FAST EARLY FLOOD WARNING SYSTEMS EXPLOITING CATCHMENT SPECIFIC BEHAVIOR

S. Rusca  
*Swiss Federal Institute of Aquatic Science and Technology, Eawag, Dübendorf, Switzerland*

J. P. Carbajal  
*Swiss Federal Institute of Aquatic Science and Technology, Eawag, Dübendorf, Switzerland*

1 INTRODUCTION

Floods due to heavy rain are among the most destructive events in hydrology and their frequency has incremented over the last decades. Numerical models are useful tools to predict floods and trigger security measures or evacuations. However, the high computation demands of these models hinder their use in real time early warning systems.

Fine tuned shallow water based simulators are able to produce trustworthy flood predictions but are unusable as early warning systems due to their high computational cost. Early warning systems should be capable of quickly providing a prediction of whether, based on current conditions and meteorological forecast, an occurring rain event will result in major flooding or not, and if yes within how much time. Fast surrogate models based on detailed simulators can provide accurate results\(^1\) with speed-up factors of up to \(10^6\) [see 1, for an application in geophysical hazards].

In this work, we present the development of a surrogate model of a detailed simulator using *FullSWOF_2D-v.1.07.00* [2] (non-linear shallow water equations).

2 METHODOLOGY

The general goal of our emulator is to take inputs related to a rain event and catchment situation, and predict the time it will take to observe a catchment discharge greater or equal to a predefined threshold discharged \(Q_t\). The threshold discharge is provided by a downstream region that is prone to flooding and the quantity of interest (QoI) is the time-to-threshold, \(t_t\).

\[Q(t, I, \Delta \theta, d) \approx Q_{\infty}(t, I, \Delta \theta) \Xi(t \leq d)\]  

where \(Q\) is the response hydrograph, \(t\) is the time variable, \(d\) is the rain event duration, and \(\Xi(x)\) is the indicator function of the interval \([0, d]\). This representation does not recover the recession of the hydrograph, but that is irrelevant for our application. Fig. 2 shows an example response obtained from the simulator. Recession of the discharge can

\(^1\)Note that accuracy is task dependent
be observed immediately after the end of the rain event.

Figure 2: Response hydrograph for $\Delta \theta = 0.5$ and $I = 32.2 \text{mm} \text{h}^{-1}$. $Q_1$ and $Q'_1$ lines highlight the discontinuous behavior of our QoI.

The discharge for infinite rain duration $Q_\infty(...)$ determines the QoI $t_1$,

$$Q_\infty(t_1, I, \Delta \theta) - Q_1 = 0$$

(2)

Although $Q_\infty$ is not available, simulations using sufficiently long rain events (6h in our case) are enough for the application at hand. This is because rain intensity is likely to change within 6h, and because high intensity rains of such duration can be considered extreme for many regions.

3 PRELIMINARY RESULTS

The QoI we are trying to predict, shows a discontinuous behavior. By observing Fig 2 this is particularly true in the region between $Q_1$ and $Q'_1$. This phenomenon makes $t_1$ very difficult to predict. Development of an ad hoc emulator represents a valid solution to the problem.

Different types of interpolators were tested and their performance was evaluated on the test points. Validation points were used in order to have an unbiased assessment of the emulator performance.

For low $I$ and $\Delta \theta$ the chosen $Q_1$ is never reached. This is shown in the figure with the orange frontier: some ($I$, $\Delta \theta$) combinations produce no $t_1$ prediction.

4 ACKNOWLEDGE

The authors would like to thank Prof. Peter Molnar and Dr. Jörg Rieckermann for their support and guidance during S. Rusca’s master thesis and the writing of this article. We thank the developers of GNU Octave and Inkscape for their excellent software tools, which were used for this article.

Source code is available at bitbucket.org/binello7/master_thesis.

Funding The research leading to these results has received funding from the Eawag’s discretionary funding program (project EmuMore kakila.bitbucket.io/emumore).

Author contributions SR developed the software, carried out simulations, and data analysis. JPC contributed to the emulator design, data analysis, and supervised SR. All authors contributed to the writing of this manuscript.

REFERENCES


