LISFLOOD-FP

User manual and technical note

Code release 2.6.2

Professor Paul Bates[1], Dr. Matt Horritt , Dr. Matt Wilson and Neil Hunter

School of Geographical Sciences, University of Bristol, University Road, Bristol, BS8 1SS, UK.

22nd June 2005

[1] Tel: +44-117-928-9108; Fax: +44-117-928-9108; E-mail: Paul.Bates@Bristol.ac.uk

Document information

Project title	LISFLOOD-FP shareware version
Document title	LISFLOOD-FP: User manual and technical note
Code release	2.6.2
Prepared by	Prof. Paul Bates, Dr. Matt Horritt, Dr. Matt Wilson and Neil Hunter
Checked by	Prof. Paul Bates, Dr. Matt Horritt, Dr. Matt Wilson and Neil Hunter
Document version	1.0
Issue date and time	22/06/2005 09:18
Original file stored at	University of Bristol PC serial no. 1699090-RG2 located in School of Geographical Sciences, Room G1.9n
File pathname	E:\Lisflood\Code\V2.6.2\LISFLOOD-FP user manual v2.6.2.doc

Disclaimer

Before using the LISFLOOD-FP software (hereafter "the Software") please read carefully the following terms for use. Extracting files from the `Lisflood v2.6.2.zip` archive on to your computer indicates you accept the following terms.

1. The University of Bristol (hereafter "the Developers") do not warrant that the software will meet your requirements or that the operation of the software will be uninterrupted or error-free or that all errors in the Software can be corrected.
2. You install and use the Software at your own risk and in no event will the Developers be liable for any loss or dammage of any kind including lost profits or any indirect incidental or other consequential loss arising from the use or inability to use the Software or from errors or deficiencies in it whether caused by negligence or otherwise.
3. The Developers accept no responsibility for the accuracy of the results obtained from the use of the Software. In using the software you are expected to make final evaluation in the context of your own problems.
4. Users are not in reliance on any statements warranties or representations which may have been made by the Developers or by anyone acting or purporting to act on their behalf.

Executive summary

This document is the user manual for the shareware implementation of the LISFLOOD-FP raster flood inundation model version 2.6.2. The code provides a general tool for simulating fluvial or coastal flood spreading, with output consisting of raster maps of water depth in each grid square at each time step and, in the case of fluvial flooding, predicted stage and discharge hydrographs at the outlet of the reach. For fluvial situations, this version of LISFLOOD-FP solves the kinematic approximation to the one-dimensional St. Venant equations to simulate the passage of a flood wave along a channel reach. Once bankful depth is exceeded, water moves from the channel to adjacent floodplains sections where two dimensional flood spreading is simulated using the Manning equation and a storage cell concept applied over a raster grid. The model therefore assumes that flood spreading over low-lying topography is a function of gravity and topography. The model is designed to take advantage of recent developments in the remote sensing of topography such as airborne laser altimetry or airborne Synthetic Aperture Radar interferometry which are now beginning to yield dense and accurate digital elevation models over wide areas.

What's new in version 2.6.2?

The LISFLOOD-FP model has undergone substantial changes since the last revision of the user manual for code version 1.2 in 2003. Numerous bug fixes have been implemented, some of which were actually critical for the correct functioning of certain applications and overall the code is more stable and reliable than previously. In addition, there have been some major improvements in the model functionality over this period. The most significant of these changes have been:

- A major rewrite of the code to allow it to include tributary channels in the 1D river model. The code is now able to simulate a dendritic river channel pattern with (theoretically) unlimited stream order. The user describes this network within the `.river` file and the structure of this file is now more complex than previously.
- Inclusion of adaptive time stepping as an alternative to the iteration control in the previous code which used a fixed time step set by the user. In the fixed time step version, setting too large a time step can result in 'chequerboard' oscillations in the solution which rapidly spread and amplify, rendering the simulation useless. For code version 2.0 onwards we include an adaptive time step option based on an analysis of the St. Venant equations and an analogy to a diffusion system (Hunter *et al*., in press) to determine the optimum time step to maintain stability. As the optimum time step reduces quadratically with grid size, for fine grids it can lead to large increases in computational cost. This modification has lead to changes in how time stepping control is prescribed by the user in the `.par` file.
- A number of command line options have also been included (such as `–nch`, `-nfp`, `-inf` etc.) which provide final override control for certain model parameters otherwise set in the input files. These options are used when the model is run in Monte Carlo mode to avoid the need for multiple input files.
- Checkpointing has been added to allow easier restarting of long model simulations following system crashes.
- Point sources have now been added to the `.bci` file to enable simulation of coastal flooding over defences.

Contents

LIST OF FIGURES

List of Tables

1 User manual

1.1 Introduction

This document describes the flood inundation model LISFLOOD-FP. LISFLOOD-FP is a raster-based inundation model specifically developed to take advantage of high resolution topographic data sets (Bates and De Roo, 2000). The model is based on a 1D kinematic wave equation representation of channel flow coupled to a 2D flood spreading model for floodplain flow. Channel and floodplain topography is discretized as a regular grid in ARC-INFO ascii raster format and floodplain flow is calculated using either the Manning equation or a weir equation applied to a storage cell concept originally proposed by Cunge *et al*. (1980). For non-fluvial plain flooding problems, such as coastal dyke breaches, the channel routing component can be suppressed. The code therefore assumes that flood spreading over low-lying topography is a function of gravity and topography. The design philosophy is to produce the simplest physical representation that can accurately simulate dynamic flood spreading when objectively compared to the best available validation data. The computational efficiency so generated allows both large areas to be modelled (e.g. Horritt and Bates 2001b) or Monte Carlo analyses of simulation uncertainty to be conducted (e.g. Aronica *et al*., 2002).

1.1.1 Development background

Estimation of reach scale flood inundation is increasingly a major task for river engineers and managers (see Penning-Rowsell and Tunstall, 1996). For most rivers sufficient observations of flood inundation extent are not available to determine such areas and recourse must be made to some sort of predictive 'model'. These can range in complexity from simply intersecting a plane representing the water surface with a Digital Elevation Model of sufficient resolution to give the flooded area (see for example Puech and Raclot, 2003) to full three-dimensional solutions of the Navier-Stokes equations with sophisticated turbulence closure (see for example Thomas and Williams, 1995; Younis, 1996). However, prediction of flood inundation is not straightforward. Out-of-bank flow in meandering compound channels is now known to be highly three-dimensional and involves the development of a strong shear layer between main channel and floodplain (Knight and Shiono, 1996) as well as spillage of water from the main channel across meander loops (Ervine *et al*., 1993; Ervine *et al*., 1994; Sellin and Willetts, 1996). Moreover, flood inundation extent is highly dependent on topography, and shallow floodplain gradients mean that small errors in modelled water surface elevations may lead to large errors in the predicted inundation front position.

Until relatively recently the most popular approaches to modelling fluvial hydraulics, and thus implicitly flood inundation, at the reach scale (5-50 km) have been one-dimensional finite difference solutions of the full St. Venant equations (see for example Fread, 1984; Samuels, 1990; Fread, 1993; Ervine and MacLeod, 1999) such as MIKE11, ISIS, ONDA, FLUCOMP and HEC-RAS. Such schemes describe the river channel and floodplain as a series of cross sections perpendicular to the flow direction and are thus well suited to parameterization using traditional field surveying methods. Numerical solution of the controlling equations for prescribed inflow and outflow boundary conditions then enables the cross section averaged velocity and water depth at each location to be calculated. More recently, two-dimensional finite difference and finite element models have been developed (see for example Feldhaus *et al*., 1992; Bates *et al*., 1992; Bates *et al*., 1995). These provide a higher order representation of river hydraulics through a full solution of the 2D St. Venant equations that is more consistent with known processes, includes a continuous representation of topography and requires no secondary processing step to determine the flood inundation. However, they have the drawback of increased computational cost and are less well suited to parameterization with traditional cross sectional surveys. Two-dimensional models are best employed in conjunction with a DEM of the channel and floodplain surface which, in conjunction with suitable inflow and outflow boundary conditions, allows the water depth and depth-averaged velocity to be computed at each computational node at each time step.

Such topography data are becoming increasingly available through techniques such as airborne laser altimetry and Synthetic Aperture Radar interferometry. Large amounts of digital elevation data are now being generated by such programmes and there is a need for hydraulic schemes which are able to directly capture as much of this information content as possible and from it generate inundation extent predictions. For reasons of computational cost, full 2D codes are not a currently viable solution here and this has lead a number of researchers to develop coupled 1D/2D codes which combine the simplicity of 1D channel routing approaches with simpler methods of treating floodplain flow that make use of improved topographic data.

LISFLOOD-FP is one such model and was originally developed by Bates and De Roo (2000) in the PC-Raster dynamic modelling language. Subsequently, the code was been re-coded in c++ in order to improve computational efficiency and allow application to larger domains (Horritt and Bates, 2001a) or multiple realisations of the same problem (Aronica *et al*., 2002). A full description of the technical basis of the model is provided in Section 3 of this document.

1.1.2 Previous studies with LISFLOOD-FP

LISFLOOD-FP is a research code to be used to explore the relationship between topography, process representation, scale and uncertainty in flood inundation modelling. As a result, significant experience in flood inundation modelling has been developed which is described briefly below.

1.1.2.1 Calibration, validation and benchmarking studies

LISFLOOD-FP has so far been validated for four river reaches: the Meuse in the Netherlands (Bates and De Roo, 2000), the Thames in the UK (Horritt and Bates, 2001a; Aronica *et al*., 2002), the Severn in the UK (Horritt and Bates, 2001b; Horritt and Bates, 2002) and the Imera basin in Sicily (Aronica *et al*., 2002). In each case the modelled inundation extent has been compared to a flood extent map derived from either air photography, satellite Synthetic Aperture Radar images or ground survey. The data sets represent the best inundation extent information currently available. The flow routing performance of the model has also been analysed in the case of the Meuse and the Severn. In addition, for many of these studies the model has been benchmarked against other standard hydraulic codes as detailed in Table 1. In each case the ability of the model to predict inundation extent is compared in terms of the measure:

$$F = \frac{A_{obs} \cap A_{mod}}{A_{obs} \cup A_{mod}} \text{x}100 \qquad [1]$$

Where A_{obs} and A_{mod} represent the sets of pixels observed to be inundated and predicted as inundated respectively.

Reach name (and length)	Validation data	Maximum LISFLOOD-FP performance (*F*)	Number of calibration runs	Benchmarked against ….
Meuse (35 km)	Air photo inundation extent, SAR inundation image, point hydrometry	82 %	1	TELEMAC-2D, planar lid approximation to free surface
Thames (3 km)	Air photo inundation extent, SAR inundation image	84 %	25	TELEMAC-2D, planar lid approximation to free surface
Severn (60 km)	SAR inundation images for two events, point hydrometry	73 %	500	TELEMAC-2D, HEC-RAS, planar lid approximation to free surface
Imera (15 km)	Ground surveyed flood extent	85 %	500	-

Table 1: Summary of LISFLOOD-FP calibration, validation and benchmarking studies for fluvial application.

The model has also been compared to observed extent data for three coastal flooding applications (see Bates *et al*., in press). Although the data quality is not as good for these tests similar conclusions have been drawn. Results of these studies are outlined in Table 2.

Area	Type of flooding	Domain size	Grid resolution	Number of cells	Event year	Model accuracy (*F*)	Computational time (on a 2.5 GHz pc)
Towyn, North Wales, UK	Defence overtopping	12.5 x 9 km	50m	~45k	1990	0.78	~60 minutes
Fleetwood, UK	Defence overtopping	2.3 x 6.3 km	10m	~145k	1977	0.54	~5 minutes
North Norfolk, UK	Defence breach	40.25 x 42 km	250m	~27k	1938	0.91	~5 minutes

Table 2: Summary of LISFLOOD-FP calibration and validation studies for coastal application.

From these studies the following general conclusions can be drawn:

- Similar to other storage cell models (e.g. Romanowicz *et al*., 1996) the fixed time step version of LISFLOOD-FP is more sensitive to channel than floodplain friction.
- Because the friction coefficients cannot be determined *a priori*, calibration is a necessity.
- When calibrated against inundation extent data LISFLOOD-FP simulates inundation resulting from an independent event as well or better than other standard hydraulic models.
- Performance at predicting inundation extent for such models is close to being within the error of the observed satellite or air photo data which we assume to be only capable of classifying correctly 90 % of the true flooded area (see Horritt *et al*., 2001).
- For the fixed time step model the calibration often needs to be optimised depending on whether one wishes to predict inundation extent or flood wave travel time (Horritt and Bates, 2001b; Hunter *et al.*, in review).
- Minimal improvement in model performance is obtained by using a more complex diffusive wave approximations for the 1D channel and 2D floodplain flow.

1.1.2.2 Scaling behaviour

In order to test the behaviour of the model with respect to changing grid scale, Horritt and Bates (2001b) conducted a scaling analysis for the River Severn reach discussed above. For the 60 km reach between the gauges at Montford Bridge and Buildwas models were constructed at 10, 25, 50, 100, 250, 500 and 1000m scales. Topography for each was parameterised using a laser altimetry survey made available by the UK Environment Agency. A ~1 in 50 year event which occurred in October 1998 was simulated and calibration studies were undertaken for each model. The results were analysed in terms of model ability to predict inundation extent and flood wave travel time through the reach. This analysis showed:

- Model performance in terms of inundation prediction reached a maximum for the 50 m resolution model. This was virtually identical in the 10 and 25 m models.
- The optimum calibration was stationary with respect to model scale for inundation extent, but non-stationary with respect to scale for wave travel time. This was considered to be due to the changing representation of near channel storage affecting travel time but not extent modelling.
- Model performance in predicting inundation extent was near identical at all grid scales below 250 m if: (i) the water levels predicted by the coarse scale model are re-projected back on to the high resolution DEM to reconstruct the detailed shoreline and (ii) the Near Channel Floodplain Storage algorithm (see Section 3) is implemented where there is a significant mismatch between the channel width and the model grid size. This implies that only a coarse

resolution model may be necessary to re-construct water levels in areas of low water surface slope such as floodplains, and that much topographic data need not be included explicitly at the model grid scale.

1.1.2.3 Uncertainty analysis

Given uncertainty over friction values, Aronica *et al*. (2002) have conducted Monte Carlo analysis of parameter uncertainty for the LISFLOOD-FP code using the Generalised Likelihood Uncertainty Estimation (GLUE) technique of Beven and Binley (1992). Dense sampling of the parameter space for the Thames and Imera models showed that there was no single well defined optimum, and instead a broad region the model's parameter space provided an acceptable fit to the observed data (see Figure 1). Further, they hypothesised that for different events these regions will be overlapping but not identical. They concluded that:

- Single deterministic predictions will represent only one of many 'behavioural' model realisations, each of which will give different predictions.
- In reality, the risk of flooding for a particular design event should be conceived as a 'fuzzy' map in which there will be significant spatial structure. Aronica *et al*. (2002) proposed a method in which the global likelihoods calculated using the *F* statistics for each model realisation could be mapped back into real space using the measure P_i^{flood} :

$$P_i^{flood} = \frac{\sum_j f_{ij} F_j^{(2)}}{\sum_j F_j^{(2)}} \qquad [2]$$

Here we take the flood state as predicted by the model for each pixel for each realisation, and weight it according to the measure of fit F to give a generalised relative risk measure for each pixel (see Figure 2), f_{ij} takes a value of 1 for a flooded pixel and is zero otherwise and F_j is the global performance measure for simulation *j*. P_i^{flood} will assume a value of 1 for pixels that are predicted as flooded in all simulations and 0 for pixels always predicted as dry.

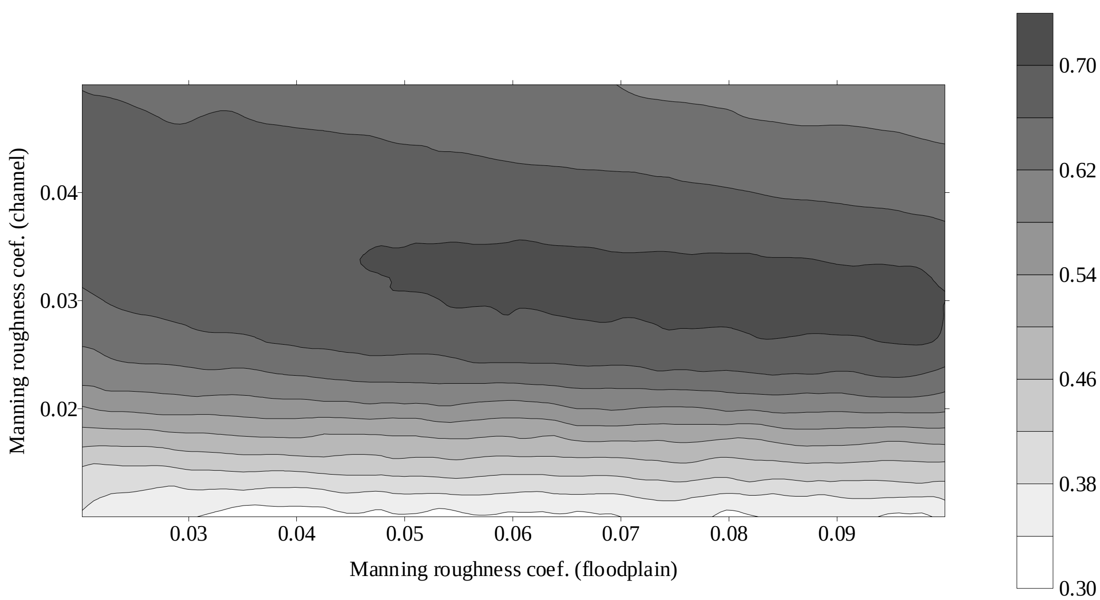

Figure 1: Plot of the F performance measure over the parameter space for the River Thames model.

1.1.2.4 Adaptive time stepping

Comparison of the fixed and adaptive time step versions of the model against analytical solutions and for real test cases have shown that this version of the model is capable of solving a number of the problems with the fixed time step version of LISFLOOD-FP (see Hunter *et al.*, in press; Hunter *et al.*, in review). These studies have concluded that:

- Unlike the fixed time step model, the adaptive version shows sensitivity to variations in floodplain friction that appear both intuitively realistic and in line with the sensitivity behaviour of full 2D solutions of the shallow water equations.
- Whilst parameter sets can be identified for both fixed and adaptive time step models that simultaneously provide acceptable simulations of flood wave travel time and inundation extent, these occur over a broader region of the parameter space for the adaptive time step model.
- Gradients of model performance measures mapped over the parameter space are steeper for the fixed time step model than for the adaptive scheme, indicating that the latter code may be easier to calibrate.
- Use of only either inundation extent or wave travel time data to calibrate the adaptive model results in a rather broad region of the parameter space being identified as capable of providing acceptable simulations. Moreover, use of only inundation extent data in conjunction with a calibration process that seeks to maximise a global measure of fit between observed and predicted inundation may result in parameter sets being identified as acceptable where the floodplain friction is significantly smaller than channel friction due to trade-offs between the two parameters. This is counter-intuitive as in reality, for most floodplain rivers one would expect this situation to be reversed. Use of inundation extent and wave travel time together as validation data sets was shown to eliminate this problem and also reduce the range of acceptable parameters, and hence the predictive uncertainty.
- As the optimum time step for the adaptive model reduces quadratically with grid size, for fine grids it can lead to large increases in computational cost.

1.1.3 Model assumptions and key limitations

The model makes the following assumptions:

- For fluvial flows we assume that the in-channel flow component can be represented using a kinematic 1D wave equation with the channel geometry simplified to a rectangle.
- We assume the channel to be wide and shallow, so the wetted perimeter is approximated by the channel width.
- For plain flooding and out-of-bank flow we assume that flow can be treated using a series of storage cells discretized as a raster grid.
- Flow between storage cells can be calculated using analytical uniform flow formulae (the Manning equation or a weir equation).
- There is no exchange of momentum between main channel and floodplain flows, only mass.
- We assume flow to be gradually varied.
- The model uses standard SI units for length (metres), time (seconds), flux (volume per time in m^3s^{-1}) etc.

The code is limited to situations where a high resolution and accurate topographic data set is available and where there is sufficient information to accurately characterise the model boundary conditions, specifically mass flux with time at all inflow points. In addition, for fluvial flows limited information on channel geometry must also be available.

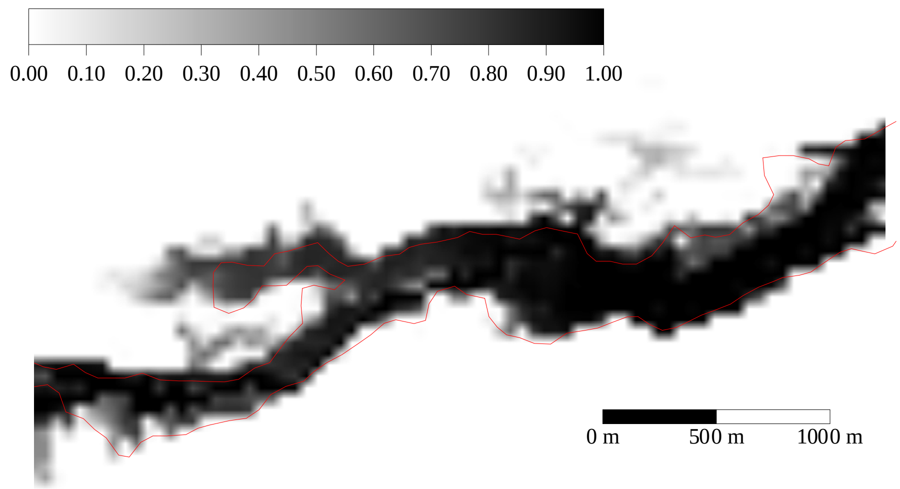

Figure 2: Probability map of predicted inundation, P_i^{flood} , for the December 1992 event for River Thames. The observed shoreline derived from interpretation of satellite Synthetic Aperture Radar data is shown as a red line.

1.2 Installation guide

The model files are provided as a WinZip archive `LISFLOOD-FP v2.6.2.zip` which should first be unpacked into a suitable directory using the WinZip shareware programme. A total of ten files are deployed from the archive as follows:

File name	**Description**
LISFLOOD-FP.EXE	Pre-compiled executable for use on Windows systems
FLOODVIEW.EXE	Windows results file viewer and animation facility
BUSCOT.PAR	Example input file containing model parameters and options
BUSCOT.WEIR	Example input file detailing location and nature of weir linkages between storage cells
BUSCOT.RIVER	Example input file detailing river location and geometry for 1D in-channel calculations
BUSCOT.N.ASCII	Example raster grid of floodplain friction coefficient values in ARC ascii format
BUSCOT.DEM.ASCII	Example raster grid of floodplain elevation heights in ARC ascii format
BUSCOT.BDY	Example input file for time varying boundary conditions
BUSCOT.BCI	Example input file identifying boundary condition types
BUSCOT.OPTS	Example file giving times of satellite overpassses

Table 3: Files deployed from the LISFLOOD-FP `v2.6.2.zip` archive.

These are the model executable, a viewer for LISFLOOD-FP results for Windows PC systems and all the files necessary to run a single example application, in this case for a 3 km reach of the River Thames downstream of Buscot weir.

Once deployed from the archive the files may be installed on a Windows PC operating system. As only the executable is distributed in the shareware release, the following sections are for information only.

1.2.1 Installation on a UNIX or LINUX system

To install the model it is simply necessary to compile the source code. This can be achieved by typing at a command prompt:

```
cc lisflood-fp.c -o lisflood-fp -lm -O3
```

to generate an executable that is speed optimized, or:

```
cc lisflood-fp.c -o lisflood-fp -lm -g
```

to generate an executable that can be used with a debugger. The generated executable is therefore called **`lisflood-fp`**. One should note that all UNIX systems are case dependent. The c++ code is reasonably standard and should work with other compilers (e.g. gcc) although this has not been fully tested.

1.2.2 Installation on a Windows NT, 98 or 2000 system

Again, installation is merely a matter of generating an executable from the `lisflood-fp.cpp` source code using an appropriate c++ compiler. Exact details will depend on the compiler used and some knowledge of these systems by the user is therefore necessary. The code has been successfully compiled at University of Bristol using the Borland Visual c++ Builder software. To use this specific compiler, the user creates a Borland project file for LISFLOOD-FP and compiles the source code. This generates an executable file `lisflood-fp.exe` that can be run from a DOS prompt. A pre-compiled executable generated in this way is provided in the `LISFLOOD-FP v2.6.2.zip` archive.

It should be noted that depending on the exact options and language standards specified by the user for their chosen compiler software, a variable number of warnings may be generated during the building of the executable.

1.2.3 Minimum hardware requirements for Windows systems

LISFLOOD-FP is a relatively simple and efficient code and should run well on a wide variety of systems. It requires little disk space to install, and tests at Bristol have shown that the Buscot test case runs in just a few minutes on even a 100 Mhz Pentium 3 machine with 48 Mb of Ram and a 800kb hard disk. Hardware requirements are much more likely to be limited by the requirements of the compiler software used to generate an executable version of the model. Nevertheless, computational times will be linearly proportional to clock speed, and the size of domain that can be simulated will depend on the amount of RAM memory available. As such a reasonable minimum hardware configuration for serious modelling would be a Pentium 4 PC with 1 Ghz processor clock speed, 128 Mb of Ram and 5 Gb hard disk.

1.3 Data requirements, input files and file formats

1.3.1 Data requirements

Model data requirements are outlined in Table 4.

Data requirement	Source	Comments
Raster Digital Elevation Model.	Typically derived from air photogrammetry or airborne laser altimetry (LiDAR).	Grid resolutions of approximately 25-100m would seem appropriate for most floodplain applications, although smaller resolutions may be preferable. Vertical accuracy of the DEM should generally be less than ±0.25 m. Experience has shown the coarse resolution models (250-500m) can produce good inundation extent predictions if the predicted water levels are projected back on to the high resolution DEM.
Boundary conditions. These can be specified in a number of ways:		
Inflow discharge hydrograph.	Gauging station records. Flow enters the model through the upstream channel cell forming the first location on each river channel vector in the `.river` file.	Model can be used in either steady state or dynamic modes, but flows should be accurate to ±10 %. For dynamic simulations, temporal resolution depends on the speed of the hydrograph rise but typically at least hourly data are required.
Flow across the domain edge	Can be based on gauging station records, spot water elevation or flux measurements, tidal curve or tide/flood frequency data.	Can be used to provide a downstream boundary condition for floodplain flows or simulate tidal forcing for coastal flooding applications.
Point sources within the domain	Can be based on gauging station records, spot water elevation or flux measurements, tidal curve or tide/flood frequency data	Used to specify point source discharges or flow over defences within the domain. Can be used to avoid simulating flow in offshore areas in coastal applications (e.g. Bates *et al.*, in press).
Channel slope.	Taken from the DEM or surveyed cross sections.	Can be set individually for each point on the channel vector if necessary.
Channel width.	Taken from the DEM or surveyed cross sections.	Can be set individually for each point on the channel vector if necessary. Need not be the same as the model grid resolution.
Bankfull depth.	Taken from the DEM or surveyed cross sections.	Can be set individually for each point on the channel vector if necessary..
Channel and floodplain friction.	User defined parameters typically chosen with reference to published tables such as those given by Chow (1959) or Acrement and Schneider (1984).	N_c typically between 0.01 and 0.05 N_{fp} typically between 0.03 and 0.15 Can be set individually for each grid cell if necessary.
Model time step	*Fixed time step version* User defined. An explicit numerical scheme is used so the stability is a function of the cell dimensions and the flow rate. As water enters the model via a single inflow cell at the head of the reach, flow rates in this cell are usually the limiting factor. *Adaptive time step version* Optimum time step to maintain stability is calculated by the code	Varies between applications but typical values are in the range 2-20 s. Calculated by the code. Optimum time step reduces quadratically with grid size. May result in substantial increase in computational cost for fine grids

Table 4: Input data required by the LISFLOOD-FP model.

These data are then assimilated into the model using the input files described in section 1.3.2.

1.3.2 Input file formats

Data is input to the model using eight file types as described below. Users should note that the file extensions are not fixed, comments can only be used in the parameter file (`.par`) and all items are case sensitive.

1.3.2.1 Parameter file (`.par`)

This file contains the information necessary to run the simulation including file names and locations and the main model and run control parameters. The following general principles apply:

- All items in the file are case sensitive.
- Items not recognised are ignored rather than generating an error message.
- The code expects one item per line only.
- If a keyword does not appear the model uses the default value specified in the code and (usually) does not generate an error message
- The order given below is not fixed.
- To comment out a line place a # in the first character space.

The following items are read in via the parameter file:

Item name	**Description**	**Value in the Buscot weir test case**
DEMfile	Digital Elevation Model file name	Buscot.dem.ascii
resroot	Root for naming of results files (e.g. root.op, root.mass, root.0001 etc)	Res (giving res.op, res.mass etc)
dirroot	Relative or absolute path for the directory where results files (excluding the `.chkpnt` file) are to be placed. The directory is created if it doesn't exist already. If this keyword is omitted the results files are placed in the directory in which the model was executed	results
simtime	Total length of the simulation in seconds (real value)	100000.0
initial_tstep	*Fixed time step model* Model time step in seconds (real value) *Adaptive time step model* Initial guess for the optimum time step	10.0
saveint	Interval in seconds at which results files are saved	1000.0
massint	Interval in seconds at which the `.mass` file is written to	100.0
checkpoint	Logical keyword which turns on checkpointing. Followed by interval in hours of computation time at which checkpointing occurs. If no value is set a default value of 2 hours is used. The model automatically looks for the file named `.chkpnt` in the directory from which the model was executed unless the `checkfile` keyword is used. The user needs to delete the `.chkpnt` or turn off this option to commence the simulation again from the beginning.	0.0001
overpass	Time in seconds at which an observed flood image is available for model validation. When specified the model writes a set of results files at this point in the simulation to allow easy model validation	100000
overpassfile	Name of file containing times of multiple satellite overpasses. See section 1.3.2.8.	Buscot.opts. Commented out so not used.
fpfric	Mannings n value for floodplain if spatially uniform	0.06
infiltration	Spatially uniform infiltration rate for the floodplain in ms^{-1}.	0.0000001. Commented out so not used.

manningfile	Name of file containing a grid of floodplain *n* values in ARC ascii raster format to allow spatially variable floodplain friction. This should have the same dimensions and resolution as the DEMfile	buscot.n.ascii. Commented out so not used.
riverfile	Name of file containing channel geometry and boundary condition information. Omit if no channel	buscot.river
bcifile	Name of file identifying floodplain boundary condition types	buscot.bci
bdyfile	Name of file containing information on time varying channel and floodplain boundary conditions	buscot.bdy Commented out so not used.
weirfile	Name of file containing information on location and nature of any weir linkages between cells	buscot.weir Commented out so not used.
stagefile	Name of file containing x, y locations of points at which stage values are to be written to a text file at each `massint`	buscot.stage Not included with Buscot test case. Commented out in `.par` file.
startfile	Name of previous results file in ARC ascii raster format used to provide initial conditions for a model simulation.	res.old. Not included with Buscot test case. Commented out in `.par` file
depthoff	Logical keyword to suppress production of depth files at each `saveint`	Commented out, therefore value is “no”. Depths written out at each saveint
elevoff	Logical keyword to suppress production of water surface elevation files at each `saveint`	Not commented so value is “yes”. Production of elevation files supressed
adaptoff	Logical keyword to suppress adaptive time stepping algorithm	Not commented so value is “yes”. Fixed time step model is used instead.

Table 5: Items read in through the parameter (`.par`) file.

An example `.par` file for the Buscot application is given below:

```
DEMfile                 buscot.dem.ascii
resroot                 res
dirroot                 results
sim_time                100000.0
initial_tstep           10.0
massint                 100.0
saveint                 10000.0
#checkpoint             0.0001
#checkfile              res_old.chkpnt
overpass                100000.0
fpfric                  0.06
#overpassfile           buscot.opts
#manningfile            buscot.n.ascii
riverfile               buscot.river
bcifile                 buscot.bci
#bdyfile                buscot.bdy
#weirfile               buscot.weir
#startfile              res.old
#stagefile              buscot.stage
elevoff
#depthoff
adaptoff
```

As this application involves a steady state simulation and a single satellite overpass, the time varying boundary condition file name (`bdyfile`) and the overpass file name (`overpassfile`) have been commented out. The simulation also uses a spatially uniform floodplain friction with no wiers and begins from the default initial conditions with no checkpointing. Stage outputs at locations within the domain are not requested. The results files all have the suffix `.res` and are placed in the directory `./results`.

1.3.2.2 Channel information file (`.river`)

This file gives information on the location and nature of the channels along the reach. For a model domain containing no channel this file is omitted. The channels are discretized as a single vector along the centreline and the model then interpolates this vector onto the raster grid specified by the user. The vector should run beyond the edge of the model domain. Each channel is described in terms of its width, Manning's n friction coefficient and bed elevation (so hence channel depth when combined with the floodplain elevation described in the DEM) and the linkages between different tributary channels are prescribed using a series of keywords. The user then has two options for prescribing this information.

- Option 1: Uniform channel

Characteristics for each channel are provided for the first and last points of the channel vector, and the code automatically fills in intermediate points by linear interpolation. By specifying the channel bed elevation at the first and last points on the channel vector the user is able to specify the (uniform) bed slope for that channel reach.

- Option 2: Spatially variable channel

Additional values can be specified at any point along the reach, but all 3 values for width, Manning's n and bed elevation must be supplied. One should note that for the kinematic approximation to in-channel flow the down reach slope must be everywhere negative (ie. the channel bed should not increase in elevation in the downstream direction).

The file is formatted as follows

Line 1:	Keyword `Tribs` followed by number of channel segments (if this line is omitted the model assumes a single channel reach)						
Line 2:	Number of data points in the channel vector (i)						
Line 3:	X_1	Y_1	$Width_1$	n_1	Bed $elevation_1$	BC	Value
Line 4:	X_2	Y_2	$Width_2$	n_2	Bed $elevation_2$	Lateral $inflow_2$	
Line 5:	X_3	Y_3	$Width_3$	n_3	Bed $elevation_3$		
etc……	…	…	…	…	…		
Line i:	X_i	Y_i	$Width_i$	n_i	Bed $elevation_i$		

Hence, values for channel width, Manning's n, and bed elevation between line 2 and line i-1 are optional. The first point on the vector must also contain a boundary condition (BC) for the inflow discharge and its value. Here again the user has two options:

- Option 1: Constant inflow.

To use this option to simulate steady state flow BC is given the keyword `QFIX` and the associated value is the inflow discharge at the upstream end of the model in m^3s^{-1}.

- Option 2: Time-varying inflow.

To use this option to simulate a dynamic flood wave BC is given the keyword `QVAR` and the associated value is a boundary identifier chosen by the user, e.g. `upstream1`. Information about the time varying boundary condition data is then held in the time varying boundary condition file `(.bdy)`.

At any point along the reach a lateral inflow may be specified as a source term to represent minor tributary inflows or other catchment hydrological processes which do not require a channel to be represented. Width, Manning's n etc do not need to be given at these points, but can be if necessary.

An example `.river` file for the Buscot application is given below:

```
Tribs 1
133
```

```
22950.000   -1930.000   20.000      0.03  68.740479   QFIX 73.0
23107.670   -1929.020
23140.552   -1924.844
23183.698   -1931.253   20.000      0.03  68.5        QVAR latinflow1
etc….       ….

26739.636   -1161.781   25.000      0.04  68.230
26759.629   -1130.894
26781.873   -1104.059   20.000      0.03  67.139
```

The file thus denotes a fixed inflow of $73m^3s^{-1}$, with channel width starting at 20m, increasing to 25m and back down to 20m, and a time varying lateral inflow at (`23183.698, -1931.253`) with values found in the `latinflow1` part of the `.bdy` file (see below).

The keyword identifier format for lateral inflows also provides the means of describing how tributary channels connect. For a `.river` file with multiple tributary channels the keyword `Tribs` on line of the river file is followed by an integer number which specifies the number of channel segments. If this line is omitted, or if this keyword equals 1, then the model assumes that there is a single channel reach. If multiple segments are present then the first channel is always the main stem. At each point along the main stem where a tributary river enters the user specifies the channel width, Manning's *n* and bed elevation and follows this by the keyword `Trib` and an integer number. This number identifies the segment number in the `.river` file which discharges into the main stem at this point. Segments are numbered sequentially in the order they appear in the `.river` file starting at 0 (which should be the main stem). Each channel segment is described in the `.river` file in exactly the same way as a single channel would be, with the exception that the x, y co-ordinates, width, Manning's *n* and bed elevation for the last point on each segment is followed by the keyword `QOUT` followed by the number of the channel segment into which this tributary discharges. The format is thus:

Line 1:	Number of data points in the channel vector (i)						
Line 2:	X_1	Y_1	$Width_1$	n_1	Bed $elevation_1$	BC	Value
Line 3:	X_2	Y_2	$Width_2$	n_2	Bed $elevation_2$	Lateral $inflow_2$	
Line 4:	X_3	Y_3	$Width_3$	n_3	Bed $elevation_3$		
etc……	…	…	…	…	…		
Line i:	X_i	Y_i	$Width_i$	n_i	Bed $elevation_i$	QOUT	Segment number

Repeating this process allows a dendritic drainage pattern with infinite stream order to be described. As an example, the following is a `.river` file for the Buscot reach assuming a single tributary joining the main stem. In addition this tributary is itself joined by a single tributary. Time varying discharge into the head of each channel segment is described by the keywords `upstream1`, `upstream2` and `upstream3`.

```
Tribs 3
133
22950.000   -1930.000   20.000      0.03  68.740479   QVAR upstream1
23107.670   -1929.020
23140.552   -1924.844
25617.870   -1428.595   20.000      0.03        68.0  TRIB  1

etc….       ….

26706.838 -1179.890
26739.636 -1161.781
26759.629 -1130.894
26781.873 -1104.059     20.000      0.03  67.139
3
24350.0     0.0         5.0         0.03  69.0        QVAR  upstream2
24900.0     -600.0      5.0         0.03  68.5        TRIB  2
25617.870 -1428.595     5.0         0.03  68.0        QOUT  0
2
22950.0     -600.0      5.0         0.03  69.0        QVAR  upstream3
```

```
24900.0     -600.0      5.0         0.03  68.5        QOUT  1
```

1.3.2.3 Boundary condition type file (`.bci`)

This file specifies boundary conditions not associated with the channel. There can be any number of boundaries on the edge of the domain or at points within the domain itself.

Column 1: Boundary identifier taking a value of N, E, S, W or P and referring to the north, east, south or west boundaries or P referring to a point source

Column 2: start of boundary segment (easting or northing in map co-ordinates) for edge boundaries or easting in map co-ordinates for a point source location

Column 3: End of boundary segment (easting or northing in map co-ordinates) for edge boundaries or northing in map co-ordinates for a point source location

Column 4: Boundary condition type

Column 5: Boundary condition value. This varies according to boundary condition type as indicated in Table 6.

Possible boundary condition types and their associated values are given in Table 6.

Boundary condition type	**Description**	Value supplied in column 5 of the `.bci` file
CLOSED	Zero-flux (default option)	None
FREE	Uniform flow	None
HFIX	Fixed free surface elevation	Free surface elevation in metres
HVAR	Time varying free surface elevation,	Boundary identifier (e.g. `downstream1`) corresponding to data in the user supplied `.bdy` file.
QFIX	Fixed flow into domain	Mass flux per unit width (m^2s^{-1}). For a boundary segment this is multiplied within the code by the length of the boundary segment to give the mass flux in m^3s^{-1}. For a point source the mass flux per unit width is multiplied by the cell width to the mass flux in m^3s^{-1}.
QVAR	Time varying flow into domain	Boundary identifier (e.g. `upstream1`) corresponding to data in the user supplied `.bdy` file

Table 6: Types of boundary condition available in the `.bci` file.

An example `.bci` file for the Buscot application is given below:

```
E           -1200              -1800      HFIX        69.000
```

This specifies a fixed free surface elevation boundary on the east side of the domain between easting `-1200` and northing `-1800`.

1.3.2.4 Time varying boundary conditions file (`.bdy`)

This file is used to specify time varying boundary conditions (keywords `QVAR` or `HVAR` in the `.river` or `.bci` files) associated with either a channel segment, boundary segment or point source. For each time varying boundary condition the format for the file is as follows:

Line 1: Comment line, ignored by LISFLOOD-FP.

Line 2: Boundary identifier (this should be consistent with notation supplied in the `.river` or `.bci` file).

Line 3: Number of time points at which boundary information is given followed by a keyword for the time units used (either ‘hours’ or ‘seconds’).

Line 4: Value$_1$ Time$_1$

Line 5: Value$_2$ Time$_2$

etc…. … …

Line i: Value$_i$ Time$_i$

Where Value$_i$ is the value of the relevant quantity for the given boundary type. For all `HVAR` boundaries Value$_i$ is a water surface elevation in metres. However, the units of Value$_i$ for `QVAR` boundaries depend on whether the given boundary identifier is specified in the `.river` or `.bci` files. This seems complex, but is a consequence of having a 1D channel model coupled to a 2D floodplain model and actually makes setting up the code a lot easier. For a `QVAR` boundary specified in the `.river` file Value$_i$ is given as mass flux with units m^3s^{-1}. By contrast, for a `QVAR` boundary specified in the `.bci` file Value$_i$ is given as mass flux per unit width with units m^2s^{-1}. In this latter case the flux per unit width is multiplied within the code either by the length of the boundary segment (for a boundary flux) or the cell size (for a point source) to give the mass flux in m^3s^{-1}.

An example .bdy file for the Buscot application is given below

```
QTBDY   Obtained from results file C:\HALCROW\KISMOD\KISL_100.ZZN
downstream1
      3                  seconds
      70.                0
      71.000             25000
      70.000             50000
```

This specifies a water surface elevation varying in time between 70 and 71m for the boundary segment identified by the keyword `downstream1`. The location of this segment is specified in the `.bci` file. Currently the only supported units are "seconds" and "hours". If an identifier specified in the `.river` or `.bci` file is not found in the `.bdy` file, or one found in the `.bdy` file has no reference in the `.river` or `.bci` file, a warning is output (verbose mode only - see below) and the boundary defaults to zero flux.

1.3.2.5 Digital Elevation Model file (`.dem.ascii`)

This file specifies the Digital Elevation Model used by the model. It consists of a 2D raster array of ground elevations in ARC ascii raster format. The file may be manipulated using either the ARC-View or ARC-Info Geographical Information System platforms or manually edited using a text editor. For full details on the ARC ascii raster format the user is referred to the ARC documentation. A brief summary of the format is provided below.

The file consists of a 6 line header followed by the numerical values of each data point on the grid as a 2D array of *i* rows and *j* columns. Each line of the header consists of a self-explanatory keyword followed by a numeric value. As an example, the header for the Buscot application is given below (comments in brackets are not part of the file format):

```
ncols         76        (Number of columns)
nrows         48        (Number of rows)
xllcorner     22950     (X cartesian co-ordinate of the lower left
                        corner of the grid in metres)
yllcorner     -2400     (Y cartesian co-ordinate of the lower left
                        corner of the grid in metres)
cellsize      50.0      (Cell size in metres)
NODATA_value  -9999     (Null value)
```

1.3.2.6 Floodplain friction coefficient file (`.n.ascii`)

This file can be used by the user to specify a spatially variable friction coefficient across the floodplain by assigning values of Manning's n to each cell on the raster grid. Again, the file format is an ARC-View ascii raster as described in section 1.3.2.5 above.

1.3.2.7 Weir cell linkage specification file (`.weir`)

In order to correctly represent embankments, weirs and structures the linkage between two given cells may be represented by a weir equation rather than the Manning formulae. Information about these linkages is given in the `.weir` file. The file format is as follows:

Line 1; number of weir-type linkages between cells (i).

Line 2;	X_1	Y_1	$Direction_1$	$Qcoeff_1$	Crest $height_1$	Modular $limit_1$	$Width_1$
Line 3:	X_2	Y_2	$Direction_2$	$Qcoeff_2$	Crest $height_2$	Modular $limit_2$	$Width_2$
etc…	…	…	…	…	…	…	…
Line i:	X_i	Y_i	$Direction_i$	$Qcoeff_i$	Crest $height_i$	Modular $limit_i$	$Width_i$

Where:

X and Y are the grid co-ordinates in Eastings and Northings of a cell with a weir linkage. X and Y can be located anywhere within the cell being identified.

Direction identifies the cell face with the linkage N, E, S or W (Obviously 10 42 W is the same as 10 41 E). If flow in only one direction is required (e.g. for a culvert), the direction may be fixed by using the tags NF, EF, SF, or WF.

Qcoeff is the weir discharge coefficient, typically ranging from 0.5-1.7 and taking a value if 1.7 for a standard broad crested weir.

Crest height is the height of the weir in m.a.s.l or the co-ordinate system being used in the model.

Modular limit is the modular limit of the weir, typically 0.9.

Width is an optional width for the weir which defaults to the grid size if not supplied.

An example `.weir` file for the Buscot application is given below. Note that the weir width is not specified so a the grid size (50m) is used as a default.

```
14
 22950      -1700       N           1.7         72          0.9
 23000      -1700       N           1.7         72          0.9
 23050      -1700       N           1.7         72          0.9
 23100      -1700       N           1.7         72          0.9
 23150      -1700       N           1.7         72          0.9
 23200      -1700       N           1.7         72          0.9
 23250      -1700       N           1.7         72          0.9
 23300      -1700       N           1.7         72          0.9
etc
```

1.3.2.8 Multiple overpass file (.opts)

This file is used to specify the times in seconds of multiple satellite overpasses during a single simulation. This option is activated by including the optional keyword `overpassfile` followed by a filename in the `.par` file. The model then outputs a set of results files at each time specified. The file format is as follows:

Line 1; Number of satellite overpasses

Line 2; Time of 1st overpass in seconds of simulation time

Line 3: Time of 2nd overpass in seconds of simulation time

etc… … … … … … … …

Line i: Time of n^{th} overpass in seconds of simulation time

An example `.opts` file is given below:

```
4
900.0
1800.0
2700.0
3600.0
```

1.4 Setting up a simulation

Setting up a simulation requires generation of the above files populated with appropriate parameter values. There is no specific order in which to attempt these tasks but the following series of steps may appropriate in many cases:

1. Generate an appropriate floodplain DEM using the ARC-View or ARC-Info systems. Typically this would consist of high-resolution topography data in some format that is then manipulated to give an ascii raster grid. Save this as a `.dem.ascii` file.
2. If spatially variable floodplain friction is to be specified use either ARC-View, Excel or a short programme to generate a further ARC ascii raster grid of the same dimensions as the `.dem.ascii` file and populate this with appropriate Manning's n values. Save this as a `.n.ascii` file.
3. Generate a vector of the channel centre line in the same co-ordinate system as used for the `.dem.ascii` file using either ARC-View, AutoCAD or some other digitising package. This information can either be taken from a LiDAR survey if available (as the water surface generates a null return to the sensor which may then be used to define the channel) or from large scale topographic maps.
4. Populate the `.river` file with channel and boundary condition information. Channel data should come from either site inspection or surveys or historic cross-sectional surveys. If the latter are used the possibility of geomorphic change should be allowed for.
5. Assign boundary condition data to the `.bci` and `.bdy` files if required.
6. Prescribe weir linkages if required in the `.weir` file.
7. Define model run time parameters and file names in the `.par` file.
8. Use the model to generate a set of initial conditions. This may be necessary for certain dynamic simulations and merely consists of the results file from a previous simulation. Specify the name of the initial conditions file after the keyword `startfile` in the `.par` file.

The model should now be ready for simulations to begin.

1.5 Running a simulation

To run the model, open a DOS or UNIX/LINUX shell and at a command prompt type the name of the executable file generated by the compiler and the name of the model parameter file.

```
lisflood-fp [command line options] model.par
```

Where 'model' is the file naming convention chosen by the user (in the case of the example application given with this code release this is buscot.par). The command line options can be used to turn on diagnostic information and warnings as the model runs or used to provide override control of certain model parameters specified in the input files. The latter facility is useful for running the model in Monte Carlo mode from a batch file as it avoids the need for multiple input file versions. Command line options implemented to date are given in Table 7 below:

Option	**Description**
-v	Verbose mode. With –v turned on the model generates a number of runtime diagnostic messages.
-version	With parameter file name omitted this option allows the user to check the version number of the executable.
-gzip	Causes model output files to be compressed on the fly. Note: this option issues a system command to run `gzip` at each `saveint`.
-dir dirname	Gives the directory name for results files. Overrides the name given after the keyword `dirroot` in the `.par` file.
-simtime value	Allows the simulation time to be specified in the command line followed by a value for the simulation time in seconds. Overrides the value given after the keyword `sim_time` in the `.par` file.
-nch value	Implements a spatially uniform channel friction for all channel segments with a value given in terms of Manning's *n*. Overrides the value given in the `.river` file.
-nfp value	Implements a spatially uniform floodplain friction with a value given in terms of Manning's *n*. Overrides the value given after the keyword `fpfric` in the `.par` file.
-inf value	Implements a spatially uniform infiltration loss across the whole floodplain with a value given in ms^{-1}. Overrides the value given after the keyword `infiltration` in the `.par` file.
-weir filename	Gives the name of the `.weir` file. Overrides the name given after the keyword `weirfile` in the `.par` file.
-checkpoint	Turns checkpointing on with default features. Code is checkpointed every 2 hours of computational time using the output file naming convention specified in the `.par` file after the keyword `resroot`. See Section 1.3.2.1. Overrides the interval given after the keyword `checkpoint` in the `.par` file.

Table 7: Command line options for LISFLOOD-FP.

The order in which command line options are used is not important.

In verbose mode the diagnostic messages are mostly self-explanatory. The exception is:

```
Smoothing bank cells with tolerance            htol
```

Where `htol` is a numeric value in metres. This refers to the operation of the `SmoothBanks` subroutine which corrects a potential source of model instability. This subroutine searches through the floodplain elevations in cells adjacent to the channel and identifies areas of low lying floodplain that are within a certain vertical tolerance (`htol`) of the interpolated channel bed elevation at that point. If found the elevation of the relevant floodplain cells are raised to the sum of the bed elevation and `htol`. For the Buscot example, `htol` is set to the default value of 0.5 m. To change the default value of `htol` the user must manually edit the relevant source code and re-compile the programme.

1.5.1 Output file formats

During a simulation the model produces a series of results files named according to the `resroot` convention given in the parameter file. These are placed in the `dirroot` directory if this keyword and a directory name are placed in the parameter file. The output files are produced at different time intervals according to specifications made by the user in the parameter file and are described below.

1.5.1.1 Mass balance output file (`.mass`)

This file gives details of the model mass balance performance and is written at the interval specified by the keyword `massint` in the parameter file. The output consists of 11 columns of data:

Column 1: Time. The time in seconds at which the data was saved.

Column 2: Tstep. Time step specified by the user (initial time step in the adaptive model)

Column 3: MinTstep. Minimum time step calculated by the adaptive model during the iteration

Column 4: ItCount. Number of time steps since the start of the simulation.

Column 5: Area. Area inundated in km^2.

Column 6: Vol, e-6. Volume of water in the domain in 10^6 m^3.

Column 7: Qin. Inflow discharge in m^3s^{-1}.

Column 8: Hds. Water depth at the downstream exit of the model domain.

Column 9: Qout. Calculated outflow discharge at the downstream exit of the model domain in m^3s^{-1}.

Column 10: Qerror. Mass balance error per time step in m^3s^{-1}.

Column 11: Inf, e-3. Cumulative Infiltration loss over the simulation in 10^3 m^3.

1.5.1.2 Water depths at time of satellite overpass file (.op)

This file consists of a grid of water depths in ARC ascii raster format for each pixel at the time of each satellite overpass specified using the parameter file keyword `overpass`, or `overpassfile` for multiple outputs (see section 1.3.2.8). Multiple overpass filenames will take the format of `outX.op`, where `out` denotes the `resroot` given in the parameter file, and `X` is the Xth overpass time given in the `overpassfile`. Numbering of overpass times commences at zero.

1.5.1.3 Channel water surface profile at time of satellite overpass (`.profile`)

This file gives the channel water surface profile at the time of each satellite overpass specified using the parameter file keyword `overpass`, or `overpassfile` for multiple outputs (see section 1.3.2.8). Multiple overpass filenames will take the format of `outX.profile`, where `out` denotes the `resroot` given in the parameter file, and `X` is the Xth overpass time given in the `overpassfile`. Numbering of overpass times commences at zero.

This is a text file consisting of three columns of data:

Column 1: Chainage. Distance along the channel thalweg from the upstream boundary in metres.

Column 2: DEM Z. Channel bed elevation in metres.

Column 3: H. Water depth in metres.

1.5.1.4 Synoptic water depth and water surface elevation files (`.0001, .0002, .0003`, `.001.elev, .0002.elev, .0003.elev` etc)

These files consist of a grid of water depths or water surface elevations in ARC ascii raster format for each pixel at each save interval (`saveint`) specified in the parameter file. Production of each set of files can be turned off by putting the logical keywords `depthoff` or `elevoff` in the `.par` file. Units are in metres.

1.5.1.5 Maximum water surface elevation file (`.mxe`) and maximum water depth (`.max`)

These files consist of a grid in ARC ascii raster format of the maximum water surface elevation (`.mxe`) predicted by the model for each pixel over the course of the simulation, or the maximum water depth (`.max`). Units are in metres.

1.5.1.6 Time of initial inundation (`.inittm`), time of maximum depth (`.maxtm`) and total time of inundation (.totaltm)

These files consist of a grid in ARC ascii raster format of the time of initial inundation for each pixel (`.inittm`), the time of maximum inundation depth in each pixel (`.maxtm`) or the total time for which a pixel is inundated (`.totaltm`). Units are in hours from the start of the simulation.

1.5.2 Visualising model results

To view the results files the user may either use the ARC-View software or the Windows visualisation and animation programme `FloodView` bundled with the installation. FloodView can be launched from Windows explorer and allows results files, DEM files and floodplain friction files to be loaded, overlain, visualised and animated.

1.6 References and bibliography

Acrement, G.J. and Schneider, V.R., (1984). *Guide for selecting Manning's roughness coefficients for natural channels and floodplains*. U.S. Department of Transportation, Federal Highways Administration, Report No. FHWA-TS-84-204, 62 pp.

Aronica, G., Bates, P.D. and Horritt, M.S., (2002). Assessing the uncertainty in distributed model predictions using observed binary pattern information within GLUE. *Hydrological Processes*, **16**, 2001-2016.

Bates P.D., Anderson M.G and Hervouet J-M., (1995). Initial comparison of two two-dimensional finite element codes for river flood simulation*. Proceedings of the Institution of Civil Engineers, Water Maritime and Energy*, **112**, 238-248.

Bates, P.D. and De Roo, A.P.J., (2000). A simple raster-based model for floodplain inundation. *Journal of Hydrology*, **236**, 54-77.

Bates, P.D., Anderson, M.G., Baird, L., Walling, D.E. and Simm, D., (1992). Modelling floodplain flow with a two-dimensional finite element scheme. *Earth Surface Processes and Landforms*, **17**, 575-588.

Bates, P.D., Dawson, R.J., Hall, J.W., Horritt, M.S., Nicholls, R.J., Wicks, J. and Hassan, M.A.A.M., (in press). Simplified two-dimensional numerical modelling of coastal flooding and example applications. *Coastal Engineering*.

Beven, A. and Binley, 1992. The future of distributed models: model calibration and uncertainty prediction. *Hydrological Processes*, **6**, 279-298.

Chow, V.T., (1959). *Open channel hydraulics*, McGraw-Hill, New York, 680pp.

Cunge, J.A., Holly, F.M. Jr. and Verwey, A., (1980). *Practical aspects of computational river hydraulics*, Pitman, London, 420pp.

Ervine D.A. and MacCleod, A.B., (1999). Modelling a river channel with distant floodbanks. *Proceedings of the Institution of Civil Engineers, Water Maritime and Energy*, **136**, 21-33.

Ervine D.A., Sellin R.H.J., and Willets B.B., (1994). Large flow structures in meandering compound channels. In: White W.R. and Watts J. (eds.), *Proceedings of the 2nd International Conference on River Flood Hydraulics*, John Wiley & Sons, 459 - 470.

Ervine D.A., Willets B.B., Sellin R.H.J. and Lorena M., (1993). Factors affecting conveyance in meandering compound flows. *Journal of Hydraulic Engineering American Society of Civil Engineers*, **119**, 1383 - 1399.

Estrela, T. and Quintas, L., (1994). Use of a GIS in the modelling of flows on floodplains. In: White W.R. and Watts J. (eds.), *Proceedings of the 2nd International Conference on River Flood Hydraulics*, John Wiley & Sons, 177-189.

Feldhaus, R., Höttges, R., Brockhaus, T. and Rouvé, G., (1992). Finite element simulation of flow and pollution transport applied to part of the River Rhine. In: *Hydraulic and environmental modelling: estuarine and river waters*, R.A. Falconer, K. Shiono and R.G.S. Matthews (eds.), Ashgate Publishing, Aldershot, 323-334.

Fread, D.L., (1984). Flood routing. In: M.G. Anderson and T.P. Burt (eds), *Hydrological Forecasting*, John Wiley and Sons, Chichester, Chapter 14.

Fread, D.L., (1993). Flood routing. In D.R. Maidment (ed*), Handbook of Applied Hydrology*, Mc-Graw Hill, New York, Chapter 10.

Horritt, M.S. and Bates, P.D., (2001a). Predicting floodplain inundation: raster-based modelling versus the finite element approach. *Hydrological Processes*, **15**, 825-842.

Horritt, M.S. and Bates, P.D., (2001b). Effects of spatial resolution on a raster based model of flood flow. *Journal of Hydrology*, **253**, 239-249.

Horritt, M.S. and Bates, P.D., (2002). Evaluation of 1-D and 2-D numerical models for predicting river flood inundation. *Journal of Hydrology*, **268**, 87-99.

Horritt, M.S., Mason, D.C. and Luckman A.J., 2001. Flood boundary delineation from Synthetic Aperture Radar imagery using a statistical active contour model. *International Journal of Remote Sensing*, **22**, 2489-2507.

Hunter, N.M., Bates, P.D., Horritt, M.S. and Wilson, M.D. (in review). Improved simulation of inundation dynamics in storage cell hydraulic models. *Institution of Civil Engineers, Journal of Water Management*.

Hunter, N.M., Horritt, M.S., Bates, P.D., Wilson, M.D. and Werner, M.G.F., (in press). An unconditionally stable explicit solution for raster-based storage cell modelling of floodplain inundation. *Advances in Water Resources*.

Knight D.W. and Shiono K. (1996). River channel and floodplain hydraulics. In : *Floodplain Processes*, Anderson M.G., Walling D.E. and Bates P.D. (eds.), John Wiley and Sons, Chichester, 139-182.

Penning-Rowsell, E.C. and Tunstall, S.M., (1996). Risks and resources: defining and managing the floodplain. In: *Floodplain Processes,* Anderson M.G., Walling D.E. and Bates P.D. (eds.), John Wiley and Sons, Chichester, 493-533.

Puech, C. and Raclot, D. 2002. Using geographical information systems and aerial photographs to determine water levels during floods. *Hydrological Processes*, **16**, 1593–1602.

Romanowicz, R., Beven, K.J. and Tawn, J., (1996). Bayesian calibration of flood inundation models. In : *Floodplain Processes,* Anderson M.G., Walling D.E. and Bates P.D. (eds.), John Wiley and Sons, Chichester, 333-360.

Samuels, P.G., (1990). Cross section location in one-dimensional models. In W.R. White (ed), *International Conference on River Flood Hydraulics.* John Wiley and Sons, Chichester, 339-350.

Sellin R.H.J. and Willets B.B., (1996). Three-dimensional structures, memory and energy dissipation in meandering compound channel flow. In : *Floodplain Processes,* Anderson M.G., Walling D.E. and Bates P.D. (eds.), John Wiley and Sons, Chichester, 255-298.

Thomas, T.G. and Williams, J.J.R., (1995). Large eddy simulation of of turbulent flow in an asymetric compound open channel. *Journal of Hydraulic Research*, **33**, 27-41.

Werner, M.G.F. & Lambert, M.F. in press. Evaluation of modelling approaches for river reach scale inundation modelling, *Journal of Hydraulic Research*.

Younis, B.A., (1996). Progress in turbulence modelling for open channel flows. In : *Floodplain Processes,* Anderson M.G., Walling D.E. and Bates P.D. (eds.), John Wiley and Sons, Chichester, 299-232.

1.7 Language standards

LISFLOOD-FP is written in ansi-C with explicit type casting where necessary to enable compilation with an ansi-C++ compiler, and is contained in a single file. No non-standard header files are required. Switched compiler directives are included at the start of the code to allow compilation as a Borland C++ Builder project, provided the correct *.bpr etc files are present.

1.8 Programme structure and subroutine map

The program consists of a main section which calls a number of subroutines with most communication via global variables (see below for a description). Execution flow is fairly linear: parameter reading, data reading, loop through time steps. The major (ones omitted should be self explanatory) subroutines are (in no particular order):

```
void SmoothBanks();
```
Removes low lying bank areas, defined as pixels next to channel <0.5m above channel bed in order to improve model stability.

```
double DomainVol();
```
Calculates volume of water in the domain (including channel) for mass balance calculation.

```
void ChannelQ();
```
Takes water depths in channel and propagates them forward by 1 time step using explicit non-linear scheme.

```
void FloodplainQ();
```
Calculates flow between floodplain cells and between floodplain and channel cells using either Manning's equation or a weir equation.

```
void UpdateH();
```
Change water depths in floodplain cells by summing floodplain flows and multiplying by the time step. Channel cells are unaffected.

```
void DryCheck();
```
Check for potentially drying cells i.e. ones with negative depth at next time step. Floodplain flows are scaled accordingly to smoothly return water depths in a cell to zero during drying without incurring a mass balance error.

```
float CalcA(float n,float s,float w,float Q);
```
Calculate cross sectional area of flow in shallow rectangular channel given Manning's n, bed slope, width and discharge.

```
float BankQ(int chani);
```
For the chanith channel cell calculates the flow *out* of the channel by summing appropriate floodplain flows.

```
float CalcQ(float n,float s,float w,float h);
```
Calculate discharge in shallow rectangular channel given Manning's n, bed slope, width and flow depth.

```
void IterateQ();
```
Iterates through time steps by calculating channel flow, floodplain flow, boundary flow, checking for drying elements and then floodplain updating depths.

```
float Newton_Raphson(float Ai,float dx,float a0,float a1,float c);
```
Non-linear solver for channel flow at next time step.

```
void LoadDEM(char *fname);
```
Loads DEM and mallocs and initialises all rasters. x and y dimensions and bottom left corner location are taken from this file – those in all other rasters are ignored.

```
void write_ascii(float *f,FILE *fp);
```
General routine to write data to ascii raster file.

```
void BCs();
```
Calculates floodplain flow at edges of the domain in response to boundary conditions (for any of imposed flow, imposed surface elevation or zero flux).

```
void LoadBCs();
```
Loads boundary conditions for floodplain flow from the .bci file and sets up the BC_Ident, BC_Val and BC_Name arrays (more information below).

```
void LoadBCVar();
```
Loads time varying boundary conditions and channel flows (if present) from .bdy file and interpolates them in time, storing results in arrays BCVarlist and QVarlist.

```
float CalcFPQx(int i,int j);
float CalcFPQy(int i,int j);
```
Calculate flows in x or y direction according to Manning's equation for adjacent non-weir floodplain cells.

```
float CalcWeirQx(int i,int j);
float CalcWeirQy(int i,int j);
```
Calculate floodplain flow between cells separated by a weir.

```
void BoundaryFlux();
```
Calculates floodplain and channel flow in and out of boundary of domain for use in mass balance calculation.

```
void write_elev();
```
Writes free surface elevation (z+h) in ascii raster format. Where h<0, a NODATA_value of –9999 is output.

1.9 Key variable names

1.9.1 General variables:

`int xsz,ysz;`	Number of cells in x,y directions
`int chsz;`	Number of channel cells
`float dx,dy,dA;`	x,y cell size, cell area
`float FPn;`	Floodplain Manning's n (uniform)
`int Nit,ts;`	Number of iterations, current iteration
`float Tstep;`	Time step
`float SolverAccuracy;`	Convergence criterion of Newton-Raphson solver
`float Qin,Qout,QChanOut,Hds;`	Flow in, out, channel out, depth at downstream end of channel
`float DepthThresh;`	Depth below which cell is considered dry
`float MaxHflow=10.0;`	Maximum flow depth (aids stability), this only affects calculation of floodplain flow, depths can be greater than this value.

`int SaveInt,MassInt,op;`	Interval at which depths and mass balance information are saved, and overpass time
`float htol=0.5;`	Tolerance for removing low lying bank areas
`float tlx,tly,blx,bly;`	Top left, bottom right corners
`int ChannelPresent,NCFS,verbose;`	Switches for channel, Near Channel Floodplain Storage and verbose mode.
`int weir;`	Switch for presence of weirs

1.9.2 Character variables

`char demfilename[80];`	
`char chanfilename[80];`	
`char startfilename[80];`	
`char Qfilename[80];`	
`char resrootname[80];`	
`char qfilename[80];`	
`char nfilename[80];`	
`char rivername[80];`	
`char bcifilename[80];`	
`char bdyfilename[80];`	
`char weirfilename[80];`	Lots of strings holding filenames
`FILE *mass_fp,*QBC_fp,*h_fp,*op_fp;`	File pointers for the above

1.9.3 2D raster arrays (xsz x ysz):

`float *DEM,*H,*Qx,*Qy,*maxH;`	Pointers to rasters for DEM, water depth, x and y floodplain flows and maximum depth
`float *Manningsn=NULL;`	Manning's n on floodplain
`int *ChanMask;`	xsz x ysz raster showing position of channel in floodplain, -1 for a floodplain cell and giving the position along the channel (0 at upstream end) for a channel cell.

1-D RASTERS (chsz):

`float *Chandx,*Shalf;`	chsz array, distance between channel cells and bed slope
`float *Chainage,*A,*NewA;`	Chainage along channel, flow cross sectional area in channel, new area (as produced by ChannelQ)
`float *ChanWidth,*ChanN;`	Channel width and Manning's n
`int *ChanX,*ChanY;`	x,y position of channel cell in DEM raster
`float *Weir_hc,*Weir_Cd;`	Crest height, discharge coefficient of weirs
`float *Weir_m,*Weir_w;`	Modulus, width available for flow
`int *Weir_Identx,*Weir_Identy;`	!=-1 for weir link between cells

1.9.4 Boundary condition variables

`int *BC_Ident;`	Identifier for each boundary cell. Starts at tl corner, facing north, proceeds clockwise, giving total of 2*xsz+2*ysz members. 0 CLOSED Zero flux

	1 FREE Uniform flow 2 HFIX Imposed height 3 HVAR Time varying h 4 QFIX Imposed flow 5 QVAR Varying flow
float *BC_Val;	Either imposed static values (HFIX or QFIX boundary) or location of time varying values in BCVarlist (HVAR or QVAR)
float **BCVarlist;	Array of pointers to array holding a time varying boundary condition with a value for each iteration. These are produced in LoadBCVar at the start of the simulation.
char *BC_Name;	Name in bdy and bci file associated with HVAR or QVAR boundary
float *Q_Val,**QVarlist; int *Q_Ident; char *Q_Name;	As above, but for flows imposed on channel (chsz rasters).

2 Example applications

2.1 Fluvial applications

2.1.1 River Thames at Buscot Weir

This test site is located on the upper Thames in Oxfordshire, UK, where the river has a bankful discharge of 40 m^3s^{-1} and drains a catchment of 1000 km^2. A short (~5 km along channel) test reach has been identified, bounded upstream by a gauged weir at Buscot (which provides the model boundary condition), and with reasonably well confined flows at the downstream end. The model topography was parameterised with a 50 m resolution stereophotogrammetric DEM (76x48 cells) with a vertical accuracy of ±25 cm, and channel information obtained from large scale UK Environment Agency maps and surveys.

In December 1992 a 1-in-5 year flood event occurred, with a peak discharge of 76 m^3s^{-1}, resulting in considerable floodplain inundation along the reach. The flood event coincided with an overpass of the ERS-1 remote sensing satellite, which acquired a SAR (synthetic aperture radar) image of the flood. This provided a map of inundation extent with boundaries accurate to ±50 m (Horritt *et al.*, 2001) approximately 20 hours after the hydrograph peak, but with discharge still high at 73 m^3s^{-1}. The broadness of the hydrograph, along with the short length of the reach, means that a dynamic model is unnecessary, and a steady state simulation is instead used with discharge corresponding to the flow at the time of the SAR overpass

2.1.2 River Ure at Boroughbridge, North Yorkshire

This test data set was provided by Halcrow Ltd (contact Jon Wicks). It consists of ~10 km of main channel within a domain 8 km x 4 km. Topography data are derived from airborne laser altimeter data made available by the UK Environment Agency. This data has been degraded to a 10 m resolution raster coverage and has vertical rms errors of the order of 15 cm. For this reach we simulate a dynamic floodwave hydrograph rising as shown in Figure 3 from an initial condition of a completely dry domain.

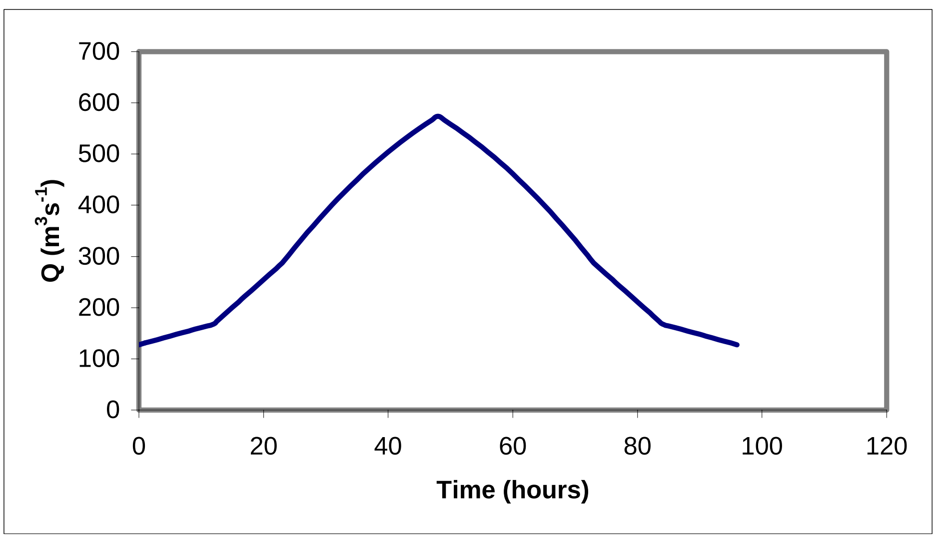

Figure 3: Inflow hydrograph for the LISFLOOD-FP application to the River Ure at Boroughbridge, North Yorkshire.

2.2 Plain flooding applications

2.2.1 Overtopping of coastal defences, Wyre district council

This application involves the simulation of flooding across a coastal plain following overtopping of sea defences. The data were made available by Halcrow Ltd. (contact Richard Mocke) and consist of an airborne laser altimeter data set at 2 m raster resolution describing the topography and dynamic inflow rates across particular identified section of the coastal defences. For model validation an observed maximum inundation shoreline is also available.

The raw DEM covers an area 8 km by 12 km and consisting of 24 million cells. It is noticeable that the vegetation removal algorithm applied to this data has not be wholly successful and numerous artefacts can be identified within the DEM which are likely to be trees on the coastal plain. Much of this domain is not inundated by the flood and therefore the first step in modelling was to crop the DEM to the area of interest. This covered a region 2.4 x 6.3 km with lower left co-ordinates at (331216, 441985) and with the western and northern boundaries lying approximately on top of the sea defences that were overtopped. Hence we can assign a mass flux to segments along this boundary. As a second step the DEM was degraded to a lower resolution to facilitate rapid model development and because it was likely that a 2 m grid would be unnecessary to replicate the observed shoreline. The cropped DEM was therefore degraded to 10, 25 and 50 m resolution grids of which we here develop an application with the 50 m model. This has 48 x 126 or 6048 cells and therefore presents a much more tractable computational problem. Inflow along various boundary segments was mapped by Halcrow and these are identified in the `wyre.bci` file, with the time varying inflows prescribed in the `wyr50mmulti.bdy` file. The simulation commences just before overtopping starts and continues for 4500 time steps of 10 s duration (or for 12.5 hours). Overtopping occurs for the first ~3 hours of the simulation, and then a ~9 hours recession period is simulated. Simulations run in just a few seconds and produce a reasonable match to the observed shoreline. The simulated flooded area contains a number of single dry pixels which occur as a result of the vegetation elements being misidentified in the DEM as part of the ground surface.

3 Technical note

The following section briefly describes the technical basis of the LISFLOOD-FP code. For further information the reader should consult Bates and De Roo, (2000), Horritt and Bates (2001a and b) and Hunter *et al.* (in press)

Channel flow is handled using a one-dimensional kinematic approach that is capable of capturing the downstream propagation of a floodwave and the response of flow to free surface slope, which can be described in terms of continuity and momentum equations as:

$$\frac{\partial Q}{\partial x} + \frac{\partial A}{\partial t} = q \qquad [3]$$

$$S_0 - \frac{n^2 P^{4/3} Q^2}{A^{10/3}} = 0 \qquad [4]$$

Q is the volumetric flow rate in the channel, A the cross sectional area of the flow, q the flow into the channel from other sources (i.e. from the floodplain or possibly tributary channels), S_0 the down-slope of the bed, n the Manning's coefficient of friction, P the wetted perimeter of the flow, and h the flow depth. We assume the channel to be wide and shallow, so the wetted perimeter is approximated by the channel width. Equations 3 and 4 are discretised using finite differences and an explicit scheme for the time dependence, and the resulting non-linear system is solved using the Newton-Raphson scheme. Sufficient boundary conditions are provided by an imposed flow at the upstream end of the reach. The channel parameters required to run the model are its width, bed slope, depth (for linking to floodplain flows) and Manning's n value and these can be varied spatially along the reach. Channel surveys provide the bed elevation profile and down reach slope, which for a kinematic channel approximation must be everywhere negative. The effective Manning's n roughness for the channel at the grid scale is left as a calibration parameter.

Each channel is discretized as a single vector along its centreline separate from the overlying floodplain raster grid. The channel thus occupies no floodplain pixels, but instead represents an extra flow path between pixels lying over the channel. Thus floodplain pixels lying over the channel have two water depths associated with them: one for the channel and one for the floodplain itself. The channel interacts with the floodplain via a Manning type flow equation (as in equation 6), allowing water to flow between channel and floodplain nodes which lie over the channel. This new scheme (referred to as the *Near Channel Floodplain Storage,* or NCFS, model by Horritt and Bates, 2001b) has proved more suitable for situations where large floodplain grid spacings are used in conjunction with a narrow (width < Δx) channel since channel width and raster grid size are decoupled. Since the NCFS scheme will also calculate floodplain flows between cells occupied by the channel, extra flow routing in near channel regions will also be represented. Along each channel vector the required channel parameters are the width, Manning's n value and bed elevation. The latter gives the bed slope and also the bankfull depth when the channel vector is combined with the floodplain Digital Elevation Model (DEM). Each channel parameter can be specified at each point along the vector and the model linearly interpolates between these. This interpolated channel is then used to identify cells in the overlying floodplain grid which have a channel lying beneath them. The only constraint on this procedure relates to the bed elevation profile. As with other channel parameters, this can have a gradient which varies along the reach, and which may also become positive (i.e. trend upwards) if the diffusive wave model is used. However, use of the kinematic wave approximation requires that the down reach slope must be everywhere negative.

Floodplain flows are similarly described in terms of continuity and momentum equations, discretized over a grid of square cells which allows the model to represent 2-D dynamic flow fields on the floodplain. We assume that the flow between two cells is simply a function of the free surface height difference between those cells (Estrela and Quintas, 1994):

$$\frac{dh^{i,j}}{dt} = \frac{Q_x^{i-1,j} - Q_x^{i,j} + Q_y^{i,j-1} - Q_y^{i,j}}{\Delta x \Delta y} \quad [5]$$

$$Q_x^{i,j} = \frac{h_{flow}^{5/3}}{n} \left(\frac{h^{i-1,j} - h^{i,j}}{\Delta x} \right)^{1/2} \Delta y \quad [6]$$

where $h^{i,j}$ is the water free surface height at the node (i,j), Δx and Δy are the cell dimensions, n is the effective grid scale Manning's friction coefficient for the floodplain, and Q_x and Q_y describe the volumetric flow rates between floodplain cells. Q_y is defined analogously to equation 6. The flow depth, h_{flow}, represents the depth through which water can flow between two cells, and is defined as the difference between the highest water free surface in the two cells and the highest bed elevation (this definition has been found to give sensible results for both wetting cells and for flows linking floodplain and channel cells.) While this approach does not accurately represent diffusive wave propagation on the floodplain, due to the decoupling of the *x*- and *y*- components of the flow, it is computationally simple and has been shown to give very similar results to a more accurate finite difference discretisation of the diffusive wave equation (Horritt and Bates, 2001a).

Equation 6 is also used to calculate flows between floodplain and channel cells, allowing floodplain cell depths to be updated using equation 5 in response to flow from the channel. These flows are also used as the source term in equation 3, effecting the linkage of channel and floodplain flows. Thus only mass transfer between channel and floodplain is represented in the model, and this is assumed to be dependent only on relative water surface elevations. While this neglects effects such as channel-floodplain momentum transfer and the effects of advection and secondary circulation on mass transfer, it is the simplest approach to the coupling problem and should reproduce the dominant behaviour of the real system.

The model time step is set by the user, however too large a time step was found to result in 'chequerboard' oscillations in the solution which rapidly spread and amplify, rendering the simulation useless. Ironically, these oscillations occur most readily in areas with low free surface gradients, where we might expect obtaining a solution to be easiest. For this reason, a flow limiter is required in order to prevent instabilities in areas of very deep water, by setting a maximum flow between cells. This flow limit is fixed so as to prevent 'over' or 'undershoot' of the solution, and is a function of flow depth, grid cell size and time step:

$$Q_x^{i,j} = \min\left(Q_x^{i,j}, \frac{\Delta x \, \Delta y \left(h^{i,j} - h^{i-1,j} \right)}{4 \, \Delta t} \right) \quad [7]$$

This value is determined by considering the change in depth of a cell, and ensuring it is not large enough to reverse the flow in or out of the cell at the next time step. This limiter replaces fluxes calculated using Manning's equation with values dependent on model parameters, and hence when the flow limiter is in use floodplain flows are sensitive to grid cell size and time step, and insensitive to Manning's *n*.

In order to overcome these problems, Hunter *et al.* (in press) have recently proposed a modified version of the LISFLOOD-FP based on adaptive time stepping. This functionality is available in LISFLOOD-FP version 2.0 and onwards. The approach seeks remove the need to invoke the flow limiter (equation 7) by finding the optimum time step (large enough for computational efficiency, small enough for stability) at each iteration. Stability depends on water depth, free surface gradients, Manning's *n* and grid cell size and thus varies in time and space during a simulation.

This method uses an analysis of the governing equations and their analogy to a diffusion system to calculate the largest stable time step. Equations (5) and (6) are essentially discretizations of the continuity and momentum equations:

$$\frac{\partial h}{\partial t} + \nabla \cdot \mathbf{q} = 0 \quad [8]$$

$$q_x = \pm \frac{h_{\text{flow}}^{5/3}}{n} \left| \frac{\partial h}{\partial x} \right|^{1/2}, \qquad q_y = \pm \frac{h_{\text{flow}}^{5/3}}{n} \left| \frac{\partial h}{\partial y} \right|^{1/2} \qquad [9]$$

where q_x and q_y are components of the flow per unit width. Equation (9) differs from the usual definition of Manning's equation in 2-D shallow water models in that the two components are decoupled, but this has been found to have negligible effect on model predictions (Horritt & Bates, 2001a). The sense of the flow is determined by whether the free surface gradient is positive or negative. Combining equations (8) and (9) we obtain:

$$\frac{\partial h}{\partial t} \underbrace{- \frac{h_{\text{flow}}^{5/3}}{2\,n} \left| \frac{\partial h}{\partial x} \right|^{-1/2} \frac{\partial^2 h}{\partial x^2} - \frac{h_{\text{flow}}^{5/3}}{2\,n} \left| \frac{\partial h}{\partial y} \right|^{-1/2} \frac{\partial^2 h}{\partial y^2}}_{\text{Diffusion Terms}} \pm$$

$$\frac{5\,h_{\text{flow}}^{2/3}}{3\,n} \left| \frac{\partial h}{\partial x} \right|^{1/2} \frac{\partial h_{\text{flow}}}{\partial x} \pm \frac{5\,h_{\text{flow}}^{2/3}}{3\,n} \left| \frac{\partial h}{\partial y} \right|^{1/2} \frac{\partial h_{\text{flow}}}{\partial y} = 0 \qquad [10]$$

The terms with the second spatial derivatives make up the diffusion part of the equation, and will dominate when free surface gradients are small and stability problems are likely to arise. The solution is unlikely to mirror the behaviour of classical diffusion problems since the diffusion coefficient varies in space and time, and is anisotropic, but we can use the analogy to estimate the most efficient time step. For the diffusion equation:

$$\frac{\partial h}{\partial t} - \alpha \left(\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2} \right) = 0 \qquad [11]$$

and its explicit discrete counterpart on a square grid (subscripts are spatial grid locations, superscripts time):

$$\frac{h_{i,j}^{t+1} - h_{i,j}^{t}}{\Delta t} - \frac{\alpha}{\Delta x^2} \left(h_{i+1,j}^{t} + h_{i-1,j}^{t} + h_{i,j+1}^{t} + h_{i,j-1}^{t} - 4\,h_{i,j}^{t} \right) = 0 \qquad [12]$$

a von Neumann stability analysis produces the following time step condition:

$$\Delta t \le \frac{\Delta x^2}{4\,\alpha} \qquad [13]$$

At equality, the $h_{i,j}^{t}$ terms in equation (11) cancel, and it becomes the well known Jacobi relaxation approach to the solution of Laplace's equation, where the value at a node is iteratively replaced by the mean of neighbouring values. This would imply that an optimal time step for the hydraulic model at a specific location is given by:

$$\Delta t = \frac{\Delta x^2}{4} \min \left(\frac{2\,n}{h_{\text{flow}}^{5/3}} \left| \frac{\partial h}{\partial x} \right|^{1/2}, \frac{2\,n}{h_{\text{flow}}^{5/3}} \left| \frac{\partial h}{\partial y} \right|^{1/2} \right) \qquad [14]$$

We thus arrive at an expression for the time step similar in form to that used by Werner & Lambert (in press) but larger by a factor of 2. In Werner & Lambert (in press) the time step was set to allow small chequerboard oscillations to decay down to a flat free surface, whereas in this analysis we counter the build up of these oscillations directly, and hence can use a larger time step. A scheme that uses this criterion can be implemented by searching the domain for the minimum time step value and using this to update h. The time step will thus be adaptive and change during the course of a simulation, but is fixed in space at each time step.

A problem with this approach is that there is no lower bound on the time step. As free surface gradients tend to zero (standing water), α tends to infinity and hence the time step also tends to zero. Furthermore, as flow reverses during the transition between the wetting and drying phases, the time step is driven to zero, causing the model to 'stall'. For a fully dynamic model, some way of dealing with this pathological behaviour as surface gradients tend to zero is required. This is avoided by introducing a linear scheme that is applied to cells where free surface elevations in neighbouring cells differ by less than a specified threshold, h_{lin} (Cunge *et al.*, 1980). The flow equation then becomes:

$$q_x = \frac{h_{\mathrm{flow}}^{5/3}}{n}\left(\frac{\Delta x}{h_{\mathrm{lin}}}\right)^{1/2}\left(\frac{\partial h}{\partial x}\right) \qquad [15]$$

with a similar expression for q_y. For cells where this linearised flow equation is applied, an equation similar to equation (14) above is used to determine the time step.

Hunter *et al.* (in press) tested this new adaptive time step scheme against analytical solutions for wave propagation over flat and planar slopes and showed a considerable improvement over the classical fixed time-step version of the model. Moreover, the scheme was shown to yield results that were independent of grid size or choice of initial time step and which showed an intuitively correct sensitivity to floodplain friction over spatially-complex topography.