iRIC Software

*Changing River Science*

# *NaysEddy ver 1.0*

# *USER MANUAL*

***By: Mohamed Nabi, Ph.D.***

# Contents

# 1. Introduction

## 1.1. What is NaysEddy?

In river engineering problems, the flow cannot always be approximated by two-dimensional models. The complexity of the flow, especially on the deformed beds and irregular banks, plays an important role in the flow structures, sediment transport and the morphological changes of the rivers. Therefore, simulation of the three-dimensional structures of the flow is essential in prediction of realistic bed shear stresses and can be a suitable estimation for sediment transport and bed morphodynamics.

Several depth integrated two-dimensional models are implemented in iRIC. They demonstrated a great success in the simulation of large scale engineering problems. However, they are not able to capture the three-dimensional structures of the flow even in the presence of complex geometries. They estimate the flow roughly ignoring the vertical flow structures and approximate the flow under the assumption of hydrostatic pressure.  The phenomena such as secondary flow, turbulence eddies, horseshoe vortices, etc cannot be observed under shallow-water estimation.

NaysCube was a solution for these problems, in which the flow can be solved as three-dimensional, using Reynolds Averaged Navier-Stokes (RANS) model. NaysCube, in its turn, is able to solve a wide range of three-dimensional problems. It is based on curvilinear coordinate system with k-epsilon model as a turbulence closure. NaysCube was successful to simulate the rage of three-dimensional problems. However, NaysCube is based on RANS which remove all flow fluctuations, and lead to a smooth flow field. This is not true in reality hence NaysCube fails in simulation of highly complex geometries, in which the turbulence gets significantly affected by the geometry.

Here, we introduce NaysEddy as a three-dimensional solver based on large-eddy simulation (LES). It solves the flow in more details using Cartesian grids with ghost-cell immersed boundary methods. This solver is flexible, accurate and it is able to solve the complex problems with a great success. It is tested under extremely complex conditions, moving boundaries, complex bed topography, and flows with relatively high Froude numbers, etc., and the solver shows its capability in the simulation of those problems. This solver applies the full Navier-Stokes equations without approximation.

However, the computational load of this solver is one order larger than that of Nays-2D. The solver is effective for reproducing local phenomena of rivers for a short time span, rather than reproducing a large section of the river for a long time span.

## 1.2. Features of NaysEddy

NaysEddy is a powerful tool in simulation of tree-dimensional flows. It includes many features, some of them are listed as following:

1. The bed topography can be selected from the existing library. In the case, it is not included in the library, it can be imported by a data file.
2. The grid is in a simple uniform Cartesian framework, which can be generated in a very easy way.
3. NaysEddy simulate the flow accurately as it is based on filtered equation, namely using LES. It gives the detailed structures of the flow.
4. It is Cartesian based solver. Solution for complex bed geometries is incorporated by using ghost-cell immersed boundary method (IBM). The complex geometries can be easily, without significant computational efforts, solved with high accuracy.
5. For IBM, the solver includes two options for interpolations: tri-linear and inverse distance square.
6. The free surface is considered as free-shear rigid-lid. It means the free surface cannot move vertically. This assumption is suitable for constant discharge.
7. The boundary conditions in the streamwise direction can be selected as periodic, or inflow and outflow, related to the nature of the problem.
8. Boundaries in the transverse direction can be selected as periodic or side walls (banks).
9. The top boundary in the vertical direction can be selected as a top wall or free surface.
10. The initial condition can be selected as smooth velocity field or be perturbed by a finite random amplitude.
11. A highly efficient multigrid method is used for the solution of the pressure correction equation. It accelerates the solver significantly. This solver can use V or W type cycles with selection of the number of pre- and post-smoothing iterations.
12. HotStar condition is implemented in the solver. It means the solver can be started from the last point which was stopped (or last saved point). This option is useful to avoid the loss of data by a sudden stopping of the solver.
13. Output date for iRIC visualization, as well as for Tecplot software is implemented. The time interval for the output can be determined.
14. The time step can be calculated automatically based on grid topology and the flow velocity. It can also set to be constant in the case a constant time step is preferred.
15. In order to simulate real cases in the rivers, the bed may need to be perturbed. This solver includes an option to be able to perturb the bed by a finite random amplitude.

# 2. Computational conditions

NaysEddy includes two parts, namely a grid generator and a solver. The grid generator of NaysEddy is essentially different than the other grid generators implemented in iRIC. This grid generator is based on Cartesian coordinate system and sets the grid points with a number of $2^n$ in all directions. This condition in essential for using multigrid solver. Multigrid is known as one of the most efficient technique for solving the elliptic-type equations, such as the Poisson pressure correction equation.

The solver reads the Cartesian grid and apply multigrid technique to solve the pressure equation. The velocity field is solved explicitly using a second-order Adams-Bashforth method. The following sections describe the settings for the grid generator and the solver separately.

## 2.1. Grid Generator

At the first step, generation of a suitable grid has to be accomplished. The generation of grid can be started by selecting [Cartesian Grid for NaysEddy xxx], as shown in Fig. 1. After the grid is selected, the settings for the grid has to be given. This grid generator creates plan grids which will be complemented to 3D later in the solver (It will be explained in section 2.2.1).

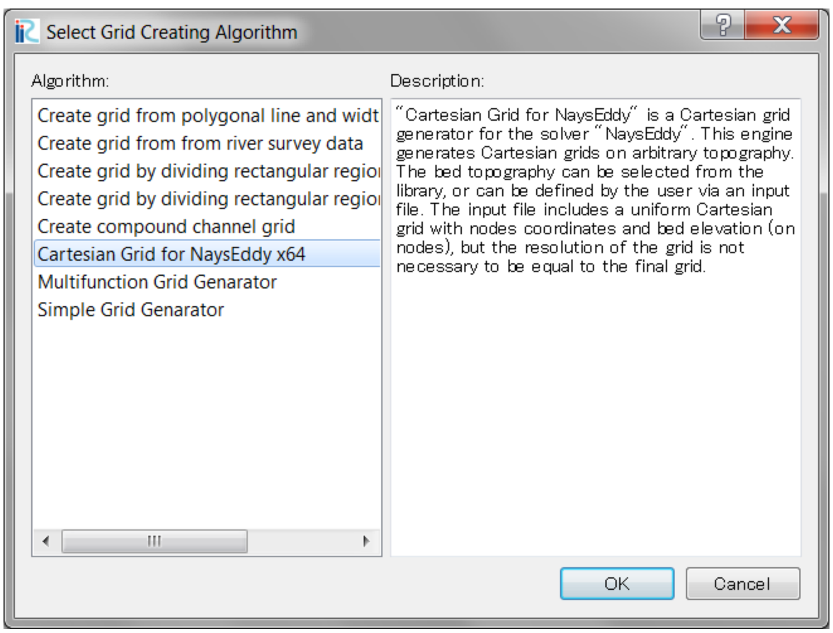



*Figure 1 Selection of the grid generator*

### *2.1.1. Settings for "Grid"*

This option sets the conditions for a Cartesian grid. Figure 2 shows the dialog of the grid generator.

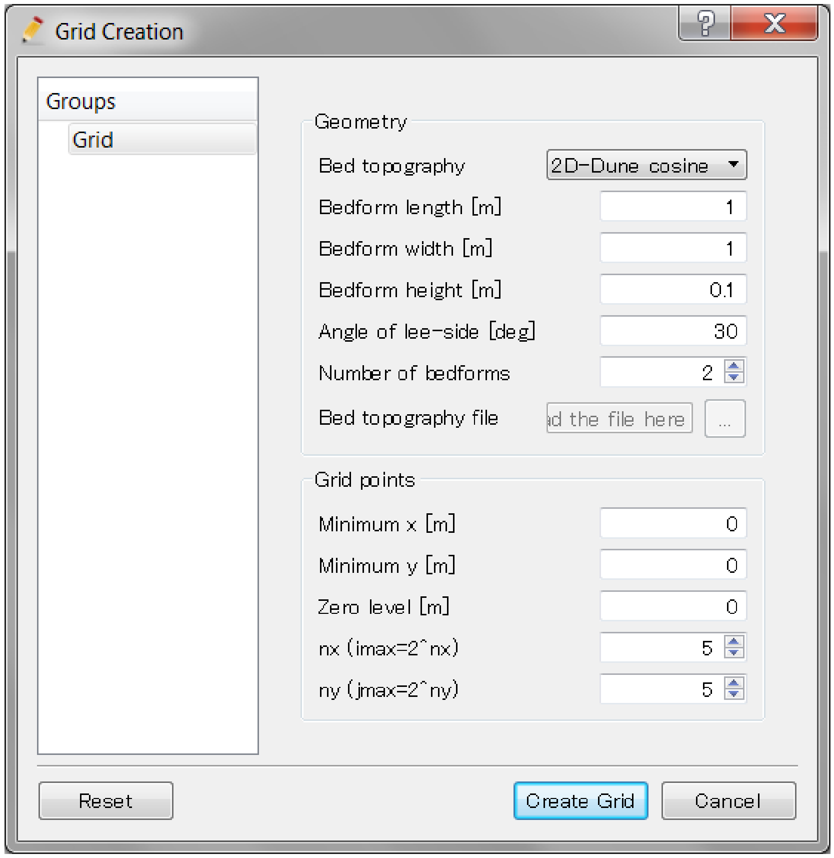



*Figure 2 Dialog of settings of "Grid"*

| # | Item | Setting method | Notes |
|---|---|---|---|
| 1 | Bed topography | Set the topography from existing library, or choose as custom | |
| 2 | Bedform length [m] | The length of bedform | Length of a single bedform (i.e. one dune) |
| 3 | Bedform width [m] | Width of bedform | Width of a single bedform |
| 4 | Bedform height [m] | Height of the bedform | |
| 5 | Angle of lee-side [deg] | Sets the angle of lee-side | In the case of dunes, angle of lee-side in necessary |
| 6 | Number of beforms | The number of sequential bedforms | |
| 7 | Bed topography file | An external file defining the bed topography | In the case, [Custom] is selected, a file has to be read(1) |
| 8 | Minimum x [m] | The minimum x-coordinate of the bed (computational domain) | |
| 9 | Minimum y [m] | The minimum y-coordinate of the bed (computational domain) | |
| 10 | Zero level [m] | Sets the zero level of the bed | The bed will be shifted in such a way the zero level pass from the middle of bedform |
| 11 | nx (imax=2^nx) | Number of grid points in x-direction | The number of grid points is $2^{nx}$. |
| 12 | ny (jmax=2^ny) | Number of grid points in y-direction | The number of grid points is $2^{ny}$. |

(1) The input file can be read from the priory generated grid. This grid in not necessary to have the same resolution as the final grid. The grid generator interpolates the data of the imported grid to find coordinated and the bed elevation on the fine grid. The format of the imported grid is shown as below. *I* and *J* are the maximum grid points in *x*- and *y*-directions, associated to the imported grid, respectively. The imported grid has to be uniform (with constant $\Delta x$ and $\Delta y$). The indices *I* and *J* can be minimally 2. There is no limitation for the maximum values of *I* and *J*. The bounds of the final grid will be a rectangle from $(x_1,y_1)$ to $(x_I,y_J)$.

| | | |
|---|---|---|
| $I$ | $J$ | |
| $x_1$ | $y_1$ | $z_{1,1}$ |
| $x_2$ | $y_1$ | $z_{2,1}$ |
| ⋮ | ⋮ | ⋮ |
| $x_I$ | $y_1$ | $z_{I,1}$ |
| $x_1$ | $y_2$ | $z_{1,2}$ |
| $x_2$ | $y_2$ | $z_{2,2}$ |
| ⋮ | ⋮ | ⋮ |
| $x_I$ | $y_2$ | $z_{I,2}$ |
| ⋮ | ⋮ | ⋮ |
| ⋮ | ⋮ | ⋮ |
| $x_1$ | $y_J$ | $z_{1,J}$ |
| $x_2$ | $y_J$ | $z_{2,J}$ |
| ⋮ | ⋮ | ⋮ |
| $x_I$ | $y_J$ | $z_{I,J}$ |

## 2.2. Solver

After the grid is generated successfully, the solver can be started. The solver can be started by selecting [NaysEddy v.1.0 xxx] from the [Select Solver] menu. It is illustrated in Fig. 3.

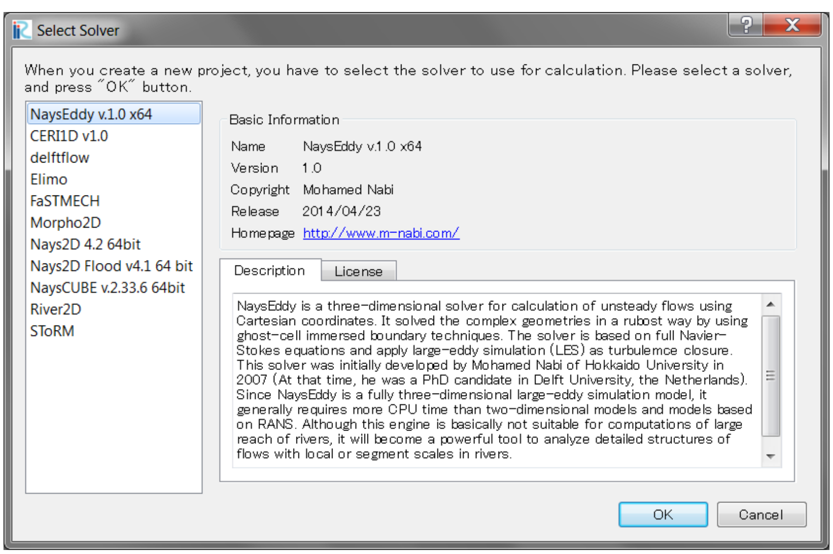


*Figure 3 Selection of the solver*

### 2.2.1. Settings for "Grid"

The grid generator of iRIC creates plane grids. The vertical direction of the grid is still not defined and hence, a definition for the grid in vertical direction is still necessary. This setting gives the possibility to define the grid in the vertical direction.

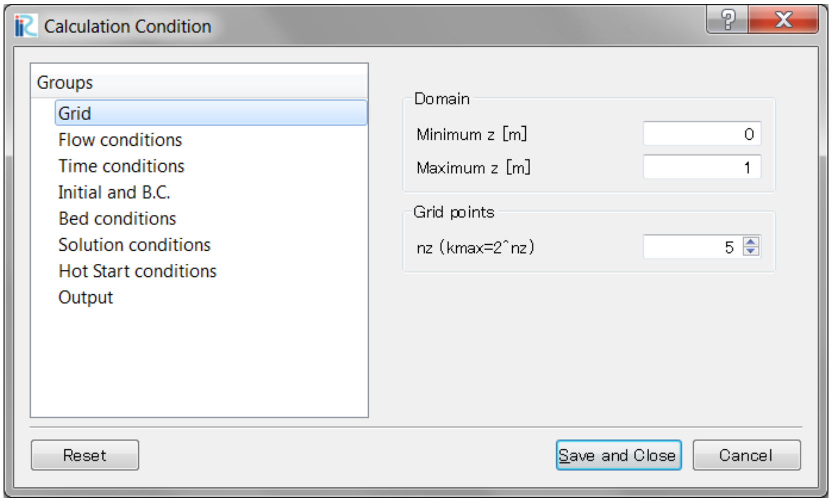



*Figure 4 Dialog settings for "Grid" in the solver settings*

| # | Item | Setting method | Notes |
|---|---|---|---|
| 1 | Minimum z | Set the minimum coordinate of computational domain in vertical direction | It must be smaller than the lowest point on the bed |
| 2 | Maximum z | Set the maximum coordinate of computational domain in vertical direction | Water surface (or wall) level. Must be greater than the highest point on the bed |
| 3 | nz (kmax=2^nz) | Set the number of grid cells in the vertical direction | The number of grid cells are a power of 2. nz is the power (note 1) |

The solver applies multigrid method for the solution of the pressure correction equation. The multigrid requires a number of cells in the form of $2^n$ to be able to coarsen the grid efficiently. To restrict the user on this condition, a power in the form of *nz* can be given. The solver calculates the number of the cells in the z-direction by applying $2^{nz}$.

### 2.2.2. Settings for "Flow conditions"

The physical parameters of the flow conditions can be set by selecting this option.

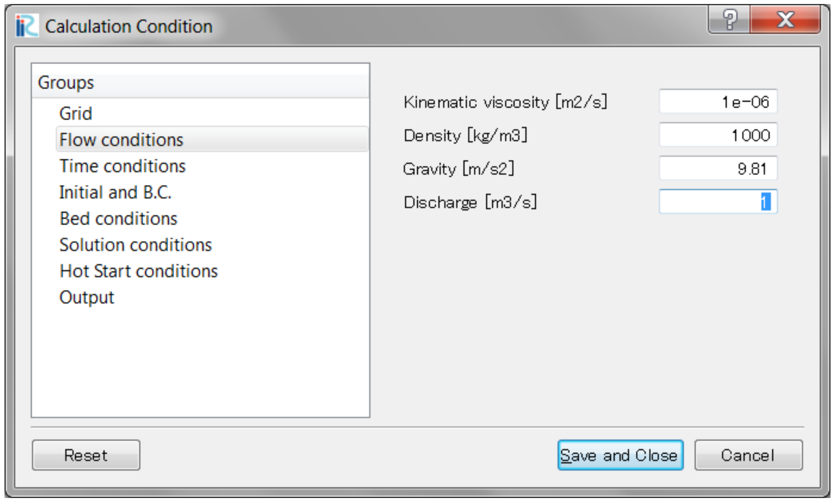



*Figure 5 Dialog settings for "Flow conditions"*

| # | Item | Setting method | Notes |
|---|---|---|---|
| 1 | Kinematic viscosity [m2/s] | Set the kinematic viscosity of flow | The default value for the water is 1e-6 |
| 2 | Density [kg/m3] | Set the mass density of the flow | The default value for the water is 1000 |
| 3 | Gravity [m/s2] | Set the gravity acceleration | The default value is 9.81 |
| 4 | Discharge [m3/s] | Set the upstream discharge | The current version of NaysEddy, supports only constant discharge |

### 2.2.3. Settings for "Time conditions"

The convergence of each solver is strongly dependent on the time step. Not only the convergence, but also the accuracy can be affected by the selection of the time step. This option describe the time-related options for the solver.

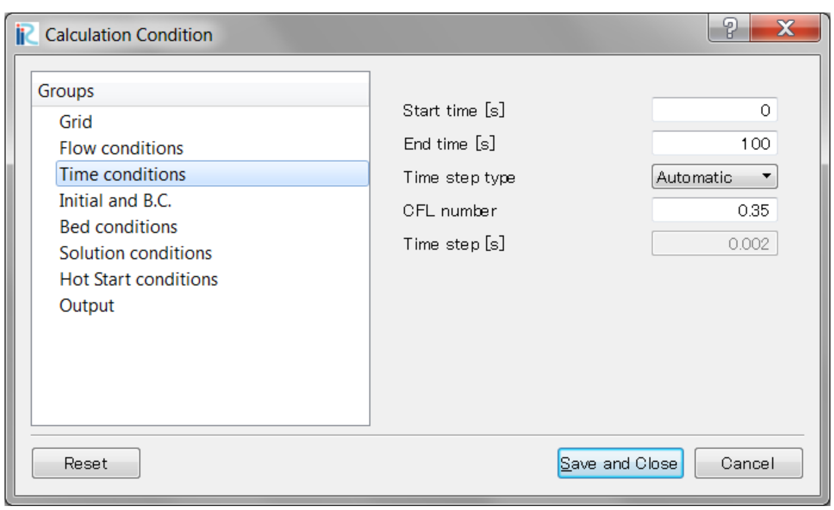



*Figure 6 Dialog settings for "Time conditions"*

| # | Item | Setting method | Notes |
|---|---|---|---|
| 1 | Start time [s] | Set the start time of the computation | The start time has no effect on the results. |
| 2 | End time [s] | Set the end time of the computation | The end time is a criterion to stop the calculation |
| 3 | Time step type | Choose among "Automatic" and "Constant" time step | The time step can be calculated (variable) or selected as constant |
| 4 | CFL number | CFL restriction for "Automatic" | The CFL number |
| 5 | Time step [s] | Set the time step if constant | Time step when the option "Constant" is selected. |

The time step can be automatically calculated based on the grid spacing and the instantaneous velocity. In this case, the time step is calculated as

$$\Delta t \le \frac{\beta}{\frac{|U|}{\Delta x}+\frac{|V|}{\Delta y}+\frac{|W|}{\Delta z}+\nu\left(\frac{1}{\Delta x^2}+\frac{1}{\Delta y^2}+\frac{1}{\Delta z^2}\right)} \qquad (1)$$

where, *U*, *V* and *W* are maximum flow velocity in the *x*-, *y*- and *z*-directions, respectively. The parameter β is a safety factor which is set to 0.35 here. Smaller value of β leads to more stability but slows down the solution. Large beta increase the speed of solution but it leads to instability. We found the value of 0.35 is a suitable choice.

### 2.2.4. Settings for "Initial and boundary conditions"

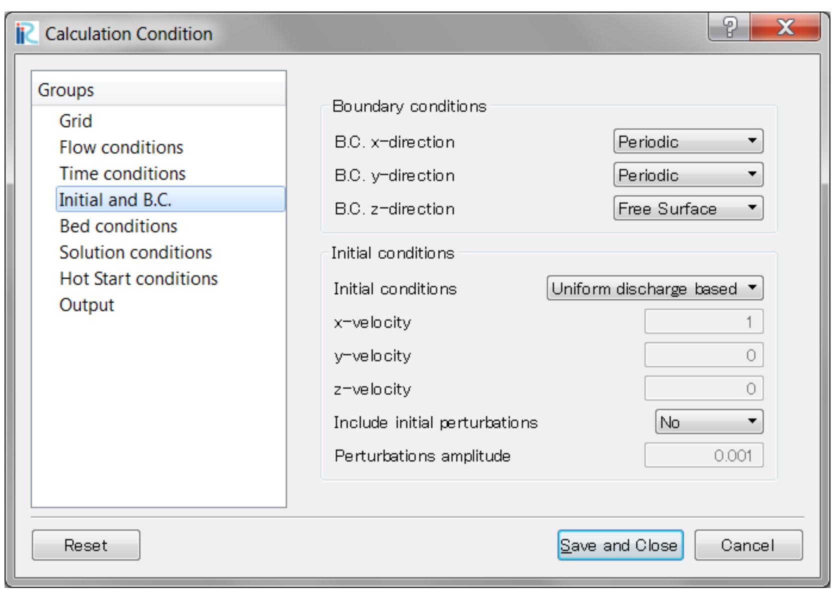



*Figure 7 Dialog settings for "Initial and boundary conditions"*

| # | Item | Setting method | Notes |
|---|---|---|---|
| 1 | B.C. x-direction | The boundary conditions in streamwise direction | Choose among “periodic” and “In and out-flow” |
| 2 | B.C. y-direction | The boundary conditions in transverse direction | Choose among “periodic” and “Side walls” |
| 3 | B.C. z-direction | The boundary conditions in vertical direction | Choose among “Free surface” and “Wall” |
| 4 | Initial conditions | The type for initial condition | Choose among “Uniform discharge based” and “Uniform custom” |
| 5 | x-velocity | Initial velocity in streamwise direction | In the case of “Uniform custom” for Initial conditions, this sets the initial velocity in x-direction |
| 6 | y-velocity | Initial velocity in transverse direction | In the case of “Uniform custom” for Initial conditions, this sets the initial velocity in y-direction |
| 7 | z-velocity | Initial velocity in vertical direction | In the case of “Uniform custom” for Initial conditions, this sets the initial velocity in z-direction |
| 8 | Include initial perturbations | Select to all initial perturbation for the velocity field | Select “Yes” if need to add initial perturbations |
| 9 | Perturbations amplitude | The amplitude of the initial perturbation | Set a value for the magnitude of perturbations |

The initial condition for the velocity can be perturbed in order to accelerate the solution to reach to turbulent regime. The velocities can be perturbed randomly by a small amplitude.

### *2.2.5. Settings for “Bed conditions”*

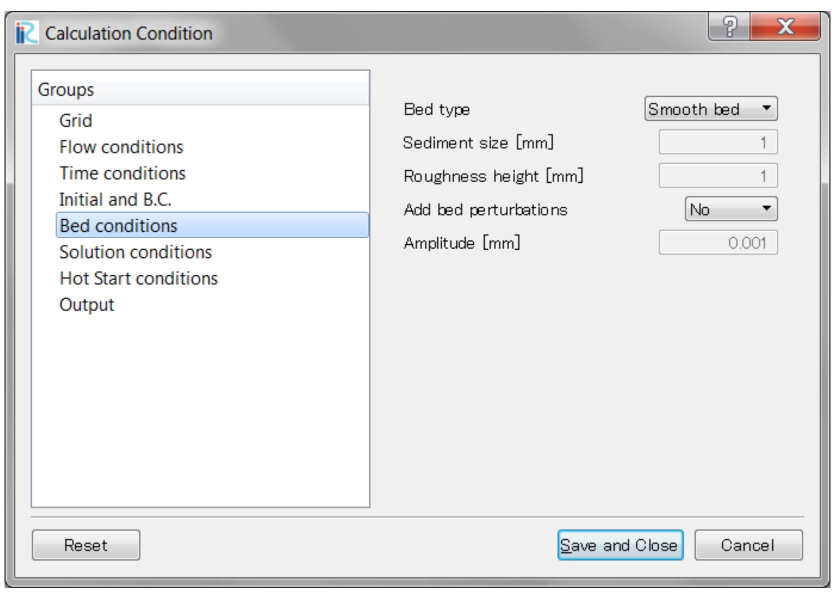



*Figure 8 Dialog settings for “Bed conditions”*

| # | Item | Setting method | Notes |
|---|---|---|---|
| 1 | Bed type | Set the bed type, either smooth or rough | Choose among “Smooth bed”, “Sediment bed” and “Rough bed” |
| 2 | Sediment size [mm] | The mean diameter of sediment on the bed | In the case of “Sediment bed” the sediment diameter has to be given |
| 3 | Roughness height [mm] | Set the height of roughness | In the case of “Rough bed” the height of roughness has to be set |
| 4 | Add bed perturbations | Option to set bed perturbation | Select “Yes” if initial bed perturbation is necessary |
| 5 | Amplitude [mm] | Set the amplitude of bed perturbations | In the case of “Yes” for bed perturbation, the magnitude of the perturbation height has to be given |

The bed of a real river is never smooth. This option adds a random perturbation to the bed. Adding perturbation to the bed increases the turbulence intensity significantly, especially close to the bed. Hence it has a considerable effect on the bed shear stress.

### *2.2.6. Settings for “Solution conditions”*

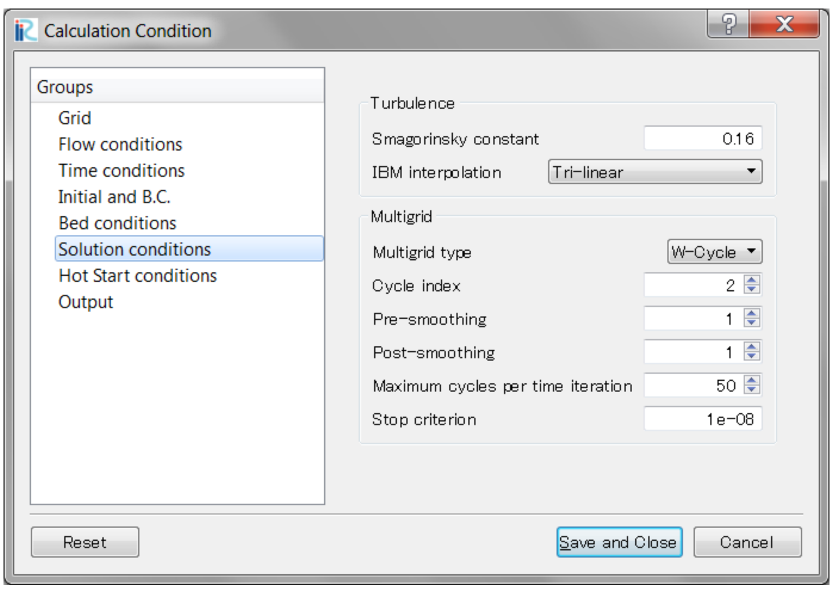



*Figure 9 Dialog settings for “Solution conditions”*

| # | Item | Setting method | Notes |
|---|---|---|---|
| 1 | Smagorinsky constant | Set the “Smagorinsky constant” for LES | The default value is 0.16 |
| 2 | IBM interpolation | Set the type of interpolation for the immersed boundary method | Select among “Tri-linear” and “Inverse distance square” |
| 3 | Multigrid type | The type of multigrid cycles | Choose among “W-cycle” and “V-cycle” |
| 4 | Cycle index | Set the cycle index | In the case of “W-cycle” the cycle index can be selected |

| | | | |
|---|---|---|---|
| 5 | Pre-smoothing | Set the number of pre-smoothing iterations | The number of smoothing times on the finer grids |
| 6 | Post-smoothing | Set the number of post-smoothing iterations | The number of smoothing times on the coarser grids |
| 7 | Maximum cycles per time iteration | Set the maximum allowed cycles per time iteration | To avoid unlimited looping in the case of non-converging |
| 8 | Stop criterion | Set the error criteria for pressure correction | A small number indicating the residual where the pressure has to stop |

In the immersed boundary method (IBM), the interpolation can be of different types. It is found tri-linear interpolation leads to a second-order accuracy. However, the tri-linear interpolation can leads to inaccuracy in extremely complex geometries (e.g. geometries with high frequency ripples), because an arranged set of flow points (to obtain tri-linear interpolation) can be in a relatively large distance from the bed. To avoid this deficiency, “inverse distance square” interpolation can be a suitable choice. It is proven that this interpolation also leads to a second order accuracy. This method does not need an arranged set of flow points, and the average is done any set of points.

### 2.2.7. Settings for “Hot start”

Hot-start is an option in which the computation does not start from the initial condition. The computation starts from a previously saved results.

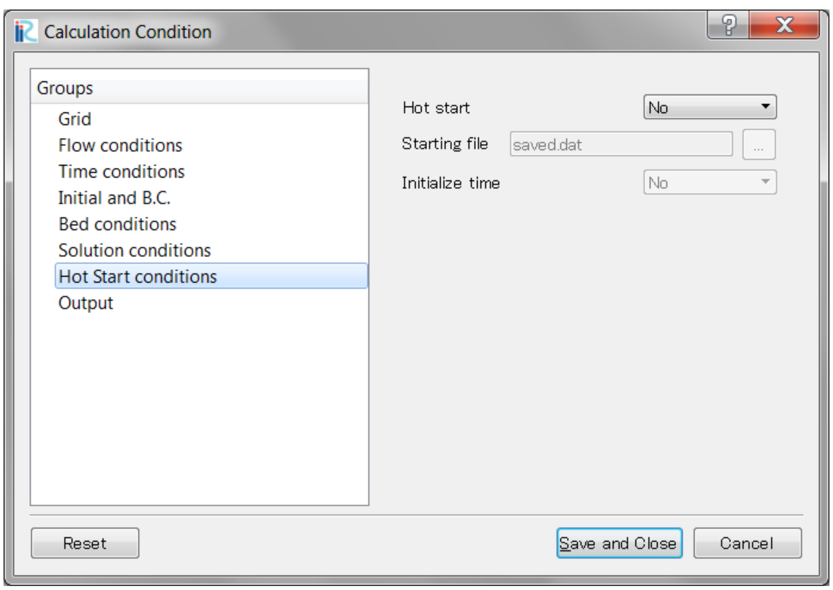



*Figure 10 Dialog settings for “Hot Start conditions”*

| # | Item | Setting method | Notes |
|---|---|---|---|
| 1 | Hot start | Set output for hot-start condition | If “Yes” is selected, the solution starts from the point defined by the file |
| 2 | Starting file | Reads the file for hot-start | Read the file from the PC for the start of the solution |

| | | | |
|---|---|---|---|
| 3 | Initialize time | Reset the time | If "Yes", the time will be reset to zero. |

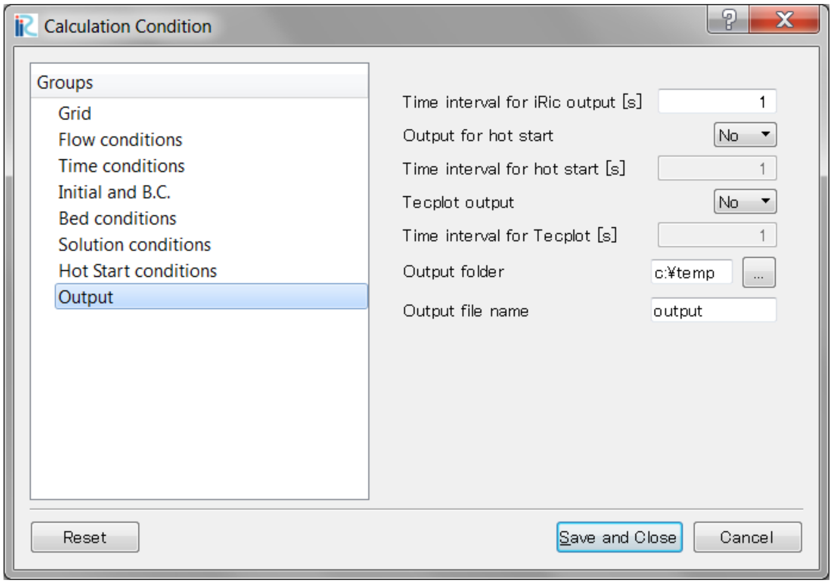



*Figure 11 Dialog settings for "Output"*

| # | Item | Setting method | Notes |
|---|---|---|---|
| 1 | Time interval for iRIC output [s] | Set the time interval for output of CGNS file | Small time leads to a large file and more output results, large time leads to less output results and smaller file |
| 2 | Output for hot start | Option for hot-start output | Select "Yes" to give output for hot-start file |
| 3 | Time interval for hot start [s] | Set the time interval for hot-start output | The time step for hot-start output file. |
| 4 | Tecplot output | Option for Tecplot output | Set to "Yes" is need Tecplot output. |
| 5 | Time interval for Tecplot [s] | Set the time interval for Tecplot output | Small time needs large space on harddisk but gives more output results |
| 6 | Output folder | Select the folder for output files | Select a folder in which the output will be saved |
| 7 | Output file name | Defines the file name | Select the file name for the output of hot-start and Tecplot |

Tecplot output can be used as a dual purpose. It can, not only be used for visualization, but also be used for averaging the flow in time (and/or in space). Although this is also possible to do by reading the data from the CGNS file, but reading the Tecplot files can be considerably easier.

# 3. Governing equations

## 3.1 Flow

The governing equations for the fluid are the full three-dimensional, unsteady, incompressible Navier-Stokes equations written in terms of primitive variables. These equations are given below in terms of volume filtered variables.

$$\frac{\partial \bar{u}_j}{\partial x_j} = 0 \tag{2}$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_j} = -\frac{1}{\rho}\frac{\partial \bar{P}}{\partial x_i} + \frac{\partial}{\partial x_j}\left\{2\left(\nu + \nu_t\right) S_{ij}\right\} \tag{3}$$

where $x_i$'s are the coordinates, $t$ is the time, $P$ the modified pressure, $\rho$ the density, $u_i$ the filtered velocity component in $x_i$-direction, $\nu$ and $\nu_t$ the molecular and turbulent viscosities and $S_{ij}$ is the resolved strain rate tensor:

$$S_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) \tag{4}$$

In large eddy simulation, the large eddies are solved directly, ignoring the smaller eddies. The smaller eddies are then modelled separately. LES uses volume filtering, allowing to filter eddies which are smaller than the grid cell volume. The filter is defined as

$$\mathbf{f}\left(\mathbf{x},t\right) = \int f\left(\mathbf{y},t\right) G_{\Delta}\left(\mathbf{x}-\mathbf{y}\right) d\mathbf{y} \tag{5}$$

The effect of the small scales upon the resolved part of turbulence appears in the sub-grid-scale (SGS) stress term

$$\tau_{ij} = \overline{u_i u_j} - \bar{u}_i \bar{u}_j \tag{6}$$

which must be modelled. It is set proportional to the strain-rate tensor $S_{ij}$, characterizing the local deformation of the resolved velocity field. The model is written as

$$\tau_{ij} = -2\nu_T S_{ij} + \frac{1}{3}\tau_{kk}\delta_{ij} \tag{7}$$

And finally, the eddy viscosity is determined as

$$\nu_t = \left(C_S \bar{\Delta}\right)^2 \left|\bar{S}\right| \tag{8}$$

Where

$$\left|\bar{S}\right| = \sqrt{2\bar{S}_{ij}\bar{S}_{ij}} \tag{9}$$

$$\bar{\Delta} = \left(\Delta x.\Delta y.\Delta z\right)^{1/3} \tag{10}$$

The value of $C_S$ is constant in Smagorinsky's model. The value of $C_S$ can be determined using the turbulence spectrum. Its value is approximately 0.17 in isotropic turbulent flow (Lilly, 1967).

Smagorinsky's model is too dissipative in the presence of walls; moreover, it does not work in particular for the transition in a boundary-layer developing on a flat plate. It artificially relaminarizes the flow if the upstream perturbation is not high enough. This is due to the heavy influence in the eddy viscosity of the velocity gradient in the direction normal to the wall and to an improper behavior of the model at the wall. Smagorinsky's model ignores the backscatter of energy, and dissipates all the energy which is transferred from the large to the small eddies.

To avoid excessive dissipation, the effect of SGS can be modelled using log-law-wall-function model. This approach interpolates the velocity close to the solid boundaries by a logarithmic function. The essence of this approach is the determination of the shear velocity $u_*$ by a logarithmic relation from the near-wall flow region and the subsequent imposition of a shear stress $u_*^2$ as a surface force in the momentum equations for the grid cells adjacent to the wall. For a hydraulically rough solid wall, the log-law has the following form

$$\frac{u}{u_*} = \frac{1}{\kappa}\ln\frac{z}{z_0} \tag{11}$$

in which $u$ is the velocity parallel to the wall at a distance of $z$ from the wall and $\kappa$ is the von Kàrmàn constant. The value of $z_0$ is defined as $z_0 = k_s/30$ where $k_s$ represents the roughness height. The diameter of the sand particles is related to the roughness height $k_s$ as 2.5 times the sand particles diameter (Van Rijn, 1984c).

For hydraulically smooth solid walls, the standard log-law wall model with viscous sub-layer ($z^+ \leq 5$), the buffer layer ($5 < z^+ < 30$) and the logarithmic layer ($z^+ \geq 30$) is given by

$$u^+ = \begin{cases} z^+ & z^+ \leq 5 \\ 5.0\ln z^+ - 3.05 & 5 < z^+ < 30 \\ 2.5\ln z^+ + 5.20 & z^+ \geq 30 \end{cases} \tag{12}$$

where $u^+$ and $z^+$ represent the non-dimensional parallel velocity to the wall and the non-dimensional normal distance to the wall respectively.

## 3.2. Immersed boundaries

An approach which has been gaining popularity in recent years is the Cartesian grid method. In this method the governing equations are discretized on a Cartesian grid which cannot fit the immersed boundaries. Cut-cell techniques and ghost-cell methods are the most popular remedies to this problem. In cut-cell techniques, the intersecting cells are cut, yielding arbitrarily shaped cells, which add complexity to the computational model. The ghost-cell methods force the fluid at the immersed boundaries. It is easy to implement and requires less computational effort than the cut-cells techniques. A ghost-cell method is implemented in this solver which is based on the application of direct forcing on the immersed boundaries. The approach is similar to the forcing used by Mohd-Yusof (1997) and Fadlun et al. (2000). The force is a function of time and space, and is defined in such way that the desired boundary conditions are satisfied at every time step.

The locations of this force are not coincident with the grid points; hence it must be extrapolated to these nodes. Many different interpolation/extrapolation techniques are adopted to determine the forcing term on the ghost cells. The solver uses two kinds of interpolations: (1) tri-linear interpolation, and (2) inverse distance-square interpolation.

### *3.2.1. Tri-linear interpolation*

In the present study, the former category is used. This means, the ghost cells are located in the fluid in the vicinity of the boundary. They are defined as cells that have at least one neighbor cell located in the solid. To classify the nodes, an integer flag is used: nodes with flag –1 are located in the solid, nodes with flag 1 in the fluid. Ghost nodes have flag zero. There are several ways to interpolate the velocities for the ghost cells to satisfy the boundary conditions. In the present work, the ghost cell is located on the bed normal vector passing through it, and it forms the imaginary point I as shown in Figure 12. The imaginary point is extended into the fluid to find the required nodes (in fluid) for a tri-linear (bi-linear in 2-D) interpolation. In the case, some of required nodes still fall in the solid, the extension can be continued. Later on, the logarithmic interpolation of equation 12 between point *I* and the bed is applied to interpolate the values for the ghost points.

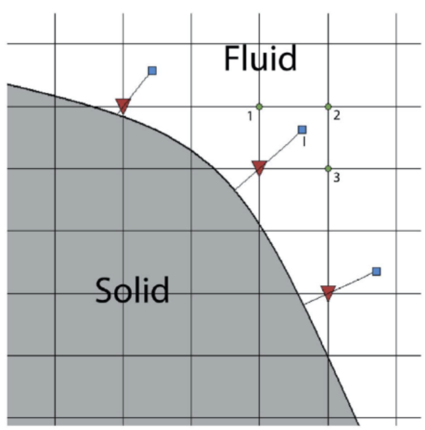



*Figure 12 Schemtic view of interpolation scheme for ghost-cell IBM method.*

### 3.2.2. Inverse distance-square interpolation

Furthermore, more elaborate, high-order schemes may be used in three-dimensions. It is well known that high-order polynomial interpolations may introduce wiggles and spurious extrema. The inverse distance weighting proposed by Franke [13] has the property of preserving local maxima and producing smooth reconstruction. This scheme is suitable for reconstructing variables that are smoothly varying without exhibiting large maxima. The interpolation at the ghost cell is

$$f_G = \sum_{m=1}^{n} w_m f_m / q \tag{13}$$

$$w_m = \left( \frac{R - h_m}{R h_m} \right)^p \tag{14}$$

$$q = \sum_{l=1}^{n} \left( \frac{R - h_l}{R h_l} \right)^p \tag{15}$$

where $f_m$ ($f_G$) represents the solution at a certain location (ghost cell), $w_m$ represents the weight and $h_m$ is the distance between the ghost cell ($f_G$) and the location of $f_m$. $p$ is an arbitrary positive real number called the power parameter (typically $p$ = 2). $R$ is the distance from the ghost point to the most distant point used in the construction and $n$ is the total number of the construction points.

It is important to note that for the forcing of Saiki and Biringen [38] and Goldstein et al. [15], the velocity at the immersed boundaries was imposed by the fictitious force. In the current approach, the boundary condition is imposed directly. This implies that, in contrast to the feedback forcing method, the stability limit of the current integration scheme is the same as that without the immersed boundaries, thus making simulation of complex three-dimensional flows practical. Higher-order extrapolation/interpolation schemes to evaluate the variables at the ghost cells can preserve at least second-order spatial accuracy.

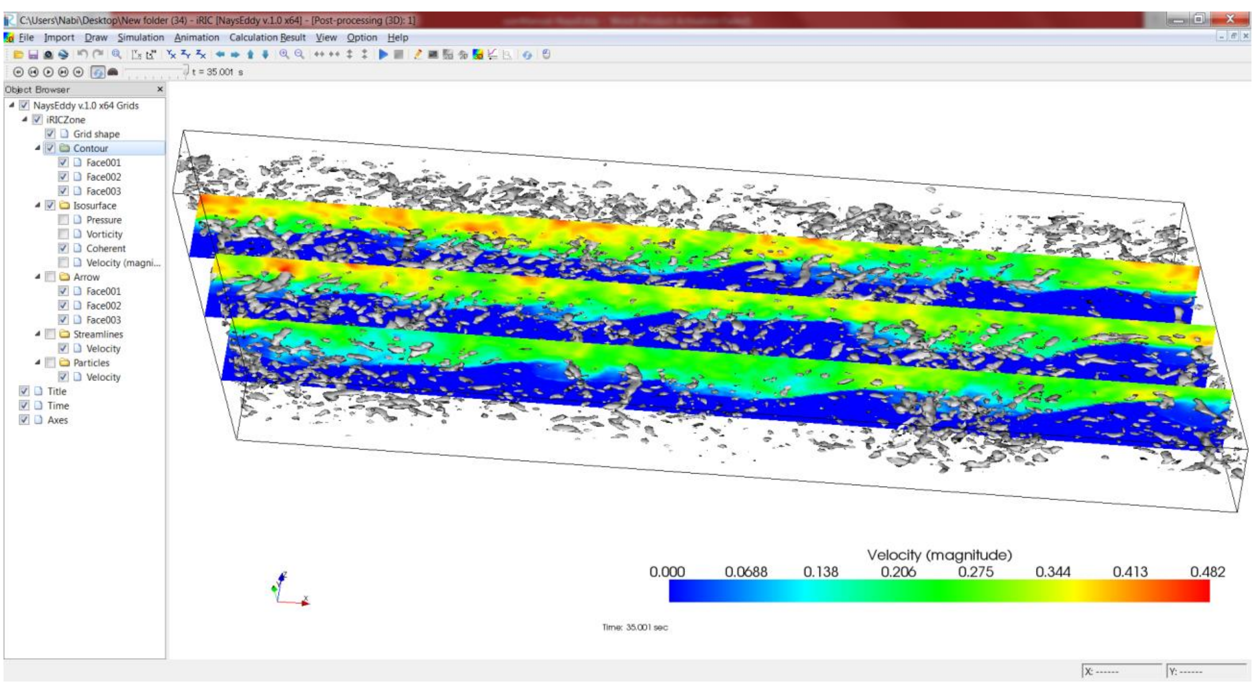
# 4. Visualization of Results

iRIC includes a post-processing engine to be able to visualize the results in an easy way without the need to third party visualization software. This postprocessor is easy to use and available for 2D and 3D visualizations.

As the computational results of NaysEddy are three-dimensional in their nature, the three-dimensional postprocessor of iRIC can be efficiently used to visualize the results. NaysEddy deploys various output parameters. By visualizing such parameters, three-dimensional computational outputs can be shown in an easy way to understand the output.

You need to open a 3D Post-Processing Window for 3D visualization. For details of the use of this window, please refer to the iRIC User's Manual or Examples Manual. Here, explanations are given about output parameters necessary for drawing a color contour or an isosurface. NaysEddy has 4 output parameters. Figure 13 is an example showing the 3D post-processing window. It includes the velocity contour and the turbulence eddies above a bed topography.

Under [Isosurface] in the [Object Browser], you can find 4 items. These items are the output parameters of NaysEddy into iRIC. In the next section, we explain these parameters in more details.



## 4.1. Output parameters in iRIC

The table below, gives the output parameters of NaysEddy into iRIC.

| # | Item | Meaning | Description |
|---|---|---|---|
| 1 | Pressure | Total water pressure (Pa) | It is given as the total pressure (static and dynamic) of the water. |
| 2 | Vorticity | Vorticity magnitude ($s^{-1}$) | It outputs the vorticity magnitude. It is suitable for clarifying the entire vortex structure. |
| 3 | Coherent | Weis function ($s^{-1}$) | It outputs the value given by the Weiss function. It is good for visualizing the turbulent coherent structures [(1)]. |
| 4 | Velocity (magnitude) | Velocity magnitude (m/s) | Set to "Yes" is need Tecplot output. |

[(1)] In order to visualize the coherent structures of the flow we follow the approach developed by Hunt et al. (1988), based on the second invariant of $\nabla u$, defined as

$$Q = -\frac{1}{2}\frac{\partial u_i}{\partial x_j}\frac{\partial u_j}{\partial x_i} = -\frac{1}{2}\left(|S|^2 - |\Omega|^2\right) \tag{16}$$

where $S$ and $\Omega$ are the symmetric and antisymmetric components of $\nabla u$. Components $S$ and $\Omega$ are defined as

$$S = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{17}$$

$$\Omega = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) \tag{18}$$

## 4.2. Output for Tecplot

NaysEddy has the feature to output the results in Tecplot format (PLT format). Tecplot is a commercial software for data visualization. In order to have output results for Tecplot, you need to choose the related option shown in Fig. 11.

The output file of Tecplot can be used not only for visualization purpose, but also for data collection. Large-eddy simulation is, unlike RANS models, solve the turbulence accurately without ensemble averaging. Hence, the results have to be collected and averaged in time (and in space, in cases with symmetric topographies) to obtain the averaged parameters such as mean velocity, turbulent kinetic energy, Reynolds shear stresses, mean vorticity, etc. It has to be noted that collection of data in time take a considerable space on the hard drive.

To take the average values of parameters, you need to develop a small subroutine. This subroutine reads the data from the PLT files and average it in time (and/or space). For instance, the averaged velocity is

$$\overline{u}_{i,j,k} = \frac{1}{T}\sum_{n=0}^{T-1} u_{i,j,k}^{n} \tag{19}$$

After the velocity is averaged, the data files need to be read again to find the Reynolds shear stresses.

$$\overline{u'u'}_{i,j,k} = \frac{1}{T}\sum_{n=0}^{T-1}\left(u_{i,j,k}^{n} - \overline{u}_{i,j,k}\right)\left(u_{i,j,k}^{n} - \overline{u}_{i,j,k}\right) \tag{20}$$

$$\overline{u'v'}_{i,j,k} = \frac{1}{T}\sum_{n=0}^{T-1}\left(u_{i,j,k}^{n} - \overline{u}_{i,j,k}\right)\left(v_{i,j,k}^{n} - \overline{v}_{i,j,k}\right) \tag{21}$$

The other components for the Reynolds shear stress tensor can be found in a similar way shown in Equations 20 and 21.

## 4.3. Output for ParaView

ParaView is another visualization software. It is powerful and free of charge. You can download this software from

http://www.paraview.org/

However, NaysEddy has no direct option for exporting the results in ParaView format (VTK format). It can be done by using the related option in the iRIC interface. It can be found in the “Calculation Result” of the menu bar, and then by selection of “Export”.

# 5. References


Fadlun, E. A., Verzicco, R., Orlandi, P., and Mohd-Yusof, J. (2000). Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics*, 161:35–60.

Hunt, J. C. R.,Wray, A. A., and Moin, P. (1988). Eddies, streams and convergence zones in turbulent flows. *CTR Annual Research Briefs NASA Ames*, Stanford University.

Mohd-Yusof, J. (1997). Combined immersed boundaries/B-splines methods for simulations of flows in complex geometries. *CTR Annual Research Briefs, NASA Ames*, Stanford University, Stanford.