

FLOOD MODELLING USING OPEN SOURCE SOFTWARE

A new methodology for flood modelling using open source software has been presented. The geographical information system software QGIS has been applied to a LiDAR image to produce a 3D file in stereolithography format.

by Dr Thomas J Scanlon



The study presents a methodology for flood modelling using entirely free-to-download, open-source software. The procedure encompasses three stages of pre-processing, solving and post-processing. In the pre-processing stage the geographic information system software QGIS is used to process a LiDAR image into a stereolithography file. This file is then used to generate a digital elevation map using the open-source computational fluid dynamics code OpenFOAM. The shallow water equations are solved within the framework of OpenFOAM using a solver called shallowFoam and post-

processed using Paraview. Results are presented for four test cases of increasing complexity and the outcomes show that the methodology produces very satisfactory agreement when compared with commercial flood models

Introduction

Given the now established link between extreme weather events and climate change (Blöschl 2019, Robins 2019) an increasing likelihood of severe flooding exists world-wide. A significant number of inundation events have occurred over the last 20 years in the UK with current flood

About Author



Dr Thomas J Scanlon
 BEng PhD CEng MIMechE
 Consultant
 MTS-CFD
 Glasgow, Scotland
 Email - tomscanlon63@googlemail.com
 Website - www.mts-cfd.com

damage costs estimated at around £1.3 billion each year (ECIU 2019). This is in addition to the human psychological damage where 1 in 6 properties in the UK are exposed to significant flooding risk (CCRA 2012). Although technical and economic barriers mean that the complete elimination of flood risk is impractical, comprehensive flood management plans must be employed to mitigate such adverse weather events. These strategies can be informed by appropriate flood modelling techniques and this is the focus of this paper.

Several commercial flood models are available (MIKE 2019, TUFLOW 2019, ISIS 2019, SOBEK 2019, INFOWORKS_ICM 2019) and in 2013 such codes were invited by the UK government Department of Environment, Food and Rural Affairs (DEFRA) to take part in a series of flood modelling exercises (DEFRA 2013). The goal of this project was to generate a series of benchmark cases for flood modelling and gauge the performance of each code with a view to establishing best practice when applied to flooding events. Among the codes tested in the DEFRA exercise was the open source code ANUGA (ANUGA 2019) developed by the Australian National University and Geoscience Australia. The code performed well in comparison with its commercial counterparts, however, it was only applied to a select number of benchmark cases. shallowFoam (Mintgen 2017) is the open source flood model solver used in this paper and represents an alternative code to ANUGA. The code has been released open source via the software repository GitHub (shallowFoam 2019). Developed within the framework of the open source computational fluid dynamics (CFD) solver OpenFOAM (OpenFOAM 2019), shallowFoam solves the shallow water equations (SWE) in a finite volume meshing environment. With an estimated user base of 10,000 (OpenORG 2019), OpenFOAM is a parallelised CFD code widely employed in the academic and

industrial communities. shallowFoam has been successfully applied to the case of a dam break (Zeng et al 2017), however, this work necessitated the use of radial basis functions in the generation of the digital elevation map (DEM). In the paper that we present, use is made exclusively of free-to-download software in the generation of the DEM and does not rely on any external mathematical manipulation.

shallowFoam Solver

The shallowFoam solver solves the 2D (x, y), depth-averaged, shallow water equations (SWE) according to the coordinate system shown in Figure 1. Here, h is the flow depth (m), z_b is the bottom surface level (m) and z_w is the water level (m).

The SWE correspond to transport equations for the conservation of mass and momentum according to equation 1.

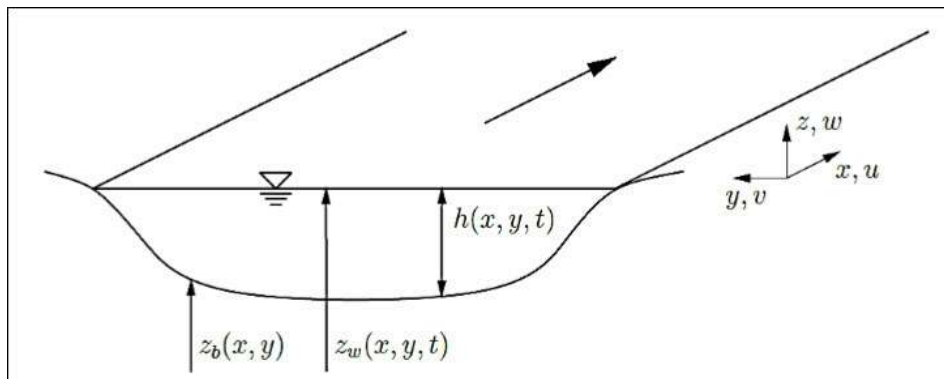


Figure 1: Coordinate system and variables for the shallow water equations (Mintgen 2017).

Equation 1

$$\frac{\partial h}{\partial t} + \frac{\partial q_i}{\partial x_i} = 0 \quad (i = 1, 2), \tag{1}$$

and,

$$\frac{\partial q_i}{\partial t} + \bar{u}_j \frac{\partial q_i}{\partial x_j} = -\frac{g}{2} \frac{\partial h^2}{\partial x_i} - gh \frac{\partial z_b}{\partial x_i} - \frac{\tau_{bi}}{\rho} + \frac{\partial}{\partial x_j} \left[v_t \left(\frac{\partial q_i}{\partial x_j} + \frac{\partial q_j}{\partial x_i} \right) \right] \quad (i=1, 2), \tag{2}$$

respectively. $q_i = h\bar{u}_i$ has been introduced as the specific discharge (m²s⁻¹), g is the gravitational acceleration (ms⁻²) and the bottom surface shear stress τ_{bi} is modelled using:

$$\frac{\tau_{bi}}{\rho} = \frac{n^2 g}{h^{\frac{1}{3}}} \bar{u}_i |\bar{\mathbf{u}}|, \tag{3}$$

where $|\bar{\mathbf{u}}|$ is the velocity vector magnitude and n is Manning's roughness coefficient. Turbulence closure is via a depth-averaged parabolic eddy viscosity model where the turbulent viscosity v_t is set as:

$$v_t = \frac{\kappa}{6} u^* h, \tag{4}$$

with the von-Kármán constant $\kappa = 0.41$ and the shear velocity u^* calculated by:

$$u^* = \sqrt{\frac{\tau_b}{\rho}} \tag{5}$$

Methodology

This section will present the general methodology for setting up, running and post-processing a flood model case. The case presented here is CASE 2, one of the four test cases considered in this paper. All other cases follow the same methodology. CASE 2 is one of the DEFRA benchmark cases and considers a dam burst scenario into a valley of approximately 17 km in length. Figure 2 shows the LiDAR image of the valley and corresponding shape file outlined in light blue which outlines to the computational domain. The ground slope is shown as an altitude-distance graph along the valley centreline in green and sampling points are shown as numbered crosses. Inlet flow conditions are supplied at the red line in the upper valley as hydrograph as shown in Figure 3.

The first step in the methodology is to create the digital elevation map (DEM). The open source geographical information system (GIS) software QGIS (QGIS 2019) is used to process the LiDAR .asc file supplied as part of the DEFRA benchmark case. A QGIS plug-in called DEMto3D is employed to generate a stereolithography (STL) file and this STL file is brought into OpenFOAM to generate a 3D computational mesh using the OpenFOAM meshing utility *snappyHexMesh*. The result of this meshing is shown in Figure 4. The height to the cell face centres from a bottom reference plane is then calculated with a new OpenFOAM utility called *Test-wallDist*. The resulting DEM is shown in Figure 5.

Following the generation of the DEM a new 2D mesh is created which covers the same computational area as the 3D one. The 2D mesh is generated using the combined OpenFOAM utilities *blockMesh*, which creates an

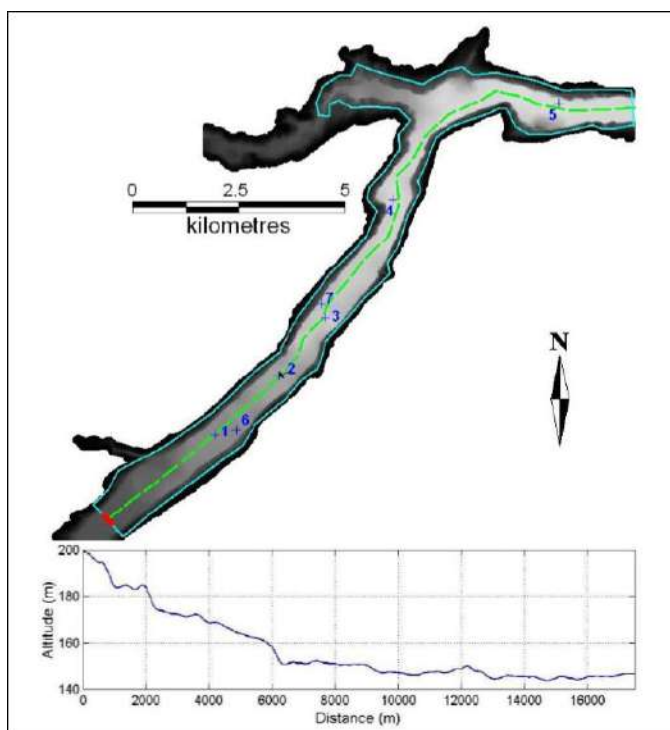


Figure 2: LiDAR image of the valley benchmark case and height-distance graph along centreline.

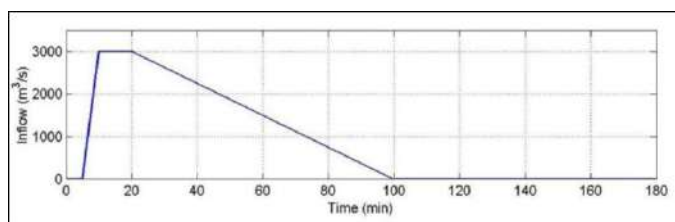


Figure 3: Inlet hydrograph for dam break scenario.

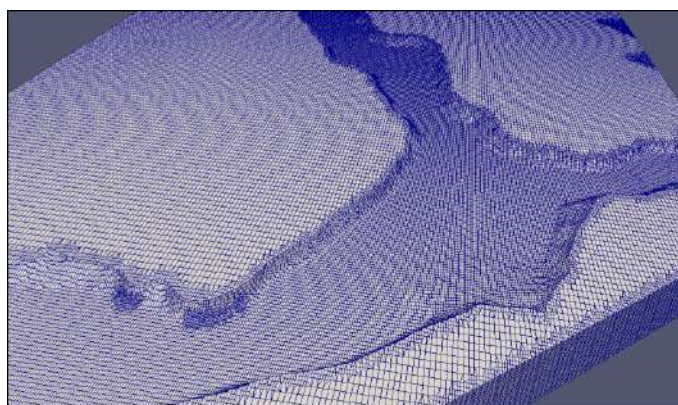


Figure 4: 3D snappyHexMesh showing the lower valley zone.

orthogonal, structured mesh and *snappyHexMesh*, which creates an unstructured polyhedral mesh. The OpenFOAM utility *extrudeMesh* is then employed generate a 2D mesh with “empty” patch types at the top and bottom surfaces. Such “empty” patches are necessary to force the OpenFOAM solution to be two-dimensional in nature. It is also at this stage that additional STL files



Figure 5: Valley DEM generated using OpenFOAM utility Test-WallDist.

may be incorporated into the 2D model to provide, for example, zones of different surface roughness for rivers, woodland, grassland, buildings

etc. – see CASE 4 for an example of this. Finally, the digital elevations are mapped from the 3D solution on to the 2D mesh using the OpenFOAM utility *mapFields*. The solver *shallowFoam* is then ready to be run, taking advantage of OpenFOAM’s unlimited parallel processing capability. Post-processing is via the open source visualisation application Paraview which is supplied with OpenFOAM.

Results and Discussion

Case 1

Test CASE 1 consists of a sloping topography with two depressions separated by an obstruction as shown in Figure 6, and of width 100m. A time-varying inflow discharge is applied as an upstream boundary condition at the left-hand side, causing a flood wave to travel down the 1:200 slope. While the total inflow volume is just sufficient to fill the left-hand side depression at $X = 150\text{m}$, some of this volume is expected to over-top the obstruction because of momentum conservation. The objective of the test is to assess the code's ability to conserve momentum over an obstruction in the topography and settle in the depression on the right-hand side at $X = 250\text{m}$.

Figures 7-9 show that the *shallowFoam* results for water level and velocity are in satisfactory concurrence with the results produced by the commercial codes MIKE_FLOOD and TUFLOW and the open source code ANUGA. This gave confidence in the *shallowFoam* code and DEM methodology to proceed to the more challenging test CASE 2.

This *shallowFoam* case was run on a single Intel Core i7-7820HK CPU @2.90 GHz. A computational mesh of 1200 cells was used, a time-step size of 1 s employed and the computational time to complete the run was 10 s.

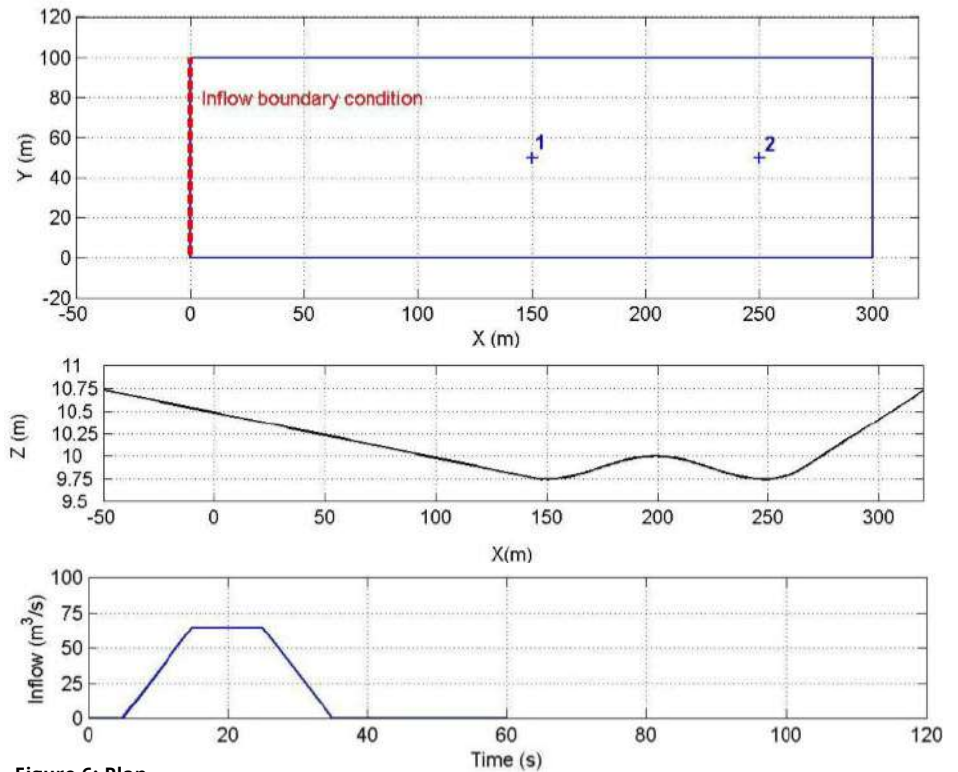


Figure 6: Plan view, elevation and inlet hydrograph for test CASE 1.

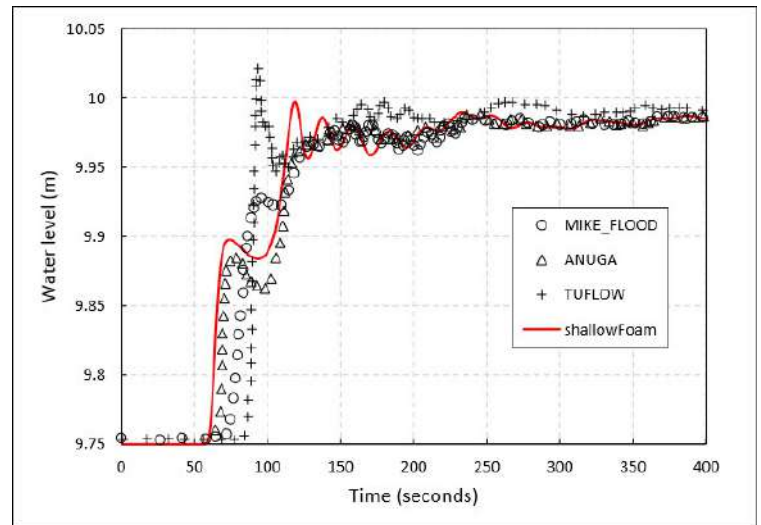


Figure 7: Water level versus time for point 1 in test CASE 1 (see Figure 6 for point locations).

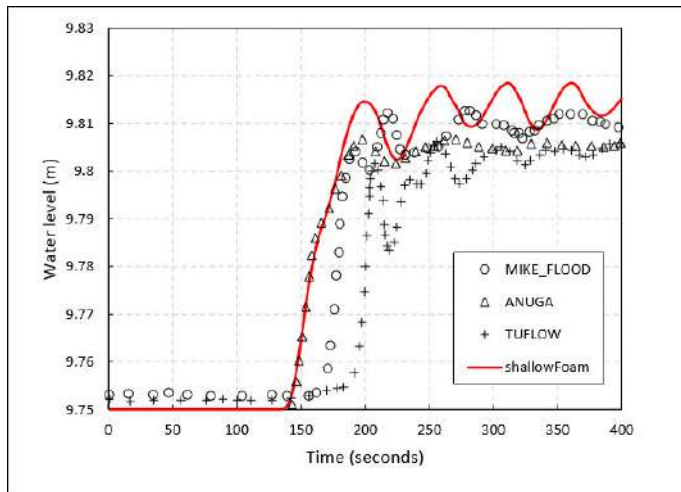


Figure 8: Water level versus time for point 2 in test CASE 1 (see Figure 6 for point locations)

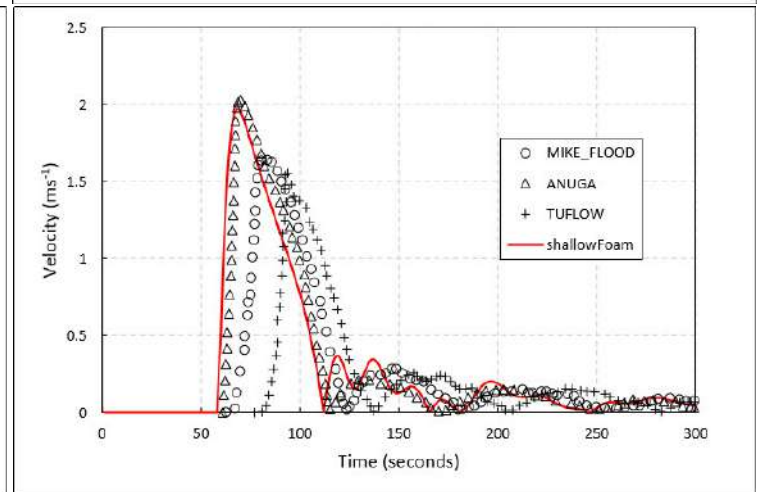


Figure 9: Water velocity versus time for point 1 in test CASE 1 (see Figure 6 for point locations).

Case 2

The geometry and boundary conditions for test CASE 2 valley dam break were outlined in the section on Methodology. Figures 10-11 for water level and velocity, respectively, show a very reasonable agreement between the shallowFoam model and the other

flood codes. A total simulation time of 30 hours was modelled in this case and Figure 12 shown contours of wetted area after 2 hours.

CPUs @2.90 GHz. A computational mesh of 246,000 cells was used, a time-step of 1 s employed and the computational time to complete the run was 2.7 hours.

This *shallowFoam* case was run in parallel using 3 Intel Core i7-7820HK

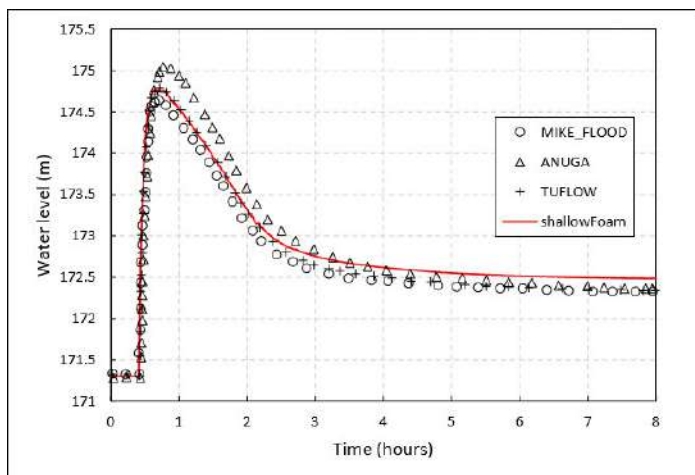


Figure 10: Water level versus time for point 1 in test CASE 2 (see Figure 2 for point locations).

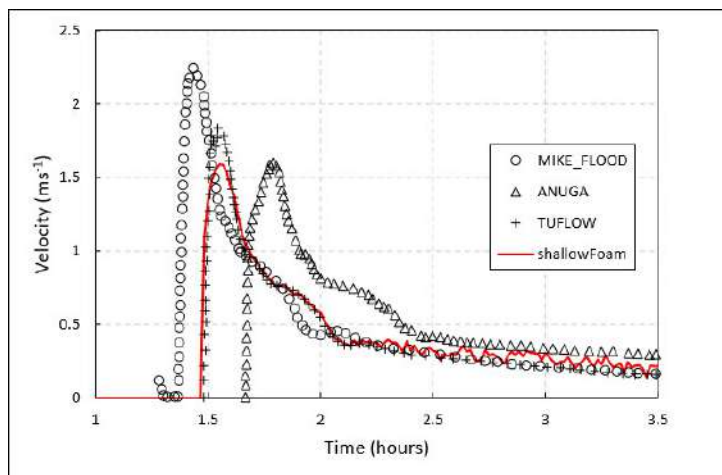


Figure 11: Water velocity versus time for point 4 in test CASE 2 (see Figure 2 for point locations).

Case 3

In this river and floodplain modelling test case the site to be modelled is approximately 7 km long by 0.75 to 1.75 km wide (see Figures 13-14), and consists of a set of three distinct floodplains FP1, FP2 and FP3 in the vicinity of the village of Upton-upon-Severn, England. In the test, the River Severn that flows through the site is modelled for a total distance of ~20km. Boundary conditions are a hypothetical inflow hydrograph (DEFRA 2013). The objective of the test is to assess the package's ability to simulate fluvial flooding in a relatively large river, with floodplain flooding taking place as the result of river bank overtopping.

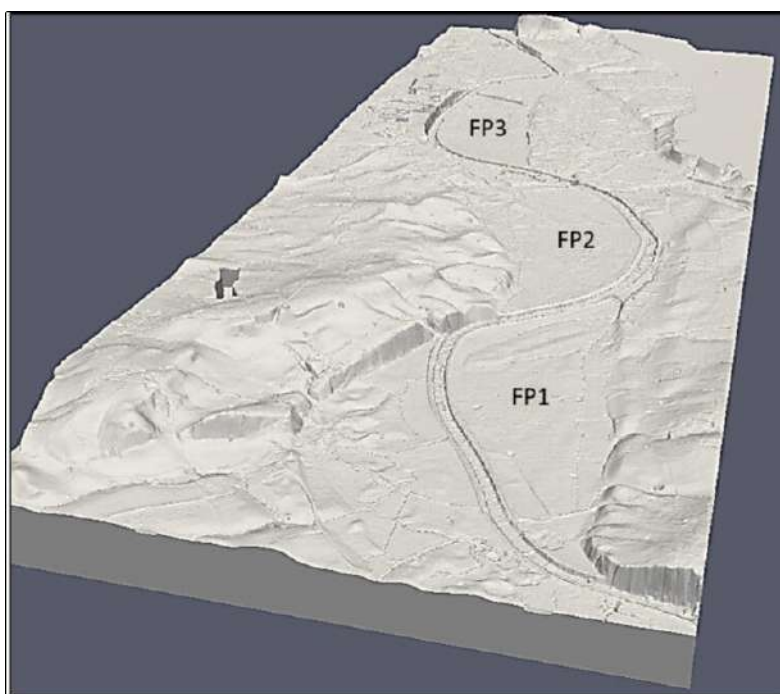


Figure 13: 3D STL file generated by the QGIS plugin DEMto3D showing floodplain regions FP1, FP2 and FP3.

It is evident that there is a relatively wide variation in the range of results for the codes tested in this complex benchmark case. The *shallowFoam* results, both qualitatively and quantitatively, appear to be in reasonable agreement across the

spectrum of data presented with peak water levels and velocities corresponding well with the commercial codes as shown in Figures 15-17.

This *shallowFoam* case was run in parallel using 3 Intel Core i7-7820HK CPUs @2.90 GHz. A computational

mesh of 244,000 cells was used, a time-step of 1 s employed cells and the computational time to complete the run was 5.05 hours.

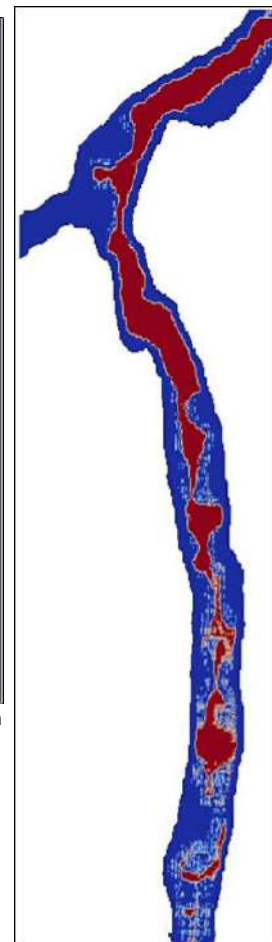


Figure 12: Contours of wetted area (red) for the *shallowFoam* simulation of test CASE 2 after a duration of 2 hours.

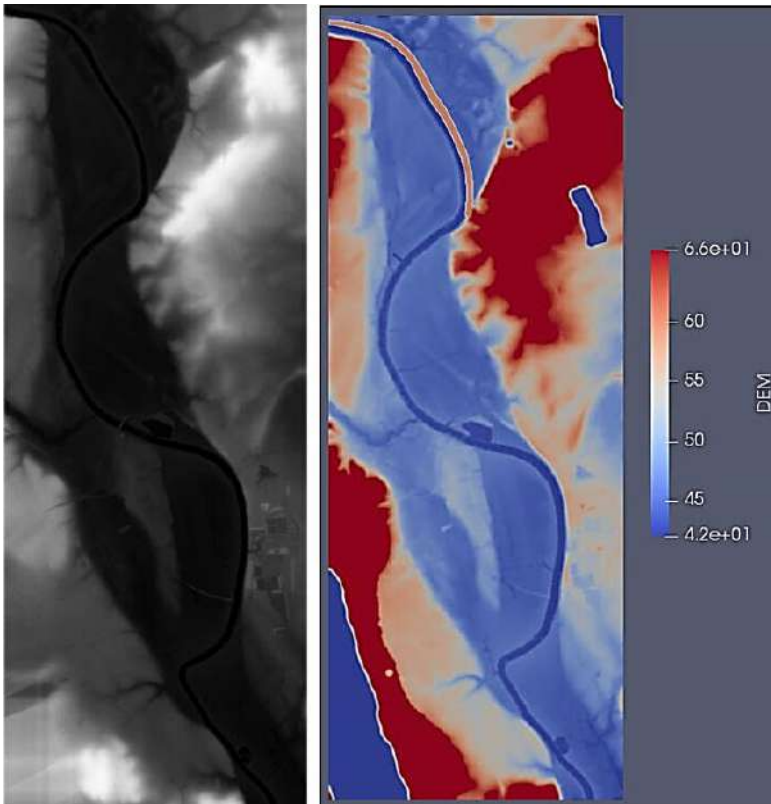


Figure 14: LiDAR image (left) and DEM generated using OpenFOAM utility Test-WallDist.

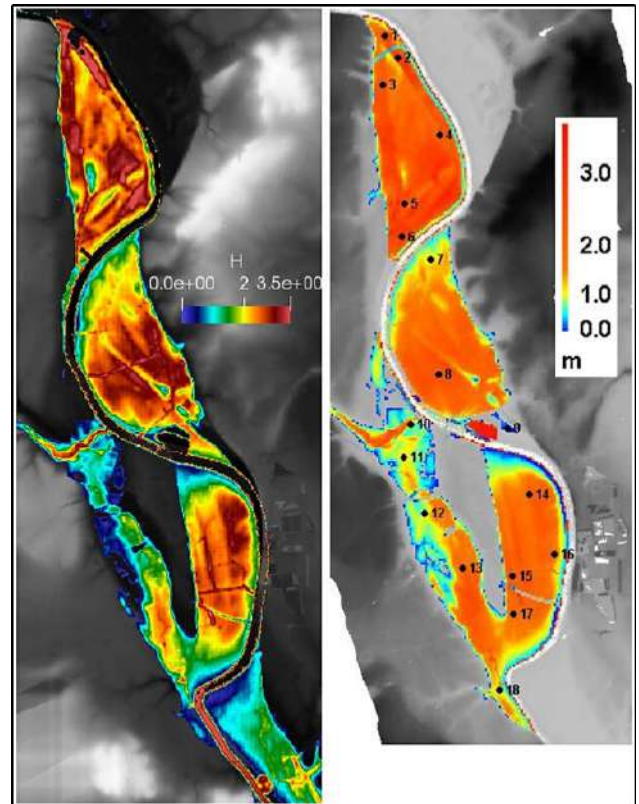


Figure 15 Peak water depths predicted by shallowFoam (left) and the commercial code ISIS-FAST (right). Sampling points P11 and P17 in floodplain FP3 are indicated on the right-hand.

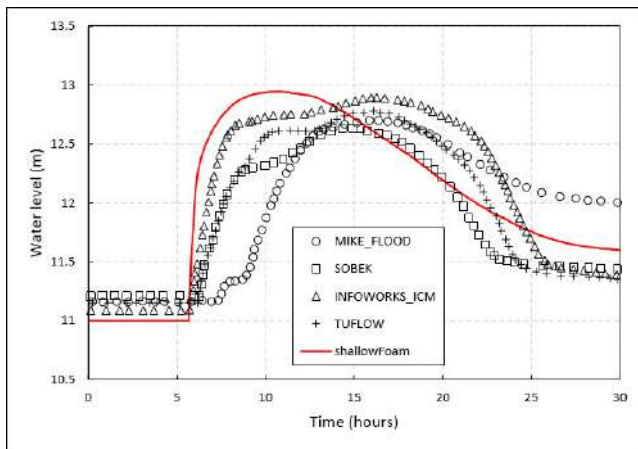


Figure 16: Water level versus time for point 11 in test CASE 3 (see Figure 15 for point locations).

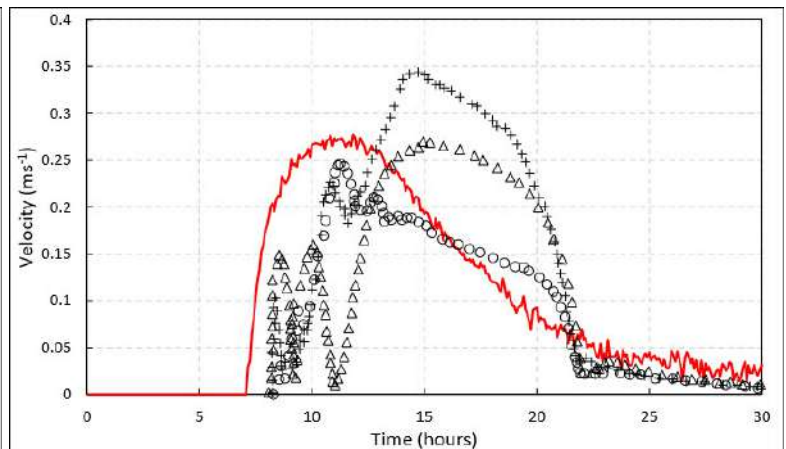


Figure 17: Water velocity versus time for point 17 in test CASE 3 (see Figure 15 for point locations).

Case 4

CASE 4 is a practical example of how the methodology presented in this paper may be used to mitigate a flooding issue. In January 2015 storm Frank caused severe flooding and damage in the town of Ballater, Scotland (FRANK 2019). The methodology has been applied to, firstly, model the flood as it occurred and suggest an possible flood defence solution. Storm Frank hydrograph inputs for the rivers Gairn/Dee and

the river Muick were included as inlet boundary conditions in the model and the shallowFoam code was run for a simulation period of 24-hours.

Figure 18 shows the extent of the computational domain and the proposed flood defence solution in red. Sampling points around the flood barrier are indicated by points P8 to P16. These points are used to extract the water level in front of the

flood defence to give the necessary barrier height to counteract flooding during such a storm.

As discussed in the section on Methodology, different roughness parameters in terms of Manning-Strickler coefficients may be introduced into the model and these roughness zones are shown in Figure 19.

Figure 20 shows the excessive inundation occurring at the time of the peak river flow rate, while Figure 21 shows the mitigation of this by using the flood barrier protection. The results from test CASE 4 show that the methodology described in the paper can be applied to a practical situation and propose solutions for flood mitigation in a complex topography.

This *shallowFoam* case was run in parallel using 3 Intel Core i7-7820HK CPUs @2.90 GHz. A computational mesh of 176,000 cells was used, a time-step of 1 s employed and the computational time to complete the run was 1.92 hours.

Conclusions

A new methodology for flood modelling using exclusively free-to-download, open source software has been presented. The geographical information system software QGIS has been applied to a LiDAR image to produce a 3D file in stereolithography format. This file was then converted to a digital elevation model in the computational fluid dynamics package OpenFOAM.

Finally, the shallow water equations were solved within the framework of OpenFOAM using the flood model *shallowFoam* to produce maps of water height and velocity. In comparison with benchmark results for a range of commercial codes, the *shallowFoam* solutions compared well and produced satisfactory results for key flooding parameters. The final test case showed how the methodology could be applied to a practical flooding problem with mitigation measures successfully implemented. Future work will include incorporation of 1D-2D links for features such as culverts and sluice gates and the implementation of a shock capturing scheme for supercritical flows. The availability of such computational technologies means that effective open source flood modelling software is now accessible for the wider academic, industrial and citizen science communities.



Figure 18: Extent of computational domain around the town of Ballater in test CASE 4. Proposed flood defence barrier (red) and probe sampling points are also highlighted.



Figure 19: Roughness zones for different Manning-Strickler coefficients in test CASE 4.

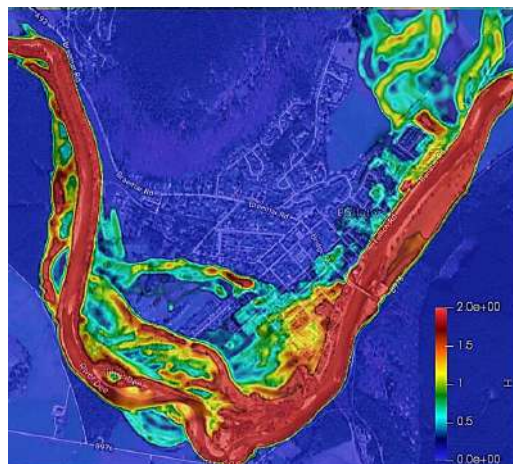


Figure 20: Contours of water height (m) at the time of maximum river flow rate (12 hours) with no flood defence.

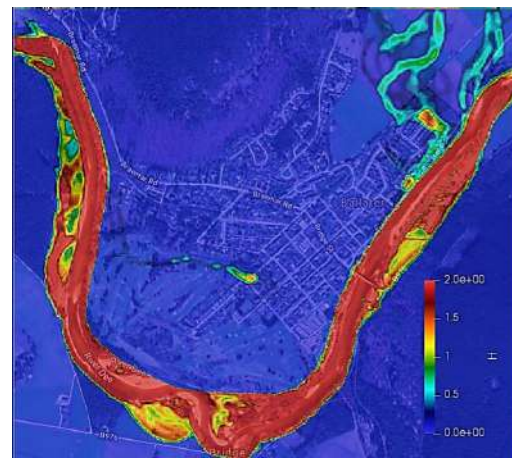


Figure 21: Contours of water height (m) at the time of maximum river flow rate (12 hours) with flood defence in place.

References

- Blöschl, G. *et al.*, 2019 Changing climate both increases and decreases European river floods, *Nature*, 573, 108–111.
- Robins, E. and Lewis, M. J. 2019 Changing Hydrology: A UK Perspective, *Coasts and Estuaries - The Future*, 611-617
- ECIU: <https://eciu.net/briefings/climate-impacts/flood-risk-and-the-uk> (accessed 25th October 2019)
- CCRA: <http://ccra.hrwallingford.com/CCRAReport/s/downloads/CCRASummaryFloods.pdf> (accessed 25th October 2019).
- MIKE-FLOOD: <https://www.mikepoweredbydhi.com/products/mike-flood> (accessed 25th October 2019).
- TUFLOW: <https://www.tuflow.com/> (accessed 25th October 2019)
- ISIS: <https://www.floodmodeller.com/> (accessed 25th October 2019)
- DEFRA 2013: <https://www.gov.uk/government/publications/benchmarking-the-latest-generation-of-2d-hydraulic-flood-modelling-packages> (accessed 25th October 2019)
- SOBEK 2019: <https://www.deltares.nl/en/software/sobek> (accessed 25th October 2019)
- INFOWORKS_ICM 2019: <https://www.innovyze.com/en-us> (accessed 25th October 2019)
- ANUGA: <https://anuga.anu.edu.au/> (accessed 25th October 2019)
- shallowFoam 2019: <https://github.com/mintgen/shallowFoam> (accessed 25th October 2019)
- OpenFOAM 2019: <https://www.openfoam.com/> (accessed 25th October 2019)
- OpenORG 2019: <https://openfoam.org/news/funding-2019/> (accessed 25th October 2019)
- QGIS 2019: <https://qgis.org/en/site/> (accessed 25th October 2019)
- FRANK 2019: <https://www.bbc.co.uk/news/uk-scotland-north-east-orkney-shetland-38332590> (accessed 25th October 2019)
- Mintgen, G. F. 2017 Coupling of Shallow and Non-Shallow Flow Solvers - An Open Source Framework, (accessed 25th October 2019) <https://mediatum.ub.tum.de/doc/1369622.pdf>