

# Developing generic hydrodynamic models using artificial neural networks

## Developpement de modèles hydrodynamiques génériques au moyen de réseaux neuronaux

YONAS B. DIBIKE, *International Institute for Infrastructural, Hydraulic, and Environmental Engineering, P.O. Box 3015, 2601 DA Delft, The Netherlands*

### ABSTRACT

Possibilities for the development of a new modelling paradigm, namely allowing models to 'construct themselves' by learning from existing numerical-hydraulic models, was investigated by extending previous works to encompass schemes that can be applied over arbitrary bathymetries with variable distances and time steps. For the simplest possible cases of one and two dimensional flow problems considered in this study, the relatively elementary technology of artificial neural network was found to provide acceptable results. Moreover, It was demonstrated that the well-trained networks could be substituted in place of the finite difference schemes in the hydrodynamic model formulation and could perform like numerical operators. This new paradigm is intended in future to supplement, and even in some instances to replace the current one

### RÉSUMÉ

On étudie ici la possibilité de développer un nouveau paradigme de modélisation – à savoir donner aux modèles la possibilité de 'se construire eux-mêmes' en se réglant sur des modèles hydrauliques numériques existants – en étendant des travaux antérieurs à la prise en compte de bathymétries arbitraires et de pas d'espace et de temps variables. Pour les cas simples d'écoulements unidimensionnels et bidimensionnels étudiés ici, il appert que la technologie relativement élémentaire des réseaux neuronaux donne des résultats acceptables. On montre de plus que des réseaux de neurones convenablement réglés peuvent être substitués aux schémas aux différences finies dans la formulation des modèles hydrodynamiques et qu'ils se comportent comme des opérateurs numériques. Ce nouveau paradigme pourrait, à l'avenir, compléter et, dans certains cas, suppléer, les approches traditionnelles.

### 1. Introduction to the context of this work

The problem of determining a fluid flow is usually divided into two stages. The first of these is concerned with a description of the flow of the fluid in such general terms that this description will hold at each and every point in the domain of the solution at all times. Such a description is said to be generic to the class of flows concerned. The result is either a so-called 'point description' such as a partial differential equation, or an 'interval description', such as an integral equation. The second stage of the problem is concerned with transforming this 'point' or 'interval' representation into a representation that is distributed over the entire domain of the solution at all times, such as is for example realised by the process of integrating a partial differential equation. The difficulties experienced in integrating over complicated domains has led to the widespread and now almost universal use of numerical methods in which point and integral descriptions are extended to finite spatial descriptions that are maintained over finite time intervals, thus providing solution procedures of finite cardinality.

However, recent developments in simulation modelling have caused this paradigm to be questioned. Numerical modelling is nowadays less and less an isolated activity, being increasingly integrated into other activities, such as field-measurement programs, data-assimilation facilities and automated calibration procedures. Similarly, such integrated systems are increasingly widely used to provide such services as real-time control and flood warning. All of these developments influence the specifications for modelling capabilities and especially on the side of the

shorter response times that are now required from the modelling systems.

Moreover, development in software engineering, and even in some branches of computer science, have led to the development of new methods, tools and environments that do not lend themselves readily to the existing numerical-hydraulic paradigm. These difficulties have led to the notion of moving over to a new modelling paradigm, that would pass better into current practices and controlling possibilities (Abbott, 1997). One essential component in this development is that of constructing new models more or less automatically from data which are obtained from existing numerical-hydraulic models and all available field measurements, while guiding future measurement programs in the direction of serving this class of learning processes.

### 2. The formulation of the classes of schemes within the existing paradigm.

The objective of the numerical-hydraulic modelling of a water system is to establish the different flow variables, such as water depths and velocities, as a function of time over the area of interest such that, knowing the initial state of the system and the different forcings, the state of the system at any future time can be simulated. The conventional way of achieving this objective is first to derive relations between flow variables from physical laws used to express the conservation of certain quantities such as mass, momentum and energy. This usually results in one or the other system of partial differential equations, which reasonably describes the flow phenomena. In the general case, these equa-

---

Revision received December 28, 2000. Open for discussion till August 31, 2002.

tions have no analytical solutions and have to be solved approximately by means of numerical methods, which results in values of the solution (flow variables) at discrete points in space and time. The way in which the computation proceeds from values of dependant variables at grid points at one time level to their values at the next time level depends on the computational scheme considered (Abbott and Basco, 1989; Abbott and Minns, 1998).

### 2.1 The simple explicit formulation used as a test case

Consider the primitive equation of de Saint Venant (1850) for 1D nearly horizontal, free surface flow in the Eulerian form:

Continuity

$$\frac{\partial h}{\partial t} + h \frac{\partial u}{\partial x} + u \frac{\partial h}{\partial x} = 0 \quad (1)$$

Momentum

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + g \frac{\partial h}{\partial x} = 0 \quad (2)$$

Where  $h$  is the water depth and  $u$  is the depth-averaged water velocity.

For very long waves with amplitudes that are relatively small as compared with the water depth, the advective acceleration term  $uu_x$  and the  $uh_x$  in the mass equation are of lower order than the other terms. In such cases, the primitive equations (1) and (2) are often used in a *linearised form*:

$$\frac{\partial h'}{\partial t} + h_0 \frac{\partial u}{\partial x} = 0 \quad (3)$$

$$\frac{\partial u}{\partial t} + g \frac{\partial h'}{\partial x} = 0 \quad (4)$$

where  $h = h_0 + h'$ , with  $h_0$  the still water depth and  $h' \ll h_0$  or, correspondingly,  $u \ll gh$ .

These equations can be simplified, in turn, to give the following equations in a single variable:

$$\frac{\partial^2 h'}{\partial t^2} - c^2 \frac{\partial^2 h'}{\partial x^2} = 0 \quad (5)$$

$$\frac{\partial^2 u}{\partial t^2} - c^2 \frac{\partial^2 u}{\partial x^2} = 0 \quad (6)$$

with  $c^2 = gh_0$

In two-dimensional flow, the continuity and momentum equations can be linearised in the same way to provide the equation:

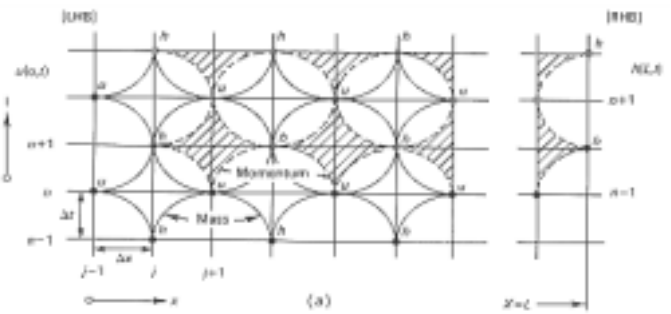


Fig. 1. Schematisation of a staggered grid (lepfrog method).

$$\frac{\partial^2 h'}{\partial t^2} - c^2 \left( \frac{\partial^2 h'}{\partial x^2} + \frac{\partial^2 h'}{\partial y^2} \right) = 0 \quad (7)$$

Consider now a staggered grid, three-time-level scheme, as depicted in Fig. 1. Using centred space and time differences, the representation in finite difference form of equations (3) and (4) is:

$$\frac{h_j^{n+1} - h_j^{n-1}}{2\Delta t} + h_0 \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0 \quad (8)$$

$$\frac{u_{j+1}^{n+2} - u_{j+1}^{n-2}}{2\Delta t} + g \frac{h_{j+2}^{n+1} - h_j^{n+1}}{2\Delta x} = 0 \quad (9)$$

Equations (8) and (9) can be solved explicitly for  $h_j^{n+1}$  and  $u_j^{n+1}$ , both of which variables are staggered in space and time. Such a three level representation is usually referred to as a *leapfrog scheme* and requires initial data at  $n$  and  $n-1$ .

A simple variation on Fig. 1 can provide a two-level scheme as shown in Fig. 2. In this case, initial values of  $u$  and  $h$  need to be known only at one time level. The corresponding finite difference equations can then be described as:

Continuity

$$\frac{h_j^{n+1} - h_j^n}{\Delta t} + h_0 \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0 \quad (10)$$

Momentum

$$\frac{u_{j+1}^{n+1} - u_{j+1}^n}{\Delta t} + g \frac{h_{j+2}^{n+1} - h_j^{n+1}}{2\Delta x} = 0 \quad (11)$$

We can solve (10) and (11) explicitly for  $h$  and  $u$  values which are staggered in space.

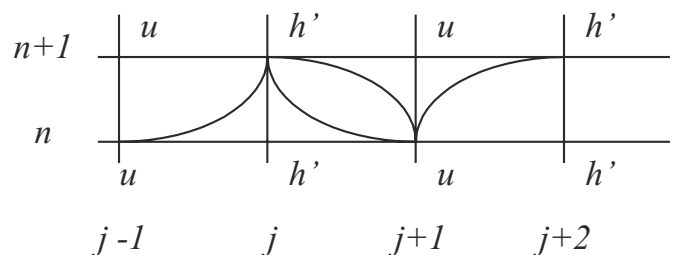


Fig. 2. A variation of the leap frog scheme on a staggered grid.

Equation (10) can be reformulated as

$$h_j^{n+1} = h_j^n - \frac{h_0 \Delta t}{2 \Delta x} (u_{j+1}^n - u_{j-1}^n) \quad (12)$$

Multiplying both the numerator and denominator of the second term in the right hand side of (12) by a gravity term  $g$  and substituting the square of celerity  $c^2$  for  $(gh_0)$ , (12) can be re-written in the following form:

$$h_j^{n+1} = k * u_{j-1}^n + l * h_j^n + m * u_{j+1}^n \quad (12a)$$

$$\text{where } k = \frac{Cr}{2} \sqrt{\frac{g}{h_0}}, \quad m = -\frac{Cr}{2} \sqrt{\frac{g}{h_0}}, \quad l = 1 \quad \text{and} \quad Cr = c \frac{\Delta t}{\Delta x}$$

In a similar manner (11) can be reformulated as follows:

$$u_{j+1}^{n+1} = u_{j+1}^n - \frac{g \Delta t}{2 \Delta x} (h_{j+2}^{n+1} - h_j^{n+1}) \quad (13)$$

Once again multiplying both the numerator and denominator of the second term in the right hand side of (13) by the still water depth  $h_0$  and substituting the square of celerity  $c^2$  for  $gh_0$ , (13) can be re-written in the following form:

$$u_{j+1}^{n+1} = p * h_j^{n+1} + q * u_{j+1}^n + r * h_{j+2}^{n+1} \quad (13a)$$

$$\text{where } p = \frac{Cr}{2} \sqrt{\frac{h_0}{g}}, \quad r = -\frac{Cr}{2} \sqrt{\frac{h_0}{g}}, \quad q = 1 \quad \text{and} \quad Cr = c \frac{\Delta t}{\Delta x}$$

In the difference schemes (12) and (13), the value of the dependent variables at one time is expressed as an explicit function of the value of the dependent variables at the earlier time. Such a scheme is then traditionally called an explicit difference scheme. In a similar manner, (5) and (7), which are the same linearised equations but described in terms of only one variable,  $h$ , can be discretised as a leapfrog scheme and represented explicitly in a finite difference form, respectively as follows:

$$h_j^{n+1} = Cr^2 h_{j+1}^n + 2(1 - Cr^2) h_j^n + Cr^2 h_{j-1}^n - h_j^{n-1} \quad (14)$$

$$h_{j,k}^{n+1} = Cr^2 (h_{j+1,k}^n + h_{j-1,k}^n + h_{j,k+1}^n + h_{j,k-1}^n) + 2(1 - Cr^2) h_{j,k}^n - h_{j,k}^{n-1} \quad (15)$$

These are the forms of the explicit schemes that should be learnt from data in the present study.

### 2.2 The Simple Implicit Formulation used as a test case

The nearly horizontal flow equations (1) and (2) can also be described in algorithmic form as follows:

$$h \frac{\partial u}{\partial t} - u \frac{\partial h}{\partial t} - (u^2 - gh) \frac{\partial h}{\partial x} = 0 \quad (16)$$

$$g \frac{\partial h}{\partial t} - u \frac{\partial h}{\partial t} - (u^2 - gh) \frac{\partial u}{\partial x} = 0 \quad (17)$$

Using centred space and time differences as in Fig. 3 to discretize the terms in these equations and rearranging them leads to the Abbott Ionescu scheme (1967):

$$A1_j h_{j+1}^{n+1} + B1_j u_j^{n+1} + C1_j h_{j-1}^{n+1} = D1_j \quad (18)$$

$$A2_{j+1} u_{j+2}^{n+1} + B2_{j+1} h_{j+1}^{n+1} + C2_{j+1} u_j^{n+1} = D2_{j+1} \quad (19)$$

For the solution of (18) and (19) one can introduce the relation:

$$h_{j+1}^{n+1} = E1_j u_j^{n+1} + F1_j \quad (20)$$

$$u_{j+2}^{n+1} = E2_{j+1} h_{j+1}^{n+1} + F2_{j+1} \quad (21)$$

With the corresponding recurrence relations:

$$E1_{j-1} = \frac{-C2_j}{A2_j E2_j + B2_j} \quad F1_{j-1} = \frac{D2_j - A2_j F2_j}{A2_j E2_j + B2_j} \quad (22)$$

$$E2_{j-1} = \frac{-C1_j}{A1_j E1_j + B1_j} \quad F2_{j-1} = \frac{D1_j - A1_j F1_j}{A1_j E1_j + B1_j} \quad (23)$$

These formulations are discussed in detail in Abbott and Basco (1989) and in Abbott and Minns (1998)

### 3. The formulation of the problem in the new paradigm

As has been mentioned above, one essential component in the development of a new modelling paradigm is that of allowing models to 'construct themselves' more or less automatically by learning from existing numerical-hydraulic models and available field measurements. So far only the possibility of realising the learning process has been investigated, and even then restricted to learning from data streams that have themselves been generated from existing numerical models for specific geographical

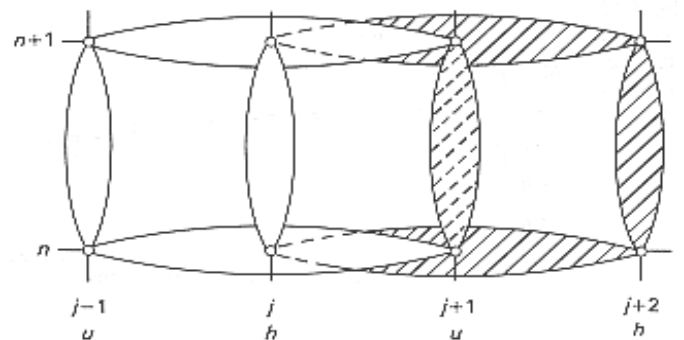


Fig. 3. Schematisation of the Abbott Ionescu staggered scheme [adapted from Abbott & Basco, 1989].

locations while using the simplest tools for automated learning – artificial neural networks – that are currently available (Dibike *et al.*, 1999b; Dibike and Abbott, 1999). A next essential step has been to extend this work to encompass schemes that can be applied over arbitrary bathymetries with distances and time steps that are also arbitrary, consistent with the inherent possibilities of numerical stability of the representation. Since the purpose so far is only to establish the feasibility of such methods, the first step towards such generic systems that forms the subject of this paper is restricted to learning schemes that mimic existing and already well-established numerical schemes. The purpose has been to establish the possibilities and limits of the new paradigm by comparing its productions with the results of the earlier paradigm, as sketched above for the simplest possible cases. The new paradigm is intended in future to supplement, and even in some instances to replace the current one. The present study is also concerned with establishing how far the relatively simple technologies of artificial neural networks can continue to provide acceptable results.

Previous works (see, for example, Minns, 1998) demonstrated that, in the simple case of pure advection with constant velocity, a linear ANN is capable of learning the exact solution by actually restoring the advection equation from the weights of the ANNs. Promising results were also obtained in the case of short-period wave propagation (Dibike *et al.*, 1999a). The present study, however, is directed towards investigating the possibility of using trained ANNs as part of computational schemes, corresponding to numerical operators, which could lead to the development of a generic hydraulic model using artificial neural networks from (in the present case, synthetic) data.

### 3.1 Experimental Studies on the New Paradigm with Explicit formulation

First, a hypothetical horizontal channel of 10,000m long and 10m wide was considered as schematised in Figure 4. The channel was taken as closed at both sides and there was no bed resistance. Flow data was then generated using a conventional numerical model. When the simulation started, the water level was taken to be inclined, with 9m depth at one side and 11m at the other with the intention of establishing a seiche flow condition. Moreover, the initial velocities were taken to be zero at all places along the channel. Different bottom configurations and grid spacing were considered as explained in the following sections:

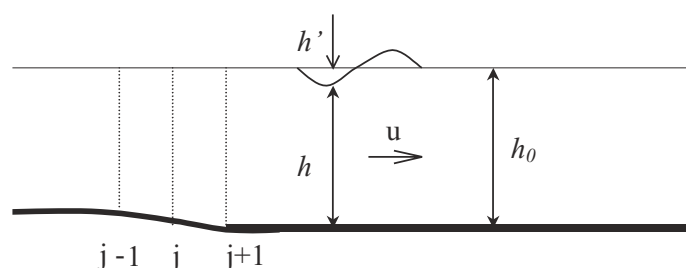


Fig. 4. Schematic view of the one dimensional channel considered.

### Case 1: 1D flow in a channel with horizontal bottom and uniform grid spacing

The first set of training experiments were performed for a uniform grid spacing  $\Delta x$ , but with different time steps  $\Delta t$ . However the Courant number was always kept below one since the input-output mapping is of explicit form. As we are dealing with a linear problem, and also in order to make the analysis easier, two-layered feed-forward networks with linear transfer functions were employed. When both flow variables  $h$  and  $u$  were considered, two networks had to be trained for each simulation since there were two sets of operators in the numerical schemes (10) and (11). For the first network  $u(j-1)$ ,  $h'(j)$  and  $u(j+1)$  at time level  $n$  were taken as input while the  $h'(j)$  at the  $(n+1)$ th time level was taken as an output. For the second network, however,  $u(j+1)$  at time level  $n$  and  $h'(j)$  and  $h'(j+2)$  at time level  $n+1$  were taken as input while the  $u(j+1)$  at the  $(n+1)$ th time level were taken as output. This means that the first network has to solve  $h'$  values at alternate grid points along the channel and then the second network uses these values to solve the  $u$  values at the intermediate grid points along the channel. After training the network with different combination of parameter values, it has finally showed an acceptable performance by reproducing the verification output data with an acceptable accuracy where RMSEs of 0.000023 and 0.000072 were obtained for  $h'$  and  $u$  respectively. The weight of the network's connections obtained through the training corresponds to the coefficients multiplying input variables and, hence directly comparable to the coefficients  $k$ ,  $l$ ,  $m$ ,  $p$ ,  $q$  and  $r$  of the finite difference schemes described in (12a) and (13a). This training exercise was repeated for the different grid spacings (or Cr values) considered. Figure 5 shows a typical performance for different Courant numbers when the numerical coefficients are compared with the coefficients obtained from the weights of ANNs. In a similar experiment a network was trained only with a single flow variable  $h'$  from the same simulation results as those in the previous case. The variable  $h'$  at grid points  $j-1$ ,  $j$ , and  $j+1$  at time level  $n$  and at grid point  $j$  at time level  $n-1$  were considered as input to the network, while the  $h'$  at grid point  $j$  at time level  $n+1$  was considered as output in a way similar to that used in the case of the leap-frog operator. Once again the training experiment was done for different grid spacings, while the Courant number was always kept below one. The weights in the final configuration of the network's connections were compared with the coefficient of the finite difference scheme (14) corresponding to the different grid spacings (or Cr) considered, and typical results are shown in Fig. 6.

In all the above cases, the plots of numerical coefficients obtained from the weights of ANN fit very well with the ones obtained from the finite difference schemes corresponding to the different Courant numbers. Nevertheless, to make the analysis more complete, the staggered grid numerical operators in the computer program were substituted with these trained networks and the simulations were performed with the same initial and boundary condition as had been used in the original model. The water surface and velocity profiles obtained along the channel at a particular time level as well as the time series of water levels at a represen-

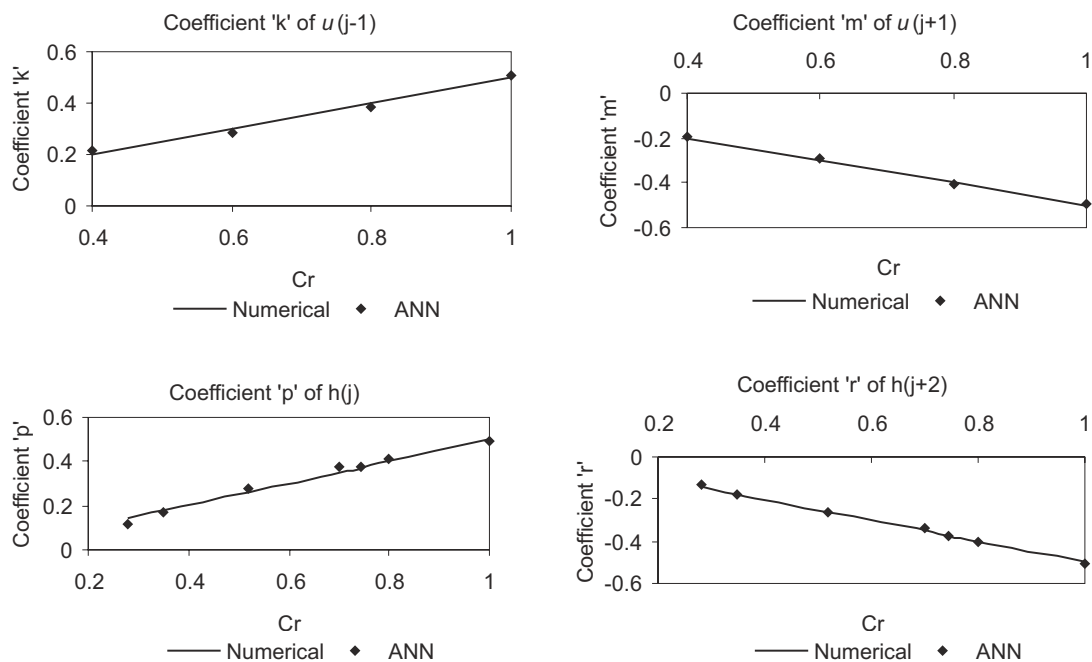


Fig. 5. Comparison of the weights of ANN with the values of numerical coefficient in (12a) and (13a).

tative point are then plotted against the corresponding values obtained with the numerical model, as shown on Figures 7 to 10.

The water surface and velocity plots on Figs. 7 and 8 show good agreement between the numerical simulations and simulations made by ANNs. However, these plots do not by any means fit perfectly. Moreover the differences (or errors) in the time series of water level and velocity on Figs. 9 and 10 keep on increasing with time. This increasing departure from the required values with time seems to be due to accumulation of error, since the output of one network at one time level serves as an input to other networks at the next time level.

However, these results can still be improved by further training the existing type of ANN employed, such as with additional hidden neurons, by using other types, and by using extended training data. The computation with ANNs also took a relatively longer time since the trained network had to run at every grid point. However, this should not be a problem when this approach is ultimately applied on non-structured grids, where the number of points in the flow field which are considered for simulation can be expected to be smaller in number.

#### Case 2: 1D flow in a channel with uneven bottom

The study has been extended to cover flow conditions in channels with non horizontal beds and also to cover flow simulations with variable grid spacings. In the case where the channel bottom was uneven, the term  $h_0$  and hence the wave celerity  $c$  along the channel were no more constant. In order to train a linear network on the simulation result, (12) was rearranged into (24) and the data was prepared to feed the network accordingly.

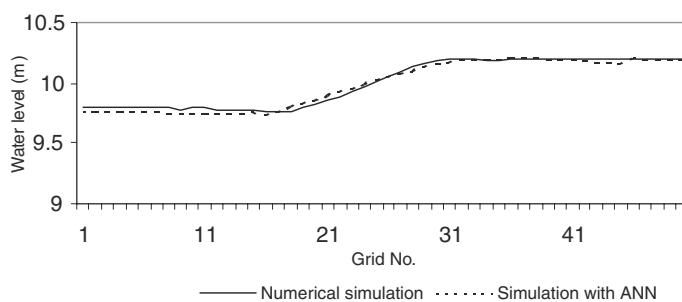


Fig. 7. Comparison of the Water level profiles in the channel after 4000 seconds.

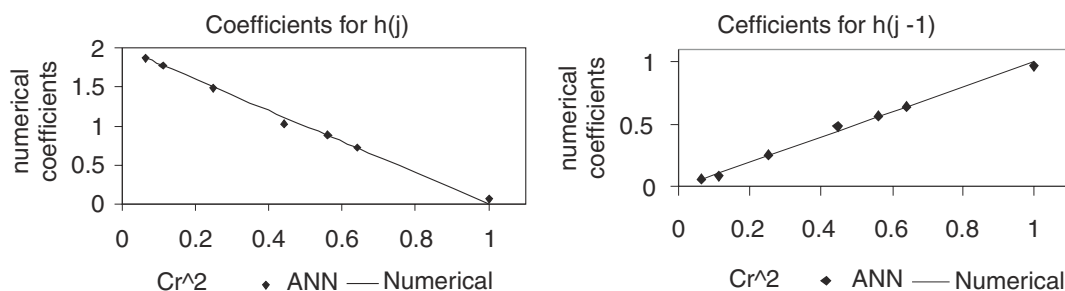


Fig. 6. Comparison of the weights of ANN with the values of numerical coefficient in (14).

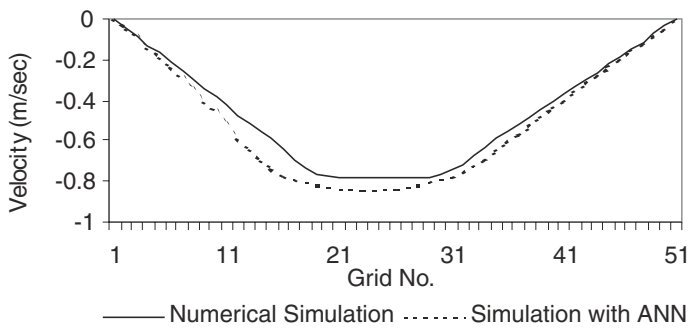


Fig. 8. Comparison of horizontal velocity along the channel after 4000 seconds.

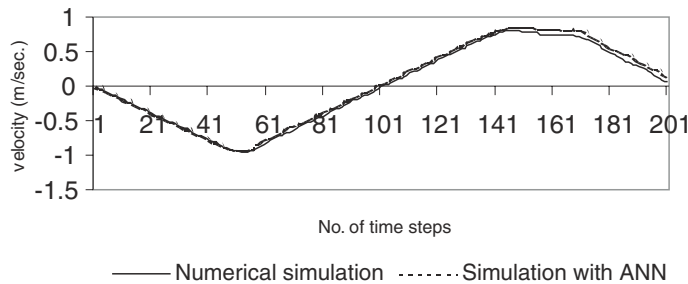


Fig. 10. Comparison of time series of the flow velocity at the middle of the channel.

$$\left(\frac{\Delta t}{\Delta x}\right)^2 h_{j+1}^n - 2\left(\frac{\Delta t}{\Delta x}\right)^2 h_j^n + \left(\frac{\Delta t}{\Delta x}\right)^2 h_{j-1}^n = \frac{1}{gh_{0j}} (h_j^{n+1} - 2h_j^n + h_j^{n-1}) \quad (24)$$

That means the value of three  $h'$  terms at time level  $n$  were provided as input to the network while the result of the expression at the right hand side of (24) were provided as output from the network. After the networks were trained with data obtained from model simulations with different  $\Delta t/\Delta x$  ratios, the weights were compared with the corresponding numerical coefficients in (14). Typical results are shown in Figure 11. Once the values of the expression at the right hand side of (24)

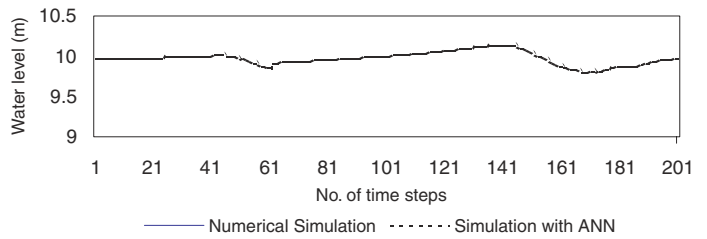
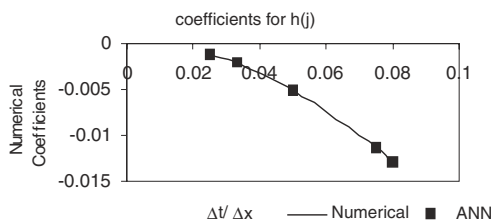


Fig. 9. Comparison of time series of the water level at the middle of the channel.

were calculated by the network from the values of the variable  $h'$  at time  $n$ , then its value at time  $n+1$  could be calculated by rearranging the terms in the right hand side of the same expression (24) at each grid point.

### Case 3: 1D horizontal bottom and variable grid spacing

If there is a change in the grid spacing along the channel, it is not possible to maintain a constant Cr. In order to incorporate this information into the linear network, the coefficients in (12) were rearranged and were rewritten as in (25):

$$\Delta t^2 h_{j+1}^n - 2\Delta t^2 h_j^n + \Delta t^2 h_{j-1}^n = \frac{\Delta x_{j-1} \Delta x_j}{gh_{0j}} (h_j^{n+1} - 2h_j^n + h_j^{n-1}) \quad (25)$$

The input and output data were then prepared accordingly, corresponding to different  $\Delta t$  values and the networks were then trained. Finally, the weights were compared with the corresponding numerical coefficients in (25) and some typical results are shown in Fig. 12. Once again, the value of the expression at the right hand side of (25) was calculated by the network from the values of the variable  $h'$  at time  $n$ , so that its value at time  $n+1$  could then be calculated by rearranging the terms in the same expression at the right hand side of (25) at each grid point.

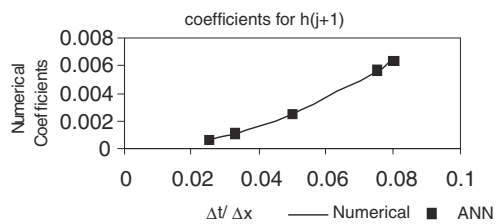


Fig. 11. Plots of numerical coefficients with respect to Cr for channels with uneven bottom.

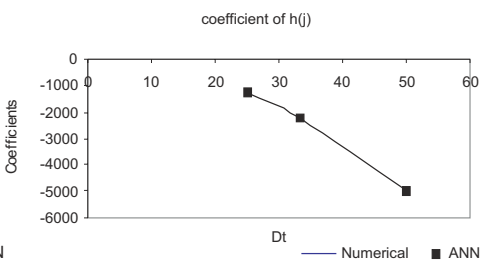
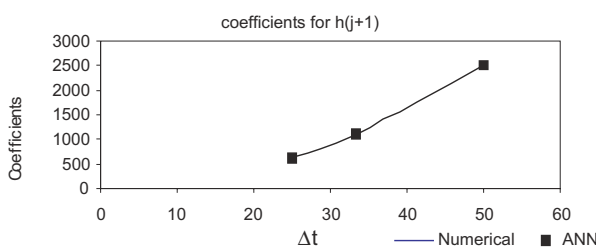


Fig. 12. Plots of numerical coefficients with respect to Cr for channels with variable grid spacing.

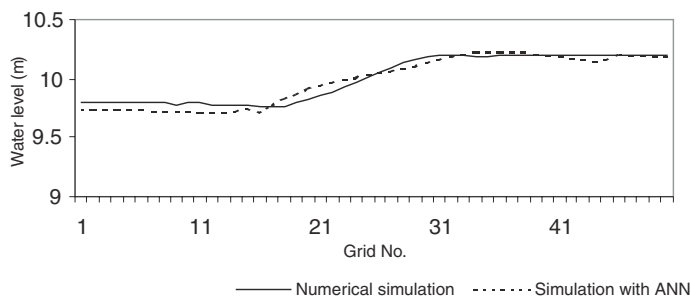


Fig. 13. Comparison of Water level profiles in the channel after 4000 seconds.

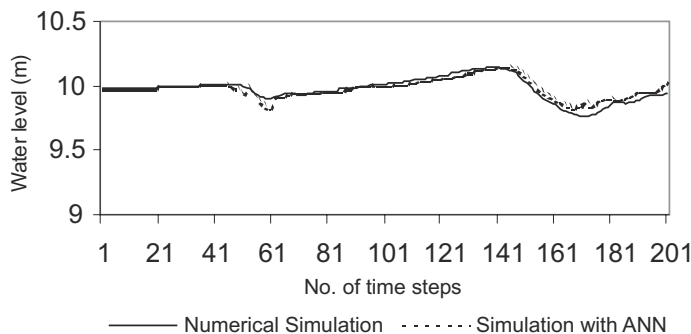


Fig. 14. Comparison of time series of the water level at the middle of the channel.

### 3.2 Experimental Studies on the New Paradigm with an Implicit Formulation

The next case considered was that of the implicit formulation, as introduced in section 2.2. In this particular case the ANN were required to model the recurrent relations in (22) and (23). In order to generate training data, 1D hydrodynamic simulations based on the equations (16) and (17) were performed. However, not only the variables  $u$  and  $h$ , but also the coefficients in the intermediate steps, were written in a text file and prepared to train the ANNs. As the recurrent relation is non linear, three-layer, feed-forward networks with tan-hyperbolic transfer functions in the hidden and output layers were considered (Dibike et al, 1999). Two networks were trained for each simulation. For the first network, all the coefficients from  $A1$  to  $F1$  at grid point  $j$  were taken as input and the corresponding values of  $E2$  and  $F2$  at grid point  $j-1$  were taken as outputs. Similarly, the inputs to the second network were the coefficients from  $A2$  to  $F2$  at grid point  $j$  while  $E1$  and  $F1$  at grid point  $j-1$  were the outputs. The networks were able to approximate the recurrence relations quite well, with RMSEs of 0.0007 and 0.0012 corresponding to (22) and (23) respectively. In order to test the performances of these trained networks, they were once again imbedded in the program in such a way that once the coefficients  $A1$  to  $D1$  and  $A2$  to  $D2$  had been calculated at alternating grid points, these two trained networks could then be run alternatively from one to the other boundary, thus functioning as numerical operators in place of (22) and (23). The simulation result obtained using these operators was compared with the one using a conventional numerical operator as shown in Figure 13. Although Figures (13) and (14) show a reasonable agreement between the simulation results obtained by the conventional method

which uses the recursive relations in equations (22) and (23), and the one which uses ANNs instead to calculate the intermediate coefficients, the difference between the two is more than the one obtained with the explicit formulation, as shown above in Figures 7 and 9. This could be attributed to an accumulation of error through the recursive process since the output from one network at a given grid point is taken to be an input to the next network at the next grid point. The computation with ANNs also took a relatively longer time since the trained network has to run at every grid point. Nonetheless, this approach promises the additional advantage that it could be implemented with greater time steps ( $Cr > 1$ ) since it is formulated in an implicit manner.

### 3.3 2D flow in a rectangular basin with horizontal bottom and uniform grid spacing

To extend this preliminary study to a two-dimensional flow case, a rectangular basin of 1500m by 2000m was considered. The basin was closed at the two longest sides while it has partially open boundaries at opposite corners of the other two sides. as shown in Fig. 15 in such a way that 2 dimensional flow could be obtained. The basin had an initial depth of 10m and a wave of constant form with 0.5m amplitude and a wave period of 5 seconds was introduced at the upstream open boundary while the downstream boundary was kept at a constant depth of 10m. A grid spacing of 50 meters in both the  $x$  and  $y$  direction and a time step of 5 seconds was chosen to make  $Cr$  near to one. The flow in the basin was then simulated using an alternating direction implicit (ADI) algorithm and flow data was generated. This flow data was used to train a linear ANN with  $h'(j,k)$ ,  $h'(j-1,k)$ ,  $h'(j+1,k)$ ,  $h'(j, k-1)$ ,  $h'(j,k+1)$  all at time level  $n$  and  $h'(j, k)$  at time level  $n-1$  as input to the network and  $h(j, k)$  at time level  $n+1$  as an out put in a way similar to the 2 dimensional leap-frog scheme employed to derive equation (15). The flow simulation and the corresponding training of ANN with the resulting flow data was repeated with successively decreasing time stapes to keep  $Cr$  less than one. The values of the weights of the ANNs and the coefficients of the corresponding finite difference scheme represented in equation (15) were then plotted with respect to  $Cr$  for each input variable as shown in Figs.16. Once again the plots of numerical coefficients obtained from the weights of ANN fits very well with the ones obtained from the finite difference schemes corresponding to the different Courant numbers.

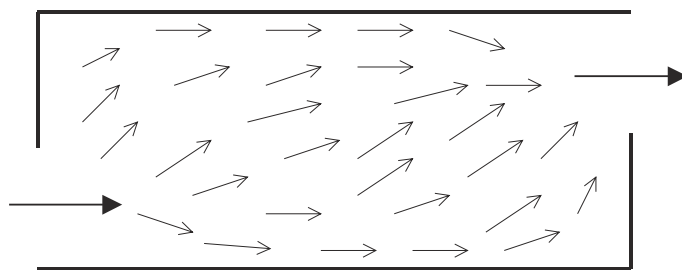


Fig. 15. Plan of the rectangular basin for 2D flow simulation.

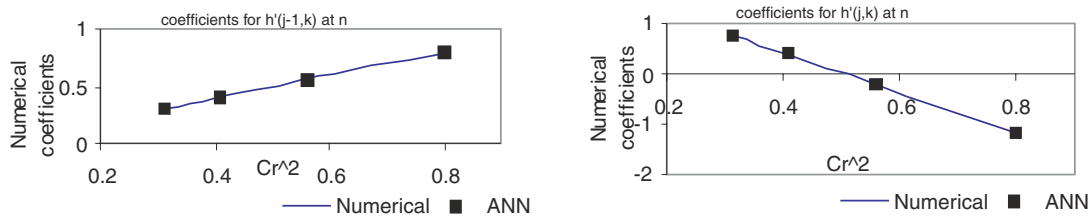


Fig. 16. Plots of numerical coefficient with respect to  $Cr$  for simulations with variable grid spacing.

#### 4. Conclusion

Possibilities of the new paradigm, namely allowing models to ‘construct themselves’ by learning from existing numerical-hydraulic models, was investigated by extending previous works to encompass schemes that can be applied over arbitrary bathymetries with distances and time steps that are also arbitrary. For the simplest possible cases considered in this study, the relatively elementary technology of artificial neural network was found to provide acceptable results. This has been demonstrated for the different cases of 1 and 2 dimensional flow problems by comparing the weights of the network’s connections with the coefficients of the corresponding finite difference schemes with different time steps (hence different Courant numbers). The study also included cases with variable bed levels and schemes with variable grid spacing and promising results were obtained.

Moreover, It was demonstrated that the well-trained networks could be substituted in place of the finite difference schemes in the hydrodynamic model formulation and could perform like numerical operators. In this case, models with trained ANNs as explicit numerical operator perform better than the once with ANNs as implicit operators. However, the implicit operators have the extra advantage that they can be implemented with relatively bigger time steps (hence  $Cr > 1$ ). In general, flow simulations with trained ANNs as numerical operators require relatively longer running times. However, This approach could be very useful if implemented for model simulation in a non-structured grid, where the number of points of interest, and hence points to be considered for the simulation, is relatively small and a reasonable amount of historical (or synthetic) data are available at those locations. In such cases, ANNs are trained based on the data at selected points in the model area and the corresponding boundary conditions. Then the trained networks could be used as numerical operators for subsequent model simulations in the same area.

#### Acknowledgement

This work was carried out as part of an IHE doctoral study supervised by Prof. M. B. Abbott

#### Notations

ANN	artificial neural network
$c$	wave celerity (m/sec.)
$Cr$	Courant number
$g$	gravitational acceleration (m/sec. <sup>2</sup> )
$h$	instantaneous depth of flow (m)
$h_0$	still water depth (m)
$h'$	elevation of free water surface with respect to the still water level (m)
$j$	number of distance grid intervals
$n$	number of time grid intervals
RMS	root mean square error
$t$	time (sec.)
$u$	depth averaged horizontal velocity
$x$	longitudinal distance (m)

#### References

- ABBOTT, M.B., 1997, Engine 2000: Research in to the next generation of computational hydraulic modelling, *Proceedings of the 27th Congress of the International Association for Hydraulic Research*, Vol. 2, pp. 859-864.
- ABBOTT, M.B. and BASCO, D.R., 1989, *Computational Fluid Dynamics*, Longman, London.
- ABBOTT, M.B., and IONESCU F., 1967, On the numerical computation of nearly horizontal flows, *Journal of Hydraulic research*, Vol. 5, pp. 97-117.
- ABBOTT, M.B., and MINNS, A., 1979/1985/1998, *Computational hydraulics* 1st ed., Pitman, London/Longman, London, and Wiley, New York/2nd edition, Ashgate, Aldershot, UK, and Brookfield, USA.
- DIBIKE, Y.B., MINNS, A.W. and ABBOTT, M.B., 1999a, Application of Artificial Neural Networks to the Generation of wave Equations from Hydraulic Data., *Journal of Hydraulic research*, Vol. 37, No. 1, pp. 81-97.
- DIBIKE, Y.B., SOLOMATINE, D., and ABBOTT, M.B., 1999b. On the encapsulation of numerical-hydraulic models in artificial neural network, *Journal of Hydraulic research*, Vol. 37, No. 2, pp. 147-161.
- DIBIKE, Y.B. and ABBOTT, M.B., 1999. Application of artificial neural networks to the simulation of two dimensional flow, *Journal of Hydraulic research*, Vol. 37, No. 4, pp. 435-446.
- MINNS, A.W., 1998, Artificial Neural Networks as Subsymbolic Process Descriptors, PhD thesis, Balkema, Rotterdam.