

# Improving Inundation Forecasting using Data Assimilation

E.S. Cooper<sup>1</sup>, S.L. Dance<sup>1,2</sup>, N.K. Nichols<sup>1,2</sup>, P.J. Smith<sup>2</sup>, J. Garcia-Pintado<sup>1</sup>

<sup>1</sup>Department of Meteorology <sup>2</sup>Department of Mathematics and Statistics

University of Reading, UK

March 2016

---

## Abstract

Fluvial flooding is a costly problem in the UK and worldwide. Real-time accurate inundation forecasting can help to reduce the damage caused by inundation events by alerting people to take necessary mitigating actions. This work is part of an effort to improve inundation forecasting using data assimilation (DA). DA is a method for combining a numerical model of a system with observations in order to best estimate the current state of the system. A river inundation model has been developed using numerical implementation of the shallow water equations with an inflow source term added; the model and implementation of the source term are described here. The model has then been used with idealised river valley topographies in order to investigate the sensitivities of the system to model parameters describing the effect of friction between water and the river channel, and the effect of the topography slope at the downstream boundary. Initial DA experiments using an ensemble Kalman Filter are also described. Identical twin experiments show that the DA as implemented in this domain can correct the water levels at the time of the observations, and that more observations lead to a better correction. However, by the time of the next observations very similar water levels are predicted, regardless of the number of observations used in the assimilation. This implies that the effective time for observations in this system is small compared to the time between observations.

## 1 Inundation Model using Clawpack

Systems of partial differential equations (PDEs) can be used to model a wide variety of physical situations. The Navier-Stokes equations are a set of such equations which describe how pressure, velocity, temperature and density change with time and in space in a moving fluid. The Navier-Stokes equations are derived from laws of conservation of momentum and mass. The equation can be simplified in certain cases to give the shallow water equations (SWE). The SWE hold in situations for which the viscosity of the fluid can be neglected, and the horizontal scale of the system is much larger than the vertical scale, i.e. the horizontal domain is large compared to the depth of the water. A further assumption is that the fluid is incompressible, so that its density is constant.

The shallow water equations for two spatial dimensions,  $x$  and  $y$ , can be written as (e.g.[9])

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{q})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{q})}{\partial y} = \mathbf{R}(\mathbf{q}), \quad (1)$$

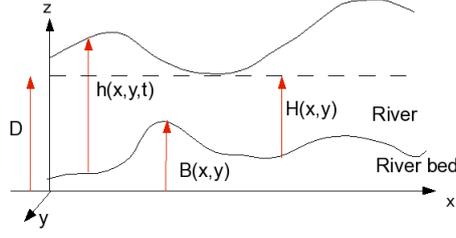


Figure 1: SWE schematic

where  $q$  is a vector of conserved quantities

$$\mathbf{q} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad (2)$$

$h$  represents depth of the fluid in the  $z$  direction (see figure 1),  $g$  is acceleration due to gravity,  $u$  and  $v$  represent velocity in the  $x$  and  $y$  directions respectively. The bathymetry of the river bed depicted in figure 1 is represented by  $B(x, y)$ , assumed here to be constant in time. It is sometimes useful to refer to a reference or steady state depth; this is shown here with a dashed line.

In equation (1),  $\mathbf{F}(\mathbf{q})$  and  $\mathbf{G}(\mathbf{q})$  represent fluxes of the conserved quantities in the  $x$  and  $y$  directions respectively. For the SWE these are

$$\mathbf{F}(\mathbf{q}) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix} \quad \text{and} \quad \mathbf{G}(\mathbf{q}) = \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix}. \quad (3)$$

In equation (1),  $\mathbf{R}(\mathbf{q})$  is a source term; when  $\mathbf{R}(\mathbf{q}) = 0$ , the system is said to be homogeneous. The homogeneous equations describe a system in which, within a given volume, any change in the conserved quantities with time is equal to the value of flux of the quantities at the boundary of the volume. When  $\mathbf{R}(\mathbf{q}) \neq 0$  in a volume, this means that there is a source of one or more of the conserved quantities within that volume. (For the case that  $\mathbf{R}(\mathbf{q})$  has negative values, this is sometimes referred to as a sink, but source will be used here for both positive and negative  $\mathbf{R}(\mathbf{q})$ ).

There are a number of inundation models available e.g. LISFLOOD-FP [1], HEC-RAS [13] and TELEMAC [6], all of which use the numerical solution of the SWE to predict flood extent as a function of time. In this work Clawpack [2] is used for inundation modelling. Clawpack is a freely available collection of code which uses sophisticated Riemann solvers to model systems of differential equations such as the SWE. The software can be downloaded by the user and adapted to fit the problem of interest.

Various methods can be used for the numerical solution of PDEs such as the SWE. These include finite element methods (FEM), finite difference methods, finite volume methods (FVM) and boundary element methods. Finite volume methods are particularly suited to situations in which the behaviour of the system is not smooth; by using the integral (also called the ‘weak’) form of conservation laws it is possible for FVM solutions to

‘capture’ or ‘track’ shocks in the solutions of the equations [9]. Clawpack [2] uses finite volume methods and can solve homogeneous and non-homogeneous equations. The following section describes how the code deals with source terms due to friction, and how a new inflow source has been implemented.

## 1.1 Source terms in Clawpack

Clawpack can solve systems of partial differential equations with source terms. The pre-existing code for a source term due to friction is described in section 1.1.1. We have developed and implemented a new source term to add water to the domain in order to model river-like flow; this new inflow source term is described in section 1.1.2.

### 1.1.1 Friction

Friction between the fluid and the channel in which it is flowing acts as a momentum source in the SWE. This is represented in Clawpack with a source term of the form

$$\mathbf{R}(\mathbf{q})_{friction} = \begin{bmatrix} 0 \\ -\gamma(hu) \\ -\gamma(hv) \end{bmatrix}, \quad (4)$$

where  $\gamma$  is given by

$$\gamma = \frac{gn^2 \sqrt{(hu)^2 + (hv)^2}}{h^{\frac{7}{3}}}. \quad (5)$$

As before,  $h$  is water depth and  $u$  and  $v$  are velocities in the  $x$  and  $y$  directions. Acceleration due to gravity is denoted  $g$  and  $n$  is Manning’s friction coefficient. This coefficient describes the roughness of the channel bed in which the water is flowing and in practical applications its value is usually determined empirically. The value of  $n$  is specified by the user in the Clawpack code, and can vary over the domain if specified in the simulation setup. The units of  $n$  are  $sm^{-3}$ .

In Clawpack, inhomogeneous sets of equations are solved using a method of fractional stepping as described in [9] p.380-395 . This method splits the equation into two simpler problems - one homogeneous conservation law and one inhomogeneous partial differential equation - which can be solved independently over the same time step. The solutions are then combined in an alternating fashion to give a solution to the whole problem. In this case, the set of problems to be solved are the homogeneous system

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{q})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{q})}{\partial y} = 0, \quad (6)$$

and the source term equation

$$\frac{\partial \mathbf{q}}{\partial t} = \mathbf{R}(\mathbf{q})_{friction}, \quad (7)$$

with  $\mathbf{R}(\mathbf{q})_{friction}$  as in equation (4). For each time step in the code equation (6) is advanced from the start of the time step,  $t_i$  to the end of the step  $t_{i+1}$  to give intermediate values of  $h_*$ ,  $hu_*$  and  $hv_*$  in  $q_*$ . The values in  $q_*$  are then used to solve equation (7) over the same time step. This introduces a ‘splitting error’ into the solution of order  $\Delta t = t_{i+1} - t_i$ , making the whole method only first order accurate. A more accurate splitting method (‘Strang splitting’) is available for implementation in the code, but the first order accuracy has been found to be sufficient in practice for a wide range of applications - e.g [9] chapter 17 or [10].

In the code, the momentum source term is only added in areas where the water depth is below a threshold, which can be set by the user. This reflects the fact that friction is likely to have a negligible effect in deep water. There is also a smaller threshold depth below which friction dominates, and momentum is set to zero. For all the simulations reported here, the default Clawpack values were used for these thresholds ( $1 \times 10^6 \text{m}$  and  $1 \times 10^{-30} \text{m}$  respectively). For each time step the code checks for these conditions in each grid cell and, where appropriate, applies the friction source term by

- extracting intermediate momentum terms calculated from the solution of the homogeneous equation (6),  $hu_*$  and  $hv_*$ ;
- calculating  $\gamma$  as in equation (5);
- calculating reduced momentum terms by the numerical solution of equation (7), using a backward Euler scheme. For small values of  $\gamma \Delta t$  this gives

$$\begin{aligned} hu &= \frac{hu_*}{1 + \gamma \Delta t}, \\ hv &= \frac{hv_*}{1 + \gamma \Delta t}. \end{aligned} \tag{8}$$

The reduced momentum terms are then returned to the code, which solves equation (6) for the next time step. An example of a simulation run with two different values for  $n$  is shown in section 2.3.1.

### 1.1.2 Incoming water

For inundation simulations, water entering the domain of interest can be modelled as a source term. In this section we describe a new inflow source term developed to model river-like flow. In operational situations, information regarding this source term may be available from an upstream gauge as a mass flow rate,  $Q$  measured in  $\text{m}^3 \text{s}^{-1}$ . In an ungauged catchment, the same information could be generated using a rainfall run-off model. A water mass flow rate can be turned into a source term,  $S$  expressed in  $\text{ms}^{-1}$  (c.f. the term ‘Sce’ in [7] p.31, which has the same units and can include rainfall, infiltration etc) as long as the area of the domain or ‘footprint’ over which the water is added is known. For water added over an area  $A$ ,  $S = Q/A$ . Now, an equation like (7) needs to be solved within the fractional stepping framework as described in section 1.1.1. The equation to be solved is

$$\frac{\partial \mathbf{q}}{\partial t} = \mathbf{R}(\mathbf{q})_{inflow}, \quad (9)$$

where

$$\mathbf{R}(\mathbf{q})_{inflow} = \begin{bmatrix} S \\ 0 \\ 0 \end{bmatrix}. \quad (10)$$

Equations (9) and (10) show that for each time increment  $\Delta t$ , the change in  $h$  due to the incoming water will depend on the value of the inflow source,  $S$ , over the same  $\Delta t$ . The extra water arriving in the domain creates extra water height, and is assumed here to arrive without any momentum; the water is subject only to hydrostatic momentum effects. An inflow source term has been implemented in the code and works in the following way

- determine in which grid cells the source term will be applied, and the total area which they cover in the domain,  $A$ . For a given mass flow rate  $Q$ , calculate  $S$  for each value of  $Q$  by dividing by  $A$ ;
- at the relevant grid points extract depth as calculated from equation (6),  $h_*$ ;
- calculate the change to  $h_*$  due to incoming water from a discretisation of (10) using a Crank-Nicholson scheme

$$h = h_* + \Delta t \frac{S(t) + S(t + \Delta t)}{2}; \quad (11)$$

- use the new value of  $h$  from equation (11) to solve for the next time step.

In this work, hourly values of  $Q$  and therefore  $S$  were used. The values of  $S(t)$  and  $S(t + \Delta t)$  were found by linearly interpolating between the hourly values.

### 1.1.3 Combining friction and inflow source terms

The source terms described in this section are applied in a sequential manner in the code. For each time step, the inflow source term calculates the new water depths in the relevant parts of the domain and then the friction source term is applied to the new water depths.

## 1.2 Boundary conditions in Clawpack

Correct specification of the solution at the boundaries of the computational domain is vital for the stability of any numerical scheme. To achieve this, Clawpack adds a user-specified number (2 by default) of ‘ghost cells’ next to each cell at a domain boundary. The domain is effectively extended in all directions by the addition of

ghost cells. The behaviour of the solution at the boundaries then depends strongly on the values of calculated model quantities in the ghost cells.

Clawpack allows for different treatment of domain boundaries, which work by setting the values of  $q$  in ghost cells beyond the edge of the computational domain. There are four options in the code for specifying the boundary conditions. For the SWE the available boundary conditions are:

- Solid wall. The momentum of the water in the ghost cells is effectively reflected about the boundary, while the water depth is extrapolated as in the non-reflecting outflow case.
- Periodic. This is for situations where the behaviour of the system is periodic so that all water leaving the system at one edge re-enters at the opposite edge. The values for  $q$  in the inflow ghost cells are therefore copied from the cells at the same distance from the outflow boundary.
- Non-reflecting outflow (extrapolating). This type of boundary acts as a free boundary, so that water can flow across the it without any (minimal) spurious reflections. This is most useful when the boundary of the computational domain does not have any physical significance but is an arbitrary edge of the domain of interest. In this case, the values of  $q$  are extrapolated from the cell next to the boundary into the ghost cells at each time step. This acts as a no-flow boundary in the case that the water is not flowing, but for non-zero velocity water flows over the boundary and leaves the domain. This is called a ‘zero order’ extrapolation in [9], and can be thought of as letting the numerical scheme collapse to a purely upwind scheme at the boundary. A first order extrapolation scheme can also be used in the code, but this has been shown to cause instabilities [9].
- User specified boundary conditions. These could be e.g. outflow conditions, if the flow across a boundary is known, or a boundary water depth condition if that is a known parameter for the system.

### 1.2.1 Topography at the boundary

The behaviour of water at the boundaries of the domain is highly dependent on the boundary conditions as described above. Another important factor is the representation of the domain topography at and across the domain boundaries. In the code used here, the value of the domain elevation is by default copied from the cells next to the boundary into the ghost cells. This represents a situation where there is no slope in bathymetry or topography across any boundaries. This is an adequate description of the left and right boundaries of the domains used here, since the domain is designed to be large enough in  $x$  that no water is expected to flow across these boundaries. The default no-slope condition is also used at the upstream boundary here, since very little water is likely to flow across it. However, at the downstream boundary the situation is very different; lots of water flows across the downstream boundary as it leaves the domain. A more physically realistic situation for the downstream boundary is therefore to extrapolate the slope of the domain into the ghost cells at the boundary. Changes have been made to the code to accommodate this, and it is possible to treat the downstream topography

slope as a model parameter, as in [5]. The effect of varying the parameter controlling the bathymetry slope at the boundary for a simple simulation can be seen in section 2.2.

## 2 Sensitivity studies

The following sensitivity studies were carried out in a domain with topography as shown in figure 2 with grid-spacing of approximately 10m in both the  $x$  and  $y$  directions.

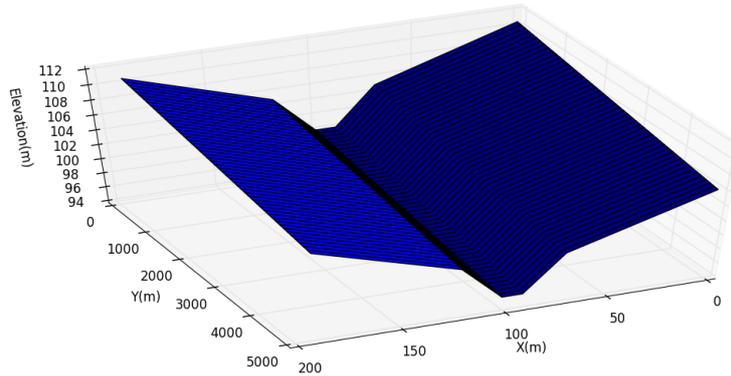


Figure 2: Elevation of the test domain in metres.

The upstream-downstream slope for the river and flood plain is 0.09% and the slope from the outside of the domain down to the river is 0.8%. The maximum depth of the channel below the bank is 8.5m and its width is 50m; these are values based on measured cross sections from the river Severn.

For the results shown in section 2, the upstream boundary of the domain was set to be a solid wall to avoid water being able to leave the domain in that direction, and also to avoid water effectively being generated in the ghost cells. The other three boundaries are free (extrapolating) boundaries with an extrapolated slope for the downstream boundary.

### 2.1 Channel friction

In the following simulations, the domain shown in figure 2 was initially empty. Water was added into the domain close to the upstream boundary at a rate of  $160m^3s^{-1}$  for a total simulation time of 3000 seconds. A gauge measuring depth (relative to topography) was simulated in a central position as shown in figure 3.

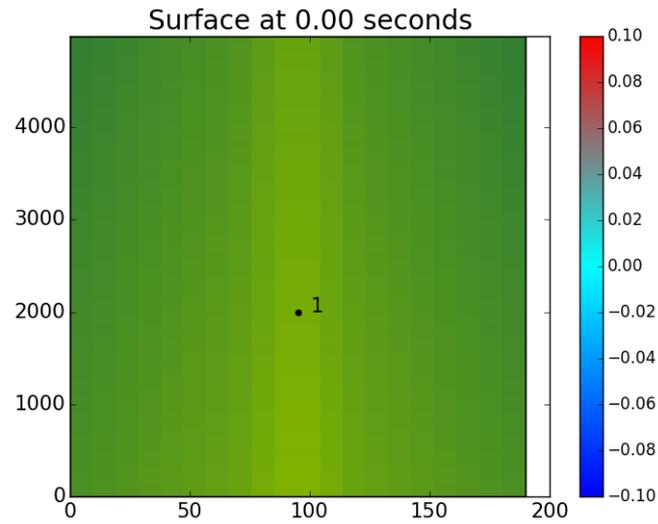


Figure 3: Location of gauge in domain marked with black dot and '1'. There is no water in the domain at the time shown; the green colours correspond to 'dry land' and are not shown on the colourbar.

The gauge recorded depth throughout the simulations, which were carried out for three different values of Manning's friction coefficient,  $n$  in the domain. The friction values used were  $n = 0.02$  (a reasonable estimate for a river channel),  $n = 0.002$  and  $n = 0.07$ . For each case the friction coefficient was the same in the whole of the domain - i.e the value of  $n$  in the rest of the domain was the same as in the channel. The results are shown in figure 4.

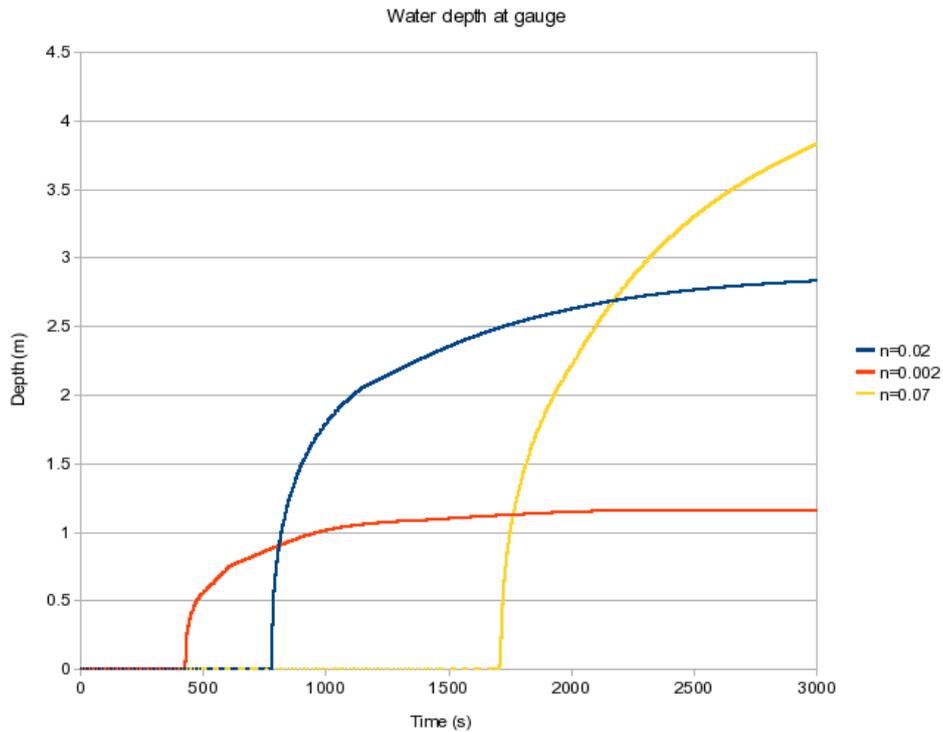
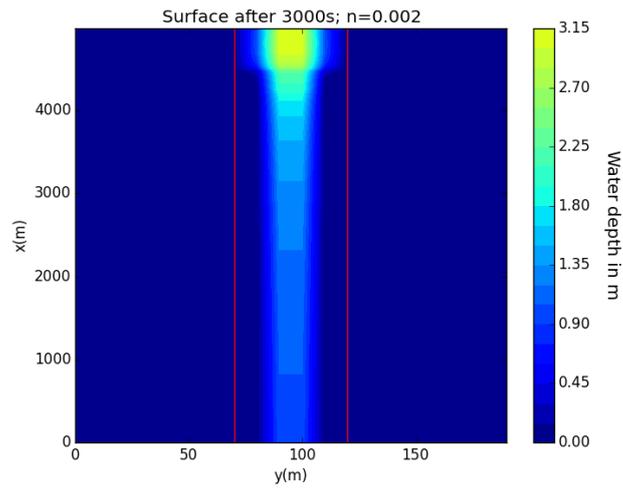
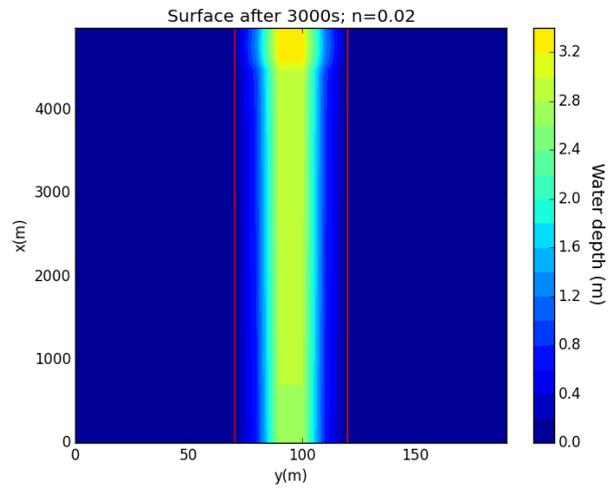


Figure 4: Water depth at gauge with time for different channel friction parameters

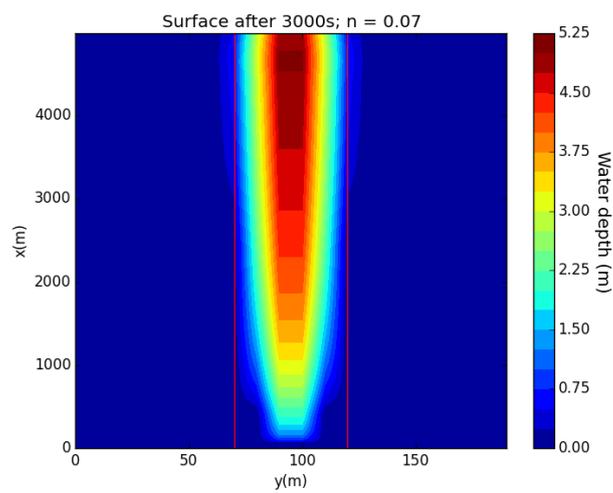
It can be seen that the channel friction parameter influences both the time taken to reach equilibrium, and the equilibrium depth in the channel, where the channel is defined as the central 5 grid cells in the  $x$  direction for all  $y$ . The final (after 3000 seconds) water levels are shown in plan view for the three different values of  $n$  in figure 5. The edges of the channel are shown by the thin red lines and we can see in figure 5c that in the case in which  $n = 0.07$ , some flooding of the domain took place (where  $x > 3000\text{m}$ ). This is because under these conditions water travels so slowly in the channel that it is forced onto the banks. In all of the cases shown here the Manning's friction coefficient on the flood plain was the same as in the channel; making these values different will clearly affect the evolution of any inundation event.



(a)  $n = 0.002$



(b)  $n = 0.02$



(c)  $n = 0.07$

Figure 5: Water depth in the domain at the end of the simulation for three different values of  $n$  in the channel. The domain is shown in plan view. Note the different scales for the colourbars.

## 2.2 Slope at downstream boundary

The effect of changing the parameter controlling the bathymetry slope at the downstream boundary can be seen in the following simulations. In each case, a symmetrical domain with a central channel was initially filled with water to a depth of approximately 4m as shown in cross section in figure 6. The domain has a slope of 0.09% which caused the water to flow downhill under the influence of gravity and leave the domain at the downstream end. No extra water was added to the domain during the simulations, which ran for 1500 seconds. Some water remained in the domain at the end of the simulations. For one case the downstream slope was extrapolated into the ghost cells, and for the other case the ‘no slope’ condition was used, where the elevation in each ghost cells is set to be the same as the domain cell next to it on the inside of the boundary.

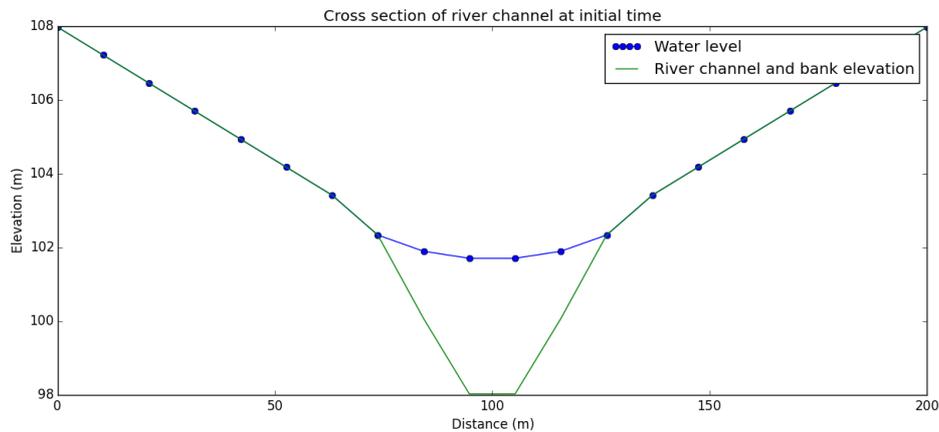
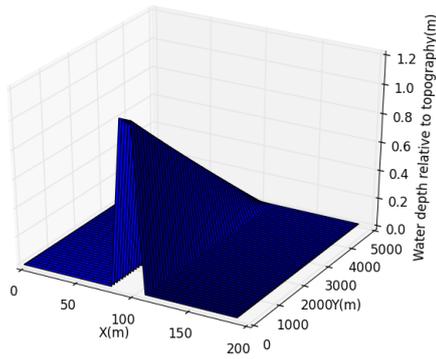


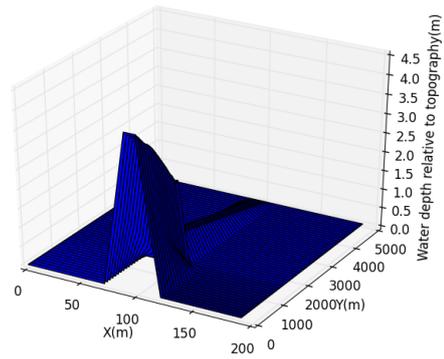
Figure 6: Cross section of the domain, showing the channel filled with water at the start of the simulations.

The green line shows the elevation of the domain and the blue points show the water depth.

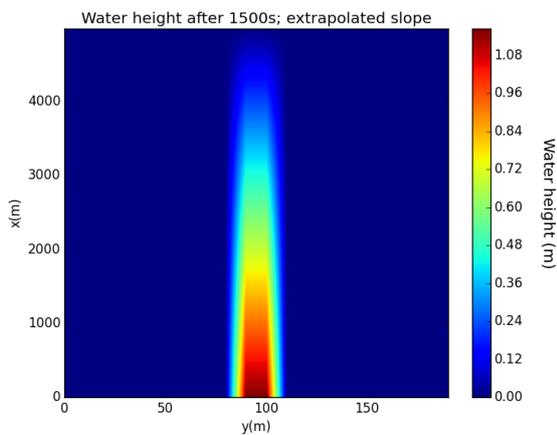
At the end of the simulations, the water profile is different for the two cases as shown in figure 7. For the extrapolated boundary slope condition, the water is able to leave the domain cleanly; the flow over the boundary is the same as elsewhere in the domain. For the no slope condition, water cannot leave the domain as fast and therefore builds up at the boundary.



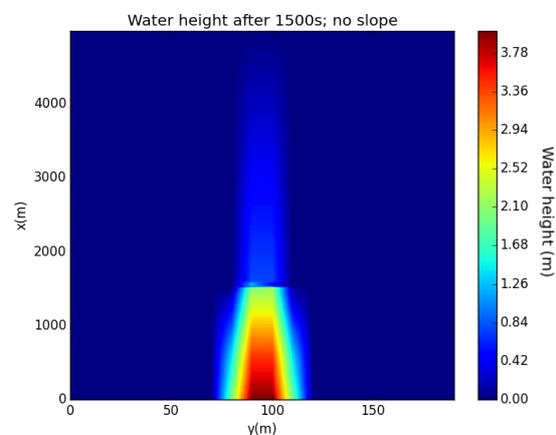
(a) Extrapolated slope; water levels relative to topography in 3D



(b) No slope; water levels relative to topography in 3D



(c) Extrapolated slope; water levels relative to topography in plan view



(d) No slope; water levels relative to topography in plan view

Figure 7: Water profiles for the slope and no slope boundary conditions after 1500s.

### 3 Assimilation experiments

Data assimilation (DA) is a powerful mathematical technique for combining a numerical model of a system with observations in order to best estimate the current true state of the system. This best estimate can then be used as the starting point for forecasting the behaviour of the system into the future. There are various different DA methods; here we use an ensemble method in which uncertainties in the system are represented by an ensemble of states.

The Ensemble Kalman Filter (EnKF), as used here, was introduced by Evensen [4] and comprises two steps: a forecasting (or prediction) step and an update step in which observations of the state are used to produce a ‘corrected’ state called the analysis. An ensemble of state vectors (in this case representing water levels in the test domain) is used to represent a statistical sample of the forecast or analysis uncertainty. In the forecasting step, each ensemble member is evolved forward in time using the forecast model (here, the numerical SWE

model), and in the update step the ensemble is updated to take account of observations of the state at that time. The EnKF has various formulations, and here we have used an Ensemble Transform Kalman Filter as described in [11].

A simplified river valley-like domain has been used to carry out some initial DA experiments. The domain is slightly modified from the version shown in section 2 but has a similar shape; it is a gently sloping symmetrical valley, 5000m by 250m with a 50m wide central river channel as shown in figure 8. The channel is defined to be the central 5 grid cells in the  $x$  direction for all values of  $y$ ; the rest of the domain is defined as the river bank. The domain is also shown in cross section in figure 9. The domain has an upstream-downstream slope of 0.08% and the slope of the floodplain towards the river is 0.8%. All of the boundaries are extrapolating boundaries for the results shown in section 3. At the downstream boundary the topography slope is extrapolated; the other boundaries have the topography no-slope condition applied.

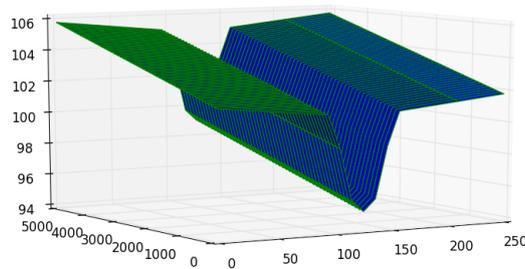


Figure 8: Elevation in m of the test domain used in assimilation experiments.

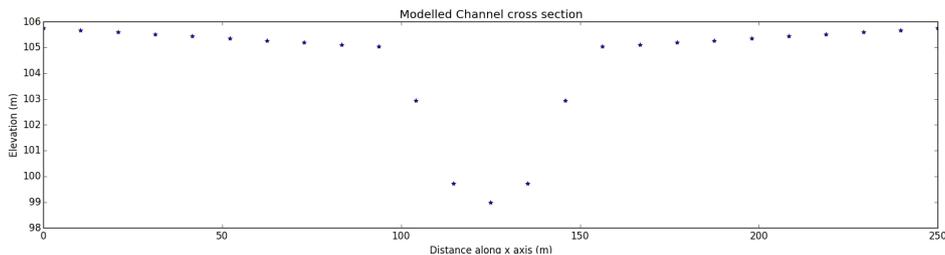


Figure 9: Cross section of the test domain shown in fig.8 and used in assimilation experiments.

### 3.1 Identical twin experiment

Python code has been written to perform data assimilation for inundation output from Clawpack using an ensemble Kalman filter; identical twin experiments have been carried out. Identical twin experiments are commonly carried out with data assimilation systems, e.g. [8], in order to test the DA system as well as to

generate information about the model to which the DA is applied. In such experiments, a numerical model (here the inundation model generated with Clawpack) is used to generate a ‘truth’ output for a set of known initial conditions and model parameters. The truth run starts at time  $t_0$  and runs to the end of the period of interest at time  $t_{final}$ . The model is then run again from time  $t_0$  with perturbed initial conditions and/or model parameters, to generate a forecast state at a time  $t_1 (> t_0)$ . Synthetic observations are taken from the truth run at  $t_1$  and combined with the forecast at  $t_1$ , to produce an improved state at called the analysis. The model is then run forward in time from  $t_1$  with the analysis as the initial condition. This is repeated for several times between  $t_0$  and  $t_{final}$ . In the case of an ensemble DA system, an ensemble of states are produced at each time step using a distribution of parameter values and/or initial conditions, and the ensemble mean is treated as the forecast state at each time step.

### 3.1.1 Model inputs

An ensemble of inflows was created to drive the truth and ensemble members. The inflow ensemble was generated by adding time-correlated errors to measured hydrograph data from the river Severn during the flooding event in 2007, as described in [5]. Water coming into the domain was then modelled using an inflow source term added to the code as described in section 1.1.2. The inflow was time-varying (one measurement per hour for 4.5 days). The first four hours of the simulation have a constant inflow equal to the first measured inflow value; this four hour period is considered as a ‘spin-up’ step where the system fills with water. The inflows with time, including the spin-up period, are shown in figure 10.

### 3.1.2 Truth run

Clawpack was used to generate a ‘truth’ for the domain shown in figures 8 and 9 with a ‘truth’ inflow, arbitrarily selected from the ensemble of inflows described in section 3.1.1. Inflow was applied inside the upstream edge of the domain. The friction in the channel was described with a Mannings friction coefficient of 0.04, which is the estimate given for a natural stream in [12]. The resulting inundation for the truth run was severe, with three peaks corresponding to the peak inflows shown in the hydrograph data in figure 10.

### 3.1.3 Forecast run

The forecast was generated with an ensemble made up of 100 members. Each ensemble member was driven by a different inflow and with a different value of Mannings friction coefficient in the channel. All other conditions are the same for each member (and the truth); in all cases the value of  $n$  outside the channel was 0.04. The channel friction coefficient for each ensemble member was taken from a Gaussian distribution centred on 0.05, a higher value than the truth, and standard deviation 0.01. There is a small risk of generating negative values of  $n$  in this way, which could be avoided using a method as described in [14]. In fact the value of  $n$  appears as  $n^2$  in the model, so no unphysical effects will occur if this happens.

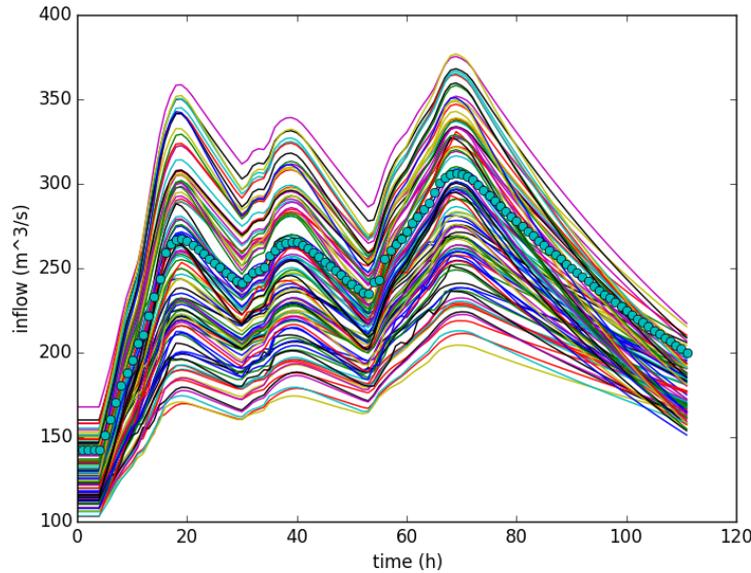


Figure 10: Inflow in  $m^3s^{-1}$  with time for the truth (circles) and ensemble members. The truth inflow was arbitrarily chosen from a distribution based on hydrograph data. Each inflow was made constant for the first 4h as a spin-up period.

### 3.1.4 Synthetic observations

Synthetic observations of water levels in the domain were taken from the truth run starting at 16 hours simulation time and then at intervals of 12 hours. Different numbers of observations were assimilated in the two cases presented here and no noise was added to the observations in either case. In case 1, 10,000 observations were used at each assimilation step, with observations of the water levels taken at almost every grid cell in the domain; only the area close to the inflow was excluded. In case 2, far fewer observations were taken, with 24 observations used at each assimilation step. In case 2 the 24 observations were all from the river bank, none were observations of water depth in the channel. This reflects the fact that operationally, satellite-derived observations of flood extent are intersected with a digital terrain map in order to get water levels in the domain for assimilation [5]. Digital terrain maps do not generally include information about river beds, so only water levels from normally dry areas are included in this way.

## 3.2 Results and discussion of assimilation

The results presented here are only for one realisation of the system - i.e. for one set of perturbed inflows and friction parameters. In order to draw firm conclusions the experiment would need to be performed a number of times, but initial results after one run are discussed here. For case 1, the EnKF performed well at each assimilation step, resulting in analysis water levels which closely matched the truth, as expected. The EnKF also performed well in case 2. Figures 11 to 13 show the results for a representative step of the assimilation cycle

for case 2. The figures show results from the 7th assimilation step (88h from the beginning of the simulation). Figure 11 shows the water levels for all forecast ensemble members and the truth. Some of the ensemble members predict severe flooding at this time, with water present 100m from the river bank and beyond. Other ensemble members predict no flooding. Figures 12 and 13 show the truth along with the forecast ensemble mean and the analysis ensemble mean. The forecast in figure 12 is the mean of the forecast ensemble shown in figure 11. The analysis is the mean of the analysis ensemble, and lies much closer to the truth than the forecast. Figure 12 shows the truth, forecast and analysis in cross section at  $y=2000\text{m}$ . The analysis and forecast both show a dip in water levels between  $x=100\text{m}$  and  $x=150\text{m}$ , coincident with the river channel. This is not a physical effect but due to some forecast ensemble members predicting relatively low water depths, as seen in figure 11. These reduce the mean forecast in this area. Outside of the channel, the water levels are constrained by the bank. This shape is also present in the analysis because the assimilation does not add any information about the depth in the channel; all the observations are on the river bank.

Figure 13 shows the differences between the truth and the forecast (fig. 13a) and between the truth and the analysis (fig. 13b) for the whole domain in plan view at  $t=88\text{h}$ . Figure 13b shows the difference in metres between the analysis and the truth in every grid cell. The biggest differences can be seen in the channel, this is because no observations were taken from this part of the domain. Outside of the channel, the analysis shows very small differences from the truth. Figure 13a shows the difference between the forecast and the truth, these differences are bigger than those in the analysis case. This means that the assimilation is moving the forecast closer to the truth, as expected.

Figure 14 shows the behaviour of the ensemble over time for case 1 (stars) and case 2 (circles). The forecast represents the prediction of the model before assimilation and the analysis represents the prediction of the model after assimilation of observations. Figure 14a shows the root mean square error (RMSE) between the forecast (ensemble mean) and the truth at each time when an assimilation was performed. Figure 14b shows the RMSE between the analysis (ensemble mean) and the truth at the same times. Note the different scales for the forecast and analysis RMSE graphs. The analysis is closer to the truth than the forecast at all cases, resulting in smaller RMSE values for the analysis. As expected, the analysis in case 1 is closer to the truth than in case 2; more observations of the system lead to a better match between the analysis and the truth. There is no significant difference between the quality of the forecast between case 1 and case 2. This is likely to be due to the fact that the time between assimilation steps is long compared to the time it takes water to flow through the test domain. Assimilating observations more frequently, altering the test domain so that the water flows more slowly (e.g. by introducing a shallower slope or higher friction coefficient) or considering a longer test domain would all likely make the forecast ability of the assimilation more long lasting by increasing the effective time for the observations [3].

Figures 14c and 14d show the trace of the covariance matrix for the forecast and analysis respectively, which is a measure of the spread of the ensemble. A larger trace indicates a larger spread in the ensemble and this corresponds to a larger variance in the predicted water levels. The spread of the analysis ensemble is very

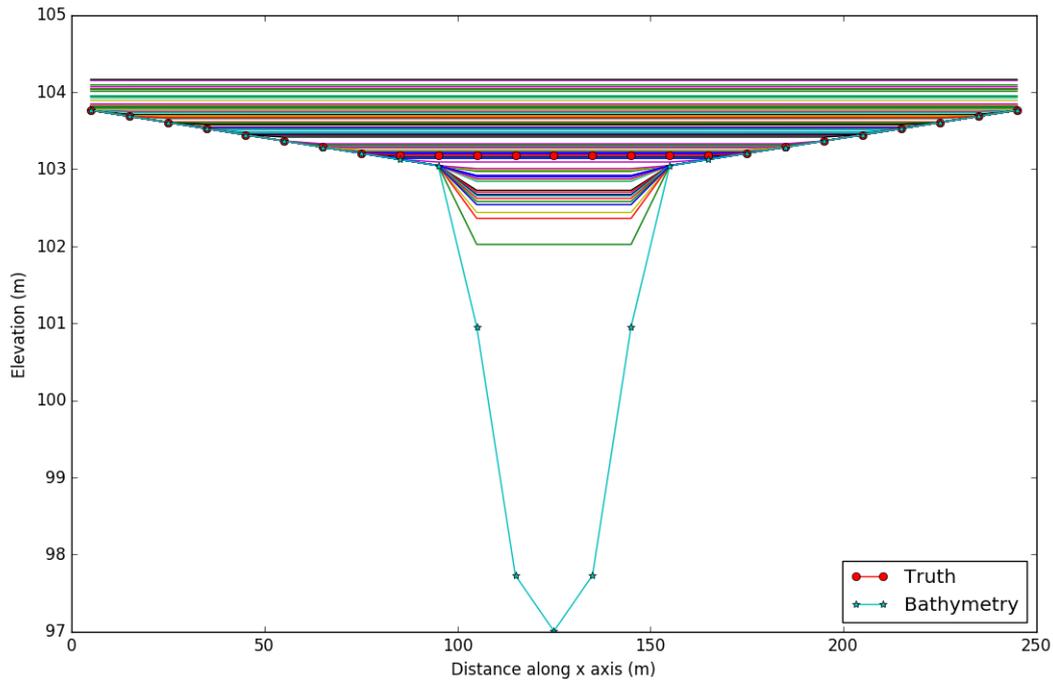


Figure 11: Cross section of the domain at  $y=2000\text{m}$ . Each line represents the water levels predicted by a different ensemble member. The water levels from the truth run are marked by red circles.

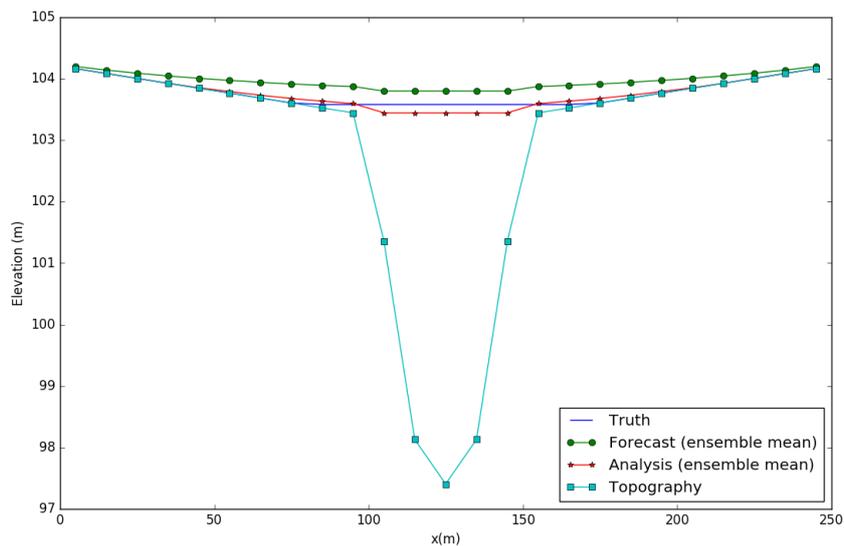
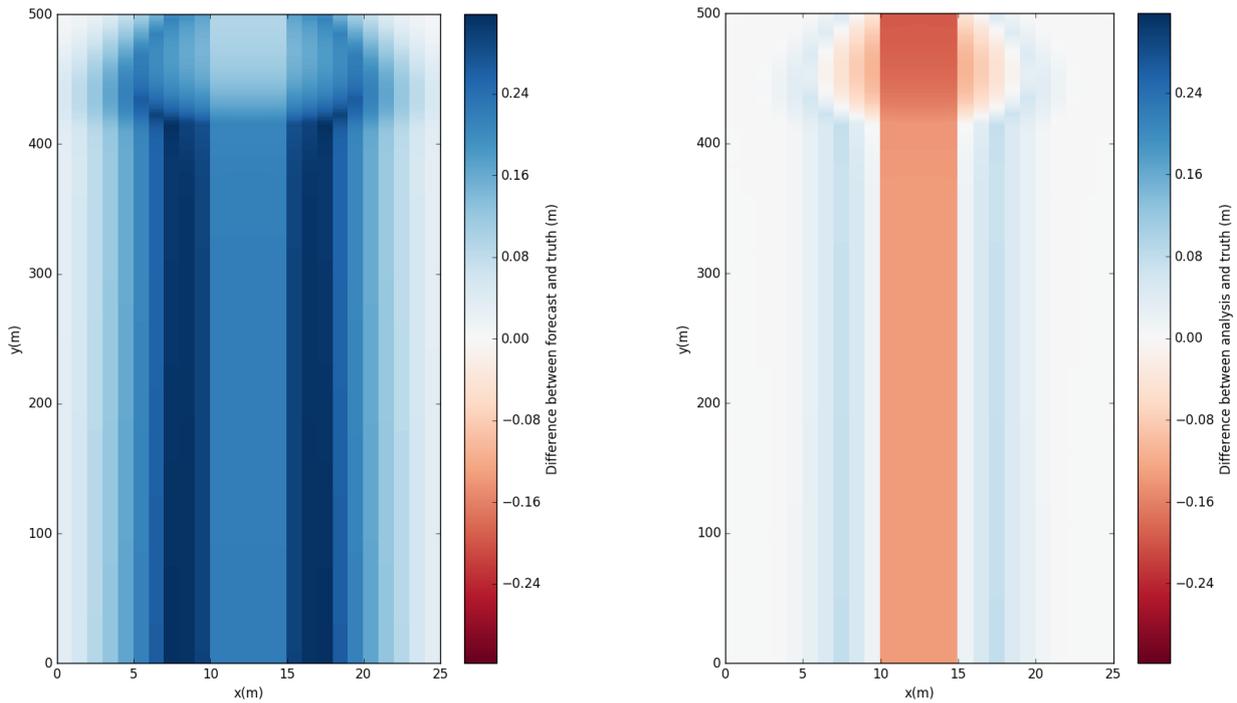


Figure 12: Cross section of the domain at  $y=2000\text{m}$  showing the truth, forecast and analysis water levels.

small, particularly in case 1. However, by the time of the next assimilation step, the spread of the ensemble has increased. This is because each ensemble member has a different inflow and a different friction coefficient, so that the ensemble members move away from each other over time during the forecasting step. Performing



(a) Difference between forecast and truth water levels at  $t=88h$       (b) Difference between analysis and truth water levels at  $t=88h$

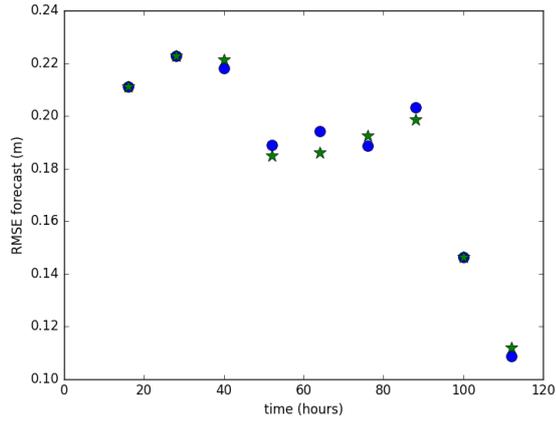
Figure 13: Difference between the forecast and the truth and the analysis and the truth in the whole of the domain. The analysis can be seen to be closer to the truth than the forecast.

joint-state parameter estimation for these parameters and updating them at each assimilation step along with the water levels would reduce the spread of the forecast ensemble and is likely to improve the quality of the forecast.

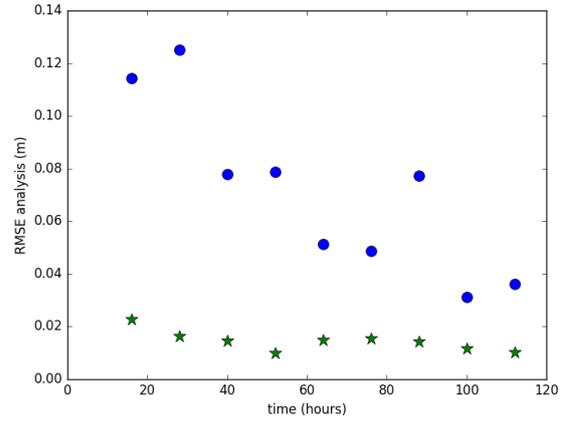
### 3.3 Conclusions

Applying data assimilation in a fluvial flooding situation has the potential to greatly improve flood forecasting, as shown in [5] using the flood modelling software Lisflood-FP. In this work, Clawpack code has been used to generate a simple river flood model by adding an inflow source term and a river-valley like topography to existing code. Clawpack is open source, robust software which uses sophisticated Riemann solvers to solve systems of equations such as the SWE.

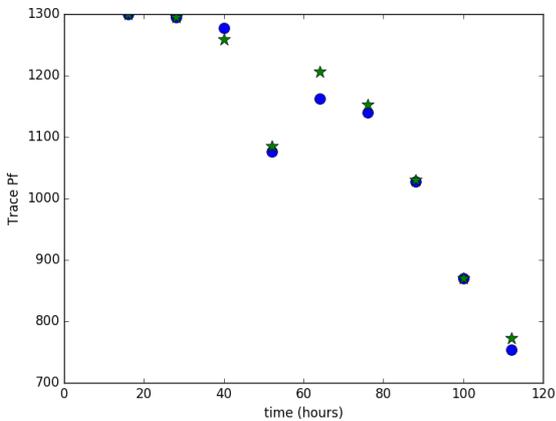
The effect of varying the channel friction parameter in the domain has been investigated, using a constant inflow. Increasing friction acts to slow down the water in the domain, as would be expected. Higher friction causes the system to take a longer time to reach equilibrium in a constant inflow situation, and makes the equilibrium depth larger. The effect of the slope at the downstream boundary has also been investigated in a simple river valley model, by letting water drain out of the domain (with no inflow added). Extrapolating the river bed slope across the boundary of the domain was shown to give physically realistic results; the water



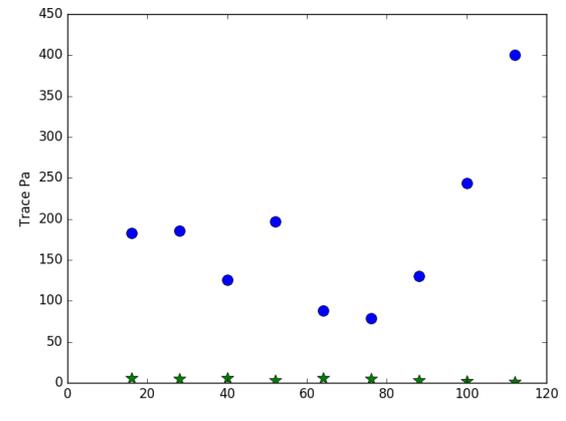
(a) RMSE of forecast and truth at each assimilation time



(b) RMSE of analysis and truth at each assimilation time



(c) Trace of the forecast covariance matrix at each assimilation time



(d) Trace of the analysis covariance matrix at each assimilation time

Figure 14: Results of the EnKF applied to an inundation model over 112 hours. Circles show the results of using 24 observations at each assimilation step. Stars show results of using 10000 observations per assimilation step.

flowed across the boundary with no build-up.

Initial identical twin DA experiments with a simplified domain show that the ETKF and observations can be used to greatly improve the water levels predicted by the model ensemble in an inundation situation. However, the greater forecast ability of the ensemble for the many observation case does not persist between assimilation steps. We believe this is because the water flows through the domain in a short time (approximately 1 hour) compared to the time between observations (12 hours), which means that the corrected water levels after an assimilation do not strongly influence the water levels by the next assimilation time. A domain with a less severe slope, or a higher friction coefficient would lead to the water moving more slowly, and likely improve the forecast ability of the ensemble. More frequent observations in the current domain would be likely to have the same result. Performing joint state-parameter estimation in order to retrieve values for the distributed model

parameters and inputs (here, channel friction and inflowing water) would also likely improve the performance of the ETKF for this domain.

## References

- [1] P.D Bates and A.P.J De Roo. A simple raster-based model for flood inundation simulation. *Journal of Hydrology*, 236(12):54 – 77, 2000.
- [2] Clawpack Development Team. Clawpack software, 2014. Version 5.2.2. <http://www.clawpack.org>.
- [3] S.L. Dance. Issues in high resolution limited area data assimilation for quantitative precipitation forecasting. *Physica D: Nonlinear Phenomena*, 196(12):1 – 27, 2004.
- [4] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical research*, 99(C5):10142–10162, 1994.
- [5] Javier Garcia-Pintado, David Mason, Sarah L. Dance, Hannah Cloke, Jeff C. Neal, Jim Freer, and Paul D. Bates. Satellite-supported flood forecasting in river networks: a real case study. *Journal of Hydrology*, 523:706–724, 2015.
- [6] Jean-Michel Hervouet. Telemac modelling system: an overview. *Hydrological Processes*, 14(13):2209–2210, 2000.
- [7] Jean-Michel Hervouet. *Hydrodynamics of Free Surface Flows*. John Wiley and Sons Ltd, 2007.
- [8] W. Lahoz, B. Khattatov, and R. Menard. *Data Assimilation: Making Sense of Observations*. Springer Berlin Heidelberg, 2010.
- [9] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.
- [10] Randall J LeVeque. Wave propagation algorithms for multidimensional hyperbolic systems. *Journal of Computational Physics*, 131(2):327–353, 1997.
- [11] David Livings. Aspects of the Kalman filter. *MSc thesis, University of Reading*. <http://www.reading.ac.uk/web/FILES/math/Livings.pdf>, 2005.
- [12] D.R. Maidment and L.W. Mays. *Applied Hydrology*. McGraw-Hill series in water resources and environmental engineering. Tata McGraw-Hill Education, 1988.
- [13] HEC-RAS Development Team. Hec-ras software. <http://www.hec.usace.army.mil/software/hec-ras/>.
- [14] M. Verlaan. Parameter estimation for a global tide and storm-surge model. *Conference abstract, Adjoint workshop June 2015*.