Concept of Turbulence Modelling and Its Application in Open Channel Flow

P. Singh[1], A. Kumar[2], K.K. Khatua[3]
1,2, M.Tech Research Scholar, Department of Civil Engineering, National Institute of Technology Rourkela, India. Email: prateek.k.singh1992@gmail.com, arunkumar5726@gmail.com
3 Associate Professor, Department of Civil Engineering, National Institute of Technology Rourkela, India. Email: kkkhatua@yahoo.co.in

ABSTRACT

The foremost objectives of the particular study are to assess the working of the most popular RANS turbulence closure model in simulating river flows. For that reason, ANSYS-FLUENT package was used considering the simulation of uniform flows in straight channel with several different RANS turbulence closure models. Two approaches were used for the turbulent region and near-wall region, which are the standard k-ϵ model and a k-ω model respectively. On the other hand, Reynold Shear Model (RSM) is also studied upon and its pros and cons are taken into consideration with respect to the models like k-ϵ model and k-ω model. In comparison to the newly conducted experimental data under controlled system, models using ANSYS–FLUENT, a Computational Fluid Dynamics (CFD) code, are found to give suitable results. Computational Fluid Dynamics (CFD) is frequently used to as technique to investigate flow structures in developing areas of a flow field for the determination of velocity, pressure, shear stresses, effect of turbulence and others. A two phase three-dimensional CFD model along with the two equations isotropic eddy viscosity model and Reynold stress model (RSM) is used to solve the turbulence equation. This study aims to validate CFD simulations of free surface flow or open channel flow by using Volume of Fluid (VoF) in multiphase scheme and viscous (laminar) scheme by comparing the data observed in by the UK Flood Channel Facility (FCF).

Keyword: - RANS turbulence model, K-ω model, K-ϵ model, RMS model, ANSYS-FLUENT

1. INTRODUCTION

Foretelling the maximum water levels and discharge is a difficult but important factor for engineers in flood modelling. The fundamental link between discharge and water level is important, since flood risk maps should give and accurate estimate of water levels and possible inundation areas (Knight 2013). Determining the stage-discharge relationship for channels with complex cross section, like compound channel, is not a simple matter (Knight and Shamseldin 2006; Knight *et al*. 2010). The modeler needs adequate turbulence data and complete awareness of the flow behavior for unfailing predictions of stage-discharge relationships.

Practitioners, in general, still use 1D model such as shallow water equation for evaluating flow conditions in real time setups, where a roughness coefficient accounts for all 3D effects (Morvan *et al*. 2008), making it specific and obliquely increasing uncertainty. Recently, a more often 2D (streamwise and lateral directions) models are being used. They unambiguously can clearly account variations in the cross-section shape and area, which includes floodplain flow and meandering channels (Wright 2001). Despite that, the roughness coefficients still bear some uncertainty, since they have to account for all the vertical processes that are not modeled (Morvan *et al*. 2008). These can be relevant when secondary flow (*i.e*. flow circulations transverse to the main downstream flow direction also known as helical flow) is important, like the ones occurring in converging diverging channel, channel bends, floodplain/channel interactions and to a lesser extent in straight channels (Wright 2001). Although, the 3D models use in real time scenario i.e. river configurations is still not common, both the increment of computational competence and the requirement of more physically based predictions will generate urgency for their use in the near future. Moreover, 3D models offer more consistent estimates of bed shear stress and other more useful information, such as the three-dimensional flow

field important for mixing processes (*cf*. Lane *et al*. 1999). In this context, the use of CFD commercial codes seems more probable to occur, rather than the use of research codes developed in the academia. Mainly, because the former are user-friendly and incorporate most of the turbulence closure models, starting from the simplest one- or two-equation models to a more advanced large eddy simulation (LES). It is important to notice that the majority of commercial codes stemmed from aerodynamics industrial applications. They often present additional shortcomings, like "hidden" default strategies (Knight 2013), using default values for the empirical coefficients that sometimes cannot be changed. Nevertheless, commercial models have demonstrated their ability in simulating laboratory open-channel flows, giving results in good agreement with the experiments (Morvan *et al*. 2002, Morvan 2005). Even so, their validation within river flow configurations is still far to be considered fully accomplished. The user will have to face a unwarranted choice about the model that should be taken into consideration, even if a CFD commercial package is accessible. The choice is completely swings between the accuracy of the results and the computational resource availability. Time required for the computation always plays a key role in the selection of the model. At present, the use of LES and Direct Numerical Simulation, DNS (this one is not usually available in commercial packages), can be discarded, due to the large quantity of data to manage and to the exceptionally high computational time required. This leaves as viable alternatives the models based on Reynolds Averaged Navier-Stokes (RANS) equations coupled with a turbulence closure model. Within this "family" of models, one can choose from less demanding one- or two-equation models to the more demanding models based on Reynolds stress transport equations.

1.2. Objectives and methodology

The main objectives of the present study are:
i) To provide users of commercial CFD packages modeling guidelines regarding RANS turbulence model and their computational domain modeling approach;
ii) Also to provide users of commercial CFD packages a clear picture of the performance of most common RANS turbulence closure models in simulating river flows;

To achieve these objectives 3D simulations of FCF data Phase A series 1 datasets were used and performed using the commercial package ANSYS-FLUENT. The simulated experiments correspond to straight trapezoidal channel configuration with high flow stages (relative depth, defined as the relation between the flow depth in the floodplain and the one in the main channel, equal to 0.092 and bed slope S=0.00103). Since the main focus was on accuracy *vs*. computational time required for different turbulence closure models, it was decided to start with the less time demanding, and most used, k-ε model and then continue with the use of more complex and time demanding turbulence closure models. At the end, two different turbulence models were used, which are, in increasing order of complexity: *RSM* model and *RSM*-ω based model. Shear stress transport model (SST) that uses a mixture of both k-ε and k-ω models is adopted and only velocity contour is shown to explain the nature of isotropic modeling.

2. MODELLING APPROACH

Computational fluid dynamics (CFD) is a branch of fluid mechanics that uses numerical methods to solve fluid flow governing differential equations by means of computer-based simulations. The fundamental bases of almost all CFD problems are the Navier-Stokes equations. One of the earliest types of calculations resembling modern CFD are those by Lewis Fry Richardson (1881-1953). For these calculations he used finite difference approximations and divided the physical space into grid cells. Although they were not fully accurate, these calculations together with Richardson's book (1965) set the basis for modern CFD.
The structure of ANSYS-CFX can be presented as a system of five cells (Table 1).

Table 1. Structure of ANSYS-FLUENT

ELEMENT OF CODE	Cell	Function
	Geometry	Generation of the geometry
Pre-processor	Mesh	Grid Generation
	Setup	Definition of initial and boundary conditions and turbulence model
Solver	Solution	Definition of the run and monitoring of the convergence
Post-processor		Visualization of the results and their treatment

After the geometry has been defined the next step is to mesh the computational domain. The objective of meshing is to divide the domain into a number of smaller elements (control volumes). The accuracy of a CFD solution partially depends on the number of cells and on the quality of the mesh. Optimal meshes are often non-uniform: finer in areas where high velocity gradients occur and coarser in regions with relatively little change (Versteeg and Malalasekera 2007). When creating the mesh, one should keep in mind that balance should exist between the accuracy of a solution through the number of grid elements generated and its cost in terms of necessary computer hardware and calculation time. The most basic form of mesh classification is based upon the connectivity of the mesh: structured (Figure 1(a)) and unstructured (Figure 1(b)). The structured mesh is considered when cells (nodes) are arranged in rows and columns (not necessary equally spaced, but following the same pattern as the geometry), thus, limiting the type of elements to quadrilaterals (for 2D geometries) and hexahedrons (for 3D geometries). Unstructured mesh is considered when cells and nodes are not arranged in rows and columns (Figure 1(b)). In unstructured meshes for 3D geometries the next types of elements can be generated: prisms, pyramids and tetrahedrons. Tetrahedral mesh might use up to six times as many elements as hexahedral one, thus resulting in more computationally expensive mesh for the same number of nodes (Chu *et al.* 2009). The disadvantages of structured grid are that it is limited to simple geometries and it is time consuming to create a high quality mesh, while unstructured grid can be generated very fast for very complex geometries.

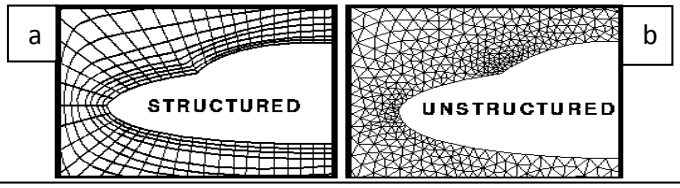

Figure 1. Represntation of structured and unstructred grid (adapted from https://www.nas.nasa.gov/Software/FAST/RND-93-010.walatka-clucas/htmldocs/chp_16.surferu.html)

The next and final step of the pre-processing is to create input required by the Solver, such as fluid properties, initial and boundary conditions, turbulence model, numerical scheme for the advection term, convergence criteria and monitoring. The component that solves the CFD problem is called **Solver**.

The numerical algorithm consists of the following steps:

1. The governing differential equations of fluid flow are integrated over all the (finite) control volumes of the region of interest.
2. Discretization - the integral equations are converted into a system of algebraic equations.
3. The system of algebraic equations is solved iteratively.

An iterative approach is required because of the non-linear nature of the equations. The solution is converged when so-called residuals - measures of the overall conservation of the flow properties - are

very small. A moderate-large number of iterations are usually required to reach a converged solution. In the present study the simulations were stopped when the convergence criteria were met. The level of convergence was evaluated based on average values of the residuals. Typically for practical simulations the aim is to decrease the residuals to an order of 10^{-4}**.** However, depending on the class of problem under investigation, it may be necessary to decrease the residuals further to 10^{-6} or even more (Knight *et al*. 2005).
The Solver produces a results file that is then passed to the **post-processor** where the resulting solution can be visualized and analyzed. At the end of the simulation the user has to judge whether the results are "good enough". It can be done by comparing the numerical results with the experimental data.

2.1. Discretization and solution theory

The CFD algorithms based on discretization of the computational domain stand on four numerical solution techniques: finite difference, finite volume, finite element and spectral methods. There are also some CFD algorithms in the field of numerical simulation that do not require discretization of the simulation domain, which are called mesh-free methods. CFX uses Finite-Element-based Finite Volume method (FVM). In this variant, the control volumes are vertex-centred, where the solution variables are calculated and stored at the vertices (nodes) of the mesh. This type of mesh is called non-staggered, or co-located, grid. The Finite Element (FE) basis comes from the use of shape functions (also known as trial functions, interpolation functions or basis functions, (Chung 2002), common in FE techniques, to describe the way a variable changes across each element. The objective of the FVM is the transformation of the partial differential equations into the system of algebraic equations, which then should be solved. The process of the discretization consists of two steps: discretization of the computational domain and discretization of the equations.

All the conservation equations can be written in the same general convection-diffusion form:

$$\underbrace{\frac{\partial}{\partial x_j}\left(\rho U_j \phi\right)}_{\text{Convection term}} = \underbrace{\frac{\partial}{\partial x_j}\left(\Gamma_\phi \left(\frac{\partial \phi}{\partial x_j}\right)\right)}_{\text{Diffusion term}} + \underbrace{S_\phi}_{\text{Source term}} \tag{1}$$

where ϕ is any variable of interest, Γ_ϕ is the diffusion coefficient, and S_ϕ is a source term.
After the computational domain has been discretized, governing equations (1) are integrated over each control volume.

$$\int_S \rho U_j \phi dn_j = \int_S \Gamma_\phi \left(\frac{\partial \phi}{\partial x_j}\right) dn_j + \int_V S_\phi dV \tag{2}$$

where V and S denote volume and surface regions of integration, respectively, and dn_j are the differential Cartesian components of the outward normal surface vector. Equation (2) represents the flux balance in a control volume. The next step in the numerical algorithm is to discretize the volume and surface integrals from equation (2). Volume integrals are converted into surface integrals using Gaussian divergence theorem. Surface integrals are discretized at the integration points (ip_n) located at the center of each surface segment within an element and then distributed to the adjacent control volumes. The surface integrals are locally conservative because they are equal and opposite for control volumes adjacent to the integration points.
After discretizing the volume and surface integrals the integral equation (2) becomes:

$$\sum_{ip} (\rho U_j \Delta n_j)_{ip} \phi_{ip} = \sum_{ip} \left(\Gamma_\phi \left(\frac{\partial \phi}{\partial x_j}\right) \Delta n_j\right)_{ip} + V\overline{S_\phi} \tag{3}$$

where the subscript ip denotes evaluation at an integration point (summations are over all integration points of the control volume), Δn_j is the discrete outward surface vector, $(\rho U_j \Delta n_j)_{ip}$ is the mass flow

rate across the surface of the control volume estimated at the integration point, ϕ_{ip} denotes the value of the variable at the integration point, V is the control volume and $\overline{S_\phi}$ is the average value of S_ϕ throughout the volume.

2.2. Numerical differentiation schemes

There are many differencing schemes available, but more accurate schemes tend to be less robust or slower. Some of the differencing schemes offered by CFX are described in this section. The advection term requires the integration point values ϕ_{ip} to be approximated in terms of the nodal values of ϕ. The advection schemes implemented into ANSYS CFX can be written in the next form:

$$\phi_{ip} = \phi_{up} + \beta \nabla \phi \Delta \vec{r} \quad (4)$$

where ϕ_{up} is the value at the upwind node and $\vec{r}$ is the vector from the upwind node to the *ip*.
The choice of β and ∇ ϕ yields different schemes as described below.

The **central differencing scheme** is one of the schemes that has been used widely to represent the diffusion terms in steady diffusion problems. It can be obtained by setting β to 1 and ∇ ϕ to the local element gradient. The central differencing scheme has second order accuracy, but it not able to identify the flow direction, which makes it not suitable for general purpose flow calculations, and thus creating the need for other schemes. This type of schemes allows to reproduce steep spatial gradients more accurately than first order schemes, but they generate non-physical oscillations (numerical dispersion) in regions of rapid solution variation.

The particular choice of β = 0 yields the **upwind differencing scheme**. The convected value of ϕ at a cell face is taken to be equal to the value at the upstream node, *i.e.* $\phi_{ip} = \phi_{up}$. The upwind differencing scheme is only first-order accurate but it accounts for the flow direction. The scheme is very robust, but it will introduce diffusive discretization errors that tend to smear steep spatial gradients. These errors are referred as a false diffusion. The false diffusion is most serious when the grid lines are inclined at 45° to the flow direction (Patankar 1980). In high Reynolds number flows, false diffusion can be large enough to produce physically erroneous results (Leschziner 1980, Huang *et al*. 1985). However, the amount of false diffusion can be reduced by refining the mesh size and by aligning the grid lines with the flow direction (Patankar 1980). Although, the upwind differencing scheme is not entirely appropriate for accurate flow calculations, it is very robust and is the best scheme to start the calculations with.

2.3. Pressure-velocity coupling

Transport equations for each velocity component (momentum equations) can be derived from the general transport equation (1) by replacing variable ϕ by *u*, *v* and *w*, respectively. Every velocity component appears in the momentum equations and also must satisfy the continuity equation. Pressure gradients appear in all three momentum equations, thus the pressure field needs to be calculated in order to be able to solve these equations. These equations are hard to solve due to non-linear terms in momentum equations and interdependence of the pressure term in all equations. If the flow field is compressible, the continuity equation may be used as the transport equation for density and the energy equation - for temperature. The pressure may then be obtained from density and temperature. However, if the flow is incompressible, the pressure is independent of density. So there is no explicit equation for pressure. The so-called pressure-velocity coupling algorithms are used to derive equations for the pressure from the momentum equations and continuity equation. A widely used pressure-velocity coupling algorithm is the SIMPLE (Semi Implicit Method for Pressure Linked Equations) algorithm proposed by Patankar and Spalding (1972). It is an iterative procedure for the calculation of pressure and velocity fields. Many CFD books describe the SIMPLE algorithm in detail (Patankar 1980, Ferziger and Peric 2001, Versteeg and Malalasekera 2007 and others), thus it will not be presented here. There are also improved versions of SIMPLE algorithm such as SIMPLER

(SIMPLE Revised) algorithm of Patankar (1980), SIMPLEC (SIMPLE Consistent) algorithm of Van Doormal and Raithby (1984) and PISO (Pressure Implicit solution by Split Operator method) algorithm proposed by Issa (1986) which have been implemented in numerous CFD codes.

2.4. Boundary conditions and near-wall modeling

When solving the Navier-Stokes (or RANSs equations coupled with a turbulence model) and continuity equations, appropriate initial and boundary conditions have to be applied. The process of solving the fluid flow can be seen in a simplified way as the extrapolation of a set of data defined on a boundary contour or surface into the domain interior (Versteeg and Malalasekera 2007). Thus, it is of extreme importance to specify correctly physically realistic boundary conditions, otherwise it will lead to incorrect results. In this subchapter a brief description of the next types of boundary conditions used in the present study will be presented:

- Inflow boundary (inlet)
- Outflow boundary (outlet)
- Periodic boundary conditions (PBC)
- Wall

At inflow boundary the distribution of all flow variables, ϕ, needs to be specified. In CFX the magnitude of the uniform profile of the inlet velocity, *U*, is specified and the direction is taken to be normal to the boundary. The turbulence intensity, $I=\sqrt{(\bar{u}+\bar{v}+\bar{w})/3}\Big/\sqrt{U^2+V^2+W^2}$ is selected to be 5%at the inlet. Thus, turbulence kinetic energy is calculated using:

$$K_{inlet}=\frac{3}{2}I^2U^2 \tag{5}$$

and the turbulence dissipation is calculated using:

$$\varepsilon_{inlet}=\rho C_\mu\frac{k^2}{\mu_t} \tag{6}$$

where $\mu_t=1000I\mu$.

It is very important to place outflow boundary at the appropriate location such that the conditions downstream have no influence on the solution. Thus, outlet should be placed far away from the inlet or any geometric obstacles such that the flow reaches a fully developed state. At the outlet, the gradients of all variables (except pressure) are zero in the flow direction.

3. RESULT & DISCUSSION

3.1. Velocity Profile of Depth Average Velocity

The Velocity profile of depth average velocity is simulated numerically with *k*-ϵ, *k*-ω , SST *K*-ω, RSM and RSM-ω based turbulence models and the FCF Phase A Series 2 data having depth ratio 0.092 have been validated with the results and the plots are shown respectively. The figure shown is non-dimensional and U_d/U^* is plotted against vertical co-ordinates and X/T is plotted against horizontal co-ordinates. Where U_d is depth average velocity, U* is the shear velocity and T is the top width.

$$U^*=\sqrt{\rho gS} \tag{7}$$

where S is the bed slope of the channel.

A practically constant value of U*=0.04080 is obtained using equation 7, while value of T=10 is obtained through channel geometry.

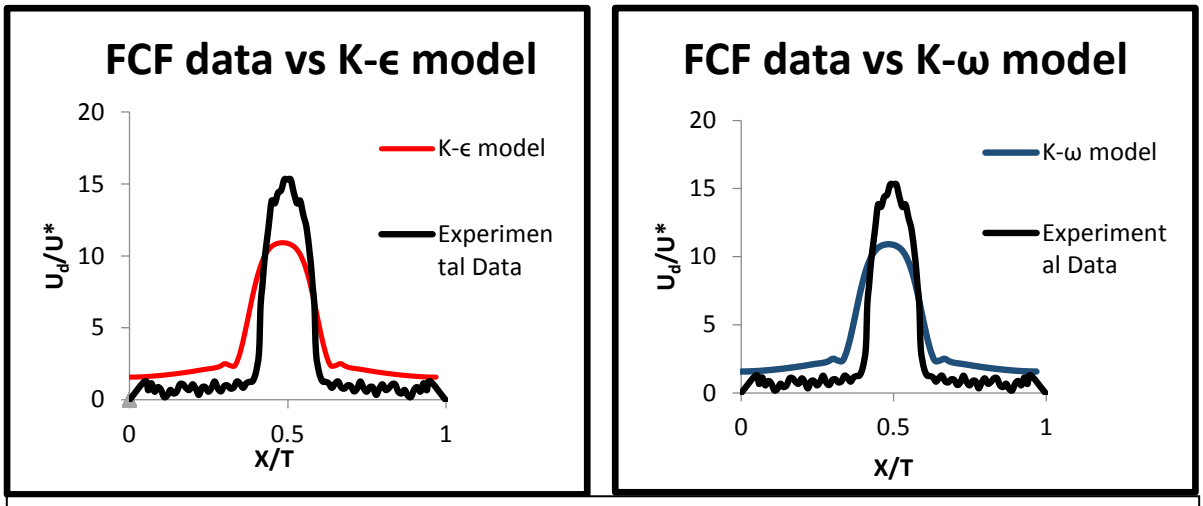

Figure 2. Measured and Simulated (*K-ϵ* and *K-ω* models) of depth average velocity in the main channel and flood plain at cross section Y=40m.

Analyzing the results presented in fig. 2 one can conclude that, both *K-ϵ* and *K-ω* both model give similar results and with good agreement with experimental data mainly in main channel section. While in floodplain both the model gives overestimated result but the similar results are obtained.

The above two equations model (i.e. *K-ϵ* and *K-ω* model) are both performs well but *K-ϵ* gives better estimation in the fully turbulent region while *K-ω* performs better near wall regions.

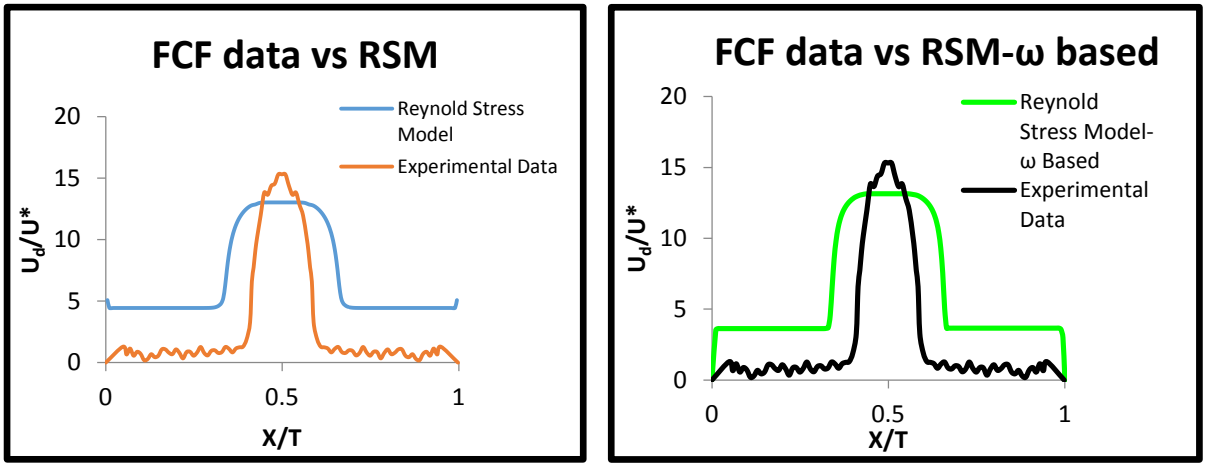

Figure 3. Measured and Simulated (*RSM and* RSM-*ω* based models) of depth average velocity in the main channel and flood plain at cross section Y=40m.

In the main channel both RSM and RSM-*ω* based models (fig. 2) give overestimated over side wall while over the bed it is slightly underestimated. But both the model gives almost similar result and results are closer to the experimental data. Another conclusion is that in the flood plain and in the middle of the main channel the results seems almost independent of the turbulence model and show good agreement with the experimental data, except near the free-surface in both fig. 1 and fig. 2. In total Reynold Stress Models (RSM) are thus more complicated than the eddy viscosity models. They provide a more accurate representation of the turbulence and are valid over a wider range of flows.

3.2 Isovel Lines

The Isovel lines obtained numerically with RSM, RSM-ω based and Shear Stress Transport K-ω turbulence model in cross-section Y=40m are presented in fig. 4.

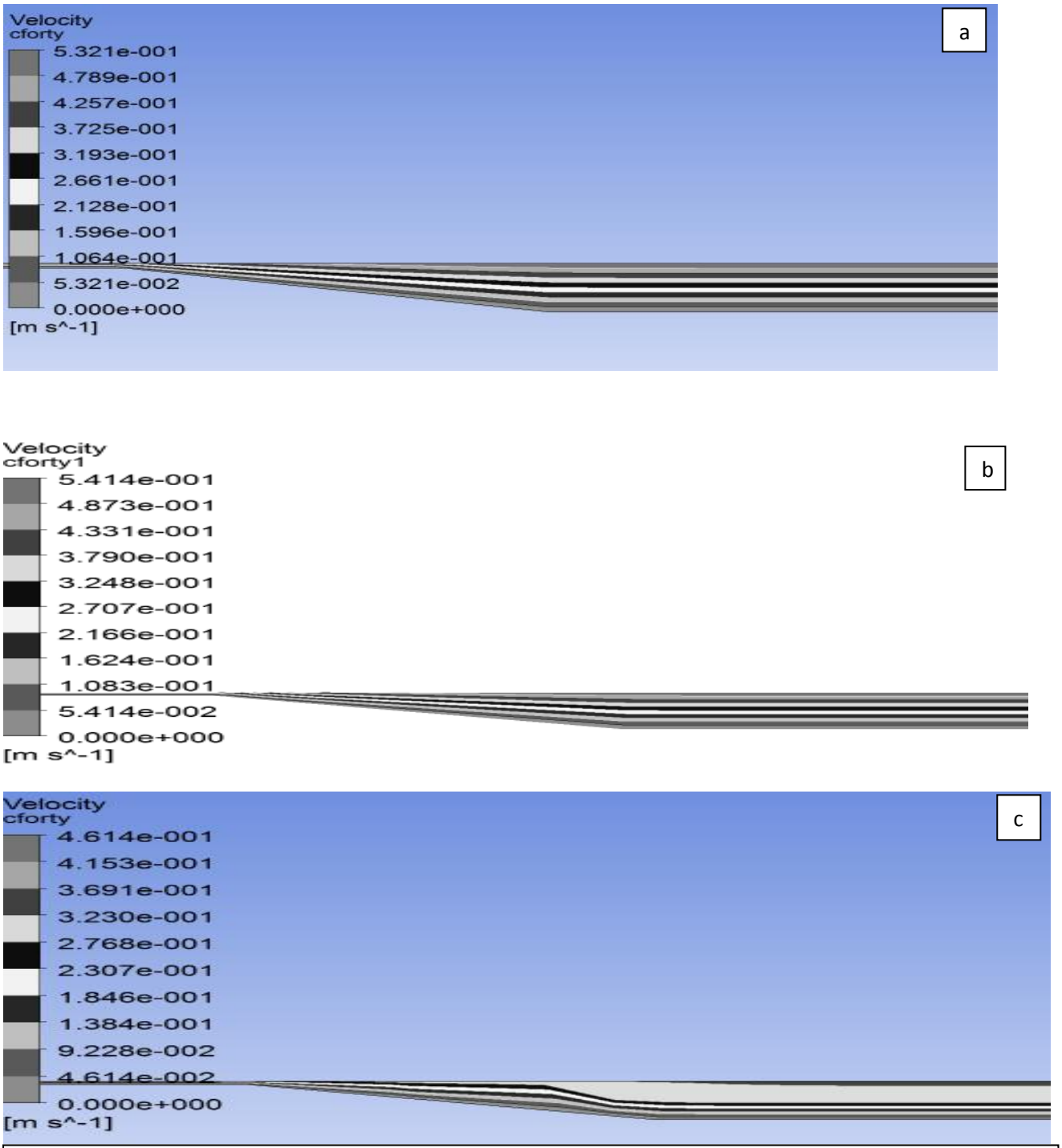

Figure 4. Contour of velocity m/s obtained numerically in cross-section Y=40m with turbulence model: a) RSM; b) RSM-ω based : c) SST K-ω model

The isovel lines of SST K-ω model (fig. 4c) moving significantly upward near upper surface of side bank as a result of finer meshing. Meanwhile, the isovel lines of RSM, RSM-ω based and SST K-ω (Fig. 4a, 4b & 4c respectively) do not bulging upward on the side bank interface because they assume isotropic turbulence and therefore cannot reproduce the secondary flow.

10. CONCLUSIONS

The numerical simulation of the experimental results with different turbulence model which are isotropic allowed to verify that the depth average velocity profile and isovel produced gives acceptable results. Using isotropic models velocity profile predicted in the interface region doesn't show secondary cell generation. Isotropic models underestimate velocities over the bed region while over interfaces (i.e. side bank) it overestimates it. In the main channel K-ϵ and K-ω model gives better result in comparison to other two models and in the floodplain all models give similar results but it is more overestimated in RSM and RSM-ω based. Two equation models (K-ϵ and K-ω) have proven that they perform reasonably well for wide range of flows of engineering interests, with some limitations that they may be accounted with the use of special bounding or damping function. Their major advantage is the simplicity, and the low computational cost compared to more complex models, such as RSM or LES (Large Eddy Simulation).

Notation

Symbol	Description	Unit
$\bar{u}$	Time-averaged streamwise velocity	[m/s]
$\bar{v}$	Time-averaged spanwise velocity	[m/s]
$\bar{w}$	Time-averaged vertical velocity	[m/s]
C	Volume fraction	[-]
g	Gravitational acceleration	[m/s^2]
i	Stands for local values in the ith cell or node	[-]
I	turbulence intensity	[-]
ip	Integration point	[-]
k	Turbulence kinetic energy	[m^2/s^2]
S	Bed slope	[-]
$S\phi$	Source term	[-]
t	Time	[s]
U	Instantaneous streamwise velocity	[m/s]
u^*	Friction velocity	[m/s]
u$'$	Streamwise fluctuation velocity	[m/s]
up	Upwind	[-]
V	Instantaneous spanwise velocity	[m/s]
v$'$	Spanwise fluctuation velocity	[m/s]

W	Instantaneous vertical velocity	[m/s]
$w^{'}$	Vertical fluctuation velocity	[m/s]
x	Cartesian coordinate in the streamwise direction	[m]
y	Cartesian coordinate in the spanwise direction	[m]
z	Cartesian coordinate in the vertical direction	[m]
nj	Discrete outward surface vector	[-]
ρ	Density of the fluid	[kg/m^3]
Γ	Secondary current term	[N/m2]
Γϕ	Diffusion coefficient	[-]

References

1. Chu, A., Fu, C. W., Hanson, A. J., and Heng, P. A. (2009) GL4D: A GPU-based Architecture for Interactive 4D Visualization. *IEEE Transactions On Visualization And Computer Graphics* **15** (6), 1587-1594.
2. Chung, T. J. (2002) *Computational Fluid Dynamics*. Cambridge University Press, N.Y.
3. Conde, J. M. P. (2015). *Numerical modelling of compound channel flow* (Doctoral dissertation, Nazarbayev University, Kazakhstan).
4. Van Doormal, J. P., & Raithby, G. D. (1984). Upstream to elliptic problems involving fluid flow. *Computers and Fluids*, *2*, 191-220.
5. Ferziger, J. H. and Peric, M. (2001) *Computational Methods for Fluid Dynamics*. 2nd ed., Springer-Verlag.
6. Filonovich, M., Azevedo, R., Rojas-Solorzano, L. R., & Leal, J. B. (2010, May). Simulation of the velocity field in compound channel flow using different closure models. In *Proceedings 1 st European IAHR Congress, Edinburgh* (pp. 6-4).
7. Hirt, C. W. and Nichols, B. D. (1981) Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.* **39** (1), 201-225.
8. Huang, P. G., Launder, B. E. and Leschziner, M. A. (1985) Discretisation of non-linear convection processes: a broad-range comparison of four schemes. *Comput. Methods Appl. Mech. Eng*. **48**, 1-24.
9. Issa, R. I. 1986 Solution of the implicitly discretised fluid flow equations by operator-splitting. J.Comput. Phys. 62, 40-65.
10. Knight, D. W. (2013) River hydraulics – a view from midstream. *J. Hydraulic Research* **51** (1), 2-18.
11. Knight, D. W. and Shamseldin, A. (2006) *River basin modelling for flood risk mitigation*. Taylor & Francis, Leiden.
12. Knight, D. W., Tang, X., Sterling, M., Shiono, K. and McGahey, C. (2010) Solving open channel flow problems with a simple lateral distribution model. In River Flow 2010, *Proc. Int. Conf. on Fluvial Hydraulics*, 41-48, A. Dittrich, K. Koll, J. Aberle, P. Geisenhainer, eds. Braunschweig.
13. Knight, D. W., Wright, N. G. and Morvan, H. P. (2005) *Guidelines for applying commercial CFD software to open channel flow*. Report based on the research work conducted under EPSRC Grants GR/R43716/01 and GR/R43723/01.
14. Lane, S. N., Bradbrook, K. F., Richards, K. S., Biron, P. A. and Roy, A. G. (1999) The application of computational fluid dynamics to natural river channels: three-dimensional versus two-dimensional approaches. *Geomorphology* **29** (1), 1-20.
15. Leschziner, M. A. (1980) Practical evaluation of three finite difference schemes for the computation of steady-state recirculating flows. *Comput. Methods Appl. Mech. Eng*. **23**, 293-312.

16. Morvan, H. P. (2005) Channel shape and turbulence issues in flood flow hydraulics. *Journal of Hydraulic Engineering*, **131** (10), 862-865.
17. Morvan, H. P., Pender, G., Wright, N.G. and Ervine, D. A. (2002) Three-dimensional hydrodynamics of meandering compound channels. *Journal of Hydraulic Engineering* **128** (7), 674-682.
18. Morvan, H., Knight, D., Wright, N., Tang, X. and Crossley, A. (2008) The concept of roughness in fluvial hydraulics and its formulation in 1D, 2D and 3D numerical simulation models. *Journal of Hydraulic Research* **46** (2), 191-208.
19. Patankar, S. V. (1980) *Numerical Heat Transfer and Fluid Flow*. Hemisphere Publishing Corporation, Taylor & Francis Group, New York.
20. Patankar, S. V. and Spalding, D. B. (1972) A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. Heat and Mass Transfer* **15** (10), 1787-1806.
21. Richardson, L. F. and Chapman, S. (1965) *Weather prediction by numerical process*. Dover Publications.
22. Versteeg, H. K. and Malalasekera, W. (2007) *An introduction to computational fluid dynamics. The finite volume method*. 2nd Ed., Pearson Pearson Education Ltd., UK.
23. Wright, N. G. (2001) *Conveyance implications for 2D and 3D modelling*. Specialist Review for Scoping Study for Reducing Uncertainty in River Flood Conveyance, R & D Technical Report by HR Wallingford to DEFRA/Environment Agency.