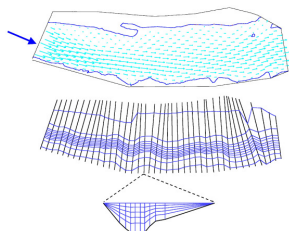
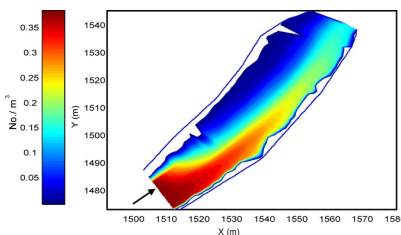


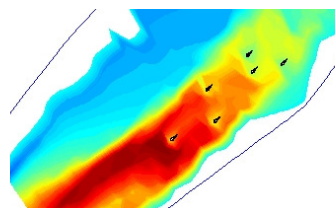
Flow Related Models for Simulating:



River Hydraulics



Invertebrate Drift Transport



Foraging Energetics of Drift-Feeding Salmonids

In association with



US Bureau of Land Management



University of Alaska

June 2012

Flow Related Models for Simulating River Hydraulics, Invertebrate Drift Transport, and Foraging Energetics of Drift-Feeding Salmonids:

User Guide (*Version 1.1*)

Lon Kelly, Joe Hay,
Nicholas Hughes, Eric Goodwin and John Hayes

Cawthron Institute
98 Halifax Street East
Private Bag 2
NELSON
NEW ZEALAND

Phone: +64.3.548.2319
Fax: +64.3.546.9464
Email: info@cawthron.org.nz

Information contained in this report may not be used without the prior consent of the client

EXECUTIVE SUMMARY

This is a guide to a modelling suite that takes a river hydraulic survey, determines the invertebrate drift transport in the study reach, and predicts the foraging energetics of drift-feeding salmonids. The *R2DStreamTubes* tool converts a two-dimensional hydraulic model (River2D) into a StreamTubes representation. The river is represented as a bundle of longitudinal streamtubes, each of which conveys an equal proportion of the river's total flow. The area of each streamtube may change along its length to reflect changes in flow velocity. The *invertebrate drift* transport model calculates the processes of entry, bulk transport, dispersal, and settling of stream invertebrates, resulting in the spatial density of invertebrate drift in the study reach. The *salmonid foraging* model predicts spatially explicit net rate of energy intake (NREI) which can be easily converted to growth potential. In this guide we focus on the way these three tools can be used together to predict how stream discharge and invertebrate drift concentration influence predicted feeding locations and the carrying capacity of a stream reach for large brown trout, but the tools we describe could also be adapted to address a variety of other questions in river ecology.

TABLE OF CONTENTS

1	INTRODUCTION	1
2	OVERVIEW OF MODELS	2
2.1	StreamTubes model.....	3
2.2	Invertebrate drift model component.....	4
2.3	Foraging model component.....	4
3	STREAMTUBES MODEL	4
3.1	Purpose of the model.....	4
3.2	Opening and displaying flow data.....	5
3.3	Creating cross-sections	6
3.4	Creating streamtubes.....	8
3.5	Subdividing vertically, and saving streamtubes	9
3.6	Querying data at points or cross-sections.....	10
4	INVERTEBRATE DRIFT MODEL	11
4.1	Purpose of the model.....	11
4.2	Mechanics	11
4.3	Drift model data requirements	13
4.4	Modelling invertebrate drift	15
4.5	Saving the drift model output.....	16
4.6	Altering the model parameters	18
4.7	Command line control.....	19
5	SALMONID FORAGING MODEL.....	20
5.1	Purpose of the model.....	20
5.2	Modelling NREI.....	21
5.3	Foraging model outputs	22
6	MODELLING CARRYING CAPACITY	25
6.1	Manual approach.....	25
6.2	Automated approach	27
6.3	Defence radius	28
7	ACKNOWLEDGEMENTS	28
8	DISCLAIMER	29
9	REFERENCES:	29

LIST OF TABLES

Table 1 Parameters for the two swimming speed equations for steelhead (rainbow) trout (from Fish Bioenergetics 3.0 (Hanson <i>et al.</i> 1997)).	39
Table 2 Parameters for the brown trout swimming cost equation (from Hayes <i>et al.</i> (2000) – a , b_1 , b_2 were sourced from Elliott (1976) and b_3 was sourced from Rand <i>et al.</i> 1993 and Stewart (1980) – cited from Rao (1968)).	39

LIST OF FIGURES

Figure 1 Flow chart of modelling process. Model names, as well as their input and output files, including their appropriate file extensions.	2
Figure 2 Velocity vectors describing a 2D flow solution produced by River2D for a surveyed pool (top) converted to cross-sections (centre, black) and streamtubes (centre, blue) using the StreamTubes model. One cross sectional view through the StreamTubes grid is shown at the bottom, not to scale.	5
Figure 3 R2DStreamTubes.exe interface window with boundary displayed and first cross-section drawn.	6
Figure 4 R2DStreamTubes.exe interface window with boundary and cross-sections displayed.	8
Figure 5 Dialog box for specifying the number of horizontal streamtube divisions.	9
Figure 6 R2DStreamTubes.exe interface window with boundary, cross-sections and horizontal streamtubes displayed in plan view.	9
Figure 7 Dialog box with prompt for number of vertical streamtubes, and tick box to specify constant velocity, rather than default logarithmic velocity, profile with depth.	10
Figure 8 Example of a StreamTubes output file. See text for explanation of values.	10
Figure 9 The lateral exchange of drift between streamtube cells in a cross-section, based on a series of Gaussian distributions. Each different colour-shaded area denotes the proportion of drift from a lateral position in source streamtube A exchanged into destination streamtubes B and C. The total proportion of drift exchanged from streamtube A with streamtubes to the left and right is based on the average overlap of the destination streamtube with the Gaussian distributions centred on all possible lateral positions within column A [see Appendix 1].	12
Figure 10 A plan view of modelled invertebrate drift dispersed down through a river reach. For this illustration the centre four streamtubes were given an initial concentration at the top of the reach of zero, so the invertebrate density in these streamtubes lower in the reach has dispersed in from streamtubes on either side. The settling rate was set to zero in this model run, and the arrow indicates the direction of flow.	13
Figure 11 An invertebrate parameters text file.	14
Figure 12 A drift density initialization text file to initialise the first cross-section with concentrations for 4 drift types.	15
Figure 13 The DriftModel.exe menu bar	15
Figure 14 An output file generated by CrossSections > ConcentrationsDump from the drift model	16
Figure 15 An output file generated by CrossSections > Dump for Foraging Model from the drift model. This file format has sufficient data to reconstruct the volumetric invertebrate density distribution for graphical display, further investigation, evaluation, validation etc. Note that the coordinates provided in the file are for the corners of cells, while the invertebrate densities specified are for the centres of the cells thus defined. Some external manipulation will be required.	17
Figure 16 A cross-section dump from the drift model	18
Figure 17 The Set H Params dialog box	19
Figure 18 Plan view of the foraging model showing the geometry of prey interception.	21
Figure 19 Cross-sectional view of the geometry used by the foraging model showing: fish position; foraging radials; and predicted prey capture area interpolated from predicted foraging radius along each foraging radial (based on prey reaction distance and water velocity) (Hughes <i>et al.</i> 2003)	21
Figure 20 A paramsIn.txt file, specifying fish and habitat parameters.	22
Figure 21 A resultsOutNREI.txt output from the foraging model	23
Figure 22 Example of resultsOutMaxNREI.txt output from the foraging model	24

Figure 23	Example of the epsOut.eps output from the foraging model	24
	Note that changes in the eps file format in different versions of the Windows operating system means that as of 2012 this output does not render correctly.	24
Figure 24	A resultsOutForaging.txt output of the foraging model.....	25

LIST OF APPENDICES

Appendix 1	Drift transport model description	33
A1-1	Values derived from the streamtube description	33
A1-2	Model of dispersion	33
A1-3	Model of settling.....	35
A1-4	Model of entry	36
Appendix 2	Swimming cost options in the NREI model	37
A2-1	Polynomial swimming cost model – coho salmon	37
A2-2	Rainbow trout swimming cost models	37
A2-3	Brown trout swimming cost model.....	39
A3-1	Matlab programs for displaying model outputs.....	41
A3-1.1	<i>Using the Matlab programs</i>	41
A3-2	Displaying outputs using River 2D	42

1 INTRODUCTION

This is a guide to a suite of software tools for modelling river hydraulics, invertebrate drift transport, and foraging energetics of drift-feeding salmonids. A *StreamTubes* representation describes river flow using an array of streamtubes, each of which conveys an equal proportion of the total discharge, and the *R2DStreamTubes* tool generates one of those representations. The *invertebrate drift* transport model predicts how the processes of entry, transport, dispersion, and settling, will determine the spatial distribution of invertebrate drift density in a river reach. The *salmonid foraging* model predicts spatially explicit net rate of energy intake for fish feeding on drifting invertebrates. In this guide we describe how these three tools can be used together to predict the effects of stream discharge and invertebrate drift concentration on predicted feeding locations and the carrying capacity of a stream reach for large brown trout. The tools we describe could be adapted to address a variety of other questions in river ecology.

When used together, these three software tools provide a process-based modelling approach capable of predicting how changes in stream discharge, bed topography, water temperature, and the abundance and behaviour of stream invertebrates will influence the quality and quantity of habitat for drift-feeding fish. The outputs from one component become the inputs for the next (Figure 1). The administrative aspects of managing the input and output files, and assembling the results, can be handled by a script in a language such as R, Matlab etc. The process-based approach is an advance over existing instream habitat procedures, such as habitat modelling within the Instream Flow Incremental Methodology (IFIM), because its predictions of growth rate and fish abundance are more readily understood and biologically realistic than abstract indices of habitat suitability (*e.g.* Weighted Usable Area).

Habitat modelling has been used for many years in attempts to predict the impact of changing flow regimes on stream life, particularly salmonids. The Instream Flow Incremental Methodology is the most common framework within which habitat models have been applied since the 1970s (Bovee 1982; Stalnaker *et al.* 1995; Annear *et al.* 2002). A recurring criticism of conventional habitat modelling has been its reliance on empirically derived habitat suitability curves, relating observed habitat use to physical habitat characteristics. Empirical habitat suitability curves and the linkage between fish and their food supply are two main areas in which conventional IFIM is weak in biological realism (Orth 1987). The impacts of aspects other than physical habitat on the maintenance of fish populations have generally not been taken into account in these models. This can potentially lead to a situation where two locations are judged to be of equal habitat value based on their physical characteristics, when in fact they might differ markedly in their ability to support fish because of spatial variation in food availability.

In contrast, our process-based modelling approach focuses on the energy balance of fish as a basis for assessing habitat quality and quantity. It recognizes that fish cannot survive, grow or reproduce if the energy available from their food intake does not exceed the energy expended in obtaining that food. The modelling process focuses on how discharge affects the availability of food to drift feeding fishes, as well as their energy expenditure while foraging. Modelling the transport and dispersion of invertebrate drift in relation to flow alleviates the need to assume uniform drift densities in the foraging model, which has been a limitation of some earlier models (*e.g.* Hughes 1992; Addley 1993; Guensch *et al.* 2001). By modelling these dynamics in a spatially explicit way, comparisons can be made between the energetic profitability of different areas for drift

feeding fishes. Energetic profitability (net rate of energy intake) can be further interpreted to predict spatially explicit growth potential, as well as numbers and the distribution of fish in a river reach.

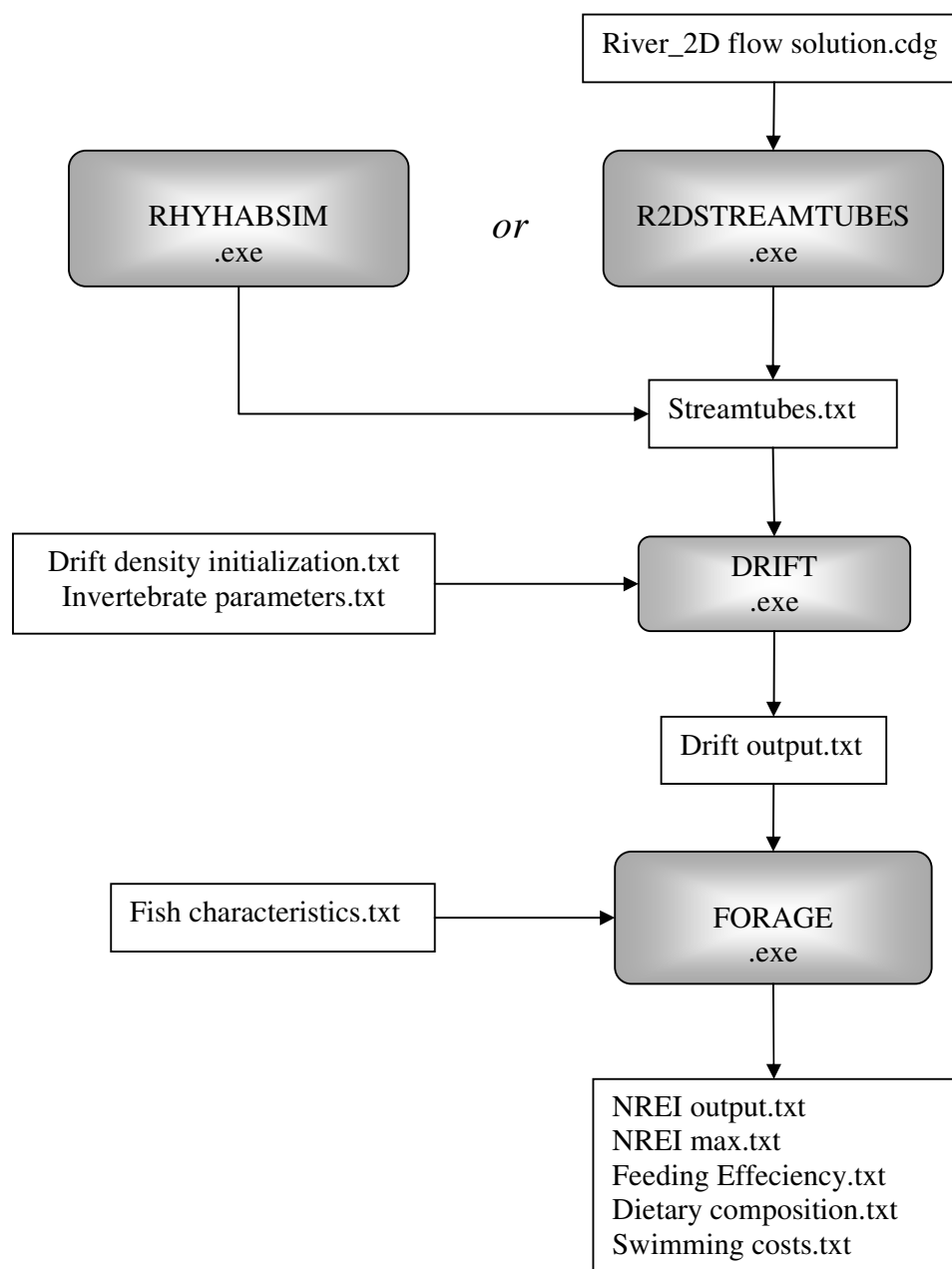


Figure 1 Flow chart of modelling process. Model names, as well as their input and output files, including their appropriate file extensions

2 OVERVIEW OF MODELS

The three model components—StreamTubes, drift transport and foraging—are implemented in software for use by technically-minded workers. They are implemented

as Windows programs. They are run sequentially, and produce considerable amounts of intermediate output. The drift and foraging models have command-line variants. The extensive and variably formatted output in combination with the ability to work through a modelling investigation step-by-step slows the modelling process; but, on the other hand, these features allow users to ensure that the programs are operating sensibly and producing outputs consistent with the inputs, and to more easily troubleshoot and customize the software (by editing and recompiling) or the data processing pipeline. Furthermore, once correct operation has been determined it can be managed by script, run in any language that can interact with the operating system, including the Windows batch language, R and Matlab. This allows multiple modelling runs to be automated, with a set of configuration data passed to the model sequentially (one after the other), and the set of results archived systematically.

2.1 StreamTubes model

The StreamTubes model generates a mathematical grid representation of the study reach. The irregular 3D grid of a StreamTubes representation provides a convenient way of describing flow through the complex topography of a modelled stream reach. A stacked array of streamtubes, each divided lengthwise into many cells, allows spatial variation in water velocities to be described in X, Y and Z directions. The streamtube¹ concept simplifies calculations relating to water flow, and hence the downstream transport of invertebrate drift and subsequent prey capture by drift-feeding fish. The flow of water (volume per time) down a streamtube is conserved along its length (from cell to cell), while the cross-section area of the streamtube (and hence velocity of its flow) may vary. Variation in cross section area provides an indication of variation in water velocity along the tube.

Our R2DStreamTubes program is designed to construct a StreamTubes representation of flow from a two-dimensional depth averaged flow output as generated by River 2D (Steffler *et al.* 2003). The River 2D surface is divided into a user-defined number of cross-sections. Spanning the river, each of these will have equal discharge through it. As these will cross the subsequent streamtubes, their number determines the number of cells per streamtube length. Each virtual cross-section is subdivided by a common user-defined number, generating a series of streamtubes with each conveying an equal fraction of the total discharge through the cross-section. In the vertical dimension, water velocities for each streamtube are calculated assuming either a logarithmic velocity profile with depth (Gordon *et al.* 1992), or constant velocity with depth.

Although this manual describes a model for constructing a StreamTubes representation from a hydrological model from River 2D, the output from 1D hydraulic surveys (as undertaken in traditional IFIM habitat modelling) can also be converted to a StreamTubes representation, and used as input to the drift and foraging models. For example, Version 3.30 of Rhyabsim (Jowett 1999) incorporates a StreamTubes conversion process. It can output simulated flow information, for a series of cross-sections in a representative reach, as a StreamTubes representation of that river reach².

¹ In this document, StreamTubes is capitalised when it refers to the software tool or the method of representing a river reach, and not capitalised when referring to an individual streamtube or the hydrological concept of a streamtube. If this seems arbitrary, think of the capitalised version as a brand. Batteries not included.

² Note though, that a difference between R2DStreamTubes and RhyHabSim, in the convention of orienting cross sections, means a separate version of the Drift model needs to be used for these two sources of StreamTubes grid files.

The StreamTubes representation of the study reach provides flow information in the format required for input to the Drift and Foraging models.

2.2 Invertebrate drift model component

The invertebrate drift model uses the flow description in the StreamTubes representation of the study reach to predict three-dimensional spatial variation in invertebrate drift density. The model is initialized by specifying invertebrate drift densities at the most upstream cell of each streamtube, along with relevant physical and behavioural characteristics of the drifting invertebrates (such as their settling velocities and rates of entry into the water column). The model then simulates the processes of entry into the flow, downstream transport, turbulent mixing and settling to estimate the concentration of invertebrate drift at each subsequent downstream cell of each streamtube.

The invertebrate drift model can be used to investigate how changes in stream discharge will influence the spatial distribution of invertebrate drift.

2.3 Foraging model component

The foraging model (a modification of that described in Hughes *et al.* 2003) uses the spatially explicit flow description in the streamtubes representation of the study reach, and the spatially explicit invertebrate drift concentrations provided by the drift model, to predict the spatially explicit net energy gain of fish drift-feeding in that study reach.

The foraging model describes the fish's prey capture rate as a function of water depth and velocity, invertebrate drift density, and the fish's visual and swimming abilities.

The foraging model estimates the energy gained due to prey capture and the energy expended due to foraging (which is also influenced by water temperature), to predict Net Rate of Energy Intake (NREI) at all cells in the modelled reach. Positive NREI values indicate that fish should be able to maintain condition or grow, at least in the short term.

The foraging model will determine the single optimum foraging location for a trout of a given size, based on maximum NREI, of all cells in the streamtubes representation in its input file. If multiple fish locations are needed the user needs to implement special program management by script, as discussed below.

3 STREAMTUBES MODEL

3.1 Purpose of the model

This process converts a River2D (Steffler *et al.* 2003) description of the flow through a stream reach into a StreamTubes representation with a number of equal discharge streamtubes. The process is embedded in a custom variant of Steffler's River2D program called R2DStreamTubes, with the interface that will be familiar to users of that software. A flow solution in River 2D CDG file format is the sole input required for this program (Figure 1).

The reach description in River2D format is first divided by a series of cross-sections, which are then subdivided, horizontally and vertically, into an array of polygons each of

which is connected with its corresponding polygon in adjacent cross-sections to form tubes, each assigned an equal proportion of the total flow (Figure 2). This StreamTubes representation of stream flow is required for the drift and foraging models.

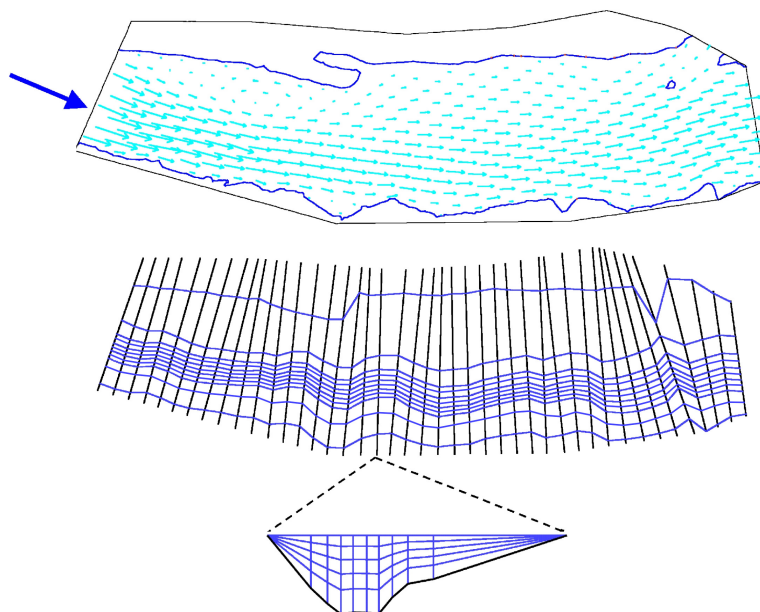


Figure 2 Velocity vectors describing a 2D flow solution produced by River2D for a surveyed pool (top) converted to cross-sections (centre, black) and streamtubes (centre, blue) using the StreamTubes model. One cross sectional view through the StreamTubes grid is shown at the bottom, not to scale.

The conversion of two-dimensional flow data, from a River 2D flow solution, into a StreamTubes representation, is performed interactively within River2D. Since the flow information in a River 2D flow solution is depth-averaged, the StreamTubes model uses a theoretical velocity / depth profile to estimate velocities at all depths from the depth-averaged input. The default profile is the text book logarithmic velocity / depth profile, with slower velocity near the bed (Gordon *et al.* 1992). However, a uniform velocity / depth profile can also be specified.

3.2 Opening and displaying flow data

Open R2DStreamTubes.exe by double clicking on its icon. You will be presented with a graphical display window based on the River 2D Bed window (Figure 3). Use the File > Open menu selection to open a flow model solution output in CDG format. The file will open with the nodes and boundary displayed.

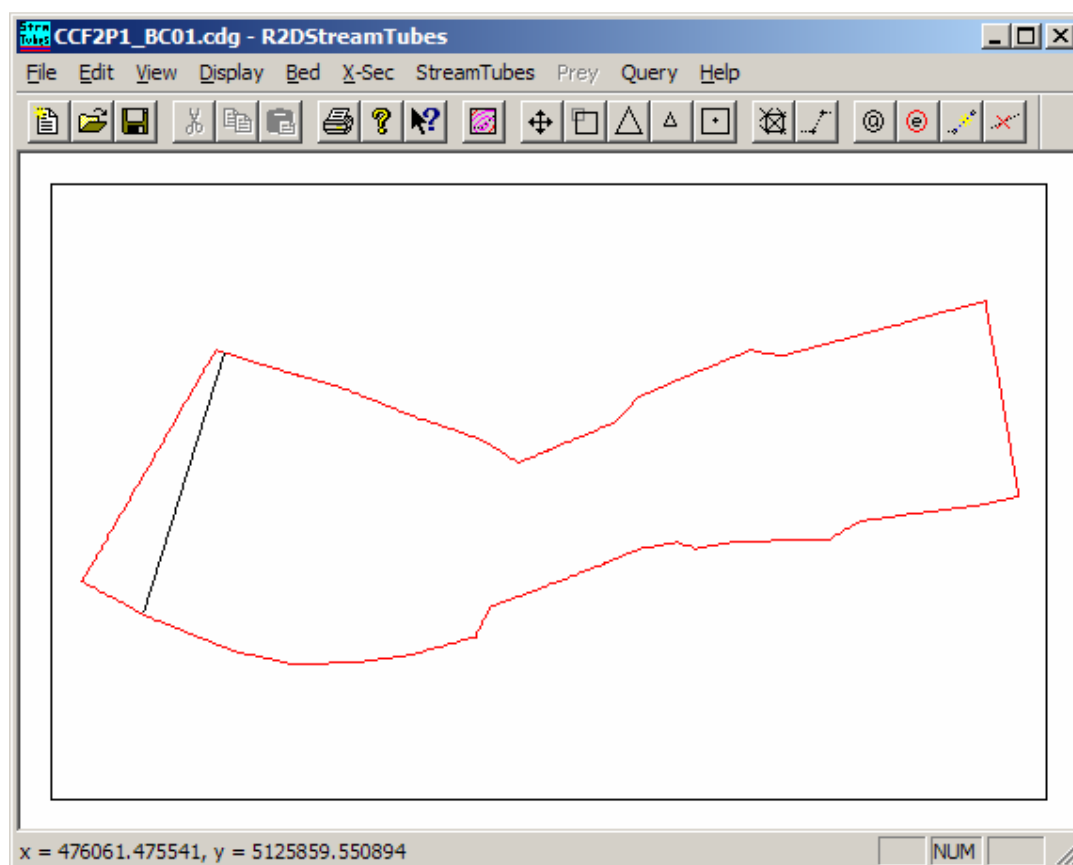


Figure 3 R2DStreamTubes.exe interface window with boundary displayed and first cross-section drawn.

The first step is to triangulate a mesh from the topography data and interpolate flow values, using the Bed > Triangulate menu item. None of the program's other functions will work until you instruct the program to triangulate the CDG mesh. The triangulation process may take a moment. The resulting triangulated mesh can be displayed by toggling on Display > Triangulation.

You are now able to display a number of properties (*e.g.* bed elevation, water depth or velocity) as contour lines or colour shading in the same way as in River 2D programs. As the mouse is moved around in the display window the Status Bar displays the X and Y coordinates, the bed elevation, water depth and flow in the X and Y directions.

3.3 Creating cross-sections

The first step in the conversion to StreamTubes representation is to divide the modelled reach by a series of cross-sections. This process is performed using the functions in the X-Sec menu. Cross-section locations can be specified explicitly by selecting X-Sec > Draw and then double-clicking on the desired location in the display window. This constructs a cross-section perpendicular to the maximum discharge at that location (black line, Figure 3). As cross-sections are created they are oriented with the first point (origin) on the true right bank³. Care should be taken to ensure that this initial cross-

³ Note that in RhyHabSim the origin of each cross section is on the true left bank.

section is at the upstream end of the reach, and spans from one no-flow boundary (a bank) to the other. X-sec > Clear all will clear all cross-sections from the modelled area, for instance if the first manually-positioned one does not span from one bank to the other.

Cross-sections can also be constructed by specifying the coordinates of their end points, by selecting Specify in the X-Sec menu and entering the endpoint coordinates of the desired cross-section into the dialog box.

The X-Sec > Fill command automatically constructs a series of cross-sections at a user-specified spacing downstream from the initial cross-section (Figure 4). The program estimates the point of maximum discharge on a cross-section, moves downstream (in the direction of this maximum flow) by the distance the user has specified for the spacing, and then orients the new cross-section at right angles to the average flow, repeating the process until the end of the modelled reach.

The optimal spacing for cross-sections depends on the scale of interest in the model, balancing resolution against processing time. If, for example, you are interested in modelling the distribution of small fish, with small foraging areas, then cross-sections should be more closely spaced. As a starting point, we have spaced cross-sections at between one and two times the length of the fish we intend to model.

Once the series of cross-sections has been constructed the X-Sec > Fix Crossing command should be employed to resolve any cross-sections that intersect one another. This is implemented by searching downstream; when a cross-section is found to intersect the cross-section upstream of it, one endpoint is moved to coincide with the endpoint of the intersected upstream cross-section. Thus the intersection is removed at the expense of having the average flow at something other than right angles to the revised cross-section. Implicit in this action is the realisation that in complicated hydraulic situations, the positioning of cross-sections is somewhat arbitrary.

It is noted that in complex river topographies (sinuous reaches or those with islands, peninsulas etc), the algorithm may fail to correctly place cross sections. If this is the case it may be necessary to manually divide the problematic reach into several sections, creating a StreamTubes representation for each, and modelling each separately.

The X-Sec > Save endpoints and X-Sec > Read endpoints selections allow for a cross-section or series of cross-sections to be saved and subsequently read back into River2D.

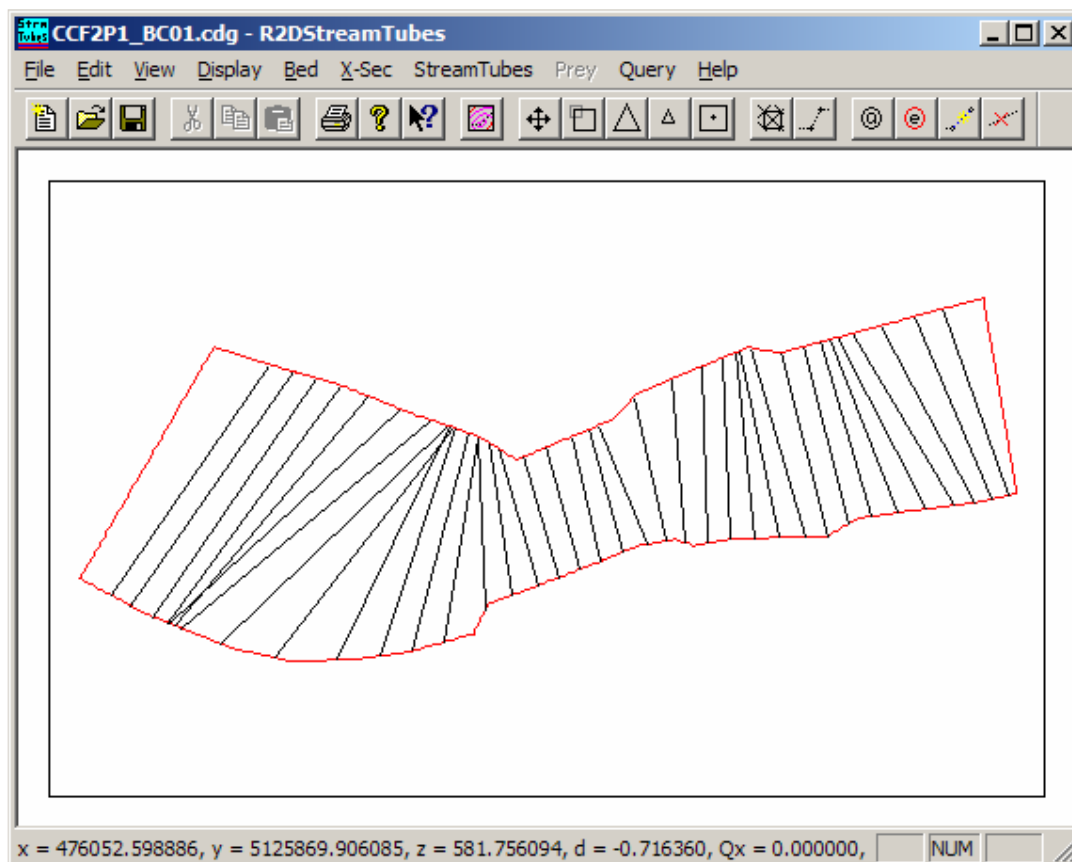


Figure 4 R2DStreamTubes.exe interface window with boundary and cross-sections displayed.

3.4 Creating streamtubes

The next step is to divide the cross-sections into a number of streamtubes, each conveying equal discharge. This is done using the **FlowTubes > Draw...** menu item. At this stage a dialog box asks you to specify the number of streamtubes you require across the stream width (Figure 5). These streamtube divisions are subsequently drawn into the display window (Figure 6). The number of streamtube divisions required in the vertical axis is specified later, along with the velocity / depth profile, during the process of saving the StreamTubes output.

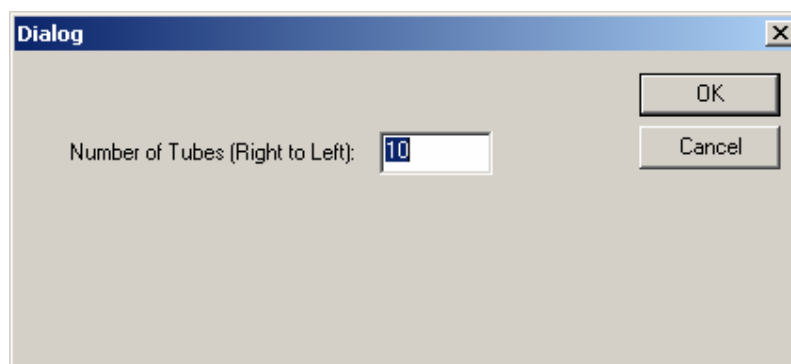


Figure 5 Dialog box for specifying the number of horizontal streamtube divisions.

The nodes defining the horizontal division of the flow can be edited interactively, to eliminate unrealistic acute angles from the streamtubes (e.g. see Figure 6), by selecting FlowTubes > EditNode and then selecting and dragging a node with the mouse. This should not be undertaken without careful consideration, as moving a node can cause unrealistically high or low velocities by changing the cross-sectional areas of streamtubes. Trying to edit nodes when there are no streamtubes will generally crash the program.

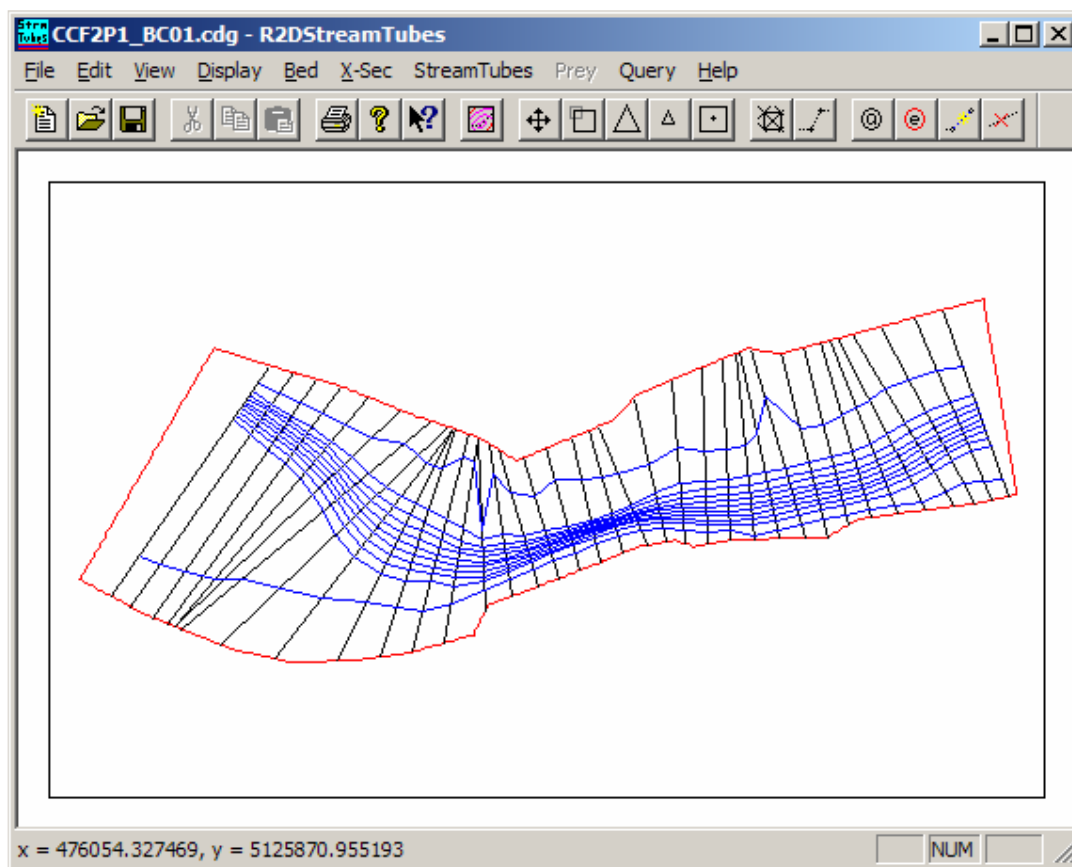


Figure 6 R2DStreamTubes.exe interface window with boundary, cross-sections and horizontal streamtubes displayed in plan view.

3.5 Subdividing vertically, and saving streamtubes

Select X-Sec > Save All as..., to save your StreamTubes in a format suitable for input to the drift model. At this point a dialog box will prompt you for the number of vertical streamtube divisions required (Figure 7). It also allows you to check a box if you want to use a constant velocity / depth profile (no change in velocity between the surface and the stream bed) to extrapolate velocities through the water column. Otherwise a logarithmic velocity profile (Gordon *et al.* 1992), with slower flow near the stream bed, will be used as the default.

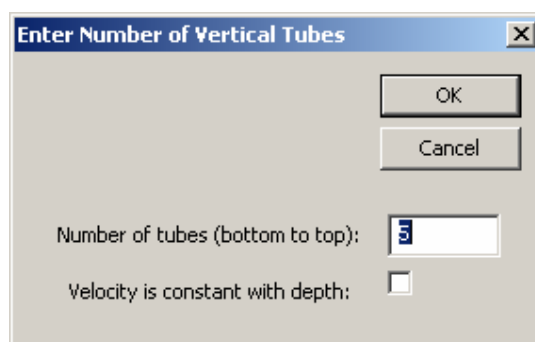


Figure 7 Dialog box with prompt for number of vertical streamtubes, and tick box to specify constant velocity, rather than default logarithmic velocity, profile with depth.

Saving the StreamTubes output creates a lightly-annotated ASCII text file (Figure 8). The filename used will need to be specified in the configuration of the drift and foraging components (described below). The format is as follows:

- number of cross-sections
- horizontal number of streamtubes
- vertical number of streamtubes

Then the following repeated for each cross section:

- cross-section number and discharge at that cross-section (this discharge will be constant)
- bed roughness height at each horizontal division across the section
- a numbered set of many nodes (X Y Z (depth below surface)) defining the corner coordinates of all streamtube polygons at that cross-section
- the node indices describing all the streamtube polygons of that cross section

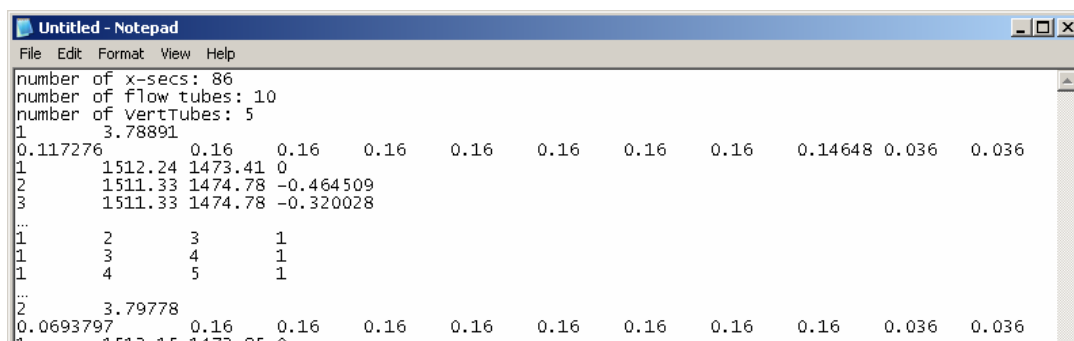


Figure 8 Example of a StreamTubes output file. See text for explanation of values.

There are memory leaks in the StreamTubes component of River2D, so do not just leave it running if you need to convert many study reaches to StreamTubes. It is best to close the program and reopen it again after each file is processed.

3.6 Querying data at points or cross-sections

The flow data can be queried at cross-sections or points using the Query menu. Use Query > Read Query XY File, to read in a text file containing the *X* and *Y* coordinates of interest in two tab separated columns. Then select Query > Process to display the results of a query of these points. The query results window will show the *X*, *Y* and *Z* coordinates of your query points, the depth (*d*), and the flow in the *X* and *Y* directions (*Q_x* and *Q_y*, respectively). You can save the results of the query by clicking the Save Results As... button in the query results window. This function can be helpful for comparing the flow solution's values with observed velocity values.

4 INVERTEBRATE DRIFT MODEL

4.1 Purpose of the model

The invertebrate drift model uses the spatial flow description (provided by the StreamTubes model), and data on initial drift concentrations and behavioural characteristics for each size class of each taxon (provided by the user), to estimate spatially explicit drift-density within the reach (Figure 10).

4.2 Mechanics

The model calculates the dispersion of drift between streamtubes in both the vertical and horizontal directions, working downstream with flow from cross-section to cross-section (Figures. 9 & 10) (See Appendix 1 for a more detailed description of the drift transport model, including supporting equations). Modelling of invertebrate drift dispersion is based on Rutherford's (1994) river mixing equations and estimates of taxon and size-specific entry rates ($\text{No.} / \text{m}^2 / \text{s}$) and settling velocities (m / s). Rutherford's turbulent mixing equations are rewritten as variations on the equation describing the Gaussian distribution, where the standard deviation of lateral or vertical dispersion is predicted from water depth, water velocity, bed roughness, and the distance between the cross-sections. By integrating this equation twice we derived an equation to predict the drift contribution that a given streamtube makes to each adjacent streamtube. After calculating the effects of dispersion on drift density between two adjacent cross-sections the model estimates the effects of settling and entry on invertebrate drift concentration.

This invertebrate drift transport model allows a significant advance in the way that spatial variation in invertebrate drift density can be dealt with in bioenergetically-oriented models of stream fish habitat. Until now these models have generally assumed uniform drift density, even though it is known that this assumption is unrealistic (Stark *et al.* 2002; Hughes *et al.* unpublished manuscript). There simply has not been an adequate way to predict spatial variability of drift. Our model overcomes this obstacle. It predicts how variation in bed topography, discharge, invertebrate abundance and behaviour will affect drift density in a spatially explicit manner. The model's predictions of drift density have been shown to match well with measured invertebrate drift distribution (Hayes *et al.* 2003, 2007).

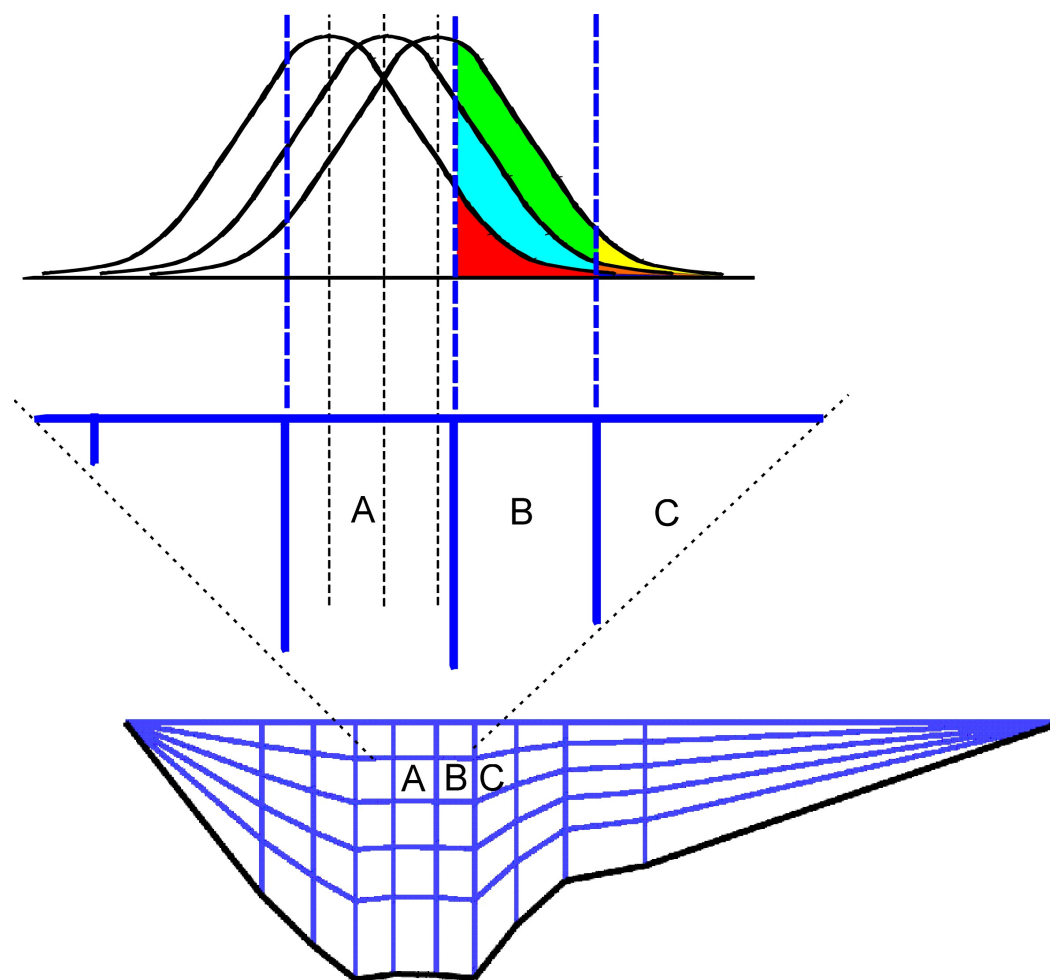


Figure 9 The lateral exchange of drift between streamtube cells in a cross-section, based on a series of Gaussian distributions. Each different colour-shaded area denotes the proportion of drift from a lateral position in source streamtube A exchanged into destination streamtubes B and C. The total proportion of drift exchanged from streamtube A with streamtubes to the left and right is based on the average overlap of the destination streamtube with the Gaussian distributions centred on all possible lateral positions within column A [see Appendix 1].

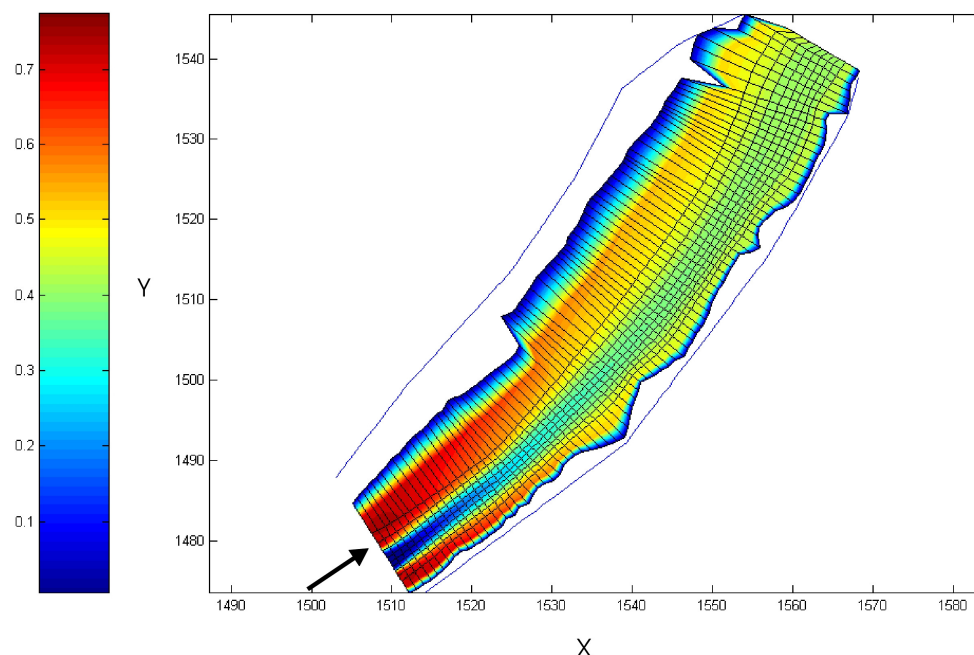


Figure 10 A plan view of modelled invertebrate drift dispersed down through a river reach. For this illustration the centre four streamtubes were given an initial concentration at the top of the reach of zero, so the invertebrate density in these streamtubes lower in the reach has dispersed in from streamtubes on either side. The settling rate was set to zero in this model run, and the arrow indicates the direction of flow.

4.3 Drift model data requirements

The drift modelling takes place in a drift model executable, which may be interface-driven, or command line driven. Note also that there are customisations of this program, to handle specific situations, such as StreamTubes files generated by R2DStreamTubes or by RhyHabSim. This program requires three input files: StreamTubes, invertebrate parameters and initial invertebrate densities. The filenames of these files can be specified when calling the program from the command line or a batch file. e.g.:

```
drift.exe -d "." -o xsIn.txt flowQ10.txt invertParams.txt
initialConc.txt
```

or specified interactively through the graphical user interface (GUI) if using that implementation of the program.

1. **StreamTubes:** a text file generated by R2DStreamTubes, as described above (see Figure 8).
2. **Invertebrate parameters:** a text file providing the following information on the invertebrates in the drift (Figure 11).
 - number of length-class * taxon classes of drifting invertebrates
 Then for each drift class:

- a name for the length / taxon class
- entry rate into drift from the bed (No. / m² / s) for the class
- settling velocity (positive if sinking, negative if rising) (m / s)
- length of invertebrate (m)
- dry weight of invertebrate (g)
- time spent near the bed following a near bed release (s)

The “time near bed” parameter in the invertebrate parameters file is included to control the area of the upstream footprint from which invertebrates entering the drift are considered in the calculations at a given cross-section. This helps alleviate the potential for inflated estimates of drift density, which would result if the entry rate (No. / m² / s) were applied to the entire area between cross-sections in slow moving water.

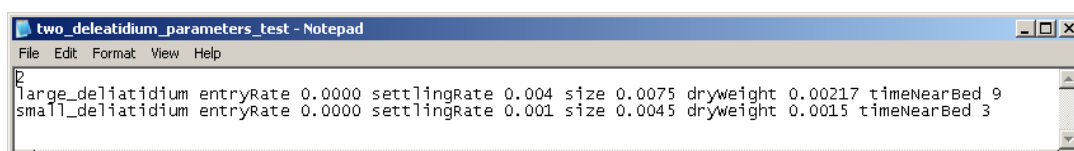


Figure 11 An invertebrate parameters text file

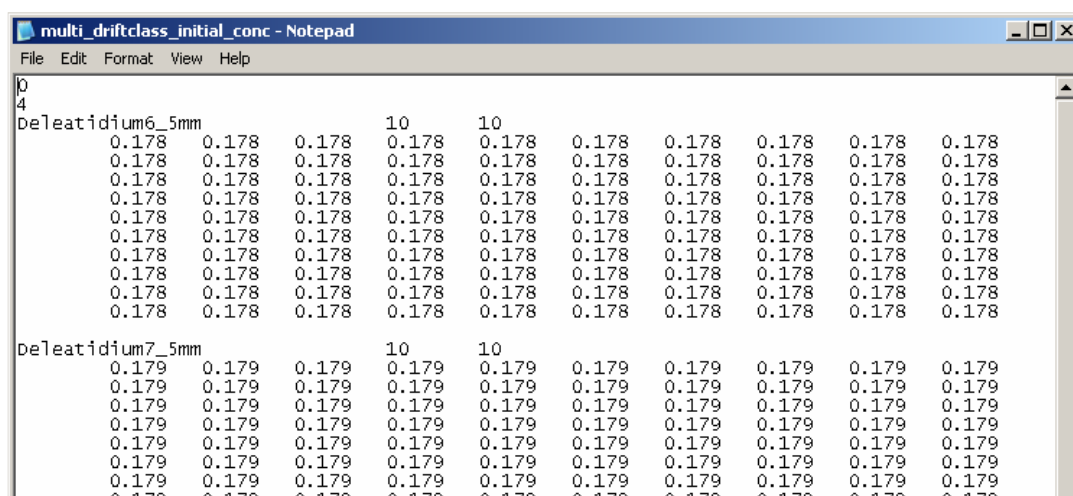
3. **Initial drift densities:** a text file providing initial drift densities (No. / m³) for each drift class at the top cross-section to be modelled for each streamtube (Figure 12). The file contains:

- the cross-section that the drift model is to be initialized at (0 being the first (upstream) cross-section in the reach)
- number of drift classes

Then the following repeated for each drift class:

- name of drift class, and horizontal and vertical number of streamtubes,
- a matrix of the initial drift densities,

Note that this file must include an individual matrix of initial drift concentrations for each class of drift. These initial concentrations should preferably be derived from the results of drift sampling in the field.



```

multi_driftclass_initial_conc - Notepad
File Edit Format View Help

10
4
Deleatidium6_5mm      10      10
0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178
0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178
0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178
0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178
0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178
0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178
0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178
0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178
0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178
0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178
Deleatidium7_5mm      10      10
0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179
0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179
0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179
0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179
0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179
0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179
0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179
0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179
0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179
0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179 0.179

```

Figure 12 A drift density initialization text file to initialise the first cross-section with concentrations for 4 drift types.

Note: This program does not model drift depletion by drift-feeding fish. However, it is possible to model this in combination with the foraging model (See section 6 Modelling carrying capacity).

4.4 Modelling invertebrate drift

Start by opening the drift modelling program, DriftModel.exe, by double clicking on its icon. The initial graphic in the display just lets you know the program is functioning. Once you open a streamtubes reach representation, the display will show the cross-sections in plan view, and coordinates will display in the status line as the mouse is moved over the display area.

The input files must be loaded in the correct order, moving from left to right across the menu bar (Figure 13). Start by reading in the streamtubes file that you created using StreamTubes, using the CrossSections > Read in menu item. Next, use Drift > Read in to load an invertebrate parameters text file. Finally, load the initial drift densities using DriftModel > Initialize command.

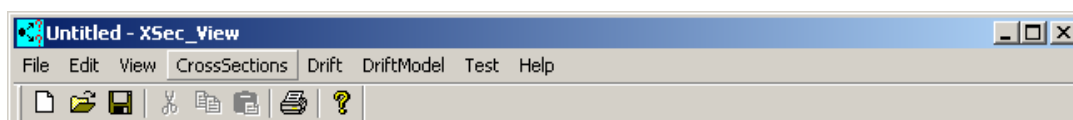


Figure 13 The DriftModel.exe menu bar

The drift transport model can now be run by selecting DriftModel > Disperse.

Note: You can load new invertebrate drift data (using the CrossSections > Read in menu item and Drift > Read in) and re-run the model (using DriftModel > Disperse) without starting over.

4.5 Saving the drift model output

There are several potential outputs from the drift model. The first two ways of extracting drift data are most frequently useful:

1. CrossSections > Concentrations Dump creates a lightly annotated text file (Figure 14). For each cross-section by drift class combination there are some meta-data (cross-section number, name of drift class, number of tubes vertically and horizontally, and the address in computer memory where the data structure was stored) followed by a tab delimited matrix of predicted drift concentrations (No. / m³), for each polygon of each cross-section.

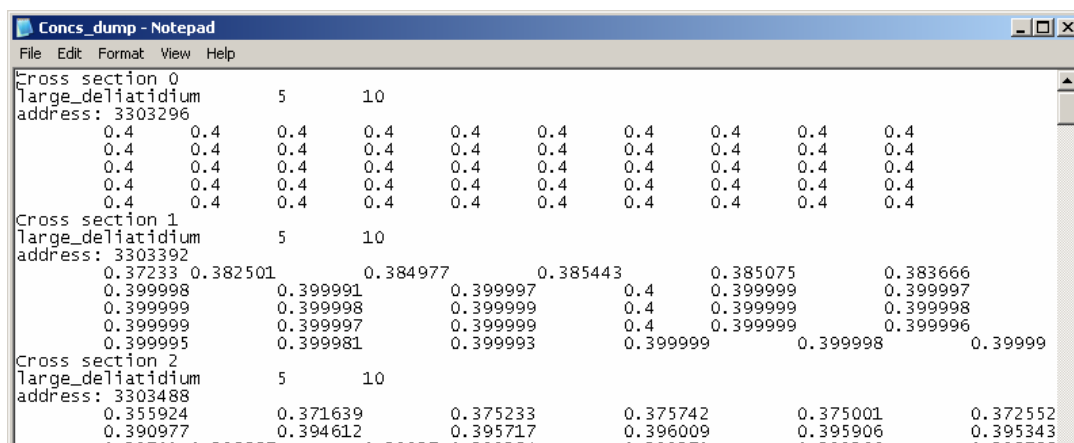


Figure 14 An output file generated by CrossSections > ConcentrationsDump from the drift model

2. CrossSections > Dump for Foraging Model outputs a non-annotated text file (Figure 15), which is suitable for input to the foraging model. The file contains the following:

- the cross-section at which the drift model run was initialized
- the total number of cross-sections

Then the following is repeated for each cross-section:

- the horizontal and vertical number of streamtubes
- X and Y coordinates of the first point on the cross-section (in plan view) and the X,Y of the last point on the cross-section
- distances to consecutive horizontal streamtube divisions across the cross-section
- a matrix of depth coordinates of the vertical divisions at each horizontal division
- a matrix of streamtube water velocities
- the number of drift classes, and the name, length and weight for each class of drifting invertebrates from the invertebrate parameters file

For each invertebrate drift class:

- a matrix of the density in each polygon in that cross section

```

xsIn - Notepad
File Edit Format View Help

0
86

10 10
1512.53 1473.64      1504.71 1485.38
0 1.64469 2.46566 3.19512 3.87187 4.57912 5.37234 6.4762 7.66603 9.23861 14.1061
0 0 0 0 0 0 0 0 0 0
0 -0.0263696 -0.0346481 -0.0420414 -0.0411084 -0.0414731 -0.03793 -0.0284187 -0.0281482 -C
0 -0.0648438 -0.0838683 -0.10076 -0.0986327 -0.0994645 -0.0913765 -0.0695656 -0.0687537 -C
0 -0.104185 -0.134157 -0.160721 -0.157377 -0.158684 -0.145969 -0.11163 -0.11026 -0.0869275
0 -0.144559 -0.185717 -0.222158 -0.217573 -0.219365 -0.201925 -0.154788 -0.15284 -0.119453
0 -0.186199 -0.238835 -0.285401 -0.279544 -0.281834 -0.25955 -0.199285 -0.196734 -0.152822
0 -0.229452 -0.293933 -0.350941 -0.343772 -0.346575 -0.319297 -0.245488 -0.242303 -0.18725
0 -0.274883 -0.351701 -0.419571 -0.411038 -0.414375 -0.381902 -0.293993 -0.29013 -0.223124
0 -0.323539 -0.413418 -0.492766 -0.482792 -0.486692 -0.448733 -0.345906 -0.341301 -0.26108
0 -0.377958 -0.482172 -0.57408 -0.562531 -0.567047 -0.523086 -0.403905 -0.39844 -0.302725
0 -0.469368 -0.593429 -0.702547 -0.688849 -0.694206 -0.642034 -0.500295 -0.492913 -0.36276

1.75316 1.51783 1.35918 1.35119 1.30183 1.20719 1.03816 1.12969 0.929884 0.655026
1.20159 1.05611 0.965683 0.966518 0.930671 0.860165 0.728174 0.781669 0.6729 0.499903
1.17511 1.0333 0.945442 0.946468 0.911357 0.842222 0.712629 0.764662 0.659272 0.490724
1.14505 1.0074 0.922456 0.923687 0.889393 0.821823 0.694962 0.745332 0.643762 0.480263
1.11023 0.977379 0.895787 0.89727 0.863927 0.798166 0.674492 0.722962 0.625785 0.468123
1.06883 0.941673 0.864029 0.865782 0.833589 0.769994 0.650123 0.696327 0.60433 0.453605
1.01759 0.897436 0.824655 0.826741 0.79596 0.735056 0.619931 0.663365 0.577714 0.435545
0.950143 0.839105 0.772613 0.775106 0.746209 0.688871 0.580076 0.619915 0.542467 0.411518
0.849522 0.751906 0.694583 0.697602 0.671525 0.619586 0.520434 0.555015 0.489467 0.375122
0.505745 0.456979 0.434811 0.440963 0.42413 0.389485 0.319871 0.334812 0.312929 0.260195

4
Deleatidium6_5mm 0.0065 0.0014
Deleatidium7_5mm 0.0075 0.00217
Deleatidium8_5mm 0.0085 0.00319
Deleatidium9_5mm 0.0095 0.00448

0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178
0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178
0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178
0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178 0.178

```

Figure 15 An output file generated by CrossSections > Dump for Foraging Model from the drift model

This file format has sufficient data to reconstruct the volumetric invertebrate density distribution for graphical display, further investigation, evaluation, validation etc. Note that the coordinates provided in the file are for the corners of cells, while the invertebrate densities specified are for the centres of the cells thus defined. Some external manipulation will be required.

A third option for data extraction is:

3. DriftModel > Query... This allows the output to be queried at user-defined points. It requires an input text file with columns defining the X and Y coordinates and the depth (negative Z) values of interest. The output must be saved as a new file. It takes the form of a tab delimited text file, where the first row gives column names (x, y, depth, drift-class name) and the following rows are the data (*i.e.*, drift concentrations (No. / m³)).

This query works by finding the cross-sections upstream and downstream of each point, the column containing the specified X and Y, and the row containing the specified depth. The concentrations are bi-linearly interpolated between the upstream and downstream cross-sections.

The final method of data extraction is:

4. CrossSections > Dump will create a text file describing the physical characteristics of the streamtubes, broken down by column (horizontal divisions) and row (vertical divisions), for each cross-section (Figure 16). This dump exists for troubleshooting internal calculations in extensive detail, and will be of little or no interest to the day-to-day user.

- column width: the average width of the column (m)
- column roughness: the effective roughness height of the bed beneath that column
- right bottom and left bottom: the XYZ coordinates of the bottom corner nodes for each column
- areas: the surface areas of the polygons defining each streamtube in the given row, across that cross-section (m^2) (*i.e.*, viewed looking through the cross-section from either upstream or downstream)
- velocities: the average velocities (m / s) in each streamtube in the given row, for each column across the cross-section
- centroids: the XYZ coordinates of the centroids of each streamtube polygon in the given row and column
- tubeHeights: the average heights (m) of each streamtube in each column
- total depth through centroid: the depth (m) of water, from the surface to the bed, measured on a line through the centroids of the given streamtube polygons

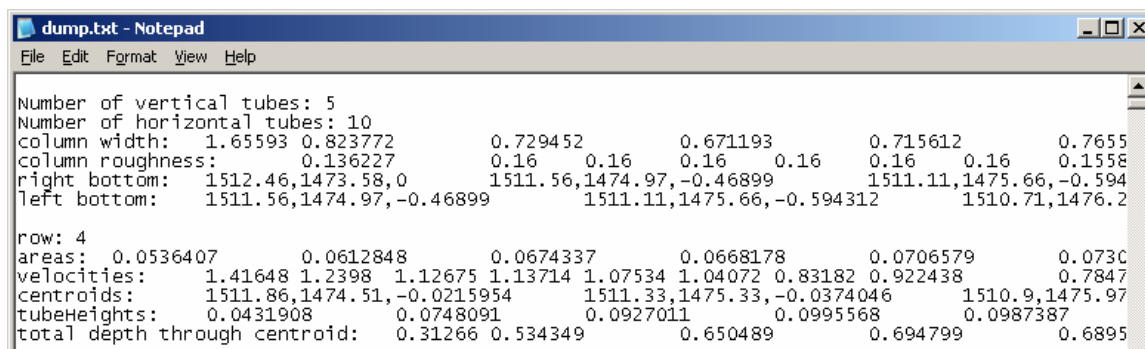


Figure 16 A cross-section dump from the drift model

4.6 Altering the model parameters

It is possible to alter the relative impact of the various parameters driving the distribution of drift within the model. This function allows for sensitivity testing of the model. DriftModel > Set H Params brings up a dialog box (Figure 17), where you can specify multipliers for the degree of dispersion in the horizontal and vertical directions, as well as for the settling velocity and entry rate of invertebrates.

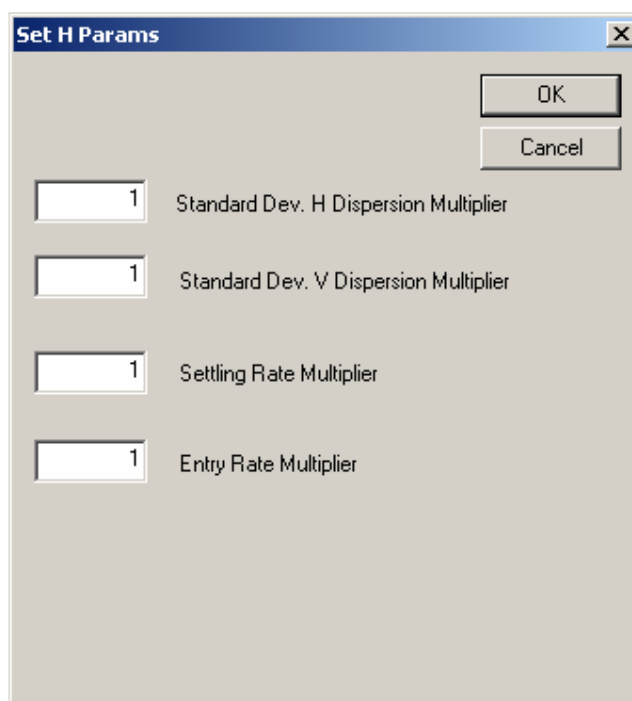


Figure 17 The Set H Params dialog box

4.7 Command line control

An alternative to the graphical interface is to control the drift model from the command line or batch script. This enables the automated repetition of model runs, altering input files or control parameters to build a set for comparisons, search for optima, investigate trends etc.

Help can be invoked for the drift model from the command line. Noting that customised versions of the drift model may have names differing from the default “drift.exe”, help is invoked by passing the “--help” flag to the drift.exe being used. For example

```
drift.exe --help
```

gives the output:

```
Usage: drift.exe [-d baseDir] [-o <output>] [--hmult=<double>]
[--vmult=<double>] [--smult=<double>] [--emult=<double>] [--help]
[--version] <streamFile> <paramsFile> <concFile> [-q
<queryInput>]
This program estimates drift concentrations based on three input
files specifying flow, characteristics of drifting insects, and
initial concentrations of drifting insects.
-d, --directory=baseDir base directory for input files
-o <output> output file (default is "-")
--hmult=<double> std dev horizontal multiplier
--vmult=<double> std dev vertical multiplier
--smult=<double> settling rate multiplier
--emult=<double> entry rate multiplier
--help print this help and exit
--version print version information and exit
<streamFile> <stream tube Input file>
<paramsFile> <drift param Input file>
```

```
<concFile>          <initial concentrations Input file>
-q <queryInput>      query input file
```

This help output details how all the customisations that can be specified through the graphical interface described above, are specified via the command line. A command is built up from the name of the executable, appended with a number of flags, some optional, to specify input and output files, and parameterisation such as the multipliers.

```
drift.exe -d "." -o xsIn.txt flowQ10.txt invertParams.txt
initialConc.txt
```

For iterative execution as implemented in a script (R, Matlab etc), a valid and complete command line call is saved as a batch file, and that batch file then called from the script. The script is responsible for the file management that ensures the correct input files are in the directory the model is run in, and the output files are moved away for archive and further analysis.

5 SALMONID FORAGING MODEL

5.1 Purpose of the model

The foraging model estimates the net energy intake of a drift feeding fish from information on water depth, water velocity, the body size and density of drifting invertebrates, the body size of the fish, and the temperature of the water. It predicts the net rate of energy intake (NREI) as the gross rate of energy intake (GREI) less the energy costs of steady swimming to maintain station at the focal point.

The foraging model is based on the model described by Hughes *et al.* (2003). It takes into account the size and velocity of the prey, and the reaction distance of the fish to prey to calculate the capture rate (Figures. 18 & 19). Hughes *et al.* estimated the foraging area by predicting water velocity dependent foraging radii over a uniform square computational grid (Figure 19). We modified this model, substituting the uniform square grid with the StreamTubes representation, of cross sections composed of triangles and quadrilaterals (*c.f.* Figure 2).

Water temperature, swimming speed (*i.e.* water velocity at the focal point) and fish size are taken into account in the calculation of swimming costs. Several options are provided for estimating swimming costs (see Appendix 2).

Hughes *et al.* (2003) used 3D underwater videography to test the drift foraging model on large brown trout in the Travers River, New Zealand. They found that the model made good predictions of the foraging area but that it overestimated prey capture rate by about 45 %, presumably because the fish did not capture every prey item that entered their foraging areas. An option is provided to correct for this over estimation (See section 5.2 Modelling NREI).

When the foraging model is run the user specifies the number of points on each cross-section at which the model will predict the fish's NREI. Interrogation points are then spaced evenly across each cross-section, and combine to provide a two-dimensional representation of predicted NREI (*i.e.*, a 2D plan view of the reach).

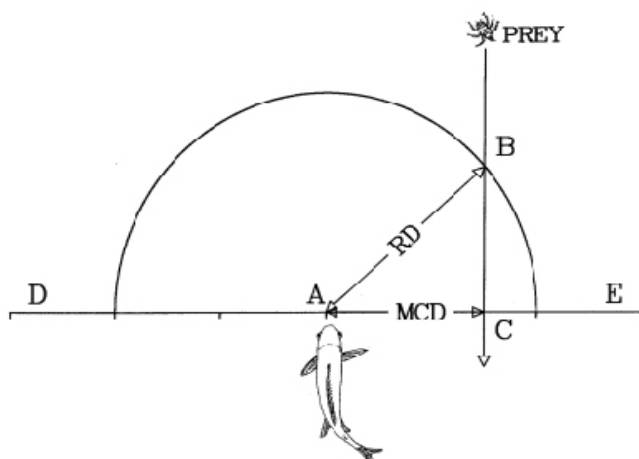


Figure 18 Plan view of the foraging model showing the geometry of prey interception.

The fish is assumed to detect prey as they hit the surface of the hemispherical reaction volume with a radius equal to its reaction distance (RD) to the length class of the prey in question. The fish intercepts prey at its maximum sustainable swimming speed (V_{max}) and may only capture those prey it is able to intercept before they cross the line D – E. Under these conditions, when water velocity is V , the maximum lateral capture distance

(MCD) is:

$$MCD = \sqrt{RD^2 - \left(\frac{V \cdot RD}{V_{max}} \right)^2} \quad (\text{Hughes et al. 2003}).$$

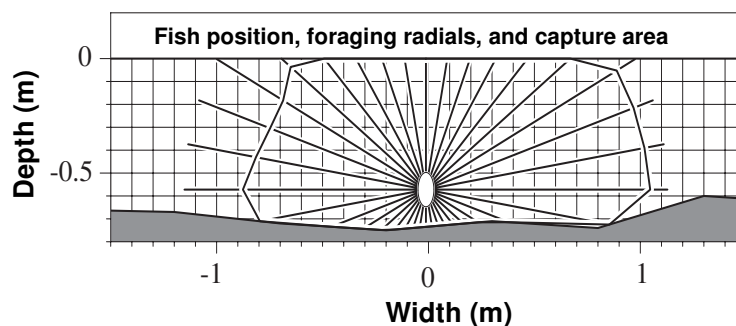


Figure 19 Cross-sectional view of the geometry used by the foraging model showing: fish position; foraging radials; and predicted prey capture area interpolated from predicted foraging radius along each foraging radial (based on prey reaction distance and water velocity) (Hughes *et al.* 2003)

5.2 Modelling NREI

NREI modelling is undertaken by the command line program ForagingModel.exe. This model requires two input files:

1. A cross-section file, which is the output from the drift model (Figure 15). This must be named xsIn.txt.
2. Fish characteristics: a text file named paramsIn.txt, similar to the invertebrate parameters input to the drift model (Figure 20). This file must contain, in order:
 - fork length of the fish (m)
 - fish weight (g)
 - minimum prey size (m) [Hayes *et al.* (2000) and Hughes *et al.* (2003) estimated this as 1.15 % of the fish length – based on gill raker spacing after Wankowski (1979)]
 - maximum prey size (m), [typically based on the gape size; 45.2 % of the fish length, *e.g.*, Hayes *et al.* (2000) and Hughes *et al.* (2003) – after Wankowski (1979)]
 - probability of prey detection (allows correction for overestimation of prey capture rate in the model)
 - height of focal point above bottom (m)
 - number of interrogations per cross-section
 - water temperature (°C)
 - defense radius (m)

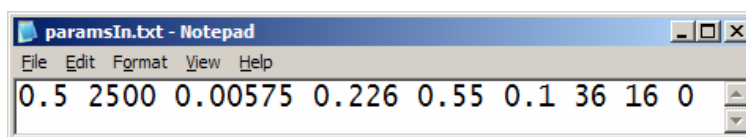


Figure 20 A paramsIn.txt file, specifying fish and habitat parameters.

Both of these files must be appropriately named and located in the same directory folder as a copy of the foraging model, ForagingModel.exe.

Double clicking the foraging model icon will open the command line window and automatically start the model running. The program will automatically create several output files in the same folder in which it has been activated.

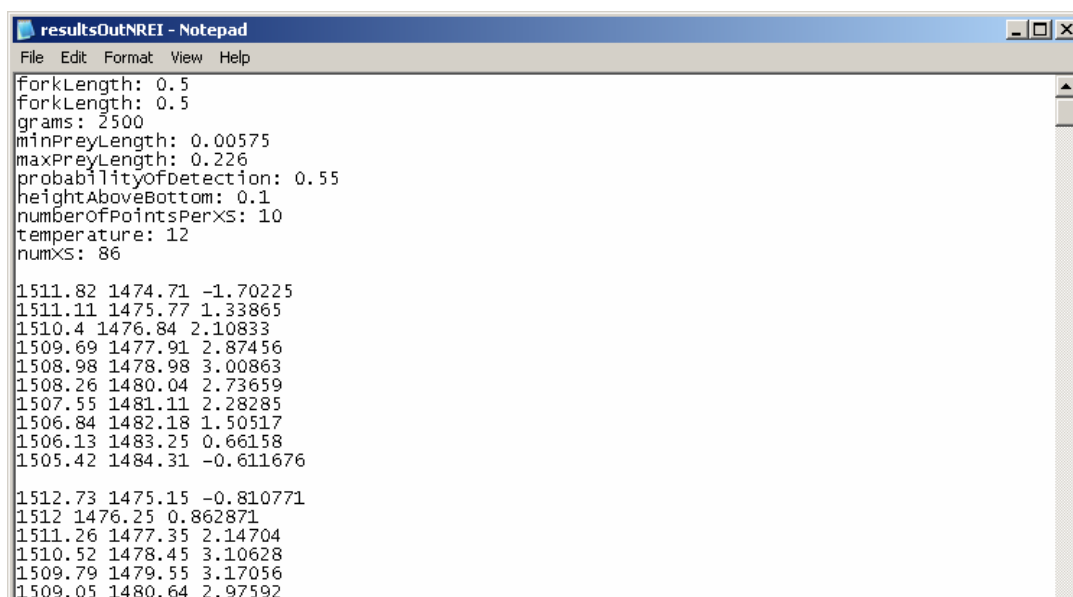
Alternatively, the executable program can be launched from a command line window where the directory has been changed to the folder containing the input files of interest. The output files will then be written to the same directory.

Alternatively, the program may be launched by a script running in an environment such as R, or Matlab, etc.

5.3 Foraging model outputs

The output named xsOut.txt is simply a duplicate of the input cross-section file (xsIn.txt) and is solely to verify that the data were properly read in. This is useful given the inconsistency between different streamtube generating programs as to which bank a cross section is initiated on. Aside from this there are six output files:

1. resultsOutNREI.txt lists the X and Y coordinates and the NREI output for each sampled position in a tab delimited text file (Figure 21).



```

resultsOutNREI - Notepad
File Edit Format View Help
forkLength: 0.5
forkLength: 0.5
grams: 2500
minPreyLength: 0.00575
maxPreyLength: 0.226
probabilityOfDetection: 0.55
heightAboveBottom: 0.1
numberOfPointsPerXS: 10
temperature: 12
numXS: 86

1511.82 1474.71 -1.70225
1511.11 1475.77 1.33865
1510.4 1476.84 2.10833
1509.69 1477.91 2.87456
1508.98 1478.98 3.00863
1508.26 1480.04 2.73659
1507.55 1481.11 2.28285
1506.84 1482.18 1.50517
1506.13 1483.25 0.66158
1505.42 1484.31 -0.611676

1512.73 1475.15 -0.810771
1512 1476.25 0.862871
1511.26 1477.35 2.14704
1510.52 1478.45 3.10628
1509.79 1479.55 3.17056
1509.05 1480.64 2.97592

```

Figure 21 A resultsOutNREI.txt output from the foraging model

2. resultsOutEFF.txt contains the same X Y coordinate data in the same format as resultsOutNREI.txt, and a measure of foraging efficiency in place of NREI. Foraging efficiency is calculated as GREI / swimming costs.
3. resultsOutCosts.txt contains the same X Y coordinate data in the same format as resultsOutNREI.txt, and the swimming costs calculated for each of these points.
4. resultsOutMaxNREI.txt presents data on the best foraging spot found in the modelled reach (Figure 22). It provides the X and Y coordinates of the best location, the cross-section in which it is located, and the NREI at this point. It also describes the drift density that remains in each of the streamtubes at this cross-section after a fish located at the best point has fed (*i.e.*, it reflects drift depleted by fish foraging at that optimum point). It provides this information in essentially the same format as the “Initial drift densities” file input to the drift model (see section 4.2 Drift model data requirements), only the initialisation cross-section is not given.

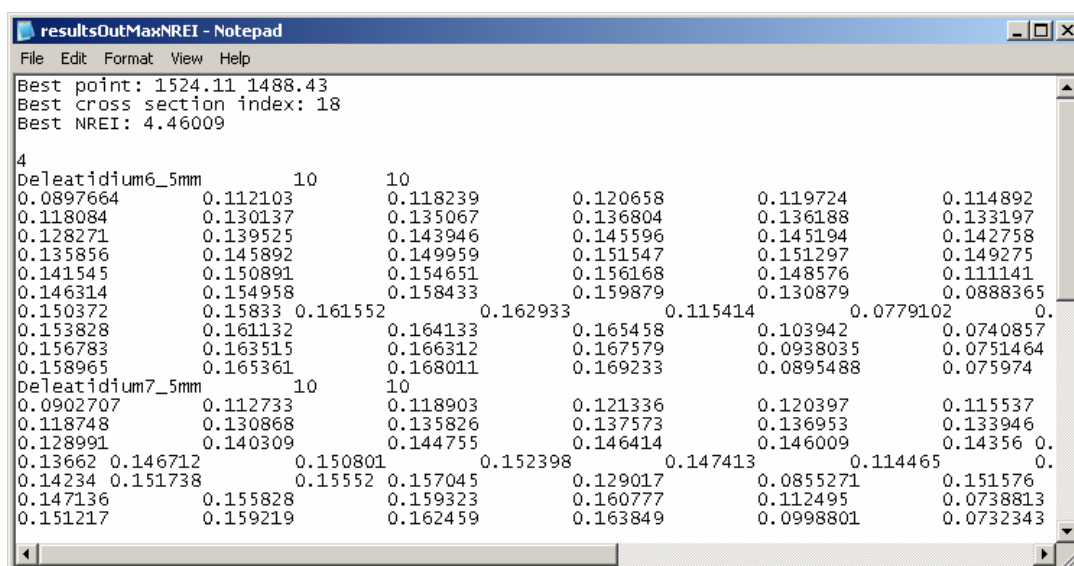


Figure 22 Example of resultsOutMaxNREI.txt output from the foraging model

- epsOut.eps is an EPS format file (Figure 23) that shows a cross-sectional view of the best foraging location. It shows the water surface, horizontal and vertical streamtube divisions, and foraging area radials.

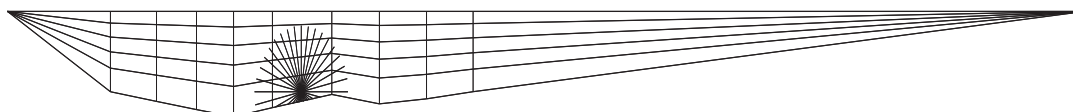


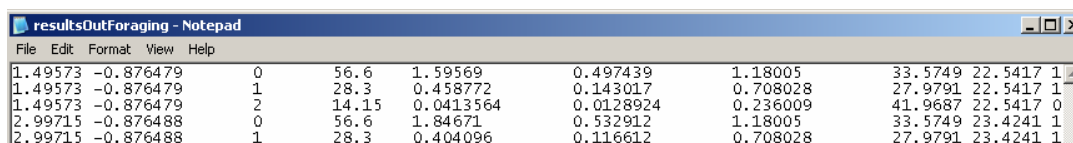
Figure 23 Example of the epsOut.eps output from the foraging model

Note that changes in the eps file format in different versions of the Windows operating system means that as of 2012 this output does not render correctly.

5. resultsOutForaging.txt provides information on the model's predictions of the size-composition of the diet, prey choice etc. (Figure 24). For every focal point by drift class combination it contains:

- the X and Y coordinates of the fish's focal point
- the drift class index
- the total energy content of the drift class (J)
- the fish's encounter rate with the given drift class per second of search (No. / s)
- the capture rate of the drift class per second of foraging (No. / s)
- the handling (interception) time (s) for the drift class (this is reaction distance/maximum swimming velocity)
- the profitability (J / s) of drift class (0.7 (total prey energy/handling time)) [N.B. 0.7 conversion factor after Elliott (1982)]
- the fish's gross energy intake at this position (J / s)

- the in-diet index (1 if the fish should eat the prey, 0 if not) – determined by the minimum and maximum prey size or by Charnov's (1976) model.



File	Edit	Format	View	Help						
1.49573	-0.876479	0	56.6	1.59569	0.497439	1.18005	33.5749	22.5417	1	
1.49573	-0.876479	1	28.3	0.458772	0.143017	0.708028	27.9791	22.5417	1	
1.49573	-0.876479	2	14.15	0.0413564	0.0128924	0.236009	41.9687	22.5417	0	
2.99715	-0.876488	0	56.6	1.84671	0.532912	1.18005	33.5749	23.4241	1	
2.99715	-0.876488	1	28.3	0.404096	0.116612	0.708028	27.9791	23.4241	1	

Figure 24 A resultsOutForaging.txt output of the foraging model

In this case (Figure 24) prey type 2 has been dropped from the diet due to morphological constraints. You can't tell from the information provided here but this prey is only 0.0015 m long, and a 0.003 m minimum prey size was used in the simulation. This prey type would not have been dropped from the diet for energetic reasons based on Charnov's (1976) prey model because the profitability of the drift class is greater than the fish's gross rate of energy intake at this position (*i.e.*, 41.9687 > 22.5417) – and so including it in the diet would increase the fish's net rate of energy intake.

6 MODELLING CARRYING CAPACITY

The drift and foraging tools can be used to investigate the number of fish supported at a certain NREI in the study reach. Two approaches are described here, one manual and the other more automated. The manual description describes an approach more suitable to shorter reaches, or reaches where the fish in the dominant position would defend all positions upstream to ensure its drift supply was undepleted. This implies that once the first fish has been placed (at the optimum position in the whole reach), subsequent fish will only be placed downstream of this. This assumption may be appropriate for modelling positions of large fish in a short pool but clearly it is not appropriate for a long modelling reach. The process involves identifying the cell in the modelled reach, when empty of fish, with the highest NREI value. A fish is placed there, depleted drift is propagated downstream, NREI recalculated downstream, and the next highest NREI cell is identified in that downstream section – and so on.

The automated approach described below allows fish to be placed upstream of the overall optimum position in the reach. This is suitable for larger reaches, where positions exist sufficiently far upstream from the optimal location that a dominant fish would not defend those positions if other fish were to occupy them. Such a scenario will occur with long modelled reaches and where the distribution of multiple size classes of fish is modelled. In the latter case, large fish will not compete with small fish because they feed on different sized prey. The automated approach is also suited to dealing with the complexities that arise when upstream fish, beyond the range of perception of the dominant fish, would deplete its drift supply sufficiently to displace it. That is, the optimal foraging position in a river empty of fish, may become non-profitable due to the presence of upstream fish in positions that are profitable, but less so than the empty-river optimum.

6.1 Manual approach

The optimal locations (and ultimately the carrying capacity) of fish within the reach can be modelled by an iterative process using the maximum NREI output from the foraging

model (resultsOutMaxNREI.txt). This process can also provide insight into the effect of depletion by feeding fish on the density of invertebrate drift downstream.

The foraging model output resultsOutMaxNREI.txt contains the best fish location in the modelled reach, as defined by maximum NREI. It also provides a matrix of the drift concentrations remaining in each streamtube at that cross-section, after a fish at that location has fed.

To model the impact of this depletion on the drift dispersion it is possible to rerun the drift model, using the same StreamTubes model and invertebrate parameters as the initial run, but using an adapted version of the resultsOutMaxNREI.txt to initialize the model run at a new cross-section.

To do this, the drift model program (DriftModel.exe) must be kept open following its initial run. This is so that it can keep track of the drift dispersion upstream of the point at which it is initialized in subsequent runs.

A new initial drift concentration file can be created for the truncated model run by copying and pasting the pertinent parts of resultsOutMaxNREI.txt into a new text file (Figure 12). This file can be created by deleting the first three lines from a copy of the resultsOutMaxNREI.txt file (except for the cross-section number). The edited file will contain:

- the number of the cross-section at which the new run is to be initialized
- the number of drift classes
- matrices of drift concentrations in each streamtube at that cross-section for each drift class, preceded by the drift class name and the number of horizontal and vertical streamtube divisions

Save this new text file. Then reinitialize the drift model with this new file, using DriftModel > Initialize. Leave the other parameters the same as in the initial run, and rerun the model by selecting DriftModel > Disperse.

The foraging model is run again, using the output from this truncated run of the drift model, to find the location of the next best fish location in the modelled reach, in terms of NREI. Net Rate of Energy Intake is still computed along all the cross-sections in the reach, but only points on cross-sections downstream of the most recent initialisation cross-section are ranked as potential fish locations. So, in the first instance, the first cross-section checked for the best location has index $0+1 = 1$, which is the second cross-section from the top of the modelled reach. If the drift model had most recently been initialized at cross-section 10 then the foraging model would predict NREI for all cross-sections, but only search for the best positions downstream of this cross-section (*i.e.* in cross-sections 11 and above).

In order to avoid the outputs of the original foraging model run being overwritten by the outputs of the new model run, set up a new folder in which to run the truncated foraging model run. This folder has to contain a copy of the foraging model (ForagingModel.exe), a paramsIn.txt file, and an output from the latest truncated run of the drift model (named xsIn.txt). Alternatively, simply rename the output files from the original model run to avoid them being overwritten.

Note that it is possible to alter the fish and habitat characteristics parameters specified in the paramsIn.txt file in subsequent foraging model runs. This makes it possible to model the distribution and abundance of a size-structured group of fish. In this situation the largest fish should be placed first, on the assumption that they will be able to defend the most profitable areas.

By iterating this process it is possible to predict the location of several, or all, the fish in the reach. Remember that a fish will not be able to maintain condition in a position where the predicted NREI is negative. So a negative NREI value at the location specified in a resultsOutMaxNREI.txt file could be interpreted as indicating that no more fish would be able to make a living in the remaining portion of the reach. Other, more conservative, cut-off points in terms of NREI requirements could be employed depending on the situation. For example, a minimum NREI value that would provide for maintenance of condition as well as accumulation of reserves for reproduction might be specified as the cut-off value.

Note: If the drift model must be shut down or crashes during the iterative process of modelling multiple fish locations, it is possible to recover the position you were in without rerunning the entire process. Simply reopen the drift model and initialize the model at the top cross-section as usual and disperse. Then proceed to re-initialize the model at successive cross-sections down the reach, using the drift initialization files that you had already created based on successive resultsOutMaxNREI.txt files, rerunning the drift model again between each initialization. When you reach the position you had obtained before closing the drift model program you can pick CrossSections > Dump for Foraging Model and carry on as usual.

6.2 Automated approach

The command-line interface to the drift model allows it to be controlled by scripts in external languages. In this way many repeated modelling runs can be incorporated into investigations by repeatedly modelling the same reach with different configurations such as flow, initial drift density etc. Script-based control also allows the fish placement to proceed slightly differently from that described under the default graphical interface approach. When run through the graphical interface, the first fish is placed at the location with the highest NREI in the whole reach, and subsequent fish are placed only downstream of existing fish. This can leave large empty areas of profitable habitat upstream of the dominant fish. To allow such territory to be occupied requires the change that is made possible by the automated approach, based on drift-modelling and then forage-modelling just one cross section at a time from the top to the bottom of the modelled reach.

The external script takes on responsibilities for the management of the input and output files, and involvement in the fish-placement decision. It passes two cross sections of topography and flow data to the drift model at a time, and initialises it with appropriate invertebrate density data. The script picks up the outputs from the drift model, and hands them to the forage model, inspects the output from the forage model to determine whether a fish will be placed in that cross section, before handing the next cross section pair to the drift model. At each stage a pair of cross sections is passed to the drift model, one as the source and one as the destination for drift dispersion.

A branch in the script is controlled by whether the foraging model reports NREI high enough to place a fish in the cross section just processed. The decision to place a fish is implemented in the script, to allow for easy control over the NREI threshold chosen for fish placement. If a fish is placed, then the depleted densities output from the forage model are passed to the drift model. If no fish is placed, then the densities output from the drift model (that had already been passed to the forage model) are passed back in to the drift model, for the next cross section.

Passing of input by the script to the model programs is of course implemented by saving files of the data to be passed under filenames that match those specified in a batch command that the script then calls. An example of an R script implementing this procedure is included as Appendix 4.

6.3 Defence radius

While the drift depletion halo effect should preclude the close packing of fish, a defence radius mechanism was introduced to ensure this. This requires the radius to be specified via an extra parameter in the paramsIn.txt file.

Number	Parameter	Example
1	Fork length of the fish (m)	0.5
2	Weight of the fish (g)	2500
3	Minimum prey size (m)	0.00575
4	Maximum prey size (m)	0.226
5	Probability of prey detection	0.55
6	Height of focal point above bottom (m)	0.1
7	Number of interrogations per cross section	36
8	Water temperature	16
9	Defence radius (m)	1

The input data in paramsIn.txt are stored in a single space-separated line, e.g. 0.5 2500 0.00575 0.226 0.55 0.1 36 16 1.0

Maintaining the defence radius spacing is handled in the model code rather than in the script code. If the location being evaluated is within defence radius of any other fish, then it will not be proposed as the best location on that cross section, and another will be evaluated until one is found outside the defence distance of any other fish.

This of course requires the positions of previously-placed fish to be available to the foraging model, which is ensured by the script writing these positions to a FishPosns.txt file that is then read in by the foraging model.

7 ACKNOWLEDGEMENTS

The R2DStreamTubes software is built on the River 2D Bed code generously provided by the River 2D group (Steffler, P.; Ghanem, A.; Blackburn, J. (2003) University of Alberta, Canada).

The foraging model software relies on the “Generic Polygon Clipping” code, written by Alan Murta, for area calculations and hit-testing. The GPC software is Copyright © 1997-1999, Advanced Interfaces Group, Department of Computer Science, University of

Manchester. This software is free for non-commercial use. It may be copied, modified, and redistributed provided that the copyright notices which appear within the library source files are preserved on all copies. The intellectual property rights of the algorithms used reside with the University of Manchester Advanced Interfaces Group.

The progress bar displayed during the drift model run is a freely available VC++ object from CodeProject, with no restrictions on use.

The Matlab programs for displaying model outputs were written by Ben Tuckey and Ben Knight, Cawthron Institute.

The automated fish placement mechanism was developed with partial support from Eco Logical Research Incorporated, Utah.

8 DISCLAIMER

Any software available is provided as is, and no warranty, express or implied, is made of its suitability for any particular purpose.

9 REFERENCES:

- Addley, R.C. 1993: A mechanistic approach to modelling habitat needs of drift-feeding salmonids. M.Sc. thesis, Utah Sate University, Logan.
- Annear, T. and 15 other authors 2002: Instream flows for riverine resource stewardship. Instream Flow Council, US.
- Brett, J. R.; Glass, N. R. 1973: Metabolic rates and critical swimming speeds of sockeye salmon (*Oncorhynchus nerka*) in relation to size and temperature. *Journal of the Fisheries Research Board of Canada* 30: 379-387.
- Bovee, K.D. 1982: A guide to stream habitat analysis using the instream flow incremental methodology. U.S. Fish and Wildlife Service Biological Services Program FWS/OBS-82/26, Instream flow information paper 12. 248 p.
- Charnov, E.L. 1976: Optimal foraging: attack strategy of a mantid. *American Naturalist* 110: 141-151.
- Ciborowski, J.J.H 1983: Downstream and lateral transport of nymphs of two mayfly species (Empheroptera). *Canadian Journal of Fisheries and Aquatic Sciences* 40:2025-2029.
- Ciborowski, J.J.H 1987: Dynamics of drift and microdistribution of two mayfly populations: a predictive model. *Canadian Journal of Fisheries and Aquatic Sciences* 44:832-845.
- Elliott, J.M. 1976: Energetics of feeding, metabolism and growth of brown trout (*Salmo trutta* L.) in relation to body weight, water temperature and ration size. *Journal of Animal Ecology* 45: 923-948.
- Elliott, J.M.; Davidson, W. 1975: Energy equivalents of oxygen consumption in animal energetics. *Oecologia (Berlin)* 19: 195-201.
- Elliott J.M. 1982: The effects of temperature and ration size on the growth and energetics of salmonids in captivity. *Comparative Biochemistry and Physiology* 73:81-91.
- Gordon, N.D.; McMahon, T.A.; Findlayson, B.L. 1992: *Stream Hydrology: An introduction for ecologists*. John Wiley and Sons, Chichester, U.K. 526p.

- Guensch, G.R.; Hardy T.B.; Addley R.C. 2001: Examining feeding strategies and position choice of drift-feeding salmonids using an individual-based, mechanistic foraging model. *Canadian Journal of Fisheries and Aquatic Sciences* 58:446-457.
- Hanson, P.C.; Johnson, T.B.; Schindler, D.E.; Kitchell, J.F. 1997: Fish Bioenergetics 3.0 for Windows. University of Wisconsin-Madison Center for Limnology and University of Wisconsin Sea Grant Institute.
- Hayes, J.W.; Hughes, N.F.; Kelly, L.H. 2003: Overview of a process-based model relating stream discharge to the quantity and quality of brown trout habitat. Proceedings of the International IFIM Workshop. Fort Collins, Colorado, June 2003.
- Hayes, J.W., Hughes, N.F., Kelly, L.H. 2007. Process-based modelling of invertebrate drift transport, net energy intake and reach carrying capacity for drift-feeding salmonids. *Ecological Modelling* 207: 171-188.
- Hayes, J.W.; Stark, J.D.; Shearer, K.A. 2000: Development and test of a whole-lifetime foraging and bioenergetics model for drift-feeding brown trout. *Transactions of the American Fisheries Society* 129: 315-332.
- Hughes, N.F. 1992: Selection of positions by drift feeding salmonids in dominance hierarchies: model and test for Arctic grayling (*Thymallus arcticus*) in subarctic mountain streams, interior Alaska. *Canadian Journal of Fisheries and Aquatic Sciences* 49:1999-2008.
- Hughes, N.F.; Hayes, J.W.; Shearer, K.A.; Young, R.G. 2003: Testing a model of drift-feeding using 3-D videography of wild brown trout in a New Zealand river. *Canadian Journal of Fisheries and Aquatic Sciences* 60: 1462-1476.
- Hughes, N.F.; Hayes, J.W.; Shearer, K.A.; Stark, J.D. Spatial variation of invertebrate drift density: influence of stream discharge, time of day, taxon and body size in a pool on a New Zealand river. Unpublished manuscript.
- Hughes, N.F.; Kelly, L.H. 1996: A hydrodynamic model for estimating the energetic cost of swimming maneuvers from a description of their geometry and dynamics. *Canadian Journal of Fisheries and Aquatic Sciences* 53: 2484-4293.
- Jowett, I.G. 1999: Rhyhabsim (River Hydraulics and Habitat Simulation) Version 3.30 – computer program.
- Kitchell, J.F.; Stewart, D.J.; Weininger, D. 1977: Applications of a bioenergetics model to yellow perch (*Perca flavescens*) and walleye (*Stizostedion vitreum vitreum*). *Journal of the Fisheries Research Board of Canada* 34: 1922-1935.
- Orth, D. J. 1987: Ecological considerations in the development and application of instream flow-habitat models. *Regulated Rivers: Research & Management* 1: 171-181.
- Railsback, S.F.; Rose, K.A. 1999: Bioenergetics modelling of stream growth: temperature and food consumption effects. *Transactions of the American Fisheries Society* 128: 241-256.
- Rand, P.S.; Stewart, D.J.; Seelbach, P.W.; Jones, M.L.; Wedge, L.R. 1993: Modelling steelhead population energetics in lakes Michigan and Ontario. *Transactions of the American Fisheries Society* 122: 977-1001.
- Rao, G.M. 1968: Oxygen consumption of rainbow trout (*Salmo gairdneri*) in relation to activity and salinity. *Canadian journal of zoology* 46: 781-786.
- Rutherford, J. C. 1994: *River mixing*. John Wiley. Chichester.
- Steffler, P.; Ghanem, A.; Blackburn, J. 2003: River2D Version 0.90 – computer program. University of Alberta, Canada. <http://www.River2D.ualberta.ca/index.htm>
- Stalnaker, C.B.; Lamb, L.; Henriksen, J.; Bovee, K.; Bartholow, J. 1995: The instream flow incremental methodology: a primer for IFIM. National Biological Service, Fort Collins, Biological Report 29. 45 p.

- Stark, J. D.; Shearer, K. A.; Hayes, J. W. 2002: Are aquatic invertebrate drift densities uniform? Implications for salmonid foraging models. *Verhandlungen Internationale Vereinigung für Theoretische und Angewandte Limnologie* 28: 988-991.
- Stewart, D.J. 1980: Salmonid predators and their forage base in Lake Michigan: a bioenergetics-modelling synthesis. PhD. thesis, University of Wisconsin, Madison.
- Stewart, D. J.; Weininger, D.; Rottiers, D. V.; Edsall, T. A. 1983: An energetics model for lake trout, *Salvelinus namaycush*: application to the Lake Michigan population. *Canadian Journal of Fisheries and Aquatic Sciences* 40: 681-698.
- Videler, J.J. 1993: *Fish swimming*. Chapman and Hall, London, U.K.
- Wankowski, J.W.J. 1979: Morphological limitations, prey size selectivity, and growth response of juvenile Atlantic salmon *Salmo salar*. *Journal of Fish Biology*. 14: 89-100.

Appendix 1

Drift transport model description

Appendix 1 Drift transport model description

The invertebrate drift model calculates the settling, entry, vertical dispersion, and lateral dispersion of drifting invertebrates in the study area, estimating their spatially explicit densities. It builds on insights provided by Ciborowski (1987). The model is based on the eulerian frame of reference provided by the equal-discharge streamtubes and the cross-sections that divide these tubes into cells. The model takes as input the densities of invertebrates in each streamtube cell at the upstream cross-section and estimates the concentrations for each of the cells in the downstream cross-sections. The model, as implemented, provides for concurrent processing of an arbitrary number of drift categories, which can vary in initial densities, settling rates, time near bed, and entry rates. The illustrations in the following explanation of the model are limited, for clarity, to processing a single drift category.

A1-1 Values derived from the streamtube description

The use of equal-discharge streamtubes provides a concise geometric description of water flow through the survey reach. The polygons on a given cross-section are either triangles or quadrilaterals. The following quantities are calculated from the 3D description of the streamtubes and the known per-cell discharge (Q):

- area (A_{cell}),
- water velocity between cells ($V_x = 0.5 \cdot Q / A_{\text{cell}} + Q / A_{\text{cell}-1}$),
- centroid (C_{cell}),
- distance to corresponding cell upstream ($D_x = |C_{\text{cell}} - C_{\text{cell}-1}|$),
- time for water to move between cells ($T_x = D_x / V_x$),
- vertical height of the cell at the centroid (H_{cell}),
- area of streambed forming the floor of the streamtube cell (A_{col}) (*i.e.* between cross-sections).

A1-2 Model of dispersion

The dispersion component of the model is similar in the lateral (y) and vertical (z) directions. The form of each analogous equation is the same, and aside from the coordinate axis, the only thing that changes is the value for the dispersion coefficient (k_y and k_z respectively), so we limit this discussion to lateral dispersion.

Rutherford (1994) gives an analytic solution for the mixing equation for constant velocity unbounded flows:

$$S_u(x, y) = \frac{m}{H \sqrt{4\pi k_y x v_x}} e^{-(v_x (y-y_0)^2 / 4k_y x)}$$

This predicts the concentration of a dispersing vertical tracer plume (injected into the flow at lateral position y_0 , with tracer mass m and height H , and travelling with velocity v_x) at downstream displacement x and lateral position y . This equation is closely related

to the normalized Gaussian function, and when we let the constant $\frac{m}{H v_x}$ represent the rate of inflow of the tracer, we can look at mixing as the product of that concentration

and a normal Gaussian function $P(y) = \frac{1}{\sigma \sqrt{2\pi}} e^{-(y-y_0)^2 / (2\sigma^2)}$ where $\sigma^2 = \frac{2k_y x}{v_x}$ as

demonstrated below.

First substitute σ^2 in the numerator:

$$P(y) = \frac{1}{\sigma \sqrt{2\pi}} e^{-(y-y_0)^2 / (2\sigma^2)}$$

$$\rightarrow P(y) = \frac{1}{\sigma \sqrt{2\pi}} e^{-(y-y_0)^2 / (2 \frac{2k_y x}{v_x})}$$

$$\rightarrow P(y) = \frac{1}{\sigma \sqrt{2\pi}} e^{-v_x (y-y_0)^2 / 4k_y x}$$

Then substitute σ^2 in the denominator:

$$\rightarrow P(y) = \frac{1}{\sqrt{\sigma^2} \sqrt{2\pi}} e^{-v_x (y-y_0)^2 / 4k_y x}$$

$$\rightarrow P(y) = \frac{1}{\sqrt{\frac{2k_y x}{v_x}} \sqrt{2\pi}} e^{-v_x (y-y_0)^2 / 4k_y x}$$

$$\rightarrow P(y) = \frac{1}{\sqrt{4\pi k_y x \frac{v_x}{v_x^2}}} e^{-v_x (y-y_0)^2 / 4k_y x}$$

$$\rightarrow P(y) = \frac{v_x}{\sqrt{4\pi k_y x v_x}} e^{-v_x (y-y_0)^2 / 4k_y x}$$

Finally introduce the tracer inflow constant to give Rutherford's (1994) mixing equation:

$$\rightarrow S_u(x, y) = \frac{m}{H v_x} \cdot \frac{v_x}{\sqrt{4\pi k_y x v_x}} e^{-v_x (y-y_0)^2 / 4k_y x}$$

$$\rightarrow S_u(x, y) = \frac{m}{H \sqrt{4\pi k_y x v_x}} e^{-v_x (y-y_0)^2 / 4k_y x}$$

This probability function calculates the likelihood that, after a downstream displacement x , a drifting invertebrate would be found to have lateral displacement y . Integrating this function yields a 'survival' distribution, which can be used to calculate the probability

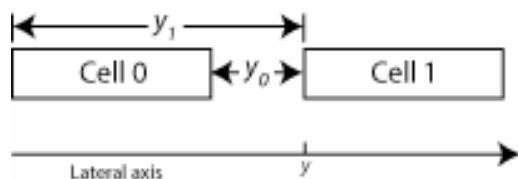
that a drifting invertebrate would disperse farther from its initial location than displacement y :

$$S(x, y) = \int_y^{\infty} P(x, y') dy'$$

Integrating again we find the probability that an invertebrate randomly placed in a cell will disperse farther than lateral coordinate y :

$$S_{cell}(x, y) = \frac{1}{y_1 - y_0} \int_{y_0}^{y_1} (0.5 - \int \frac{e^{-y'^2 / 2\sigma^2}}{\sigma\sqrt{2\pi}} dy') dy = \frac{1}{y_1 - y_0} \int_{y_0}^{y_1} (0.5 - S(x, y')) dy$$

In this equation, y_1 and y_0 are respectively the maximum and minimum distances from the cell to lateral coordinate y as shown here:



This equation is the basis of the model. Using it we can predict the concentrations that disperse from a given cell, across the stream and into and beyond laterally adjacent cells as the water moves between cross-sections. If $S_{cell_0}(x, y')$ is the proportion of the drift that disperses from $cell_0$ at least as far as $cell_1$ and $S_{cell_0}(x, y'')$ is the proportion that disperses past $cell_1$ then $S_{cell_0}(x, y') - S_{cell_0}(x, y'')$ is the proportion of the drift in $cell_0$ that ends up in $cell_1$ at distance x downstream. This computation is made for each pair of cells laterally, and moving from cross-section to cross-section downstream. The equation assumes unbounded flow, and we account for the fact that the flow is not unbounded by using the mirroring method described by Rutherford, where we “reflect” dispersion from the banks. The equation also assumes constant flow, and our model accounts for the fact that velocities vary between cells as follows: First, the effect of dispersion is calculated for each cell individually (to predict the contribution that each cell would make to its neighbours if conditions in the neighbouring cells were the same as in the originating cell). Second, the dispersions predicted for each pair of cells are averaged. The result is the amount of interchange predicted to occur between each pair of neighbouring cells, given their different values of σ and velocity.

The contributions of each cell to the concentrations of cells to the left and right (including mirror cells) are accumulated in an array, and a similar process is followed for vertical dispersion.

A1-3 Model of settling

Settling is considerably simpler than dispersion, as it is modeled as a linear time-dependent process. The settling component of the drift model begins with knowledge of the drift concentrations at the cells upstream and the settling velocity (V_z) for the drift category. The proportion of the concentration in the cell immediately upstream that settles into the next cell below (or, in the case of the bottom-most cell, the proportion removed from the water) is taken as the minimum of 1.0 or $H_{cell} / (T_x V_z)$. The

contributions from and to each cell from settling are added to the array where we stored the contributions of dispersion. Calculations proceed cross-section by cross-section down the stream.

A1-4 Model of entry

The entry component of the model operates only on the concentrations of the bottom tier of streamtubes: those following the bed of the stream. Entry is the only component of the model where we deal explicitly with numbers of insects in addition to concentrations. The inputs are a rate of entry $RE = No. / m^2 / s$ and a time (T_{nb}) that an insect is allowed to spend near the bed. T_{nb} allows us to avoid having large numbers of re-entering insects overwhelm settling in very slow water, and is consistent with Ciborowski's (1983) experimental finding that insects released near the bed spend a constant time in the flow before settling out. The number of insects entering the flow between cross-sections is modeled as $RE \cdot A_{col} \cdot \text{Max} (1, (T_{nb} / T_x))$. The effects of entry are converted to concentrations, and accumulated in the array storing the cumulative effects of dispersion and settling.

Appendix 2

Swimming cost options in the foraging model

Appendix 2 Swimming cost options in the NREI model

The foraging model code includes four options for calculating swimming costs. The models and their parameters, and their origins, are presented in the following sections:

- A polynomial model fitted to Brett and Glass's (1973) swimming cost data for sockeye salmon (Hughes and Kelly 1996).
- Swimming cost equations 1 (Stewart *et al.* 1983) and
- 2 (Kitchell *et al.* 1977) parameterized for steelhead (rainbow) trout from Fish Bioenergetics 3.0 (Hanson *et al.* 1997).
- Swimming cost equation for brown trout in Hayes *et al.* (2000) based on an equation and parameters for rainbow trout in Rand *et al.* (1993) and parameters for brown trout from Elliott (1976).

A2-1 Polynomial swimming cost model – coho salmon

This model is a polynomial fit to Brett and Glass's (1973) swimming cost data for coho salmon fry, developed by Hughes and Kelly (1996).

It takes the form:

$$SC = \frac{t}{3600} \cdot \frac{W}{1000} \cdot OQ \cdot \left(e^{\ln(SMR) + u \cdot \frac{\ln(AMR) - \ln(SMR)}{u_{ms}}} \right)$$

where:

SC is the swimming cost (Joules).

t is time (s).

W is fish mass (g).

OQ is the oxycaloric equivalent (Joules.mg⁻¹ of O₂, taken as 14.1 after Videler 1993).

SMR is the standard metabolic rate (mg of O₂ kg⁻¹.hr⁻¹) oxygen consumption at rest.

AMR is the active metabolic rate (mg of O₂ kg⁻¹.hr⁻¹) oxygen consumption at maximum swimming speed u_{ms} .

And u is the swimming speed (in this case taken to equal u_{ms} , since the fish is assumed to swim at its maximum sustainable swimming speed throughout its foraging forays).

A2-2 Rainbow trout swimming cost models

These models were sourced from Fish Bioenergetics 3.0 (Hanson *et al.* 1997) and are reproduced below. The parameters for the equations are listed in Table 1.

The basic form of the respiration function is:

$$R = ((RA \cdot W^{RB} \cdot f(T) \cdot ACTIVITY) \cdot 13565) / 86400$$

where:

- R is energy cost (Joules.s⁻¹). The original units in Fish Bioenergetics 3.0 are g O₂.g⁻¹.d⁻¹ (*i.e.*, specific rate of respiration). In the above equation a coefficient of 13565 J.g O₂⁻¹ is used to convert to energy units (Joules.s⁻¹) (Elliott and Davidson 1975) and J.d⁻¹ is converted to J.s⁻¹ by dividing by 86400.
- W is fish mass (g)
- RA is the intercept of the allometric mass function (*i.e.*, specific weight of oxygen (g O₂.g⁻¹.d⁻¹) consumed by a 1 g fish at 0°C and zero swimming speed) (g.g⁻¹.d⁻¹)
- RB is the slope of the allometric mass function for standard metabolism (g.g⁻¹.d⁻¹)
- $f(T)$ is the temperature dependence function
- T is water temperature (°C)
- $ACTIVITY$ is an activity multiplier

Fish Bioenergetics 3.0 gives two equations for calculating $f(T)$ and $ACTIVITY$:

Equation 1: Exponential with swimming speed (Stewart et al. 1983)

This equation predicts a continuous exponential increase in respiration costs as a function of temperature and swimming speed.

$$f(T) = e^{(RQ_1 \cdot T)}$$

$$ACTIVITY = e^{(RT \cdot VEL)}$$

where:

- VEL swimming speed (m.s⁻¹)
- RQ_1 approximates the Q_{10} (the rate at which the function increases over relatively low water temperatures)
- RT is the coefficient for swimming speed dependence on metabolism (s.cm⁻¹)

Equation 2: Temperature dependent with activity multiplier (Kitchell et al. 1977)

This equation predicts respiration increasing to a maximum and then declining as a function of temperature with swimming cost ($ACTIVITY$) as an exponential multiplier. This is a departure from Equation 2 in Bioenergetics 3.0 which used a constant activity multiplier (ACT). The exponential activity multiplier that we use is taken from Equation 1.

$$f(T) = V^X \cdot e^{(X \cdot (1-V))}$$

$$ACTIVITY = e^{(RTO \cdot VEL)}$$

where:

$$V = (RTM - T) / (RTM - RTO)$$

$$X = (Z^2 \cdot (1 + (1 + 40/Y)^{0.5})^2) / 400$$

$$Z = LN(RQ_2) \cdot (RTM - RTO)$$

$$Y = LN(RQ_2) \cdot (RTM - RTO + 2)$$

RQ_2 approximates the Q_{10} (the rate at which the function increases over relatively low water temperatures)

RTO ($^{\circ}C$) is the optimum temperature for respiration (where respiration is highest)

RTM ($^{\circ}C$) is the maximum (lethal) water temperature

Table 1 Parameters for the two swimming speed equations for steelhead (rainbow) trout (from Fish Bioenergetics 3.0 (Hanson *et al.* 1997)).

Parameter	Value
RA	0.00264
RB	-0.217
RQ_1	0.06818
RQ_2	2.2*
RT	0.0234
RTO	22*
RTM	26*

* Parameters sourced from Railsback & Rose (1999). In our opinion the value for RTO (22) from this source may be unrealistically high for general application. We suspect that it may be most appropriate for warm water acclimated rainbow trout populations on an energy rich diet. However, it may overestimate the optimal temperature for growth for colder water adapted populations – especially when trout are feeding on invertebrates rather than a fish diet. In our opinion, further research is required to produce reliable energetics equations for rainbow trout.

A2-3 Brown trout swimming cost model

This model is sourced from Hayes *et al.* (2000). It is based on an equation and parameters for rainbow trout in Rand *et al.* (1993) and parameters for brown trout from Elliott (1976).

$$R = ((a \cdot W^{b_1 \cdot T} \cdot e^{b_2 \cdot T} \cdot e^{b_3 \cdot V}) \cdot 4.1868) / 86400$$

where:

R is energy cost (Joules.s⁻¹)
 W is fish weight (g)
 T is temperature ($^{\circ}C$)
 V is velocity (swimming speed) (m.s⁻¹)
 a, b_1, b_2, b_3 are parameters listed in Table 2

Table 2 Parameters for the brown trout swimming cost equation (from Hayes *et al.* (2000) – a, b_1, b_2 were sourced from Elliott (1976) and b_3 was sourced from Rand *et al.* 1993 and Stewart (1980) – cited from Rao (1968).

Temperature ($^{\circ}C$)	a	b_1	b_2	b_3
3.8-7.1	4.126	0.734	0.192	2.34
7.1-19.5	8.277	0.731	0.094	2.34

Appendix 3

Displaying model outputs

The XYZ data contained in the text file outputs of the foraging model and the drift model can be visually displayed as 2D colour contour plots. This involves interpolating between data points from the model output to produce an image that gives the impression of a smooth continuous distribution, which allows for more effective visualization of the position of favourable drift-feeding habitat in the modelled reach.

There are several options for contour plotting the NREI output, including:

- contouring functions in a GIS package,
- purpose written Matlab programs,
- or using the triangulation and contouring functions in River 2D.

The Matlab and River 2D options are discussed here.

A3-1 Matlab programs for displaying model outputs

The following two purpose built matlab programmes allow drift and foraging model output to be visually displayed: flowtubeplot.m, plotfish.m.

“Flowtubeplot.m” produces a series of images, depicting both the drift concentrations and NREI values in the modelled reach as colour contour plots. This program produces both plan view and cross-sectional images, with drift concentrations broken down by drift class and amalgamated.

“Plotfish.m” allows predicted fish positions to be plotted onto the NREI output from a multi-fish simulation.

A3-1.1 Using the Matlab programs

These programs require four input files to be located in the same directory.

1. A boundary file: a text file defining the boundaries of the modelled reach, made using Save Endpoints As ... under the X-Sec menu in R2DStreamTubes (named Boundary.txt)
2. A StreamTubes output file: as used as input to the drift model (named StreamTubes.txt)
3. A drift model output file: as used as input to the foraging model (named xsIn.txt)
4. An NREI output file (named resultsOut.txt)

The command “flowtubeplot” in Matlab will produce a graphic showing a plan view of the cross-sections and streamtubes produced in R2DStreamTubes.

Double-clicking on any of the cross-sections will produce a series of graphical outputs, including both cross-sectional (for the cross-section selected) and depth averaged plan views (for the entire reach) of:

1. the modelled NREI values
2. the modelled drift densities for:
 - a) each drift class individually (No./ m³)
 - b) the overall drift density, summed over all drift classes (No./ m³)
 - c) the total drift biomass (µg/ m³)

Returning to the original cross-section and streamtube plot and double clicking on a new cross-section will change the cross-sectional graphics to the new cross-section.

Clicking outside of the modelled area will exit the program.

Note: The plan view NREI output also shows the total NREI available from the entire reach. This is derived by summing all the positive, area weighted, NREI values.

Entering the command “plotfish” in Matlab will plot the fish locations obtained from a multi-fish simulation over the active plan view NREI graphic. To do this, additional input files are required. One file must contain the *X* and *Y* locations as returned in the resultsOutNREIMax.txt files for the multi-fish model runs. These data make up the first two columns of a tab delimited text file called locations.txt. A third column should contain the lengths of each fish (m). A second file called newfish.txt, which describes the shape of the fish icons used in the plot, is also required.

These graphics can be exported in a format that can be inserted into other documents *etc.* When exporting graphics the Portable Network Graphic (*.png) file type is recommended for best visual results.

A3-2 Displaying outputs using River 2D

It is possible to use the triangulation and contouring functionality of the freely available River 2D software to display the output from the foraging model. This involves using the outputs of the foraging model to create a pseudo-bed file, which than can then be displayed in River 2D.

The first step is to copy the *X* and *Y* values from a resultsOutNREI.txt file into a new file. Add the *X* and *Y* coordinates of the model boundaries (captured using Save Endpoints As ... under the X-Sec menu in R2DStreamTubes) to the end of these columns.

This file can now be used to extract bed elevation values for the series of points, using Read Query XY File under Query in R2DStreamTubes.

Next add a column before the *X* and *Y* columns, containing a series of arbitrary node numbers (running from 1 to *n*). Then add another column after all the others, containing the NREI values for the nodes (from the original resultsOutNREI.txt file). Give the boundary nodes an NREI value of zero.

Finish the file off by typing “no more nodes.” at the end. Save it and change its file extension from “.txt” to “.bed”.

This file is now a pseudo-bed file, with NREI values in the column were Bed Roughness values would normally be. It is ready for display in River 2D.

Open the flow solution used as the basis for this modelling in River 2D. Display colour contours of Bed Roughness clipped to the waters edge. Now use Mesh Edit >Load Bed File ..., to load your newly created “bed” file. Toggle off the Bed Contours under the Display menu. River 2D will now be displaying a colour contoured image of NREI values through the modelled reach.

Note: The water boundaries may not display particularly accurately using this method.

Appendix 4

Control script

This is the core of an R script that takes a line at a time from a StreamTubes output file and passes it the drift model, then the outputs of that to the foraging model. It examines the output of the foraging model, and places a new fish if the highest NREI in the cross section just modelled is higher than the threshold set at the start of this script. It then runs the drift model again, either with the drift-dispersed densities from the drift model of the previous cross section, or the foraging-depleted densities from the foraging model. This script is provided as an inspiration only, and will require customisation to the user's specific requirements.

```
NREI_Limit=1000 #Place a fish at spots greater than this
fishpos=data.frame(NA,NA) #This will store the fish positions
names(fishpos)=c('x','y')
fishCount=0
sforageRun='ForageOneXS' #Specifies the foraging model to be used
MaxNREIout='resultsOutMaxNREI'

#set the first initial conc file
if (file.exists('initialConc.txt')) {file.remove('initialConc.txt')}
if (file.exists('streamtubes.txt')) {file.remove('streamtubes.txt')}
if (file.exists('fishPosns.txt')) {file.remove('fishPosns.txt')}

#the top reach uses the first initial concentration,
#the next reaches carry the altered concentration forward
if (secn==1) {
  file.copy('initialConc0.txt ','initialConc.txt')
}
if (secn==2) {
  file.copy(paste(workingDir,'/',sectionDir[1], '/',flowDir[flown], '
/initialconc.txt',sep=''),paste(workingDir,'/',sectionDir[2], '/',
flowDir[flown], '/initialconc.txt',sep=''))
}
if (secn==3) {
  file.copy(paste(workingDir,'/',sectionDir[2], '/',flowDir[flown], '
/initialconc.txt',sep=''),paste(workingDir,'/',sectionDir[3], '/',
flowDir[flown], '/initialconc.txt',sep=''))
}

icstore=file('icstore.txt','wr')
velstore=file('velstore.txt','wr')

#get the streamtubes data
openfile=file(STREAMTUBES,'r') #set filename here
flowDat1=readLines(openfile)
close(openfile)
rm(openfile)

nCrossSecs=as.double(strsplit(flowDat1[1], ' ')[[1]][4])
Ncolrow=c(as.double(strsplit(flowDat1[2], ' ')[[1]][5]),
as.double(strsplit(flowDat1[3], ' ')[[1]][4]))

NREI_Val=vector("double",nCrossSecs)
NREIstore=matrix(data=0,nrow=0,ncol=3)
```

```
#one cross section at a time
crossSec=1
while(crossSec<nCrossSecs) {
  #grab a single cross section from the streamtubes.txt data
  #and write it out for drift.bat
  targetLinesStart = 4+ (crossSec-1) *
    ((3+(Ncolrow[1]*(Ncolrow[2]+1))-Ncolrow[2]) +
    ((Ncolrow[1]*(Ncolrow[2]+1))-Ncolrow[1]))
  targetLinesEnd = 3+ (crossSec+1) *
    ((3+(Ncolrow[1]*(Ncolrow[2]+1))-Ncolrow[2]) +
    ((Ncolrow[1]*(Ncolrow[2]+1))-Ncolrow[1]))
  newFileData1=flowDat1[c(1:3,targetLinesStart:targetLinesEnd)]
  newFileData1[1]='number of x-secs: 2'
#MODIFY THIS FILENAME TO MATCH THAT IN YOUR DRIFT.BAT
  openfile=file(paste('streamtubes.txt',sep=''),'w')
  writeLines(con=openfile,text=newFileData1)
  close(openfile)
  rm(openfile)

  rm(xsindata)
  # run drift model
  print('Running Drift...')
  #THE STREAMTUBES FILE IN THIS BATCH FILE SHOULD MATCH
  #THAT SPECIFIED IN THIS SCRIPT
  #For some reason show output has to be true for this to work
  system('drift.bat',ignore.stdout=TRUE, show.output.on.console
  =TRUE)
  #creates xsIn.txt for the forage model to work on

  openfile=file('xin.txt')
  xsindata=readLines(openfile)
  close(openfile)
  rm(openfile)

  #Run forage model. The forage model used has to be able to read
  in existing fish positions
  print('Running Forage...')
  #sforageRun stores the name of the foraging model to be used
  system(command= sforageRun,ignore.stdout=TRUE,
  show.output.on.console =FALSE)
  #creates xsOut.txt et al

  #The NREI max file produced by forage.exe contains the depleted
  #drift for the next cross section...
  openfile=file('resultsOutMAXNREI.txt','r')
  NREIFiledata=readLines(openfile)
  close(openfile)
  rm(openfile)

  # ... NREI max val ...
  NREI_Val[crossSec]=as.double(substr(NREIFiledata[3],11,1000))

  #...and the best fish position
  #Note that the NREI threshold for placement is only managed
  #in this file, not in the executable
  #so a fish position will be reported even if it's a negative
  #(or too low) NREI
  #this script then has to accept or reject that suggestion based
  #on the NREI value at that position relative to the threshold
  if (NREI_Val[crossSec]>NREI_Limit) {
    fishCount=fishCount+1;
    fishpos[fishCount,]=c(as.double(strsplit(NREIFiledata[1],
    '')[[1]][3]),as.double(strsplit(NREIFiledata[1], ' ')[[1]][4]))
  }
}
```

```

write(fishCount, 'FishPosns.txt')
for (fishwrite in 1:fishCount) {
  write(paste(sep=',', fishpos[fishwrite,1], fishpos[fish
    write,2]), append=TRUE, 'FishPosns.txt')
}
}

#make a numbered backup of initialConc drift file
#file.copy('initialConc.txt
', paste('initialConc', as.character(crossSec), '.txt', sep=''))

#Write depleted concentration to initialConc, for the next drift
run
#note that the grid may be 'reversed' (if working from Rhyhabsim)
whereas the output concs never are, so we reverse them here if
necessary

file.remove('initialConc.txt')
openfile=file('initialconc.txt', open='w')
#if a fish is placed, use the depleted concentrations for the
next drift run, ...
if (NREI_Val[crossSec]>NREI_Limit) {
  NREIFileddata[4]=0
  write(NREIFileddata[4:5], openfile)
  for (driftwrite in 1:(as.numeric(NREIFileddata[5]))) {
    cat(NREIFileddata[6+(driftwrite-
      1)*(Ncolrow[2]+1)], append=TRUE, file=openfile)
    cat('\n', append=TRUE, file=openfile)
    for (driftRowwrite in 1:Ncolrow[2]) {
      cat(rev(unlist(strsplit(
        NREIFileddata[6+(driftwrite-
          1)*(Ncolrow[2]+1)+driftRowwrite], '\t'))), append
        =TRUE, file=openfile)
      cat('\n', append=TRUE, file=openfile)
    }
    cat('\n', append=TRUE, file=openfile)
  }
  close(openfile)
  rm(openfile)
} else {
  #...otherwise use the drift-modified concentrations
  #extract the concentrations from xsin.txt and
  #write them out in the right format for the drift model
  line4=unlist(strsplit(xsindata[4], ' '))
  nvert=as.numeric(line4[2])
  nhori=as.numeric(line4[1])
  ndrftline=6+1+nvert+1+nvert+2
  ndrft=as.numeric(xsindata[ndrftline])
  ndrftnamelines=(ndrftline+1):(ndrftline+ndrft)
  rm(ndrftline)
  ndrftnames=xsindata[ndrftnamelines]
  writeLines(con=openfile, '0')
  writeLines(con=openfile, as.character(ndrft))
  for (dr in 1:ndrft) {
    #drift name output
    outline=paste(unlist(strsplit(ndrftnames[dr], '
      '))[1], line4[1], line4[2], '\n', sep='\t')
    cat(outline, file=openfile, append=TRUE)
    #drift concentrations output
    drconclines=(ndrftnamelines[ndrft]+2+(dr-
      1)*(nvert+1)):(ndrftnamelines[ndrft]+(dr-
      1)*(nvert+1)+nvert+1)
    for (dl in 1:length(drconclines)) {

```

```

        #reverse the concentrations read to match the
        reversed grid file (Rhyhabsim) we're reading in
        outline=rev(unlist(strsplit(
        xsindata[drconclines[dl]], ' ')))
        cat(outline,file=openfile,sep='\t',append=TRUE)
        cat('\n',file=openfile,append=TRUE)
    }
    cat('\n',file=openfile,append=TRUE)
}
close(openfile)
rm(openfile)
}

#store the concentrations
openfile=file('initialconc.txt',open='r')
concstowrite=readLines(openfile)
close(openfile)
rm(openfile)
for (nnn in 1:length(concstowrite)) {
    cat(concstowrite[nnn],file=icstore,append=TRUE)
    cat('\n',file=icstore,append=TRUE)
}
#store the velocities
for (nnn in 1:Ncolrow[2]) {
    cat(xsindata[7+Ncolrow[2]+1+nnn],file=velstore,append=TRUE)
    #eg for five layers, this gives rows 14 to 18 incl
    cat('\n',file=velstore,append=TRUE)
}
#store the NREI for this cross section in an archive for all
cross sections
openfile=file('resultsOutNREI.txt')
newNREI=readLines(openfile)
close(openfile)
rm(openfile)

newNREI=as.numeric(unlist(strsplit(newNREI[12:length(newNREI)], '
')))
dim(newNREI)=c(3,length(newNREI)/3)
newNREI=t(newNREI)
NREIstore=rbind(NREIstore,newNREI)

crossSec=crossSec+1
}
close(icstore)
close(velstore)

#write out the archived NREI for all cross sections in this section at
this flow
write.csv(NREIstore,file='NREIstore.txt',row.names=FALSE,col.names=NA)

```