually input each of the times input each time you set up a run.
(5)`Generic filename for these RESTART output files (no extension)`	This is the name that will form the basis for the name that the files to be output will be called. The way the filename is constructed is if `junk` is input here (and files are required at 100 and 200 timesteps) then the output restart files will be called `junk-100.rst2` and `junk-200.rst2`. i.e. <generic filename>–<time>`.rst2`. The file name cannot have spaces in it.

(6)`Absolute Start Time`	This time is used in the construction of the RST2 filenames above and is the absolute time that the run starts at. For instance, the run may be continuing a previous run that output a result at 1000 timesteps and we intend to output for this run at 1000 timesteps from the start. The output file then corresponds to 2000 timesteps in absolute time and in this case you input 1000 here. In this way we can build up a set of output files from a consistent series of separate computer runs.
If a restart file was specified above then skip to question 10	
(7)`INPUT the random number =`	This is a random number for any initial random perturbation on the elevations. By inputting the same random number two runs will be identical and so runs are reproducible (this will also be true across different computers since SIBERIA uses its own random number generator not the machine's).
(8)`INPUT no of outlet nodes =`	This is the number of points within the catchment that will have fixed elevations. For a normal catchment this will 1 (i.e. the outlet) though for various reasons it can be useful to specify more than 1 here.
(9)`INPUT the position of the initial node`	The X,Y nodal coordinates (in pairs) for each of the specified fixed elevation nodes.
(10)`New parameter #`	<u>If you want to modify the default parameters</u> then this is the first place to do it. Specify the parameter # here (see Sections 3.2.1-3.2.3 for more details) and it will tell you what the old parameter value was and prompt you for a new value. <u>If you are happy with the parameters</u> input 0 here to proceed to the next question.

<table>
<tr><td colspan="2">Parameter Values. At this stage you will get a large table of all the current parameter values plus some information of the modes set in the model. You then have a prompt to start the program</td></tr>
<tr><td><code>(11)? 0=New Parameters, 1=Start</code></td><td>Here you may either start the program calculating (enter 1), or further modify the parameters (enter 0 and you go into the modification mode described for the previous question 10).
The filenames are modified by inputting the negative of the filename to be modified (i.e. -1 will modify filename 1).</td></tr>
</table>

3.1.2 Interpreting the program output

Once you have started the program you will get a large table of summary statistics (see below) and line printer contours of various properties of the states (elevations, CIF, slopes, etc) and a plot of drainage directions. The contours are interpreted as ‘ ’ (i.e. a space) lowest through ‘a’, ‘b’ to ‘t’ giving 20 equally spaced contours. For the erosion/deposition picture upper and lower case letters are used to distinguish erosion from deposition. For the graph of the channels ‘ ’ indicates hillslope and ‘t’ channels with letters in between indicating either developing channels (deterministic channels) or the probability of a channel being at that point (for stochastic channels). For the drainage directions a character ‘0’ indicates a self draining node, ‘/’ a node draining either in a NE or SW direction, ‘[‘ a node draining in a NW or SE direction, ‘-’ a node draining E or W, and ‘l’ a node draining north or south.

Most of the numbers in the table of statistics are self explanatory. One statistic of particular use is the potential change in elevation at the outlet/unit time. This says how much the outlet elevation would change if all the sediment transport out of the catchment were to be piled onto that node. Thus dividing this value by the number of nodes draining through that point gives the average decline of the catchment per timestep. Applying that value as a uniform tectonic uplift will very quickly yield a catchment in dynamic equilibrium.

Sample Program Output

(for no initial RST2 file, user responses are bolded)

```
 Scalar Double Precision SIBERIA  V 7.04
  Copyright 1994, G. R. Willgoose
  All rights reserved

  CPU seconds =  9.9999998E-03
 Initial RST file ? (filename)  : <return>
  No input RST file
 BOUNDARIES file ?  (filename)  : <return>
  No input boundary file
 No of times for RESTART output : 10
 Input of each time from run start :
         1000
         2000
         3000
         4000
         5000
         6000
         7000
         8000
         9000
        10000
 END of times input

 Enter Generic filename (no extension) for RST2 output files
 or a filename with ".raw" extension for Mindraft output : test
 Absolute Start Time : 0
 INPUT the random number   1
 INPUT no of outlet nodes  : 1
 INPUT the position of the outlet node :  2 2
New PARAMETER #  0

  INTEGER Parameters
  ------------------
      10000=RunTime       1000=StatTime        10=kx              10=ky
          0=ModeI          100=TimeUp           5=ModeSolv          1=ModeDir
          0=ModeUp           0=ModeRn           0=ModeErod          0=ModeROff
          0=ModeCh           0=ModeDP           0=                  0=
          0=                 0=                 0=                  0=

  REAL Parameters
  ---------------
     0.00000=dZ          1.00000=dZn         0.00000=dZHold      0.00000=QsHold
     1.00000=FactMx      1.00000=FRanMn      0.00030=1/at        1.00000=YFix
     0.00000=FRanCV      0.00000=b3sds       0.00000=b3sdl       0.00000=Tamp
     0.00000=Tperid      0.00000=Tphase      0.00500=FRanZ       1.00000=a1
     1.00000=m3          0.20000=b3          0.01000=b1          1.80000=m1

     2.10000=n1          1.00000=BlkDen      0.02500=TimeStep    2.50000=b5
     0.30000=n5         10.00000=Zinit       0.40000=m5          0.10000=YHold
  1000.00000=notch       0.00000=            0.00000=            1.00000=dtime
     0.10000=otime       1.00000=GridXY      0.00000=East        0.00000=North
     0.00000=b6          0.00000=m6          0.00000=            0.00000=

     0.00000=            0.00000=            0.00000=            0.00000=
     0.00000=            0.00000=            0.00000=            0.00000=
     0.00000=            0.00000=

  User Defined Filenames
  ----------------------
```

```
 -1:Erode :
 -2:Runoff:
 -3:Uplift:
 -4:CtrBnk:
 -5:      :
 -6:      :
 -7:      :
 -8:      :
 -9:      :
-10:Others:
  ----------------------
  Deterministic channels mode set
 ? 0=New Parameters, 1=Start
```

…

etc

…

```
 Restart file test-10000.rst2                          output
 Time =  10000 iterations =     400000
   Max. CIF                =    3.201165934649938
   Range Overland CIF      =   0.0000000000000000E+00   3.201165934649938
   Range Channel CIF       =   0.0000000000000000E+00  0.0000000000000000E+00
   Max. Normalised CIF     =   0.3841399121579925
   Max. Area               =           50
   Slope Range             =   3.7225381786813115E-04  0.6467226307185001
   Elevation Range         =    10.00216125160456       10.99243219363890
   Max. Elevation change   =   2.9527342311917884E-11
   Min. Elevation change   =  -1.9877620485308714E-06
   Potential Elevation change at outlet/unit time =   6.2784971814983808E-04
   Sum of cyclic erosion  =   0.0000000000000000E+00

  Stability No. =     0.000164
  Diffusion Peclet No =   0.0000000000000000E+00

   Relative Mass Balance =    1.000054445493391
   Absolute Mass balance =   8.5454317870809642E-10
   Zin, Zout  =  -1.5695388644459491E-05  1.5696243187638199E-05
   Total Mass =    883.0540539953805
   Hypsometric Integral           =   0.9085802797814257
   Pred. D.E. Hypso. Int. =   0.0000000000000000E+00  -100.0000000000000    %

   Total Area              =           81
   Drainage Density        =   1.2345679012345678E-02
      Channels
-----------------------------------------------------------------------------
 t











-----------------------------------------------------------------------------
      CIF
-----------------------------------------------------------------------------
  kmliffab
 nttpktfb
 mtmttjlbb
```

```
 mtppoiccb
 ltlhfd bb
 jttlcbcbb
 lihgcbb b
 cccbbbbcb
  b b b ca
-----------------------------------------------------------------------------
      Sediment EROSION/Deposition
-----------------------------------------------------------------------------
  ACB
 ECGHAO
 DDCNS
 BFFFC
 AHA
  JI



-----------------------------------------------------------------------------
      s0
-----------------------------------------------------------------------------
  jpmeaa
 tcdmida
 qdoeega
 ndkkie
 kekca
 ggec
 bdcb


-----------------------------------------------------------------------------
      Elevations
-----------------------------------------------------------------------------
  fprsssss
mmcfosrsss
qqfpknssss
rriprsssss
sslsssssss
ssqqssstss
ssssssssss
ssssssssss
ssssssssss
-----------------------------------------------------------------------------
      Drainage Directions
-----------------------------------------------------------------------------
OO-|//|/[|
O|[--//-/O
O-|-[--[-|
O-|-|[[|[-
O/|[[[[O-/
O/|[--/||/
O/||[[|-O-
O|[|/|[||[
OO-O-O-O--
-----------------------------------------------------------------------------
  END
  CPU seconds =   1807.640
```

3.2 SIBERIA parameters.

There are a large number of parameters for SIBERIA. These parameters control the (1) mode of operation of the mode (i.e. what algorithms to use, what exogenous inputs to apply); (2) the actual physical parameters in the model; and (3) numerical behaviour of the model (e.g. the number of timesteps and frequency of output of statistics, the size of the timestep).

In modifying the parameters they are referred to by a number, the first parameter is #1, the second is #2 and so on. The parameters are split into three groups.

- The first group are the integer parameters which are the first 20 parameters (i.e. #1 to #20). These parameters typically control the mode of operation of the model and what physical models are used for the simulation.
- The second group are the real parameters which are the next 50 parameters (i.e. #21 to #70). The parameters are the actual physical parameters for the model.
- The third group are the 10 model filenames referred to by negative of the file number (i.e. #-1 to #-10). These are typically the filename associated with each of the major model components within SIBERIA (erosion, runoff, tectonics, soils, etc)

A brief description of each parameter is given in the table below. A more detailed description follows in Section 4.

In the table below some parameters are blank because they are currently not used in the code: they exist to provide compatibility with future versions of SIBERIA that may need further parameters to control new modes of behaviour. In this way the format of the restart files is guaranteed not to change for some time. This is done because many visualisation packages require the header have a fixed format or that it has a fixed number of values so that it can skip the header before reading the data (e.g. Silicon Graphics Explorer) and by having a set number of parameters that we will fill in over time it means that the input routine for the visualisation package does not need to be changed for each minor update of SIBERIA. Future versions will simply fill in the blanks in the table below.

3.2.1 Integer parameters

No	`Parameter`	Description
1	`RunTime`	• No of time steps to solve for in this run. • Negative values are for the automated completion criteria. If `RunTime`=-1 then this this termination criteria is achievement of declining equilibrium, if `RunTime`=-1 then it is dynamic equilibrium. • See also parameter 2.
2	`StatsTime`	• Line printer and statistics output of contours every `StatsTime` timesteps. `RunTime/StatsTime` must be integer. If the input value of `StatsTime < RunTime` then the program sets `StatsTime = RunTime` • See also parameter 1.
3	`kx`	• X dimension of the rectangular grid. • See also parameter 3.
4	`ky`	• Y dimension of the rectangular grid. • See also parameter 4.

5	`ModeI`	• Type of initial condition to be applied to initial elevations when no initial RST file is specified for the run.
		  • 0 = flat level surface with initial value = `ZInit`
		  • 1 = surface sloping upwards in the +ve X direction with max height = `ZInit`
		• If a RST2 file is input this parameter is ignored.
		• See also parameters 6, 46

6	`TimeUp`	• End time for application of the tectonic uplift. The rate of uplift is $U = \begin{cases} \frac{Z_{init}}{Notch} & time < TimeUp \\ 0 & time \geq TimeUp \end{cases}$ • See also parameters 5, 6, 46 and 49. • NB: this uplift is superimposed on top of the uniform uplift of the cyclic uplift function (parameters 32, 33, and 34).

7	`ModeSolver`	• Mode of solution of the sediment transport relation • 0 (disabled)=> solution. for elevations by Taylor Series • 1 (default) => solution. of the physical transport equation • 3 => Corrected version of `ModeSolver`=5 with correct BC for Area=1. As for `ModeSolver`=4 except that two fluvial processes occur everywhere. • 4 => Corrected version of `ModeSolver`=5 with correct BC for Area=1. As for `ModeSolver`=3 except that two fluvial processes switch. 1st process occurs only on hillslope, 2nd process occurs only in channel. The default solver for versions later than V8.02. • 5 => solution. by explicit/analytic solution (nonlinear predictor/corrector). The default solver until V8.02. • 6 => Shear stress driven source limitation model. • 7 => equivalent to `ModeSolver`=5 except that there is no erosion at on hillslopes.

8	`ModeDir`	• Mode of solution of the directions in the diranal routine
		• 0 => directions as steepest slope.
		• 1 (default) => directions for the channels are writ in stone provided that channels drain into channels
		• 2 => directions as steepest slope with constraints on slope provided by contour banks.
		  • See also parameters 8, -4.
		• 3 => directions as specified in the input RST2 file and do not change during the simulation.
		• 4 => directions as steepest slope with constraints on slope provided by contour banks.
		• 5 => highly optimised version of `ModeDir`=0 for RISC processors.
		• 6 => average flow direction for the region rather just the adjacent 8 nodes.
		  • See also parameters 8, 16.
		• 7 => directions as per D∞.

9	`ModeUplift`	• Mode of **Uplift Model**.
		$$U = T_{amp} \sin\left(\frac{2\pi\left(T_{phase} + timestep\right)}{T_{period}}\right)$$
		  • 0 (default) => No perturbation
		  • 1 => Sinusoidal uplift *rate* with amplitude T_{amp}, Period T_{period}, and initial phase T_{phase}
		  • 2 => Square wave uplift *rate* with parameters as for sinusoidal
		  • 3 => Impulse uplift with height (note not a *rate* but the amount of uplift per impulse) T_{amp}, Period T_{period}, and initial phase T_{phase}
		  • -ve => user defined model
		• See also parameters 9, 32, 33, 34.
		• NB: this uplift is superimposed on top of the uniform uplift of `ZInit` and `Notch` (parameters 6, 46, and 49). Note that `Notch` is only used to stop the uniform uplift so that the cyclic uplift applies for the complete duration of the simulation.

10	`ModeRandom`	• Mode of Random perturbation using `FRanCV` and `FRanMn`.
		  • 0 (default) => No Perturbation
		  • 1 => *β_1 (i.e. random erosion)
		  • 2 => *β_5 (i.e. random channel threshold)
		• See also parameters 10, 26, 29.

11	`ModeErode`	• Mode for the **Erosion Model**
		  • 0 (default) => spatially constant erodibility, dependent only on the channelisation (ie. β_1 for the channel and $\beta_1 O_t$ for the hillslope)
		  • 1 => depth dependent armouring model
		  • 3 => spatially variable erodibility regions specified by RGN file
		  • +20 => erodibility β_1 specified per unit width rather than per grid point.
		  • -ve => user defined model
		• See also parameters 11, 24, 39, 40 and 41.

12	`ModeRunoff`	• Mode for the **Runoff Model**
		  • 0 (default) => spatially constant RUNOFF
		  • 1 => spatially variable runoff rate input from file (see Section 4.3).
		  • 2 => known inflows from offsite
		  • 3 => spatially variable runoff regions specified by RGN file
		  • +20 => runoff rate □3 specified per unit width rather than per grid point.
		  • -ve => user defined model
		• See also parameters 12, 37 and 38.

13	`ModeChannel`	• Mode for the **Channel Model**
		  • The depth of the channel is given by
		$depth = b_6 X^{m_6}$
		where
		  • 0 (default) => zero depth channel (i.e. *depth*=0)
		  • 1 => *X*=area
		  • 2 => *X*=discharge
		  • 3 => turns off ALL channel calculations (including CIF determination).
		• See also parameters 13, 57, and 58.

14	`ModeDP`	• Mode for the user defined **Dependent Model**
15	`ModeMC`	• Mode for Monte Carlo simulation of landforms (PVM version of SIBERIA only)
16	`DirReg`	• Size of the region used to determine the slope and drainage directions. Default value is 1 (i.e. only adjacent nodes are examined).
17	`ModeSoil`	• Mode for the **Soil Model** • 0 (default) => under development
18-20	.	•

3.2.2 Real parameters

No	Parameter	Description
21	`dZ`	• Coefficient of diffusion D_z in sediment transport. $Q_s = \begin{cases} \dfrac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} - D_t & \dfrac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} > D_t \\ 0 & \dfrac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} \leq D_t \end{cases}$ • See also parameters 21, 22, 23 and 51.
22	`dZn`	• Exponent of nonlinearity D_{zn} in the diffusion of sediment transport (default=1) $Q_s = \begin{cases} \dfrac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} - D_t & \dfrac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} > D_t \\ 0 & \dfrac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} \leq D_t \end{cases}$ • See also parameters 21, 22, 23 and 51.
23	`dZHold`	• Threshold D_t below which diffusive sediment transport does not occur (default=0) $Q_s = \begin{cases} \dfrac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} - D_t & \dfrac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} > D_t \\ 0 & \dfrac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} \leq D_t \end{cases}$ • See also parameters 21, 22, 23 and 51.
24	`QsHold`	• Threshold Q_t below which fluvial sediment transport does not occur (default=0) $Q_s = \begin{cases} \beta_1 Q^{m_1} S^{n_1} - Q_t & \beta_1 Q^{m_1} S^{n_1} > Q_t \\ 0 & \beta_1 Q^{m_1} S^{n_1} \leq Q_t \end{cases}$ • See also parameters 11, 24, 39, 40 and 41.
25	`FactMx`	• The maximum value for FACTOR in SEDANAL used in the calculation of O_t. (default = 1) • See also parameters 25, 53.

26	`FRanMn`	• Mean for random fluctuations. • See also parameters 10, 26, 29.
27	`1/at`	• Coefficient on CIF in differentiation equation 0.025/c1 = CIF threshold $c_1\square_5(a_1Q)^{m_5}S^{n_5} = \frac{\square_5(a_1Q)^{m_5}S^{n_5}}{a_t}$ $\frac{\square_5(a_1Q)^{m_5}S^{n_5}}{a_t} > 1 \quad \square \; channel$ $\frac{\square_5(a_1Q)^{m_5}S^{n_5}}{a_t} > \quad \square \; hillslope$ • See also parameters 27, 36, 44, 45 and 47.

28	`YFix`	• The CIF allocated to fixed elevation points
		  • 0 => all fixed elevation points are hillslope
		  • 1 => all fixed elevation points are channels (default)

29	`FRanCV`	• Scaling factor for random fluctuations. Maximum value =2. If FranCV=1 then it is unformly distributed between 0.5 and 1.5. • See also parameters 10, 26, 29.
30	`b3SDs`	• Standard deviation for short term variations in the runoff rate (used in variation of saturation from below regions runoff model)
31	`b3SDl`	• Standard deviation for long term variations in the runoff rate (used in variation of channel head position in stochastic channels model)
32	`Tamp`	• Amplitude of the cyclic uplift. $U = T_{amp}\sin\left(\frac{2\square\left(T_{phase} + timestep\right)}{T_{period}}\right)$ • See also parameters 9, 32, 33, 34. • NB: this uplift is superimposed on top of the uniform uplift of `ZInit` and `Notch` (parameters 6, 46, and 49). Note that `Notch` is only used to stop the uniform uplift so that the cyclic uplift applies for the complete duration of the simulation.

33	`TPeriod`	• Period of the cyclic uplift (timesteps). $U = T_{amp} \sin\left(\frac{2\pi\left(T_{phase} + timestep\right)}{T_{period}}\right)$ • See also parameters 9, 32, 33, 34. • NB: this uplift is superimposed on top of the uniform uplift of `ZInit` and `Notch` (parameters 6, 46, and 49). Note that `Notch` is only used to stop the uniform uplift so that the cyclic uplift applies for the complete duration of the simulation.
34	`TPhase`	• Phase of the cyclic uplift at t=0. $U = T_{amp} \sin\left(\frac{2\pi\left(T_{phase} + timestep\right)}{T_{period}}\right)$ • See also parameters 9, 32, 33, 34. • NB this uplift is superimposed on top of the uniform uplift of `ZInit` and `Notch` (parameters 6, 46, and 49). Note that `Notch` is only used to stop the uniform uplift so that the cyclic uplift applies for the complete duration of the simulation.
35	`FRanZ`	• Standard deviation for random fluctuations in the elevation initial conditions when no `.RST2` file is specified as input
36	`a1`	• Factor a_1 relating the discharge used in sediment transport and the channel initiation function. (Default =1) • See also parameters 27, 36, 44, 45 and 47.
37	`m3`	• Exponent on the area in discharge used in the sediment transport. $Q = \beta_3 A^{m_3}$. • See also parameters 12, 37 and 38.

38	`b3`	• Coefficient between discharge and area in the sediment transport formula. $Q = \beta_3 A^{m_3}$ • See also parameters 12, 37 and 38.
39	`b1`	• Coefficient β_1 in the fluvial sediment transport formula. $Q_s = \begin{cases} \beta_1 Q^{m_1} S^{n_1} - Q_t & \beta_1 Q^{m_1} S^{n_1} > Q_t \\ 0 & \beta_1 Q^{m_1} S^{n_1} \le Q_t \end{cases}$ • See also parameters 11, 24, 39, 40 and 41.
40	`m1`	• Exponent on the discharge m_1 in the sediment transport formula. $Q_s = \begin{cases} \beta_1 Q^{m_1} S^{n_1} - Q_t & \beta_1 Q^{m_1} S^{n_1} > Q_t \\ 0 & \beta_1 Q^{m_1} S^{n_1} \le Q_t \end{cases}$ • See also parameters 11, 24, 39, 40 and 41.
41	`n1`	• Exponent on this slope n_1 in the sediment transport formula. $Q_s = \begin{cases} \beta_1 Q^{m_1} S^{n_1} - Q_t & \beta_1 Q^{m_1} S^{n_1} > Q_t \\ 0 & \beta_1 Q^{m_1} S^{n_1} \le Q_t \end{cases}$ • See also parameters 11, 24, 39, 40 and 41.
42	`Bulk`	• Effective porosity of soil which relates the volume rate of transport with the actual amount of elevation decrease. This is only used in the conversion of elevation changes to tonnes/Hectare erosion loss in RSU output.
43	`InitTimeStep`	• If this parameter is positive then this is the size of the time step size to be used for the run. You must ensure that `1/` InitTimeStep is an integer. • If this parameter is set to negative value then the adaptive timestepping algorithm is used to determine the step size.

44	`b5`	• Coefficient β_5 on the channel initiation function. • See also parameters 27, 36, 44, 45 and 47.
45	`n5`	• Exponent m_5 on slope used in the channel initiation function equation. • See also parameters 27, 36, 44, 45 and 47.

46	`ZInit`	• *For a new run* (i.e. no RST2 used for starting) this is the initial elevation of the whole region. Uplift then continues as per the continuing run case.
		  • See also parameters 5, 46.
		  • NB: this uplift is superimposed on top of the uniform uplift of the cyclic uplift function (parameters 32, 33, and 34).
		• *For a continuing run* (i.e. a RST2 is being used to start the run) this is the uniform uplift being applied over the time `notch`.
		  • For the continuing uplift the rate of uplift
		$$U = \begin{cases} \dfrac{Z_{init}}{Notch} & time < TimeUp \\ 0 & time \geq TimeUp \end{cases}$$
		  • See also parameters 6, 46 and 49
		  • NB: this uplift is superimposed on top of the uniform uplift of the cyclic uplift function (parameters 32, 33, and 34).

47	`m5`	• Exponent n_5 on area in the channel initiation function equation. • See also parameters 27, 36, 44, 45 and 47.
48	`Yhold`	• Threshold used to determine when a hillslope goes to being a channel.

49	`Notch`	• Time over which `ZInit` elevation change is applied to the notch for a uniform uplift with time. The rate of uplift is $$U = \begin{cases} \frac{Z_{init}}{Notch} & time < TimeUp \\ 0 & time \geq TimeUp \end{cases}$$ • See also parameters 6, 46 and 49. • NB: this uplift is superimposed on top of the uniform uplift of the cyclic uplift function (parameters 32, 33, and 34).
50	`Cover`	• The cover factor for erosion (ala. USLE). Default = 1
51	`S0Max`	• The maximum stable slope $S_{threshold}$ used in the diffusive transport model. $$Q_s = \begin{cases} \frac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} & \frac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} \leq D_t \\ D_t & \frac{D_z S^{D_{zn}} S_{threshold}}{S_{threshold} - S} > D_t \\ 0 & \end{cases}$$ • If this parameter is less than or equal to 0 then no maximum stable slope is applied; i.e. $S_{threshold}$ is considered infinite. • See also parameters 21, 22, 23 and 51.
52	`Dtime`	• Rate at which a node changes from hillslope to channel transition once the CIF is exceeded. Default=1
53	`Otime`	• Factor to adjust the relative rates of overland to channel sediment transport adjustment. $$Q_{s,\ hillslope} = Q_{s,\ channel}\, O_t.$$ • See also parameters 25, 53.
54	`GridXY`	• The grid spacing in the X and Y directions in the units of the simulation (e.g. metres) (default =1) • See also parameters 54, 55, and 56.
55	`East`	• The easting of the SW corner of the grid in the units of `GridXY` (default=0) • See also parameters 54, 55, and 56.
56	`North`	• The northing of the SW corner of the grid in the units of `GridXY` (default=0) • See also parameters 54, 55, and 56.

57	`b6`	• Coefficient b_6 in the channel depth model (`ModeDP`) $depth = b_6 X^{m_6}$ where • `ModeChannel` =1 => X=area, • `ModeChannel` =2 => X=discharge • See also parameters 13, 57, and 58.
58	`m6`	• Exponent m_6 in the channel depth model (`ModeDP`) $depth = b_6 X^{m_6}$ where • `ModeChannel`=1 => X=area, • `ModeChannel` =2 => X=discharge • See also parameters 13, 57, and 58.
59	`b12`	• For the second fluvial erosion process this is the b1 parameter. • If `b12`=0 then the second fluvial erosion process is not used. • See also parameter 39 and 60
60	`m12`	• For the second fluvial erosion process this is the m1 parameter • If `b12`=0 then the second fluvial erosion process is not used. • See also parameter 40 and 59.
61	`SoilRate`	• Soil parameter. • Model under development. • See also parameters 61, 62, 63 and 64.
62	`SoilExp1`	• Soil parameter. • Model under development. • See also parameters 61, 62, 63 and 64.
63	`SoilExp2`	• Soil parameter. • Model under development. • See also parameters 61, 62, 63 and 64.
64	`SMThreshold`	• Soil moisture parameter. • Model under development. • See also parameters 61, 62, 63 and 64.
65-70	.	•

3.2.3 File name parameters

No	Parameter	Description (and control parameter)
-1	FileErode	Filename for the erosion model. (see ModeErode)
-2	FileRunoff	Filename for the runoff model. (see ModeRunoff)
-3	FileUplift	Filename for the tectonics model. (see ModeUplift)
-4	FileCtrBank	Filename for user defined drainage directions and contour bank model. (see ModeDir)
-5--9	.	•
-10	FileOthers	Filename for the dependent variables model (see ModeDP)

3.3 File formats.

3.3.1 Introduction

There are 3 standard file formats that SIBERIA creates and uses.

The first two file formats contain the digital terrain map (DTM) data and contain sufficient data for SIBERIA to be able to restart from this without having to reinput the parameters (the user might want to change some of the parameters controlling the length of run though). These two file formats have identical and interchangeable data; their difference is in their internal format. The two different file formats basically perform the same purpose; one is in plain text format while the other is a substantially compressed binary format. The common name is that these file are **restart** files. Many other ancillary programs use these as input (e.g. FEOUT and VIEWER) for their analysis of the data.

The text restart file (with a standard file extension `.RST2`) is the standard input/output file for runs with SIBERIA. This file format is normally used when files need to be routinely transferred from one computer to another. The binary restart file (with a standard file extension `.RST3`) is not output by SIBERIA but can be input by SIBERIA. This file format is normally used when disk space is a premium and files will not be transferred from one computer to another (because the binary representation of numbers can vary from one computer to another).

The third file format used by SIBERIA is the boundary information for an irregularly shaped region, has a standard extension of `.BND` and is called a boundary file. This file is a plain text file and can be routinely transferred from one computer to another. Caution should be used when transferring this file type with KERMIT because the file can have lines longer than 80 characters (i.e. `kx` > 80) and KERMIT normally truncates all lines longer than 80 characters.

Appendix C has a listing of a shell program that may be used as the basis of a program to analyse the RST and BND files.

3.3.2 The RST2 data file

The RST2 file is the main data file format used by SIBERIA. It includes all data require to rerun or restart a new run. Thus it has in it all the states of the model and the parameters. The filename always ends with the file extension `.rst2` (lower case).

The actual format of the RST2 file has changed over time as SIBERIA has developed. The version of the file described below is that output by version 8 of the code. There are substantial differences in format of the parameters for version 6 and 7 of the code, however, SIBERIA V8 can input V6 and V7 files (though it only outputs V8 file formats). While this file can be modified manually great care should be taken. In particular, the first line and the values of `kx` and `ky` should not be modified.

The **1st line** of the RST2 file is an identifying line that states what version of what code created this file. In the case of SIBERIA the first thing on the line is the string SIBERIA (starting in either the 1st or 2nd column and all upper case depending on the computer that the file was generated on). The second item on the first line is a real number identifying the version of the program that created this file.

The **2nd to 14th line** of the .RST2 file are the parameters of the run that generated this file. See Section 3.2 for an explanation of the parameters. The 2nd to 5th line are the first 20 parameters which are all the integer parameters. The 6th to 14th line are the next 50 parameters which are all the real parameters. Note that the values of kx and ky will be overridden by the values in the boundary file if irregular boundaries are input (see below).

Lines 15 to 24 are lines with text on them. Each line is a filename for one of the model modes, and the use of the text on each line is outlined in Sections 4. In Version 8 five lines are active (see the table above for the filename parameters), the other 5 should be left blank. Even if filenames are not required to be input the lines must be input and left blank (i.e. there must always be 10 lines in this section)..

The **25th line** specifies how many nodes in the grid have fixed elevations (i.e. catchment outlets). This number, `init`, will always be 0 or greater, with a maximum number of 1500. The **26th to (25+**`init`**)th line** specify the `x`, `y` of each node with fixed elevation. If you are also inputting irregular boundary conditions (using a boundary file) carefully read how this portion of the file may be affected and how these values may be overridden by those in the `.BND` file (Section 3.3.4). Note that some visualisation programs (e.g. IRIS Explorer) cannot cope with `init=0` so that at least one fixed elevation node will be required.

The **remaining lines** input the states of the grid solved by SIBERIA. The data are listed in 8 columns with each line being the listing of the grid along the `x` direction first. The 8 columns in the file are (left to right) slope, random field, channelisation, elevation, area, drainage direction, channel depth and soil depth.

- The slope is calculated assuming the Δx and Δy of the finite difference grid are 1 units.
- The random field is that used to perturb the coefficients of sediment transport and CIF (see parameter 10, `ModeRn`).
- The channelisation state indicates directly whether a point is a channel or a hillslope (for the deterministic channels mode the values will either be

approximately 0 or greater than 1; for the stochastic channels mode there will be a distribution of values from 0 to 1 reflecting the proportion of time that grid point is a channel).

- The drainage directions (Figure 3.1) indicate which of the 8 adjacent nodes the node drains into (the interpretation is 1=(-1,0) i.e. one node negative in x and 0 in y, 2=(0,-1), 3=(1,0), 4=(0,1), 11=(-1,1), 12=(-1,-1), 13=(1,-1), 14=(1,1), 0 and 5-9=(0,0) i.e. drain to themselves). If irregular boundaries are specified then those nodes that lie outside the domain will have all 0's for the states and 5 for the drainage direction (see the sample below).
- The channel depths are in the units of the vertical elevations. If no channel model is specified then this column is zero.
- The soil depth are in the units of vertical elevations. If no soil model is specified then the column is zero.

Sample code that reads in restart files is listed in the Appendices.

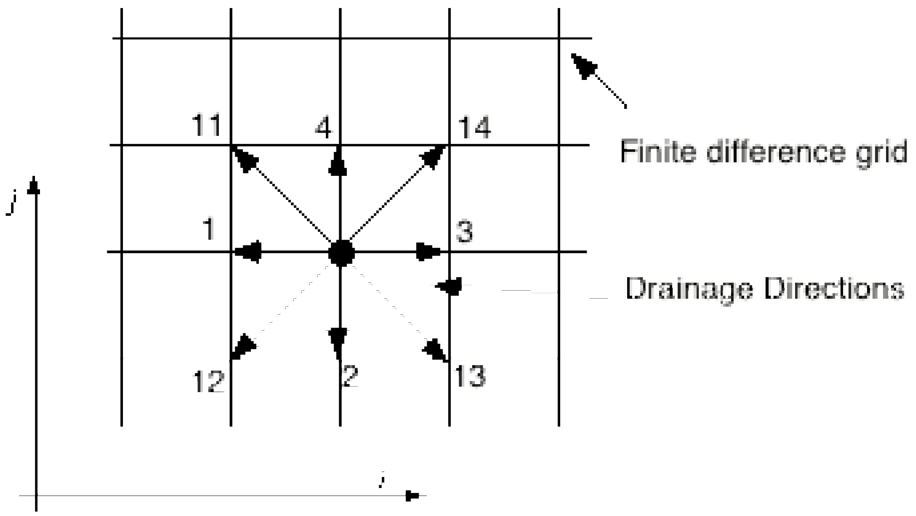


Figure 3.1: Drainage direction definitions on the finite difference grid.

Example .RST2 File

```
SIBERIA         8.01
    20000      2000         59         36          0          0
        5         1          0          1          0          0
        0         0          0          0          0          0
        0         0
        .000000       1.000000        .000000        .000000       1.000000
       1.000000        .000000       1.000000        .100000        .000000
        .000000        .000000        .000000        .000000        .005000
       1.000000       1.000000        .200000        .010000       1.800000
       2.100000        .300000        .250000       2.500000        .300000
      10.000000        .400000        .100000       2.000000        .000000
        .000000       1.000000        .100000        .000000        .000000
        .000000        .000000        .000000        .000000        .000000
        .000000        .000000        .000000        .000000        .000000
        .000000        .000000        .000000        .000000        .000000
                                                Filename for user erosion module
                                                Filename for the user runoff module
                                                Filename for the user tectonics module
                                                Filename for the drainage directions module
                                                ]
                                                ]
                                                ]  These lines not currently used
                                                ]
                                                ]
                                                Filename for dependent variables
1                                               No of fixed elevations
59 29                                           x,y of fixed elevation pt
0  0  0  0  0  5  0  0
0  0  0  0  0  5  0  0
0  0  0  0  0  5  0  0
0  0  0  0  0  5  0  0
0  0  0  0  0  5  0  0                          An example of a node outside the domain
0  0  0  0  0  5  0  0
0  0  0  0  0  5  0  0
0  0  0  0  0  5  0  0
.4572E+00 1.0548  .0010  .1484124E+02    1 14 0.0 0.0        Node  inside the domain
.5340E+00 1.0869  .0010  .1472861E+02    1  4 0.0 0.0
.4953E+00 1.0198  .0010  .1440068E+02    1 14 0.0 0.0
.5920E+00 1.0988  .0010  .1429211E+02    1  4 0.0 0.0
.6363E+00 1.0032  .0010  .1432361E+02    1  4 0.0 0.0
…
…
```

3.3.3 The RST3 data file.

The format of the .RST3 file is a direct binary conversion of the .RST2 file. The filename always ends with the file extension `.rst3` (lower case). This file format is typically an order if magnitude smaller in size than the .rst2 format. The same values appear in the same fashion within the file. See the sample code in Appendix A for how to input this file format. Sample code that reads in restart files is listed in the appendices. The drainage directions are INTEGER*2 while all other integers (i.e. area, fixed elevation points, parameters) are INTEGER*4. All reals are REAL*4. The text character strings for the user-defined functions are CHARACTER*80. '`SIBERIA`' is a CHARACTER*20, and the version number is INTEGER*2.

SIBERIA does not output RST3 format because binary files are typically incompatible between different types of computers. However, many of the support package for SIBERIA support the rst3 file format. Using the UNIX compress facility typically yields a file size comparable with the RST3 files.

3.3.4 The RSU data file

The RSU file format is a flexible format for storing spatially distributed data from SIBERIA runs. Typically if there is an RSU file there is also a corresponding one with the RST2 file extension. For example, if there is a file '`run5-1-0001000.rsu`' then there will also be a file '`run5-1-0001000.rst2`'. Programs like VIEWER will automatically look for an RSU file when they input an RST2 file and will input it if it exists. The filename always ends with the file extension `.rsu` (lower case).

The file format is as follows

- **Line 1:** A header starting with '`SIBERIA`' identifying the version of SIBERIA this file compatible with.
- **Lines 2-4:** Three lines of title information. These three lines can be used for any purpose and might for instance give a short description of what this file is.
- **Line 5:** Three numbers which from left to right are (1) the x size of the grid, (2) the y size of the grid and (3) the number of data that follow.
- **Line 6:** The headings for the columns that follow. The headings are in free format so an enclosed in apostrophes.
- **Line 7 and beyond:** The data in column-wise format where the x coordinate is read first and both x and y coordinates are in ascending order starting from x=1 and y=1.

The first few lines of a sample file follow

Sample RSU File

```
SIBERIA     8.08000000000000
  RSU output from SIBERIA


           76          92           2
 'Area/Slope_No'  'Fluvial/Diffusion_No'
  0.000E+00  0.000E+00
  0.000E+00  0.000E+00
  0.000E+00  0.000E+00
0.136E+01  0.700E-04
  0.131E+02  0.411E-02
  0.363E+01  0.408E-03
  0.272E+01  0.243E-03
  0.427E+00  0.865E-05
```

3.3.5 The BND boundary file.

The boundary file is the means of inputting irregular boundary conditions. If a rectangular domain is all that is required then it is generally unnecessary to use this file, just specify the `kx` and `ky` as the size of the domain. The filename always ends with the file extension `.bnd` (lower case)

- The **1st line** is a heading describing the contents of the file.
- The **2nd line** has two numbers with the x and y dimensions of the grid (exactly the same as `kx` and `ky` in the parameter list). If these values are bigger than the values of `kx` and `ky` specified in the restart file then the input values in the boundary file <u>will override</u> those in the restart file.
- The **remaining lines** give the domain of the problem. The data is input on a rectangular grid of the same dimensions as input in the 2nd line. The first line is y=1, the second y=2, etc; i.e. if the file is printed out then the catchment will be upside down when viewed in normal Cartesian coordinates. The notation is that a ‘`.`’ indicates a point outside the domain. Any other value is within the domain. The character ‘`*`’ is a generic point within the domain. The character ‘`^`’ is a fixed elevation point. Other symbols are planned for specific purposes and will be implemented as need arises (see below). Note that fixed elevation points input here (i.e. ‘`^`’) <u>will override</u> those input in the restart file (this is the interaction between boundary and restart files noted in Section 3.2.2).

Note that for various algorithmic reasons the first column (i.e. y=1) and first row (i.e. x=1) MUST be outside the domain (i.e. ‘`.`’). A sample file follows. Sample code that reads in boundary files is listed in the Appendices.

The boundary file can include definition of regions for which hydrology and sediment transport information can be requested. The regions are designed to make accounting and monitoring of the quantities of runoff and sediment transport within the domain easier. They can (1) represent **retention ponds** that capture sediment or (2) **accounting regions** for which runoff and sediment transport information is required. *This capability is no longer fully implemented in SIBERIA and may not work correctly*.

- For the **retention ponds** SIBERIA tracks the amount of runoff and sediment flowing into each region and will return erosion statistics for them. Elevations within the retention ponds are fixed and do change with time.
- For the **accounting regions** runoff and transport through the region is calculated. For accounting regions the elevation changes in the accounting regions are not affected by the specification of accounting. They respond to erosion and deposition in the normal fashion.

Each region is represented in the boundary file by a number or lower case letter for the nodes for which they will represent; retention ponds are defined by the numbers 0-9, the accounting regions by the lower case letters a-z. For instance, the regions can be applied around the edge of the boundary and they will then report the sediment transport into and out of the domain. In the sample file '`4`' represents a retention pond, '`d`' represents part of the boundary with changing elevations for which runoff and sediment will be reported, while '`1`' represents another part of the boundary for which elevations are fixed and for which transport will be reported.

Example .BND File

```
 Boundary File from Xnetworks 20 m grid from v2 triangulation
          59          36
...........................................................
...............................*****.......................
...........................**********......................
......................****************.....................
.....................******************....................
.................***********************...................
................*************************..................
.....*........****************************.................
...*****.....******************************................
.****************************444************...............
.*****************************44*************..............
..**************************4444**************.............
..****************************444**************............
..**********************************************...........
...**********************************************..........
...***********************************************.........
....***********************************************........
.....***********************************************.......
......***********************************************......
.......***********************************************.....
.......d**********************************************.....
........d**********************************************....
.........d*********************************************....
..........ddddd*****************************************...
................*****************************************..
..................***************************************..
...................**************************************..
....................**1111111111111***********************.
....................11.............1**********************^
.....................................1*********************
......................................*******************..
..........................................**************...
...........................................************....
............................................**********.....
............................................****...........
.............................................**............
```

3.3.6 The RGN region file

The RGN file is a modified version of the BND file used for storing information about the spatial extent of some characteristic. The filename always ends with the file extension `.rgn` (lower case).

In appearance it is identical to the boundary file except that the catchment outlets (i.e. the letter '^') do not need to be identified. It's main use is a way of inputting information about different regions of runoff and erosion properties. A region is identified by any symbol other than a '.' and using the runoff and erosion models (see Section 4) to apply a different value of runoff and erosion over that region. RGN files are normally generated by selecting a region in VIEWER as its rather tedious to generate them by hand.

3.4 Controlling the Operation of SIBERIA

The overall operation of SIBERIA is controlled by two files that are read immediately after SIBERIA starts.

The first file is one called 'default-directory.txt' which changes the directory from which all files are read and written. This file must be placed in the directory in which SIBERIA starts up (see table below) and allows all files to read from and written to a different location. Typically this is modified once to configure SIBERIA and then only modified infrequently thereafter. The format of the file is as follows. Note that the directory format is the default text line format for the appropriate operation system. Note also that the line MUST be terminated by the operating system directory seperator (\ in Windows and / in UNIX).

Windows

```
DEFAULT
D:\siberia-runs\
```

UNIX (all versions including LINUX and Mac OSX)

```
DEFAULT
~/siberia-runs/
```

Table: Default Startup Locations for SIBERIA

Operation System	Method of Startup	Location
Windows	From Start bar as part of EAMS install	The default directory for the installation of programs. Normally something like C:\PROGRAM FILES\TELLURICRESEARCH\EAMS\
	Double clicking on the executable in a directory called D:\HERE1\	D:\HERE1\
	Double clicking on an alias in a directory D:\HERE1\ where the original executable is in C:\HERE2\	C:\HERE2\
Mac OSX	Click on SIBERIA.COMMAND	Your home directory. That is the result of typing `cd ~` at the UNIX prompt (in the TERMINAL application) or the location when Go>Home is selected in FINDER.
	Start at UNIX command line in the TERMINAL application	The directory given by the `pwd` command irrespective of the executable location

Other UNIX	Start at UNIX command line	The directory given by the `pwd` command irrespective of the executable location

The second file that is read in at the start of the run by SIBERIA called '`siberia.setup`' (all lower case). As a general rule anything that changes what SIBERIA does without actually impacting on the values output for the simulated landforms is input in this file. The reasoning behind this nowhere in any of the output files are the parameters in `siberia.setup` actually output so only parameters that change what SIBERIA outputs are allowed here. Operating specific issues are dealt with here as is the output of .rsu files. A sample file is provided with every implementation and the user needs only uncomment (i.e. remove the leading # on the line) those lines that they need from the file.

For instance, you may wish to keep a copy of everything you see on the screen during a SIBERIA run. To do this simply uncomment out the line starting ECHO. The file name is not allowed to have spaces in it. On a UNIX computer this is effectively equivalent to using output redirection (i.e. specifying a file name after > when you run the program).

To provide extra information on the variation of simulation properties in space and time a range of extra data output are allowed. These are output in a .rsu file at the same time that a .rst2 file is generated. Thus for every .rst2 file there is an .rsu file with the same name. The data in this .rsu file can be examined with EAMS-Viewer or most data visualization packages (e.g. IDL). The data that can be output are listed in the file and include

An example of the standard `siberia.setup` file follows. There are some options in `siberia.setup` that are not immediately obvious what they do. A general explanation is provided below.

- ECHO: this command opens a file with the specified name and then prints everything from the screen to this file. This is effectively a log of the SIBERIA run. Of most importance is that this file logs the time of the run and any WARNING messages output by the code. Typically WARNINGs do not terminate the operation of the code but may indicate an unexpected input that may change the results. If a superseded mode parameter is specified SIBERIA will generally output a warning but will continue with a replacement mode (which may or may not give exactly the same answer). ECHO_INCR appends a number to the specified filename, and increments this number for each run, ensuring that output files from previous runs are not overwritten.
- PAUSE_AT_END: When running from a double click on some windowing systems (Windows and MacOSX spring to mind) the window opened by the code when running is deleted immediately upon completion of the code. This open allows the window to remain open after a successful run. Unfortunately it will **not** keep a window open if the code crashes (if this is a problem you will need to open a command window and run SIBERIA from the command line).

- RST_OVERWRITE: By default SIBERIA does not typically overwrite an existing RST2 file (i.e. it will destroy output from a previous set of calculations). This allows the user to override this behaviour. If set ON then it will also overwrite existing RSU and LAYER files.
- NO_THREADS: For users running on single processor PCs this command can be ignored. SIBERIA has parallel computation capability using a protocol called openMP. Most parallel supercomputers, and/or the queuing system they use, require that the user specify how many CPUs will be used by the run. This command specifies how many CPUs to request.
- XYZ file: This option outputs a gridded elevation DEM in an text form at each time a RST output is requested. The format of the file is a series of lines with each line being a triple of the X,Y, Z coordinates of each node in the grid. It is useful visualisation and CAD packages where you only want elevation data in a simple format.
- OUTPUT & OUTPUT_BIN: This options specifies the output of simulation data in addition to what is output in the RST2 file. The file is a column format where is column corresponds to an OUTPUT command. If OUTPUT is specified then all that data is put into one file, the RSU file. If OUTPUT_BIN is specified then in addition to output in the RSU file the specified data is output into a binary file. Each different data type (corresponding to a individual OUTPUT_BIN command) is output into a separate file with a name constructed from the RST2 file and an abbreviation corresponding to that data set. The OUTPUT_BIN option is primarily provided because it is a convenient way to stream data into visualisation packages under UNIX and LINUX.
- A complete explanation of the OUTPUT capabilities is provided in the table below. Some of the output options are not operational unless the particular model option is turned on. For instance, to output soil depths you must have the soils model operational. Ditto for LAYER information. For the layer model the output options below are abbreviated. The user should look at the layer model for a complete list of layer output capabilities.

Table: Output Option Explanation

OUTPUT option	Explanation
OUTPUT OPTIONS THAT ARE ALWAYS AVAILABLE	
`SED_FLUX`	
`SED_FLUX_POTENTIAL`	
`SED_FLUX_ACTUAL`	
`YIELD`	The amount of material eroded/deposited at that node during the preceding timestep. Sign convention is positive for deposition (i.e. an increase in elevation). Units are the units of elevation.
`AVEYIELD`	The spatially averaged amount of material eroded/deposited in the catchment draining through that node during the preceding

	timestep. Sign convention is positive for deposition (i.e. an increase in elevation). Units are the units of elevation.
ZCHANGE	The cumulative change in elevation at that node over the duration of the simulation. Sign convention is positive for deposition (i.e. an increase in elevation). Units are the units of elevation.
AVEZCHANGE	The spatially averaged amount of material eroded/deposited in the catchment draining through that node over the duration of the simulation. Sign convention is positive for deposition (i.e. an increase in elevation). Units are the units of elevation. The catchment is defined based on the drainage analysis at the time of output.
GULLYPOT	The channel initiation function.
LOGGULLYPOT	The log (base 10) channel initiation function.
TONNESHECTARE	This is essentially YIELD converted to Tonnes/Hectare. This conversion assumes that sediment flux is expressed in cubic metres of sediment/metre width/timestep, the bulk density is expressed in tones/cubic metre, and the gridXY (i.e. grid spacing) is in metres.
AVETONNESHECTARE	As for TONNESHECTARE but using AVEYIELD rather than YIELD.
DOMAIN	This outputs a 0 if that node is outside the domain and 1 if the node is inside the domain. Useful if you want to filter other data in the RSU file (typically some form of automated statistical analysis) for whether that point is inside the domain or not.
DISCHARGE_MEANANNUAL	This is simply b3*area^m3.
SURFACE_B1	The erodibility of the landform surface. See also LAYER_1_B1 in the layer model.
SOILS MODEL OUTPUT (SOILS MODEL MUST BE OPERATIONAL)	
SOILMOISTURE	The soil moisture calculated using the topographic index and soil depth.
BEDROCK_Z	The elevations of the base of the soil layer (i.e. the bedrock surface)
BEDROCK_SLOPE	The slopes of the bedrock surface.
BEDROCK_AREA	The area draining through that node using the bedrock surface. The same algorithm used for area analysis of the landform surface is used for this calculation.
BEDROCK_DIRECTIONS	The flow directions using D8 for the bedrock surface.
LAYER MODEL OUTPUT (LAYER MODEL MUST BE OPERATIONAL)	
FLOW_B1	The erodibility of the sediment in the flow.
LAYER_NO	The number of soil layers at that node.

LAYER_1_B1	The erodibility of the top layer (i.e. 1st, surface layer). This is identical to SURFACE_B1 though SURFACE_B1 can be called without using the layer model.
LAYER_2_B1	The erodibility of the 2nd layer.
LAYER_3_B1	Ditto for the 3rd layer.
LAYER_4_B1	Ditto for the 4th layer.
LAYER_5_B1	Ditto for the 5th layer.
LAYER_1_Z	The elevation of the base of the top layer (i.e. 1st, surface layer).
LAYER_2_Z	Ditto for the 2nd layer.
LAYER_3_Z	Ditto for the 3rd layer.
LAYER_4_Z	Ditto for the 4th layer.
LAYER_5_Z	Ditto for the 5th layer.
LAYER_1_DEPTH	The depth of the top layer (i.e. 1st layer, surface layer). This is the vertical (i.e. measured in the vertical direction) distance from the base of the 1st layer to the surface. If there is no layer beneath it (i.e. there is only one layer) then this number is large (exact value implementation dependent).
LAYER_2_DEPTH	The depth of the 2nd layer. This is the vertical (i.e. measured in the vertical direction) distance from the base of the 2nd layer to the bottom of the 1st layer. If there is no layer beneath it (i.e. it is the bottom layer) then this number is large (exact value implementation dependent).
LAYER_3_DEPTH	Ditto for the 3rd layer.
LAYER_4_DEPTH	Ditto for the 4th layer
LAYER_5_DEPTH	Ditto for the 5th layer
DETACHMENT MODEL (LAYER MODEL MUST BE OPERATIONAL)	
LAYER_FLOW_DETACHMENT	The relative detachability of the sediment in the flow.
LAYER_1_DETACHMENT	The relative detachability of the top layer (i.e. 1st layer, surface).
LAYER_2_DETACHMENT	Ditto for the 2nd layer. If there is no 2nd layer then this is 0.
LAYER_3_DETACHMENT	Ditto for 3rd layer.
LAYER_4_DETACHMENT	Ditto for 3rd layer.
LAYER_5_DETACHMENT	Ditto for 3rd layer.
INTERNAL MODEL STATES (NOT NORMALLY NEEDED BY USERS)	
PREDICTOR_SED	This is the elevation change estimated by the predictor step in the fluvial erosion numerical solver.
CORRECTOR_SED	This is the elevation change estimated by the corrector step in the fluvial erosion numerical solver.

PREDCORRECT_SED_DIFF	This is simply PREDICTOR_SED-CORRECTOR_SED.
PREDCORRECT_SED_RATIO	This is simply PREDICTOR_SED/CORRECTOR_SED.
DINFWEIGHTS	The weights for the Dinfinity drainage direction solver.
STABILITY	This an estimate of the numerical stability of the fluvial erosion solver.

Sample `siberia.setup` File

The extract from a `siberia.setup` illustrates only the key aspects of the file. The file itself is fully documented in the file installed as part of the EAMS install. Note that not all the options are listed in this example. The user should refer to the version of `siberia.setup` in their EAMS install.

There is one aspect of this file that EAMS Viewer users should remember. If they are going to use the 'Difference file …' option in Viewer then the OUTPUT options for the two files must be exactly the same. They must have exactly the same output options and these options should appear in the `siberia.setup` file in exactly the same order. Viewer does not check that these conditions are met and if not the results of the differencing will be incorrect.

```
#
#
#                        HOW TO USE THIS FILE
#                       ----------------------
#
#    This file controls the operation of SIBERIA. Its name should always
#    be 'siberia.setup' (all in lower case on Unix or Mac OSX machines)
#    and it should be situated in the directory in which SIBERIA is
#    being run. If the file exists in that directory then SIBERIA reads
#    it automatically. If the file does not exist then SIBERIA simply
#    continues on without it, choosing default values where necessary.
#
#    To make this file easier to use all of the allowable commands are
#    listed below. The commands are the lines all in UPPER CASE while
#    the comments and explanations are the lines in lower case.
#
#    If a line starts in column 1 with either of # or ! character then
#    that line is treated as a comment and is ignored by SIBERIA. To
#    make a command active all you have to do is to uncomment the
#    appropriate line (i.e. remove the # or ! from the first column).
#    To inactivate it you simply add the # or ! to the first column
#    again. Explanations for the commands are provided immediately above
#    the commands.
#
#    There are a number of commands that turn some mode in the model on
#    or off. There are always three options for these modes. ON = (turn
#    that mode on), OFF = (turn that mode off), DEFAULT = (do whatever
#    the code decides is best in the circumstances). If you do not
#    enable one these three options then the code chooses DEFAULT
#    automatically. NOTE: the default action may not always be the same
#    as it may vary with size of the problem being solved, whether
#    SIBERIA detects that it is being run on a multiprocessor machine, etc,
#    so if you absolutely must have some form of behaviour then specify
#    it otherwise SIBERIA may run differently on different machines
#
#                        FILE REVISION HISTORY
#                       -----------------------
#          - updated for V8.28  5/ 4/2005 (GRW)
#
#
#
# ========================================================================
# -- To echo whatever is output to the screen to a file called (in the
```

```
#    example command below it is 'junk.output') uncomment the line
#    starting ECHO
# -- To NOT echo to a file uncomment the line starting NOECHO
# -- ECHO_INCR appends a unique number to the filename to ensure that
#    it doesn't overwrite the output file from previous runs of siberia
# ============================================================================
#
#ECHO junk.output
```

```
#NOECHO
#ECHO_INCR siberia.output
#
ECHO_INCR siberia.output
#
# ============================================================================
# -- To have the program halt at the end of the run without the window
#    automatically closing then uncomment the line PAUSE_AT_END ON.
# -- To have the window automatically close at the end of the run then
#    uncomment the line PAUSE_AT_END OFF
# -- To have the program do whatever its default behaviour is with the
#    window at the end of the run then uncomment line then uncomment
#    PAUSE_AT_END DEFAULT.
# ============================================================================
#
#PAUSE_AT_END ON
#PAUSE_AT_END OFF
#PAUSE_AT_END DEFAULT
#
PAUSE_AT_END ON
#
# ============================================================================
# -- To allow existing RST (and corresponding RSU amd LAYER) output files
#    to be overwritten uncomment the line RST_OVERWRITE ON.
# -- To stop RST output files from being overwritten uncomment the line
#    RST_OVERWRITE OFF
# -- To have the program do its default behaviour (typically this
#    is to NOT overwrite the RST files) uncomment RST_OVERWRITE DEFAULT.
# ============================================================================
#
#RST_OVERWRITE ON
#RST_OVERWRITE OFF
#RST_OVERWRITE DEFAULT
#
RST_OVERWRITE DEFAULT
# ============================================================================
#                       PARALLEL OPTIONS
#                       ----------------
#    This option is to set the maximum no of threads that the parallel
#    implementation of SIBERIA can use. The code will use this number
#    of threads and attempt to get that many number of processors from
#    the computer. This option is ignored if the standard serial version
#    of code is being used. If this option is not used then the code grabs
#    a default (typically small but > 1) number of processors. On shared
#    parallel supercomputers choosing a large number of threads may slow
#    the starting of the code until the requested number of processors
```

```
#    become available.
# ============================================================================
#
NO_THREADS 1
#
# ============================================================================
#                       EAMS COMPATIBILITY OPTIONS
#                       --------------------------
#  Output the elevation data in an xyz format (identical to the format
#  read by EAMS) in addition to the output in the .rst2 files. This option
#  is also useful in EAMS for output back to mine management and CAD
#  (e.g. AutoCad) packages.
# ============================================================================
#
```

```
#XYZ_FILE
#
#
#=============================================================================
#                        OUTPUT OF SUPPLEMENTARY CALCULATION DATA
#                        ----------------------------------------
#  The following OUTPUT commands provide supplementary information to what is
#  in the RST file. The data below are output in a .RSU file. A maximum of 10
#  datsets may be output.
#
#  There are two forms of the OUTPUT command
#   OUTPUT          : This outputs the specified data set into an RSU file
#                     which is a text column format used by all of the
#                     software in the EAMS suite and which is easily readable
#                     into data analysis programs (e.g. EXCEL, Kaleidograph,
#                     SigmaPlot).
#   OUTPUT_BIN      : In addition to the RSU file this form also outputs the
#                     dataset into a binary file (the filename is
#                     name.abbrev.bin where 'name' is the  same as the RST
#                     and RSU files, 'abbrev' is a self evident abbreviation
#                     for the dataset requested) that can be streamed into
#                     visualisation packages like IDL, EXPLORER etc.
#                     The format is 2 4byte integers (the x and y dimensions
#                     of the grid) followed by the data in 4byte floating
#                     point (by the x dimension first). Note if you request
#                     more than one dataset to be OUTPUT_BIN then each
#                     dataset requested goes into a separate file with the
#                     appropriate name.
# ============================================================================
#
# the amount of sediment being transported (cubic metres/timestep/m width)
# analytically derived from the transport equation.
#
#OUTPUT SED_FLUX
#
#
# the amount of sediment removed/timestep in units of height at any pt in
# the grid at the requested time.
#
#OUTPUT YIELD
```

```
#       .
#       .
#       .
# … More output options …
#       .
#       .
#       .
#       .
#
#                SOILS MODEL OUTPUTS
#                -------------------
#   The outputs that follow can only be output when the soils model is
#   turned ON.
#
#OUTPUT SOILMOISTURE
#
#   - Bedrock properties can only be output for ModeSoils=2
#
#OUTPUT BEDROCK_Z
#OUTPUT BEDROCK_SLOPE
#OUTPUT BEDROCK_AREA
#OUTPUT BEDROCK_DIRECTIONS
#
#
#
#                LAYERS MODEL OUTPUTS
#                -------------------
#   The outputs that follow can only be output when the layering model is
#   turned ON.
#
#
#   The B1 of the surface Layer (equivalent to OUTPUT LAYER_1_B1 except
#   SURFACE_B1 will give the surface B1 even if the layers model is not used)
#
#OUTPUT SURFACE_B1
#
#   The B1 of the flow
#
#OUTPUT FLOW_B1
#
#   Number of layers at that node
#
#OUTPUT LAYER_NO
#
#   Layer properties for the top 5 layers
#    ... note if the layer doesn't exist then zeros are output
#    ... OUTPUT LAYER_1_B1 is equivalent to OUTPUT SURFACE_B1 and is
#        provided to display B1 when layers is not used (SURFACE_B1 doesn't
#        need the layers model while LAYER_1_B1 does).
#
#OUTPUT LAYER_1_B1
#OUTPUT LAYER_2_B1
#
#OUTPUT LAYER_1_Z
#OUTPUT LAYER_2_Z
#
#                The Detachment-Limitation Model
#                ------------------------------
#   This model is only available in combination with the LAYERS module
#   so the output commands are a subset of the LAYERS commands (and LAYERS
```

```
#   model has to be activated to enable the detachment model)
#
#   The detachment rate for the material in the flow
#OUTPUT LAYER_FLOW_DETACHMENT
#
#   The detachment rate of the material for the various layers (note LAYER=1
#   is equivalent to the detachment rate for the surface)
#
```

```
#OUTPUT LAYER_1_DETACHMENT
#OUTPUT LAYER_2_DETACHMENT
#
#
#               INTERNAL MODEL STATES
#               ---------------------
#   These commands provide diagnostic output of the internal model
#   operations. They are primarily available to aid debugging of the
#   code operation and provided here as a memory aid for the developer.
#   It is not anticipated, nor recommended, that the user use these options.
#
#   States that control the stability/mass balance of the solver
#
#   Elevation changes of predictor
#
#OUTPUT PREDICTOR_SED
#
#   Elevation changes of corrector
#
#OUTPUT CORRECTOR_SED
#
#   Difference between elevation changes of predictor and corrector
#
#OUTPUT PREDCORRECT_SED_DIFF
#
#   Relative difference between elevation changes of predictor and corrector
#
#OUTPUT PREDCORRECT_SED_RATIO
#
#
#   The weights generated by the Dinfinity algorithm
#
#OUTPUT DINFWEIGHTS
#
#   A domain mask 0=outside computational domain, 1=inside computational
#   domain
#
#OUTPUT DOMAIN
```

4 SIBERIA Extended Models

4.1 General comments about SIBERIA Extended Models

A degree of flexibility has been provided in SIBERIA to allow the user to extend the various capabilities of the model in a simple fashion. The means by which this can be done is that the coefficients for the processes, or in some cases the processes themselves, can be calculated in external FORTRAN routines that can be modified, independently of the main program. These routines may have a number of different alternative models for determining the required data.

The upside of these extended models is that they allow a significant degree of flexibility in the operation of SIBERIA, beyond the standard simple models. The downside is that they (generally) require a file of input data that is separate from the RST2 files so it is possible to get very confused about what input data is used for any particular run. For instance, RST2 file only knows that another file has the data required to run the model, and its entirely possible that that file has been modified from the time at which that run was made.

If you are interested in using spatially variable erosion and/or runoff properties you should read the sections on the runoff, erosion and layering models carefully. Much work is going into enhancement of this capability and it is anticipated that the current erosion and runoff models will be replaced by the more general layering model. If possible use the layering model in preference to the runoff and erosion models, as it is likely in the future, when the layering model is fully operational, that the runoff and erosion models will be removed.

These extended capabilities are controlled by two parameters.

• **The first parameter** is an integer controlling parameter (`Mode`… in Section 3.2.1) that is passed to the routine by the program. This controlling parameter tells the routine which model in the routine to choose to determine the data to be sent back to the program. Some general rules need to be complied with. A value of 0 for the controlling parameter selects the default model for determining the parameters. In general, this is a spatially uniform and temporally constant process. Positive values of the controlling parameter are used for selecting standard capabilities that are or will be provided by the author in SIBERIA; these values are reserved and will be used in future version of SIBERIA. Negative values for the controlling parameter are used for selecting capabilities that may be coded by the individual user. Users are free to use any negative values they feel fit. Only if a negative value is chosen will SIBERIA call the user-defined subroutine – if 0 or positive numbers are input then the internal subroutines are called.

• **The second parameter** is a character string (the filename parameter in Section 3.2.3). This string can be used in any way the user sees fit. However, it is anticipated that this string will contain a filename that is used by the user defined module to input data (e.g. spatial data). This file could, in turn, contain further filenames if more than 1 file of data is required by the module. This filename is typically used for a filename (called a “model file”) by the standard models when a positive value for the controlling mode parameter is input (see description of standard models below). For some ideas about how these filenames can be used see the built in models in the section below.

The user defined function is called every time around the computation loop within the code, and the internal states of SIBERIA are passed out (e.g. elevation, slopes, drainage area, time, etc) for use by the user defined subroutine. These internal states MUST NOT BE MODIFIED, but can be used in the calculation of the module outputs. Note, that because of the internal numerics the subroutine may be called a number of times at the same simulation time, though the states will be different. This is not an error but a feature of the numerical solver in SIBERIA (see Section 2.6).

4.2 The Built-in Erosion Models

This module determines the value of $\square_1$ (Equation 2.1.5 and 2.1.6) to be applied at each point in the grid at point in time. The mode parameter for this model is `ModeErode`.

At some stage in the future this Model will be eliminated and be replaced by the layering model (see Section 4.9). The layering model is undergoing staged development, and it may not yet provide all the capability in this section. However, if your needs are satisfied by the layering model, it is recommended that you use the layering model in preference to this model.

ModeErode=0

This is the default fluvial erosion module. $\square_1$, m_1 and n_1 are spatially and temporally constant. The fluvial erosion rate in the channel is $K=\square_1$, and the fluvial erosion rate on the hillslope is $K=\square_1 O_t$.

$$q_s = Kq^{m_1}S^{n_1} \qquad (4.2.1)$$

If `ModeRandom=1` then K is multiplied by a random perturbation as well (see Section 3.2.1).

ModeErode=1

This is the depth dependent erosion model, a simple approximation to armouring reduction of the erodibility. The default erodibility, K, (see `ModeErode=0` for the definition of the default erodibility K) is adjusted by a factor that is dependent on the cumulative depth of erosion since the start of the simulation. The equation for the erodibility at that point is

$$K_{new} = \frac{K_{default}}{C_1(Z_e)^{C_2} + 1} \qquad (4.2.2)$$

where K_{new} is the value of K used in the modelling at that point, K_{new} is the value of K defined for `ModeErode=0` and C_1, C_2 are the coefficient and exponent of the erodibility reduction respectively, and Z_e is the depth of erosion at that node in the domain since the start of the simulation. The model ensures that for zero depth erosion the erodibility starts at the default value for erodibility but that as the depth of erosion increases the erodibility reduces asymptotically to zero.

To evaluate the cumulative depth of erosion, Z_e, the elevation at that point at the start of the simulation is simply subtracted from the current elevation at that point; positive values indicate erosion. Thus this model does not account for any uplift of the elevations by tectonic uplift. The model implicitly assumes that there is no tectonic uplift.

If there is net deposition at that spot in the catchment then no adjustment is made.

The parameters are input in a file specified by the file parameter –1, `FileErode`. The format is:

Line 1: A line that consists only of the string ‘ `SIBERIA`’. This is used to identify the file and ensure that it is a SIBERIA file.

Line 2-4: 3 title lines for the file

Line 5: C_1, C_2 in free format, separated by spaces.

A sample file follows

```
 SIBERIA
This is a sample file of the simple armouring model
Data for the gully experiment at Ranger 1997-1998
Run 3
10.0 1.0
```

ModeErode=3

This mode allows the input of regions that have different erosion properties. A region is input (using a RGN file) and then the parameters of the erosion process are input. These parameters are then applied over the whole of the region specified. All of this information is input in a file specified by the file parameter –1, `FileErode`. This file is called a model file. A sample model file is given below (this is the file '`test.model`' sample file). Note that the model file may include regional information for other processes (e.g. runoff variations in space) in addition to erosion. You might want to do that to keep everything organised (i.e. `FileErode` is then the same as `FileRunoff`) or you may wish to have erosion information in a separate file from runoff information (i.e. `FileErode` is then different from `FileRunoff`).

The format of the model file is as follows (note all lines starting with a # in the 1st column are comment lines … you may put comment lines anywhere):

Line 1: The string 'SIBERIA'. This line must be exactly as shown.

Line 2-4: These are 3 lines of title information. Anything may be entered here.

Subsequent lines: Commands that identify the erosion data to be input.

The general format of the erosion command is

`EROSION <modification keyword>` β_1 m_1 n_1 `<region filename>`

The `<modification keyword>` specifies how the values of β_1, m_1 and n_1 are to be modified. If it is `ABSOLUTE` then for every point on the grid covered by the region file then previous values of β_1, m_1 and n_1 are replaced by the new values. If it is `RELATIVE` then for every point on the grid covered by the region file the previous values of β_1, m_1 and n_1 are multiplied by the new values. Note that it is not possible to change the second fluvial process parameters.

The `<region filename>` is the file name of the region file that has the region to be modified. The filename should not include spaces.

The order of the commands in the model file is important. SIBERIA modifies the erosion parameters in the order in which the commands are presented in the file. If you have region files that overlap and you use `ABSOLUTE` then the order of modification will affect the final values for the erosion parameters.

```
SIBERIA
heading 1      ! the heading on the first line is required ... do not change!
heading 2      ! the 3 lines of heading are required and may contain
heading 3      ! anything (e.g. file identification).
#
#  -------------------------------------------------------------
#  This model file contains the information for  SIBERIA on regional
#  variations in runoff, erosion and tectonics.
#  -------------------------------------------------------------
#
#  The format of this file is as a series of example commands giving the
#  general format of the commands. The commands can be in order and there
#  are no limits on the number of commands in the file.
#
#  We suggest that you copy the appropriate command line examples and
#  edit copies so that you always have copies of the original correct
#  form of the command.
#
#  NB
#     1. All types commands are independent of each other so that runoff
#  and erosion commands can be entered independently. However, the runoff
#  commands are not independent of each other with subsequent runoff
#  commands working on the result of previous runoff commands if the
#  regions over which they apply are overlapping. The same is true of
#  the erosion commands. In particular, if you have regions
#  of different material you may change the runoff for that region
#  and not the erosion model if that is appropriate (and vice versa) or
#  you may change both if appropriate,
#  or you may change one 'absolute' and one 'relative' if desired.
#
#     2. These commands assume that the erosion model and the runoff model
#  are initially everywhere uniform and determined by the b1, m1, n1, b3,
#  n3 parameters specified in the RST2 file.
#
#                                ----------------
#
#  EROSION commands
#  ----------------
#
#   These commands are read in SIBERIA when the parameter UserFt=3 and
#   the file parameter=-1 (ie the filename for the ERODE parameter) is set
#   to this file.
#
#   The general form of the erosion commands from left to right is
#     - 'erosion' indicating  this is an erosion command (starts in column 1)
#     - one of either 'absolute' or 'relative' indicating
#          absolute: the erosion parameters are as given
#          relative: the erosion parameters given are multiplied with
#                   the erosion parameters at that point previously
#                   given . ie. this changes the parameters by a
#                   relative amount
#     - the parameters b1, m1, n1. For relative these are interpreted
#          as multipliers
#     - the region file for applying those parameters
#
#
#   The command below replaces the erosion model over the region defined
#   by the region file test1.rgn with an erosion model with
#   b1(new)=0.01, m1(new)=0.6, n1(new)=0.7
#
```

```
#EROSION ABSOLUTE 0.01 0.6 0.7 'test1.rgn'
#
#   The command below replaces the erosion model over the region defined
#   by test2.rgn with an erosion model with
#   b1(new)=0.05, m1(new)=0.2, n1(new)=0.3
#   NB. Because this command after the test1.rgn command where
#       test1.rgn and test2.rgn overlap test2.rgn overwrites test1.rgn
#
#EROSION ABSOLUTE 0.05 0.2 0.3 'test2.rgn'
#
#   The command below modifies the erosion model over the region defined
#   by test3.rgn with an erosion model with b1(new)=b1(old)*0.1
#                               m1(new)=m1(old)*0.7
#                               n1(new)=n1(old)*1.2
#   NB. Because this command after the test1.rgn and test2.rgn commands where
#       test1.rgn, test2.rgn and test3.rgn overlap test3.rgn overwrites
#       the other files
#
#EROSION RELATIVE 0.1 0.7 1.2 'test3.rgn'
#
#
#   RUNOFF commands
#  -----------------
#
#   These commands are read in SIBERIA when the parameter UserRO=3 and
#   the file parameter=-2 (ie the filename for the RUNOFF parameter) is
#   set to this file.
#
#   The general form of the erosion commands from left to right is
#     - 'runoff' indicating  this is an runoff command (starts in column 1)
#     - one of either 'absolute' or 'relative' indicating
#          absolute: the runoff parameters are as given
#          relative: the runoff parameters given are multipLied with
#                   the runoff parameters at that point previously
#                   given .... ie. this changes the parameters by a
#                   relative amount
#     - the parameters b3, m3. For relative these are interpreted
#       as multipliers
#     - the region file for applying those parameters
#
#
#
#
#
#   The command below replaces the runoff model over the region defined
#   by test0.rgn with a runoff model with
#   b3(new)=1.0, m3(new)=0.8
#
#RUNOFF ABSOLUTE 1.0 0.8 'test0.rgn'
#
#
#   The command below updates the runoff model over the region defined
#   by test1_2.rgn with a runoff model with
#   b3(new)=b3(old)*2.0, m3(new)=m3(old)*0.9
```

```
#
#RUNOFF RELATIVE 2.0 0.9 'test1_2.rgn'
#
#
#   The command below updates the runoff model over the region defined
#   by test0.rgn with a runoff model with
#   b3(new)=b3(old)*0.5, m3(new)=m3(old)*1.0
#
#RUNOFF RELATIVE 0.5 1.0 'test0.rgn'
```

ModeErode=20 and greater

In this case the β_1 parameter is interpreted as being the rate parameter/unit width of hillslope. The parameter is then internally converted within SIBERIA to the rate parameter for the grid resolution adopted using the value of `GridXY` input. All the `ModeErode` models for less than 20 are still available (simply subtract 20 from `ModeErode` to determine the models from above). **Note** that once one of `ModeErode or ModeRunoff` is given a value greater than 20 **all** parameters are assumed to be given in rates/unit width.

4.3 The Built-in Runoff Models

This module determines the coefficient β_3 to be applied at each point in the grid and at each point in time. The mode parameter for this model is `ModeRunoff`.

At some stage in the future this Model will be eliminated and be replaced by the layering model (see Section 4.9). The layering model is undergoing staged development, and it may not yet provide all the capability in this section. However, if your needs are satisfied by the layering model, it is recommended that you use the layering model in preference to this model.

ModeRunoff=0

This is the default fluvial erosion module. Both β_3 and m_3 are spatially constant.

$$q = \beta_3 A^{m_3} \qquad (4.3.1)$$

ModeRunoff=1

This mode allows for the input of a spatially variable, but constant in time, runoff. The spatial variability of runoff, β_3, is determined by

$$\beta_3(x,t) = \overline{\beta}_3 \, \square_3(x,t) \qquad (4.3.2)$$

where $\overline{\beta_3}$ is the value of β_3 input in the parameter list and $\beta_3(x,t)$ is the multiplier on $\overline{\beta_3}$ to yield the spatially varying runoff. Note that the runoff at any point is determined by the equation

$$Q = \beta_3(x,t)\ A^{m_3} \tag{4.3.3}$$

The file that has the multipliers in it is specified in `FileRunoff` (file parameter –2, Section 3.2.3). The format of this file is

line 1: The header string ‘ SIBERIA RUNOFF’ starting column 2 (i.e. the first column is blank).

lines 2-4: 3 lines of information about the file. You may input anything you like.

line 5: The x and y dimensions of the runoff grid (this will be interpolated to the size of the computational grid so these numbers can be different from kx and ky).

line 6 and beyond: The runoff data starting at point (1,1) reading the x dimension fastest (i.e. columnwise for a matrix).

The first time the subroutine is called the runoff data is read and interpolated onto the computational grid. The input data is scaled and interpolated as necessary so that the data will fit onto the computational grid. In subsequent calls the runoff of individual nodes is summed to yield the average runoff over the area draining to that node.

Note that a significant computational penalty is paid on supercomputers for `ModeRO`=1 because some unvectorisable, unparallelisable algorithms need to be calculated at each timestep. On a workstation a penalty (though somewhat lesser) will still be paid with the average runoff calculations that need to done at each timestep being approximately doubled.

ModeRunoff=2

This mode allows you to specify known runoffs from offsite into a node. Thus if you have a large flow coming from offsite and you are not unduly concerned about modelling the evolution of that part of the catchment then you may simulate it as an offsite inflow node. For instance, we have simulated a site of about 2 sq km that had a large river flowing across the bottom corner that drained about 100 sq km. We modelled that large river as an offsite inflow node.

The file that has the multipliers in it is specified in `FileRunoff` (file parameter –2, Section 3.2.3). The format of this file is

line 1: The header string ‘ SIBERIA RUNOFF’ starting column 2 (i.e. the first column is blank).

lines 2-4: 3 lines of information about the file. You may input anything you like.

line 5: The number of offsite inflow nodes that are in the file.

line 6 and beyond: One line is input for offsite inflow node. Each line consist of 4 numbers. The first two numbers are the (x,y) coordinates on the grid. The Third number is the area of the inflow. The fourth number is the slope of the inflow.

The code then calculates the inflow discharge into the grid by use of β_3 and m_3 and the default runoff equation. The inflow sediment transport is also calculated by use of β_1, m_1 and n_1 and the default sediment transport equation using the input area and slope. The elevation change of the code at which the inflow is specified is then calculated by the imbalance between the sediment inflow and the outflow from that node in the same way that it is done in all of the other nodes of the domain. Note that even though the elevation of the inflow node may change during the simulation (and then perhaps the slope draining into that node and upstream), you cannot change the inflow slope specified for the inflow node.

No checking is done that the inflow nodes are on the edge of the domain so it is possible to input sediment and runoff into the centre of the domain. That may be useful if sediment is being transported into the centre of the domain as is done in a tailings dam.

ModeRunoff=3

This mode allows the input of regions that have different runoff properties. A region is input (using a RGN file) and then the parameters of the runoff process are input. These parameters are then applied over the whole of the region specified. All of this information is input in a file specified by the file parameter –2, `FileRunoff`. This file is called a model file. A sample model file is given in the erosion model section for `FileErode`=3 (this is the file '`test.model`' sample file). Note that the model file may include regional information for other processes (e.g. erosion variations in space) in addition to runoff. You might want to do that to keep everything organised (i.e. `FileRunoff` is then the same as `FileErode`) or you may wish to have erosion information in a separate file from runoff information (i.e. `FileRunoff` is then different from `FileErode`).

The format of the model file is as follows (note all lines starting with a # in the 1st column are comment lines … you may put comment lines anywhere):

Line 1: The string 'SIBERIA'. This line must be exactly as shown.

Line 2-4: These are 3 lines of title information. Anything may be entered here.

Subsequent lines: Commands that identify the runoff data to be input.

The general format of the erosion command is

```
RUNOFF <modification keyword> β3 m3 <region filename>
```

The `<modification keyword>` specifies how the values of β_3 and m_3 are to be modified. If it is `ABSOLUTE` then for every point on the grid covered by the region file then previous values of β_3 and m_3 are replaced by the new values. If it is `RELATIVE` then for every point on the grid covered by the region file the previous values of β_3 and m_3 are multiplied by the new values.

The <region filename> is the file name of the region file that has the region to be modified. The filename should not include spaces.

The order of the commands in the model file is important. SIBERIA modifies the erosion parameters in the order in which the commands are presented in the file. If you have region files that overlap and you use ABSOLUTE then the order of modification will affect the final values for the erosion parameters.

ModeRunoff=20 and greater

In this case the β_3 parameter is interpreted as being the rate parameter/unit width of hillslope. The parameter is then internally converted within SIBERIA to the rate parameter for the grid resolution adopted using the value of GridXY input. All the ModeRunoff models for less than 20 are still available (simply subtract 20 from ModeRunoff to determine the models from above). **Note** that once one of ModeErode or ModeRunoff is given a value greater than 20 **all** parameters are assumed to be given in rates/unit width.

4.4 The Built-in Tectonics Models

This module determines the tectonic uplift to be applied at every node in the grid. The mode parameter for this model is ModeUplift.

ModeUplift=0

This is the default tectonics module. The uplift is spatially constant, and does not respond to isostasy. The uplift has two components that are simply added together (1) a time invariant uplift parameterised by Zinit and notch, and (2) a time varying uplift parameterised by Tamp, Tphase, Tperiod. For more detail see Section 2.2.

ModeUplift=1

This is identical to ModeUplift=0. If ModeUplift=0 is specified then the model defaults to ModeUplift=1. The time varying uplift function is sinusoidal.

ModeUplift=2

This is similar in concept to ModeUplift=1. The time varying uplift function is a square wave with the centre of the square wave coinciding with the peaks of the sinusoid.

ModeUplift=3

This is similar in concept to `ModeUplift=1`. The time varying uplift function is a impulse function with the impulses coinciding with the positive peaks of the sinusoid in `ModeUplift=1` . The magnitude of the pulse uplift is `Tamp`.

4.5 The Built-in Drainage Directions and Contour Banks Models

This is an extension to SIBERIA that requires the input of a file for some modes and controls the way water is deemed to have flowed within the domain. The mode parameter for this model is `ModeDir`. Unless otherwise noted they are based on the D8 algorithm (i.e. flow is deemed to flow to the node point for which the downslope slope is steepest).

ModeDir=0,1

These are the standard modes described in Section 3.2.1. Drainage directions are generally unconstrained and drainage proceeds in the steepest downslope direction. For `ModeDir=1` a channel is not allowed to change directions if that means that it will change from flowing into a channel to flowing onto a hillslope. `ModeDir =0` allows this behaviour.

ModeDir=2

This mode allows the input of contour banks. In this case the filename #4 (`FileCtrBank` in Section 3.2.3) contains the information of the allowable drainage directions for each node. The file format consists of

Line 1: A header that says `' SIBERIA DIRECTIONS'`. If this line is not included then the file is rejected as not being a valid input file of allowable drainage directions.

Lines 2-4: 3 lines of titles and comments for the file.

Line 5: The x and y dimensions of the grid of data in the file. These must exactly match the values of `kx` and `ky` otherwise the file is rejected.

Line 6 onwards: The allowable drainage directions. Each line consists of 8 numbers, each representing one of the 8 flow directions from that point (see Figure 3.1), either 0 or 1 in the format like '00011011' with no spaces between the numbers. A 1 indicates that direction is allowed, while a 0 indicates that direction is not allowed. The order of the directions from 1 to 8 is given by the array `DIR` in the FORTRAN include file `DIRDEFN.INC` (i.e. The 1st direction is direction 3 in Figure 3.1, then proceeding anticlockwise). The x dimension is read first in the increasing direction down the file.

ModeDir=3

In this mode the drainage directions in the simulation are as specified in the input rst2 file. They do not change with time.

ModeDir=4

This is a highly optimised version of `ModeDir`=0. This is about 50% faster than `ModeDir`=0, with about a 15% speedup of the overall code performance on a typical RISC based machine.

ModeDir=5

This is crude implementation of the random 8 direction algorithm (i.e. R8). The weighting scheme is simply in proportion to the downstream slope (i.e. a direction twice as steep will be weighted twice as highly).

ModeDir=6

This is crude method to allow for non local slopes (perhaps I should call it D24). Instead of looking at simply the 8 adjacent nodes for the steepest downslope direction it looks at the next 16 nodes that are the boundaries of the box two nodes out. An average slope of the 24 points surrounding the node are examined in the 8 directions and the flow allocated to the directly adjacent node that has the steepest downslope direction as accounted for by the 24 surrounding nodes. This has only been used once in Hancock (1997) as a crude way of allowing for momentum effects where the direction of flow is not simply dependent on the local slope conditions but on more wide scale conditions.

ModeDir=7,8

These modes are currently under active development, and cannot at this time be guaranteed bug-free. These modes select the Tarboton (1997) D∞ multiple flow drainage direction algorithm.

4.6 The Built-in Channel Models

This model determines how the depth of the channel is determined. The main use of this is the slope in the channel is determined by examining the nominal elevation (the surface elevation at that node determined by SIBERIA) minus the channel depth. On the hillslope the nominal elevation simply the surface elevation determined by the model. For a node flowing from the hillslope into a channel the upstream elevation is the surface elevation, while the downstream elevation is the surface elevation minus the channel depth. The mode parameter for this model is `ModeChannel`.

ModeChannel=0

In this mode channels are mapped in space but their dimensions are not calculated. The slope in a channel is the slope based on the difference in the elevations between adjacent nodes. Thus slopes in channels are calculated in exactly the same way as for hillslopes.

ModeChannel=1

The depth of the channel is calculated with regime equation

$$depth = b_6 A^{m_6} \qquad (4.6.1)$$

The slope in a channel is the slope is then the difference between the (elevations-channel depth) between the adjacent nodes. Thus slopes in channels are calculated differently in channels compared with hillslopes.

ModeChannel=2

The depth of the channel is calculated with regime equation

$$depth = b_6 Q^{m_6} \qquad (4.6.2)$$

The slope in a channel is the slope is then the difference between the (elevations-channel depth) between the adjacent nodes. Thus slopes in channels are calculated differently in channels compared with hillslopes.

ModeChannel=3

In this mode the sediment transport in channels and hillslopes is calculated ignoring the parameter O_t. Thus the erodibility of both the hillslopes and channel is identical, irrespective of what value of O_t is input. No channel calculations at all are calculated so the code is a little faster in this mode than the other modes.

4.7 The Built-in Soil Models

These models are as implemented by Saco, Willgoose and Hancock (2005) (in review, Journal of Geophysical Research) and are as yet undocumented.

4.8 The Generic Dependent Model

This module is to allow for the calculation of any variable that may be dependent on the output from SIBERIA but which in itself does effect SIBERIA calculations (e.g. soil water values, vegetation, etc). However, conceivably a user defined erosion module could be linked to a user defined vegetation module leading to powerful and general modules not anticipated in the original construction of SIBERIA. There are no standard modules for this case at this time so only negative values of `ModeDP` are valid.

4.9 The Built-in Layers Models

The layering model is a major project that was first implemented in SIBERIA V8.28. In the future it will eventually replace `ModeErode=3` and `ModeRunoff=3` as it allows more realistic modelling of erosion and runoff physics, particularly for cases where deposition of eroded sediment is a key characteristic of the simulations.

4.9.1 The Science underpinning the Layering Model

The geology is represented by a series of layers of material with specified characteristics (e.g. erosion, runoff, etc). A series of capabilities are provided for initial specification of layers and their properties. In V8.28 only *fluvial erosion* properties change from layer to layer but in future version non-fluvial erosion and also runoff properties will be included. The soils model cannot be used at the same time as the layering model.

As the landform erodes the model tracks in the flow the characteristics of the material being transported. This has the consequence that the material being transported by the flow determines the transport capabilities of the flow. The characteristics of the flow reflect the mixing of material being transported from upstream and the material being eroded at that point. Since at the current time SIBERIA determines erosion characteristics based on a characteristic diameter of the eroded sediment (e.g. d50) then a variety of mixing models are provided to allow simulation of different kinds of diameter dependency of sediment transport. (NB It is anticipated that at some later date SIBERIA will be extended to allow modelling of a grading distribution for the sediment).

When deposition occurs the characteristics of the material being deposited are those of the material in the flow at that point and time. Deposition is assumed to be instantaneous.

Since the characteristics of the material being deposited typically change over time and space (and with cumulative upstream erosion exposing new layers, or eroding previously deposited material), the changing characteristics of the deposited material are tracked and a profile of layers of deposited material is created at that point

If an area of previous deposition is eroded at some later stage of the evolution then the characteristics of the entrained sediment are those of the layer currently being eroded. As the layers are eroded the characteristics of the entrained sediment change to reflect the current layer being eroded.

The layers are applied, deposited, eroded, etc at each point independently of any other point. The model does not *impose* any spatial layering structure (i.e. linking of a layer at one point with some layer at another point) as deposition simply reflects deposition characteristics at each individual point. However, since sediment characteristics change slowly as your proceed down a drainage path there is likely to be some spatial pattern to deposited sediment. This reflects the physics, not any structure imposed by the layering model.

The user may be confused that an apparent exception to this lack of explicit spatial coupling is in the input of new layers in the layer model file at the start of the run. In many cases it is convenient to create a layer in the initial landform that reflects some sort of spatial pattern (e.g. putting a layer of capping material on the surface of a landform to protect it). The commands are designed to make this type of data input easy BUT once these layers are input the model ignores this implicit spatial coupling and treats each point in the domain independently.

Layers are modeled to a specified resolution (the maximum layer depth). In general deposited sediment is mixed into the surface sediment layer until the layer resolution is reached at which time a new layer is created on the surface which sediment then deposits into. Thus in general the stratigraphy generated by SIBERIA will be a series of layers of depth equal to the maximum layer depth, and each layers properties reflect the average of all the sediments that were deposited into that layer. The only exception to this rule is when a large amount of homogeneous sediment is deposited in one single timestep. In this case because the sediment is homogeneous the deposited layer is allowed to exceed the maximum layer depth.

Two transport models are provided:

- Transport-Limitation: This is conceptually based on the link being sediment grading and transport capacity of the flow. As the grading of the sediment in the flow gets coarser then the transport capacity decreases and the ability to eroded material from the surface decreases, and vice versa.
- Detachment-Limitation: The transport limitation model is used and the "potential" entrainment rate is calculated. The detachment limitation model then provides a relative detachment reduction factor for the entrainment rate. If this factor is for example 0.1 then for the sediment eroded at that point and time the potential transport limited entrainment rate is calculated and the actual entrainment rate is 0.1 times that potential rate.

4.9.2 The Transport Limited Model

The default transport limited model operates by interpreting the b_1 input as the coefficient on the transport limited capacity of the material on the surface. The actual transport capacity of the flow is then calculated by mixing the b_1 from the sediment entrained at the surface and the b_1 of the sediment being transported by the flow from upstream. The process of determining the b_1 result from that mixing is based on b_1 being determined by the transport capacity of the non-cohesive granular flow where the difference in b_1 solely reflect the differences in the grading of the sediment in the flow. A simple model for the relationship between b_1 and the sediment grading is

$$b_1 = \frac{1}{d_{50}^{\square}}$$

where $\alpha = 0.75$ and d_{50} is the median size of the sediment grading distribution. More complex expressions have been published but to first order are mostly of this form. Note that this expression is a function only of the d_{50} of the flow. If for a moment we consider two masses of sediment, M_1 and M_2, with mean diameters $\bar{d}_1$ and $\bar{d}_2$ the average diameter of the sediment after they are mixed is

$$\bar{d}_{mix} = \frac{M_1\bar{d}_1 + M_2\bar{d}_2}{M_1 + M_2}$$

and if we note that if there is constant relationship between the $\bar{d}$ and d_{50} pre- and post-mixing (i.e. $\bar{d} = Kd_{50}$ with constant K) then this mixing relationship applies also for d_{50}. This allows us to write a mixing equation for b_1 as follows

$$\beta_{mix} = \left[\frac{M_1 + M_2}{M_1\left(\frac{1}{\beta_1}\right)^{\frac{1}{\alpha}} + M_2\left(\frac{1}{\beta_2}\right)^{\frac{1}{\alpha}}} \right]^{\alpha} = \beta_2\left[\frac{M_1 + M_2}{M_1\left(\frac{\beta_2}{\beta_1}\right)^{\frac{1}{\alpha}} + M_2} \right]^{\alpha}$$

where the β_{mix} is the erodibility coefficient after the mixing of the two types of sediment. Note that the only assumption here is the constant relationship between $\bar{d}$ and d_{50}.

4.9.3 Computational Notes

This maximum layer depth can be controlled by the user though the user should be aware that layer storage can take an enormous of memory, and a high resolution will likely result in out-of-memory errors during the run (particularly for 32 bit operating systems like Windows), or very long run times (if the consumed memory exceeds the physical memory in your computer and virtual memory is triggered). If virtual memory is not triggered then the runs times are largely independent of the number of layers.

Relative to a run without layers the transport limited models adds about 20-30% to run times, while requesting detachment limitation as well adds a further 20-30%.

4.9.4 The Layering Model File

The general format of the model file follows that of the model files for `ModeErode`=3 and `ModeRunoff`=3 discussed in the sections above. The layering model is used for erosion when `ModeErode`=4. For completeness some of this introduction for `ModeErode`=3 is repeated below.

The layer model is read whenever a filename is given for `FileLayer`. The trigger the use of layering data for the erosion model you should input `ModeErode`=4 (in V8.30 there is no way to trigger its use on the runoff model though this will be implemented in future versions). There are no further modes for control of how the layering model works. Instead the layering model is controlled by commands within the layer model file. A sample model file is given below (this is the file '`layer.model`' sample file). Note that the model file may include information for other models (e.g. erosion, runoff variations in space) in addition to layering information, and where requested will read this data as well as the specific layer commands You might want to do that to keep everything organised (i.e. `FileLayer` is then the same as `FileErode`) or you may wish to have erosion information in a separate file from runoff/erosion information (i.e. `FileRunoff` is then different from `FileErode`). Eventually in the future version of SIBERIA the layer model file will supercede the use of erosion and runoff model files but this will only occur when the required capabilities are fully built into the layering model in SIBERIA.

NOTE: For compatibility with EAMS if `ModeErode`=4 then IF `FileLayer` is left blank (i.e. no layer model file specified) THEN SIBERIA will try to find layer commands in the erosion model file `FileErode`. If `FileErode` is also blank (i.e. no erosion model file specified) then SIBERIA will report an error. If `FileErode` is not blank then NO error will be reported even if there are no layer commands in the erosion model file.

The format of the model file is as follows (note all lines starting with a # in the 1st column are comment lines … you may put comment lines anywhere):

- **Line 1:** The string 'SIBERIA'. This line must be exactly as shown starting in column 1.
- **Line 2-4:** These are 3 lines of title information. Anything may be entered here.
- **Subsequent lines:** Commands that identify the runoff data to be input.

The general format of a layer command is

```
LAYER <keyword> <data>
```

The `<keyword>` specifies what the layer command is. The `<data>` is data used by that layer command with the specifics varying from command to command as listed below. The specifics will be explained below in the discussion of the commands.

The order of the commands in the model file is important. SIBERIA interprets the layer commands in the order in which the commands are presented in the file. Thus one layer command may change the behaviour of the subsequent commands. In fact it is this very order dependence that provides the power of the layer commands that are described below.

Note that some commands in the file example are noted as being "not operational in the layering model of V8.30". These notes indicate commands that will be implemented shortly, which will be read correctly by V8.30 but which do not do anything in V8.30.

```
SIBERIA EXTERNAL
    This is a siberia model input file. The first line of the file
    above is fixed and should not be edited. These three lines
    are for file description data and can be modified by the user
#
#
==============================================================================
#       This file contains the extended models information for SIBERIA
#      (including regional variations in runoff, erosion, tectonics
#       ,aggradation and layering)
==============================================================================
#
#  The format of this file is as a series of example commands giving the
#  general format of the commands. The commands can be in any order and there
#  are no limits on the number of commands in the file. Each command is a
#  single line of information.
#
#  We suggest that you copy the appropriate command line examples and
#  edit copies so that you always have copies of the original correct form
#  of the command.
#
#  NB 1. All commands are independent of each other so that runoff and
#     erosion commands can be entered independently. However, the runoff
#     commands are dependent on each other with subsequent runoff commands
#     working on the result of previous runoff commands if the regions over
#     which they apply are overlapping. The same is true of the erosion
#     commands, tectonics, layers, etc.
#
#     In particular, if you have regions of different material you may
#     change the runoff for that region and not the erosion model if that
#     is appropriate (and vice versa) or you may change both if appropriate,
#     or you may change one 'absolute' and one 'relative' if desired.
#
#     2. These commands assume that the erosion model and the runoff model
#     are initially everywhere uniform and determined by the b1, m1, n1,
#     b3, n3 parameters specified in  the RST2 file.
#
#                     Compatible with SIBERIA V8.30
#                     ==============================
#
#
=============================================================================
#                          EROSION commands
#
=============================================================================
#
#
#                   The EROSION commands come in two forms:
#                   --------------------------------------
#
#                          GENERAL EROSION MODEL
#                          ---------------------
#
#       Any commands for the general erosion model (i.e. commands starting
```

```
#       with EROSION) are currently ignored in SIBERIA V8.30. In some
#       future version of SIBERIA they will be interpreted.
#
#
#                     ERODIBILITY ONLY EROSION MODEL
#                     ------------------------------
#
#   A more specific form of the erosion command only modifies the erodibility
#   and is of the form
#     - 'ERODIBILITY' indicating  this is an erosion command (starts in
#       column 1)
#     - one of either 'ABSOLUTE' or 'RELATIVE' indicating
#          ABSOLUTE: the erosion parameters are as given
#          RELATIVE: the erosion parameters given are multipied with
#                   the erosion parameters at that point previously
#                   given .... ie. this changes the parameters by a
#                   relative amount
#     - the parameter b1. For RELATIVE these are interpretted as multipliers
#     - the region file for applying those parameters (it must inside '')
#
#   The command below modifies the erosion model over the region defined
#   by test3.rgn with an erosion model with b1(new)=b1(old)*0.1
#
#ERODIBILITY ABSOLUTE 0.001 'control\test.rgn' (not operational in the
#                                              layering model in V8.30)
#
#
#ERODIBILITY RELATIVE 0.1 'd:junk.rgn' rgn'   (not operational in the
#                                              layering model in V8.30)
#
#
#
#
============================================================================
#                           RUNOFF commands
#
============================================================================
#
#
#        In SIBERIA V8.30 runoff commands in the layer model are
#        are ignored (i.e commands starting with RUNOFF).
#        However, it is anticipated that in future
#        versions of SIBERIA they will be interpreted (in a currently
#        unanticipatable way … but capturing the intent of the commands).
#
#
============================================================================
#                           CHANNEL commands
#
============================================================================
#
#
#         Any channel commands (i.e. commands starting with CHANNEL)
#         in the layer model file are ignored. It is not anticipated they
#         be read by any future version of the layering model in SIBERIA.
#
```

```
#
============================================================================
#                          UPLIFT commands
#
============================================================================
#
#
#         Any uplift commands (i.e. commands starting with UPLIFT)
#         in the layer model file are ignored. It is not anticipated they
#         be read by any future version of the layering model in SIBERIA.
#
#
============================================================================
#               KNOWN AGGRADATION/DEGRADATION rate commands
#
============================================================================
#
#
#         Any aggradation/degradation commands (i.e. commands beginning
#         with AGGRADATION) in the layer model file are ignored. It is
#         not anticipated they be read by any future version of the
#         layering model in SIBERIA.
#
#
============================================================================
#                         LAYERS MODULE commands
#
============================================================================
#
#  Whenever a LAYERS file is specified the file is read for all commands
#  that start with LAYERS. ALL lines starting with other commands (e.g.
#  RUNOFF, UPLIFT) are ignored, other than what is specified above.
#
#  LAYER commands can be divided into four kinds.
#
#    - LAYER CONTROL. These commands allow the user to control the internal
#         computational behaviour of the LAYER model (e.g. maximum thickness
#         of layers)
#    - LAYER PARAMETERS. These commands provide information about the
#         erosion/runoff model parameters that are to be used for subsequent
#         LAYER commands (or until they are superseded by a new LAYER
#         parameter command).
#    - LAYER CREATION. These commands input the elevation properties of
#         the layer being created. The properties of the LAYER being
#         created are those input by the most recent LAYER PARAMETER
#         commands.
#    - LAYER MASKING. These commands input information on the spatial
#         extent of the LAYER currently being created. These commands
#         allow you to create a LAYER that is restricted in spatial extent
#         so that it doesn't have to cover the entire computational domain.
#
#
#   LAYER CONTROL COMMANDS
#   ---------------------
#
#   - the maximum thickness of a layer created by SIBERIA during deposition.
#     This does not preclude the user from inputting a thicker layer but
#     all layers generated by the computations will have a maximum
#     thickness as below.
#
```

```
#LAYER THICKNESS 0.1
#
#     Relative Detachment Rate
#     =========================
#     Turn detachment limitation ON. Once ON it cannot be turned off.
#     NOTE: If you wish to use the detachment model you MUST set it ON
#     before you input any layers otherwise the layers input before this
#     command will have the default detachment rate. (Default is 1.0)
#
#LAYER DETACHMENT ON
#
#     These commands set the detachment rate. NOTE: You must set these values
#     before you input layers otherwise the layers will have the
#     default detachment rate
#
#LAYER DETACHMENT RELATIVE 0.6
#LAYER DETACHMENT ABSOLUTE 0.01
#LAYER DETACHMENT DEFAULT
#
#
#   LAYER PARAMETER COMMANDS
#   ----------------------
#
#   - the general format of these commands is the same as the EROSION/RUNOFF
#     models for region input at the top of this file, except that a
#     region file is not input as part of the command line. For instance,
#     for EROSION parameters input is:
#     - 'EROSION' indicating this is an erosion command (starts in column 1)
#     - one of either 'ABSOLUTE' or 'RELATIVE' indicating
#          ABSOLUTE: the erosion parameters are as given
#          RELATIVE: the erosion parameters given are multipied with
#                    the erosion parameters at that point previously
#                    given .... ie. this changes the parameters by a relative
#                    amount (initially the parameters are those set in
#                    the parameters input at the start)
#          DEFAULT: sets the parameters back to the default values (those
#                    input in the parameters input at the start). This can
#                    be handy for resetting parameters when you have input
#                    lots of layers with RELATIVE parameters.
#     - the parameters. For RELATIVE these are interpretted as multipliers
#       on the last value for the parameters.
#
#
#   - Erosion model parameters 'b1,m1,n1' (not operational in V8.30)
#LAYER EROSION RELATIVE 0.1 0.2 0.3
#LAYER EROSION ABSOLUTE 1.0 2.0 3.0
#LAYER EROSION DEFAULT
#
#   - Erodibility parameter 'b1'
#LAYER ERODIBILITY RELATIVE 0.1
```

```
#LAYER ERODIBILITY ABSOLUTE 0.3
#LAYER ERODIBILITY DEFAULT
#
#
```

```
#   - Runoff model parameters 'b3,m3' (not operational in V8.30)
#LAYER RUNOFF RELATIVE 0.5 0.95
#LAYER RUNOFF ABSOLUTE 0.1 0.5
#LAYER RUNOFF DEFAULT
#
#   - Maximum slope parameter 's0max' (not operational in V8.30)
#LAYER ANGLE_OF_REPOSE RELATIVE 0.5
#LAYER ANGLE_OF_REPOSE ABSOLUTE 0.2
#LAYER ANGLE_OF_REPOSE DEFAULT
#
#   - Creep parameter 'dZ' (not operational in V8.30)
#LAYER CREEP RELATIVE 0.6
#LAYER CREEP ABSOLUTE 0.01
#LAYER CREEP DEFAULT
#
#
#   LAYER ELEVATION COMMANDS
#   ------------------------
#
#   - the commands for input of the elevations for the layers. NB the input
#     elevations are for the base of the layer unless otherwise noted
#     (i.e. the layer extends upwards from the elevations input.
#
#   - a layer covering the surface of thickness given. The top of the
#     capping layer is the landform surface and the bottom of the capping
#     is 'thickness' below that surface
#
#LAYER CAPPING 2.1
#
#   - a layer with base horizontal and with the layer's base at the
#     elevation given
#
#LAYER Z 25.2
#
#   - a layer that is a bilinear spline. the 1st 4 values are the
#     coordinates of the four corners of the rectangular region (in node
#     coordinates), and the 2nd four values are the elevations at the
#     four corners in the order SW, SE, NW, NE corners. NB. the spline
#     extends over the whole domain and the corner coordinates are
#     ONLY used to determine the elevations of the interpolated/extrapolated
#     surface not the spatial extent of the surface
#
#LAYER BILINEAR 1 20 1 30 10.0 20.0 22.0 35.0
#
#   - a layer that has as its base the elevations as read from the rst2 file,
#     with the elevations offset by the specified value (i.e. a negative
#     offset is a lower elevation). NB the areal extent of the RST2 file
#     must match the RST2 file of the landform DEM.
#
#LAYER DEM -2.1 'TEST.RST2'
```

```
#
#
#   LAYER MASKING COMMANDS
#   ----------------------
```

```
#
#   - these commands specify over what part of the domain the layer will be
#     created. If a mask is input and active then the layer will be created
#     for the region specified by the mask and will not be created outside
#     the mask. For all other cases the layer will cover the entire region.
#   - the mask is that part of the domain specified by the region file.
#     The mask is automatically activated after input of the region file.
#LAYER REGION_MASK 'TEST.RGN'
#
#   - Any mask that has been input can be inactivated (i.e. turned off).
#     In the event that the is needed again then it can activated by
#     setting REGION_ACTIVE ON. NB. This command does not delete the mask
#     ... it only turns it off ... it can always be turned back on again
#     later (unless the mask is overwritten with another RGN file in
#     the meantime by a #LAYER REGION_FILE command).
#LAYER REGION_ACTIVE OFF
#
#   - If a mask has been input from a region file then this activates it
#      (i.e. turns it on). If no region file has been input then the
#      command is ignored.
#LAYER REGION_ACTIVE ON
```

4.9.5 Model File Input Options: Control Options

4.9.6 Model File Input Options: Landform Input Options

how the values of $\square_3$ and m_3 are to be modified. If it is ABSOLUTE then for every point on the grid covered by the region file then previous values of $\square_3$ and m_3 are replaced by the new values. If it is RELATIVE then for every point on the grid covered by the region file the previous values of $\square_3$ and m_3 are multiplied by the new values.

The `<region filename>` is the file name of the region file that has the region to be modified. The filename should not include spaces.

4.9.7 Model File Input Options: Landform Creation and Masking

The layer creation commands create a layer over (1) the entire domain or (2) part of the domain if a mask file has been input and is active.

First we will deal with the concept of the masking file. By default when a layer is created it is created over the whole domain for all those nodes where a layer is possible (clearly is not possible to add a layer if the elevation input for the layer is above the soil surface, but there are also other exceptions we will discuss later). It is possible to restrict the spatial extent of the layer. The masking file is a region file (i.e. created by EAMS-Viewer) that indicates the spatial extent for which you wish to create a layer. The process of creating this mask is as follows.

- The region is input with the command LAYER REGION_MASK. This inputs the default region over which ALL subsequent layers will be created (and they will not created outside this region). NB: By default this mask is active as soon as it is input (see next dot point)
- A simple rectangular masking region can be input with the command LAYER REGION_CLIP. This rectangular mask overwrites any region input with LAYER REGION_MASK. The mask is automatically activated.
- If you have input a mask but do not to want to use it you can deactivate by the command LAYER REGION_ACTIVE OFF. All subsequent layers will then created for the entire region. To turn the same mask back on again (i.e. activate it) for subsequent layer creation use the command LAYER REGION_ACTIVE ON. This activation and deactivation can be done as often as required. Note that activation/deactivation does not change the stored mask region, only whether that mask region is used in defining the spatial extent of any layer subsequently created.
- The mask that is activated/deactivated is the mask input by the last LAYER REGION_MASK command. Each time an LAYER REGION_MASK command is executed the previous mask is overwritten.

We now discuss the commands controlling the creation of layers. The general principle with layer creation is that the layer is created by specifying what the elevation is of the bottom of the layer and everything above that elevation to the soil surface is part of that layer. The characteristics of the material in that layer are either (1) the default values specified by the run parameters or (2) the values for these defaults modified by previous layer parameter commands (e.g. LAYER ERODIBILITY).

- The simplest layer command is LAYER Z. This command inputs a horizontal layer. The base of the layer is the elevation input. If at some point in the landscape the input elevation is above the surface (e.g. you are creating a geologic strata that is present on the tops of a hills but which is above the bottom of a valley) then a layer at that low point is not created. The following command creates a layer with base elevation 11.2m.

```
LAYER Z 11.2
```

- LAYER CAPPING puts a capping layer on the surface. The thickness input is the thickness of the capping layer. The elevation of the base of the layer is calculated as the elevation of the landscape surface minus the capping thickness. Thus if the capping is input as 2 then a layer 2m thick is placed across the surface (within the masking region if one has been activated) with the bottom elevation for the layer being 2m lower than the landscape surface. A multilayer cap can be input by first inputting the lowest layer, then the next layer up, working your way upwards to the surface. For instance to create a three layer cap (with top layer 1m thick, 2^{nd} layer 0.6m thick and bottom layer 0.2m thick) with the top layer having 0.1*<default erodibility>, 2^{nd} layer 0.5*<default erodibility> and the 3^{rd} layer being 10*<default erodibility> the following commands are input

```
LAYER ERODIBILITY DEFAULT
LAYER ERODIBILITY RELATIVE 10.0
LAYER CAPPING 1.8
LAYER ERODIBILITY DEFAULT
LAYER ERODIBILITY RELATIVE 0.5
LAYER CAPPING 1.6
LAYER ERODIBILITY DEFAULT
LAYER ERODIBILITY RELATIVE 0.1
LAYER CAPPING 1.0
```

- LAYER DEM allows the input of a spatially variable elevation provided by a DEM. The DEM is in a RST2 file (which must have the same grid dimensions as the domain). The layer can be offset vertically from this DEM. The sign convention of the offset is a positive offset has the layer base is above the DEM elevations, while negative is below. As for the other layer commands if the layer base is above the landscape surface at a point no layer is produced at that point. The follow command reads the DEM from file 'test.rst2' (note that on UNIX machines upper and lower case are important in the file name) and creates a layer 1.2 m higher than the elevations of the DEM.

```
LAYER DEM 1.2 'test.rst2'
```

- LAYER BILINEAR 1 20 1 30 10.0 20.0 22.0 35.0 allows the input of a smooth surface defined by a bilinear surface. The 1^{st} 4 parameters define the (east, north) coordinates of the bottom left and top right corner of a box while 2^{nd} 4 parameters define the elevations at the 4 corners (1^{st} is bottom left corner, 2^{nd} bottom right, 3^{rd} top left, and 4^{th} top right corner). Note that these 8 parameter ONLY define the elevations of the bilinear surface not the spatial extent of the layer (i.e. the layer will extend outside the bound box defined by the 1^{st} 4 parameters). To restrict the spatial extent of the layer you should input a masking region.

5 Monte-Carlo Modelling

Monte-carlo risk assessment modelling is controlled by the parameter `ModeMC`. For `ModeMC`=0 the monte-carlo capabilities are disabled.

5.1 ModeMC=0

This mode is for non Monte-carlo runs. This is the default mode for running SIBERIA.

5.2 ModeMC=1

This mode is the mode that has been historically used for Monte-Carlo modelling. Because of fundamental incompatibilities of the old Monte-Carlo Control file and the current one this mode has now been disabled. If this value is selected then the code issues a warning and sets `ModeMC`=0. All the capabilities of this mode and more are available in `ModeMC`=2 in a slightly different format.

5.3 ModeMC=2

Currently undocumented.

6 Calibration of SIBERIA

There are a number of ways of calibrating SIBERIA to data, most of which are complicated enough in detail to refer the interested user to the original papers. Calibration can be done in two ways (1) fiddling the parameters of the model until the model fits known landforms in some measurable (and hopefully defensible) way, (2) calibration of the processes within the model directly and then using those parameters to predict landform characteristics. We will only address the second of the methods here as the first methods are (at least in principle if not in practice) relatively obvious and straightforward. Moreover, here we will address only the calibration of the fluvial sediment transport parameters. At this time there are few reliable methods to calibrate the diffusive processes.

6.1 Calibration to erosion plot data

This section describes the calibration of the transport limited fluvial sediment transport model in SIBERIA to erosion plot data. The equations we need to calibrate are the runoff equation

$$Q = \beta_3 A^{m_3} \qquad (6.1.1)$$

and the fluvial sediment transport equation

$$Q_s = \begin{cases} \beta_1 Q^{m_1} S^{n_1} - Q_t & \beta_1 Q^{m_1} S^{n_1} - Q_t > Q_t \\ 0 & \beta_1 Q^{m_1} S^{n_1} - Q_t \le Q_t \end{cases} \qquad (6.1.2)$$

where it should be noted that these are the mean annual equations. That is the sediment transport that simulated is that occurring on average every year not instantaneously at given time.

An important observation for the discussion below is that Equation 6.1.2 can be formulated as

$$Q_s = \begin{cases} \left(\beta_1 \beta_3^{m_1}\right) A^{m_1 m_3} S^{n_1} - A_t & \left(\beta_1 \beta_3^{m_1}\right) A^{m_1 m_3} S^{n_1} > A_t \\ 0 & \left(\beta_1 \beta_3^{m_1}\right) A^{m_1 m_3} S^{n_1} \le A_t \end{cases} \qquad (6.1.3)$$

or

$$Q_s = \begin{cases} \beta A^{m} S^{n_1} - A_t & \beta A^{m} S^{n_1} > A_t \\ 0 & \beta A^{m} S^{n_1} \le A_t \end{cases} \qquad (6.1.4).$$

This form directly relates the long term sediment transport with the area and slope. In this form we do not need to evaluate each of the parameters $\beta_1,\ m_1,\ n_1,\ Q_t,\ \beta_3,\ m_3$ separately. In this form there are only 4 independent parameters to be estimated, $\beta,\ m,\ n_1,\ A_t$. If you are *not* interested in simulating the runoff characteristics of the DEM then you can simply simulate erosion only. In this way the model calibration task is simplified somewhat. In fact, internally within SIBERIA, the net result of the way the solution is calculated is effectively the same as Equation (6.1.4) for erosion.

6.1.1 Brute Force Calibration

The brute force calibration method involves the simulation of runoff and sediment transport record for DEM for an long period (at least 20 years, preferably more). This long term record is then analysed and the parameters in the long term erosion model evaluated by nonlinear regression in Equation (6.1.4). The most important aspect of this approach is that you must be able to simulate the discharge and sediment transport record for a range of different catchment areas within the DEM so as to get the scaling exponent on area. As suggested above you can set $\beta_3 = 1$ and $m_3 = 1$ if you are *not* specifically interested in simulating the hydrology, and then evaluate the remaining parameters in SIBERIA from the calibrated values of $\beta,\ m,\ n_1,\ A_t$.

To date this approach has not been widely used because of the computation demands of determining the scaling parameter on area *m*.

6.1.2 Scaling Approach

Willgoose et al. (1989) and Willgoose and Riley (1993, 1998a,b) showed the runoff and sediment that can be measured instantaneously (e.g. during a storm event) and can be related to the mean annual sediment transport in a conceptually simple fashion. A similar result has been recently proposed by Tucker and Bras (1999) where they related the statistical characteristics of rainfall and infiltration to derive at a model for mean annual sediment load in terms of rainfall and infiltration characteristics. Willgoose et al. (1989) related the equation for the sediment flux (note that his equation assumes there is no sediment transport threshold so that $A_t = 0$) at any instant in time, q_s,

$$q_s = \beta_2 q^{m_2} S^{n_2} \tag{6.1.5}$$

to the long term erosion rate, $\bar{q}_s$,

$$\bar{q}_s = \beta_1 q_p^{m_1} S^{n_1} \tag{6.1.6}$$

where

$$\beta_1 = \beta_2 \omega \left(\overline{T}_p \overline{q}_p \int \left[q'(t') \right]^{m_1} dt' \right) \left[1 + m_1 \left(m_1 - 1 \right) \frac{\sigma^2_{q_p}}{\overline{q}_p^2} + m_1 \frac{\sigma^2_{q_p T_p}}{\overline{q}_p \overline{T}_p} \right]$$

$$m_1 = m_2 \qquad (6.1.7)$$

$$n_1 = n_2$$

The term in the parentheses in the erodibility in Equation (6.1.7) is a function of (from left to right in the equation) (1) the intrinsic erodibility of the material β_2, (2) the rate of arrival of significant erosive events, ω, (3) the unit hydrograph of the catchment, reflected in the integral of the characteristic hydrograph of the catchment, $\int \left[q'(t') \right]^{m_1} dt'$, scaled by the mean duration and peak discharge of the hydrograph $\overline{T}_p \overline{q}_p$ and (4) the mean peak discharge and correlation with the duration of the event, related to the rainfall and runoff statistics of the catchment. Willgoose et al. (1989) and Willgoose at al. (1991a) provide a detailed development of this equation. The discharge in Equation (6.1.6) and (6.1.7) is the mean peak discharge/unit width, q_p. The mean peak discharge is approximately equal to the 1 in 2.33 year recurrence peak discharge event for that catchment. In the index method for flood prediction the mean peak discharge Q_p is related to a power of the area of a catchment

$$Q_p = \beta_3 A^{m_3} \qquad (6.1.8)$$

where the coefficient β_3 is the flood frequency index of the discharge and changes with the recurrence interval of the flood event. Typically it is found that the power on area A^{m_3} is independent of the recurrence interval of the flood discharge and is typically in the range 0.6-0.8. The dependence on the mean peak discharge arises because of the nonlinearity of the discharge exponent in Equation (6.1.5). That $m_1 >> 1$ means that larger runoff events have a disproportionately high input into the mean annual sediment flux and smaller flood events can be ignored with little error.

To convert from the mean peak discharge to mean peak discharge/ unit width an effective flow width is required. For hillslope elements that is likely to be come proportion of the element width (the wetted perimeter perhaps) while for a channel element width, which might be derived from regime equations.

Using this model, calibration can be simplified into a three stage process (Figure 6.1).

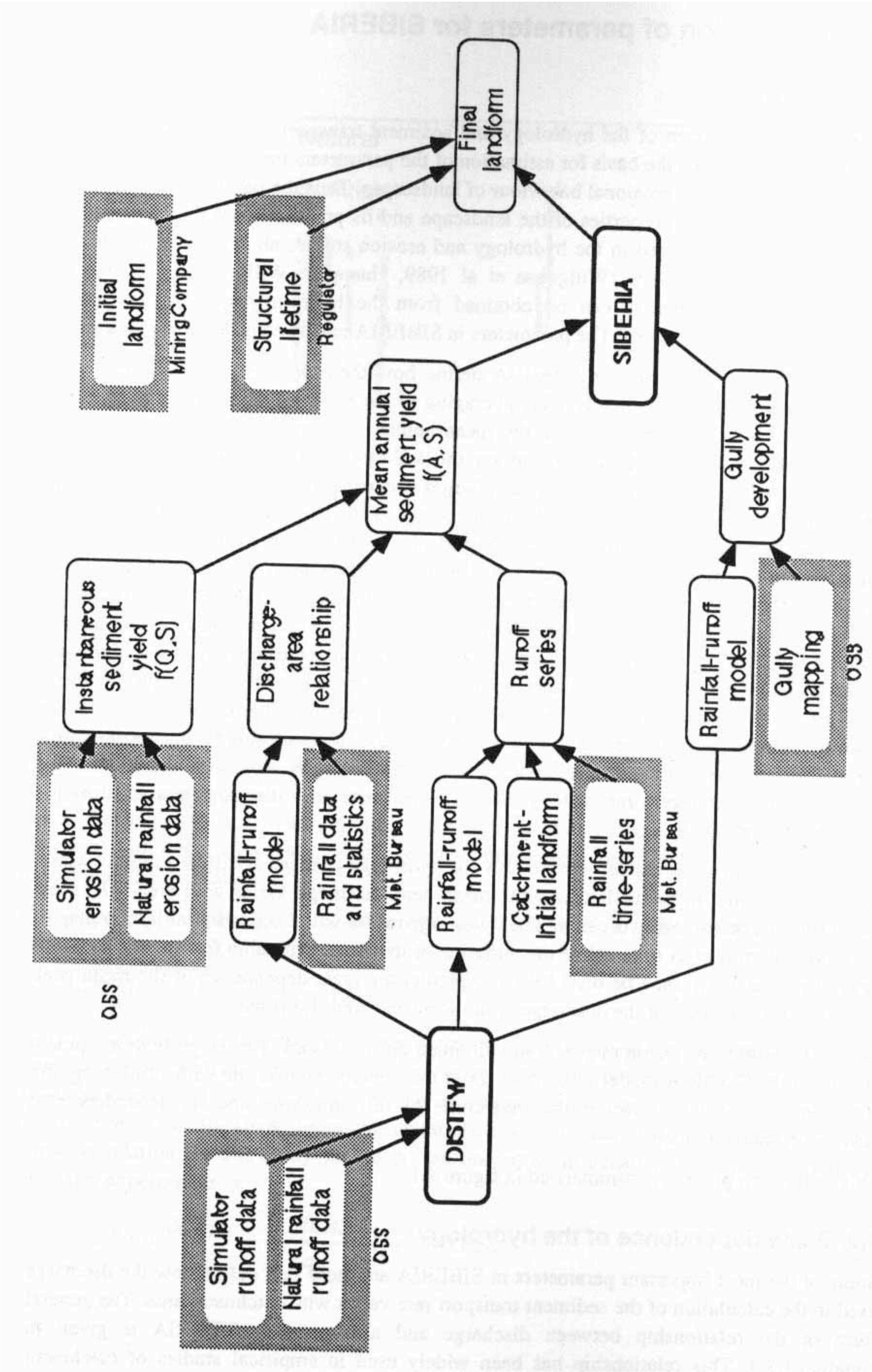


Figure 6.1: Schematic of the calibration procedure for SIBERIA when using field data (from Willgoose and Riley, 1998b)

- **Stage 1: Mean peak discharge-area relationship**: The mean peak discharge-area relationship (equation 6.1.8) is determined by simulating 1 in two year storm peak discharge for a DEM of the site and doing a regression between the peak discharge and area. Note that the discharge used in this regression is the critical duration peak discharge so that the complete range of duration design rainfall events must be used and at each point in the DEM the peak of the discharges given by the different duration events is selected. The exponent, m_3, is determined by plotting the peak discharge against the catchment area for every point in the DEM on a log-log plot and the slope of the relationship determined. The coefficient, β_3, is determined from the intercept of this plot for $A = 1$. See Figure 6.2 for a typical result of this process.

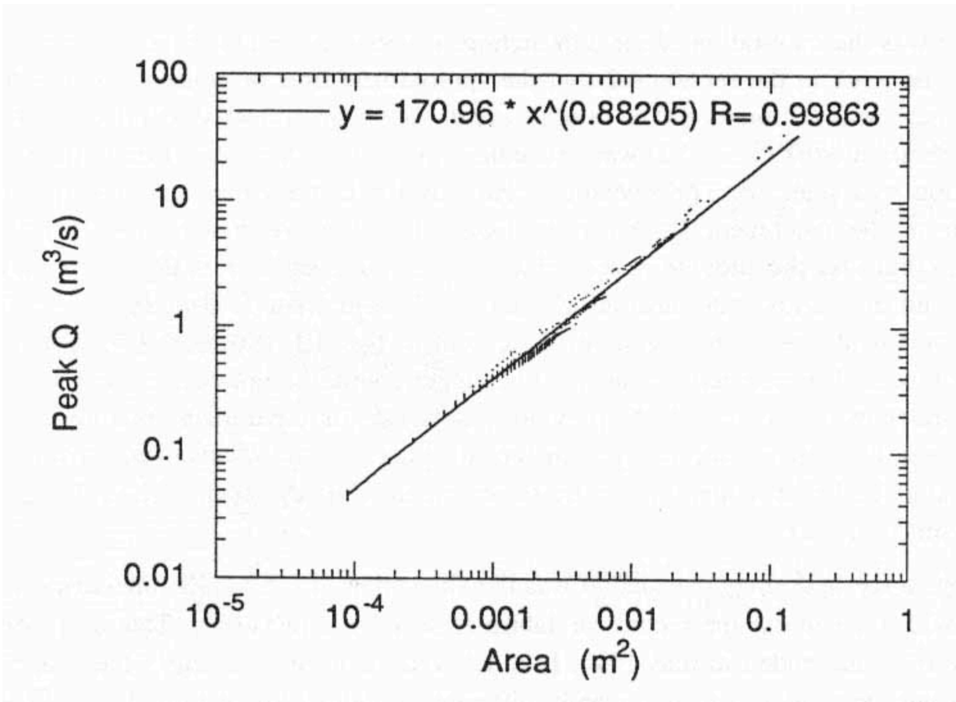
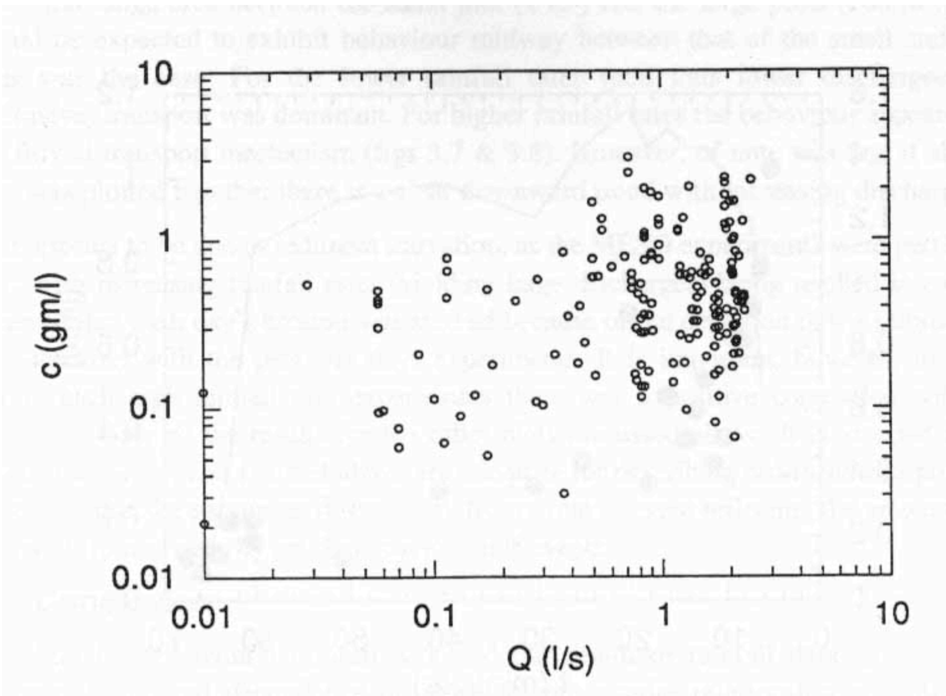
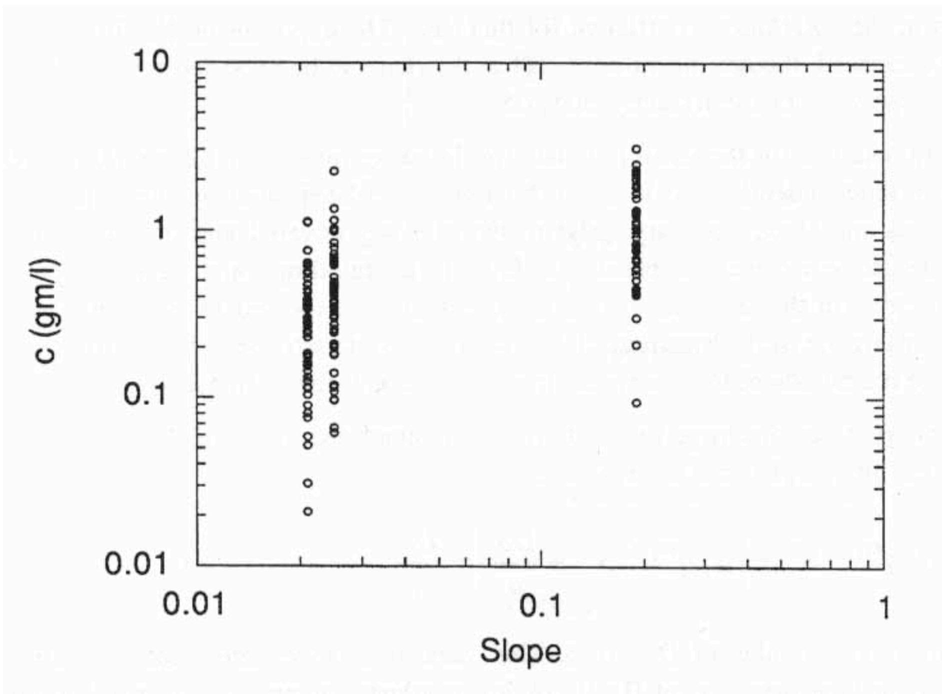


Figure 6.2: Example Mean Peak Discharge versus Area plot from a kinematic wave model (from Willgoose and Riley, 1998b).



(a)



(b)

Figure 6.3: Example Field data showing (a) Concentration versus discharge and (b) Concentration versus slope (from Willgoose and Riley, 1998b).

Stage 2: At a station instantaneous sediment concentration-discharge-slope relationship: This is the stage that fits the parameters β_2, m_2, and n_2 in Equation (6.1.5). This is simply achieved by multiple regression of sediment fluxes per unit width (or sediment concentration , which is just sediment flux divided by discharge) against slope and discharge (see Figure 6.3 for some typical field data). For this to work you must have plot or catchment runoff and erosion data for at least two different slopes (preferably more, though cost normally prohibits too many plots at different slopes) and different discharges (in rainfall simulator this is easy to achieve by applying different rainfall, or run-on, rates for the same plot). If the data has been collected for a series of natural rainfall events (as compared to simulators which normally apply a constant rainfall for 30 minutes or so) then the range of discharge within a rainfall event is normally sufficient to allow estimation of m_2. The value of m_2 should be accurate provided that they are reasonably large events and thus indicative of the larger dominant erosion events. It is worth noting here that our experience is that two or three good events where the whole hydrograph (the full range of observed discharges for both the rising and falling limbs so you capture any hysteresis) is measured are much more valuable than a large number of poorly sampled events. Thus a short intensive field campaign is more effective than a longer term but less intensive monitoring arrangement.

- **Stage 3: At a station long-term sediment yield**: This is the empirical determination of the relationship between parameters β_2 and β_1 in Equation (6.1.7). In this process a pluviograph record is used with a calibrated rainfall-runoff model to simulate a long term runoff record at high temporal resolution for the site. The sediment transport equation from Stage 2 is then used to determine the sediment yield arising from this runoff record and the value of β_1 determined. In principle this is very similar to the process used in the USDA models CREAMS and WEPP to determine a long term mean annual erosion rate. There are a number of nontrivial intermediate steps in this process:
  - **Stage 3a: The pluviograph record**: The pluviograph record should be of high enough resolution that it captures all of the critical duration runoff and erosion events in the record. For instance, in a small catchment of a few hectares it would be unusual for a catchment to take more a few hours to respond (ala time of concentration) to a rainfall event. In this case the rainfall record should be of sufficient resolution that it captures all of the important rises and falls to be found in these critical storm hydrograph. This would suggest that the rainfall record to be used should be sub-hour in resolution for a small catchment. By way of comparison CREAMS generally simulates rainfall on a daily basis which will fail to capture the detail of the critical erosion events. The rationale for this is that historically the vast majority of available rainfall records are aggregated daily rainfall. WEPP uses either pluviograph or dissaggregated daily rainfall (e.g. see Bras and Rodriguez-Iturbe, 1985, for a basic discussion of rainfall dissaggregation methods) to simulate these critical duration events. It is possible for this temporal resolution error in the simulated erosion record to be as much as an order of magnitude in size.

- **Stage 3b: The calibrated hydrology model**: As for the rainfall record the rainfall-runoff model must be able to simulate runoff at the resolution of the critical duration events for long term records. This model can be calibrated to observed rainfall–runoff data for an existing catchment, as per standard hydrologic practice. A more difficult situation arises in the mine rehabilitation area where the model is be used for simulation of erosion on sites that have not yet even been built so that appropriate hydrology data does not exist for this calibration. In this case it may be possible to use rainfall simulator data to calibrate the model and to scale this model up the catchment scale (Willgoose and Riley, 1998a,b). While infiltration properties are easily calibrated by this method, calibration of runoff-routing parameters (e.g. kinematic wave parameters) is particularly error-prone (Willgoose and Kuczera, 1995). Evidence to date suggests that the effects of errors in routing parameters are most apparent in the scaling equation (Stage 1) and that the effects of differences in the surface condition between the plot and full scale field catchment overwhelm scale effects in Stage 3. For instance, field scale catchments on mine sites are often quite rough, primarily to reduce runoff and encourage seed germination, while experimental plots are generally smooth to allow impermeable plot boundaries to be constructed, and the routing and infiltration properties of the two surface are dramatically different (e.g. Evans, 1998).

A simplified variant of this approach was used to develop the Queensland Coal Association Version 1.0 erosion database (`qca-v1.sdb`; Gyasi-Agyei and Willgoose, 1997). In this case erodibilities for a range of different mine wastes and different surface materials were determined from an extensive field campaign with a rainfall simulator over a three year period (Bell, at al. 1993). Because of a lack of data for different slopes (data was only available for plots with slope of 20% at the time) and different discharges (a single constant rainfall rate was applied for 30 minutes so that only a small variation in discharge, that resulting from a time varying infiltration rate, was observed) $m_1 = 1.5$ and $n_1 = 2$ were assumed, being about the values to be expected for a rilled surface (Willgoose at al., 1989). No erosion threshold was observed in the data so that $A_t = 0$ could be assumed and the calibration method above was used. The erodibility parameter β_2 was then simply estimated by relating the sediment removed from the plot during the 30 minute rainfall simulation experiment and the runoff/unit width measured. This equation was then used with local pluviograph records (and the measured infiltration characteristics of the plots) to generate a long runoff and erosion series from which β_1 was then estimated.

6.2 Calibration to other erosion models

This section discusses the calibration of SIBERIA's transport limited fluvial sediment transport model to other fluvial erosion models. The equations we need to calibrate are the runoff equation

$$Q = \beta_3 A^{m_3} \qquad (6.2.1)$$

and the fluvial sediment transport equation

$$Q_s = \begin{cases} \beta_1 Q^{m_1} S^{n_1} - Q_t & \beta_1 Q^{m_1} S^{n_1} > Q_t \\ 0 & \beta_1 Q^{m_1} S^{n_1} \leq Q_t \end{cases} \tag{6.2.2}$$

where it should be noted that these are the mean annual equations. Now as previously noted in the preceding section the parameters $\beta_3,\ m_3$ and $\beta_1,\ m_1,\ n_1,\ Q_t$ need to be evaluated. This can be done by generating a synthetic set of erosion data using another sediment transport model (e.g. CREAMS, WEPP) for a range of different slopes, catchment areas and rainfall rates. Since there are 6 parameters then at least 6 (and preferably more to check the adequacy of the representation used by SIBERIA) different sets of data need to be evaluated though if the model that is being used to generate the data outputs both runoff and erosion data then only 4 independent sets of data are necessary. Though again more would be desirable. Of most importance is that there be different slopes (to determine n_1), different catchment areas (to determine m_3) and different runoff rates (to determine m_1).

Now an important observation to make is that the values of the parameters $\beta_3,\ m_3$ and $\beta_1,\ m_1,\ n_1,\ Q_t$ may not be the obvious ones discussed in the literature. For instance, in Willgoose et al. (1989) it was noted that river engineering suggests that $m_1,\ n_1$ should be about 1.8 and 2.1 respectively for the Einstein Brown sediment transport equation for a wide channel (i.e. the wetted perimeter of the flow does not change with flow depth). This ignores channel geometry and Willgoose et al. noted (as have other authors, for example, Moore and Burch, 1986) that the value for m_1 should be (and is also observed to be) about 1.3-1.5 depending on rill geometry in overland and channel geometry in channel flow. More recently, Willgoose and Riley (1993, 1998a,b) observed for an armoured surface on a mine site that a fitted value for n_1 of about 0.7 was best. Evans (1998) showed that this result was consistent with measurements of the armours on the surfaces where the steeper surfaces had a coarser armour than the flatter surfaces. Thus the erosion rate on the steeper surfaces did not increase as much as would be expected from blind application of the river equations (which assume that the grading of the transported material is independent of slope). Since the ratio of $\frac{m_1 - 1}{n_1}$ determines the characteristic shape of the landform over the long term such considerations are important in the calibration. The parameters calibrated for SIBERIA will not reflect these characteristics if these issues are not addressed either by the structure of the model or by the way the model is used for calibration. Thus the calibrated parameters will only be as good as the model used for generating the data and the intelligence with which that model is subsequently used. At this stage it seems reasonable to suggest that for most natural surfaces m_1 will be in the range 1.3-1.5, a general recommendation for the value of n_1 is not possible, though data, backed up physical reasoning would suggest that a value less than 2 and perhaps as low 0.5-1 might be appropriate. It should be noted, however, that there is a consensus that the concavity of most fluvial erosion dominated surfaces (parameterised in SIBERIA by $\frac{m_1 m_3 - 1}{n_1}$) is in the range 0.4-0.6. If m_3 is about 0.8 and m_1 is about 1.5 then this suggests a value of n_1 in the range 0.5-0.8 is appropriate, which is consistent with the findings of Willgoose and Riley (1993, 1998a,b) and Evans (1998). However, until further studies are published on the interaction between slope and armouring this conclusion should be treated with some caution.

One simple way we commonly use to calibrate the model is to generate a series of linear hillslopes 1 node wide and X (some number) nodes long, simulating a slope X m long and 1 m wide. Simulations are then done on this hillslope with the model simulating the erosion data and using a DEM for this same hillslope the parameters of SIBERIA can be determined. The parameters so derived are then per unit width.

One particularly simple method is possible if only the erosion rate (and not the runoff) is of interest. In the output to SIBERIA is a diagnostic called TOTAL MASS. This is simply all of the elevations of the DEM node points added up and allows you to track the average elevation of the catchment over time (just the ‘total mass’ divided by the ‘total area’). If the erosion model predicts that the catchment average loss is 0.5mm/year (typically these models will give tonnes/Ha which can be converted to mm loss by using the bulk density of the surface soil) then for a unit slope (i.e. S=1) hillslope and assuming β_3, m_3 to be both unity (remember we are only interested in the erosion rate, not the discharges) β_1 can be adjusted up and down until SIBERIA yields 0.5mm of erosion over the same catchment over one timestep (one timestep being equivalent to one year in this case). The discharge on the hillslope is then doubled and tripled to determine the parameters m_1, Q_t and the slope doubled and tripled to obtain the parameter n_1. A few extra slopes, discharges and areas are typically generated to check that the calibration over the range of values to be simulated is satisfactory.

7 REFERENCES

Bell L C, Loch R J, Haneman D, and Willgoose G R. 1993. "A post-mining landform research program for open-cut mines". *Australian Minerals Industry Council Environmental Workshop*. Melbourne, Australian Institute of Mining and Metallurgy.

Bras R L and Rodriguez-Iturbe I. 1985. *Random Functions and Hydrology*, Addison-Wesley, New York.

Davis W M. 1954. *Geographical Essays*, Dover, New York.

Dunne T. 1989. "Hydrology, mechanics and geomorphic implications of erosion by subsurface flow", *Groundwater Geomorphology*, Higgins, (Ed) Geological Society of America,

Evans K G. 1998. *Runoff and erosion characteristics of a post-mining rehabilitated landform at Ranger Uranium Mine, Northern Territory, Australia and the implications for topographic evolution*. Ph.D. thesis, The University of Newcastle.

Gilbert G. 1909. "The convexity of hillslopes", *Journal of Geology*, 17, pp. 344-350.

Gyasi-Agyei Y, and Willgoose G R. 1997. *Calibration of SIBERIA parameters for 15 mine sites using runoff and erosion data obtained from the QDPI rainfall simulator: Derivation of the Version 1 Database*, Research Report 146.05.1997. Department of Civil, Surveying and Environmental Engineering, The University of Newcastle, Callaghan, NSW.

Hancock G R. 1997. *Experimental testing of the SIBERIA landscape evolution model*. Ph.D. thesis, The University of Newcastle.

Horton R E. 1945. "Erosional development of steams and their drainage basins; hydrophysical approach to quantitative morphology", *Bulletin of the Geological Society of America*, 56, pp. 275-370.

Huggett R J. 1988. "Dissipative systems: Implications for geomorphology", *Earth Surface Processes and Landforms*, 13, pp. 45 - 49.

Kirkby M J. 1971. "Hillslope process-response models based on the continuity equation", in *Slopes: form and process*, Institute of British Geographers Special Publications, 3, London, Institute of British Geographers, pp. 15 - 30.

Kirkby M J. 1987. "Modelling some influences of soil erosion, landslides and valley gradient on drainage density and hollow development", Ahnert F, *Geomorphological Models: Theoretical and Empirical Aspects,* Aachen, Germany, 1986.

Knisel W. 1980. *CREAMS: A field-scale model for chemicals, runoff, and erosion from Agricultural Management Systems*, Conservation Research Report No. 26, US Department of Agriculture, Washington, USA

Loewenherz-Lawrence D S. Hydrodynamic description for advective sediment transport processes and rill initiation. *Water Resources Research*. 1994; 30(11), 3203-3212.

Meinhardt H A. 1982. *Models of biological pattern formation*, Academic Press, Berlin.

Moore I D, and Burch G J. 1986. Sediment transport capacity of sheet and rill flow : Application of unit stream power theory. *Water Resources Research*, 22(8), pp 1350-1360.

Shreve R L. 1966. “Statistical law of stream numbers”, *Journal of Geology*, 74, pp. 17-37.

Smith T R and Bretherton F P. 1972. “Stability and the conservation of mass in drainage basin evolution”, *Water Resources Research*, 8(6), pp. 1506 - 1529.

Strahler A N. 1964. “Quantitative geomorphology of drainage basins and channel networks”, *Handbook of applied hydrology*, Chow V T, (Ed.) New York, McGraw Hill, pp. 4-39 - 4-76.

Tarboton D G. 1997. “A new method for the determination of flow directions and upslope areas in grid digital elevation models”. *Water Resources Research*, 33(2), pp 309-19.

Tucker G and Bras R L. 1999.

Valdes J B, Fiallo Y and Rodriguez-Iturbe I. 1979. “A rainfall-runoff analysis of the geomorphologic IUH”, *Water Resources Research*, 15(6), pp. 1421-1434.

Willgoose G R. 1993. “Hydrology and erosion”, *Proceedings of the Symposium on the Management and Rehabilitation of Waste Rock Dumps,* Darwin, 7-8 October, 1993.

Willgoose G R. 1994a. “A physical explanation for an observed area-slope-elevation relationship for declining catchments”, *Water Resources Research*, 30(2), pp. 151-159.

Willgoose G R. 1994b. “A statistic for testing the elevation characteristics of landscape simulation models”, *Journal of Geophysical Research*, 99(B7), pp. 13987-13996.

Willgoose G R, and Kuczera G A. 1995. “Estimation of sub-grid scale kinematic wave parameters for hillslopes”. *Hydrological Processes*, 9(3/4), 469-82.

Willgoose G R and Riley S. J. 1993. “The assessment of the long-term erosional stability of engineered structures of a proposed mine rehabilitation”, Chowdhury R N and Sivakumar M (Ed), *Environmental Management: Geowater and Engineering Aspects,* Wollongong University, 8-11 February, 1993.

Willgoose G R and Riley S J. 1994. “Long-term erosional stability of mine spoils”, *Australian mining looks north - The challenges and choices,* Australian Institute of Mining and Metallurgy National Conference, Darwin, 5-9 August, 1994.

Willgoose G R, and Riley S J. 1998a. An assessment of the long-term erosional stability of a proposed mine rehabilitation. *Earth Surface Processes and Landforms*, 23, 237-59.

Willgoose G R, and Riley S J. 1998b. *Application of a catchment evolution model to the prediction of long term erosion on the spoil heap at Ranger Uranium Mines: Initial analysis*. 107p . Supervising Scientist Report 132. Canberra, Australian Government Publishing Service.

Willgoose G R, Bras R L and Rodriguez-Iturbe I. 1989. *A physically based channel network and catchment evolution model*, TR 322, Ralph M. Parsons Laboratory, Dept. of Civil Engineering, MIT, Boston, MA.

Willgoose G R, Bras R L and Rodriguez-Iturbe I. 1991a. “A physically based coupled network growth and hillslope evolution model: 1 Theory”, *Water Resources Research*, 27(7), pp. 1671-1684.

Willgoose G R, Bras R L and Rodriguez-Iturbe I. 1991b. “A physically based coupled network growth and hillslope evolution model: 2 Applications”, *Water Resources Research*, 27(7), pp. 1685-1696.

Willgoose G R, Bras R L and Rodriguez-Iturbe I. 1991c. “A physical explanation of an observed link area-slope relationship”, *Water Resources Research*, 27(7), pp. 1697-1702.

Willgoose G R, Bras R L and Rodriguez-Iturbe I. 1991d. “Results from a new model of river basin evolution”, *Earth Surface Processes and Landforms*, 16, pp. 237-254.

Willgoose G R, Bras R L and Rodriguez-Iturbe I. 1994. “Hydrogeomorphology modelling with a physically based river basin evolution model”, *Process Models and Theoretical Geomorphology*, Kirkby M J (Ed), Chichester, Wiley, pp. 271-294.

APPENDICES

Appendix A: Revision History

A full revision history with details of the modifications at each update is provided in the file `siberia-revision-history.txt` provided in the EAMS install.

Version 6 (pre 1992)

The first version of SIBERIA. V6.28 was mostly used for the results of the Ph.D. results of Willgoose et al. (1989).

Later version of V6 (notably V6.34) were extended under an Australian Water Research Advisory Council (AWRAC) research grant to implement the code for mine site applications. These improvements included

- Rewrite of the numerical solver to improve the range of parameters for which it would work.
- Change the underlying basis of the solver so that it was flux based to allow the easy implementation of irregular catchment boundary geometries.
- The first version of the code to be used for the long term erosion assessment of a mine site (Willgoose and Riley, 1993, 1998a,b)

Version 7 (1992-1997)

Version 7 was the first version extensively used for mine site rehabilitation applications. While being loosely based on the V6 code it was extensively rewritten to

- Further improve the speed, accuracy and generality of the numerical solver.
- First version of adaptive time stepping algorithm
- Key physical processes in the model were modularised (the models in Section 4) to allow the code to be easily extended for new applications.
- While on sabbatical at University of Lancaster in 1995 I coded a version (V7.05) for use by Parallel Virtual Machine software (PVM) to allow us to easily using Monte Carlo risk assessment techniques to look at stochastic aspects of the landform evolution resulting from spatial variability of runoff and erosion properties, waste rock settlement, poorly specified initial conditions and chaotic aspects of the landform evolution processes. This version has not been kept up to date with the non-PVM version and is effectively frozen at V7.05.
- Various extensions to the code for Hancock (1997) to allow comparison of SIBERIA with experimental data from a small scale landform erosion simulator.
- Implementation of channels with real dimensions. Slopes in the channels can now be calculated from the elevation of the bottom of the channel rather than the overall elevations

of the simulated DEM (i.e. hillslope elevations). This resulted in there being an extra state (7 columns now) in the RST2 files, the depth of the channel.

- V7.05 was the first version of SIBERIA to be properly documented.

Version 8 (1998-date)

The major improvement in V8 is the implementation of a soil model. This work is ongoing and the soils model is still in the process of testing and validation. The major implication of this is the addition of an extra state (8 columns now) in the RST2 files, the depth of the soil.

V8 is also the first version to be made widely available to other researchers.

Independently of the soils model a number of other improvements have been made.

- V8.05: Simple armouring model implemented. Erosion rate decreases with increased erosion depth.
- V8.08: Rate parameters for runoff and erosion can now be defined independently of the grid resolution. Input parameters are defined as being per unit width and the code internally uses the grid resolution information to convert to the nondimensional grid used by the model.
- V8.09: Improved dynamic timestepping and comprehensive testing of its accuracy. A major improvement over the previous technique.
- V8.10: First version to be advertised as available on the web for researchers. Last version of SIBERIA in F77.
- V8.11: Incorporation of Tarboton's D∞ drainage direction algorithm. First version of SIBERIA in F90.
- V8.25: First release version of full dynamic memory. This allows better use of memory particularly for calculation of large domains (greater than a million nodes). It also provides the required memory management foundations for the layering model.
- V8.27: Pre-release version of detachment limitation for Los Alamos National Labs.
- V8.28-.29: Layering model engine implemented.
- V8.30: First release version of layering model.

Appendix B: Code to read in restart files.

The sample code that follows is the code segment of SIBERIA that reads in the restart files. This code is available in module INOUT. It is somewhat more general than is technically required. In particular, it reads in files from version 6 of SIBERIA which has a different header than V7. In addition, it has subroutines that adjust the interpretation of parameters depending on the version of the code that created the file. Previous versions of SIBERIA interpreted some parameters slightly differently from that described in this manual.

```
c
c======================================================================
c  Reads in the Restart file
c======================================================================
c
      SUBROUTINE readin(slope,vz,y,z,area,grid,filenm,init,ixxx,iyyy,
     &    direct,FilenameUser,MaxUser)
      IMPLICIT NONE
c
c  slope = slopes for DTM
c  vz    = random field for DTM
c  y     = channelisation of the DTM
c  z     = elevations of the DTM
c  area  = area draining through the DTM
c  direct= flow directions for the DTM
c  grid  = storage size of the arrays above (x,y the same)
c  filenm= name of restart file TO be READ
c  init  = no of fixed elevation points
c  ixxx,iyyy = x,y coordinates of the init fixed points
c
      INTEGER MaxNoInts,MaxNoReals
      PARAMETER (MaxNoInts=20,MaxNoReals=50)
C
      INTEGER grid,area(grid,grid),ixxx(*),iyyy(*)
     &    ,direct(grid,grid),MaxUser
      REAL*8 slope(grid,grid),vz(grid,grid),y(grid,grid),z(grid,grid)
      CHARACTER*(*) filenm
      CHARACTER*80 FilenameUser(MaxUser)
c
      INTEGER intvar
      REAL*8 realvar
      COMMON / D / Intvar(MaxNoInts),realvar(MaxNoReals)
c
      INTEGER unitno,RstMode,NoPar,init
      INTEGER noreals,noints,lgthname,lgthstr
      REAL*8 iversion
      CHARACTER*30 Line
c
      noreals=MaxNoReals
      noints=MaxNoInts
      unitno=10
      lgthname=lgthstr(filenm,80)
c
c  RST1 files no longer supported in V7 of SIBERIA
c
      IF (filenm(lgthname-4:lgthname).eq.'.rst2'.or.
     &      filenm(lgthname-4:lgthname).eq.'.RST2') THEN
        rstmode=2
      ELSE
        IF (filenm(lgthname-4:lgthname).eq.'.rst3'.or.
     &        filenm(lgthname-4:lgthname).eq.'.RST3') THEN
          rstmode=3
        ELSE
          WRITE(*,*) 'FATAL ERROR ',filenm(1:40)
     &               , ' is not a supported restart file'
          STOP
        END IF
      END IF
      IF (rstmode.le.2) THEN
c========================================================
```

```
c  Reading the ASCII version of the restart file
c=========================================================
        OPEN (unit=unitno,file=filenm,status='old',err=9999)
c
c   In later versions (>=V6.34) of the code the first line idenbtifies
c   the version of thecode that created the restart file. Early versions
c   DO not have this header line and start straight in with the parameters
c
        READ (unitno,6000) Line
 6000   FORMAT (a30)
        IF (line(2:7).eq.'BRANCH'.or.line(1:6).eq.'BRANCH'
     &     .OR.line(2:8).eq.'SIBERIA'.or.line(1:7).eq.'SIBERIA') THEN
          READ(line(9:30),6010) iversion
 6010     FORMAT(f21.0)
        ELSE
          CLOSE(unit=unitno,status='keep')
          OPEN (unit=unitno,file=filenm,status='old',err=9999)
C
C   The last version of the code not TO INCLUDE the header line is V6.33
C
          iversion=6.33
        END IF
        IF (iversion.lt.7.00) THEN
          CALL rst2v6(iversion,unitno,noints,noreals,nopar
     &         ,Intvar,realvar
     &         ,init,ixxx,iyyy,slope,vz,y,z,area,direct,grid)
        ELSE
          CALL rst2v7(iversion,unitno,noints,noreals,nopar
     &         ,Intvar,realvar
     &         ,init,ixxx,iyyy,slope,vz,y,z,area,direct,grid
     &         ,FilenameUser,MaxUser)
        END IF
        CLOSE(unit=unitno)
      ELSE
c=========================================================
c   READ the binary version of the file
c=========================================================
        OPEN (unit=unitno,file=filenm,status='old',err=9999,
     &         form='unformatted')
        READ (unitno) nopar
        IF (Line(1:4).eq.'BRAN'.or.Line(1:4).eq.'SIBE') THEN
          READ(unitno) Line (5:20)
          READ(Line(7:20),6010) iversion
          READ(unitno) NoPar
        ELSE
          iVersion=6.33
        END IF
        IF (iversion.lt.7.00) THEN
          CALL rst3v6(iversion,unitno,noints,noreals,nopar
     &         ,Intvar,realvar
     &         ,init,ixxx,iyyy,slope,vz,y,z,area,direct,grid)
        ELSE
          CALL rst3v7(iversion,unitno,noints,noreals,nopar
     &         ,Intvar,realvar
```

```
     &        ,init,ixxx,iyyy,slope,vz,y,z,area,direct,grid
     &        ,FilenameUser,MaxUser)
        END IF
```

```
        CLOSE(unit=unitno)
      END IF
      WRITE (*,6020) '  -- SIBERIA InputFile Version ',iversion
 6020 FORMAT(a31,f10.2)
c
c  DO any change in interpretation in parameters from old versions
c  of SIBERIA and BRANCH.
c
      CALL updatev6(iversion,IntVar,Realvar,FilenameUser)
      CALL updatev7(iversion,IntVar,Realvar)
      RETURN
 9999 WRITE(*,*)' -- Unable TO OPEN the RESTART input file'
      STOP
      END
C
c  =====================================
c  Modify any change in PARAMETER interpretation
c  from old versions of SIBERIA (V6)
c  =====================================
c
      SUBROUTINE updatev6(version,IntVar,Realvar,FilenameUser)
      IMPLICIT NONE
c
      INTEGER Intvar(*)
      REAL*8 realvar(*),version
      CHARACTER*80 FilenameUser(10)
c
      INTEGER i,k
c
c  The interpretation of PARAMETER REALVAR(24) (or B5) has changed
c  from version 6.33 TO correct an inconsistency (see SUBROUTINE CONST)
C
      IF (version.le.6.33) THEN
        RealVar(24)=RealVar(24)/(RealVar(18)**RealVar(27))
        RealVar(5)=0.0
      END IF
      IF (version.le.6.38) THEN
        RealVar(3)=0.0
        RealVar(4)=0.0
        RealVar(8)=0.0
        RealVar(10)=0.0
        RealVar(11)=0.0
        RealVar(12)=0.0
        RealVar(13)=0.0
        RealVar(14)=0.0
        Realvar(16)=1.0
        Realvar(17)=1.0
      END IF
      IF (version.le.6.41) THEN
        RealVar(5)=1000000
      END IF
```

```
c
c  YFIX is set TO channel for all V6 files. IMPLICIT in V6.
c
```

```
      Realvar(8)=1
c
c  user defined erosion module not available in v6
```

```
c
      do 1001 k=1,10
        DO 1000 i=1,80
          FilenameUser(k)(i:i)=' '
 1000   CONTINUE
 1001 continue
      RETURN
      END
C
c  =====================================
c  Modify any change in PARAMETER interpretation
c  from old versions of SIBERIA (V7)
c  =====================================
c
      SUBROUTINE updatev7(version,IntVar,Realvar)
C  This SUBROUTINE assumes that all modifications for V6 changes
c  have already been done. Only V7 modification are in here.
      IMPLICIT NONE
c
      INTEGER Intvar(*)
      REAL*8 realvar(*),version
C
c  fix gridsize, easting and northing TO default values
c    (DO this for all versions)
c
      IF (Realvar(34).eq.0.0) THEN
        realvar(34)=1
        realvar(35)=0
        realvar(36)=0
      END IF
c  Change in interpretation from soil porosity (never used) TO Bulk Density
      IF (version.le.7.00)realvar(22)=1
      RETURN
      END
C
c  =====================================
c  input of RST2 generated by SIBERIA V6
c  =====================================
c
      SUBROUTINE rst2v6(version,unitno,noints,noreals,nopar
     &    ,Intvar,realvar
     &    ,init,ixxx,iyyy,slope,vz,y,z,area,direct,grid)
      IMPLICIT NONE
```

```
C
      INTEGER grid,area(grid,grid),init,ixxx(*),iyyy(*)
     &    ,direct(grid,grid),Intvar(*),unitno,NoPar,noints,noreals
      REAL*8 slope(grid,grid),vz(grid,grid),y(grid,grid),z(grid,grid)
     &    ,realvar(*),version
c
      INTEGER i,j
C
        WRITE (*,*) ' -- Input V6 RST2 file'
        DO 1010 i=1,noints
          IntVar(i)=0
```

```
 1010   CONTINUE
        DO 1020 i=1,noreals
          RealVar(i)=0
```

```
 1020   CONTINUE
        noints=8
        noreals=33
        READ (unitno,*,err=9999,END=9998) nopar
        nopar=noreals+noints
        READ (unitno,*,err=9999,END=9998) (Intvar(i),i=1,noints)
        READ (unitno,*,err=9999,END=9998) (Realvar(i),i=1,noreals)
        READ (unitno,*,err=9999,END=9998) init
c
c  READ the rest of the restart file
c
        DO 1110 i=1,init
           READ(unitno,*,err=9999,END=9998) ixxx(I),iyyy(i)
 1110   CONTINUE
        DO 1120 j=1,Intvar(4)
          DO 1130 i=1,Intvar(3)
            READ(unitno,*,err=9999,END=9998)slope(i,j),vz(i,j),y(i,j)
     &                    ,z(i,j),area(i,j),direct(i,j)
 1130     CONTINUE
 1120   CONTINUE
      RETURN
 9999 WRITE (*,*) ' -- Error reading RST2 file'
      STOP
 9998 WRITE (*,*) ' -- Premature END TO RST2 file'
      STOP
      END
C
c  =====================================
c  input of RST2 generated by SIBERIA V7
c  =====================================
c
      SUBROUTINE rst2v7(version,unitno,noints,noreals,nopar
     &    ,Intvar,realvar
     &    ,init,ixxx,iyyy,slope,vz,y,z,area,direct,grid
     &    ,FilenameUser,MaxUser)
```

```
      IMPLICIT NONE
C
      INTEGER grid,area(grid,grid),init,ixxx(*),iyyy(*)
     &    ,direct(grid,grid),Intvar(*),unitno,NoPar,noints,noreals
     &    ,MaxUser
      REAL*8 slope(grid,grid),vz(grid,grid),y(grid,grid),z(grid,grid)
     &    ,realvar(*),version
      CHARACTER*80 FilenameUser(MaxUser)
c
      INTEGER i,j
C
        WRITE (*,*) ' -- Input V7 RST2 file'
        DO 1010 i=1,noints
          IntVar(i)=0
 1010   CONTINUE
        DO 1020 i=1,noreals
          RealVar(i)=0
 1020   CONTINUE
        noints=20
```

```
        noreals=50
c        READ (unitno,*,err=9999,END=9998) nopar
        nopar=noreals+noints
```

```
        READ (unitno,*,err=9999,END=9998) (Intvar(i),i=1,noints)
        READ (unitno,*,err=9999,END=9998) (Realvar(i),i=1,noreals)
        IF (version.ge.7.03) THEN
          DO 1021 i=1,MaxUser
            READ (unitno,6060,err=9999,END=9998) FilenameUser(i)(1:80)
 6060       FORMAT(a80)
 1021     CONTINUE
        ELSE
          READ (unitno,6060,err=9999,END=9998) FilenameUser(1)(1:80)
        END IF
        READ (unitno,*,err=9999,END=9998) init
c
c  READ the rest of the restart file
c
        DO 1110 i=1,init
           READ(unitno,*,err=9999,END=9998) ixxx(I),iyyy(i)
 1110   CONTINUE
        DO 1120 j=1,Intvar(4)
          DO 1130 i=1,Intvar(3)
            READ(unitno,*,err=9999,END=9998)slope(i,j),vz(i,j),y(i,j)
     &                     ,z(i,j),area(i,j),direct(i,j)
 1130     CONTINUE
 1120   CONTINUE
        CLOSE(unit=unitno)
      RETURN
 9999 WRITE (*,*) ' -- Error reading RST2 file'
      STOP
```

```
 9998 WRITE (*,*) ' -- Premature END TO RST2 file'
      STOP
      END
C
c  =====================================
c  input of RST3 generated by SIBERIA V6
c  =====================================
C
      SUBROUTINE rst3v6(version,unitno,noints,noreals,nopar
     &    ,Intvar,realvar
     &    ,init,ixxx,iyyy,slope,vz,y,z,area,direct,grid)
      IMPLICIT NONE
C
      INTEGER grid,area(grid,grid),ixxx(*),iyyy(*),intvar(*)
     &    ,direct(grid,grid),noreals,noints,unitno,nopar,init
      REAL*8 version,slope(grid,grid),vz(grid,grid),y(grid,grid)
     &    ,z(grid,grid)
     &    ,realvar(*)
c
      INTEGER i,j
      INTEGER*2 IParameter(8),i1,i2
c
      NoInts=8
      noreals=33
      READ (unitno,err=9999,END=9998) (IParameter(i),i=1,6)
      READ (unitno,err=9999,END=9998) (IParameter(i),i=7,8)
      DO 1150 i=1,NoInts
```

```
        IntVar(i)=IParameter(i)
 1150 CONTINUE
      READ (unitno,err=9999,END=9998) (Realvar(i),i=1,5)
```

```
      READ (unitno,err=9999,END=9998) (Realvar(i),i=6,10)
      READ (unitno,err=9999,END=9998) (Realvar(i),i=11,15)
      READ (unitno,err=9999,END=9998) (Realvar(i),i=16,20)
      READ (unitno,err=9999,END=9998) (Realvar(i),i=21,25)
      READ (unitno,err=9999,END=9998) (Realvar(i),i=26,30)
      READ (unitno,err=9999,END=9998) (Realvar(i),i=31,33)
      READ (unitno,err=9999,END=9998) init
      DO 1101 i=1,init
        READ(unitno,err=9999,END=9998) ixxx(I),iyyy(i)
 1101 CONTINUE
      DO 1120 j=1,Intvar(4)
        DO 1130 i=1,Intvar(3)
          READ(unitno,err=9999,END=9998)slope(i,j),vz(i,j),y(i,j)
     &          ,z(i,j),i1,i2
          area(i,j)=i1
          direct(i,j)=i2
 1130   CONTINUE
 1120 CONTINUE
      RETURN
```

```
 9999 WRITE (*,*) ' -- Error reading RST3 file'
      STOP
 9998 WRITE (*,*) ' -- Premature END TO RST3 file'
      STOP
      END
C
c  =====================================
c  input of RST3 generated by SIBERIA V7
c  =====================================
C
      SUBROUTINE rst3v7(version,unitno,noints,noreals,nopar
     &    ,Intvar,realvar
     &    ,init,ixxx,iyyy,slope,vz,y,z,area,direct,grid
     &    ,FilenameUser,MaxUser)
      IMPLICIT NONE
C
      INTEGER grid,area(grid,grid),ixxx(*),iyyy(*),intvar(*)
     &    ,direct(grid,grid),noreals,noints,unitno,nopar,init,MaxUser
      REAL*8 version,slope(grid,grid),vz(grid,grid),y(grid,grid)
     &    ,z(grid,grid)
     &    ,realvar(*)
      CHARACTER*80 FilenameUser(MaxUser)
c
      INTEGER i,j
      INTEGER*2 IParameter(20),i1,i2
c
      NoInts=20
      noreals=50
      READ (unitno,err=9999,END=9998) (IParameter(i),i=1,6)
      READ (unitno,err=9999,END=9998) (IParameter(i),i=7,12)
      READ (unitno,err=9999,END=9998) (IParameter(i),i=13,18)
      READ (unitno,err=9999,END=9998) (IParameter(i),i=19,20)
      DO 1150 i=1,NoInts
        IntVar(i)=IParameter(i)
 1150 CONTINUE
```

```
      READ (unitno,err=9999,END=9998) (Realvar(i),i=1,5)
      READ (unitno,err=9999,END=9998) (Realvar(i),i=6,10)
      READ (unitno,err=9999,END=9998) (Realvar(i),i=11,15)
```

```
      READ (unitno,err=9999,END=9998) (Realvar(i),i=16,20)
      READ (unitno,err=9999,END=9998) (Realvar(i),i=21,25)
      READ (unitno,err=9999,END=9998) (Realvar(i),i=26,30)
      READ (unitno,err=9999,END=9998) (Realvar(i),i=31,35)
      READ (unitno,err=9999,END=9998) (Realvar(i),i=36,40)
      READ (unitno,err=9999,END=9998) (Realvar(i),i=41,45)
      READ (unitno,err=9999,END=9998) (Realvar(i),i=46,50)
      IF (version.ge.7.03) THEN
        DO 1021 i=1,MaxUser
          READ (unitno,err=9999,END=9998) FilenameUser(i)(1:80)
 1021   CONTINUE
```

```
      ELSE
        DO 1200 i=1,80
          FilenameUser(1)(i:i)=' '
 1200   CONTINUE
      END IF
      READ (unitno,err=9999,END=9998) init
      DO 1101 i=1,init
        READ(unitno,err=9999,END=9998) ixxx(I),iyyy(i)
 1101 CONTINUE
      DO 1120 j=1,Intvar(4)
        DO 1130 i=1,Intvar(3)
          READ(unitno,err=9999,END=9998)slope(i,j),vz(i,j),y(i,j)
     &           ,z(i,j),i1,i2
          area(i,j)=i1
          direct(i,j)=i2
 1130   CONTINUE
 1120 CONTINUE
      RETURN
 9999 WRITE (*,*) ' -- Error reading RST3 file'
      STOP
 9998 WRITE (*,*) ' -- Premature END TO RST3 file'
      STOP
      END
c-----------------------------------------------------------------
c Routine TO left-justify a string from one CHARACTER variable
c TO another.
c-----------------------------------------------------------------
      SUBROUTINE moveb(strng1, strng2, i1, i2)
c
c Parameters:
c
c Input:
c           strng1    input string
c        i1        start CHARACTER position in strng2
c Output:
c        strng2    destination CHARACTER variable
c        i2        last CHARACTER position used in strng2
c
      CHARACTER strng1*(*), strng2*(*)
      INTEGER   i1, i2
      INTEGER   k, l
c
      l = len(strng1)
```

```
      k = 1
      DO while(strng1(k:k) .eq. ' ' .and.  k .lt. l)
        k = k + 1
```

```
      ENDDO
c
      i2 = i1 + l - k
```

```
IF(i2 .gt. len(strng2)) THEN
   i2 = len(strng2)
   l  = i2 - i1 + k
ENDIF

strng2(i1:i2) = strng1(k:l)
RETURN
END
```

Appendix C: Code to read in boundary files

The code below is the input routine for boundary files from SIBERIA v7. This code is available in module INOUT.

```
c
c======================================================================
c  Input the boundaries files for irregular grid
c======================================================================
c
      SUBROUTINE InputBoundaries(BoundFile,Domain,Grid,kx,ky
     &                       ,init,ixxx,iyyy,Regions,NoRegions
     &                       ,MaxRegions,Region)
c
c  BoundFile = name of file TO be READ
c  domain    = LOGICAL array with the DATA of what nodes are in the domain
c  Grid  = the declared size of domain (same in x and y directions)
c  kx,ky     = x and y sizes of the domain
c  init      = no of fixed elevation points
c  ixxx,iyyy = the x and y coordinates of the fixed elevation points
c
      INTEGER NoOutlets
      PARAMETER (NoOutlets = 1500)
c
      INTEGER Grid, kx, ky, init,ixxx(*),iyyy(*),Regions(*)
     &    ,NoRegions,MaxRegions,Region(grid,grid)
      LOGICAL Domain(Grid,Grid)
      CHARACTER*(80) BoundFile
c
      INCLUDE 'gridsize1.inc'
c
      INTEGER FIOUnit, Noout,Area,ix,iy,i,j,kxm,kym,num,ord
      CHARACTER*1 stuff(GridSize1,GridSize1)
      DATA FIOUnit / 10 /
c
      OPEN(unit=FIOUnit, file=BoundFile, status='old')
      READ (FIOUnit,1000)
 1000 FORMAT(a1)
      READ (FIOUnit,*) kxm, kym
      IF (kx.lt.kxm.or.ky.lt.kym) THEN
        WRITE (*,*) ' *** The grid is too small for input boundaries'
        WRITE (*,*) '     KX, KY increased'
      END IF
      kx=kxm
      ky=kym
      ix=kx
      iy=ky
      NoOut=0
      Area=0
      DO 2000 i=1,iy
        READ(10,1010) (stuff(j,i),j=1,ix)
 1010   FORMAT(201a1)
        DO 2010 j=1,ix
          Region(j,i)=-1
          IF (stuff(j,i)(1:1).ne.'.') THEN
            domain(j,i)=.true.
            area=area+1
c
```

```
c  Fixed elevation outlet BC points.
c
            IF (stuff(j,i)(1:1).eq.'^') THEN
              NoOut=NoOut+1
              IF (NoOut.gt.NoOutlets) THEN
                WRITE (*,*) ' No of Outlet nodes in',
     &                       ' boundary file too large.'
                WRITE (*,*) '   Max Allowed = ',NoOutlets
                STOP
              END IF
              ixxx(NoOut)=j
              iyyy(NoOut)=i
            END IF
c
c  Fixed elevation regions BC regions (eg. reservoirs)
c
            num=ord(stuff(j,i)(1:1))
            IF (num.ge.1.and.num.le.10) THEN
              region(j,i)=num
              call addregion(num,Regions,NoRegions,MaxRegions)
            end if
          ELSE
c
c  Outside domain
c
            Domain(j,i)=.false.
          END IF
 2010   CONTINUE
 2000 CONTINUE
      IF (NoOut.eq.0) THEN
        WRITE (*,*) 'No Outlet node has been specified in'
     &             ,' the boundary file'
        STOP
      ELSE
        init=NoOut
      END IF
      CLOSE(unit=FIOUnit, status='keep')
      DO 1030 i=1,ky+1
        Domain(kx+1,i)=.false.
 1030 CONTINUE
      DO 1040 i=1,kx+1
        Domain(i,ky+1)=.false.
 1040 CONTINUE
      WRITE (*,*) ' Irregular boundaries have been input'
      WRITE (*,*) ' Total catchment area = ',area
      WRITE (*,*) ' No of specified catchment outlets',init
      RETURN
      END
c
c
===========================================================================
c   Finding the ordinal number of an input character. Returns a -ve value if
c   the input character is not recognised.
c
===========================================================================
c
      integer function ord(tr)
      implicit none
       character*1 tr
c
```

```
      integer no,i
      parameter (no=36)
       character*(no) ta
       data ta / '1234567890abcdefghijklmnopqrstuvwxyz' /
c
      do 1001 i=1,no
        if (ta(i:i).eq.tr) go to 1100
 1001 continue
      ord=-1
      return
 1100 ord=i
      return
       end
c
c
============================================================================
c   Add  a  region  to  the  array  keeping  track  of  what  regions  have  been
specified
c
============================================================================
c
      subroutine addregion(num,regions,noregions,Maxregions)
      implicit none
c
      integer Maxregions,num,regions(Maxregions),noregions
c
      integer i
c
      do 1000 i=1,noregions
        if (regions(i).eq.0) go to 1100
        if (regions(i).eq.num) return
 1000 continue
      if(noregions.eq.Maxregions) then
        write (*,*) ' -- Too many regions in InputBoundaries > '
     &               ,Maxregions
        stop
      else
        noregions=noregions+1
        regions(noregions)=num
      end if
 1100 write (*,*) ' -- Internal error in  AddRegion'
      stop
      end
```

Appendix D: Shell program to analyse RST and BND file formats

This is a sample code that can read in RST and BND files for analysis. It uses the routines in INOUT.

```
      PROGRAM SiberiaShell
      IMPLICIT NONE
c
c            Created by G Willgoose 16/4/93.
c            ------------------------------
c
c   This code segment provides a shell that reads in a SIBERIA
c   boundary and data file names, and reads in the files. It uses
c   the routines INOUT.F from SIBERIA to read version 7.03 type RST2 and BND
c   files. The model parameters are stored in a COMMON pulled in from
c   SIBERIA version 7.03.
c
c
C   REQUIREMENTS
c   ============
c
c     To link and compile this code segment you must have the files
c        INOUT.F         - the input/output routines
c        STRSUPPORT.F    - a string manipulation package required by
c                          INOUT.F
c        PARAMETERS.INC  - an include file with the simulation
c                          run parameters
c        GRIDSIZE1.INC   - an include file required by INOUT.F
c
c
c
c
c   EXPLANATION OF THE PARAMETERS USED IN THE INPUT ROUTINES
c   ========================================================
c
c  SUBROUTINE READIN
C  -----------------
c
c  slope        = slopes for DTM
c  vz           = random field for DTM
c  channels     = channelisation of the DTM
c  Elevations   = elevations of the DTM
c  area         = area draining through the DTM
c  directions   = flow directions for the DTM
c  grid         = storage size of the arrays above (x,y the same)
c  RSTfile      = name of restart file to be read
c  init         = no of fixed elevation points
c  ixxx,iyyy    = x,y coordinates of the init fixed points
c  FilenameUser(MaxUser) = user defined filenames
C
C  SUBROUTINE INPUTBOUNDARIES
C  -------------------------
c
c  BoundFile = name of file to be read
c  domain    = logical array with the data of what nodes are in the domain
c  gridsize  = the declared size of domain (same in x and y directions)
c  kx,ky     = x and y sizes of the domain
c  init      = no of fixed elevation points
```

```
c  ixxx,iyyy = the x and y coordinates of the fixed elevation points
c
c  NB: init and ixxx,iyyy input in InputBoundaries overwrite the same
c      data input by readin. This is normal practice.
c
c
c
c  EXPLANATION OF THE PARAMETERS OF THE SIMULATION
C  ===============================================
c
1:  RunTime     = total no of time steps TO solve for
c             -1,-2 are automated convergence criteria
c   2:  StatsTime   = line printer output of contours at this time increment
c   3:  kx    = X DIMENSION of the rectangular Grid
c   4:  ky    = Y DIMENSION of the rectangular Grid
c   5:  modeIC = type of initial condition TO be applied TO initial
c              elevations
c                0  =  flat level surface with initial value = SInit
c                1  =  surface sloping upwards in the +ve X direction
c                      with max height = SInit
c   6:  TimeUp= END of time for application of the tetconic uplift
c   7:  ModeSolver = mode of solution of the sediment transport relation
c             0 (disabled) => soln. for elevations by Taylor Series
c             1            => soln. of the physical transport equation
c             3            => as for ModeSolver=4 except the
c                               TwoFluvialProcesses occur everywhere
c                               (additive sediment transport processes)
c             4            => corrected Version of ModeSolver=5 with Correct
c                               BC implementation for Area=1.
C                               For TwoFluvialProcesses
c                               channel/hillslope processes switch
c                              (switching sediment transport proceses)
c             5            => soln. by explicit/analytic soln
c                               (predictor/corrector) - nonlinear
c             6            => shear stress driven source limitation model
c             7            => equivalent to ModeSolver=5
c                               except there is no erosion
c                               at all on the hillslope
c   8:  ModeDir = mode of solution of the directions in the DirAnal routine
c             0 (default) => directions as steepest Slope.
c             1           => directions for the channels are writ in stone
c                              provided that channels drain into channels
c             2           => use ModeDir=1 except READ in allowable drainage
c                              directions from a user defined file (used TO
c                              model contour bank constraints on drainage)
c             3           => directions DO not change (set from RST2 file)
c             4           => Highly optimised version of ModeDir=0
c                              for RISC processers
c             5           => R8 implementation
c             6           => Average flow direction for a region
c                              rather just adjacent nodes
c             7           => Dinifinity implementation. Not done yet.
c   9:  ModeUplift = mode of tectonic uplift perturbation
c             0 (default) => No perturbation
```

```
c              1           => Sinusoidal with amplitude TAmp, Period Tper,
c                             and initial phase TPhase
c              2           => Square wave with parameters as for sinusoidal
c              3           => impulse uplift with height TAmp, Period Tper,
c                             and initial phase TPhase
c   10: ModeRandom = mode of Random perturbation using FRanCV
c              0 (default) => No Perturbation
c              1           => *b1
c              2           => *b5
c
c   11: ModeErode = mode for use in the user-defined FACTOR routine
c              0 (default) => spatially constant FACTOR
c              1           => depth dependent armouring erodibility
c                             (1st used in J Bell BE thesis)
c              3           => spatially variable erodibility based on.
C                             RGN input
c             +20          => erodibility with correction for
c                             physical Grid dimensions
c
c   12: ModeRunoff = mode for use in the user-defined RUNOFF routine
c              0 (default) => spatially constant RUNOFF
c              1           => spatially variable runoff from a file
c                             of grid data (first
c                             used in Hancock (1997) PhD).
c              2           => known inflows from offsite
c                            (first used in J Bell BE thesis)
c              3           => spatially variable runoff based on .RGN input.
c             +20          => runoff with correction for physical
c                             Grid dimensions
c   13: ModeChannel = mode for incorporation of GULLY/Channel dimensions
c              0 (default) => zero dimensions
c              1           => width and depth power fns of Area
c              2           => width and depth power fns of discharge
c              3           => simply ignore any variations in erodibility
c                             between channel and hillslope
c                        (i.e. assume Ot=1 no matter what you sent Ot to be).
c   14: ModeDP = mode for incorporation of DEPENDENT variables
c               (no modes implemented to date).
c   15: ModeMC = mode of MonteCarlo run generation
c              0           => default deterministic run.
c              1           => multiple input .RST2 files for MC assessment of
initial conditions
c                             (only impelmented in the PVM version
c                               ... latest version 7.05).
c   16: DirReg = how large is the Region used TO calculate the drainage
c                directions relative TO using the adjacent points in
c                ModeDir=6 (default =1).
c   17: ModeSoil = the soil development model TO be used (work in progress)
c                  0 => no modelling
c                  1 => SM and depth model
c   18-20: idummy18-idummy20 = N/A
c
C
c                 REAL PARAMETERS
C                 ==============
c  21: dZ     = Diffusivity in sediment transport
c  22: dZn    = power of nonlinearity in the diffusion of sediment transport
```

```
c  23: dZHold = Threshold below which diffusion transport does not occur
c  24: QsHold = Threshold below which fluvial transport does not occur
c  25: FactMx = The maximum value for FACTOR in SedAnal used in the
c              calculation of Ot.
c  26: FRanMn= Mean factor for random fluctuations (see also FRanCV)
c  27: 1/at  = factor on activator in differentiation equation
c              0.025/c1 = activator threshold, a
c                                                 t
c  28: YFix  = The CIF allocated TO fixed Elevation points
c              0  =>  all fixed Elevation points are hillslope
c              1  =>  all fixed Elevation points are channels
c  29: FRanCV= coeff. of var for random fluctuations (see also FRanMn)
c  30: b3SDs = Standard Dev. for short term variations in the runoff rate
c              (used in variation of saturation from below Regions)
c  31: b3SDl = Standard Dev. for long term variations in the runoff rate
c              (used in variation of Channel head position)
c  32: TAmp  = Amplitude of the cyclic uplift
c  33: TPeriod = Period of the cyclic uplift (timesteps)
c  34: TPhase= Phase of the cyclic uplift at t=0
c  35: FRanZ = standard dev. factor for random fluctuations in the Elevation
c              initial conditions
c  36: a1    = factor relating the discharge used in Q and a
c                                                      s
c                     m
c                      3
c          Q  =  b  A                      (discharge-Area formula)
c                 3
c
c  37: m3    = power on the Area in discharge
c  38: b3    = coefficient between discharge and Area in the sediment
c              transport formula
c
c                         m   m     n
c                          3   1     1
c          Q  = b . (b .A   )    .S              (sediment transport
c           s    1    3                              formula )
c
c
c  39: b1    = coefficient on the front of the sediment transport formula
c  40: m1    = power of the discharge in the sediment transport formula
c  41: n1    = power on the Slope in the sediment transport formula
c
c  42: Bulk =  Bulk density of soil. Relates the mass rate of
c               transport with the actual amount of Elevation change.
c  43: InitTimeStep  =  +ve  => time step size TO be used
c                       -ve  => % error criteria to used in adaptive
c                               timestepping
c
c
c
c                             m   m     n
c                              3   5     5
c          a  =  b .  (a1 b A    )   . S    (Channel initiation equation)
c                 5        3
c
c          PRIOR TO V6.34 the interpretation of this equation was
c             (NOTE parameters for Restart files for these early Versions
c                   are automatically adjusted TO comply with the
c                   interpretation above. see SUBROUTINE ReadIn)
c
```

```
c                       m     n
c                        5     5
c           a  =  b . A   . S         (Channel initiation equation)
c                  5
c
c
c  44: b5    = coefficient on CIF
c  45: n5    = power on the Slope used in the CIF equation
c  46: SInit = for uniform initial Elevation gives starting value
c  47: m5    = power on Area in the CIF equation
c  48: YHold = threshold used in the determination of the drawdown in the
c              Channel when going from "overland" TO Channel.
c  49: Notch = time over which SInit Elevation change is applied TO the Notch
c  50: Cover = Vegetation Cover factor TO adjust b1 TO reflect veg effects
c              on soil erodibility (ala USLE)
c  51: s0max = Maximum Slope that can be reached due TO landslide stability
c              (default=0.0 => inactive)
c  52: DTime = factor TO adjust the time scale of Channel formation
c              timescale = 1/DTime
c  53: OTime = factor TO adjust the relative rates of overland TO
c              Channel sediment transport adjustment
c  54: GridXY= the spacing of the XY Grid in m
c  55: East  = the easting of the bottom left hand corner of the Grid
c  56: North = the northing of the bottom left hand corner of the Grid
c  57: b6    = the coefficent in the Channel dimensions model
c  58: m6    = the power in the Channel dimensions model
c
c                            m6
c              depth = b  A              A is used for ModeChannel=1
c                       6                Q is used for ModeChannel=2
c
c  59: b12   =  b1 for a second fluvial sediment transport process
c               IF b12=0.0 THEN model does not use a second process
c               default b12=0
c  60: m12   =  m1 for a second fluvial sediment transport process
c               (IF b12 <> 0)
c  61: SDRate = the rate PARAMETER for the soil development model
c  62: SDExp1 = the First exponent on the soil development model
c               SDModel=1 => exponent of soil moisture
c  63: SDExp2 = the second exponent on the soil development model
c               SDModel=1 => exponent on the soil depth
c  64: SMThreshold = the threshold for saturation excess runoff
c  65-70: rdummy45-rdummy50 = N/A
c
c
c                        FileName PARAMETERS
c                       =====================
c      These filenames are generically named FilenameUser(index 1..10) and
```

```
c      are used in the user defined modules in the model. The internal
c      ModeSolver
c      they are used for determination of follow.
c
c   1:  Factor     = Used in SedAnal TO determine the b1 in the fluvial
c                    erosion module. Accessed in User-Factor.
```

```
c   2:  Runoff     = Used in SedAnal, CtrOut and Finite TO determine the
c                    relationship for discharge at any point. Accessed in
c                    User-Runoff.
c   3:  Uplift     = Used in SedAnal TO determine the tectonic uplift at any
c                    time. Used in User-Uplift.
c   4:  Directions = Used in DirAnal for determine constraints on drainage
c                    directions TO determine slopes and areas. Used in
c                    User-DirAnal.
c   5-9: N/A.
c   10:  Others    = Used in SIBERIA main loop TO determine user-defined
c                    dependent variables that depend on SIBERIA output but
c                    DO not impact the operation of SIBERA except potentially
c                    through other user defined modules. Accessed in User-
c                    Others.
c  MaxUser should not be changed
c  Both grid and NoFix can be increased. They should not decreased below 200
c  and 1000 respectively otherwise you may not be able to read some RST2
files.
c
      INTEGER grid,NoFix,MaxUser, MaxRegions
      REAL*4 version
      PARAMETER (grid=200,NoFix=1000,MaxUser=10, MaxRegions=26, version 8.00)
c
      INCLUDE 'parameters.inc'
c
      REAL*8 slope(grid,grid),vz(grid,grid),channels(grid,grid)
     &      ,Elevations(grid,grid),cDepth(grid,grid),SoilDepth(grid,grid)
      INTEGER Area(grid,grid),directions(grid,grid),RegionMap(MaxRegions)
     &      ,init,ixxx(Nofix),iyyy(NoFix),LowX,HighX,LowY,HighY
      LOGICAL IrregularBoundaries,Domain(grid,grid)
      CHARACTER*80 RSTfile,BoundFile,FilenameUser(MaxUser)
c
      CHARACTER*5 ofext
      LOGICAL detcif, FlipLR, FlipTB
c
      IrregularBoundaries=.false.
      write (*,*) 'Input RST filename'
      read(*,6000) RSTfile
      Boundfile(1:5)='     '
      write (*,*) 'InputBND filename'
      read(*,6000) Boundfile
 6000 format(a80)
c
c    read in RST files
c
      FlipLR=.false.
      FlipTB=.false.
      CALL ReadIn(slope,vz,channels,elevations,Area,grid,grid
     &      ,RSTfile,iInit,iXXX,iYYY,Directions,cDepth,SoilDepth
     &      ,FlipLR,FlipTB)
c
c    read in boundary files
c
      IF (Boundfile(1:5).ne.'     ') THEN
        CALL InputBoundaries(BoundFile,Domain,Grid,Grid,kx,ky
     &         ,iInit,iXXX,iYYY,Regions,NoRegions,MaxRegions,Region
     &         ,RegionMap,LowX,HighX,LowY,HighY)
        IrregularBoundaries=.true.
      ELSE
        LowX=2
```

```
        HighX=kx
        LowY=2
        HighY=ky
      END IF
c
c   PUT YOUR STUFF HERE
c

c  ------------------------------------------------------------
c  if you want to write out a RST2 file use the following line
c  some things are not intialised by the reading programs so need
c  values put in
c  ------------------------------------------------------------
c  filename should be the name of the file to be output, say 'junk'.
      filename='junk'
      detcif=.true.
c  We are generating a RST2 file.
      ofext='.RST2'
c  'time,iscale,er,nr,cvar,sed' are not needed when generating RST2 files.
      Time=0.0
      Iscale=1
      Er=0
      Nr=0
      CALL WrtOut(vz,channels,Elevations,Area,grid,grid,FileName
     &               ,iInit,iXXX,iYYY,Direct,Time
     &               ,IrregularBoundary,Domain,slope,DetCIF,Version
     &               ,ofext,iscale,er,nr,Sed,cDepth,SoilDepth)
      STOP
      END
```