

LISFLOOD-FP

User manual and technical note

**Code release**

Paul Bates[[1]](#footnote-1), Tim Fewtrell, Mark Trigg and Jeff Neal

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

25th April 2008

Document information

|  |  |
| --- | --- |
| Project title | LISFLOOD-FP shareware version |
| Document title |  |
| Code release |  |
| Prepared by | Paul Bates, Tim Fewtrell and Jeff Neal |
| Checked by | Mark Trigg |
| Code contributions | Paul Bates, Matt Horritt, Matt Wilson, Neil Hunter, Tim Fewtrell, Mark Trigg and Jeff Neal |
| Document version | 1.2 |
| Issue date and time | 26/09/2012 16:40 |
| Original file stored at | <http://source.ggy.bris.ac.uk/websvn/> |

Disclaimer

Before using the LISFLOOD-FP software (hereafter “the Software”) please read carefully the following terms for use. Extracting files from the LISFLOOD-FP.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 damage 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 . 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 or diffusive approximations to the one-dimensional St. Venant equations to simulate the passage of a flood wave along a channel reach. Once bankfull 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.

Major Version History

Ver Date Details

---- ------- ----------------------------------------------------------------------------------------------------

4.3 Sep 04 2009 Dynamic & diffusive steady state 1D solution added & tested by Tim Fewtrell

4.1 Nov 10 2008 TRENT solver added but not tested. Integrated version tested by Jeff Neal

3.6 Jul 31 2008 Decouple river channel timestep from floodplain timestep by Mark Trigg

3.5 Jun 13 2008 OpenMP version implemented and tested on Bustcot by Jeff Neal

3.4 Apr 21 2008 Double precision version by Mark Trigg

3.3 Jan 11 2008 Diffusive channel solver & Bug fixed branching channels by Mark Trigg

3.1 Oct 08 2007 Fully tested and bug fixed modular code by Mark Trigg

3.0 May 25 2006 Modularised the code and added porosity scaling algorithm by Tim Fewtrell

2.7 Feb 25 2005 Evaporation and Infiltration added by Matt Wilson

2.6 Dec 20 2004 Add more output file and command line options by Matt Wilson

2.5 Nov 25 2004 Checkpointing functionality added by Matt Wilson

2.0 Jun 08 2004 Adaptive timestep implemented by Neil Hunter

1.0 2003 First public release version by Matt Horritt

0.9 2003 Increase output file and command line options by Matt Wilson

0.8 2001 Prototype C++ version created by Matt Horritt

0.5 2001 Original version created by Paul Bates and Ad De Roo

What’s new in version ?

The LISFLOOD-FP model has undergone substantial changes since the last revision of the user manual for code version 2.6.2 in 2005. 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 improvements to the model functionality over this period. The most significant of these changes have been:

Version 4.3.6

* Fully tested and working diffusive channel solver now implemented. Set with “diffusive” in par file. New downstream boundary conditions for channel diffusive solver HFIX, HVAR and FREE with downstream slope or user supplied slope.
* “ch\_dynamic” and “-dynsw” flags in parameter file. Both flags enable the use of the full 1D de St Venant equations for the steady state initial calculation for the 1D channel solver. Incorporated mainly for possible forward compatibility and useful for very shallow flow situations.
* Recommendation to always use “startq” option in diffusive mode to initialise model with steady state diffusive solution.
* OpenMP support added to the floodplain solver and to any loop around all the floodplain cells. Use of newer compilers with this option will also auto-parallelise portions of the 1D channel solver. See Neal *et al.* (2009) for details of implementation and testing.
* Decoupled channel and floodplain timesteps to speed up model computation for grids with gradually varying rivers and high grid resolutions/areas of deep water. Activated by “ts\_multiple” followed by the value in the parameter file. Tests show results up to 10x are stable but 100x shows considerable loss of accuracy.
* Bug fix for boundary flows that caused the use of the linearised form of Manning’s on the floodplain at the boundary and ended up with very small depths. An HFIX or HVAR boundary on the floodplain will be initiated once water has reached that particular cell from the rest of the floodplain.
* “qlimfact” added to allow user to specify a relaxation factor for Qlimit such that more water can move between floodplain cells. When using the option, ensure that the resulting model solution is still stable and believable.
* Checkpointing functionality has now been improved with added checks for LISFLOOD-FP and checkpoint version numbers to ensure compatibility.
* Added some code to make river channel chainage independent of cell size and thus uses the straightline chainage between entered cross-sections rather than that derived from the DEM. This option can be turned off using “chainageoff” in the parameter file.

Version 3.4.0

* Fully tested and working diffusive channel solver now implemented. Set with “diffusive” in par file. New downstream boundary conditions for channel diffusive solver HFIX, HVAR.
* All calculations are now in double precision which reduces the mass error by an order of magnitude, particularly for smaller grid sizes and timesteps. No speed penalty as compilers’ and cpus’ native precision is now double.
* New "-log" command line option - redirects screen output to a log file. New par file option htol. New options in par file, "ch\_start\_h" and “startq”, to set starting water depth in channel.
* More detailed river channel profile output. Output of fluxes now with offset asciiheaders to locate output at boundaries.
* New "time to complete run" estimate line (in minutes) output to screen at every save write interval.
* Debug parameter and command-line now outputs 3 files; the final dem after burning in the channel and bank mods (\*.dem), the channel mask (\*.chmask) and the channel segment mask (\*.segmask).
* Numerous bug fixes in branched channel code. Hds in mass file is now from main channel, not last one calculated. Now adds only main channel output to boundary flux instead of all channel outputs. Trib connections were out by one cell resulted in incorrect slope. Where more than one trib on one channel, outflows are now assigned to the correct locations. Slope mod for kinematic solver which set -ve slopes to very small slope resulting in very high water levels replaced with simple positive slope of same magnitude giving more reasonable (but still wrong!) results (added -ve slope warning to user).
* Checkpoint function no longer overwrites previous results files when restarting and new mass file output lines are appended with a clear break point.
* Fixed porosity technique implementation to scale both volume and flux to maintain mass conservation. Test against fully blocked cell provides identical solutions. Fixed minor bug in FloodedArea calculation introduced by porosity scaling.
* Boundary condition bug which stops water exiting north and west under FREE conditions and bug which displaces south and west FREE boundaries by one cell is now fixed. Bug in boundary flux calc for north boundary (added wrong way) now fixed
* Fixed bug in counter increment for profiles output. Counter was incrementing before profiles output completed. Bug fixed in the storage and output of Qlim and Trec arrays. Bug fix for domains without a channel. Now correctly assigns outflows when ChannelPresent=OFF.

Version 3.1.1

* The code has been ported into a version control system based on an open source software called Subversion (see <http://subversion.tigris.org/>) and now resides on a secure, password protected server located at the School of Geographical Sciences at the University of Bristol. The benefits of this are that: (1) it allows multiple developers to work on the same code at the same time without generating conflicts; (2) it prevents multiple working versions of the code by allowing developers to get immediate access to latest code versions and at the same time easily commit their main changes to the code reference version; (3) it captures documentation on code changes and logs these and (4) it allows to easily “roll back” code changes to recover a previous version. Use of Subversion commenced on 11th May 2006 and since this time there have been 21 separate bug fixes and minor improvements. Subversion also delivers an RSS feed to alert developers to code changes by others in the development team. For details on using Subversion see <http://source.ggy.bris.ac.uk/wiki/Subversion_server>.
* Time variable evaporation has been added. This is switched on using the keyword “evaporation” in the parameter file followed by the filename of the evaporation data to be read in and used in the calculations.
* The time of initial inundation counter can be reset at a specified time by using the keyword resettimeinit in the parameter file, followed by the time in seconds. This is useful for calculating time of inundation of a specific event within a multi-event sequence.
* Profiles of channel water surface elevation can now also be written at each save interval (as opposed to only at the time of a satellite overpass) using the keyword profiles in the parameter file.
* Date and time stamps have been added at the start and end of runs to enable better tracking of jobs.
* Optional raster output for the flux values in the x and y Cartesian directions have been added. These are turned on using the keyword qoutput in the parameter file and will cause the program to output ascii raster grids with file extensions .Qx and .Qy, at the saveint file output interval. By default these output flags are turned off.
* Optional raster output for the per cell optimal time step (equation [14]) and flow limiter (equation [8]) values calculated during adaptive time stepping have been added. These are turned on using the parameter file keywords touput and qloutput and will cause the program to output ascii raster grids with file extensions .Tx, .Ty, .QLx and .QLy at the saveint file output interval. By default these output flags are turned off.
* The file extension for ascii raster grids of water depth has been changed to .wd for clarity and to allow easier analysis of multiple files.
* A facility has been added to output water depth time series at particular x,y locations.
* Various new command line and parameter file options have been added.
* Experimental porosity techniques for urban flood modelling have been added.
* Code has been restructured to be fully modularised and extensive testing carried out, including against a new analytical test case for channel flow.

Contents

Document information 2

Disclaimer 3

Executive summary 4

Major Version History 5

What’s new in version 4.3.6? 6

Contents 9

List of Tables 12

1 User manual 13

1.1 Introduction 13

1.1.1 Development background 13

1.1.2 Previous studies with LISFLOOD-FP 14

1.1.2.1 Calibration, validation and benchmarking studies 14

1.1.2.2 Scaling behaviour 15

1.1.2.3 Uncertainty analysis 16

1.1.2.4 Adaptive time stepping 17

1.1.3 Model assumptions and key limitations 17

1.2 Installation guide 18

1.2.1 Installation on a UNIX or LINUX system 18

1.2.2 Installation on a Windows Vista, XP, NT, 98 or 2000 system 19

1.2.3 Minimum hardware requirements for Windows systems 19

1.3 Data requirements, input files and file formats 20

1.3.1 Data requirements 20

1.3.2 Input file formats 21

1.3.2.1 Parameter file (.par) 21

1.3.2.2 Channel information file (.river) 24

1.3.2.3 Boundary condition type file (.bci) 27

1.3.2.4 Time varying boundary conditions file (.bdy) 28

1.3.2.5 Digital Elevation Model file (.dem.ascii) 28

1.3.2.6 Floodplain friction coefficient file (.n.ascii) 29

1.3.2.7 Weir cell linkage specification file (.weir) 29

1.3.2.8 Multiple overpass file (.opts) 30

1.3.2.9 Stage output data file (.stage) 30

1.3.2.10 Evaporation data file (.evap) 30

1.3.2.11 Alternative ascii header file (.head) 31

1.3.2.12 Virtual gauge output data file (.gauge) 31

1.4 Setting up a simulation 31

1.5 Running a simulation 32

1.5.1 Checkpointing 33

1.5.2 Output file formats 34

1.5.2.1 Mass balance output file (.mass) 34

1.5.2.2 Water depths at time of satellite overpass file (.op) 35

1.5.2.3 Channel water surface profile (.profile) 35

1.5.2.4 Synoptic water depth, water surface elevation and flow flux files (-xxxx.wd and -xxxx.elev) 35

1.5.2.5 Maximum water surface elevation file (.mxe) and maximum water depth (.max) 36

1.5.2.6 Time of initial inundation (.inittm), time of maximum depth (.maxtm) and total time of inundation (.totaltm) 36

1.5.2.7 Flux values (-xxxx.Qx and -xxxx.Qy) 36

1.5.2.8 Adaptive time step (-xxxx.Tx and -xxxx.Ty) and flow limiter (-xxxx.QLx and -xxxx.QLy) values 36

1.5.3 Visualising model results 36

1.6 References and bibliography 36

1.7 Language standards 39

1.8 Programme structure and subroutine map 39

1.9 Key variable names 40

1.9.1 General variables: 40

1.9.2 Character variables 41

1.9.3 2D raster arrays (xsz x ysz): 41

1.9.4 Boundary condition variables 41

2 Example applications 43

2.1 Fluvial applications 43

2.1.1 River Thames at Buscot Weir 43

2.1.2 River Ure at Boroughbridge, North Yorkshire 43

2.2 Plain flooding applications 44

2.2.1 Overtopping of coastal defences, Wyre district council 44

3 Technical note 45

List of Figures

Figure 1: Plot of the F performance measure over the parameter space for the River Thames model. 16

Figure 2: Probability map of predicted inundation,  , 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. 18

Figure 3: Inflow hydrograph for the LISFLOOD-FP application to the River Ure at Boroughbridge, North Yorkshire. 41

List of Tables

Table 1: Summary of LISFLOOD-FP calibration, validation and benchmarking studies for fluvial application. 14

Table 2: Summary of LISFLOOD-FP calibration and validation studies for coastal application. 15

Table 3: Files deployed from the LISFLOOD-FP.zip archive. 18

Table 4: Input data required by the LISFLOOD-FP model. 20

Table 5: Items read in through the parameter (.par) file. 23

Table 6: Types of boundary condition available in the .bci file. 27

Table 7: Command line options for LISFLOOD-FP. 31

# User manual

## 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 or diffusive wave equation representation of channel flow coupled to a 2D flood spreading model for floodplain flow. Channel and floodplain topography is discretised 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).

### 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.

### 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.

#### 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:

 []

Where Aobs and Amod 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 : 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 : 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.

#### 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.

#### 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 ). 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:

 []

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 to give a generalised relative risk measure for each pixel (see Figure 2), *fij* takes a value of 1 for a flooded pixel and is zero otherwise and  is the global performance measure for simulation *j*.  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 : Plot of the F performance measure over the parameter space for the River Thames model.

#### 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.

### 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 or diffusive 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 discretised 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 m3s-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 : Probability map of predicted inundation,  , 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.

## Installation guide

The model files are provided as a WinZip archive LISFLOOD-FP.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 overpasses |

Table : Files deployed from the LISFLOOD-FP.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.

### 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:

g++ LISFLOOD-FP.cpp -o LISFLOOD-FP -lm -O3

to generate an executable that is speed optimized, or:

g++ LISFLOOD-FP.cpp -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 although this has not been fully tested.

The newer modular versions (3.0 and later) of LISFLOOD-FP comes with a makefile to compile on Linux. Versions from 3.5 onwards are enabled with OpenMP support. In order to compile with OpenMP support use the following compilation flags:

-parallel -openmp

### Installation on a Windows Vista, XP, 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, Visual c++ and Intel c++ compilers, and of these our current preference is for the Intel software because of its speed. All these software then generate an executable file LISFLOOD-FP.exe that can be run from a DOS prompt. A pre-compiled executable generated in this way using the Intel c++ compiler is provided in the LISFLOOD-FP.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.

### 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 PC with 3 Ghz processor clock speed, 1 Gb of Ram and 100 Gb hard disk.

## Data requirements, input files and file formats

### Data requirements

Model data requirements are outlined in .

|  |  |  |
| --- | --- | --- |
| **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). | *Nc* typically between 0.01 and 0.05  *Nfp* 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. | Varies between applications but typical values are in the range 2-20 s. |
|  | *Adaptive time step version*  Optimum time step to maintain stability is calculated by the code | Calculated by the code. Optimum time step reduces quadratically with grid size. May result in substantial increase in computational cost for fine grids |

Table : 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.

### 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.

#### 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.wd 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)  *Acceleration and Adaptive time step model*  Initial guess for the optimum time step and maximum possible time step. | 10.0 |
| saveint | Interval in seconds at which results files are saved. Note each file is saved with a sequential number stamp, eg results-0001.wd | 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 1 hour is used. When the model starts it automatically looks for and reads in the default file named *“resroot”*.chkpnt in the directory from which the model was executed, unless the loadcheck keyword with alternative filename is used. The user needs to delete the .chkpnt or turn off this option to commence the simulation again from the beginning. | 0.0001 |
| loadcheck | Name of an alternative file used to start the checkpointing. By default, the program uses a single file which is overwritten at the checkpointing interval. This alternative start file allows you to start from a file that does not get overwritten by the checkpoint function. |  |
| 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 (depth \*.op and channel water surface profile \*.profile) 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. |
| porfile | Name of file containing porosity details to apply to model. There are currently 4 different methods implemented. Please see Tim Fewtrell’s Porosity manual for full details. Note - while the code for this options works fine, the methodology is still at research stage. | 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. This should be a water depth file. | Res.old. Not included with Buscot test case. Commented out in .par file. |
| evaporation | Name of file containing evaporation data. | Not included with Buscot test case. Not present 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 suppressed. |
| adaptoff | Logical keyword to suppress adaptive time stepping algorithm | Not commented so value is “yes”. Fixed time step model is used instead. |
| resettimeinit | Resets the time of initial inundation counter to zero at a specified time by the user. The keyword should be followed by the time in seconds at which the reset should take place. | Not used in the Buscot test case so does not appear in the .par file. |
| profiles | Keyword which forces the model to write out the channel water surface profile at each saveint. | Not used in the Buscot test case so does not appear in the .par file |
| qoutput | Keyword which forces the model to write out ascii raster grids of the flux values in the x and y Cartesian directions. Grids are output at each saveint. | Not used in the Buscot test case so does not appear in the .par file |
| voutput | Keyword which forces the model to write out ascii raster grids of the velocity values in the x and y Cartesian directions. Grids are output at each saveint. | Not used in the Buscot test case so does not appear in the .par file |
| toutput | Keyword which forces the model to write out ascii raster grids of the per cell optimum time step values calculated by the adaptive time stepping routine using equation [14]. Grids are output at each saveint and separate values are calculated for the x and y Cartesian directions. | Not used in the Buscot test case so does not appear in the .par file |
| qloutput | Keyword which forces the model to write out ascii raster grids of the per cell flow limiter values calculated by the adaptive time stepping routine using equation [8]. Grids are output at each saveint and separate values are calculated for the x and y Cartesian directions. | Not used in the Buscot test case so does not appear in the .par file |
| ascheader | Name of file containing alternative header information for output of ascii raster grids. Useful for switching to lat/long format. | Not used in the Buscot test case so does not appear in the .par file |
| debug | **O**utputs 3 files; the final dem after burning in the channel and bank mods (\*.dem), the channel mask (\*.chmask) and the channel segment mask (\*.segmask). | Not used in the Buscot test case so does not appear in the .par file |
| ch\_start\_h | By default, the channel solver will start with a water depth of 2m for the whole channel. The user can override this by using this option and a value. This can speedup the spinup time of the model. | Not used in the Buscot test case so does not appear in the .par file |
| startq | In kinematic mode, the model will calculate a water level for each section given the inflow at the top of the reach. In diffusive mode, the model will iterate to the initial steady state solution given a downstream boundary condition and an upstream inflow. Will dramatically decrease spin up time for complex channels. See “ch\_dyanmic” below for more details | Not used in the Buscot test case so does not appear in the .par file |
| diffusive | LISFLOOD-FP by default uses the kinematic solver for the river channel unless this is present in the .par file. | Not used in the Buscot test case so does not appear in the .par file |
| htol | Optional parameter to override default 1m bank smoothing. | Not used in the Buscot test case so does not appear in the .par file |
| ch\_dynamic | startq will automatically use the diffusive steady state solution in diffusive mode. Use this keyword to activate full dynamic steady state initial condition. Mainly incorporated for forward compatibility and very complex channel systems. Can only be used in conjunction with “startq” | Not used in the Buscot test case so does not appear in the .par file |
| dhlin | Linearisation threshold for adaptive version. Increasing the value, reduces run time and accuracy. Default value is related to dx but calculated from a gradient of 0.0002 from Cunge et al., 1980 and Hunter et al., 2005 | Not used in the Buscot test case so does not appear in the .par file |
| depththresh | Option to change the depth at which a cell is considered wet. | Not used in the Buscot test case so does not appear in the .par file |
| mint\_hk | Keyword to allow calculation of maxH, maxHtm, totalHtm and initHtm at the mass interval rather than every time-step. Useful for parallel solutions and should decrease computation time. | Not used in the Buscot test case so does not appear in the .par file |
| ts\_multiple | Decouples the channel and floodplain time step. Enter a value after the keyword to invoke more than x1. Tests show up to x10 gives almost identical results to x1. If used, check sensitivity of results. | Not used in the Buscot test case so does not appear in the .par file |
| chainageoff | Default code now makes river channel chainage independent of cellsize and uses straight line distance between entered sections. Use this keyword to revert to the old calculation which used cell dx dimensions. | Not used in the Buscot test case so does not appear in the .par file |
| comp\_out | Keyword to initiate model time/computation time ratio output to standard out buffer. Details in section 1.5 below. | Not used in the Buscot test case so does not appear in the .par file |

Table : 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.

#### 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 should be negative (or positive downhill) (ie. the channel bed should not increase in elevation in the downstream direction). LISFLOOD-FP will allow uphill slopes for the kinematic solver, but just pretend they are downhill and give a warning. The diffusive solver can handle uphill slopes so no warnings are issued.

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: X1 Y1 Width1 n1 Bed elevation1 BC Value

Line 4: X2 Y2 Width2 n2 Bed elevation2 Lateral inflow2

Line 5: X3 Y3 Width3 n3 Bed elevation3

etc…… … … … … …

Line i: Xi Yi Widthi ni Bed elevationi

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 m3s-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 73m3s-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: X1 Y1 Width1 n1 Bed elevation1 BC Value

Line 3: X2 Y2 Width2 n2 Bed elevation2 Lateral inflow2

Line 4: X3 Y3 Width3 n3 Bed elevation3

etc…… … … … … …

Line i: Xi Yi Widthi ni Bed elevationi 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

**Downstream Boundary Conditions for the Diffusive Channel Solver**

Unlike the kinematic solver, the diffusive channel solver requires a downstream boundary condition. For tributaries this is handled automatically by LISFLOOD-FP, which uses the water level from the downstream receiving channel. However, for the main channel a boundary condition will have to be provided by the user – and you will be warned if it is not present. Currently there are two fully tested options for this.

* Option 1: Normal depth calculation

To use this option, use the keyword FREE to force the model to calculate the normal depth for the downstream water level. There are two options available of which the latter is considerably more stable. Option a is to allow the model to calculate the slope used for the normal depth calculation which uses the slope between the last two river sections. eg.

22950.0 -600.0 5.0 0.03 69.0

24900.0 -600.0 5.0 0.03 68.5 FREE

Option b is to specify a user determined slope which is normally taken as the overall valley slope. eg.

22950.0 -600.0 5.0 0.03 69.0

24900.0 -600.0 5.0 0.03 68.5 FREE 0.0006

* Option 2: Constant water level.

To use this option to simulate a steady state water level BC, use the keyword HFIX and the associated water ELEVATION value at the downstream end of the model in m. eg.

22950.0 -600.0 5.0 0.03 69.0

24900.0 -600.0 5.0 0.03 68.5 HFIX 38.345

* Option 3: Time-varying water level.

To use this option to simulate a dynamic flood wave BC, use the keyword HVAR and the associated value is a boundary identifier chosen by the user, e.g. downstream1. Information about the time varying boundary condition data is then held in the time varying boundary condition file (.bdy).

Note that there is currently one additional boundary setting which is in development. There is a stability issue associated with this under certain conditions, so use at your own risk. The boundary condition is RATE (rating curve) with a boundary identifier and associated table in the .bdy file.

#### 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 | Free surface or valley slope (optional) |
| 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 (m2s-1). For a boundary segment this is multiplied within the code by the length of the boundary segment to give the mass flux in m3s-1. For a point source the mass flux per unit width is multiplied by the cell width to the mass flux in m3s-1. |
| QVAR | Time varying flow into domain | Boundary identifier (e.g. upstream1) corresponding to data in the user supplied .bdy file |

Table : 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 northing co-ordinates -1200 and -1800 (i.e. on the y axis).

#### 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 ‘days’, ‘hours’ or ‘seconds’).

Line 4: Value1 Time1

Line 5: Value2 Time2

etc…. … …

Line i: Valuei Timei

Where Valuei is the value of the relevant quantity for the given boundary type. For all HVAR boundaries Valuei is a water surface elevation in metres. However, the units of Valuei 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 Valuei is given as mass flux with units m3s-1. By contrast, for a QVAR boundary specified in the .bci file Valuei is given as mass flux per unit width with units m2s-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 m3s-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.

#### 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 ARCGIS 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)

#### 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-Info ascii raster as described in section 1.3.2.5 above.

#### Weir & bridge 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.

**Weir flow equation**

Q = CL(2gH)1.5

Where:

C = Weir flow Coefficient. Default value would be 1.4

L = weir breadth across channel

H = The energy head upstream of the weir.

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; X1 Y1 Direction1 Qcoeff1 Crest height1 Modular limit1 Width1

Line 3: X2 Y2 Direction2  Qcoeff2 Crest height2 Modular limit2 Width2

etc… … … … … … … …

Line i: Xi Yi Directioni Qcoeffi Crest heighti Modular limiti Widthi

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 1.4 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 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

Weir limitations and notes:

1. Note that currently the weir calc in lisflood uses the water depth rather than energy head (thus ignoring approach velocity). This is a reasonable approximation for low Fr number hydraulics. However, you should find it reasonably easy to add the velocity/energy head if this was important to your model.
2. The flow across the cell boundary is totally controlled by the weir calculation within the subgrid channel. There is no floodplain component. This can lead to localised instabilities around the weir if there is no cells around the weir cell that can carry bypass flow. This arises as flow may be out of subgrid bank upstream of the weir (and hence on the floodplain) and then at the weir is force back in the channel and over the weir. We recommend placing a stage output location upstream and downstream of the weir in order to check for this if the weir is critical. The code could be changed to allow for the out of bank flow and this should be straightforward if you wish to do this for your model.
3. If in doubt build a simple test model of your bridge and ensure you understand how it is represented and behaving in lisflood-fp. See the testing directory 16 for examples of bridge testing setups.
4. The drowned out weir uses a slightly modified form of the weir flow equation, but this has not been tested fully and we suspect the modular limit implementation is wrong.

**BRIDGES (currently subgrid channel version only)**

Bridges can also be represented explicitly (since version 5.6.5). The aim with the lisflood-fp implementation of bridges is to allow the hydraulic effects of a bridge (abutments/deck etc.) to be represented realistically with a few simple parameters. It should be noted here that it is NOT intended as an engineering tool for detailed modelling of bridge hydraulics. Hydraulic modelling of bridges can be a complicated subject in itself and a tool such as Hecras may be a more appropriate choice for such purposes. Currently bridges have only been implemented in the subgrid channel version. However if you wished to extend the bridge functionality to normal floodplain flow cells it should be fairly straightforward. Extension of bridges to the 1d diffusive solved would be more of a challenge.

The bridge modelling method used is the pressure flow method which implements an orifice equation to calculate the flow through the bridge when the bridge deck obstructs flow. This is a widely used method for modelling bridges and is the default bridge modelling method used in Hecras (against which the lisflood-fp implementation has been tested).

**Orifice flow equation**

Q = CdA(2gH)0.5

Where:

Cd = Coefficient of discharge for a fully submerged pressure flow. Default value would be 0.8.

A = Net area of bridge opening

H = The difference between the energy gradient elevation upstream and the water surface elevation downstream.

Figure : Bridge as implemented in lisflood-fp.

W = bridge opening width

db = bridge opening depth

dus,ds = upstream, downstream depth of flow

egus = upstream energy grade depth (Vus2/2g) . Bridge approach velocity Vus is calculated from qus divided by channel area (not bridge area)

qb = bridge flow

qus,ds = upstream, downstream flow

WLus,ds = upstream, downstream water level

A = bridge open area (W x db)

H = orifice head (WLus – WLds + egus)

Zr = upstream depth to opening ratio (dus/db). Zr = 1.0 when water level is at soffit level.

The actual calculation used by lisflood-fp at a bridge location will depend upon the water level at the bridge. For water levels below the bridge soffit (Zr<1.0), the normal open channel flow method is used (using the bridge opening flow area not the channel area). For water levels well above the soffit, the orifice calculation is used. There is a transition zone between the two types of flow (roughly between Zr 1.0 and 1.5) where a weighted combination of the two flow types is used. This transition zone is notoriously difficult to model for various reasons (see Hecras manual). The approach used here is simple and robust and in tests compares well with the hecras sluice approach for this transition zone.

Like weirs, information about bridge linkages is also given in the .weir file. The file line format for bridges is as follows:

X1 Y1 Direction1 Cd1 Soffit elevation1 Transition zone1 Width1

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). When stating the direction you must put n, s, e, w (north, south ...) followed by a b for bridge.

Cd is the coefficient of discharge for a fully submerged pressure flow, typically 0.8.

Soffit elevation is the underside of the bridge deck elevation.

Transition zone is the upper end of the zone for which lisflood-fp will take a weighted mean of the open channel flow and pressure flow (the lower end of the zone has a value of 1.0 and represents the point where the water elevation is equal to the soffit elevation). Typically for a bridge this should be a value of 1.5. If the hydraulics approaching/at the bridge are particularly extreme (eg Fr>0.75) you may find extending this to a higher value eg 1.7) may provide extra stability at the expense of accuracy.

Width is the width of the bridge opening.

Bridge limitations and notes:

1. For more irregular bridges it is up to the user to distil the geometry to an appropriate simple representation that can be used in lisflood. For example, if a bridge has piers you can subtract the pier area from the bridge opening area and put the net area into lisflood-fp.
2. While a bridge is placed between two cells, in reality, a bridge must be placed in the centre of 4 contiguous cells. This is because the calculation uses the flow fluxes at the boundaries of cells 1 and 2 and cells 3 and 4 in order to calculate approach velocities and hence energy grade. It is also a good idea to ensure that the 4 cells are not part of some other process such as a boundary or confluence etc.
3. If in doubt build a simple test model of your bridge and ensure you understand how it is represented and behaving in lisflood-fp. See the testing directory 16 for examples of bridge testing setups.
4. Lisflood-fp does not take into account contraction and expansion losses before and after the bridge. This means that if your bridge width is significantly less that of the channel, then the head (afflux) upstream of the bridge constriction will be underestimated. This does not affect the pressure flow calculation, only the open channel flow calculation when water elevations are below the bridge deck.
5. There is currently no provision for overtopping of the bridge deck when water elevations upstream are very high. You can easily extend the Lisflood-fp bridge code using the weir equation for this case if you require this functionality for your model.

#### 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, with the file naming including a simple counter (beginning at 0000) to signify each overpass requested. It is important to remember that the model time that the overpass counter signifies is not the same as that of the regular file output interval counter. 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 nth overpass in seconds of simulation time

An example .opts file is given below:

4

900.0

1800.0

2700.0

3600.0

#### Stage output data file (.stage)

This file is used to specify the x,y locations of points where the user wishes the model to output a time series of water depths. This option is activated by including the keyword stagefile in the .par file and following this with the name of the .stage file to be read. For each location specified in the file the water depth value is written out at each massint interval. The format of the file is as follows:

Line 1; Number of stage points at which water depth output time series are required

Line 2; x and y locations of 1st point

Line 3: x and y locations of 2nd point

etc… … … … … … … …

Line i: x and y locations of nth point

An example .stage file is given below:

3

388869.59 233696.3

386307.41 239076.1

383681.45 245652.34

#### Evaporation data file (.evap)

This file is used to specify a time-varying evaporation rate and is read when the keyword evaporation appears in the .par file. This sink term is then applied to every model grid cell at each time step to give a spatially uniform evaporation loss over the domain. The file format is similar to the .bdy file:

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

Line 2: Number of time points at which boundary information is given followed by a keyword for the time units used (either ‘days’, ‘hours’ or ‘seconds’).

Line 3: Value1 Time1

Line 4: Value2 Time2

etc…. … …

Line i: Valuei Timei

Where Valuei is evaporation rate in mm day-1 and Timei is the time at which this value occurs in the units specified on line 2. The model then linearly interpolates these values to give the evaporation rate at each time step.

#### Alternative ascii header file (.head)

This file is used to an alternative 6 line header for all ascii raster file output by the model and is read when the keyword ascheader appears in the .par file. This is particularly useful for switching between different coordinate systems (e.g. UTM to lat/long). The format is identical to that given in Section 1.3.2.5 and each line of the header consists of a self-explanatory keyword followed by a numeric value.

#### Virtual gauge output data file (.gauge)

This file is used to specify the x,y locations and lengths of cross-sections where the user wishes the model to output a time series of discharge crossing the section. This option is activated by including the keyword gaugefile in the .par file and following this with the name of the .gauge file to be read. For each location specified in the file the direction identifies the cell face from which discharge will be measured and the direction of positive flow (e.g. N, E, S or W). The width is then the length of the cross section in an easterly direction for measuring flows to the north and south, and a southerly direction for flows to the east or west (note that the distance will be rounded up to nearest cell width). The discharge value is written out at each massint interval. The format of the file is as follows:

Line 1; number of virtual gauge sections.

Line 2; X1 Y1 Direction1 Width1

Line 3: X2 Y2 Direction2  Width2

etc… … … … …

Line i: Xi Yi Directioni Widthi

An example .gauge file is given below:

3

388869.59 233696.30 N 100

386307.41 239076.10 E 50

383681.45 245652.34 S 200

## 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.

## 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\_win [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. Linux only option, ignored in windows. It assumes you have gzip installed. If not it generates an error but otherwise files are created ok, just not compressed.** |
| -dir dirname | **Gives the directory name for results files. Overrides the name given after the keyword dirroot in the .par file.** |
| resroot | **Root for naming of results files (e.g. root.op, root.mass, root-0001.wd etc)** |
| -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 or the values given in the .n.ascii 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 hour of computational time by default using the output file naming convention specified in the .par file after the keyword resroot. See Section . If specified, the interval given after the keyword checkpoint in the .par file is used. Although this would also switch on checkpointing anyway making the use of this command line option unnecessary .** |
| -loadcheck filename | **Forces program to read in an alternative checkpoint filename at start. Useful for when you don’t want the start checkpoint file overwritten by the program as it goes along. Also turns checkpointing on with default features (as option –checkpoint). If specified, the interval given after the keyword checkpoint in the .par file is used.** |
| -log | **Redirects screen output to a log file in the results directory.** |
| -debug | **Outputs 3 files; the final dem after burning in the channel and bank mods (\*.dem), the channel mask (\*.chmask) and the channel segment mask (\*.segmask).** |
| -dynsw | **Implements the full dynamic wave steady state initial solution for the 1D diffusive channel solver** |
| -dhlin value | **Overwrites the linearization threshold value for the adaptive version which is currently set as a function of dx and a slope of 0.0002 from Cunge et al., 1980 and Hunter et al., 2005.** |
| -kill value | **Forces the model to exit after a given length of computation time (in hours) which is useful on clusters which put limits on maximum run time.** |

Table : Command line options for LISFLOOD-FP.

The order in which command line options are used is not important. Just remember that the parameter file is the last argument on the command line.

If the verbose mode is on and “comp\_out” keyword specified, LISFLOOD-FP will output a time to completion estimate to the screen at every save interval. This is useful when trying to work out when the run will complete. Times are in minutes, an example is shown below.

T(mins): M: 500.0, C: 5.3, M/C: 94.94, ETot: 17.6, EFin: 12.3

M: model time

C: computer time (real world minutes spent processing)

M/C: Time ratio (In this case, 100model minutes are processed for every real world minute)

ETot: Estimated total time for run

EFin: Estimated time to completion of current run.

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 1 m. The user can override the default value by using the htol parameter in the .par file.

### Checkpointing

LISFLOOD-FP has a very useful checkpointing facility. This allows it to write out a file containing the current state of the model. This file is repeatedly overwritten at a default or user defined computation time interval. If the program crashes or is killed during the run, this allows the run to restart from when the last checkpoint write occurred rather than from the beginning again. This facility is turned on by using the checkpoint option in the parameter file. The default interval is 1 hour computation time. If the user requires a different interval, this number (in hours) should be placed after the checkpoint keyword.

There is also a -checkpoint command line option, although this does not allow the user to specify an interval on the command line and uses the default 1 hour. Note, if an interval is specified using the checkpoint option in the parameter file, this will be used. However, this makes the use of the command line -checkpoint option superfluous anyway!

If checkpointing is on, then when the model starts it automatically looks for the default file named *“resroot”*.chkpnt in the directory from which the model was executed. If it finds the file, it will assume that it is from a previous partial run and attempt to read it in and then restart from that point. If it does not find the file it will assume that this is a fresh run and create the file. If you do not want to restart the run from the checkpoint, just delete the \*.chkpnt file.

It is also possible to start the checkpointing from an alternative filename, which does not then get overwritten by the checkpoint facility. You do this by using the the command line option –loadcheck “filename” or the loadcheck “filename” option in the parameter file. Note, if there is a default named checkpoint file existing when LISFLOOD-FP starts, it will assume that this is newer (ie later on in the run) than the alternative starting point and load this to start the run. Just delete the default checkpoint file if you want to start again from your alternative starting checkpoint file. The loadcheck option switches on the checkpointing by default, so there is no need to also specify this at the same time, unless you want to dictate a user defined interval.

The checkpointing facility writes a copy of all important variables to a binary file. This saves space compared to an ascii file and maintains model precision. However, it does mean you may not be able to use the checkpoint file on a different machine (eg Linux then Windows). LISFLOOD-FP may well crash if the new machine uses a different binary convention (known as little or big endian). You may also experience a crash if you change some of the run parameters and expect LISFLOOD-FP to restart from a checkpoint file written with different parameters. LISFLOOD-FP does do some basic parameter checks when reading in a checkpoint file, such as domain size, but mostly assumes the basic parameters don’t change. Importantly, if the LISFLOOF-FP version number or checkpoint version number has changed since the checkpoint file was created, the code will issue a warning and exit. This is to prevent problems of forward and backward compatibility.

A checkpoint is made at the end of the simulation as well as during it - this makes it possible to, for example, run the model in steady state for a period, then run multiple different hydrographs from that point - the new hydrograph should include the period of steady state in the timings.

Important Note: After a checkpoint restart, the output written to the mass file is appended to the file rather than overwriting the previous lines. A checkpoint break line is added before the new lines are written, and this will let you see where it started up again, but leads to a discontinuous mass record. You can manually edit the mass file after the run to remove the overlap if you want the data continuous. The stage output file behaves in a similar fashion. Numbered results files continue to be output at the correct time.

### 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.

#### 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, space separated:

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: NumTsteps. Number of time steps since the start of the simulation.

Column 5: Area. Area inundated in km2..

Column 6: Vol. Volume of water in the domain.

Column 7: Qin. Inflow discharge in m3s-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 m3s-1.

Column 10: Qerror. Volume error per second in m3s-1.

Column 11: Verror. Volume error per mass interval (massint variable in the parameter file) m3.

Column 11: Inf+Evap. Cumulative Infiltration loss over the simulation in 103 m3.

#### 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 out-xxxx-T.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.

#### Channel water surface profile (.profile)

This file gives the channel water surface profile at either: (a) the time of each satellite overpass if this is specified using the parameter file keyword overpass, or overpassfile for multiple outputs (see section 1.3.2.8) or (b) at each time defined by the parameter file keyword saveint If the keyword profiles appears in the .par file.. Multiple overpass filenames will take the format of out-xxxx-T.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. For profile output at each saveint then X is the sequential output file number (0001, 0002, 0003 etc.). This is a text file consisting of eleven columns of data for each channel segment:

Column 1: ChanX – channel segment X location

Column 2: ChanY – channel segment Y location

Column 3: Chainage - distance along the channel thalweg from the upstream boundary in metres.

Column 4: Width – channel width in m

Column 5: Mannings – channel mannings

Column 6: Slope – channel slope

Column 7: BankZ – Bank elevation in m

Column 8: BedElev – bed elevation in m

Column 9: WaterElev – water elevation in m

Column 10: WaterDepth – water depth in m

Column 11: Flow – flow in cumecs

#### Synoptic water depth, water surface elevation and flow flux files (-xxxx.wd and -xxxx.elev)

These files consist of a grid of water depths and water surface elevations values in ARC ascii raster format for each pixel at each save interval (saveint) specified in the parameter file. In this naming convention xxxx is the saveint number. By default these output options are turned on but production of each set of files can be suppressed by putting the logical keywords depthoff or elevoff in the .par file. Units are in metres.

#### 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.

#### 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.

#### Flux values (-xxxx.Qx and -xxxx.Qy)

These files consist of a grid in ARC ascii raster format of the flux values in the x and y Cartesian directions. Grids are output at each save interval (saveint) specified in the parameter file and xxxx is the saveint number. By default these files are not produced and are only output if the keyword qoutput appears in the .par file.

#### Adaptive time step (-xxxx.Tx and -xxxx.Ty) and flow limiter (-xxxx.QLx and -xxxx.QLy) values

These files consist of a grid in ARC ascii raster format of the optimum adaptive time step (equation [14]) and flow limiter values (equation [8]) in the x and y Cartesian directions. Grids are output at each save interval (saveint) specified in the parameter file and xxxx is the saveint number. By default these files are not produced and are only output if the keywords tpouput and qloutput appear in the .par file.

### 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.

## 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.

Fewtrell, T., Bates, P.D., Horritt, M. and Hunter, N. 2008. Evaluating the effect of scale in flood inundation modelling in urban environments. *Hydrological Processes*, 22, 5107–5118.

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., 2006. Improved simulation of flood flows using storage cell models. *Proceedings of the Institution of Civil Engineers, Water Management*, 159 (1), 9-18.

Hunter, N.M., Bates, P.D., Neelz, S., Pender, G., Villanueva, I., Wright, N.G., Liang, D., Falconer, R.A., Lin, B., Waller, S., Crossley, A.J. and Mason, D. 2008. Benchmarking 2D hydraulic models for urban flood simulations. *Proceedings of the Institution of Civil Engineers: Water Management*, 161 (1), 13-30.

Hunter, N.M., Horritt, M.S., Bates, P.D., Wilson, M.D. and Werner, M.G.F. 2005. An adaptive time step solution for raster-based storage cell modelling of floodplain inundation. *Advances in Water Resources*, 28, 975-991.

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.

Neal, JC, Fewtrell, TJ & Trigg, MA. (2009) 'Parallelisation of storage cell flood models using OpenMP', *Environmental Modelling & Software*, **24**, 872-877

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.

Programmers guide

## 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.

## 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.

double 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.

double BankQ(int chani);

For the chanith channel cell calculates the flow *out* of the channel by summing appropriate floodplain flows.

double 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.

double 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.

double CalcFPQx(int i,int j);

double CalcFPQy(int i,int j);

Calculate flows in x or y direction according to Manning’s equation for adjacent non-weir floodplain cells.

double CalcWeirQx(int i,int j);

double 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.

## Key variable names

### General variables:

int xsz,ysz; Number of cells in x,y directions

int chsz; Number of channel cells

double dx,dy,dA; x,y cell size, cell area

double FPn; Floodplain Manning’s n (uniform)

int Nit,ts; Number of iterations, current iteration

double Tstep; Time step

double SolverAccuracy; Convergence criterion of Newton-Raphson solver

double Qin,Qout,QChanOut,Hds; Flow in, out, channel out, depth at downstream end of channel

double DepthThresh; Depth below which cell is considered dry

double 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

double htol=0.5; Tolerance for removing low lying bank areas

double 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

### 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

### 2D raster arrays (xsz x ysz):

double \*DEM,\*H,\*Qx,\*Qy,\*maxH; Pointers to rasters for DEM, water depth, x and y floodplain flows and maximum depth

double \*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):

double \*Chandx,\*Shalf; chsz array, distance between channel cells and bed slope

double \*Chainage,\*A,\*NewA; Chainage along channel, flow cross sectional area in channel, new area (as produced by ChannelQ)

double \*ChanWidth,\*ChanN; Channel width and Manning’s n

int \*ChanX,\*ChanY; x,y position of channel cell in DEM raster

double \*Weir\_hc,\*Weir\_Cd; Crest height, discharge coefficient of weirs

double \*Weir\_m,\*Weir\_w; Modulus, width available for flow

int \*Weir\_Identx,\*Weir\_Identy; !=-1 for weir link between cells

### 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

double \*BC\_Val; Either imposed static values (HFIX or QFIX boundary) or location of time varying values in BCVarlist (HVAR or QVAR)

double \*\*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

double \*Q\_Val,\*\*QVarlist; As above, but for flows imposed on

int \*Q\_Ident; channel (chsz rasters).

char \*Q\_Name;

# Example applications

## Fluvial applications

### 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 m3s-1 and drains a catchment of 1000 km2. 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 m3s-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 m3s-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

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

## Plain flooding applications

### 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.

# 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)

**Kinematic 1D Channel Flow**

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:

 []

 []

*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), *S0* 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.

**Diffusive 1D Channel Flow**

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

 []

 []

The term in brackets is the diffusion term, which forces the flow to respond to both the bed slope and the free surface slope. Usually, Q is chosen as the dependent variable (Chow, 1988, p. 296), as it results in smaller relative errors in the estimation of discharge. However, as we are interested primarily in water levels (which dictate the flood extent), the cross-sectional area A is used as the dependent variable. This simplifies the inclusion of the diffusion term. The diffusion equation is solved using an explicit LU decomposition scheme - Crout's method (Horritt and Bates, 2001a). When using the FREE boundary, Manning’s equation is substituted into the Jacbian for Q in the last channel section to allow correct matrix solving and to allow communication with the upstream channel sections.

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 < Δ*x*) 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):

 []

 []

where *hi,j* is the water free surface height at the node (*i,j*), Δ*x* and Δ*y* are the cell dimensions, n is the effective grid scale Manning’s friction coefficient for the floodplain, and *Qx* and *Qy* describe the volumetric flow rates between floodplain cells. *Qy* is defined analogously to equation 6. The flow depth, *hflow*, 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:

 []

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:

 []

 []

where *qx* and *qy* 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:

 []

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:

 []

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

 []

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

 []

At equality, the  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:

 []

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:

 []

with a similar expression for *qy*. 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.

1. Tel: +44-117-928-9108; Fax: +44-117-928-9108; E-mail: [Paul.Bates@Bristol.ac.uk](mailto:Paul.Bates@Bristol.ac.uk) [↑](#footnote-ref-1)