

DIFFERENTIAL QUADRATURE TECHNIQUES FOR SOLVING SHALLOW WATER FLOW USING SPREADSHEETS

M. EZZELDIN¹, A. S. AL-GHAMDI² & ABDULRAZAK H. AL-MALIKI³

^{1,2}Professor, Department of Civil Engineering, College of Engineering, King Abdulaziz University (KAU),
Jeddah Saudi Arabia

³Research Scholar, Department of Civil Engineering, College of Engineering, King Abdulaziz University (KAU),
Jeddah, Saudi Arabia

ABSTRACT

There has been a growing interest in using the differential quadrature method over the last 15 years and spreadsheet simulation modeling field of water resource modeling. This Current work provides an unsteady surface water modeling using spreadsheet simulation (USMS) model for (solving surface water problems where the user can edit the upstream discharges of the river to get the discharge at any point on the channel automatically. The present project is proposing applying the differential quadrature method on the simulating flow in one dimension prismatic open channel through solving the diffusion wave models. Their high specific accuracy and stability for solving PDE is one of the main reasons for choosing them. USMS is a practical method and introduction to unsteady surface water modeling that uses spreadsheets instead of conventional surface water model codes. Unsteady flow, variable boundary and initial conditions, one dimension prismatic open channel, are evaluated in the USMS. Various flow management scenarios could be studied by obtaining the surface water level of the different times. In order to verify the DQM solution and the USMS, flood data found in published paper Bajracharya et al., 1996) is used as a benchmark case and the results of DQM solution is compared with Crank Nickerson method and McCormick schemes which are the most two accurate implicit and explicit Finite Difference schemes. DQM solution shows good agreement with benchmark case.

KEYWORDS: Differential Quadrature, Saint Venant equation, Spreadsheets, Preissmann and McCormick Scheme

INTRODUCTION

Shallow water modeling is one of the most important topics in Engineering. Generally, the main issue in hydrology and hydraulics is to understand the flood wave theory and solve the governing equations accurately for a given set of parameters, boundary conditions and scenarios. So finding accurate solutions for the governing equations of flood wave propagation plays an important role in the hydrology and hydraulics area. Therefore this is an important subject because of its vast applications in different hydraulic problems like flood forecasting, design of spillways, waterways and dams. Thus, there is wide variety of academics and practitioners have expressed about interest in solving them by the numerical models.

The mathematical representation of the unsteady flow is governed by the hyperbolic fully non-linear Saint-Venant equations. As they are highly non-linear and there is no analytical solution in the literature for these equations. So the solution of these equations can be obtained only numerically [2–4]. Due to the nonlinearity of the governing equations and the numerical techniques are more commonly used as compared to analytical solutions.[5]

Lal, A.M, (2012) studied the limitations of DW based models when simulating kinematic flow conditions. [6]Tommaso et al., (2012) studied the applicability of the KW and (DW) approximations for the two types of downstream boundary conditions, critical-flow depth and zero-flow depth gradient for steady flow in prismatic channels, by using a second-order two-step Lax-Wendroff numerical scheme coupled with the characteristic method at the boundaries.[7] (2010) (Akbari G. and Firoozi B.) investigated two different numerical methods, namely; Preissmann and Lax diffusive schemes for numerical solution of Saint- Venant equations that govern the propagation of flood wave, in natural rivers, with the objective of the better understanding of this propagation process.

The finite difference techniques are commonly used to solve the partial differential equations through numerical methods and for solving Saint-Venant Equations. Flood wave prediction is a very important aspect in hydraulic concerns, thus there is a need to find the numerical schemes that have higher stability, accuracy and convergence to solve those equations. The differential quadrature method (DQM) which is applied in this research is stable and has high level of accuracy [1-12].

The differential quadrature method (DQM) was developed by Richard Bellman (Bellman et al., 1971) [8]. This method can solve equations in a differential form of any system, including boundary/initial conditions in the equations. Richards (1992) uses DQM in some applications of fluid mechanics and the bending and twisting of beams [9]. In the recent studies, Fung (2001) has concentrated on the application of DQM in problems which may be faced in fluid mechanics and heat transfer [10]. Hashemi, Abedini and Malekzadeh (2006) used the differential quadrature method to solve full dynamic equations which is applied to simulating flow in a prismatic rectangular open channel and regular wetted bed [14]. Kaya (2010) has discovered the use of DQM on the solution of the Advection Diffusion Equation [12,]. Kaya and Arisoy (2010) have applied that method in solving full dynamic equations in order to apply it to simulating flow in a prismatic trapezoidal open channel and regular wetted bed [13]. Meral[15] (2013) computed weighting coefficients by using the Lagrange polynomials by which the solution of the differential quadrature of heat- and mass-transfer equations has been found. The solution showed the applicability of DQM space- RungeKutta method time procedure for the one- and two-dimensional domain. Jiwari[16] (2013) proposed a solution of Burgers' equation based on the weighted average differential quadrature method and the Lagrange polynomials which are used as the test functions to compute weighting coefficients.

The computer technology development ease solving the shallow water flow equations. Spreadsheet is software and it used for solving the PDEs which is one of the best and available tools. The Spreadsheet has some important Capabilities for instance, fast calculating, numerical and visual feedback. In addition to input and output through the Spreadsheet that can be plotted at the same worksheet. And its graphical interface is easing the understanding of data and results. One of the important advantage of spreadsheet is that Spreadsheets are easy to program. Any changes in the solution domain will be appeared in the solution workspace due to Spreadsheets graphical representation capability.

During the last 15 years, spreadsheets are used as solver in the solution of many different fields of engineering problems such as in solution of PDEs [20]. One-dimensional transient heat-conduction problems [24], and free-surface seepage problems [19].Using of spreadsheet in the engineering problems that have been growth recently. There are many published papers in the literature [17–24] for engineering applications, and some applications of spreadsheet in water resource engineering fields. HalilKarahan [19] has concentrated on the application of spreadsheet in groundwater problems which may be faced and showed that spreadsheet is useful tool in this field. HalilKarahan [19] proved that excel

spreadsheets introduce an easy approach for modeling and understanding groundwater problems instead of using conventional groundwater model codes. He solved two-dimensional transient unsteady-state groundwater problems for homogeneous aquifer parameters with constant sinks and source terms.

Solving the saint vent equation which includes time dimension with higher grid points is difficult. Therefore, there is a need for using more efficient and simpler computational tools for solving the surface modeling system in the unsteady-state case. In surface water modeling on spreadsheet, there is no need to write an equation code in each cell to solve equations. Spreadsheet has features of copy and paste, DQM equation can be copied to the next time step cells without writing the equations to each cells. Calculations are started iteratively and simntnously once the equation pasted to the next time step cells of the solution domain which may cause decreasing the computational time.

The main goal of this study is to deliver flexible shallow water modeling simulation algorithm using the DQM. Therefore, an unsteady surface water modeling using spreadsheet simulation (USMS) model has been presented.

DESCRIPTION OF THE DIFFERENTIAL QUADRATURE METHOD

The differential quadrature method (DQM) is a numerical scheme for solving the partial differential equations. The Differential Quadrature Method (DQM) was developed by Richard Bellman (Bellman et al., 1971) [2, 3] in the 1971 and, since then, the technique has been used in finding the solutions of many problems in applied and sciences [1-4]. This method can solve the equations in a differential form of any system, including boundary/initial conditions in the equations.

The idea of a differential quadrature method is based on the derivation of a function with respect to a space or time variable at any point along a mesh line which is approximated as a weighted linear summation of the functional values at all others sampling points in the mesh of that variable.

For showing the mathematical representation of the DQM, we consider one dimensional domain variable $f(x)$ in a domain specified by $a = x_1 \leq x \leq x_N = b$ see figure 1.

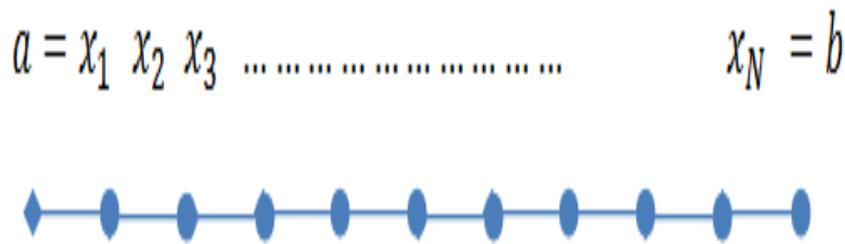


Figure 1: One Dimensional Field Variable $f(x)$

Assume $f_i = f(x_i)$ are the function values specified in a distributed set of N sampling points x_i $i = 1, 2, \dots, N$ of the domain in which the end points x_1 and x_N are in

$$f^{(r)}(x_i) = \frac{d^r f(x_i)}{dx^r} = \sum_{j=1}^N A_{ij}^{(r)} f_j \quad (i = 1, 2, \dots, N) \tag{1}$$

where $A_{ij}^{(r)}$ are the weighting coefficients of the $r^{(th)}$ -order derivative of the function with respect to points x_i .

The equation above is the Differential Quadrature Method rule for solving any derivative.

The advantage of this method depending on the literature review which are:

- It is rapid, convergent and accurate method.
- Solution results can be gotten by using less grid points in space and time domain.
- Its stability is not sensitive to time step like Courant number(Cr)
- Solution results can be obtained smoothly and accurately.

The only disadvantage is that the results are sensitive to grid distribution in time and space domain.

Mathematical and Application of Differential Model

The governing differential equation for this research can be expressed as following:

$$\frac{\partial Q'}{\partial t} + C \frac{\partial Q'}{\partial x} = D_h \frac{\partial^2 Q'}{\partial x^2} \quad 0 < x < L, 0 < t < T \quad (2)$$

equation is gained (Yen & Tsai,2001).

By applying the rule of Quadrature Method in Eq 1, the linearized St. Venant equations can be discretized in terms of differential quadrature as:

$$\sum_{r=1}^R A_{r,s} Q_{i,r} + [C \sum_{j=1}^N B_{j,i} - D_h \sum_{j=1}^N B_{j,i}^{(2)}] Q_{j,s} = 0 \quad i = 1, 2 \dots N ; s = 1, 2, \dots, R \quad (3)$$

Eq. 3 is called the differential quadrature (DQ). Where $A_{r,s}$ and $B_{i,j}$ are matrix represent the weighting coefficients of each points on the time and space domain respectively, and N is the number of grid points in the whole time domain, R is the number of grid points in the whole space domain in(x direction). It should be noted that the weighting coefficients $A_{r,s}$ and $B_{i,j}$ is the 1st order derivative with respect to x at each point on the grid and are different at different locations of xi and ts and their calculation depends on sample spacing between the grids (the location).The distribution is used usually for determine the location of each point in the domain. The key procedure in DQ approximation is to use the proper grid points distribution and then determine the weighting coefficients $A_{r,s}$ so, this research shows the determination of the weighting coefficients in detail and (USMS) is used for programming the problems and they will be used for achieving the stable solution. The boundary conditions at the initial and boundary conditions are given by:

$$Q_{(0,x)} = F1_{(t)}, 0 < x < L \text{ and } Q_{(0,t)} = F2_{(x)}, 0 < t < T$$

$F1_{(t)}$ and $F2_{(x)}$ are function of time and space respectively.

After involving the initial and boundary condition in Eq 3 then will yield:

$$\sum_{r=2}^R A_{r,s} Q_{i,r} + [C \sum_{j=2}^N B_{j,i} - D_h \sum_{j=2}^N B_{j,i}^{(2)}] Q_{j,s} = A_{1,s} Q_{i,1} + [CB_{1,i} - D_h B_{1,i}^{(2)}] Q_{1,s} \quad i = 2, N ; s = 2, 3, \dots, R \quad (4)$$

The procedure of Applying DQM as following

- Form the network domain by Discretize the space domain in x axis and time in y axis and using Gauss-Lobatto-Chebyshev sampling points distribution which is recommended for numerical stability.
- Approximate the derivatives in a differential equation (saint venat equation Eq.2) by using the equation (1). By substituting the equation (1) into the governing equations.

- Apply DQM on each unknown point in the domain, so one has one equation for each point, for each unknown. The derivative of any function with respect to a space or time variable at any point on the domain is determined as a weighted linear sum of the functional values at all others discrete points in the domain of that variable.
- Form a system of linear equations.
- Matrices must be formed from a system of linear equations which can be solved by using of Gauss elimination or other methods.
- Solving the system of linear equation yields the desired unknowns.

Weighting Coefficients

Weighting coefficients is the key factor of solving PDE by using the differential quadrature method. Every grid points in the domain effect each other by weight which can be calculated by using the different types of polynomial interpolation. The accuracy of the differential quadrature method that depends on the accuracy of the weighting coefficient which is associated with the order of the derivative of the PDE terms and grid points distribution.

Quan and Chang [25] have developed an explicit formulae of the Lagrangian- interpolation-function-based on weighting coefficients for the higher derivatives which is the reason of not using them in this research. Belmans approach presents approximation for first order derivative and as the PDE under study includes second order derivative term, this approach is not good chosen.

For first or higher order derivatives, Shu [28] proposed a general relationship by considering the test function in the Lagrangian interpolation process. This explicit formulae is that highly accurate weighting coefficients could be obtained for any number of the distributed grid sampling points. Villadsen, Michelsen[29],Quan and Chang [15] presented the weighting coefficients of r (th) -order derivatives of the Lagrangian interpolation test functions. According to the literature {[25], [26], [27], [28] [29], } there is no effect of using different types of weighting coefficients on the accuracy of the solutions, but the accuracy of weighting coefficients depends on grid points distributions and order of the derivatives.

The differential quadrature weighted coefficients can be solved by using several techniques like Bellman's first approach, Shu's general and Quan Chang's approach in the present study, Shu's general approach which is a numerical discretization technique that delivers logical and accurate numerical solutions. Shu's general approach is based on (Legendre polynomials) then the weight coefficient matrix can be written in general as (Shu et al. 2004):

$$A_{i,j}^{(r)} = \frac{d^r}{dx^r} l_j(x_i) (i, j = 1, 2, \dots N) \tag{5}$$

Where,

$$l_j(x) = \frac{\phi(x)}{(x-x_j) \phi^{(1)}(x_j)}; \phi(x) = \prod_{m=1}^N (x - x_m) (i, j = 1, 2, \dots N) \tag{6}$$

By substituting 6 in 5 yields;

$$A_{i,j}^{(1)} = \frac{d^1 l_j(x_i)}{dx^1} = \frac{\phi^{(1)}(x_i)}{(x_i-x_j) \phi^{(1)}(x_j)}; \phi(x) = \prod_{m=1}^N (x - x_m) (i, j = 1, 2, \dots N \ i \neq j) \tag{7}$$

In this work, the weight coefficient matrix can be written as (Shu et al. 2004):

$$B_{i,k} = \frac{L_i}{L_k(x_i - x_k)}, Li = \prod_{j=1}^N(x_j - x_i) \tag{8a}$$

$$Lk = \prod_{j=1}^N(x_j - x_k) \quad i \neq k \tag{8aa}$$

$$B_{i,i} = -\sum_{k=1}^N B_{i,k} \quad i \neq k \tag{8b}$$

$$B_{i,k}^{(2)} = 2 \left[B_{i,k} B_{i,k}^{(1)} - \frac{B_{i,k}}{x_i - x_k} \right] \tag{9a}$$

$$B_{i,i}^{(2)} = -\sum_{k=1}^N B_{i,k}^{(2)} \quad i \neq k \tag{9b}$$

A matrix of time domain can be written like B matrix in space domain as

$$Li = \prod_{j=1}^R(t_j - t_i) \tag{10}$$

$$A_{i,k} = \frac{L_i}{L_k(t_i - t_k)} \quad i \neq k \tag{11}$$

$$A_{i,i} = -\sum_{k=1}^R A_{i,k} \quad i \neq k \tag{12}$$

Note that the Legendre interpolation shape functions $l_j(x)$ have the following properties.

$$l_j(x) = 1 \text{ if } i = j \tag{13}$$

And then Eq. 4 has been applied in the formulation of the matrix, as shown below.

$$\begin{aligned} & \begin{bmatrix} A_{2,3} & A_{2,4} & \dots & A_{2,s=R} \\ A_{3,3} & A_{3,4} & \dots & A_{3,s=R} \\ \vdots & \vdots & \dots & \vdots \\ A_{R,3} & A_{R,4} & \dots & A_{r=R,s=R} \end{bmatrix} * \begin{bmatrix} Q_{1=2,r=2} & Q_{2,3} & Q_{2,4} & \dots & Q_{2,r=R} \\ Q_{3,2} & Q_{3,3} & Q_{3,4} & \dots & Q_{3,r=R} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ Q_{1=N,2} & Q_{1=N,3} & Q_{1=N,4} & \dots & Q_{1=N,r=R} \end{bmatrix} \\ & + \left[c * \begin{bmatrix} B_{j=2,i=2} & B_{2,3} & B_{2,4} & \dots & B_{2,i=N} \\ B_{3,2} & B_{3,3} & B_{3,4} & \dots & B_{3,i=N} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ B_{j=N,2} & B_{N,3} & B_{N,4} & \dots & B_{j=N,i=N} \end{bmatrix} - D_h * \begin{bmatrix} B_{j=2,i=2}^{(2)} & B_{2,3}^{(2)} & B_{2,4}^{(2)} & \dots & B_{2,i=N}^{(2)} \\ B_{3,2}^{(2)} & B_{3,3}^{(2)} & B_{3,4}^{(2)} & \dots & B_{3,i=N}^{(2)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ B_{j=N,2}^{(2)} & B_{R,3}^{(2)} & B_{R,4}^{(2)} & \dots & B_{j=N,i=N}^{(2)} \end{bmatrix} \right] * \begin{bmatrix} Q_{j=2,s=2} & Q_{2,3} & Q_{2,4} & \dots & Q_{2,s=R} \\ Q_{3,2} & Q_{3,3} & Q_{3,4} & \dots & Q_{3,s=R} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ Q_{j=N,2} & Q_{N,3} & Q_{N,4} & \dots & Q_{j=N,s=R} \end{bmatrix} \\ & = -[A_{r=1,s=2} \quad A_{1,3} \quad A_{1,4} \quad \dots \quad A_{1,R}] * \begin{bmatrix} Q_{i=2,1} \\ Q_{3,1} \\ Q_{4,1} \\ \vdots \\ Q_{j=N,1} \end{bmatrix} - \{ [c * [B_{j=1,i=2}] - D_h * [B_{j=1,i=2}^{(2)}]] * [Q_{i=1,s=2} \quad Q_{1,3} \quad Q_{1,4} \quad \dots \quad Q_{1,s=R}] \} \tag{14} \end{aligned}$$

Where $Q_{i,r}$ and $Q_{j,s}$ are N and R unknown function values. The previous matrices is solved for the $Q_{i,r}$ and $Q_{j,s}$ which are unknown values. Approximate derivatives in the differential equation by these N unknown function values. Note that the boundary and initial conditions have been constructed in the form of the Weighting coefficient matrix as shown in the right hand side above. After taking the matrices of unknown values $Q_{i,r}$ and $Q_{j,s}$ as a common factor for the equation above yields:

$$\begin{aligned} & \begin{bmatrix} Q_{i,j=2,r=s=2} & Q_{2,3} & Q_{2,4} & \dots & Q_{2,rand=s=R} \\ Q_{3,2} & Q_{3,3} & Q_{3,4} & \dots & Q_{3,rand=s=R} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ Q_{i,j=N,2} & Q_{i,j=N,3} & Q_{i,j=N,4} & \dots & Q_{i,j=N,r,s=R} \end{bmatrix} * \begin{bmatrix} A_{r=2,s=2} & A_{2,3} & A_{2,4} & \dots & A_{2,s=R} \\ A_{3,2} & A_{3,3} & A_{3,4} & \dots & A_{3,s=R} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ A_{r=R,2} & A_{R,3} & A_{R,4} & \dots & A_{r=R,s=R} \end{bmatrix} + \\ & \left[c * \begin{bmatrix} B_{j=2,i=2} & B_{2,3} & B_{2,4} & \dots & B_{2,i=N} \\ B_{3,2} & B_{3,3} & B_{3,4} & \dots & B_{3,i=N} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ B_{j=N,2} & B_{N,3} & B_{N,4} & \dots & B_{j=N,i=N} \end{bmatrix} - D_h * \begin{bmatrix} B_{j=2,i=2}^{(2)} & B_{2,3}^{(2)} & B_{2,4}^{(2)} & \dots & B_{2,i=N}^{(2)} \\ B_{3,2}^{(2)} & B_{3,3}^{(2)} & B_{3,4}^{(2)} & \dots & B_{3,i=N}^{(2)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ B_{j=N,2}^{(2)} & B_{R,3}^{(2)} & B_{R,4}^{(2)} & \dots & B_{j=N,i=N}^{(2)} \end{bmatrix} \right] = -[A_{r=1,s=2} \quad A_{1,3} \quad A_{1,4} \quad \dots \quad A_{1,R}] * \begin{bmatrix} Q_{i=2,1} \\ Q_{3,1} \\ Q_{4,1} \\ \vdots \\ Q_{j=N,1} \end{bmatrix} - \{ [c * \\ & [B_{j=1,i=2}] - D_h * [B_{j=1,i=2}^{(2)}]] * [Q_{i=1,s=2} \quad Q_{1,3} \quad Q_{1,4} \quad \dots \quad Q_{1,s=R}] \} \tag{15} \end{aligned}$$

The boundary condition in Eq. (4) can be written in the matrix form as:

$$-[A_{r=1,s=2} \quad A_{1,3} \quad A_{1,4} \quad \dots \quad A_{1,R}] * \begin{bmatrix} Q_{i=2,1} \\ Q_{3,1} \\ Q_{4,1} \\ \vdots \\ Q_{j=N,1} \end{bmatrix} \tag{16}$$

The initial condition in Eq. (4) can be written in the matrix form as:

$$-\{[c * [B_{j=1,i=2}]] - D_h * [B_{j=1,i=2}^{(2)}]\} * [Q_{i=1,s=2} \quad Q_{1,3} \quad Q_{1,4} \quad \dots \quad Q_{1,s=R}] \tag{17}$$

Note that the initial and boundary conditions have been applied in the formulation of the matrixes.

SAMPLING POINT'S SELECTION

In case of using the Differential Quadrature, non-equal intervals sampling points gives more accuracy for the solutions there for, Shu is recommend unequally sampling points like Chebyshev-Gauss-Lobatto grid points (Shu, 2000). For a domain specified by $r_1 \leq x \leq r_N$ and with using Gauss-Lobatto-Chebyshev sampling points (non-uniform grid), then the coordinate of any point (i) can be determined by:

$$r_i = 0.5(1 - \cos \frac{i-1}{N-1} \pi) \tag{18}$$

The location of each grid point on the space domain can be determined as following:

$$x_i = \frac{r_i - r_1}{r_N - r_1} \tag{19}$$

In time domain as following:

$$t_i = \frac{r_i - r_1}{r_R - r_1} \tag{20}$$

MODEL DEVELOPMENT

The flowchart of the DQM solution for the (USMS), which uses DQM, can be seen in figure 2. The structure of the program consists of two main separate excel spreadsheet files, one for calculating the weighting coefficient matrices in space domain and the other for calculating the weighting coefficient matrices in time domain. The two files are easily connected by using the excel capabilities and both of them have the same general structure, see figure 2.

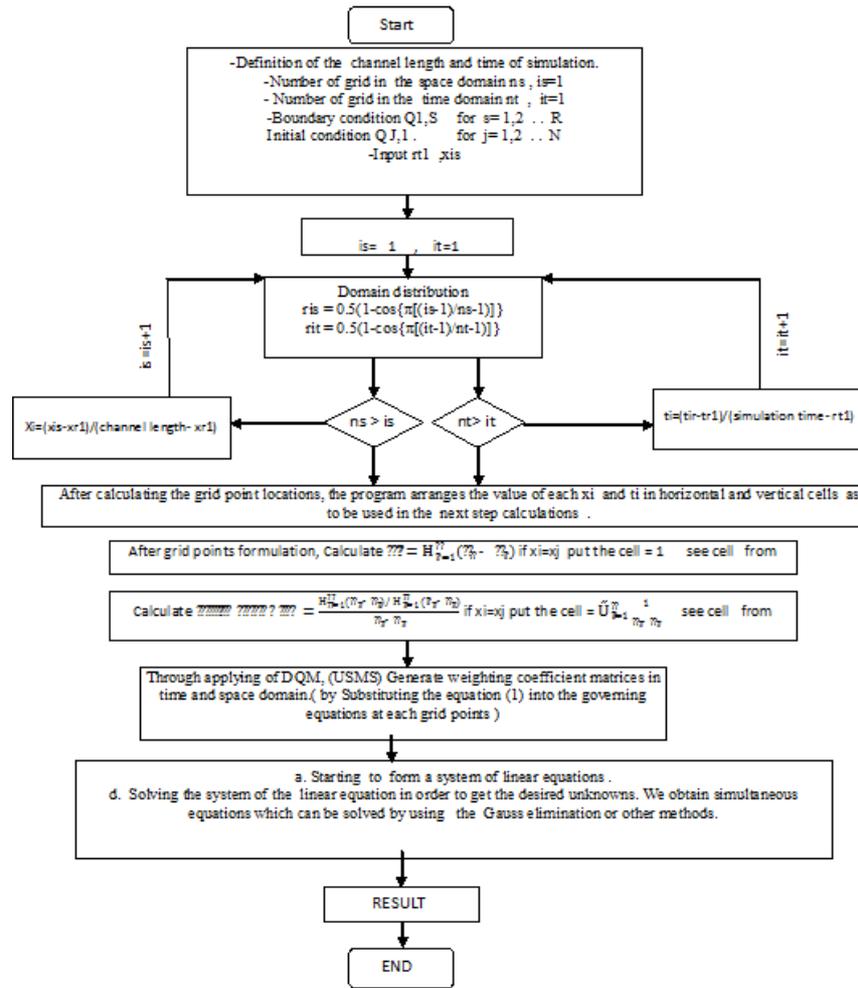


Figure 2: The (USMS) Flowchart

The structure is consisting of three matrices, each of them consists of the same number of the columns and the rows which are equals the number of grid points in the desired domain wither space or time domain. The first row from up and the first column from left are representing the space or time domain and the space domain can be generated by using equation (18) and its code statement is:

$$AZ64 = \$AA\$6 + (1 - \text{COS} ((AZ63 - 1) / (\$Z\$6 - 1) * 3.14)) / 2 * \$AB\$6$$

The function of the first matrix is to calculate Equation 8aa in space domain and the equation 10 in the time domain and the program statement is:

$$=IF (BB12=BB13; AZ12=1; \$BBS13-BB12)$$

The statement of the last row is:

$$=PRODUCT (AY12:AY61)$$

The initial condition was assumed as: $Q_{(x,0)} = 0$. As well as the values of the celerity, c , the diffusion coefficient and D were taken as 1 m/s and 100 m²/s, respectively. So the length of channel under study is 5000 m and the simulation time is 10000 second.

THE RESULTS AND DISCUSSIONS

Stability Analysis of DQM Solution

The numerical stability is a result of the accuracy of the numerical algorithm. The numerical error (erroneous initial conditions, local truncation or round-off errors) and its propagation affect the numerical stability. The numerical solution is said to be unstable if errors were clearly observed at some parts in the calculations. So any small change in the input data or the method parameters may produce correspondingly big changes in the final results and at which the solution considered unstable. Therefore the method is considered stable if small change in input data causes small or the same change in magnitude in output results. So the numerical stability, means that the numerical solution has to be chosen to the benchmark case solution through all stages of calculations. In the present project, the different changes in input items of the method have been done to discover which parameter has big effect on the accuracy of DQM solution and then the numerical stability.

The accuracy and stability of the results are affected by two factors, one of them is grid point distribution types and the other is test functions types (Bert and Malik, 1996) and the affection is presented in this research. Basically in Figure 9 shows the effect of grid point distribution types on the accuracy of the solution which the DQM solution shows good agreement with benchmark case with using Gauss-Lobatto-Chebyshev sampling point's distribution (line) and poor agreement with using others. Figure 9 also shows the effect of the test functions types on the accuracy of the solution which the DQM Solution shows good agreement with benchmark case with using polynomials and poor agreement with using others.

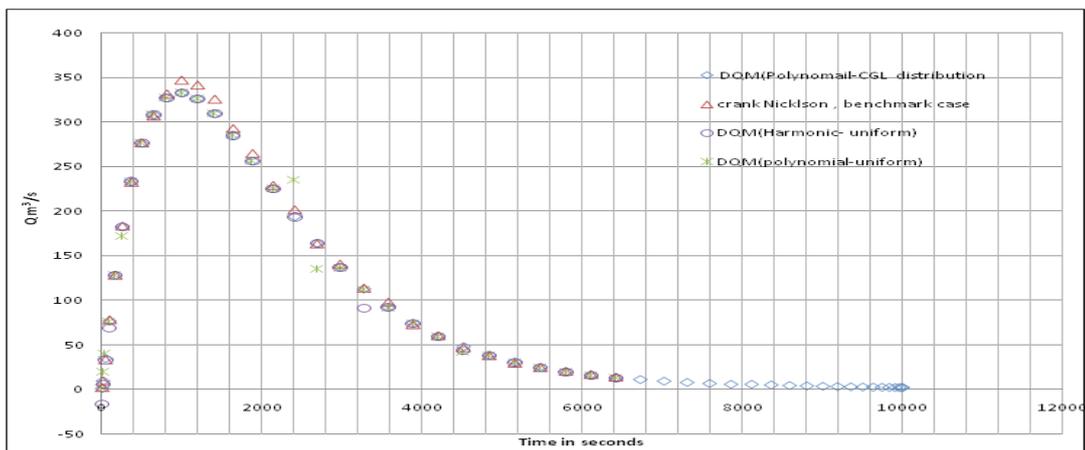


Figure 9: Effect of Grid Distribution and Test Functions on the Solution, Discharge at x=490 m

Figure 9 shows the stability of DQM solution and its smoothness against the benchmark case solution. The error which is the difference between the DQM solution (flow rate) and the benchmark case solution (flow rate) it can be seen in table 1. The DQM has high errors at the peak flow rate and get smaller after the peak as it is shown in figure 10. The error is almost zero until the peak flow rate is reached and it increased at the peak and after the peak, the error back to near zero

through the last stage of the solution as it is shown in figure 12. The error is calculated by the normal error equation as follows: $\|e\|_2 = [\sum_i |e_i|^2]^{1/n}$, $e_i = q_i^{DQM} - q_i^{benchmark}$, n = number of observations.

Where n is the number of observations. The normal error of are 1.3084 and 1.259489 for MacCormack and DQ methods, respectively.

Table 1: Accuracy of Mac Cormack Scheme Solution and DQM Solution at Distance 490 m

Time in Seconds	Crank Nicolson Method Solution	Mac Cormack Scheme Normal Error =1.3084		DQM Solution Normal Error = 1.259489	
		Absolute Error(e_i)	Relative Error	Absolute Error (e_i)	Relative Error
0	1.67	27.21	16.2934131	0.982424182	-0.588277953
10.26	9.1354	24.3054	2.66057315	0.011352482	0.001242691
41.00	32.9	42.9	1.30395136	1.514213589	0.046024729
92.11	78.001	0.316	0.00405123	1.579630081	-0.020251408
163.3	127.879	-0.421	-0.0032921	0.344591808	0.002694671
254.4	183.24	0.14	0.00076402	0.706604773	-0.003856171
365.0	232.42	0.39	0.00167799	1.048597659	0.00451165
494.6	276.58	-0.12	-0.0004338	0.051039039	-0.000184536
642.7	306.74	0.262	0.00085414	1.632596437	0.005322411
808.7	331.8	-0.24	-0.0007233	4.540763223	-0.013685242
991.96	347.32	-0.28	-0.0008061	14.48871129	-0.041715741
1191	341.75	-0.25	-0.0007315	15.46296368	-0.045246419
1406.8	325.84	-0.2	-0.0006137	16.38755486	-0.050293257
1636.9	292.9	-0.108	-0.0003687	7.765151154	-0.026511271
1880.7	264.874	-0.326	-0.0012307	8.555858624	-0.032301617

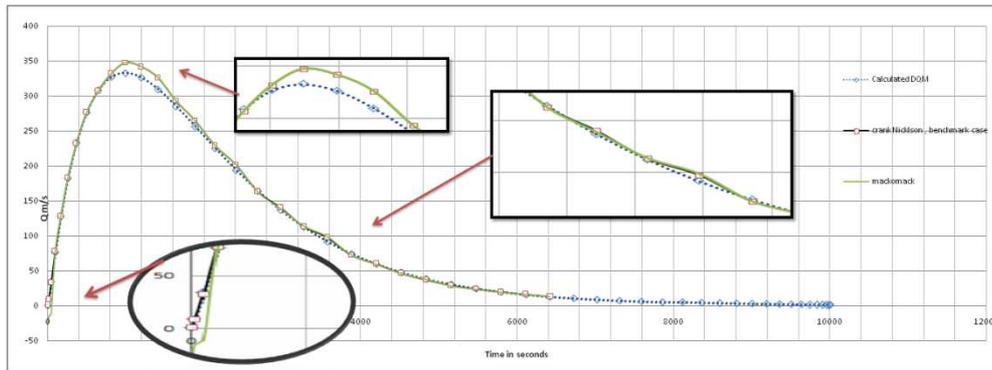


Figure 10: Solution of Real Life Case Using DQM for the Diffusion Wave, Discharge Hydrograph for the Numerical Example at x=490

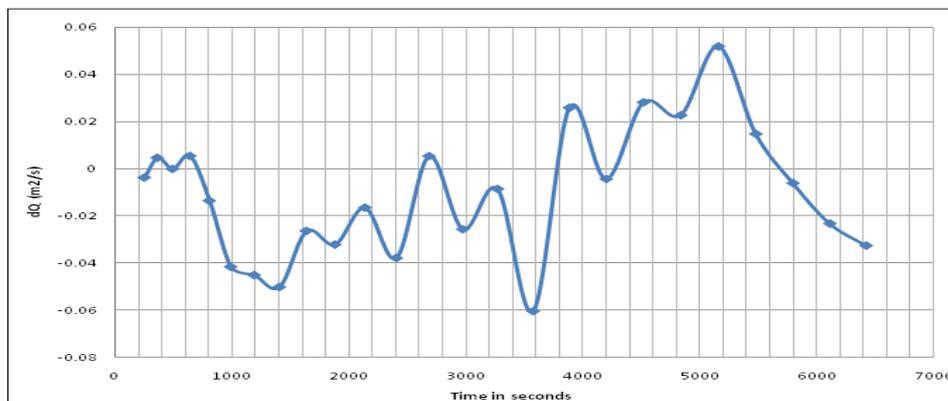


Figure 11: Numerical Relative Errors of the DQM

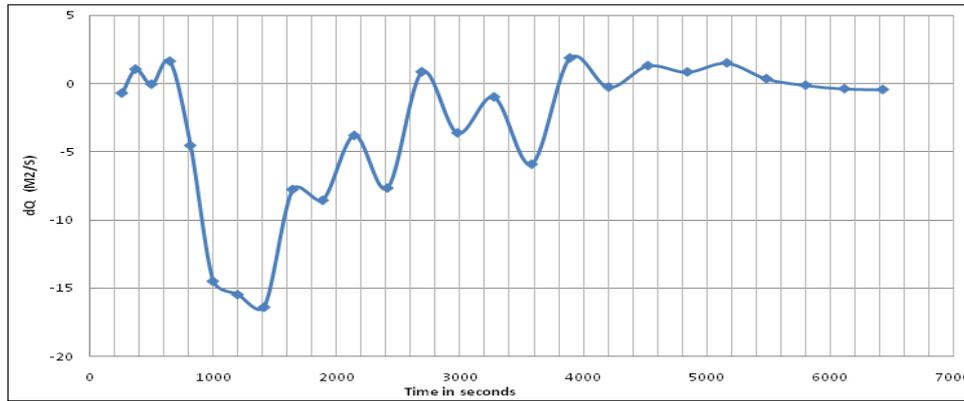


Figure 12: Numerical Absolute Errors of the DQM

Convergence Analysis of the Solution

The numerical scheme is considered as a convergent if the different is between its solution and the exact solution approach goes to zero. In the DQM solution, in order to reach the good convergence, it must be seek for sufficient grid points in both the space domain and the time domain. In the current example, the sufficient grid points in x direction is 5 and in t direction is 50 as it is seen in figure 13, 14. The convergence of the DQM solution increases with increasing the number of grid points in t direction only as can be seen in figure 13. As it is shown in figure 14, there is no effect of increasing the number of grid points in x direction on accuracy and stability of the solution.

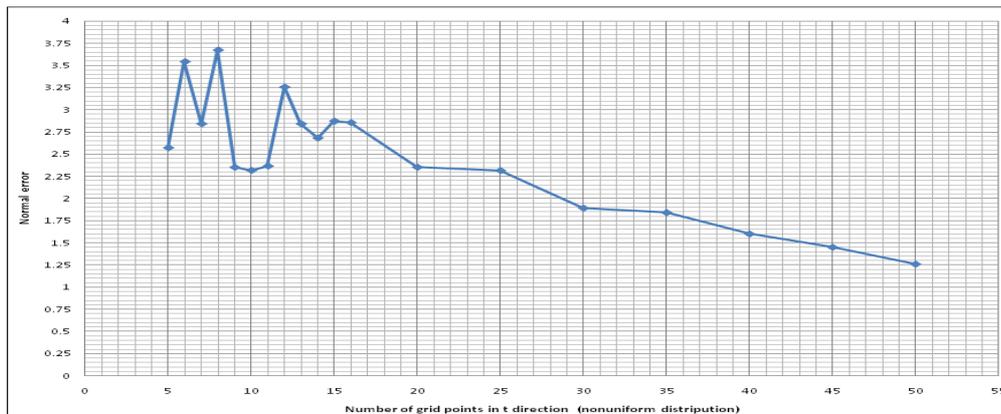


Figure 13: The Effect of Number of Grid Points in t Position

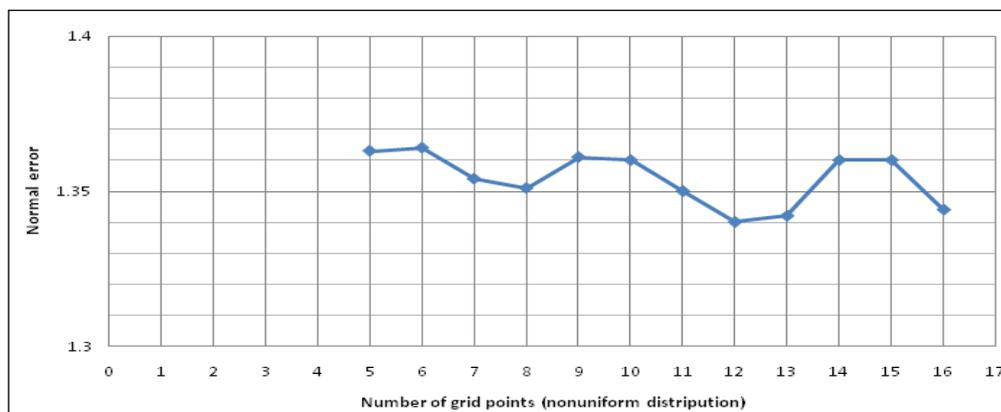


Figure 14: The Effect of Number of Grid Points in x Direction

CONCLUSIONS

In the present work, the DQM is used to solve the unsteady state one-dimensional diffusion wave model by using spreadsheet with friendly interface. This method discretizes the partial differential equations into a set of algebraic equations which can be solved by using Gauss elimination or other methods. The current work presents (USMS) model to solve shallow water-modeling problems. The excel and Matlab is integrated to produce the model results. The model results is tested with a real life example and two different schemes and its results shows good potential for solving Diffusion wave model by using DQ method. Subroutines and macros are not used in (USMS) model. Input data like initial and boundary conditions can be written in user friendly windows which is constructed on Matlab program.

The current method is applied on one example and the results are compared with their benchmark case solutions and the other methods. The results show good agreement with benchmark case. And through applying the DQM, the normal and relative errors resulted from DQM by using (USMS) model is less than the other methods. From the results above, the finite difference schemes results show poor agreement with the exact results at the beginning of simulation whereas DQM solution get away from benchmark case solution at the peak flow rate values. Moreover, the potential of the DQ method for solving Diffusion wave model was proved in stability and accuracy.

REFERENCES

1. Bajracharya, K. and Barry, D. (1997). Accuracy criteria for linearised diffusion wave flood routing. *Journal of Hydrology*, 195, pp. 200-217.
2. X. Ying, A. A. Khan, S. S. Y. Wang, Upwind conservative scheme for the Saint Venant equations, *J. Hydraul. Eng. ASCE* 130 (10) (2004) 977–987.
3. Collier, N, Radwan, H, Dalcin, L, and Calo, V. M. (2011). Diffusive Wave Approximation to the Shallow Water Equations: Computational Approach. *Procedia Computer Science*, 4, pp. 1828–1833.
4. T.C. Lackey, Sotiropoulos, Role of artificial dissipation scaling and multigrid acceleration in numerical solutions of the depth-averaged free-surface flow equations, *J. Hydraul. Eng. ASCE* 131 (9) (2005) 755–769.
5. A. M. Wasantha Lal, (2012) “Numerical Errors in Diffusion Wave Models When Simulating Kinematic Flow.” *World Environmental and Water Resources Congress 2012*.
6. Tommaso Moramarco¹, Claudia Pandolfo, and Vijay P. Singh (2012). “Accuracy of Kinematic Wave and Diffusion Wave Approximations for Flood Routing. I: Steady Analysis.”
7. Akbari G. and Firoozi B. (2010) “Implicit and Explicit Numerical Solution of Saint-Venant Equations for Simulating Flood Wave in Natural Rivers.” 5th National Congress on Civil Engineering, May 4-6, 2010, Ferdowsi University of Mashhad, Mashhad, Iran
8. Bellman, R. & Casti, J. (1971). Differential quadrature and long-term integration, *Journal of Mathematical Analysis and Applications*, Vol. 34, 235-238.
9. Shu, C. & Richards, B. E. (1992). Application of generalized differential quadrature to solve two-dimensional incompressible Navier-Stokes equations, *International Journal for Numerical Methods in Fluids*, Vol. 15, 791-798.

10. Fung, T.C. (2001). Solving initial value problems by differential quadrature method – part I: First-order equations, *Int. Journal for Numerical Methods in Eng*, No. 50, 1411-1427.
11. Kaya, B. (2010) Solution of the advection-diffusion equation using the differential quadrature method, *KSCE Journal of Civil Engineering*, Vol. 14, No. 1, 69-75.
12. Kaya, B; Arisoy, Y. & Ulke, A. (2010). Differential Quadrature Method (DQM) for Numerical Solution of the Diffusion Wave Model, *Journal of Flood Engineering*.
13. B. Kaya and Y. Arisoy, Differential quadrature solution for one-dimensional aquifer flow, *Math. Comput. Appl*, 16(2)(2011), 524-534.
14. Hashemi, M. R; Abedini, M.J. & Malekzadeh, P. (2006). Numerical modelling long waves in shallow water using incremental differential quadrature method, *Ocean Engineering*, Vol.33, 1749-1764.
15. G. Meral, Differential quadrature solution of heat-and mass-transfer equations, *Appl. Math. Modelling* 37(2013), 4350-3459.
16. R. Jiware, R. c. Mittal and K. K. Sharma, A numerical scheme based on weighted average differential quadrature method for the numerical solution of Burger’s equation, *Appl. Math. Comput*. 219(2013), 6680-6691.
17. Abdelwahab Kharab; Use of a spreadsheet program in a two-dimensional heat conduction problem , *Computer Methods in Applied Mechanics and Engineering*, Volume 122, Issues 1–2, April 1995, Pages 173-181
18. Esmail M. A. Mokheimer; Spreadsheet numerical simulation for developing laminar free convection between vertical parallel plates , *Computer Methods in Applied Mechanics and Engineering*, Volume 178, Issues 3–4, August 1999, Pages 393-412
19. Halil Karahan, M. Tamer Ayvaz; Transient groundwater modeling using spreadsheets, 4 January 2005, *Advances in Engineering Software* 36 (2005) 374–384.
20. Halil Karahan; Implicit finite difference techniques for the advection–diffusion equation using spreadsheets, *Advances in Engineering Software*, Volume 37, Issue 9, September 2006, Pages 601-608 .
21. Halil Karahan; A third-order upwind scheme for the advection–diffusion equation using spreadsheets, *Advances in Engineering Software*, Volume 38, Issue 10, October 2007, Pages 688-697.
22. Halil Karahan; Unconditional stable explicit finite difference technique for the advection–diffusion equation using spreadsheets, *Advances in Engineering Software*, Volume 38, Issue 2, February 2007, Pages 80-86.
23. Mustafa Tamer Ayvaz, Alper Elçi; Application of the Hybrid HS–Solver Algorithm to the Solution of Groundwater Management Problems, *Metaheuristics in Water, Geotechnical and Transport Engineering*, 2013, Pages 79-97
24. Enzo Martinelli, Eduardus A.B. Koenders, Antonio Caggiano; A numerical recipe for modelling hydration and heat flow in hardening concrete Cement and Concrete Composites, Volume 40, July 2013, Pages 48-58.
25. Quan, J. R. and Chang, C. T. 1989. New insights in solving distributed system equations by the differential quadrature method-I: analysis, *Computers and Chemical Engineering*, 13: 779-788.

26. Roos, H. G, Stynes, M. and Tobiska, L. 1996. "Numerical Methods for Singularly Perturbed Differential Equations", Springer, Berlin.
27. Shu. C, 1991. Generalized differential integral quadrature and application to the simulation of incompressible viscous flows including parallel computation. Ph. D Thesis, University of Glasgow, UK.
28. Shu, C and Richards, B. E. 1992. Application of Generalized differential quadrature to solve two-dimensional incompressible Navier-Stokes equations, International Journal for Numerical Methods in Fluids, 15. 7: 791-798.
29. Villadsen, J. V. and Michelsen, M. L. 1978. "Solution of differential equation models by polynomial approximation", Prentice-Hall Englewood Cliffs, New Jersey.