ITTC Quality System Manual

Recommended Procedures and Guidelines

Guideline

Practical Guidelines for Ship CFD Applications

7.5	Process Control
7.5-03	CFD
7.5-03-02	Resistance and Flow
7.5-03-02-03	Practical Guidelines for Ship CFD Applications

Edited /Updated	Approved
Quality Systems Group of the 28th ITTC	27th ITTC 2014
Date 06/2017	Date 09/2014

Table of Contents

Practical Guidelines for Ship CFD Applications

1. INTRODUCTION

This guidelines describe recommendations and common practices for ship CFD applications in a general way. Separate specific guidelines cover the CFD process for other applications in detail.

In this guideline the CFD process is divided into three steps: pre-processing, computation, and post-processing. The pre-processing step involves the definition of the geometry, the computational domain, and the computational grid. The choice of the governing equations to be solved, the most suitable numerical techniques to be used to solve them and the real computational work are performed in the computational step. The post-processing step is the visualisation, the analysis, and the verification and validation of the results.

All these steps may be significantly different for different naval hydrodynamic problems, such as hull resistance, propulsion, manoeuvrability, sea keeping, sail aerodynamics, etc. However, the present document aims at providing good practice guidelines, which can be applied to most ship hydrodynamic applications. ITTC specific guidelines are available for ship resistance (7.5-03-02-04), self-propulsion (7.5-03-03-01) and RANS calculations of nominal wakes (7.5-03-03-02).

For the applications covered by the present guidelines, the flow is mostly turbulent and the Reynolds numbers (*Re*) are of the order of 10^6 in model scale and 10^9 in full scale,

$$Re = \frac{\rho V L}{\mu} \tag{0}$$

where V is the ship speed, L the length between the perpendiculars, ρ and μ are the density and viscosity of the fluid, respectively.

The current guidelines are intended for applications with high Reynolds numbers, typical of ship hydrodynamics.

For free surface flow simulations, where the free surface between air and water is solved, the Froude numbers (*Fr*) are typically of the order of 0.1 and rarely above 1,

$$Fr = \frac{V}{\sqrt{gL}} \tag{0}$$

where g is the acceleration of gravity.

Both air and water are typically assumed to be incompressible, allowing uncoupling momentum and continuity equations from the energy equation. Therefore, velocity and pressure fields can be computed independently from the

temperature field. This assumption remains valid for Mach numbers (*Ma*) smaller than about 0.3,

$$Ma = \frac{V}{a} \tag{0}$$

where a is the speed of sound.

Finally, these guidelines are written assuming that the solver is grid-based (either structured or unstructured) and based on the Reynolds Averaged Navier-Stokes equations (RANSE), as found in most commercial and academic CFD packages. Use of mesh-free methods is not covered in this guidelines. For a complete overview on the computational techniques, the reader is referred for example to Hirsch (1989 and 1990) and Ferziger & Peric (2002).

2. PRE-PROCESSING

2.1 Geometry

It is common practice to define the geometry in a CAD (Computer Aided Design) software, which allows for a high control of the surfaces. The geometry can be exported from the CAD software and imported into a CFD pre-processor software (grid generator) with a range of formats. One format compatible with most commercial software is IGES. The accuracy of the geometry should be checked to ensure that the imported surface definitions are reasonably smooth and connect within a given tolerance.

Indicative tolerances for the geometry are $\pm 10^{-5} L$, where L is the ship length for the hull for $Re \sim 10^6$. This tolerance may need to be reduced when smaller scale appendages, such as rudder or propellers, are to be included in the flow computations or when the flow computation requires smaller grid cells than the indicative CAD tolerance. Conversely, as this tolerance is based on the distance to the wall of the first grid point (see §2.3.6) if wall functions are used the geometry tolerance can be increased appropriately.

Due care and attention are required to resolve geometry features such as trailing edges that may be less than an order of magnitude larger than the geometry tolerance. Geometry features that are smaller than the geometric tolerance should be removed from the geometry before it is exported. It is recommended that any additional surfaces, curves or any other geometrical entities that are produced within the pre-processor software, be generated with the same tolerance of the original geometry.

The origin and orientation of the reference coordinate system should be chosen with care because the equations of motion will be solved with respect to this coordinate system and so will be the computation of forces, moments, etc. Therefore, if the aim of the simulation is the estimation of a force (for example the resistance), it is recommended to have one coordinate axis aligned with that force, this to minimize the possibility of mistakes and to have the opportunity

to monitor directly the quantity of interest, if an iterative procedure (in time, for example) is involved. On the other hand, if body motion is involved (as in the case of manoeuvring or sea keeping computations) the best choice of the frame of reference could be driven by the requirement for the rigid body motion solver. Moreover, in order to minimise chances of errors, it is recommended that the same coordinate system is used in both the CAD software and the pre-processor software.

2.2 Computational Domain and Boundary Conditions

In ship hydrodynamics, most of the problems involve external flows, i.e. the fluid is not contained into physical boundaries. Therefore, the computational domain, which represents the volume of fluid to be modelled, must be closed by additional surfaces that do not represent any physical boundary. In the more general case, the tested object is completely surrounded by fluid. In this case a prismatic computational domain can be considered. However, different geometries could better suit different grid topologies. Therefore, the shape of the computational domain should be dictated by the grid strategy in place (see §2.3.1).

The computational domain should include inflow and an outflow surfaces. A Dirichlet condition, i.e. a known velocity field, should be imposed on the inflow boundary. When no specific experimental data is known, the boundary condition should be placed sufficiently far from any solid wall in order to assume an undisturbed far field velocity. When significant lift force is generated with respect to the far field velocity direction, computing the correct deflection of the upstream streamlines is critical to the correct computation of the lift force. If L is the length of the lifting surface, the inlet face should be positioned at least $10L$ upstream of the lifting surface. Conversely, when the lift is negligible and the body is streamlined, the inlet face can be positioned as close as one length upstream the body.

Different boundary conditions can be used on the outflow face, though the most robust option is a Neumann condition (zero gradient) for velocity and pressure. The distance between the outflow face and the body mainly depends on the generation of lift and, for free surface computations, on the reflection of gravity waves. When large lift force is generated, the outlet face should be at least $20L$ downstream the lifting surface, while for other cases the outlet face should be at least L downstream the geometry.

The other non-physical boundaries of the domain should also be placed at least L away from the geometry. For towing tank walls symmetry, imposed velocity or slip conditions can be used, with symmetry the most commonly used.

2.3 Grid

Details on the grid generation process will largely depend on the solver and the type of grid that it can handle (structured multi-block, unstructured, overset, etc.); here are some general guidelines that apply to most solvers.

2.3.1 Structured Grids

Structured grids employ quadrilateral in 2D and hexahedra in 3D and show regular connectivity. Grid topology is the mapping relation between the grid in the physical space (x, y, z) and the computational space (i, j, k). The physical space can be mapped into one computational space or it can be broken down into several spaces. The latter approach is known as a multi-block approach.

For grids made of a single block around a single isolated geometry, O-O or H-O topology is adopted in most cases, although C-O and H-H grids can be applied; see Thompson *et al.* (1985) for examples of these topologies. In either topology, the grid lines in the girth-direction are O-type. The longitudinal grid lines in O-O grids wrap around the geometry whereas those in H-O grids start from the inflow boundary and go through regions ahead of the geometry, side of and aft of it. Thus, when the total number of grid points and the number of grid points along the lines in the normal and in the girth directions are fixed, O-O grids can accommodate more grid points along a geometry than H-O grids. Also, O-O grids can be adapted more easily to a rounded geometry. On the other hand, since the grid lines in the normal directions spread to the outer boundary in O-O grids, the grid resolution in the wake region and the region away from the geometry tends to be lower than the H-O grids. Therefore the choice of grid topology should be based on the nature of the geometry and on consideration of which areas or features of the flow field are more important than others

2.3.2 Unstructured Grids

Unstructured grids do not show a defined topology and can be generated much more easily, resulting in a big advantage for unstructured grid technology. The cells can have several possible shapes (tetrahedral, hexahedra and other polyhedral). However, the resolution of key flow details such as boundary layers and wakes can lead to significantly higher number of cells than equivalent structured grids (see §2.3.4 and §2.3.5) .

Unstructured solvers require typically more memory, are slightly slower and have less accuracy than an equivalent structured solver. This is due to the extra information needed to store and process the grid connectivity and implementation complexities that prevent the use of higher-order schemes.

2.3.3 Hybrid Grids and Other Techniques

Hybrid grids can use both structured and unstructured grids in different parts of the domain. Non-conformal grids are those where the points or cells on a surface do not match. Non-conformal grids are often used at the boundaries between blocks discretized with different grid densities, or on moving (sliding) grids. These types of grids are often required for wake flow analysis where it is important to capture small flow features.

The flow solution algorithm must use special coding in order to interpolate between the non-connected grids. For some methods this interpolation scheme may be defined by interfaces, where the interpolation method is defined across grid boundary surfaces, for other methods the interpolation scheme is defined by overlaps, where the interpolation is defined across local grid volumes. For both of these types of interpolation schemes the formal order of accuracy is likely to be reduced, especially when there are large differences between the grid resolutions and topologies of donor and receptor grids. However, this type of approach can be used to considerably simplify the grid generation process so that locally better quality grids can be produced around the various geometry components and assembled together to form a complete grid.

Non-conformal grids should be used with care because undesirable wave reflections can occur at the interfaces or overlapping regions if the interpolation scheme is unable to resolve correctly the change in grid. This can be alleviated by ensuring that the local change in the grid across the non-conformal region is minimized with similar grid resolution and spacing used for both grid regions. Non-conformal grids can also use different cell types to assist the grid generation process, for example a grid for a detailed rudder with skeg and end plates can be produced using a local prism/tetrahedral grid which is embedded inside a multiblock grid for the hull.

Overset grids are formed by blocks that overlap in space. These grids are particularly convenient when the different blocks experience a relative motion and therefore the overlapping changes with time. For instance, overset grids can be used for ship propulsion where one grid is used for the hull and another is used for the propeller.

2.3.4 Cell Types

Quadrilateral (2D, 4-sided) and hexahedral (3D, 6-sided) cells are supported by almost all CFD codes. Topological attributes of these cells – presence of opposing faces, and relative locations of cell centres and face centres – have been found to be beneficial to spatial accuracy of numerical solutions. In structured grid-based solvers, the presence of stencils (e.g., *i, j, k* coordinates in the computational domain) readily accommodates high-order discretization schemes (e.g., fifth-order convection scheme) that can

enhance spatial accuracy. They are also efficient in terms of usage of cells, since they can be clustered and/or stretched as needed to economically increase the resolution in particular regions (such as boundary layers, or region where large velocity gradients occur). The main shortcoming of these structured grid cells is that it is often very hard to generate high-quality grids for complex geometries.

Unstructured grids give more flexibility in the choice of cell types, simplifying grid generation for complex geometries. The majority of unstructured CFD solvers allow use of arbitrary polyhedral cells such as quadrilaterals and triangles in 2D; and hexahedra, tetrahedra, wedges, pyramids and prisms in 3D to name a few, and combination of all them (hybrid grids). In a typical unstructured grid, the boundary layer is discretized using a prism grid grown out of a triangular grid on the wall, and a tetrahedral grid is used elsewhere away from wall. Compared to typical structured grids, gridding time can be dramatically reduced with unstructured grids. Spatial accuracy for unstructured grid cells such as triangles, tetrahedra, and pyramids can be lower than that for quadrilateral and hexahedral cells. Unstructured grids usually require a larger number of computational points than structured grids in order to achieve a comparable accuracy. Furthermore, spatial accuracy of the majority of unstructured grid finite volume solvers is limited to second order.

Which cell types to use for a given problem really depends on many factors such as the solver (what cell types can the solver support), objective of the computation (do fine details of the flow need to be resolved?) and hardware resources (are computational resources available to run cases involving large grids?). For relatively simple geometries, consider using a high-quality hexahedral grid. However, if body-motion (e.g. free sinkage and trim) is involved, consider using overset grids if your CFD solver includes this capability.

For complex geometries (e.g. a fully-appended ship) for which a high-quality structured grid is difficult to generate, consider using an unstructured grid (preferably a hybrid unstructured grid) or overlapping grids.

Avoid using tetrahedral cells for boundary layers, near the free surface, and in the regions where high resolution is required. The use of hexahedral or prismatic cells would result in higher accuracy and better convergence rates.

2.3.5 Grid Resolution

The number of grid points, and therefore the resolution of the grid, should be decided based on the chosen turbulence model and on the temporal and spatial scales of the flow features of interest. In the following some practical guidelines are provided but interested readers can find in Pope (2000) and Sagaut (2006) more rigorous discussion based on the physics of the resolved

time and statistical averages of the flow field, and the modelled turbulent fluctuations.

Whenever possible, use hyperbolic grid generators to guarantee as much as possible an orthogonal grid near the wall. Grids orthogonal to the domain boundaries, where the boundary conditions are imposed, are recommended. For some boundary conditions, such as symmetry and wall conditions, this orthogonality condition may be mandatory for some solvers.

Provide refinement where flow features of interest are expected, in accordance with the size of the feature to be simulated. Where the flow features of interest are not known before-hand, it may be necessary to use an iterative process to establish the existence of the key flow features for a given geometry. This requires obtaining and examining an initial flow solution, and subsequent grid refinement in the regions of interest. This process should be carried out until some measure of grid refinement is satisfied to ensure that flow features are sufficiently resolved.

If overset grids are used, check that overlap is sufficient for the number of fringes that are needed in the code and the order of accuracy. If motions are computed, this requirement holds for all time steps.

Ensure sufficient resolution where high curvature geometry is present, especially around leading and trailing edges. An appropriate grid structure can enable more efficient use of computing resources but at the expense of increased grid generation time and complexity.

In free surface flow simulations, when the free surface is modelled, always use orthogonal grids to resolve it if possible. Use no less than 40 grid points to resolve the shortest wave length when a second-order spatial accuracy is used, though this condition may be extremely expensive and unnecessary for low Froude number flows. The same resolution can be achieved with different spatial accuracies varying linearly the spatial resolution. For instance, for a fourth-order spatial resolution, using 20 grid points is equivalent to using 40 grid points with a second-order spatial resolution.

2.3.6 Near-wall Region

The number of grid points within the boundary layer is determined by the level of accuracy required, the turbulence model chosen, and whether wall functions are used.

Guidelines are provided in terms of the non-dimensional wall distance of the first point from the wall, y^+. We define

Friction velocity:

$$u^* = \sqrt{\frac{\tau_w}{\rho}} \tag{0}$$

Non-dimensional wall velocity:

$$u^+ = \frac{u}{u^*} \qquad (0)$$

Non-dimensional wall distance:

$$y^+ = \frac{\rho u^* y}{\mu} \qquad (0)$$

where τ_w is the skin friction on the wall, u the local streamwise velocity component, and y is the wall-normal coordinate.

A near-wall turbulence model resolving the laminar sub-layer needs at least 3 points inside it, which using $y^+ = 2$ results in an expansion ratio of 1.5, the largest acceptable, but a maximum value closer to 1.2 is recommended. In most cases a $y^+ \leq 1$ will be used with expansion ratios around 1.1. Wall functions start farther out in regions of smaller velocity gradients and can use larger expansion ratios, as large as 1.2 for coarse grids. When direct integration is used the total number of points within the boundary layer can be very large, more than 20, while when wall functions are used, then less than 15 grid points could suffice. For wall functions, it is recommended that the first point from the wall is well within the logarithmic layer of the boundary layer, therefore: $30 < y^+ < 100$. Table 1 summarizes the recommended values.

	First point	Expansion ratio	Points within boundary layer
Near wall	$y^+ \leq 1$	1.2	20
Wall functions	$30 < y^+ < 100.$	1.2	15

Table 1. Recommended values for grid design at the wall.

Once the desired y^+ has been chosen (either to comply with wall function requirements or to resolve the boundary layer), the distance y of the first point from the wall can be computed as follows:

$$y = \frac{y^+ L}{Re_L \sqrt{\frac{C_f}{2}}} \qquad (0)$$

where C_f is the friction coefficient, which can be estimated *a priori* considering an equivalent boundary layer on a flat plate with zero pressure gradient. In particular, for laminar flow conditions

$$C_f = \frac{1.328}{\sqrt{Re_L}} \qquad (0)$$

for transitional flow conditions

$$C_f = \frac{0.455}{\left[\ln\left(Re_L\right)\right]^{2.58}} - \frac{1700}{Re_L} \qquad (0)$$

for turbulent flow on a smooth surface

$$C_f = \frac{0.455}{\left[\ln\left(Re_L\right)\right]^{2.58}} \qquad (0)$$

and for fully turbulent flow on a rough surface with roughness height ε

$$C_f = \frac{1}{\left[1.89 - 1.62\ln\left(\frac{\varepsilon}{L}\right)\right]^{2.5}} \qquad (0)$$

See §3.3 for more details on the roughness height.

Equations (8-11) do not take into account the surface curvature and the pressure gradient. For the friction coefficient over a ship hull, the 1957' ITTC formulation allows a better approximation of full scale friction resistance from model scale friction resistance:

$$C_f = \frac{0.075}{\left[\log_{10}\left(Re_L\right) - 2\right]^2} \qquad (0)$$

It should be noted that Eq. (12), though widely used and accepted, was developed for typical hull forms in existence before 1957 and thus it may not produce good results for modern and unconventional hull forms.

Since Eqs. (7-12) are only generic approximations, y^+ should always be checked *a posteriori* once the solutions are obtained and if necessary the grids should be modified accordingly.

2.3.7 Grid Quality

Check the grid quality to guarantee that all volumes are positive (positive Jacobian in structured grids), skewness and aspect ratio are acceptable, and that orthogonality is nearly satisfied in most places.

Typically the 3x3 determinant for structured grids should be greater than 0.3, as a measure of the Jacobian and associated skewness. However, it may be necessary to have a few small cells where the 3x3 determinant is no better than 0.15. For these cases it may be necessary to use a smaller time step or increased under-relaxation in order to achieve converged results.

3. COMPUTATION

3.1 Turbulence Models

3.1.1 Turbulent Scales

As mentioned in the Introduction, these guidelines assume that the incompressible Navier-Stokes equations for Newtonian fluids are solved. The Reynolds number range is such that laminar to turbulent transition occurs and turbulent flow must be modelled. In the present guide

it is assumed that the turbulence is modelled with a RANS approach, where the instantaneous velocity is split into the sum of its statistical average and a turbulent fluctuation which is modelled by the turbulence model. It has to be highlighted that, when dealing with unsteady flow fields, it is assumed that a clear separation between the time scales of the mean motion and time scales of turbulent motion exists. The fundamental assumption in unsteady RANS computations is that the averaging time is bigger than turbulent time scales (which are averaged) but much smaller than mean flow time scales (which are solved).

It is worth noting more computationally demanding approaches are becoming more and more common in recent years. Direct Numerical Simulation (DNS) solves the Navier-Stokes equations to the resolution of the smallest turbulent scales but needs grids and time resolution which are not yet achievable for high Reynolds numbers, such as those typical of ship hydrodynamic applications.

Large Eddy Simulation (LES) does not average the Navier-Stokes equations in time, but filters them in space. This results in transient computations on extremely dense grids as they aim to resolve the larger turbulence scales and modelling the scales smaller than the grid resolution. LES is used currently for highly demanding flows where the transient nature of the turbulence needs to be resolved to smaller scales, like in the case of acoustic noise.

Detached Eddy Simulation (DES) is a hybrid method that reduces the required computational effort by solving the (unsteady) RANS equations in the near-wall region and applying LES in the rest of the domain.

3.1.2 RANS Models

Turbulence modelling has been an important research topic over the past decades. A large number of models have been proposed, tested and applied, but no universal model has been developed. Thus one is forced to choose the best model available for each specific application. The majority of turbulence models are based on the so-called Boussinesq hypothesis, which defines a turbulent or eddy viscosity (as opposed to the molecular viscosity) to account for the effect the turbulence motion has on the mean flow. For a detailed description of turbulence models, see for example Wilcox (2006).

Zero-equation, or algebraic models express the eddy viscosity in terms of the mean flow variables and mean flow gradients without solving any additional equations. They are hardly ever used in ship hydrodynamic applications.

One-equation models solve one additional equation (i.e. in addition to the momentum and mass conservation equations) for the eddy viscosity. Regularly used in ship hydrodynamics are models by Menter and by Spalart-Allmaras. These models are sometimes extended with a

correction for vortical flow, to improve wake field predictions.

Two-equation models solve two additional equations for the eddy viscosity, one for the turbulence kinetic energy (k), and one for its dissipation rate (typically ε or ω). These models have shown to be able to give accurate predictions in ship hydrodynamics, especially certain versions of the $k-\omega$ model and are by far the most applied ones (80% of the submissions for the Gothenburg 2010 Workshop, see Larsson *et al.*, 2013).

An important class of turbulence models, not based on the Boussinesq hypothesis, are the Reynolds-stress models, and versions thereof. Rather than introducing an eddy-viscosity, they aim to solve the equations for the six Reynolds stress components directly. Apart from that, additional equations have to be solved, since terms in these equations require modelling as well. Consequently Reynolds-stress models are more computationally intensive, and often harder to converge, compared to one or two-equation models. However, they contain more physics and are therefore be able to capture flow features that cannot be solved by eddy-viscosity models. Explicit algebraic Reynolds stress models are based on the algebraic expressions for each stress component which can be determined from two equation turbulence models. They are considered to be a reasonable compromise between complexity and performance.

3.2 Near-wall Modelling

In the near-wall region flow variables have very large normal gradients. In the viscous sub-layer, viscous effects suppress turbulent fluctuations and have a dominant role in momentum transfer. This requires proper modelling of the near-wall region. Two approaches are available:

1. use of near-wall turbulence models that are able to resolve the flow all the way down to the wall;
2. use of wall functions, semi-empirical formulae that used to bridge the solution variables in the fully turbulent log-law region and the corresponding quantities on the wall, without actually resolving the flow in between (i.e. the viscous-affected sub-layer).

Resolving flow by a near-wall model is a more rational approach. However, it requires grid nodes highly refined towards walls ($y^+=1$ as discussed in §2.3.6). This often leads to extremely high aspect ratio for the near-wall grid cells, not only resulting in large grids but also posing numerical instability problems in the solution procedure, making the computations substantially heavy. This is especially the case for full scale calculations.

Using wall functions eases the need of extreme refinements near the wall. In terms of y^+ value, it suffices to have the first grid node in the range $30<y^+<100$ (see §2.3.5). However, wall functions are based on two-dimensional flow

typically at zero pressure gradients and it is well-known that these analytical expressions become less valid, or even invalid, with increasing adverse pressure gradients. Though corrections exist for wall functions in the presence of pressure gradients, these make the wall functions less stable. Thus it is a trade-off between accuracy and computational effort. Wall functions should be avoided if possible, and used with care when necessary.

3.3 Surface Roughness

The flow in the boundary layer is not affected by the surface roughness when the physical roughness height k_s is small compared to the thickness of the laminar sublayer. In particular, the effect of the surface roughness is negligible when k_s is lower than approximately $2\nu / u^*$, in this case the surface is said hydrodynamically smooth. For hydrodynamically rough surfaces, the logarithmic Law of the Wall is modified adding a term, Δu^+, which is a function of k_s. Unfortunately this function depends on the type of roughness and no universal formulation is known.

In most ship CFD applications, surfaces are assumed to be hydraulically smooth. Otherwise the effect of the roughness can be modelled either modifying the wall functions or the turbulence boundary conditions. When wall functions are used, then different functions for Δu^+ are used depending on the range of k_s and the types of roughness, such as sand grains or spheres. Interested readers are invited to consult Schlichting and Gersten (2000). No general guidelines can yet be given for wall-resolved boundary layers. However, the roughness profile is typically not resolved, the blockage effect of the roughness is taken into account using a modified wall unit, $y^+_{mod} = y^+ + k_s / 2$, and therefore the wall-adjacent cell should be higher than k_s.

3.4 Numerical Schemes

In the majority of industrial CFD codes, diffusion terms in the governing equations are discretized using a second-order (central differencing) scheme by default. Thus, spatial accuracy is largely determined by discretization scheme used for convection terms.

The first-order upwind (FOU) scheme, offered in many commercial CFD codes often as a default scheme, is very stable. However, it introduces large numerical (artificial) diffusion - that is why it is so stable. Its use can only be justified when the physical diffusion (turbulent + molecular) is much higher than the introduced numerical diffusion. This is true within the boundary layer, allowing use of hybrid schemes that use lower order near the wall and higher order farther out.

The majority of high-order convection discretization schemes in popular use today formally have a second-order of accuracy with an upwind bias. All these second-order upwind (SOU) schemes differ from one another in terms of the flux limiter used to suppress unphysical oscillations in the solutions. Still higher-order schemes such as fifth-order scheme exist; however, not all CFD solvers offer such higher-order schemes. Moreover, it has to be taken into account that higher order schemes are, in general, less robust than first or second order accurate schemes. Therefore, their use should be considered with special care. These schemes can be used in regions of high-quality grid (near orthogonal, low aspect ratio, low expansion ratio), like in Cartesian overset refinements, while lower-order schemes can be used in others regions. Higher-order schemes are standard in immersed boundary methods in Cartesian grids.

The SOU scheme is both reasonably accurate and robust, and for that reason is an industrial workhorse for convection discretization in body-fitted approaches. The SOU scheme is therefore recommended for all convection-diffusion transport equations.

First order accurate schemes in time may be used only if a steady-state is sought.

Although outside the scope of the present guidelines, it is worth mentioning that second-order central differencing (CD) scheme is often used in large eddy simulation (LES) and direct numerical simulation (DNS) for its low-dissipation that is critical to accurately resolve turbulent structures. However, CD scheme is inherently unstable, raising convergence issues for cases involving fine grids and small effective viscosity (large local Reynolds number). One should consider using a stabilized form of central differencing.

3.5 Free Surface

There are two major categories in free surface models. For interface fitting approaches the numerical grid is conformed to the free surface shape. Interface capturing approaches define the free surface as an iso-surface of a marker function and thus the grid does not have to conform to the free surface.

In general, the interface fitting approach is more accurate and efficient than the capturing approach, since free surface boundary conditions can be applied in the exact free surface location. Therefore the interface fitting model may be selected whenever possible. However, it should be noted that a deformation/re-gridding procedure is essential in order to keep the gridlines following the deformation of the free surface. This may cause severe distortion of gridlines even though the initial grid conforming to the undisturbed free surface has a good quality.

Difficulties in grid generation and/or re-gridding can be avoided with interface capturing approaches. Also, for large deformation of free

surface, such as overturning or breaking waves, interface capturing methods should be used since surface-tracking methods cannot resolve the change in surface topology. Since the capturing methods demand finer grid resolution near the interface, grid generation requires more attention. The choice of level-set method or volume-of-fluid method in for capturing approach has little impact in the final solutions. Although the numerical methods are different, well implemented flow solvers provide similar results for both models.

When a volume-of-fluid method is used, the volume fraction equation requires special care, since the transported quantity is essentially a step function across the free surface, and traditional convection schemes designed for convection-diffusion equations perform poorly in this case. Though geometric reconstruction offers the best possible solution, it is expensive and frequently pure transport is used, which does not guarantee a sharp interface throughout the computation. It has been found that convection schemes with some degree of downwind bias resolve the sharp interface much better.

3.6 Time Step

In explicit solvers the time step is chosen to satisfy the Courant–Friedrichs–Lewy (CFL) condition or to resolve the flow features of interest, whatever results smaller. The CFL condition is a condition on the CFL number, which represents the number of points travelled by a particle of fluid within a time interval equal to the time step:

$$\frac{|\bar{u}|\Delta t}{\Delta x} < C_{max} \tag{0}$$

where $|\bar{u}|$ is the local velocity, Δt is the time step, and Δx is the linear cell size in the flow direction. In explicit solvers, where $C_{max} < 1$, the CFL condition is more demanding than in implicit solvers, where C_{max} can be larger than 1. In this case the time step is decided by the flow features. As a rule of thumb:

- for periodic phenomena (e.g. roll decay, vortex shedding, incoming waves etc.) use at least 100 time steps per period;
- for complex unsteady phenomena (e.g. wetted transom instabilities), use at least 20 time steps per period for the highest frequency to be resolved;
- for rotating propellers use at least 180 time steps per revolution;
- Δt must be smaller than 0.01 L/U if one or two equation turbulence models are used, while it should be smaller than 0.001 L/U if Reynolds stress turbulence model is used.

3.7 Parallel Computing

In modern computers, as a rule of thumb choose the number of cores so that you use from 50,000 to 500,000 grid points per core. In some

cases fewer points per core can be used if the CFD code scales well to obtain faster turnaround, or more points per core if processor memory allows it and turnaround time is not critical.

To maximize performance, try to distribute the load evenly between nodes. For instance, if running in a Linux cluster with 2 quad-core processors (8 cores) per node, and your case needs 10 cores, you can distribute your load in two nodes using 8-2 or 5-5 configurations. The second balances the load per node better since it provides more memory bandwidth per process.

Modern workstations with shared memory are available with up to 64 processors, though much larger specialized systems are produced. High-performance clusters are typically cheaper per processor for large systems (thousands of cores) but use distributed memory. Shared-memory systems allow all processors access all memory, resulting in easier programming and better scalability of most applications. On the other hand, distributed memory systems provide massive number of processors for very large computations.

3.8 Convergence

A number of convergence criteria should be defined and examined in order to ensure reliable convergence of solution. While different levels of convergence are acceptable, the uncertainty due to the convergence should always be estimated with a verification procedure.

Convergence should be achieved for both steady-state and transient computations. In the case of implicit transient computations, convergence must be evaluated at each time step.

The level of convergence should be assessed by the history of residual variations for the mass and momentum equations. Residuals indicate how far the present approximate solution is away from perfect conservation of mass and momentum. Thus the residual for a discretized equation is defined as the L_1-Norm of the imbalance between the left and the right hand side of that equation over all the computational points. Usually the residual is scaled by a reference value. Sometimes the L_2-Norm or L_∞-Norm are used to define the residual.

CFD users do not need to worry about the definition of residuals, as they are often pre-defined by code developers. Instead, attention should be paid to the selection of convergence criteria. The recommended criterion is the drop of scaled residuals by at least three orders of magnitude off their initial values. However it is acknowledged that due to complexity of some problems or oscillatory convergence this criterion may not be satisfied. In such cases, the convergence of forces and moments, and the convergence of the unknowns in key regions of the flow field (e.g. at propeller plane for nominal wake calculations) can be monitored.

4. POST-PROCESSING

4.1 Visualization

A number of post processing plots should be used as a minimum sub-set of information to ensure that the correct settings have been used for each computation. This should include contour plots of the pressure coefficient, skin friction coefficient and y^+ on the geometry surface, and vector plots of the boundary layer profiles along the geometry.

Reasonable checks should be carried out to ensure that the solution is physically sound and that it does not show the footprint of the grid.

4.2 Verification and Validation

The verification and validation of numerical simulations is an essential tool for the interpretation of the results and the identifications of those aspects of the simulation that can be improved.

The numerical uncertainty is due to the numerical error introduced by the method used to solve the chosen set of equations with the given boundary and initial conditions, while the modelling uncertainty is associated to the modelling error, which is the difference between the exact solution of this set of equations and the reality. The numerical uncertainty can be estimated performing a set of numerical simulations, while experimental data is necessary to estimate the modelling error.

The ITTC procedure 7.5-03-01-01 provides "methodology and procedures for estimating the uncertainty in a simulation result". The procedure includes a *verification* procedure for the estimation of the numerical uncertainty, and a *validation* procedure for the estimation of the modelling error.

Any numerical results should be considered together with its numerical uncertainty. This becomes of critical importance when different design solutions are ranked based on the CFD results, since the uncertainty reflects, and allows the computation of, the probability that the ranking is correct.

The estimate of the uncertainty must be based on a verification and validation procedures but it can be assumed that the uncertainty does not change significantly with small changes of the boundary and initial conditions. Therefore, once the uncertainty has been estimated for one simulation, this can often be used as a reasonable estimate of the uncertainty of similar simulations. For example, if a parameter study on the hull geometry is performed, verification can be performed for only one or few values of such parameter. Care should be taken on these extrapolations that no significant changes on the flow field are observed between similar cases.

It must be noted that the difference between the uncertainties of different quantities computed with the same simulation can be higher than one order of magnitude. For instance, the uncertainty of the ship resistance is expected to be significantly lower than the uncertainty of the local flow speed in one point of the domain. Also, the higher the order of the derivatives of the primitive unknowns, the higher the uncertainty expected. For instance, the uncertainty on the velocity is expected to be lower than the uncertainty on the vorticity. Therefore, the uncertainty must be estimated for every quantitative result gathered from the simulation.

5. REFERENCES

Hirsch, C. (1989) Numerical computation of internal and external flows: Fundamentals of numerical discretization, Wiley–Blackwell.

Hirsch, C. (1990) Numerical computation of internal and external flows: Computational methods for inviscid and viscous flows, Wiley–Blackwell.

Ferziger, J.H. and Peric, M. (2002) Computational Methods for Fluid Dynamics, Springer.

Thompson, J.F., Warsi Z.U.A., Mastin, C.W. (1985) Numerical Grid Generation: Foundations and Applications.

Pope, S. (2000) Turbulent Flows, Cambridge University Press.

Sagaut, P. (2006) Large Eddy Simulation for Incompressible Flows, an Introduction, Springer.

Schlichting, H., and Gersten, K. (2000) Boundary-layer theory, Springer.

Wilcox, D.C. (2006) Turbulence Modeling for CFD (Third Edition), DCW Industries.

Larsson, L., Stern F., Vissonneau, M. (2013) Numerical Ship Hydrodynamics: An assessment of the Gothenburg 2010 Workshop, Springer.

6. REFERENCES TO OTHER CFD GUIDELINES

AIAA, 1988, AIAA guide for the verification and validation of computational fluid dynamics simulations, AIAA G-077-1998.

ASME Committee PTC-61, 2009, ANSI Standard V&V 20. ASME Guide on Verification and Validation in Computational Fluid Dynamics and Heat Transfer.

ERCOFTAC Best Practice Guidelines. http://www.ercoftac.org/publications/ercoftac_best_practice_guidelines/

ITTC Recommended Procedures and Guidelines. Uncertainty analysis in CFD, Guidelines for RANS codes, No. 7.5-03-01-02 (1999).

ITTC Recommended Procedures and Guidelines. CFD User's Guide, No. 7.5-03-01-03 (1999).

ITTC Recommended Procedures and Guidelines. CFD Verification, No. 7.5-03-01-04 (1999).

ITTC Recommended Procedures and Guidelines. Uncertainty Analysis in CFD Verification and Validation. Methodology and Procedures, No. 7.5-03-01-01 (2008).

ITTC Recommended Procedures and Guidelines. Practical Guidelines for Ship Resistance CFD, No. 7.5-03-02-04 (2014).

ITTC Recommended Procedures and Guidelines. Practical Guidelines for Ship Self-Propulsion CFD No. 7.5-03-03-01 (2014).

ITTC Recommended Procedures and Guidelines. Practical Guidelines for RANS Calculation of Nominal Wakes, No. 7.5-03-03-02 (2014).