Chapter

Examination of Hydrologic Computer Programs DHM and EDHM

Theodore V. Hromadka II and Prasada Rao

Abstract

The Diffusion Hydrodynamic Model or DHM is a coupled one- and two-dimensional (2-D) surface flow model based upon a diffusion formulation of the well-known Navier–Stokes equations, developed by research hydrologists of the USGS (United States Geological Survey) for use in modeling floodplains and dam-break situations. The Fortran 77 source code and various applications were published in 1987 by the USGS as a Technical Report authored by Hromadka and Yen. The DHM program led to the development of several subsequent computational programs such as the FLO-2D computational model and other similar programs. The original DHM program had a limit of applications to problems with no more than 250 nodes and modeling grids. That limitation was recently removed by a program version named EDHM (Extended DHM), which provides for 9999 nodes and grids. However, the computational code is preserved in order that the baseline code algorithmic procedures are untouched. In this paper, the DHM and EDHM are rigorously compared and examined to identify any variations between the two Fortran codes. It is concluded from this investigation that the two sets of algorithm codes are identical, and outcomes from either program are similar for appropriately sized applications.

Keywords: legacy Fortran codes, computational economy, large scale application, overland flow, flow through a constriction

1. Introduction

Legacy Fortran 77 codes that have been developed in the 1980s continue to have a wide spread audience (for both research and commercial applications) across all the Computational Fluid Dynamics disciplines. Their popularity can be largely attributed to the extensive validation that these models have been subjected to with analytical and experimental data for single and multi-dimensional flows. The trust that the audience have in the end results from these legacy codes, and their ability to meet the user goals are other driving factors for making them popular among the modeling community. Few of the legacy Fortran 77 codes, developed by various groups include MFIX [1] (Open source software for simulating Multiphase Flow with Interphase eXchanges), VOF 2D [2] (Two- dimensional, transient, free-surface incompressible fluid dynamics program), FUN3D [3] (fully unstructured-grid fluid dynamic simulations spanning incompressible flow to transonic flow), LAURA [4] (Langley Aerothermodynamic Upwind Relaxation Algorithm for structured, multi-block, computational aerothermodynamic simulations) and INS3D [5] (incompressible Navier–Stokes equations in three dimensions for steady and transient flows).

Although theoretically, these legacy codes are on a firm footing, computationally, they are uneconomical. When applied over a large scale application, characterized by thousands of nodes, these codes are constrained by varying degrees. The limitations arise partly from (a) the lack of object-oriented tools and the absence of abstract modeling capabilities in Fortran 77 and (b) the required CPU time when these models are applied across large domains. Balancing the accuracy of simulation with acceptable CPU is a crucial element that the current modelers are looking for. Capturing the physics of some flow phenomena necessitates that the equations be applied across the small spatial grid and temporal scales, which can be an issue in applying legacy codes. Since the codes were written (at that time) for the then available computational resources, applying them, as they are, for large scale domains might not be feasible either due to the large CPU time that they need to complete the simulation or because of the array limitations or modifications that need to be made so that the codes can be compiled using the currently available Fortran 77 compilers. Addressing these limitations will result in a 'modernized' version of the legacy codes. Modernization does not mean a better numerical formulation or a more accurate code. It only refers to a computationally efficient code with perhaps a better user interface for input and for visualizing the results through colorful multi-dimensional graphs and tables. In fact, using any modernized code without extensive benchmark testing can lead to erroneous solutions.

The above first limitation was addressed by researchers either by rewriting the entire code from scratch using an object-oriented programming language or by using an incremental approach in which the computationally intensive modules in legacy codes are identified and replaced with their computationally efficient counterparts. For instance, in an implicit finite difference or finite element formulation, where the system of equations are assembled in the form Ax = B, much of the computational time is spent on solving for 'x' [6]. While for the application audience, a solution module merely represents a means to an end of solving the flow equations, for a solver developer, the application is a source of sparse equations to be solved. Using an appropriate solver and a preconditioner can significantly cut down the simulation time, thus partly addressing the limitation.

The advent of high-performance computing tools and their availability to all the audience (based on commodity processors) saw the evolution of legacy Fortran codes to their full or semi-parallel versions, thus addressing the second limitation in legacy codes. There are many advantages to using parallel codes. For an application modeler using a well-written parallel code, the primary advantage is the reduction in the computational time. This reduction is on the order of the number of processors used in the communicator. Parallelization allows a program to be executed by as many processors available within the sub-complex simultaneously, thus facilitating a steep reduction in the computational time required for the solution to converge to the desired simulation time. This reduction enables the code to be applied over domains with millions of nodes or cells, to better capture the physics of the flow. Using standard parallel libraries that use MPI protocols like PETSc [7], ScaLAPACK [8], and BLAS [9], existing serial codes can be converted to parallel codes with reduced effort. Alternatively, highly optimized parallel versions of legacy codes can be written from scratch, which involves higher costs for developing and testing the code.

Diffusion Hydrodynamic Model (DHM) is a legacy model developed in the 1980s for USGS by the first author and his colleague [10]. The report and the Fortran 77 source code are also available at the DHM companion web page http://

diffusionhydrodynamicmodel.com/. The model was extensively tested in the 1980s for various free surface flow scenarios, and its back engine has laid foundation blocks for other popular models like FLO-2D [11]. The DHM was originally developed as the first (or one of the first) 3-D CFD computational programs but was subsequently revised into the form published due to computer limitations of the day. In the last 30 years, DHM has proven to be a practical and reliable tool for predicting two-dimensional surface flow characteristics associated with gradually varied flows and is popular among the hydraulic community. DHM solves a simplified two-dimensional diffusion wave equation, the solution of which is sufficient for many free surface flows that are commonly encountered. For gradually varied flows, the model predicts the values of flow depth and velocity. The model does include any turbulence terms, and it cannot be used for rapidly varying flows. In this work, we address the computational limitations in DHM so that it can be applied over larger domains. The modified DHM, herein, is referred to as Enhanced DHM (EDHM).

The layout of this document is as follows. In Section 2, the flow equations and other salient characteristics in DHM are briefly described. Section 3 lists the computational limitations in DHM. In Section 4, the modifications done in DHM to arrive at EDHM are detailed. Performance tests to validate the reliability of EDHM are discussed in Section 5. Conclusions are presented in Section 6.

2. Overview of DHM

The two-dimensional flow continuity and momentum equations along the X and Y axis (assuming a constant fluid density without sources or sinks in the flow field and hydrostatic pressure distribution) can be written as [10].

$$\frac{\partial H}{\partial t}+\frac{\partial q_x}{\partial x}+\frac{\partial q_y}{\partial y}=0 \qquad (1)$$

$$\frac{\partial q_x}{\partial t}+\frac{\partial}{\partial x}\left[\frac{q_x{}^2}{h}\right]+\frac{\partial}{\partial y}\left[\frac{q_x q_y}{h}\right]+gh\left[S_{fx}+\frac{\partial H}{\partial X}\right]=0 \qquad (2)$$

$$\frac{\partial q_y}{\partial t}+\frac{\partial}{\partial y}\left[\frac{q_y{}^2}{h}\right]+\frac{\partial}{\partial y}\left[\frac{q_x q_y}{h}\right]+gh\left[S_{fy}+\frac{\partial H}{\partial X}\right]=0 \qquad (3)$$

in which q_X, q_y are the unit flow rates along the spatial directions; S_{fx}, S_{fy} represents friction slopes; and h, H, h, g denote flow depth, water surface elevation, and gravity, respectively.

The local and convective acceleration terms can be grouped and Eqs. 2 and 3 are rewritten as

$$\mathrm{m_Z}+\left[S_{fz}+\frac{\partial H}{\partial Z}\right]=0, \mathrm{z}=\mathrm{x},\mathrm{y} \qquad (4)$$

where $\mathrm{m_Z}$ represents the sum of the first three terms in Eqs. 1,2 divided by gh. Using Manning's formula to calculate the frictional slope, the flow equation can be simplified to

$$q_Z = \frac{1.486}{n} h^{5/3} S_{fz}{}^{1/2},\ z = x, y \tag{5}$$

Eq. 5 can be rewritten in the general case as

$$q_Z = -K_Z \frac{\partial H}{\partial Z} - K_Z m_Z,\ \ z = x, y \tag{6}$$

where

$$K_Z = \frac{1.486}{n}\ h^{5/3} \Big/ \left|\frac{\partial H}{\partial S} + m_S\right|^{1/2} \quad z = x, y \tag{7}$$

The symbol S in Eq. 7 indicates the flow direction which makes an angle of $\theta = \tan^{-1}\left(q_y / q_x\right)$ with the positive x-direction. By assuming the value of m to be negligible, the diffusion model can be expressed as,

$$q_Z = -K_Z \frac{\partial H}{\partial Z},\ \ z = x, y \tag{8}$$

Two-dimensional DHM is formulated by substituting Eq. 8 into Eq. 1

$$\frac{\partial H}{\partial t} = \frac{\partial}{\partial X} K_x \frac{\partial H}{\partial X} + \frac{\partial}{\partial y} K_y \frac{\partial H}{\partial y}. \tag{9}$$

If the momentum term groupings were retained, Eq. 9 can be written as

$$\frac{\partial H}{\partial t} = \frac{\partial}{\partial x} K_x \frac{\partial H}{\partial x} + \frac{\partial}{\partial y} K_y \frac{\partial H}{\partial y} + S \tag{10}$$

where

$$S = \frac{\partial}{\partial x}\left(K_x m_x\right) + \frac{\partial}{\partial y}\left(K_y m_y\right),$$

and $K_x,\ K_y$ are also functions of $m_x,\ m_y$ respectively.

To maintain continuity in the discussion, while salient aspects of the DHM numerical algorithm are presented here, readers are referred to [10] for a detailed description of the numerical formulation and the input file format. The domain is divided into uniform square grids or cells. For each interior grid, its connectivity with adjacent grids along the North, East, South, and West directions is specified. For grids that are on the boundaries, '0' is specified along the directions that do not have adjacent nodes. The flow equation is solved using the nodal domain integration method. Apart from the grid connectors, at the center of each cell, the required input variables are the roughness value, ground elevation, initial flow depth. The number of inflow cells and the inflow hydrograph at each of them need to be inputted. The number and the outflow boundary cell numbers should be identified. Since the formulation is explicit, the choice of time step (Δt) is limited by the Courant-Friedrich-Lewy stability condition. Starting from time = 0, the explicit solution is marched in the direction of time, until the required transient time level is reached. DHM gives the option of printing the output variables at any time level. The output at the center of each cell includes the flow depth, elevation, and flow velocities along the four directions.

3. Computational limitations in DHM

For modeling flows over large sizes domain, the two primary shortcomings in DHM are

a. The maximum number of cells (nodes) that can be accommodated in DHM is limited to 250

b. Inflow and outflow boundary nodes are limited to 10

Both these limitations were largely due to the computational resources that were available to the developers in the 1980's. Application of DHM over large flow domains would require using a higher number of nodes in the computational domain, warranting modifications to DHM, as discussed next.

4. Features in extended DHM (EDHM)

The changes made in DHM can be grouped into three categories, as detailed below. No changes were made to the format or structure of variables input and output files.

4.1 Major enhancements

a. Increased the array size of variables (FP,FC) that stores the location of cells and the initial depth, average elevation, roughness coefficient values, velocity (VEL), maximum water depth, and the corresponding time at the cell (DMAX, TIMEX) from 250 to 9999.

b. Increased the array size of variables that store the stage curve data for the channel (NOSTA, STA) from 10 to 99.

c. Increased effective rainfall intensity data pairs from 10 to 99.

d. Increased the array size of variables relating to the inflow and outflow hydrograph nodes (KIN, KOUT), depth of the specified stage-discharge curve (HOUT), inflow boundary condition nodal points (KINP) and the inflow hydrograph details (HP) from 10 to 99

e. Increased of the array which stores the nodal points where outflow hydrographs are being printed (NODFX, NODCC) from 50 to 99

The changes in the above array sizes were done in the main code and the associated subroutines FLOODC, QFP, QFC, and CHANPL.

DHM		EDHM	
Line #	Content	Line #	Content
12	COMMON/BLK 1/FP(250,8),FC(250,6)	12	COMMON/BLK 1/FP(9999,8),FC(9999,6)
14	COMMON/BLK 2/KIN(10),H(10,15,2), KOUT(10),HOUT(10,15,3)	14	COMMON/BLK 2/KIN(99),H(99,15,2), KOUT(99),HOUT(99,15,3)
16	COMMON/BLK 3/NOSTA(10),STA(10,15,2),NODFX(50)	16	COMMON/BLK 3/NOSTA(99),STA(99,15,2),NODFX(99)
18	COMMON/BLK 4/DMAX(250,2),TIMEX(250,2)	18	COMMON/BLK 4/DMAX(9999,2),TIMEX(9999,2)
20	COMMON/BLK 5/KINP(10),HP(10,15,2)	20	COMMON/BLK 5/KINP(99),HP(99,15,2)
26	DIMENSION NODDC(50),VEL(250,4),R(10,2),Q(4)	26	DIMENSION NODDC(99),VEL(9999,4),R(99,2),Q(4)
271	FORMAT(10X,5I4,1X,F6.4,2X,F6.1,1X,F5.1)	271	FORMAT(10X,5I5,1X,F6.4,2X,F6.1,1X,F5.1)
281	FORMAT(/,10X,'INFLOW HYDROGRAPH AT NODE #’,I3,/,	281	FORMAT(/,10X,'INFLOW HYDROGRAPH AT NODE #’,I4,/,
291	FORMAT(10X,I3,1X,I3)	291	FORMAT(10X,I4,1X,I4)
297	FORMAT(10X,I3 ,2X,F5.4,1X,F7.1,1X,F7.1,1X,F7.1,5X,F7.1)	297	FORMAT(10X,I4 ,2X,F5.4,1X,F7.1,1X,F7.1,1X,F7.1,5X,F7.1)
303	FORMAT(10X,'OUTFLOW NODE # ‘,I3,	303	FORMAT(10X,'OUTFLOW NODE # ‘,I4,
311	FORMAT(/,10X,'STAGE CURVE AT NODE #’,I3,/,	311	FORMAT(/,10X,'STAGE CURVE AT NODE #’,I4,/,
333	FORMAT(10X,'INFLOW RATE AT NODE ‘,I3,’ IS EQUAL TO ‘,F10.2)	333	FORMAT(10X,'INFLOW RATE AT NODE ‘,I4,’ IS EQUAL TO ‘,F10.2)
335	FORMAT(/,5X,'NODE’,7X,10(I3,8X))	335	FORMAT(/,5X,'NODE’,7X,10(I4,7X))
357	FORMAT(10X,'OUTFLOW RATE AT NODE ‘,I3,’ IS EQUAL TO ‘,F10.2)	357	FORMAT(10X,'OUTFLOW RATE AT NODE ‘,I4,’ IS EQUAL TO ‘,F10.2)
1275	COMMON/BLK 1/FP(250,8),FC(250,6)	1275	COMMON/BLK 1/FP(9999,8),FC(9999,6)
1277	COMMON/BLK 2/KIN(10),H(10,15,2), KOUT(10),HOUT(10,15,3)	1277	COMMON/BLK 2/KIN(99),H(99,15,2), KOUT(99),HOUT(99,15,3)
1279	COMMON/BLK 3/NOSTA(10),STA(10,15,2),NODFX(50)	1279	COMMON/BLK 3/NOSTA(99),STA(99,15,2),NODFX(99)
1281	COMMON/BLK 4/DMAX(250,2),TIMEX(250,2)	1281	COMMON/BLK 4/DMAX(9999,2),TIMEX(9999,2)
1567	COMMON/BLK 1/FP(250,8),FC(250,6)	1567	COMMON/BLK 1/FP(9999,8),FC(9999,6)
1647	COMMON/BLK 1/FP(250,8),FC(250,6)	1647	COMMON/BLK 1/FP(9999,8),FC(9999,6)

DHM		EDHM	
Line #	**Content**	**Line #**	**Content**
1727	COMMON/BLK 1/FP(250,8),FC(250,6)	1727	COMMON/BLK 1/FP(9999,8),FC(9999,6)

Table 1.
Detailed listing of the differences between the DHM and EDHM source codes.

4.2 Minor enhancements

The two minor enhancements that were made in DHM code are (a) to accommodate the increased number of cells in EDHM, the fixed format output descriptor has been expanded by one digit and (b) to better align the variables in the output file, the inter variable spacing was decreased by one digit. A detailed listing of all the major and minor changes made in the DHM source code, along with the corresponding line numbers, is shown in **Table 1**.

4.3 Compiler details

After reviewing the currently available compilers in Windows for Fortran 77 codes, we have chosen the Intel Fortran Compiler within the Microsoft Visual Studio integrated development environment (IDE) to make the enhancements in DHM and for generating the.EXE file. This interface is ideal to debug and execute Fortran 77 programs. The compiler can optimize the performance of source codes for Intel CPUs. It offers broad support for current and previous Fortran standards and also tools by which a robust, high-performance code can be created in serial and parallel environments. The Math Kernel Library (Intel MKL) and the Debugger tools in the compiler, creates a solid foundation for building robust, high-performance codes. The end executable file (.EXE) although optimized for Intel CPUs, can also run on x86 compatible CPUs such as those from AMD.

5. Application of EDHM

EDHM is qualitatively and quantitatively evaluated by comparing its results with the output from DHM, for two test simulations. The focus was to check if the EDHM solution resembles DHM output for these two cases for varying inflow and other model parameters.

Case 1: Flow in a transition.

Open channel flow through a linear contraction under the framework of two-dimensional flow is a common phenomenon and has drawn the attention of many experimental and numerical studies. The flow characteristics in the contraction depend on the Froude number at the upstream end. Flow in a contraction has acted as a benchmark simulation in investigations [12] that compared the performance of various CFD and hydraulic models. **Figure 1** is the definition sketch of the test problem. The rectangular channel is 380 ft. long and 260 ft. wide. The constriction

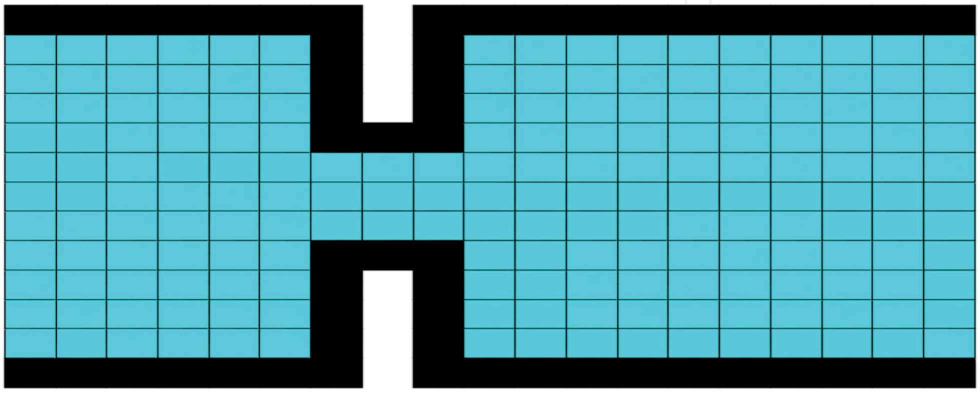


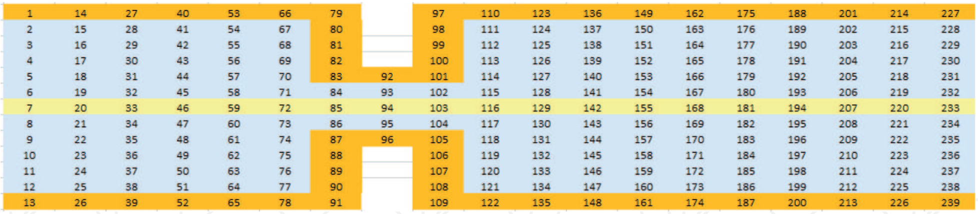
Figure 1.
Definition sketch of the test problem (channel length = 380 ft., channel width = 260 ft., cell size = 20 ft).

portion of the channel is 60 ft. x 60 ft. The channel length before and after constriction is 120 ft. and 200 ft., respectively. The cell size in the domain is 20 ft., and the total number of cells are 239. **Figure 2** illustrates the cell numbers in the domain. The elevation of cells along the north and south boundaries was assigned a high value to physically denote that they are walls. The flow is confined within these boundaries. At the upstream end, cells 3-11 (nine cells) were specified with a constant inflow of 33.33 cfs. At the downstream end, a free outfall boundary was specified. The transient simulation was carried out until time = 1 hour.

Figure 3 plots the water depth profile for the two models along the channel centerline at time = 0.9 hours for a channel bottom Manning's roughness value of 0.015. The close agreement of results gives confidence that the changes made to arrive at EDHM did not lead to different output data. The analysis over varying inflow discharge and bottom roughness values did not change the trend of the results. **Figures 4** and **5** compare the depth profile for the roughness coefficient of 0.024 and 0.04. A similar agreement in output data (including velocity components) was observed across other longitudinal sections.

Case 2: Overland flow.

Overland flow over a hill slope generated by a rainfall event is characterized by varying hydraulic properties, roughness values, topography, and physical features in the domain. For predicting the hydraulic and hydrologic properties of flow, various models that solve a range of equations from a one-dimensional hydrodynamic equation for homogeneous place surfaces [13, 14] to 2D full non-linear shallow water Equations [15] have been applied. **Figure 6** shows the flow domain. The number of cells in the domain is 56 and are 30 ft. in size. The roughness value ranged between 0.015 to 0.03Inflow hydrograph (**Figure 7**) was applied at cells 1 and 15. The transient model was run until time = 2 hours. The elevation drop along the north and south end of the domain is 26 ft., and the drop along the west and east boundaries is 5 ft. Cells 42 to 56 were specified as critical depth outflow



1	14	27	40	53	66	79		97	110	123	136	149	162	175	188	201	214	227
2	15	28	41	54	67	80		98	111	124	137	150	163	176	189	202	215	228
3	16	29	42	55	68	81		99	112	125	138	151	164	177	190	203	216	229
4	17	30	43	56	69	82		100	113	126	139	152	165	178	191	204	217	230
5	18	31	44	57	70	83	92	101	114	127	140	153	166	179	192	205	218	231
6	19	32	45	58	71	84	93	102	115	128	141	154	167	180	193	206	219	232
7	20	33	46	59	72	85	94	103	116	129	142	155	168	181	194	207	220	233
8	21	34	47	60	73	86	95	104	117	130	143	156	169	182	195	208	221	234
9	22	35	48	61	74	87	96	105	118	131	144	157	170	183	196	209	222	235
10	23	36	49	62	75	88		106	119	132	145	158	171	184	197	210	223	236
11	24	37	50	63	76	89		107	120	133	146	159	172	185	198	211	224	237
12	25	38	51	64	77	90		108	121	134	147	160	173	186	199	212	225	238
13	26	39	52	65	78	91		109	122	135	148	161	174	187	200	213	226	239

Figure 2.
Flow domain with the cell numbers. The wall boundaries are identified by orange-colored cells. There are 239 cells in the domain. The centerline cells (7,20,..220,233) where the depth profiles are compared are highlighted.

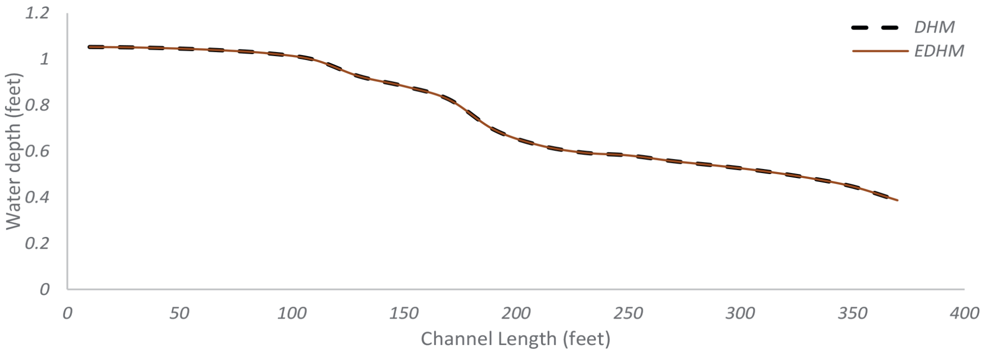


Figure 3.
Comparison of depth profiles along the centerline of the channel at time = 0.9 hours (roughness = 0.015).

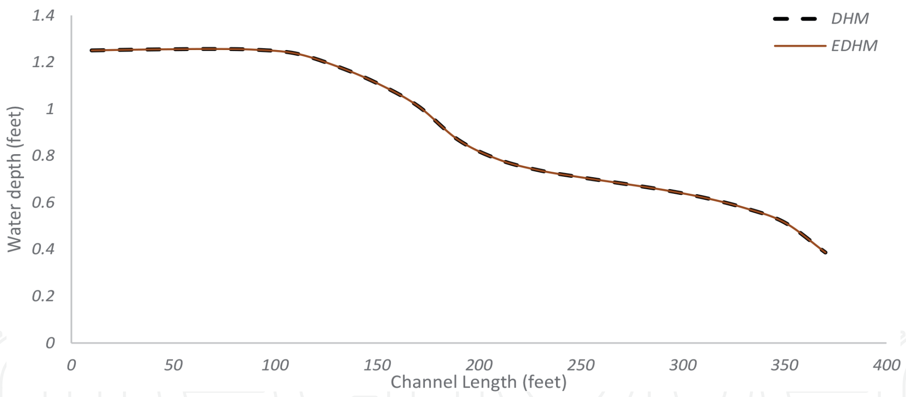


Figure 4.
Comparison of depth profiles along the centerline of the channel at time = 0.9 hours (roughness = 0.024).

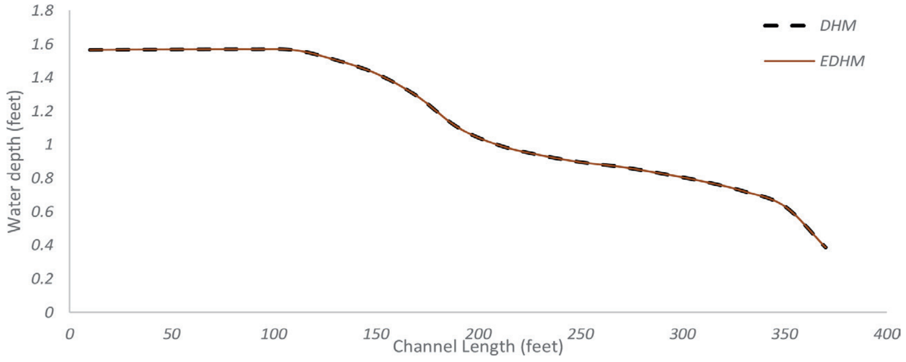


Figure 5.
Comparison of depth profiles along the centerline of the channel at time = 0.9 hours (roughness = 0.04).

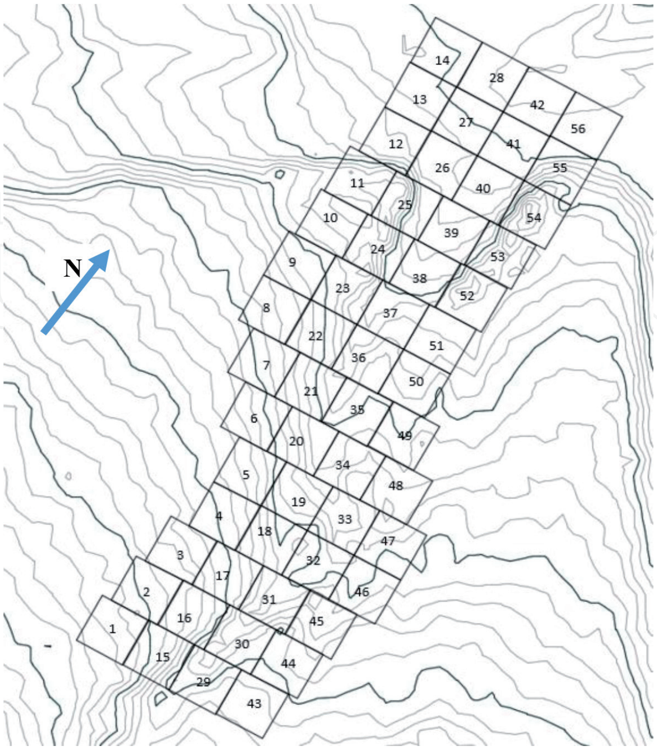


Figure 6.
Overland flow domain with the cell numbers. The domain has 56 cells.

nodes. **Figures 8** and **9** compare the transient depth profiles from DHM and EDHM (time = 0 to 2 hours) at cells 36 and 26, respectively. The trend of the depth and velocity values at other cells also indicated the close agreement of flow data from the two models.

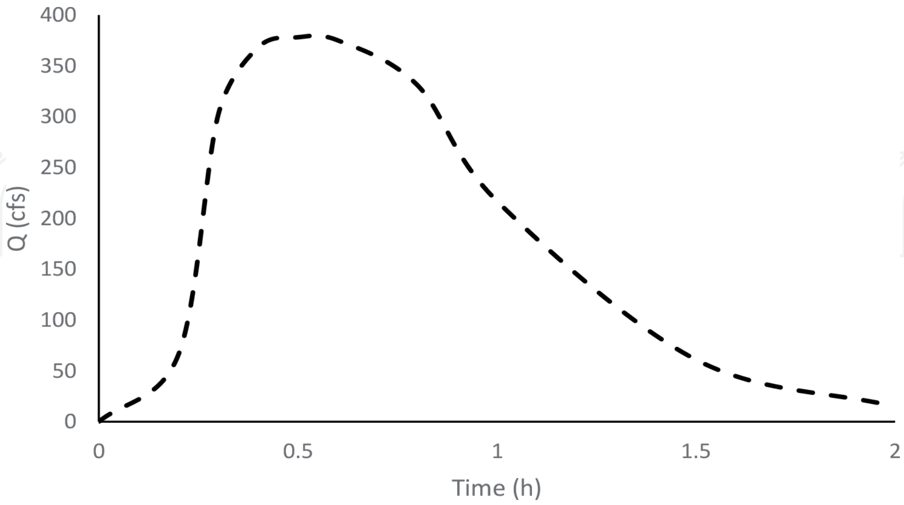


Figure 7.
Inflow hydrograph data at cells 1 and 15.

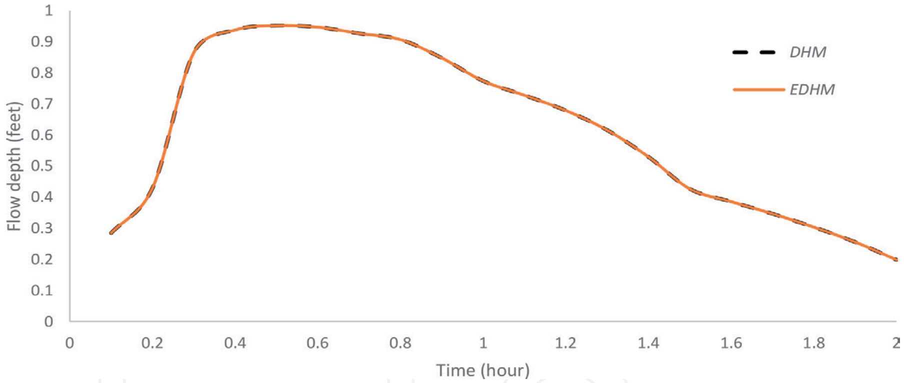


Figure 8.
Transient depth profiles at cells 36.

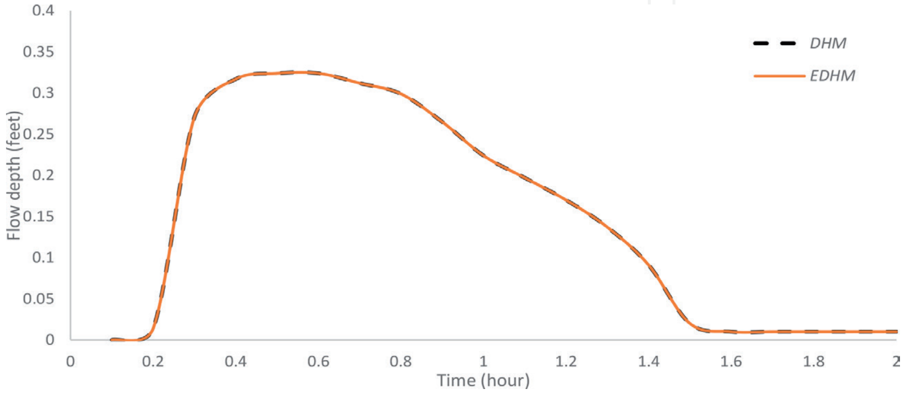


Figure 9.
Transient depth profiles at cell 26.

6. Conclusions

USGS Diffusion Hydrodynamic Model is a legacy Fortran 77 that has been widely applied for multiple one and two-dimensional flow scenarios. The computational limitations in the model which prevent its application over large domains have been addressed in this paper. The enhanced model can accommodate 9999 cells and 99 inflow and outflow nodes. Based on the analysis that was carried out and the close agreement of the results between the two models gives confidence in the reliability of the extended model. Current findings encourage future development of parallel EDHM in order to reduce the computational time.

Author details

Theodore V. Hromadka II [1] and Prasada Rao[2]*

1 Department of Engineering-Mathematics, United States Military Academy, West Point, NY, United States of America

2 Department of Civil and Environmental Engineering, California State University, Fullerton, CA, United States of America

*Address all correspondence to: mprasadarao@fullerton.edu

References

[1] Syamlal, M. MFIX documentation. User's manual, U.S. Dept. of Energy, DOE/METC-95/1013, DE95000031, https://mfix.netl.doe.gov/doc/mfix/20.1.0/, 2004.

[2] Torrey MD, Cloutman LD, Mjolsness RC and Hirt, CW.NASA-VOF2D: A Computer Program for Incompressible Flows with Free Surfaces.LA-10612-MS(December 1985) http://www.oecd-nea.org/tools/abstract/detail/nesc9644/

[3] Nielsen E, Park MA, Rumsey CL, Thomas JL, and Wood WA. FUN3D Manual: 13.0," NASA TM-2016-219330, Langley Research Center, Aug. 2016. https://fun3d.larc.nasa.gov/

[4] Mazaheri A, Gnoffo PA, Johnston CO, and Kleb B. LAURA Users Manual: 5.5-65135, NASA TM-2013-217800, February 2013. https://ntrs.nasa.gov/search.jsp?R=20130009520

[5] Rogers, SE, Kwak D, and Chang JL. INS3D-an Incompressible Navier-Stokes Code in Generalized Three-Dimensional Coordinates, NASA TM-100012. 1987. https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19880003609.pdf

[6] Eriksson K, Estep D, Hansbo P. Johnson C. Computational Differential Equations: Cambridge Univ. Press; 1996

[7] Balay S, Brown J, Buschelman K, Eijkhout V, Gropp WD, Kaushik D, Knepley MG, McInnes LC, Smith BF, Zhang H. PETSc users manual. Tech. Rep. ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013.

[8] Blackford LS, Choi J, Cleary A, D'Azevedo E, Demmel J, Dhillon I, et al. Guide. SIAM; 1997

[9] Chtchelkanova A, Gunnels J, Morrow G, Overfelt J, van de Geijn RA. Parallel implementation of BLAS: General techniques for level 3 BLAS. In: TR-95-40, Department of Computer Sciences, University of Texas. Oct. 1995

[10] Hromadka II TV, Yen CC. A diffusion hydrodynamic model. Water resources investigations report. U.S. geological survey; 1987: 87– 4137. https://pubs.er.usgs.gov/publication/wri874137

[11] O'Brien JS, Julien PY, Fullerton WT. Two-dimensional water flood and mudflow simulation. Journal of Hydraulic Engineering ASCE. 1993;**119**:244-261

[12] Paudel M, Roman SB, Pritchard J. A Comparative Study of HEC-RAS 2D, TUFLOW, & Mike 21, Presented at ASFPM 2016 Annual National Conference. MI: Grand Rapids; 2016

[13] Liggett JA, Woolhiser DA. Difference solutions of the shallow-water equation, J. Eng. Mech. Div. Am. Soc. Civ. Eng. 1967;**9J**(EM2):39-71

[14] Akan AO, Yen BC. Mathematical modeling of shallow water flow over porous media. J. Hydraul. Div. Am. Soc. Civ. Eng. 1981;**W7**(HY4):479-494

[15] Singh J, Altinakar MS, Ding Y. Numerical modeling of rainfall-generated overland flow using non-linear shallow-water equations. Journal of Hydrologic Engineering. 2015;**20**(8). DOI: 04014089, 10.1061/(ASCE)HE.1943-5584.0001124