

Notes on Running the Cases with CFD

1. It is important to note that most of the cases described on this website were run using **NONDIMENSIONAL** CFD codes that solve the compressible Reynolds-averaged Navier-Stokes equations for an ideal, calorically-perfect gas. Therefore, the descriptions of the cases give a **Reynolds number** per unit length (referenced to the grid provided) and a **Mach number**. These parameters completely define the problem (a temperature is also specified for determining the dynamic viscosity via Sutherland's law, but this does not have as significant an influence on the resulting flowfield as Re and M). If you are running using a dimensional code on the provided grids, it is crucial that you at least **set conditions to match the prescribed Reynolds number and Mach number exactly**.

Let us use the [Flat Plate Verification Case](#) as an example. That case specifies $M=0.2$, $Re_L=5$ million, where L is unit 1 of the grid, and the plate portion of the grid is 2 units long. (The $T_{ref}=540R=300K$ is also specified; it is used solely as an input to [Sutherland's law](#) (for example) for determining the dynamic viscosity.) These parameters completely define the problem.

For a dimensional code, one must choose units, and force the input choices to yield the correct Re and M. For this case the goal is to put this plate in a "virtual wind tunnel" so as to achieve $Re=5$ million per unit length and $M=0.2$.

- For example, one could assume that the flat plate grid is in meters. Choose a speed of sound. It can be anything, but say it is 342 m/s. (Depending on your particular code, other inputs would have to be consistent with this choice, assuming an ideal gas.) Thus, reference velocity must be chosen as 0.2 times 342 = 68.4 m/s. Then, reference kinematic viscosity must be chosen so that Re per meter is 5 million: i.e., $\nu = \mu/\rho$ must be $1.368 \times 10^{-5} \text{ m}^2/\text{s}$.
- As another example, assume that the flat plate grid is in feet. Choose speed of sound to be (say) 1122 ft/s. Reference velocity must be 0.2 times 1122 = 224.4 ft/s. Reference kinematic viscosity must be chosen so that Re per foot is 5 million: i.e., $\nu = \mu/\rho$ must be $4.488 \times 10^{-5} \text{ ft}^2/\text{s}$.

Further details about the nondimensional codes used on this website: the Prandtl number Pr is taken to be constant at 0.72, and turbulent Prandtl number Pr_t is taken to be constant at 0.9. The dynamic viscosity is computed using Sutherland's Law (See White, F. M., "Viscous Fluid Flow," McGraw Hill, New York, 1974, p. 28). In Sutherland's Law, the local value of dynamic viscosity is determined by plugging the local value of temperature (T) into the following formula:

$$\mu = \mu_0 \left(\frac{T}{T_0} \right)^{3/2} \left(\frac{T_0 + S}{T + S} \right)$$

where $\mu_0 = 1.716 \times 10^{-5} kg/(ms)$, $T_0 = 491.6R$, and $S = 198.6R$. The same formula can be found [online](#) (with temperature constants given in degrees K and some small conversion differences). Note that in terms of reference quantities, Sutherland's Law can equivalently be written:

$$\left(\frac{\mu}{\mu_{ref}} \right) = \left(\frac{T}{T_{ref}} \right)^{3/2} \left(\frac{T_{ref} + S}{T + S} \right)$$

where μ_{ref} is the reference dynamic viscosity that typically corresponds to the freestream, and T_{ref} is given for each problem of interest. This latter form may be more convenient for nondimensional codes. (Specific details regarding an implementation of Sutherland's Law in nondimensional codes can be found in [handwritten notes describing Sutherland's Law in CFL3D and FUN3D](#).)

2. As discussed elsewhere on this site, note that all cases are **compressible flow** verification and validation cases. Although many are at low Mach numbers such that the flow is "essentially" incompressible, if you run with an incompressible code your results will not exactly correspond with the compressible code results given here.

3. Throughout the webpages, there are many comparisons with experimental $\overline{u'v'}$, here referred to as "turbulent shear stress."

(Officially, the turbulent shear stress is $\overline{\rho u'_i u'_j}$, with this giving the proper units for a stress. However, for low-speed flows with density approximately constant, it is very common to also refer to $\overline{u'v'}$ as the

"turbulent shear stress," even though its units are velocity squared.) The equations for $\overline{u'v'}$ are as follows. **First, use the**

definition: $\tau_{ij} \equiv -\overline{\rho u_i'' u_j''}$ (see [Implementing Turbulence Models into the Compressible RANS Equations](#) page). Here, the double prime is used for Favre-averaged variables. **Since most cases on the website are at low Mach numbers such that the flow is "essentially" incompressible, we will use $\overline{\rho u_i'' u_j''}$ and $\overline{\bar{\rho} u_i' u_j'}$ interchangeably. Thus:**

$$\overline{u_i' u_j'} \approx -\frac{\tau_{ij}}{\bar{\rho}}$$

Depending on the turbulence model, the τ_{ij} may be solved directly (e.g., full RSM), may be given by a complex nonlinear equation (e.g., explicit algebraic stress), or may be given by the simple Boussinesq relation:

$$\tau_{ij} = 2\hat{\mu}_t \left(\hat{S}_{ij} - \frac{1}{3} \frac{\partial \hat{u}_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \bar{\rho} k \delta_{ij}$$

where $\hat{S}_{ij} = (\partial \hat{u}_i / \partial x_j + \partial \hat{u}_j / \partial x_i) / 2$, and $\hat{\mu}_t$ is the eddy viscosity obtained by the turbulence model.

Therefore, for the turbulence models that use the Boussinesq relation, $\overline{u'v'}$ can be computed from:

$$\overline{u'v'} = -\frac{\hat{\mu}_t}{\bar{\rho}} \left(\frac{\partial \hat{u}}{\partial y} + \frac{\partial \hat{v}}{\partial x} \right)$$

4. There is sometimes confusion with regard to the form of the production term when the Boussinesq relation is invoked. The Boussinesq relation is:

$$\tau_{ij} = 2\hat{\mu}_t \left(\hat{S}_{ij} - \frac{1}{3} \frac{\partial \hat{u}_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \bar{\rho} k \delta_{ij}$$

as described in (3) above. Now, for example, the k production term is:

$$P = \tau_{ij} \frac{\partial \hat{u}_i}{\partial x_j}$$

For incompressible flows, this exactly becomes:

$$P_{inc} = \frac{1}{2} \hat{\mu}_t \left(\frac{\partial \hat{u}_i}{\partial x_j} + \frac{\partial \hat{u}_j}{\partial x_i} \right) \left(\frac{\partial \hat{u}_i}{\partial x_j} + \frac{\partial \hat{u}_j}{\partial x_i} \right) = \hat{\mu}_t \frac{\partial \hat{u}_i}{\partial x_j} \left(\frac{\partial \hat{u}_i}{\partial x_j} + \frac{\partial \hat{u}_j}{\partial x_i} \right) = 2 \hat{\mu}_t \hat{S}_{ij} \hat{S}_{ij} \equiv \mu_t \hat{S}^2$$

Many references cite the above incompressible form of the production term. Except for very high Mach number flows (e.g., arguably $M > 2 - 5$, depending on the case), P_{inc} is often a very good approximation for P .

5. All of the cases on this website are intended to be run as "fully turbulent," in the sense that the RANS turbulence model should be active and engaged. However, it is recognized that most models inherently "trip" on their own over some finite distance (which may be code and grid dependent for a given case). See, e.g., AIAA Journal, Vol. 47, No. 4, April 2009, pp. 982-993, <https://doi.org/10.2514/1.39947>. The main intent is that the tripping occurs reasonably quickly over the body so that transition is a negligible issue for the cases given.

Some models give specific recommended freestream levels of turbulence for all situations, some ask for the user to input values for freestream turbulence intensity (Tu) and freestream turbulence length scale or freestream eddy viscosity, and some make no recommendations. In the cases on this site, various approaches are taken. For CFL3D and FUN3D using the SA and SST/SST-V models, the freestream levels are described on the VERIFICATION pages (see, e.g., the [SA Expected Results for Flat Plate](#) page or the [SST Expected Results for Flat Plate](#) page). Additional details can be found in the [CFL3D User's Manual, Appendix H](#). These same freestream settings were used by these two codes for the VALIDATION cases as well. (Note that although freestream k is typically fixed in CFL3D and FUN3D for two-equation models, due to the nondimensionalization used the resulting turbulence intensity (Tu) is a function of reference Mach number, so Tu varies from case to case).

For some of the other contributed code results, different freestream turbulence settings were used, and for some codes the freestream turbulence settings are not known. This inconsistency/uncertainty in freestream turbulence levels may contribute to some of the (usually minor) code-to-code differences in results on the VALIDATION pages. Presumably, however, the very close agreements between codes is an indication that the freestream turbulence levels used by the various codes are sufficient to ensure adequate "fully turbulent"

flow while not being so unphysical as to corrupt the turbulent solutions. For example, see the bottom of the page [SST Expected Results for Flat Plate](#) for a quantification of the effects that freestream turbulence levels have on fully turbulent flat plate solution measures of interest.

6. Many of the cases are evaluated based on surface skin friction coefficient, surface pressure coefficient, lift coefficient, and/or drag coefficient. The definitions of these are "standard" and can be found in many textbooks or on websites such as [CFD-ONLINE](#) or [Wikipedia](#). Surface skin friction coefficient is defined by:

$$C_f = \frac{2\tau_w}{\rho_{ref}U_{ref}^2}$$

where typically the "ref" density and velocity are freestream, but this may depend on the case. The τ_w is the wall shear stress

$$\tau_w = \mu_w \left(\frac{\partial u'}{\partial n} \right)_w$$

where μ is the dynamic viscosity, u' is the velocity along the boundary (parallel to the wall), n is the direction normal to the wall, and subscript "w" indicates at the wall. Because (for most aerospace applications) the x-direction is generally used as the "downstream" direction, the skin friction coefficient is often defined to be positive or negative based on the sign of the u -velocity component of u' (i.e., based on the sign of the x-direction component of the skin friction coefficient). In 3-D, this practice may be somewhat ambiguous.

Surface pressure coefficient is defined by:

$$C_p = \frac{2(p_w - p_{ref})}{\rho_{ref}U_{ref}^2}$$

Here, the "ref" pressure, density, and velocity are again typically freestream, but this may depend on the case. Lift coefficient is defined by:

$$C_L = \frac{2L}{\rho_{ref}U_{ref}^2 A}$$

where L is the lift force and A is the reference area (in 2-D it is reference length, or area per unit span). Similarly, drag coefficient is defined by:

$$C_D = \frac{2D}{\rho_{ref}U_{ref}^2 A}$$

where D is the drag force.

7. Many of the cases on this website were run using the NASA LaRC CFD codes [CFL3D](#) and [FUN3D](#). These codes both solve

the [compressible Reynolds-averaged Navier-Stokes \(RANS\)](#) equations. As described in detail in Note 1 above, they both use Sutherland's law for the dynamic viscosity (White, F. M., "Viscous Fluid Flow," McGraw Hill, New York, 1974, p. 28), Prandtl number Pr is taken to be constant at 0.72, and turbulent Prandtl number Pr_t is taken to be constant at 0.9. These and other details can be found on the respective websites for these codes. Unless otherwise noted, the runs using both CFL3D and FUN3D throughout the TMR web pages were made with the following:

- **First order turbulence advection in the turbulence transport equations.** As the grid is refined, results using 1st order turbulence advection will go to the same results as that for 2nd order, but on a given grid the former results may be less accurate.
- **Use of P_{inc} for the "exact" production term in two-equation models (see note 4 above).** This approximation neglects two terms that are exactly zero for incompressible flows: $-2/3 \mu_t (du_k/dx_k)^2$ and $-2/3 \rho k (du_k/dx_k)$. For low Mach number flows, this approximation makes negligible difference, but the approximation may have some noticeable influence as the Mach number increases. Currently, all of the TMR [verification cases](#) are at $M=0.5$ or lower.
- **In linear two-equation models, the $(2/3)\rho k$ term is neglected in the Boussinesq approximation for τ_{ij} in the momentum and energy equations.** (See the [Implementing Turbulence Models into the Compressible RANS Equations](#) page.)

(More recent versions of the CFL3D and FUN3D codes allow most of the above defaults to be overridden.)

Note: as of the time of this writing, both CFL3D and FUN3D solve the nonconservative form of all turbulence equations by default (e.g., carrying k and ω rather than ρk and $\rho \omega$).

8. In the description of the various turbulence models, it is implicitly presumed that variables that appear in denominators never reach zero (i.e., division by zero should never occur). In practice in CFD codes, this may be enforced in different ways; but the precise method is generally not specified in the description of the turbulence model itself. Thus, potential inconsistencies in implementation may exist at this level between codes that otherwise have identical coding. It is assumed that this level of inconsistency would yield insignificant differences. Note that (similar to the mean flow quantities density and

pressure), many of the turbulence quantities should always remain non-zero and positive in the flowfield away from boundaries, including k , ω , k_L , μ_t , R_t , and ϵ .

9. Unless otherwise explicitly stated, the equations on this site assume an inertial frame of reference. If you are computing in a non-inertial frame (for example, in a rotating reference frame), then you will need to appropriately account for it in the mean flow and turbulence equations. In the turbulence equations, the most common impacted term is the vorticity. The SA-RC and SST-RC models include accounting for a rotating reference frame.

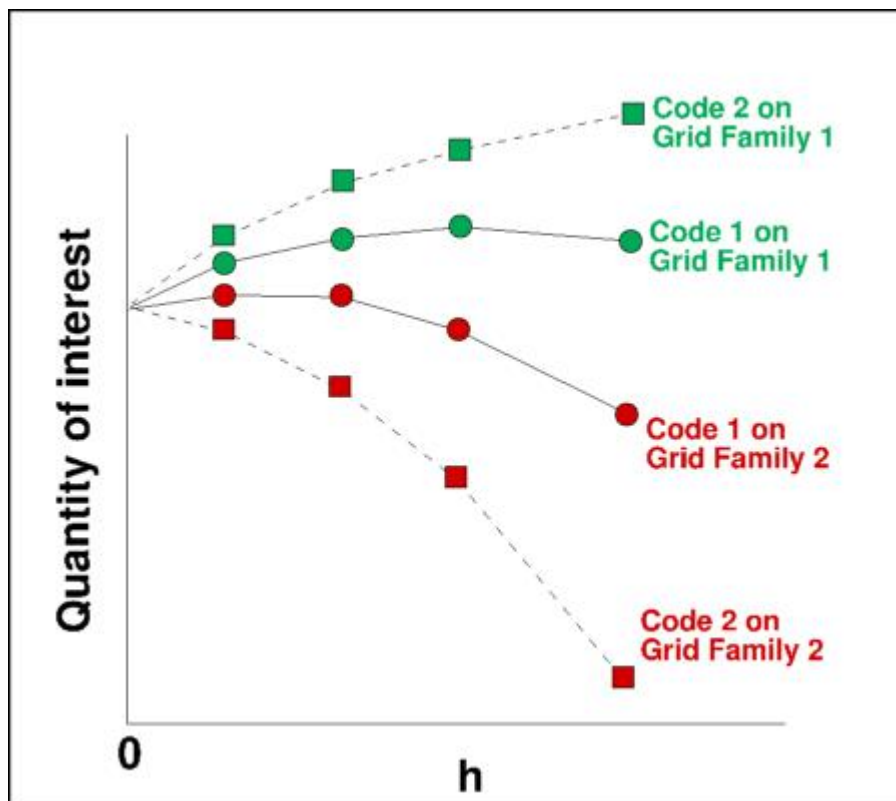
10. A note on "verification." The type of verification used for this site (verification by code-to-code comparisons along with grid-convergence study) is relatively weak, but it can at least indicate if there is a likely problem in your turbulence model implementation. A better method of verification is Method of Manufactured Solutions (MMS), briefly talked about on the [Information from Lisbon "Workshop on CFD Uncertainty Analysis" series](#) page. You can also find information in the open literature.

In any case, a properly-conducted grid-convergence study is crucial to drawing the right conclusions. The following aspects are very important. First of all, note that:

- Even if coded correctly, different codes/schemes may (and likely will) yield different results on the same grid. But on finer and finer grids of the same family, they will eventually approach the same result.
- The same code/scheme run on different grid families may yield different results for the same number of degrees of freedom (same grid sizes), but as each family is refined they will eventually approach the same result.
- If the grid is too coarse such that it does not lie in the "asymptotic range of grid convergence," then the solution may not approach the true (infinite-grid) solution with any order property. In other words, the solution may get worse before it (eventually) gets better. It can be difficult to tell if a given grid is fine enough to lie in the asymptotic range.

These concepts are illustrated in the sketch below. Here, two different codes/schemes are run on two different grid families. When plotting the quantity of interest as a function of some measure of the

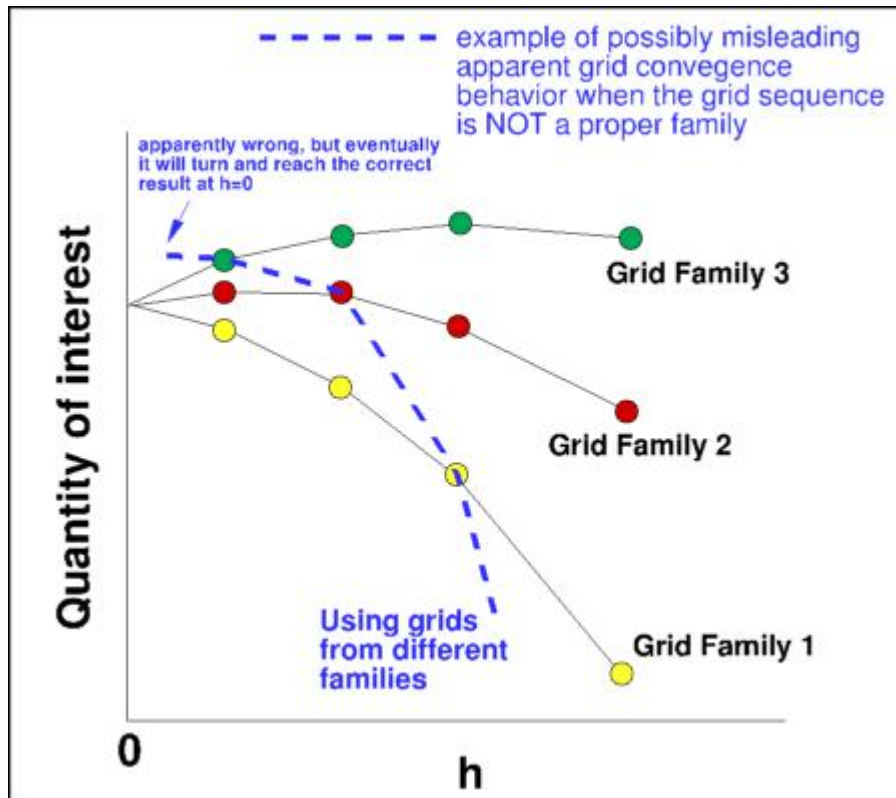
mean grid spacing (h), you can see that there is only one correct answer on an infinite grid ($h=0$). But both codes and both grid families approach the infinite-grid result differently. Even on the finest grids illustrated here (left-most symbols), there is a difference between results for the same code on different grids, and there is a difference between results for different codes on the same grid. One can never reach $h=0$, so the goal is typically to achieve a level of grid refinement that makes the error reasonably small.



Note that some level of "intelligent grid design" is important. A grid with horribly misplaced points may yield very poor results even with h very small.

It is crucial to understand what is meant by a "grid family" when performing global grid refinement. **In a proper grid family, each successively coarser grid is a subset of the next finer one, with every grid cell size changing consistently in all directions.** A proper grid family can be made with structured grids very easily, by making the finest possible grid, then removing every other point in each coordinate direction for the next coarser level. (This is done for most of the verification cases on this site.) It is also possible to create a grid family with other-than-factor-of-2-in-each-coordinate-direction

variation, but this takes more work. It is much more difficult to create a proper grid family for unstructured grids (see, for example, the [VERIF/2DMEA](#) case). A proper grid family will ALWAYS have all of its grid cell spacings in every region of the flow continually and consistently decreasing with each successively finer grid. When one makes use of grids that are NOT of the same family, it may be difficult to tell where the solution is heading with grid refinement. This concept is notionally illustrated in the following sketch:



Here, the solid lines indicate CFD results from a given code/scheme using 3 different grid families. They all approach the same infinite-grid result, as expected. On the other hand, the dashed line indicates CFD results on finer and finer grids that are NOT of the same family. As can be seen, without a proper grid family, it is possible to be misled regarding what the infinite-grid solution should be, particularly if the finest grid used is not close enough to $h=0$. This is because it is not possible to establish an order property or have a true asymptotic range in this case. Note: if you create a sequence of grids all with the same fixed minimum spacing at the wall, then the sequence is NOT of the same family. In fact, if ANY region of the grid does not refine uniformly and multidirectionally with successively finer versions, then it is not a proper grid family.

Naturally, some regions of the flow may benefit from grid refinement more than others; but it is a bad idea to try to guess. Creating a grid family avoids the guesswork, although it can be expensive (especially for 3-D flows). Adaptive mesh refinement (not covered here) is a technique that automatically and accurately determines how/where grid refinement is needed, and therefore has the capability to approach the infinite-grid solution with fewer grid points than with global grid refinement.

This discussion did not go into order-of-accuracy analysis, which can also be an important component of a grid-refinement study. There is much in the open literature on this. See, for example, Roy et al., *Journal of Aircraft* Vol. 55 No. 4, 2018, pp. 1338-1351, <https://doi.org/10.2514/1.C034856> or Oberkampf, W. L., and Roy, C. J., Verification and Validation in Scientific Computing, Cambridge Univ. Press, Cambridge, England, U.K., 2010.