Chapter Five: Best Practices of the CFD and Turbulence Trade

"The result was very happy." Osborne Reynolds upon the initial formulation of his dimensionless number, 1883.

"*Given the erratic track record of most turbulence models, new ideas are always welcome." David Wilcox, 2006.*

It was said in the 1970s in regard to finances that when "E. F. Hutton talks, people listen". Said in an overtly-enthusiastic fashion, "If CFD calculations are done correctly, then nature listens!" The correct fluid dynamics equations, coded correctly, and applied correctly, will certainly mimic nature, and experimental data will inevitably follow pre-test calculations. And under such careful modeling approach, on the uncommon instance when computational and experimental output do not match, there is a strong probability that the source is experimental, such as faulty pressure gauges, incorrect experimental procedures, and so forth. The sections that follow endeavor to provide guidelines that increase the likelihood that computational output will accurately reflect system behavior and its experimental data. But the converse is also true, and too common: when CFD is done incorrectly, without checks and balances, the GIGO acronym becomes valid—garbage in, garbage out. CFD is not to be treated as a black box. In this chapter, numerous guidelines and rules-of-thumb are provided, with the goal of increasing the computational accuracy of the simulations.

5.1 Developing a Bullet-Proof Mesh

WLOG, consider a 3D system. The mesh represents the system geometry that has been parsed (divided) into computational elements or nodes, with each reflecting the system behavior of the primitive variables and a numerous set of derived computational quantities. Each finite element or finite volume represents (reflects, maps) a small region of space for the system in question. Each element is in turn comprised of discrete computational points, or nodes, that have no volume. But, in their agglomeration, the computational nodes somewhat abstractly represent the physical behavior of a small volume where conservation of mass, momentum, and energy are applied. In the cumulative sense, the entire set of discrete, volumeless computational nodes represent the entire contiguous volume. In this virtual world, each computational node simultaneously behaves as if it were a thermocouple for recording temperature, as a pressure transducer for recording pressure, as a flow a meter to obtain mass flow rates and fluid velocity distribution, and so forth.

Without doubt, anyone can develop a mesh. But how is a *quality* mesh developed, and how can it be defined and measured? Moreover, what is "mesh quality", and is it not subjective? In the words of P. M. Knupp [Knupp, 2007],

"Mesh quality concerns the characteristics of a mesh that permit a particular numerical PDE simulation to be efficiently performed, with fidelity to the underlying physics, and with the accuracy required for the problem."

Many mesh metrics are available to quantify and control the quality of a computational mesh, and are summarized in Table 5.1. The table describes over two dozen mesh metrics for hexahedral elements, including their definition and "acceptable range". A metric's "acceptable range" is considered as a reasonable "rule-of-thumb" that generally ensures that a mesh will provide defensible output (or at least, not provide additional, unacceptable errors). However, it is emphasized that a single mesh metric is akin to a reasonable rule-of-thumb, and does not necessarily, in of itself, guarantee that the mesh will be "bullet proof". *In fact, the shrewd analysist will demonstrate that the mesh in question satisfies a set of reasonably-independent mesh metrics, and not just a single metric.* That is, multiple mesh aspects must be tested, and improved as necessary. For example, having a good aspect ratio says nothing about element angle, and vice-versa. Thus, it is possible for a mesh to simultaneously have a great aspect ratio and a poor skew angle, and so forth. In fact, skew is "insensitive to length or aspect ratios", having sole dependence on the element's angles that are formed by the element faces [Knupp, 2003]. Thus, no *single* (unweighted) mesh metric can capture all the geometric issues associated with elements, because element geometry involves several exclusive parameters such as length, angle, etc. This point is showcased in Figure 5.1, which zooms into an airfoil region, showing both skew and aspect ratio. In this situation, regions with good aspect ratios have skew magnitudes that are approaching the high limit, and regions with higher skew and aspect ratio are not always at the same location. Figure 5.2 shows an unzoomed region of the airfoils colored with aspect ratio, with the highest magnitudes occurring where the geometry has the sharpest changes. This situation is typical, as regions with the sharpest geometrical changes tend to have the worst mesh metrics. Thus, those are regions where the analyst should focus attention, as well as the boundaries, as will be discussed later.

Table 5.1. Mesh metric definition and acceptable range for *hexahedral* elements.

Metric	**Definition**	**Acceptable Range (units)**	**References**
Angle (Minimum)	The smallest element angle formed by the intersecting planes (dihedral angle). Too small of an angle increases numerical stiffness.	45-90 (degrees)	[Stimpson et al., 2007; Brewer and Marcum, 2008; Zigh and Solis, 2013]
Angle (Maximum)	The largest element angle formed by the intersecting planes (dihedral angle). Too large of an angle increases numerical error.	90-135 (degrees)	[Stimpson et al., 2007; Brewer and Marcum, 2008; Zigh and Solis, 2013]

Aspect Ratio	The ratio of maximum vs. minimum edge length. The aspect ratio seeks to ensure that quantities such as momentum and heat are transferred appropriately throughout the system in question.	1-5 (unitless); can reach up to 10 if gradient is small (e.g., longer length parallel to flow direction, and smaller length perpendicular to the wall)	[Robinson, 1987; Andersson et al., 2012; Cubit 2017]
Condition Number	Jacobian matrix condition number based on the maximum value of the four element corners.	1-4 (unitless)	[Knupp, 2000; Stimpson et al., 2007; Cubit 2017]
Distortion	The minimum of the Jacobian determinant times the ratio of the local (transformed) and global (actual) areas. This represents the element surface's deviation from a square.	0.4-1.0 (length squared)	[SDRC, 1988; Lawry, 2000; Stimpson et al., 2007; Cubit 2017]
Element Area	Jacobian determinant magnitude at the element's center.	None (length squared)	[Robinson, 1987]
Element Volume	Jacobian determinant magnitude at the element's center.	None (length cubed)	[Cubit 2017]
Expansion Ratio	Element growth rate between adjacent elements.	$\leq$ 1.5 (unitless)	[Fluent, 2012; Zigh and Solis, 2013]
Jacobian	The Jacobian matrix relates how the computational variables map linearly onto their spatial location, e.g., the computational nodes. The matrix has geometrical information such as volume, shape, and orientation [Knupp, 2001]. The Jacobian determinant is calculated to gauge the relative stretching of the local spacing in an element. It is also a measure of the orientation of the surface	None (length squared)	[Knupp, 2000; Stimpson et al., 2007; Cubit 2017]

	normals relative to each other. To obtain its relative goodness metric, it is scaled vs. a perfect element; refer to “Scaled Jacobian” in this Table.		
Oddy	Oddy represents the largest metric tensor variation in the four corners.	0.0-0.5 (none)	[Stimpson et al., 2007]
Orthogonal Quality	The normalized dot product minimum of the element area vector and the centroid vector based on the element’s face or that of the adjacent element.	0.15-1.0 (unitless). A value approaching 0 is unacceptable.	[Fluent, 2018]
Quality Index	A code-defined approach to factor the relative impact of a number of mesh metrics into a single metric (e.g., HyperMesh uses 12 different mesh metrics with user-defined weight factors).	The acceptable range is classified as “ideal” and “good”, depending on the user-defined weight factors. Suspicious elements are flagged as “warn”, while bad elements are tagged as “fail” and “worst”.	[HyperMesh, 2018]
Relative Size	J is the weighted Jacobian matrix determinant. Then, the relative size represents the minimum of J and its inverse, $1/J$.	0.3-1.0 (unitless)	[Knupp, 2003; Cubit 2017]
Scaled Jacobian	The Jacobian minimum is scaled by dividing it by the lengths of two element-edge vectors. It is used in many CFD codes when inverting system matrixes. If the scaled Jacobian is less than 0.5, the calculation may abort; 1.0 refers to a cube, and a value approaching zero implies a highly-distorted (and undesirable element). Negative Jacobians refer to	0.3-1.0 (unitless)	[Knupp, 2000; Stimpson et al., 2007; Cubit 2017]

	inverted, concave, or bowed elements, and should be avoided at all cost. See “Jacobian” in this Table.		
Shape	2 divided by the magnitude of the condition number of the weighted Jacobian matrix.	0.3-1.0 (unitless)	[Knupp, 2003; Stimpson et al., 2007; Cubit 2017]
Shape and Size	The product of the shape and the relative size.	0.2-1.0 (unitless)	[Knupp, 2003; Stimpson et al., 2007; Cubit 2017]
Shear	2 divided by the magnitude of the condition number of the Jacobian skew matrix.	0.3-1.0 (unitless)	[Knupp, 2003; Stimpson et al., 2007; Cubit 2017]
Shear and Size	The product of the shear and the relative size.	0.2-1.0 (unitless)	[Knupp, 2003; Stimpson et al., 2007; Cubit 2017]
Skew	The maximum of $\|\cos \alpha\|$, where α represents the angle between the edges at the element’s center. A perfect element with 90° has cos(90°)=0, while an element with 60° has a value of 0.5. Thus, the smaller the skew, the better.	< 0.5 (unitless)	[Robinson, 1987; Knupp, 2003; Stimpson et al., 2007; Cubit 2017]
Skewness	Skewness compares the shape difference between a given element and that of a perfect hexahedral of the same volume. The larger skewness is, the larger the numerical error and the potential for instabilities.	<0.9 (unitless); average mesh value should approach < 1/3	[Fluent, 2009, Fluent, 2012; Andersson et al., 2012]
Squish Index	Calculates how much the faces of an element diverge from an ideal, orthogonal face.	< 0.9 (unitless)	[Fluent, 2009]
Stretch	$\sqrt{2}$ times the ratio of the element’s minimum edge	0.25-1.0 (unitless)	[FIDAP, 1999; Stimpson et al.,

	length and the maximum diagonal length.		2007; Cubit 2017]
Taper	Maximum ratio of element lengths at opposite sides.	0.0-0.7 (unitless)	[Robinson, 1987; Stimpson et al., 2007; Cubit 2017]
Warp	Cosine of the smallest dihedral angle. That is, this represents the angle formed by the element planes that intersect diagonally.	0.9-1.0 (unitless)	[Cubit, 2018]
Warpage	1 – the cosine of the smallest dihedral angle. That is, the angle formed by the element planes that intersect diagonally.	0.0-0.7 (unitless)	[Stimpson et al., 2007]

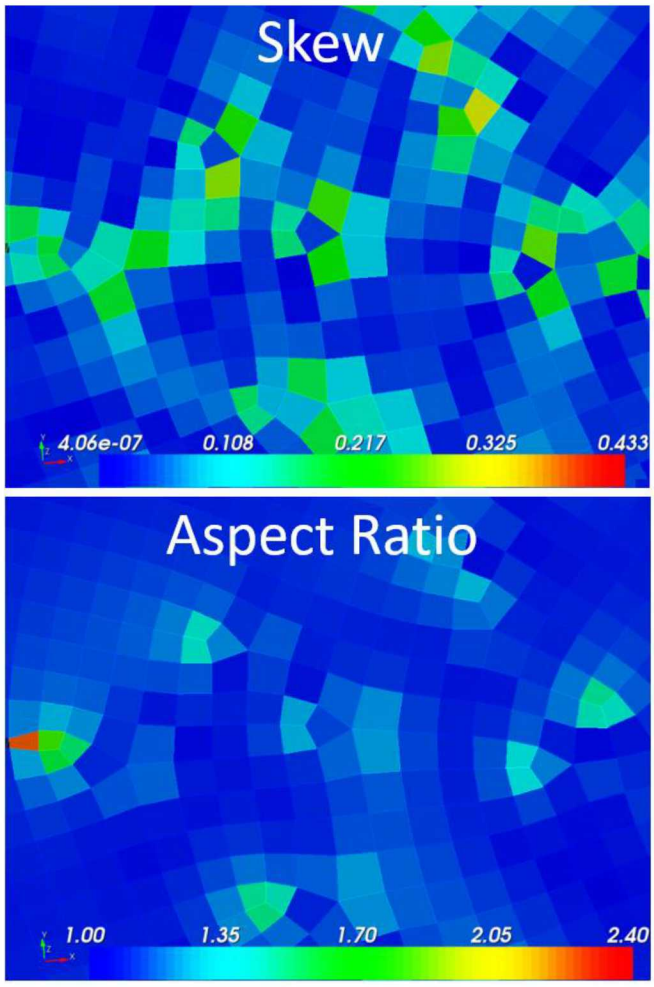

Figure 5.1. Skew vs. aspect ratio over same region.

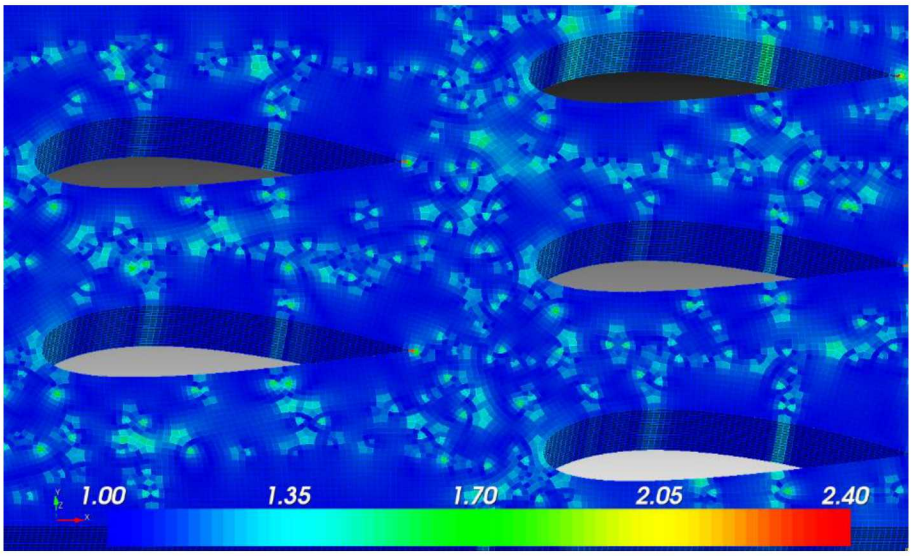

Figure 5.2. Overlay of mesh colored with aspect ratio.

As of 2018, no unanimous consensus exists as to what qualifies as a sufficient set of mesh metrics (but the literature does express some ideas). To make matters worse, is also clear that many mesh metrics are not independent of each other.

The first modern CFD meshes were boldly generated, without much regard for mesh metrics; who knew at the time that there would be mesh issues? However, it was quickly discovered that mesh controls were required to generate defensible output. The first attempts involved a single metric, and in particular, the aspect ratio [Robinson, 1986]. By the mid '80s, some researchers recommended aspect ratio, skew, and taper.

Other metrics may include distortion, which is the minimum of the Jacobian determinant times the ratio of the local (transformed) and global (actual) areas. Said in more geometric terms, this represents the element surface's deviation from a square. Some software packages offer a combination of quality metrics that are weighted and factored into a single metric, such as HyperMesh's "quality index". This software uses a combination of 12 different mesh metrics, including aspect ratio, skew, Jacobian, warpage, and angle [HyperMesh, 2018]. On the other hand, Fluent recommends skewness, aspect ratio, and squish (a measure for how much the faces diverge from an ideal, orthogonal face) [Fluent, 2009]. This brief mesh metrics overview shows a commonality for certain mesh metrics, such as aspect ratio and skew, as well as a commonality in geometric parameters that should be qualified. In effect, by the start of the 21st century, the general notion was, and continues to be, that mesh metrics show that elements are not unduly [Fluent, 2009; Fluent, 2012]

- stretched (length issues),
- distorted (angle issues), and
- transitioned (distance issues between adjacent elements).

For these reasons, ideal mesh metric sets (or equivalent) tend to include the aspect ratio, skew, expansion ratio, and the scaled Jacobian, because this set considers length ratios, element angles, distance between adjacent elements, and computational variable mapping onto node distribution, respectively.

For these reasons, ideal mesh metric sets (or equivalent) tend to include

- aspect ratio (considers length ratios),
- skew (factors in element angles),
- expansion ratio (gauges distance between adjacent elements), and
- scaled Jacobian (a measure of the computational variable mapping onto node distribution).

As shown in Table 5.1, there is a bewildering number of mesh metrics, whose intent is to provide guidelines for the generation of quality meshes. Thus, Table 5.1 is not meant to intimidate users, but is instead intended to offer many options, especially because CFD tools tend to be associate with diverse metrics. For illustration purposes, Table 5.2 shows various mesh metric output for a compact heat exchanger with airfoil surfaces. Notice that a more concise mathematical expression for mesh metrics can be pinpointed if the average, minimum, maximum, and standard deviations are calculated by the meshing package. For convenience, Table 5.3 lists key mesh and flow metrics that should be estimated by the analyst *prior* to beginning CFD analysis.

Table 5.2. Cubit output showing mesh metrics for a compact heat exchanger with airfoil surfaces.

Mesh Metric	**Average**	**Standard Deviation**	**Minimum (Element Number)**	**Maximum (Element Number)**
Aspect Ratio	1.093E+00	9.885E-02	1.000E+00 (95714)	2.405E+00 (51721)
Skew	5.867E-02	5.856E-02	4.058E-07 (553135)	4.335E-01 (368)
Taper	4.620E-02	6.404E-02	3.399E-06 (497420)	4.025E-01 (22026)
Element Volume	2.081E-13	3.856E-14	5.344E-14 (107610)	5.336E-13 (10085)
Stretch	9.228E-01	6.818E-02	3.568E-01 (55611)	9.973E-01 (66323)
Diagonal Ratio	9.626E-01	3.645E-02	6.949E-01 (56257)	1.000E+00 (664913)
Dimension	3.394E-05	2.307E-06	1.892E-05 (163499)	4.389E-05 (10085)
Condition Number	1.024E+00	5.678E-02	1.000E+00 (513435)	1.765E+00 (502167)
Jacobian	1.965E-13	4.031E-14	3.543E-14 (107610)	5.141E-13 (10085)

Scaled Jacobian	9.837E-01	3.562E-02	6.910E-01 (742)	1.000E+00 (384691)
Shear	9.837E-01	3.562E-02	6.910E-01 (742)	1.000E+00 (384691)
Shape	9.793E-01	4.231E-02	6.212E-01 (107610)	1.000E+00 (513435)
Relative Size	8.062E-01	1.898E-01	6.596E-02 (107610)	1.000E+00 (92062)
Shear and Size	7.974E-01	1.976E-01	5.785E-02 (107476)	9.999E-01 (92062)
Shape and Size	7.957E-01	2.004E-01	4.097E-02 (107610)	9.998E-01 (45726)
Distortion	9.392E-01	8.792E-02	4.862E-01 (75966)	1.000E+00 (50308)

Table 5.3. Tabulation of Key Mesh Metrics for Laminar and Turbulent Flows.

Parameter	**Coarse (0.5x)**	**Medium (x)**	**Fine (2x)**	**Very Fine (4x)**
Number of computational nodes				
Average aspect ratio (Desired range: 1.0 to 5.0)				
Maximum aspect ratio (Desired range: 1.0 to 5.0)				
Average skew (Desired range: 0.0 to 0.5)				
Maximum skew (Desired range: 0.0 to 0.5)				
Average node spacing (m) (Desired range: Taylor eddy size < node spacing < integral eddy size)				
Average condition number (Desired range: 1.0 to 4.0)				

Re =
If turbulent:
Kolmogorov eddy size =
Taylor eddy size =
Integral eddy size =
At y^+ = 7, *y* =
At y^+ = 30, *y* =
First computation node at *y* =

While no mesh metric could ever be “perfect” or “universal”, mesh metric guidelines should be viewed as more than “rules of thumb”. The metrics generally have a strong mathematical basis founded upon computational principles [Knupp, 2003], and as such,

are intended to increase confidence in the computational output. Certainly, some meshes will produce reasonable results even when metrics are ignored, but more often than not, ignoring the guidelines will result in poor output. Some of the most curious results that have been pinned down directly to poor meshes include stainless steel that ignited at 400 K, flows exceeding the speed of light, regions with no flow that suddenly accelerate, levitating flows that only Houdini could explain, temperatures below absolute zero, negative densities, and countless other nonsense. If this discussion has not yet generated a healthy dose of caution when developing meshes, then consider Figure 5.3, which shows how drastically solutions can diverge if mesh metric guidelines are not followed. In this situation, flow around a cylinder is considered, whereby the cylinder has D = 0.1016 m, U_∞ = 2.44 m/s, and ν = 4.95x10^{-4} m^2/s. The fluid flows from left to right, at Re=500. Two calculations were run, with everything being the same, except that the mesh on the RHS was developed with good mesh metrics (e.g., aspect ratio ≤ 5, skew ≤ 0.5, and an expansion ratio ≤ 1.5), while the mesh on the LHS was purposely developed with poor mesh metrics (e.g. aspect ratio ≥ 50, skew ≥ 5, and expansion ratio ≥ 15).

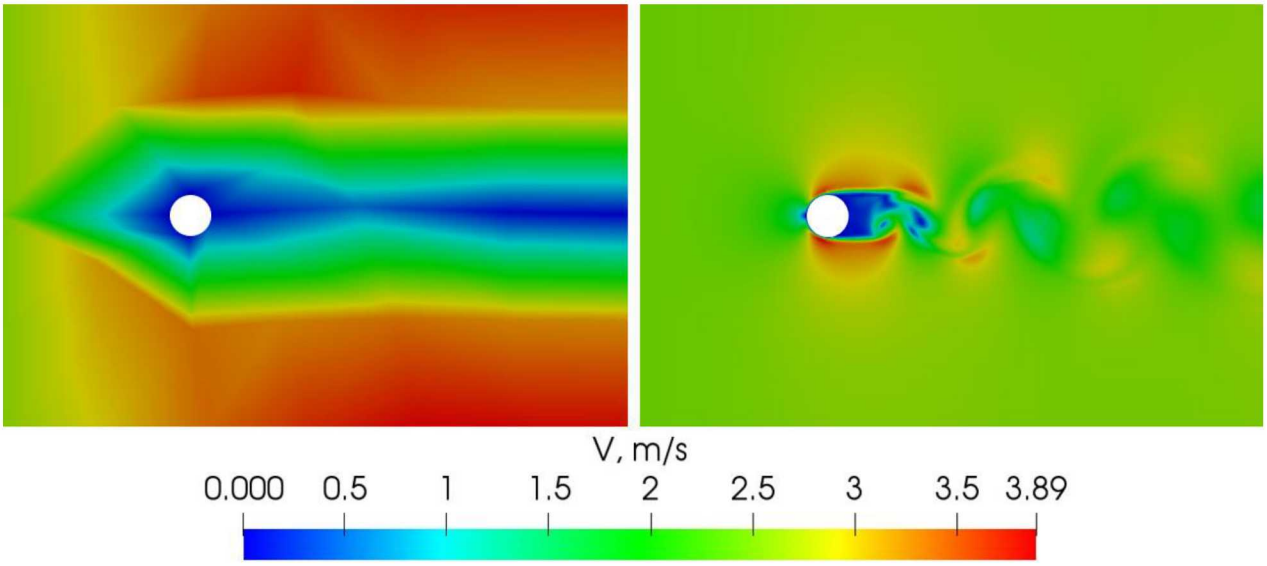

Figure 5.3. Divergent solutions using noncompliant mesh metrics (LHS) vs. compliant (RHS).

5.1.1 Additional mesh guidelines

- Highly distorted elements will inevitably produce highly distorted output.
- Regions with large gradients should have a higher element density (e.g., the boundary layer near the wall, free surfaces).
- The element face should be as close to perpendicular to the wall. That is, this is one of the worst regions to have skewed elements. For this reason, tetrahedral elements are not recommended at the wall.
- Similarly to the criteria above, the element faces should be close to perpendicular to the main flow direction.

- The longer side of an element should be oriented in the direction of the flow, with the smaller side perpendicular to the large velocity gradient (e.g. a boundary layer flow).
- Numerical error is only compounded when elements have multiple, independent poor mesh metrics (e.g., an element having both poor aspect ratio and large skew).
- Always ensure that the grid is refined spatially (coarse, medium, and fine meshes). Refer to Chapter 6 for guidelines.
- The usage of Richardson extrapolation greatly increases modeling confidence.

5.1.2 Computational node spacing for RANS models

The spatial distribution of computational nodes in RANS models is crucial for generating defensible solutions. This is especially so for phenomena-rich calculations involving complex flows, such as strong wall shear, swirl, rotational surfaces, backflows, turbomachinery, drag, and lift.

If no wall function is used, then it is important that CFD meshes have the first computational node at $y^+ = 1$ [Fluent, 2012]. Generally speaking, this is the case for most RANS models, but there are exceptions. For example, Wilcox recommended that the first computational node be placed at $y^+ = 5$ if using his 2006 k-ω turbulence model [Wilcox, 2006]. For turbulent, low-*Re* flows, the first node can be as far as $y^+ = 4$ [Andersson et al., 2012]. Mesh biasing can be used to reduce the number of computational nodes, as shown in Figure 5.4. If biasing is used, it is crucial that the growth between nodes not exceed 1.5. But, just how large should the node spacing be, especially near the wall? Certainly, because RANS is not concerned with individual eddy behavior (except for Myong-Kasagi RANS and the like), then it is not necessary that node spacing be less than or equal to the size of the Taylor and Kolmogorov eddies. On the other hand, including an additional computational node at $y^+ \leq 30$ will help capture the complex turbulence motions occurring in the buffer layer, as explained in Section 3.6. Conservative researchers recommend as many as 5 to 10 computational nodes in the region bounded by $y^+ < 20$ [Andersson et al., 2012]. This is certainly consistent with the use of node biasing with a growth factor that does not exceed 1.5.

So, the above guidelines specify a minimum spacing limit. What should the maximum spacing limit be? The maximum nodal distance should be decreed by the node spacing needed to achieve sufficient spatial discretization. For example, if cutting the node size in half reduces the computational error by less than some reasonable metric acceptable to the analyst (say < 2%), then the current maximum node spacing is satisfactory.

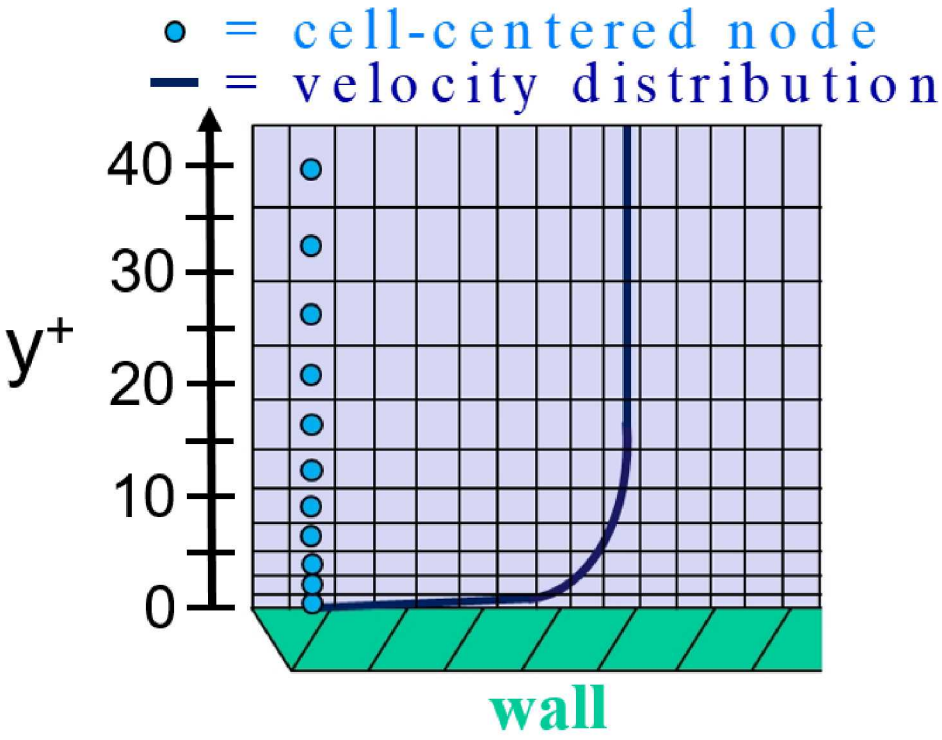

Figure 5.4. Biased node distribution based on y^+ location.

As noted in Section 3.6, different flow physics, with critical impact on the flow, occur in very narrow regions referred as the viscous sublayer and the buffer sublayer. For example, drag and lift are significant at and near the wall. Therefore, being able to model the viscous sublayer in the range of $0 \le y^+ \le 5$ (to as high as 8) is of outmost importance. As also noted in Section 3.6, crucial turbulence phenomena occur in the buffer layer, which is at approximately in the range of $5 \le y^+ \le 30$. This is where lots of nonisotropic effects occur, where eddy production and decay predominate, and where sizable eddy fluctuations occur. *Therefore, it is not surprising that calculations that fail to include at least one computational node in the viscous sublayer and at least two in the buffer layer generally fail to deliver*, especially if complex flows are involved. Certainly, following these guidelines will result in a large mesh that will require much computational time. Therefore, the analyst in encouraged to run coarse meshes to at least get some preliminary results, and submit the fine meshes as early as possible.

5.1.3 Wall functions

Wall functions are used to more accurately calculate certain behaviors at the wall, such as shear stress, wall friction, wall heat flux, wall temperature, and so forth. And for some turbulence models, it serves as the only way of ensuring that the velocity near the wall is calculated adequately (e.g., SKE). Wall functions can also serve as a means to decrease the number of computational nodes. This practice is common when the SKE is used in low-*Re* flows. Wall functions can be included in LES and DES calculations as a means of reducing node count, but their usage is counter to the LES modeling philosophy, as

discussed in Section 5.4.1. By contrast, a wall function is not required for the 2006 k-ω model; indeed, its usage will likely corrupt the calculation.

If a wall function is present, then the first computational node can be at $y^+ < 5$ (instead of 1) if a linear wall function is used. If a log wall function is used, then the first computational node can be at $30 < y^+ < 500$ [Zigh and Solis, 2013]. Of course, wall functions will not work if the flow is detached (separated). In such cases, SKE should be avoided, and the 2003 SST, 2006 k-ω, LES, or DNS can be used.

5.1.4 Computational domain size

Another critical aspect of simulations is to determine a computational space that is sufficiently large. If the computational domain is too small, important phenomena will be missed, and if the domain is too large, the calculation will be needlessly slow.

It is important in CFD calculations that the mesh is sufficiently large to capture recirculation/entrainment/secondary flow. For illustration purposes, the top of Figure 5.5 shows a computational region for an expansion where the recirculation pattern is truncated, while the bottom section shows a domain that is sufficiently large, such that it captures the recirculation pattern. Figure 5.6 shows a similar situation for a jet model. In particular, the LHS shows conceptually what happens if the mesh is not wide enough to capture the jet as it spreads, while the RHS shows a more satisfactory computational domain.

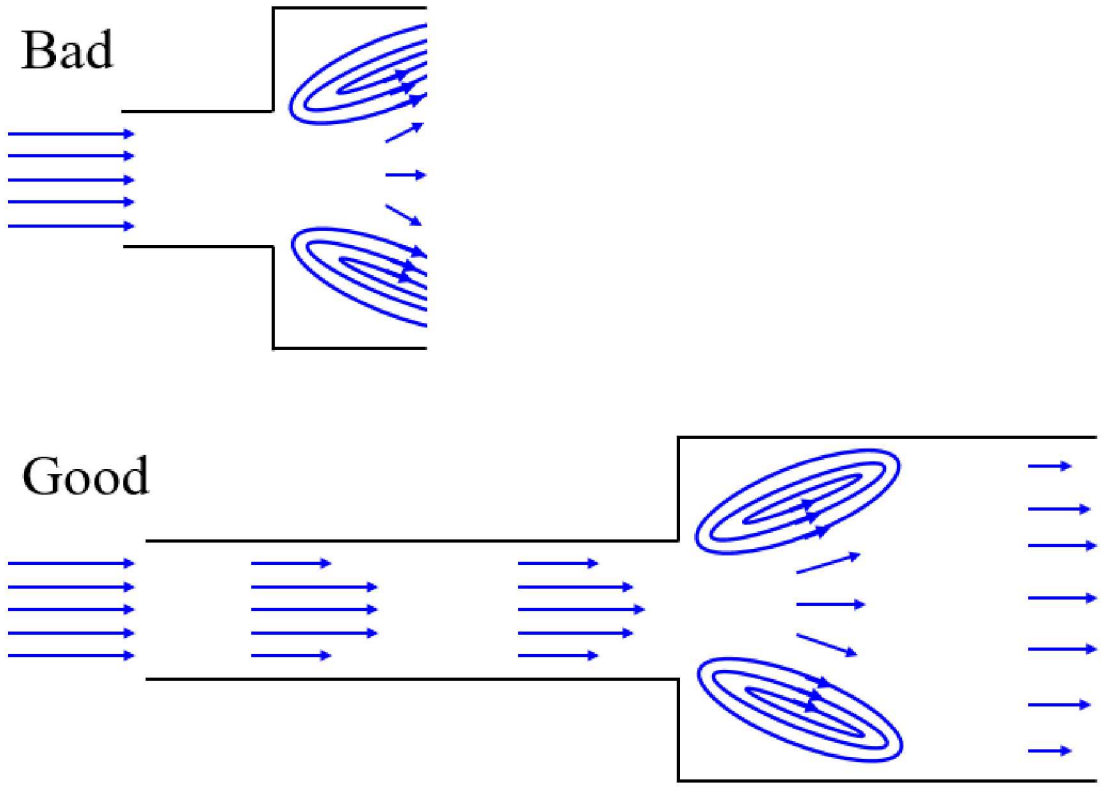

Figure 5.5. Good and bad mesh domains for a flow contraction.

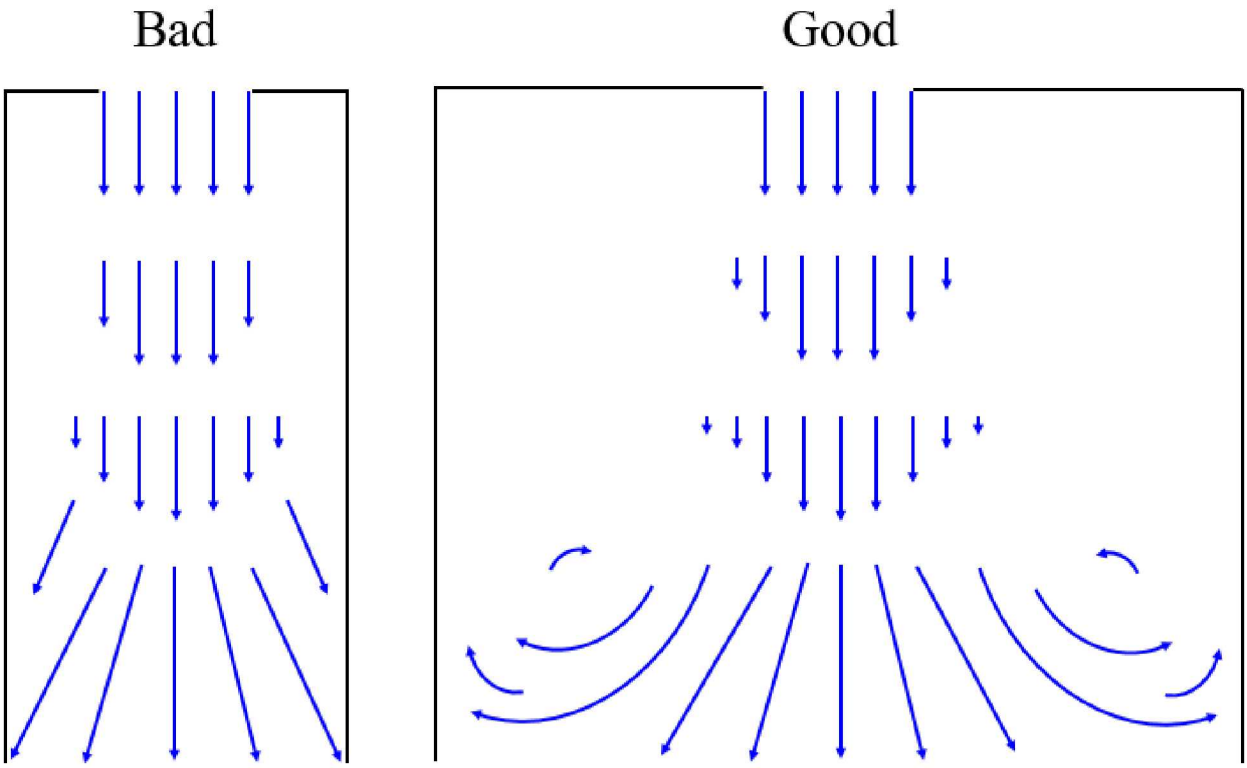

Figure 5.6. Good and bad mesh domains for a jet.

Generally, a visualization tool will rather quickly and intuitively show whether a mesh has an adequate domain or not. Fortunately, there are specific guidelines that can serve as general guiding principles, without having to submit the calculation first [Tutar and Holdo, 2001]. Some rules-of-thumb for flow around a body with hydraulic diameter D_h are as follows [Tutar and Holdo, 2001],

- Distance between the inflow boundary and the body centerline = $7D_h$,
- Distance between the body centerline and the perpendicular side boundaries = $7D_h$, and
- Distance between the body centerline and the flow exit boundary = $15D_h$.

Similar guidelines can be found elsewhere for other systems [Leonardi and Castro, 2010], implying a universality.

Finally, upon running a calculation, the P and u gradients normal to the BC should be relatively small. If not, then it will be necessary to further extend the mesh.

5.2 Boundary and Initial Condition Recommendations

Another key to successful CFD modeling includes the proper selection of boundary conditions (BCs) and initial conditions (ICs). For instance, though a simulation with an incorrect BC may run to completion, it will nevertheless yield an incorrect solution. Recall that BCs provide an infinite family of solution curves, but only one specific BC will provide the correct solution for a specific system. In other words, BCs determine unique solutions

for the ordinary and partial differential equations (ODEs and PDEs) found in conservation equations, such as mass, momentum, and energy. Under ideal conditions, a bad BC will immediately cause a code abort, and thereby raise red flags that something is wrong. But, under many situations (Murphy's Law), the code will be happy to proceed in the generation of useless data. Besides generating bad output, incompatible BCs will result in other unintended consequences, including solution instabilities and significantly-longer computational times. Thus, extreme caution is desirable when selecting BCs.

For mature CFD codes, one of the most significant sources of boundary errors arises from the user via poor meshes and/or BC choices. For these reasons and others, it is incumbent that analysts spend a reasonable amount of time reflecting on BC selection as the input model is developed and tested. Thereafter, the analyst could consider running additional simulations with alternative BCs, to explore the system behavior.

5.2.1 Boundary conditions

Boundary conditions (BCs) enable unique solutions to be obtained from ODEs and PDEs. BCs can also be viewed as solution drivers because they force the calculation into a unique solution. The number of required BCs for ODEs and PDEs is based on the largest order for each spatial coordinate being solved. For example, a second order ODE in 3D space will require two BCs for each of the three spatial coordinates, for a total of six. Note that BCs are associated with the system behavior at the spatial boundaries, that is, at the system edges, while ICs are specified for the entire domain.

Example 5.1. Consider the following ODE, $\frac{d^2T}{dx^2} + \cos(3\pi x)\frac{dT}{dx} - 10T = 0$, where $T=T(x)$, and $0 \le x \le L$. How many BCs are needed?

Solution. Two BCs are needed to appropriately solve this system because the highest spatial derivative is of order two and only one space coordinate is considered (x-direction). As an example of such system, one possibility is that it has fixed-temperature conditions, e.g., $T(x=0)$ = 450 K on the LHS and $T(x=L)$ = 600 K on the RHS.

Example 5.2. Consider a system in 3D Cartesian space, whereby $T=T(x,y,z)$, $0 \le x \le L$, $0 \le y \le M$, and $0 \le z \le N$. The PDE is $\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = 0$. How many BCs are needed?

Solution. This system is second order for each of the three space derivatives, and there are three directions. Therefore, a total of six BCs are needed: two in the x-direction, two in the y-direction, and another two to take care of the z-direction.

More formally, BC types for a given spatial domain of length $[a,b]$ are classified as follows:

- Dirichlet (first type). This type associates the primitive variables (e.g., u, v, w, T, P, ρ, etc.) with a fixed value or scalar field. For example, $w(a)$ = 5.5 m/s.
- Neumann (second type). This BC type associates a derivative with zero, meaning that the spatial gradient is zero. For example, if no heat transfer occurs, then

$\frac{dT}{\partial x} = 0$, so the system is adiabatic, which means $T_{wall} = T_{node\ adjacent\ to\ wall}$. In theory, it is impossible to have a perfectly adiabatic boundary, but for engineering purposes, this is a great approximation if the system is reasonably insulated at the boundary.

- Robin (third type). This BC is a superposition (generalization in this case) of the terms associated with a Dirichlet and Neumann BC. For instance, a convective/conductive BC can be represented as $k\frac{dT}{dx} + hT = C_1$.

5.2.2 BC types in CFD codes

There are many types of BCs used in commercial CFD codes. *However, it is up to the analyst to justify the usage of all input model BCs and to perform sensitivity studies to quantify their impact on the solution. This is the case because BC input is one of the top sources of modeling errors. Furthermore, boundaries are one of the worst places to have poor mesh metrics.* For this reason, it is highly recommended that analysts use aspect ratios near 1.0 for elements normal to boundaries. Likewise, having skew less than 0.5 is highly recommended, and more so if skew approaches 0. A selection of major BC types is discussed next.

A symmetry boundary is ideal for use when the flow field has geometric symmetry *and* the flow is symmetric. (Asymmetric flow can occur in situations with symmetric geometry, but with different BCs). Where appropriate, use symmetry to reduce the element count. Symmetry BCs are ideal for DNS analysis and are also suitable in the spanwise direction of boundary layer flows, far away from the wall. For example, a free-stream boundary allows the user to model fluid conditions far away from boundary layer.

Periodic boundaries are ideal for repetitive geometry, such as occurs in the periodic flow lattice of a nuclear reactor core [Rodriguez and Turner, 2012; Joshi and Nayak, 2019].

An inflow (or outflow) BC allows the user to impose velocity or mass flow rate at the inlet (or outlet). The quantity can be fixed or variable, and can be rather complex, involving functions, subroutines, tables, and so forth. Often, inflows and outflows should be normal to the BC surface, but there are exceptions, such as a swirl BC [Rodriguez, 2011].

An open boundary (used to compute inflow or outflow), allows the fluid to exit the domain, without impacting the interior solution (hopefully!). In theory, such is the case, whereby the momentum distribution within the system is not impacted. In practice, these boundaries can generate spurious reflective waves. Therefore, careful inspection should be undertaken to ensure a defensible solution. The reader is encouraged to consult the literature in this specialized research topic, which includes many areas regarding reflectiveless BCs and wave-mitigation techniques [Thompson, 1987; Spalart, 1990; Nordstrom, Nordin, and Henningson, 1999].

Wall BCs are used to define mathematically how the flow is shaped by surfaces. This includes characteristics such as wall roughness, with the “smooth wall” being the general CFD code default. Wall BCs are generally “no slip”, meaning that the fluid is attached to

the wall. Therefore, the fluid velocity at the wall matches the wall velocity, which is normally 0; the same situation applies to temperature *T*. Slip is present in situations with small Knudsen number (*Kn*), 0.01 ≤ *Kn* ≤ 0.3, which is usually in the realm of micro and nano flows. In such cases, $u_{fluid} \neq u_{wall}$ and $T_{fluid} \neq T_{wall}$. Rather, the fluid velocity at the wall is greater than zero when slip occurs! It is thought that slip is present as a result of thin, trapped gas sheets that behave as a lubricant between a liquid and the wall surface roughness [Tabeling, 2009; Bolaños and Vernescu, 2017]. If slip is present, a reasonable wall BC for liquid flow is [Tabeling, 2009]

$$u = L_N \frac{\partial u}{\partial z}$$

where

L_N = Navier length ~ $\frac{\nu}{u_{therm}} \sim \frac{\nu}{u_s}$,

In this context,

ν = *kinematic viscosity of the gas trapped between the liquid and the wall and*

$u_{therm} \sim u_s$, the sound velocity.

Kn is defined as

$$Kn = \frac{\lambda}{\ell_{char}}$$

where

λ = mean free path, which is the average distance travelled by particles between collisions and

ℓ_{char} = characteristic length of the flow channel.

For liquids, λ is approximately on the magnitude of the liquid's molecular size. For water, λ ~ 0.02 nm [Wang, Yang, and Zhao, 2014; Bolaños and Vernescu, 2017]. Recently, researchers have considered the impact of surface roughness on slip [Bolaños and Vernescu, 2017].

Example 5.3. Find the Navier length for water at 30 °C and 1 atmosphere in a cylindrical tree root. How small can the hydraulic diameter be such that slip occurs?

Solution. Water has the following physical properties: u_s = 1,507 m/s and $\nu = 0.801 \times 10^{-6}$ m²/s. Hence, $L_N \sim \frac{\nu}{u_s} = \frac{0.801x10^{-6}\ \mathrm{m^2/s}}{1{,}507\ \mathrm{m/s}} = 5.32x10^{-10}\ \mathrm{m}$. Solving for the smallest hydraulic diameter implies that *Kn* < 0.3. Therefore, $\ell_{char} = \frac{\lambda}{Kn} \sim \frac{0.2x10^{-9}\ \mathrm{m}}{0.3} = 6.67x10^{-10}\ \mathrm{m}$, which is clearly in the nano region.

Example 5.4. Consider a rectangular duct with H = 0.005 m, W = 0.025 m, average water velocity = 0.247 m/s, and Re = 11,000. It is desired to use various RANS models, so k, ε, ω, and ν_t must be calculated for usage as input values. Apply the LIKE algorithm to estimate the values of the four variables.

Solution. $D_H = \dfrac{4FA}{WP} = \dfrac{4HW}{2H+2W} = \dfrac{2HW}{H+W} = \dfrac{2*0.005*0.025}{0.005+0.025} = 0.00833.$

$$\ell = 0.07\mathrm{D_H} = 0.07*0.0083 = 0.000583 \text{ m}$$

$$I = 0.16\mathrm{Re_H^{-1/8}} = 0.16(11{,}000)^{-1/8} = 0.05$$

Now it is straightforward to calculate the input values:

$$k = \frac{3}{2}(\bar{u}I)^2 = \frac{3}{2}(0.247*0.05)^2 = 2.29x10^{-4} \text{ m}^2/\text{s}^2$$

$$C_\mu = 0.09.$$

$$\varepsilon = C_\mu \frac{k^{3/2}}{\ell} = 0.09\frac{\left(2.29x10^{-4}\right)^{3/2}}{0.000583} = 5.35x10^{-4} \text{ m}^2/\text{s}^3$$

$$\omega = \frac{k^{1/2}}{\ell} = \frac{\left(2.29x10^{-4}\right)^{1/2}}{0.000583} = 26.0 \text{ 1/s}$$

$$\nu_t \equiv \frac{\mu_t}{\rho} = \frac{C_\mu k^2}{\varepsilon} \text{ (Prandtl-Kolmogorov)} = \frac{0.09\left(2.29x10^{-4}\right)^2}{5.35x10^{-4}} = 8.82x10^{-6} \text{ m}^2/\text{s}$$

As a “sanity check”,

$$\nu \equiv \frac{\mu}{\rho} = \frac{2.32x10^{-4}}{942} = 2.46x10^{-7} \text{ m}^2/\text{s}$$

$$\frac{\nu_t}{\nu} = \frac{8.82x10^{-6}}{2.46x10^{-7}} = 35.9$$

That ν_t >> ν is consistent with our expectation that the more turbulent the flow is, the larger ν_t will be.

5.2.3 Initial conditions

Initial conditions (ICs) are needed to solve initial value problems that march in time as the solution progresses. Analogous to BCs, the number of required ICs for ODEs and PDEs is based on the maximum order of the time derivative. For example, consider a chunk of ice. Its mass as a function of time can be determined if it is known just how much ice

there was initially—this is an IC (and of course, just how much heat the ice is absorbing, perhaps a BC such as a heat flux or Robin).

The mathematical utility of an IC is that it specifies the initial quantity (value) of a given parameter, usually at time 0. For example, *T(t=0)* = 310 K specifies the temperature distribution at time zero when the transient was initiated for the domain in question.

Often, it is better to set the initial velocities to 0, and just let the code calculate consistent values based on the BCs. Otherwise, there may be inconsistencies between the pressure and velocity fields, as specified by the ICs and BCs, and these can cause the code to run longer before it converges, and may even result in solution divergence! In other words, failure to include ICs that are consistent with the values in the BCs may cause numerical issues.

While not needed for LES or DNS, RANS methods require IC input for various turbulence quantities, depending on the model being used:

- *k* for the *k* PDE (e.g., Prandtl *k*, *k*-ε, *v2-f*, and *k*-ω models),
- ε for the ε PDE (e.g., *SKE*, realizable *k*-ε, RNG *k*-ε, and Myong-Kasagi *k*-ε models),
- ω for the ω PDE (e.g., Kolmogorov *k*-ω, 1988 *k*-ω, 1998 *k*-ω, and 2006 *k*-ω models), and
- ν_t for Prandtl-Kolmogorov closure models (SKE).

k is the turbulent kinetic energy in m^2/s^2, which describes the *velocity scale* of the *large* eddies. ε is the kinetic energy dissipation rate in m^2/s^3, and is a measure of the eddy *length scale*. ω is the eddy specific dissipation frequency in 1/s, and is a measure eddy *time scale*. Finally, ν_t is the turbulent kinematic viscosity in m^2/s, which is a metric for the relative degree of the turbulence, and can be *orders of magnitude larger* than ν. These input parameters have a significant impact near open BCs, but not as much far away from inlets and outlets. The best approach is to obtain *k*, ε, ω, and ν_t from experimental data. But such data is usually not available. The next best approach is to estimate input values using the LIKE algorithm; refer to Sections 3.4 and 3.5. The approach of last resort is to use the CFD code default values. But such values are generalized, and are therefore not always suitable for the specific situation of interest of all analysts.

Example 5.5. Consider the following PDE,

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+\mathrm{v}\frac{\partial u}{\partial x}+w\frac{\partial u}{\partial x}=\frac{\mu}{\rho}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)-\frac{1}{\rho}\frac{\partial P}{\partial x}.$$ How many ICs are required?

Solution. This is the Navier-Stokes equation, with a first order partial derivative for time. Therefore, only one IC is needed.

Example 5.6. Consider the following Navier-Stokes PDE,

$$u\frac{\partial u}{\partial x}+\mathrm{v}\frac{\partial u}{\partial x}+w\frac{\partial u}{\partial x}=\frac{\mu}{\rho}\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)-\frac{1}{\rho}\frac{\partial P}{\partial x}.$$ How many ICs are required?

Solution. This equation is the steady state (SS) version of Navier-Stokes, so the time partial derivative is zero, and therefore no ICs are needed.

Example 5.7. Consider the following PDE,

$\frac{\partial^2 u}{\partial t^2} = c^2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right)$. How many ICs are required?

Solution. This is the wave equation with a second order time derivative. Therefore, two ICs are needed.

5.2.4 Compatible vs. incompatible BCs

When various BCs are used, the following Western classic comes to mind, "The Good, the Bad, and the Ugly". In particular, certain BC combinations for inlets and outlets are generally defensible and reliable, while other are notoriously untrustworthy. This is summarized in Table 5.4.

Table 5.4. Inlet and outlet BC combinations.

The Good…	**The Bad…**	**And the Ugly**
The specification of *u* at the inlet and static *P* at the outlet yields reasonable results. These BCs are highly compatible.	Usage of total *P* at the inlet and static *P* at the outlet can result in numerical instabilities.	The specification of *both* inlet and outlet *u* will result in incorrect velocity distributions. In any case, what is driving the momentum here?
Because mass flow rate is proportional to *u*, the specification of mass flow rate and static *P* at the outlet yields reasonable results for incompressible situations.	---	The usage of total *P* at the inlet, while using an outflow BC at the outlet can result in flow instabilities and incorrect momentum calculation.
P can be specified at the inlet, so long as the code calculates the exit conditions.	---	Usage of mass flow rate at the inlet and an outflow BC at the outlet is not recommended. This is particularly so when ρ is not constant, i.e., when the flow is compressible.

For convenience, the total static pressure is defined as,

$$P_{total} = P_{static} + \frac{\rho u^2}{2}.$$

A set of BC conditions must be self-consistent, and along with the PDEs/ODEs being solved, the system must not form an ill-posed problem [Rempfer, 2006]. This can result when the BC fails to provide additional independent information, such as when n+1 unknowns are solved with n equations. Additionally, BC intersections, such as the interphase where curves and surfaces meet, can be prone to numerical issues if the intersecting BCs are not compatible.

5.3 RANS Modeling Recommendations

If there is a choice between the 1988, 1998, and 2006 k-ω models, the latter is much superior, and is therefore the preferred version. Unfortunately, the coefficients for the Kolmogorov k-ω model are not well documented, so this version is not recommended.

If only k-ε models are available, the Myong-Kasagi is a fairly-recent, promising contender with much potential, but not as much validation as the SKE; yet, because of its promising characteristics (See Section 4.6.3.6), it is highly recommended. On the one hand, the realizable and RNG k-ε models resolve various issues associated with the SKE. However, issues associated with the ε PDE regarding eddy scales continue to impact *all* k-ε models (except Myong-Kasagi, and similar models), as discussed in Section 4.7 and its subsections.

For great, all-around RANS models, the 2006 k-ω, 2003 SST, and Myong-Kasagi k-ε are recommended, while the SKE is not; refer to Chapter 4 for additional details and reasoning behind such choices. The v2-f model is quite useful, but encounters instability issues on occasion. Table 5.5 summarizes some of the pros and cons of various turbulence models.

Table 5.5. Some guidelines towards turbulence model selection.

Turbulence Model	Pros	Cons
Zero-equation (algebraic models; mixing length hypothesis)	Fastest, great for getting analytical solutions from PDEs. (Ex. Prandtl mixing-length model; Baldwin-Lomax is good for turbomachinery/aerospace applications, and attached, thin, boundary layers.) Simplest form of turbulence model, easiest to implement, and very robust numerically.	Very limited applicability and successful mostly for very simple flows. No transport of turbulent scales (e.g., v and ℓ). Do not use near solid boundaries, unless using a damping function (Van Driest), etc.

One-equation (*k*-algebraic model)	Fast. Computes turbulent length scale. Spalart-Allmaras is good for turbomachinery/aerospace applications; Prandtl *k* PDE is useful for fast calculations, test cases.	No transport of the length scale. More limited compared to two-equation models (e.g. k-ε, k-ω).
Two-equation models	Compute velocity turbulence scale, and some other key scale, such as length, etc. Generally provides good results for many flows, with varying degrees of success.	Shear stress diffusion and non-homogeneity are not calculated. Cannot compute eddy dynamics.
Reynolds Stress Model (RSM)	Zero-, one-, and two-equation RANS models assume isotropic viscosity via the Boussinesq approximation. RSM is anisotropic and Boussinesq-less. Great for strong swirl, adverse pressure gradients, and anisotropic turbulence.	RSM uses six PDEs to solve each of the six components in the stress tensor. Theory is great, but successes are limited, as it has many coefficients that require justification. Computationally costly.
Standard k-ε (SKE)	The most widely used model prior to ~2005. Good for isotropic (high *Re*) flows, simple flows, plane and radial jets (but NOT round jets), and plumes.	Poor results for round jets (round jet anomaly), far wakes, strongly-curved surfaces, swirl, flow separation, sudden acceleration, and low-*Re*.
Renormalization group (RNG) k-ε	Improves standard k-ε for low *Re*, separation, and swirling flows with an extra dissipation term.	Not good for round jets and plumes. Not as stable as the standard k-ε.
Realizable k-ε (RKE)	Improves standard k-ε for separated and swirling flows, boundary flows, strong streamline curvature, and round jets. The realizability constraints only yield positive normal stresses. Better than RNG for separated flow and secondary flows. Solves the round jet anomaly.	Not as stable as the SKE.

V2-f	Has the same k-ε PDEs as SKE, but dissipation is different. Models the wall region without using wall or damping functions. Good for strong swirl.	Some misses at low swirl. Notoriously unstable.
2006 k-ω	Great for adverse pressure gradients, separated flows, swirl, and low-Re (no wall functions). Solves the round jet anomaly. Best all-around RANS model.	Requires a fine mesh near the wall, as it does not use wall functions (which is actually a pro). It is recommended that the first node be at $y^+ < 5$.

5.4 LES Modeling

The larger eddies obtain their kinetic energy from the bulk fluid energy, contain most of the turbulent kinetic energy (~80%), transfer kinetic energy to the smaller eddies by stretching them and breaking them up ("cascading"), and are responsible for the majority of the diffusive processes such as mass, momentum, energy, and stress. The larger the eddy, the higher its non-isotropic nature. For these reasons, the simulation of large eddies is highly desirable. On the other hand, the smaller eddies take the kinetic energy from the larger eddies and transfer their energy back to the fluid through viscous shear. For high *Re*, the small-scale turbulent eddies are *statistically isotropic.* Therefore, they are "more universal", independent of the boundary conditions, and the mean flow velocity than the larger eddies. Thus, simulation of the smaller eddies is also desirable.

So, why not simulate (resolve) the larger eddies, and approximate (model) the behavior of the smaller eddies? To this effect, large eddy simulation (LES) models have been developed for several decades [Smagorinsky, 1963; Leonard, 1974; Germano et al., 1991; Kim and Menon, 1995; Nicoud and Ducros, 1999; You and Moin, 2007]. Figure 5.7 conceptually shows that LES resolves integral and Taylor eddies up to a user- or mesh-defined minimum eddy size Δ, while DNS resolves the integral, Taylor, and Kolmogorov eddies (all scales). In this context, the scale Δ determines the minimum size for which eddies will be resolved, thereby acting as a filter for the subgrid scale where eddies smaller than Δ are modeled.

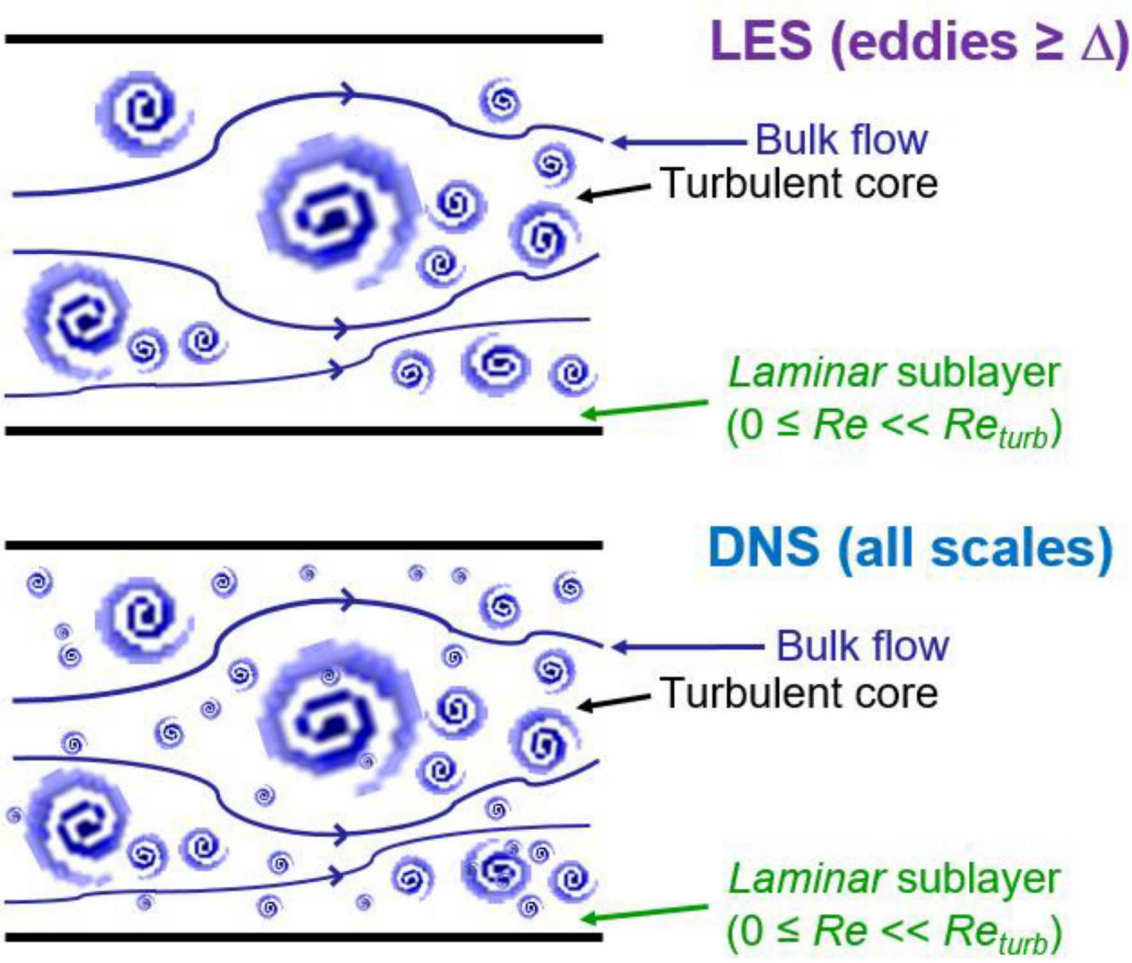

Figure 5.7. LES (top) and DNS (bottom) eddy modeling.

Figure 5.8 shows the instantaneous velocity based on resolved LES and DNS calculations. Note that LES will capture a significant number of velocity fluctuations associated with the larger eddies, while DNS will capture all the LES fluctuations, as well as the Taylor eddies that were cut off from the LES (in the interest of a faster calculation), and the Kolmogorov eddy fluctuations as well. In other words, DNS resolves the entire eddy spectrum: integral, Taylor, and Kolmogorov eddies. Hence, the DNS instantaneous velocity is more "jagged", and of course, synonymous with the fluctuations seen in experimental data. *Note that RANS will not calculate the dynamic eddy behavior, but is instead a representative behavior of "non-dynamic, average" eddy behavior.* As such, the RANS instantaneous velocity as a function of time (and space) can never have the wiggles that are resolved by LES or DNS, as shown in Figure 5.9.

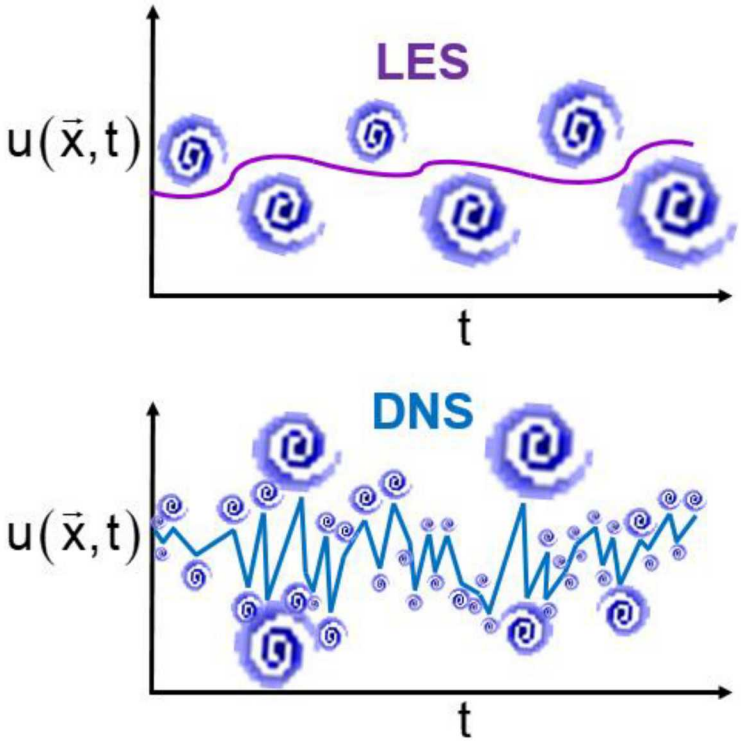

Figure 5.8. LES (top) and DNS (bottom) instantaneous velocities.

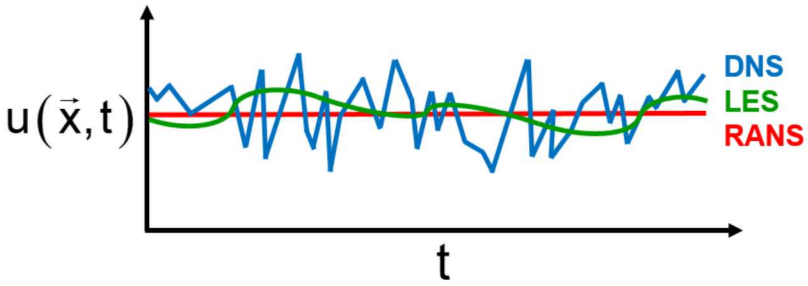

Figure 5.9. A comparison of RANS, LES, and DNS instantaneous velocities.

Finally, Figure 5.10 compares the velocity distribution for a jet at *Re*=3,220, which was simulated using RANS, LES, and DNS. A close inspection shows significant differences and similarities in velocity distribution. The main idea is that although all three provide useful details of the velocity field, RANS is faster than LES, which is faster than DNS; conversely, DNS is more detailed than LES, which is more detailed than RANS. It is also worthy of mention, that theoretically speaking, DNS will calculate *all* turbulence cases, while LES can do so for most situations, whereas RANS is much more limited in applicability. The choice of turbulence model is ultimately based on financial resources, time constraints, computational resources, and the necessary level of output required of the simulation. Said more pragmatically, use the fastest model that gets reasonable accuracy for the system of interest.

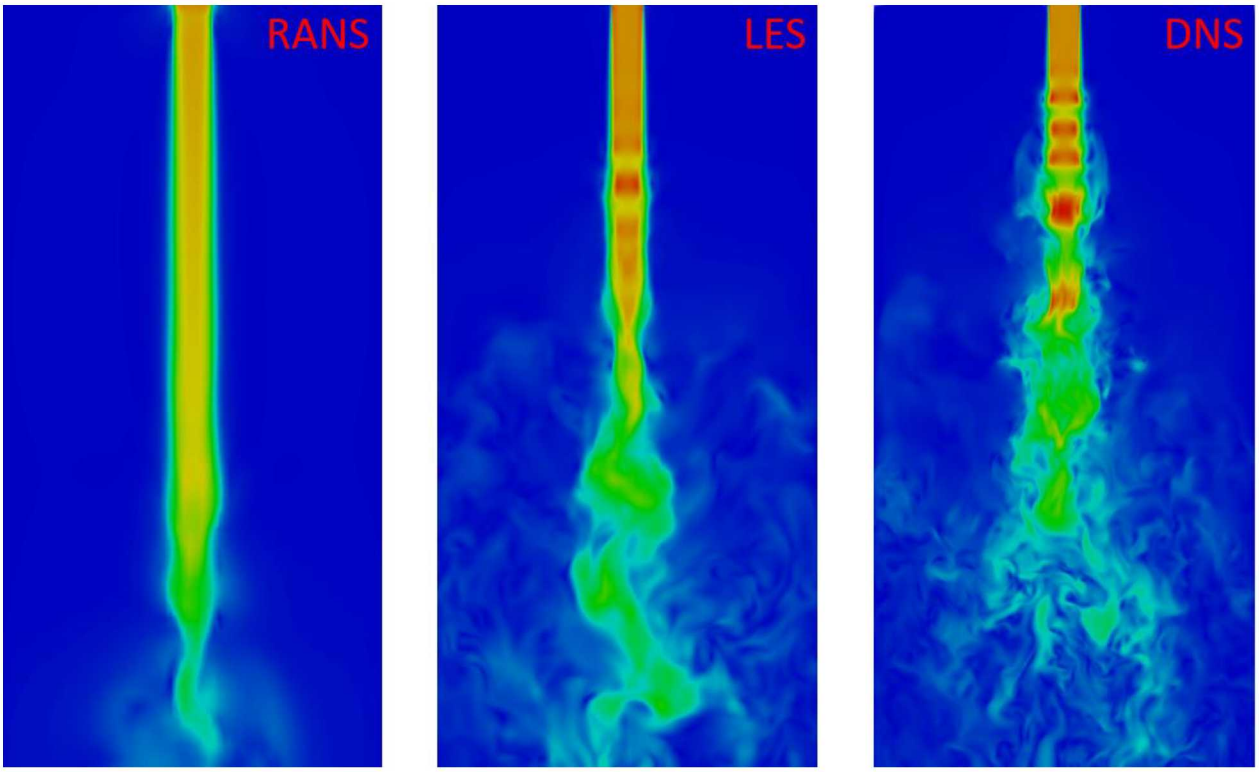

Figure 5.10. Velocity distribution for a turbulent jet using RANS (LHS), LES (middle), and DNS (RHS).

For the reasons discussed above, LES can be considered as an intermediate methodology between RANS and DNS, as a balance between output and computational effort. In general, LES is about an order or two of magnitude more expensive than RANS, but is an order or two cheaper than DNS. The reason for LES's computational cost is primarily due to the number of elements used (i.e., a simulation may require refinement up to the smaller eddies in the Taylor scale). On the other hand, RANS elements are much larger because they are not trying to resolve the integral and Taylor eddies that LES resolves. Unlike RANS, LES captures the behavior of the larger, energy-carrying eddies. Like RANS, the LES models typically employ the Boussinesq assumption (which is applied to the smaller eddies in the subgrid scale).

LES is great for adverse pressure gradients, complex surfaces, and swirl, but can be expensive in the boundary layer [Afgan, 2007; Rodriguez, 2011]; hence the use of DES. Due to its success in theoretical and, increasingly in engineering calculations, there are dozens of LES models in the literature. The interested reader is encouraged to pursue this subject matter. A partial list of such models includes

- Standard Smagorinsky model [Smagorinsky, 1963],
- Algebraic dynamic model (aka "dynamic subgrid-scale model" or "dynamic Smagorinsky") [Germano et al., 1991; Lilly, 1992],
- Localized dynamic model [Kim and Menon, 1993],
- Wall-adapting local eddy-viscosity (WALE) model [Nicoud and Ducros, 1999],

- Dynamic global-coefficient model [You and Moin, 2007],
- RNG-LES model [CFD-Online, 2018], and
- Kinetic energy subgrid-scale model (KSGS) [Fuego, 2016A; Fuego, 2016B].

5.4.1 How LES Works—A Brief Overview

As anticipated, the LES methodology divides the simulation into two areas. One portion calculates the velocity field of the larger eddies, thereby resolving their behavior explicitly, while the subgrid portion represents the smaller eddies, which are modeled (approximated). This is shown conceptually in Figure 5.11.

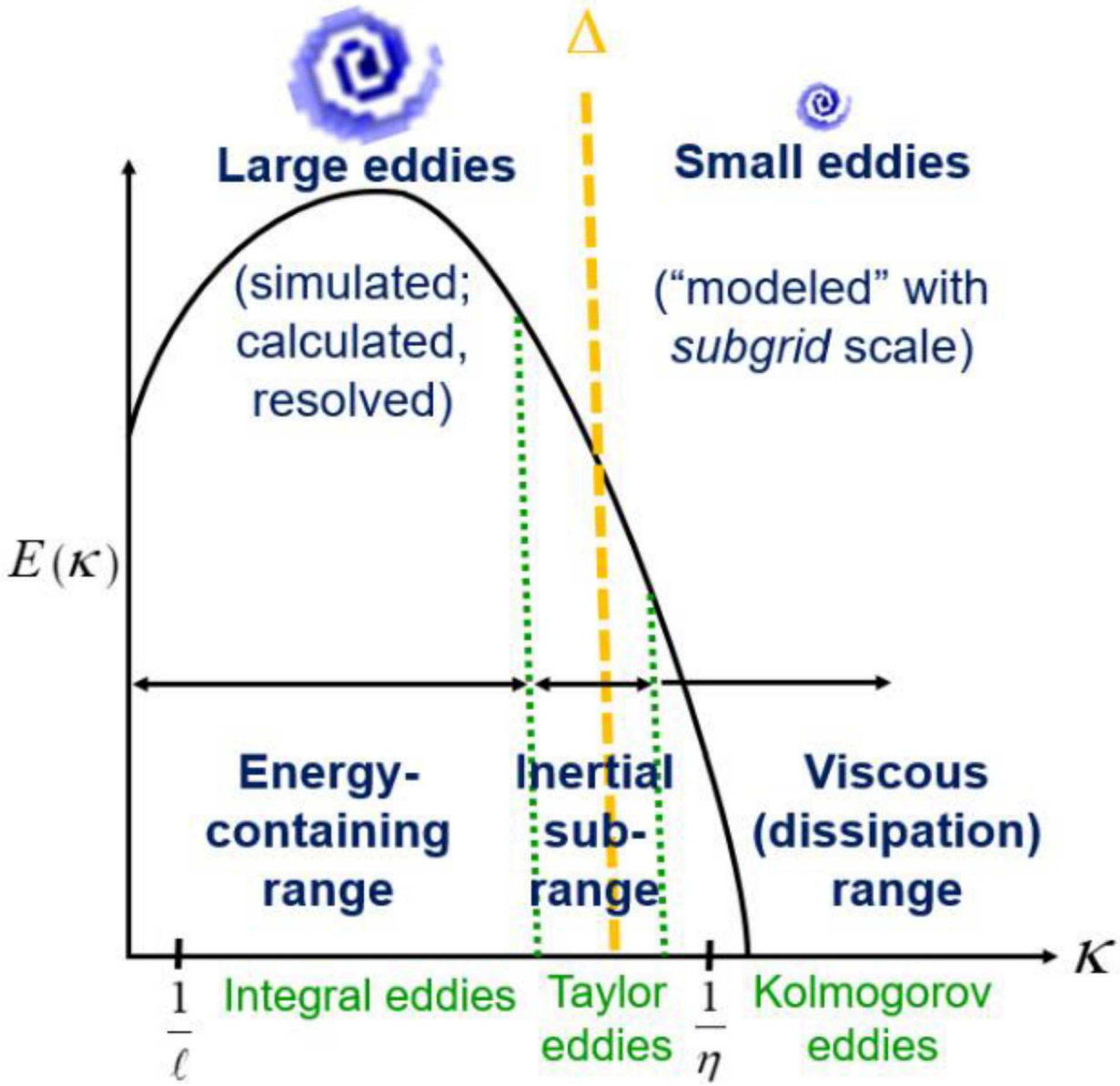

Figure 5.11. Size filtering of LES model eddies.

The computational approach is made by choosing some length scale Δ. In particular, if the eddy size $\geq \Delta$, such eddy belongs to the larger eddies that are resolved, and if the eddy is smaller than Δ, it belongs to the smaller, modelled category. That is, the larger eddies are simulated (calculated, resolved), while the smaller eddies are "modeled" (more like lumped as a homogeneous-like group for a RANS-like treatment). First proposed by Joseph Smagorinsky [Smagorinsky, 1963], the smaller eddies are modeled with *subgrid* scale (SGS) models. The two domains (the larger and the smaller subgrid eddies) are then coupled together to seamlessly simulate the entire turbulent flow space.

A generic "convolution filter" for the space filtering approach is as follows [Leonard, 1974; Shaanan, Ferziger, and Reynolds, 1975],

$$\overline{u}_i\left(\vec{x},t\right)=\iiint G\left(\vec{x},\vec{x}'\right)u_i\left(\vec{x},t\right)d^3\vec{x}\,.$$

A convolution takes two functions and mathematically changes (maps) them onto a new relationship that is related to one of the original functions. In this case, a convolution takes the instantaneous velocity u_i and filters it per eddy dimension, Δ, to produce the filtered larger eddy velocity, $\overline{u}_i$ and the smaller eddy velocity u_i'. So now, the instantaneous velocity is split as follows,

$$u_i = u_i(\vec{x},t) = \text{instantaneous velocity} = \overline{u}_i + u_i'\,,$$

where $\overline{u}_i$ is the resolved velocity field for the larger eddies and u_i' is the subgrid velocity for the smaller, modeled eddies; this concept is illustrated in Figure 5.12. Note that the velocity-field superposition notation is identical to Reynolds decomposition, *but has a different meaning*, viz., $\overline{u}_i$ is the velocity of the resolvable scale (the larger-eddy velocities), while u_i' is the modeled subgrid velocity (eddies whose size is < Δ). This means that the LES velocity field is

$$u_i = \sum\left(\text{resolved} + \text{subgrid}\right) = \sum\left(\text{larger eddies} + \text{smaller eddy approximation}\right).$$

Note that the convolution filter methods allow the analyst to employ GCI and Richardson extrapolation.

Some authors use the following notation for the filter function to more explicitly indicate the existence of a mapping transformation based on parameter, Δ,

$$G\left(\mathrm{x},\mathrm{x}'\right)\equiv G\left(\vec{\mathrm{x}}\text{-}\vec{\mathrm{x}}';\Delta\right).$$

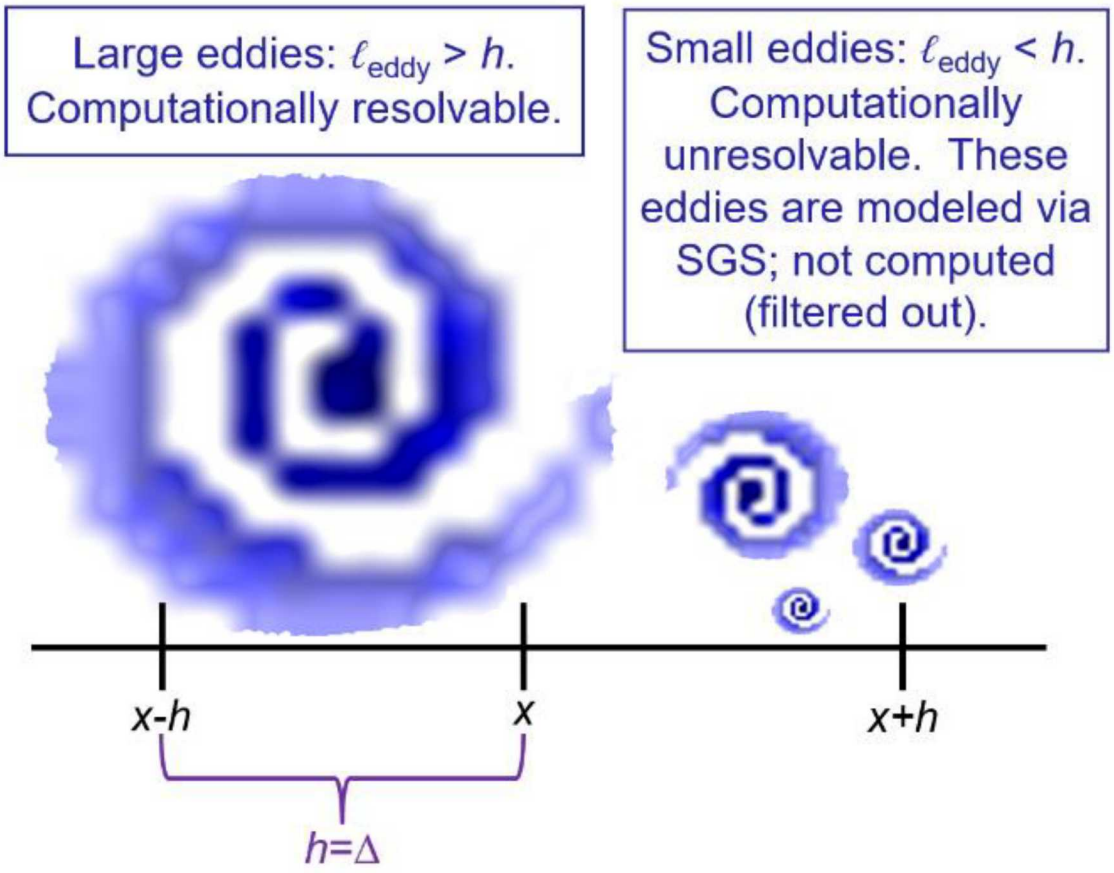

Figure 5.12. Resolvable vs. filtered eddies.

Many types of filters are used in the literature, and a general consensus is not yet fully formulated, primarily because the filters have properties that are not generically applicable [Deardorff, 1970; Shaanan, Ferziger, and Reynolds, 1975; Ferziger, 1977; Galperin and Orszag, 1993; Stefano and Vasilyev, 2002; Lesieur, Metais, and Comte, 2005]. Said more explicitly, the choice of filter will impact the solution, unfortunately. In particular, the SGS model output depends on whether the filter has a smooth or sharp cutoff. For example, smooth filters permit the filtered region to be fuzzier, in the sense that the cutoff is not sharply divided between the larger eddies that are resolved, and the smaller eddies that are modeled. By contrast, a sharp filter behaves more like a binary switch, being either "on" or "off", and thereby providing a well-defined separation cutoff [Stefano and Vasilyev, 2002]. Because turbulence is not smooth and fuzzy, but is instead sharp and chaotic, the sharp filter resonates well with the present authors. Some common filters are discussed next.

A simple, yet very effective filter is known as the volume-averaged box filter (aka "running mean filter") [Deardorff, 1970; Shaanan, Ferziger, and Reynolds, 1975; Fuego, 2016B]. The filter includes the element volume Δ^3 in its filtering criteria for cut-off; because the eddy is either inside or outside of the unit volume, this filter is of the sharp type, and is popular in CFD tools,

$$G\left(\vec{\mathrm{x}}\text{-}\vec{\mathrm{x}}';\Delta\right)=\begin{cases}\dfrac{1}{\Delta^3}, & \left|\mathrm{x}\text{-}\mathrm{x}'\right|<\dfrac{\Delta x}{2}\\ 0, & \text{otherwise}\end{cases}.$$

Another sharp, spectral filter employs a sine function to filter out the eddies [Garnier, Adams, and Sagaut, 2009],

$$G\left(\vec{\mathrm{x}}\text{-}\vec{\mathrm{x}}';\Delta\right)=\frac{\sin\left[\dfrac{\pi\left(\vec{\mathrm{x}}\text{-}\vec{\mathrm{x}}'\right)}{\Delta}\right]}{\pi\left(\vec{\mathrm{x}}\text{-}\vec{\mathrm{x}}'\right)}.$$

A Gaussian filter was developed to account for eddy distribution [Shaanan, Ferziger, and Reynolds, 1975], and is therefore a smooth filter [Stefano and Vasilyev, 2002],

$$G\left(\vec{\mathrm{x}}\text{-}\vec{\mathrm{x}}';\Delta\right)=\left(\frac{6}{\pi\Delta^2}\right)^{3/2}\exp\left(-\frac{6\left|\vec{\mathrm{x}}\text{-}\vec{\mathrm{x}}'\right|^2}{\Delta^2}\right).$$

Finally, the box filter is also known as the "grid filter" and as "top hat filter" [Wilcox, 2006; Garnier, Adams, and Sagaut, 2009]. This filter employs the Heaviside function *H*, which integrates the Dirac delta function, and therefore applies a cutoff at the halfway width, $\Delta/2$. This filter is also considered as a smooth filter [Stefano and Vasilyev, 2002], and is defined as,

$$G\left(\vec{\mathrm{x}}\text{-}\vec{\mathrm{x}}';\Delta\right)=\frac{1}{\Delta}H\left(\frac{\Delta}{2}-\left|\vec{\mathrm{x}}\text{-}\vec{\mathrm{x}}'\right|\right),$$

where

$$H=H\left(x\right)=\int_{-\infty}^{x}\delta\left(s\right)ds$$

and

δ = Dirac delta function.

5.4.2 LES mass and momentum conservation

As was done with the derivation of RANS, assume an incompressible, Newtonian flow. Use SS mass conservation and the transient "laminar" (i.e. unmodified, unfiltered) NS equation. WLOG, simplify further by letting $\mu\neq\mu\left(\vec{r}\right)$. Then, the unfiltered conservation of mass PDE is

$$\frac{\partial u_i}{\partial x_i}=0.$$

The unfiltered momentum conservation PDE is represented with the following NS formulation,

$$\frac{\partial u_i}{\partial t}+\frac{\partial\left(u_i u_j\right)}{\partial x_j}=-\frac{1}{\rho}\frac{\partial \tau_{ij,lam}}{\partial x_j}-\frac{1}{\rho}\frac{\partial P}{\partial x_i}=\frac{\mu}{\rho}\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)-\frac{1}{\rho}\frac{\partial P}{\partial x_i}.$$

Note that in its full form,

$$\tau_{ij,lam}=-\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right),$$

and not

$$\tau_{ij,lam}=-\mu\frac{\partial u_i}{\partial x_j}.$$

Now, perform the filter (bar) operation over the unfiltered NS equation to begin the derivation of the LES momentum equation; do not apply $u_i=\overline{u}_i+u_i'$ yet. Then,

$$\frac{\partial \overline{u}_i}{\partial t}+\frac{\partial\left(\overline{u_i u_j}\right)}{\partial x_j}=\frac{\mu}{\rho}\frac{\partial}{\partial x_j}\left(\frac{\partial \overline{u}_i}{\partial x_j}+\frac{\partial \overline{u}_j}{\partial x_i}\right)-\frac{1}{\rho}\frac{\partial \overline{P}}{\partial x_i}.$$

Now substitute $u_i=\overline{u}_i+u_i'$ into the convective term, $\overline{u_i u_j}$ (which is currently unknown), and perform the FOIL multiplication,

$$\overline{u_i u_j}=\overline{\left(\overline{u}_i+u_i'\right)\left(\overline{u}_j+u_j'\right)}=\overline{\left(\overline{u}_i\overline{u}_j+\overline{u}_i u_j'+u_i'\overline{u}_j+u_i'u_j'\right)}=\overline{\overline{u}_i\overline{u}_j}+\overline{\overline{u}_i u_j'}+\overline{u_i'\overline{u}_j}+\overline{u_i'u_j'}$$

Caution: unlike Reynolds averaging, $\overline{\overline{u}}_i\neq\overline{u}_i$ under LES bar operation; that is, the space filtering of a space-filtered velocity yields an entirely different velocity (whereas the RANS filtering of a filtered quantity yields the same quantity). Furthermore, under LES operation [Clark, Ferziger, and Reynolds 1979],

$\overline{u_i u_j}\neq\overline{u}_i\overline{u}_j$.

Instead,

$$\overline{u_i u_j}=\overline{\overline{u}_i\overline{u}_j}+\overline{\overline{u}_i u_j'}+\overline{u_i'\overline{u}_j}+\overline{u_i'u_j'}.$$

Note that Wilcox uses $\overline{\overline{u}_j u_i'}$ for the second term [Wilcox, 2006]; this is appropriate for symmetric tensors.

At this point, more succinct and fairly universal notation can be used, namely that

$$\overline{u_i u_j}=\overline{\overline{u}_i\overline{u}_j}+\overline{\overline{u}_i u_j'}+\overline{u_i'\overline{u}_j}+\overline{u_i'u_j'}=\overline{u}_i\overline{u}_j+C_{ij}+L_{ij}+R_{ij}$$

where the following standard notation is used:

$C_{ij} = \overline{\bar{u}_i u_j'} + \overline{u_i' \bar{u}_j}$ = cross-term stress [Clark, Ferziger, and Reynolds 1979], $L_{ij} = \overline{\bar{u}_i \bar{u}_j} - \bar{u}_i \bar{u}_j$ = Leonard stress arising from the convective term [Leonard, 1974; Wilcox, 2006], and

$R_{ij} = \overline{u_i' u_j'}$ = SGS Reynolds stress [Clark, Ferziger, and Reynolds 1979].

In summary, the space-filtered NS reduces to

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial\left(\overline{u_i u_j}\right)}{\partial x_j} = \frac{\mu}{\rho}\frac{\partial}{\partial x_j}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \frac{1}{\rho}\frac{\partial \bar{P}}{\partial x_i},$$

with the hitherto unknown term $\overline{u_i u_j}$ now expressible in a more manageable form as

$$\overline{u_i u_j} = \bar{u}_i \bar{u}_j + L_{ij} + C_{ij} + R_{ij}.$$

Next, substitute the expression for $\overline{u_i u_j}$ into the space-filtered NS and simplify the expression as follows,

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial\left(\bar{u}_i \bar{u}_j + L_{ij} + C_{ij} + R_{ij}\right)}{\partial x_j} = \frac{\mu}{\rho}\frac{\partial}{\partial x_j}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \frac{1}{\rho}\frac{\partial \bar{P}}{\partial x_i}.$$

Now rearrange it to obtain the sought-after space-averaged LES PDE,

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial\left(\bar{u}_i \bar{u}_j\right)}{\partial x_j} = \frac{\mu}{\rho}\frac{\partial}{\partial x_j}\left[\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \frac{1}{\rho}\frac{\partial \bar{P}}{\partial x_i} - C_{ij} - L_{ij} - R_{ij}\right].$$

It is worthwhile noting that the C_{ij} cross-term decreases as *Re* increases [Clark, Ferziger, and Reynolds 1979]. In addition, $\bar{u}_i \bar{u}_j$ goes away once L_{ij} is included, because the Leonard stress has a negative $\bar{u}_i \bar{u}_j$ term. Note that the Leonard stress can be approximated as being on the order of the truncation error associated with second-order finite differences [Shaanan, Ferziger, and Reynolds, 1975; Wilcox, 2006], and is ignored by some for that reason [Wilcox, 2006]. On the other hand, the Leonard stress term ought not to be discounted so readily, as it has a significant role regarding the transfer of the eddy turbulent kinetic energy during cascading [Shaanan, Ferziger, and Reynolds, 1975]. For low-*Re* applications, L_{ij} can be approximated as [Leonard, 1974; Clark, Ferziger, and Reynolds 1979],

$$L_{ij} \approx \frac{\gamma_L}{2}\nabla^2\left(\bar{u}_i \bar{u}_j\right),$$

where γ_L is calculated as the second moment of the filter function, *G*,

$$\gamma_L = \iiint \left|\vec{x}\right|^2 G(\vec{x}) d^3\vec{x}.$$

Finally, in shrewd anticipation of using various SGS closure relationships, the previous space-filtered NS form is conveniently re-formulated as follows,

$$\frac{\partial \overline{u}_i}{\partial t} + \overline{u}_j \frac{\partial \overline{u}_i}{\partial x_j} = \frac{\mu}{\rho}\frac{\partial}{\partial x_j}\left[\underbrace{\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right)}_{\text{Large resolved eddies}} + \underbrace{\tau_{ij}}_{\text{Small modeled eddies}}\right] - \frac{1}{\rho}\frac{\partial P}{\partial x_i},$$

where

$$\tau_{ij} \equiv -Q_{ij} + \frac{1}{3}Q_{kk}\delta_{ij},$$

$P \equiv \overline{P} + \frac{1}{3}\rho Q_{kk}$, and

$$Q_{ij} \equiv C_{ij} + L_{ij} + R_{ij} = SGS \text{ stresses}.$$

At this point, the LES filtered NS requires an analytical expression to calculate τ_{ij}. For this reason, there are dozens of SGS models in the literature, with the first model originating over half a century ago. The first model, now called the standard Smagorinsky, is analogous to the Newtonian shear model [Smagorinsky, 1963], and is expressed as,

$$\tau_{ij} \equiv 2\nu_t \overline{S}_{ij} = \left(C_S \Delta\right)^2 \sqrt{\overline{S}_{ij}\overline{S}_{ij}}.$$

The resolved strain rate is

$$\overline{S}_{ij} = \frac{1}{2}\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right),$$

while the Smagorinsky eddy kinematic viscosity is

$$\nu_t = \frac{\mu_t}{\rho} \equiv \left(C_S \Delta\right)^2 \sqrt{\overline{S}_{ij}\overline{S}_{ij}},$$

where

$\Delta = \left(\Delta x \Delta y \Delta z\right)^{1/3}$ and

C_s = the Smagorinsky constant.

Some authors use $0.1 \le C_s \le 0.3$ [Tutar and Holdo, 2001], while others use $0.1 \le C_s \le 0.24$ [Fuego, 2016A]. Additional guidelines regarding C_s are found in Section 5.4.3.

5.4.3 Miscellaneous LES modeling recommendations

- Computer processing unit (CPU) time is strongly dependent on the minimum element size, Δ (and number of elements, of course).
    - The smaller Δ is, the longer it will take to run the simulation.
- For very large simulations, RANS can be submitted first to get faster, preliminary results.
    - Then, the LES calculations ought to be submitted as soon as possible, as they can take 10 to 100 times more computational time than RANS.
- A good starting point for an LES simulation is to let $\Delta \approx \lambda$ (i.e., simulate down to the Taylor eddies).
    - The logic behind this approach is that LES resolves the larger eddies, which carry about 80% of the total turbulent kinetic energy. Therefore, setting $\Delta \approx \lambda$ means that sufficient computational nodes are included in the region occupied by the integral eddies, as well as a significant fraction of the Taylor eddies that can also carry a significant fraction of *k* (recall that the Taylor eddy spans a wide spatial range).
- The integral and Taylor eddy dimension can be estimated using the LIKE algorithm (refer to Sections 3.4 and 3.5).
- The usage of higher order elements such as hexahedrals is highly recommended (conversely, avoid lower-order tetrahedrals) [Tyacke et al., 2014].
- Caution: C_S magnitude has a strong dependency on geometry and flow conditions—unfortunately, there is no universal value that is valid for all scenarios.
- Some reasonable guidelines and values for C_S for various specific cases include:
    - C_S = 0.1 for internal flow in ducts [Rogallo and Moin, 1984],
    - C_S = 0.15 for flow around a sphere [Tutar and Holdo, 2001],
    - The Fuego CFD code has C_S = 0.17 as a default value, which is the average of its minimum and maximum range [Fuego, 2016A], in an attempt towards a good, all around value, and
    - C_S = 0.21 for flows where buoyancy dominates or flows with low mean shear [Rogallo and Moin, 1984],
    - C_S should be decreased for situations where the mean strain rate increases, and
    - If C_S is too large, the eddies will undergo excessive damping.
- The dynamic Smagorinsky model automatically calculates C_S in both space and time as the calculation proceeds, whereby $C_S = C_S\left(\vec{x}, t\right)$.

- Given a choice of the standard Smagorinsky vs. the dynamic Smagorinsky model, use the latter; a "constant" C_S can never fully represent a *dynamic* situation involving spatial and temporal changes.
- Unfortunately, the *standard* Smagorinsky calculates a non-zero value for ν_t at the wall, so some LES models are forced to use a damping function such as van Driest. Nevertheless, this issue can be overcome by using sufficiently-discretized resolution near the wall, without having to use damping functions [Rodi et al., 1997; Tutar and Holdo, 2001].
- A wall function can be detrimental for LES simulations with separated flow [Rodi et al., 1997].
- The addition of wall and damping functions tend to induce a counterintuitive flavor for LES purists. Namely, LES theory adheres to the notion that flow ought to be adequately calculated using a reasonable Δ that includes sufficient quantities of the larger and intermediate-sized eddies (i.e., a good fraction of the Taylor eddies), plus the behavioral contributions of the lumped, smallest eddies. So why "fudge" the simulation with extraneous functions? Particularly, any numerical accuracy that damping and wall functions may add to the simulation, can doubtlessly be obtained using sufficient spatial resolution, higher order elements, and good mesh metrics, especially near the wall.
- As for what is sufficiently discretized for LES, the limit can be based on purely theoretical arguments: the first computational node ought to be the minimum of: either the smallest eddies being resolved (say the Taylor eddies) *or* the size determined by the physics in question (e.g., the viscous layer is at $y^+ = 5$).
- Large *Re* simulations should include a mechanism for calculating the Leonard stress [Shaanan, Ferziger, and Reynolds, 1975].

5.5 DNS Modeling Recommendations

RANS calculates *time-averaged* turbulence effects, but no eddy dynamics. By contrast, direct numerical modeling (DNS) is a turbulence approach that solves the unsteady Navier-Stokes equations such that *all* turbulence scales are resolved, and unlike LES, no subgrid model is employed [Wilcox, 2006; Afgan, 2007]. As a result, DNS often-times requires multiple tens of millions to billions of computational nodes [Day, 2009]. DNS is therefore used sparingly, especially in large systems under high *Re*; the higher the *Re*, the higher the required node count. Nevertheless, time favors this turbulence method, especially as computers and algorithms become faster. As of 2018, DNS is generally *at least* two to three orders of magnitude more expensive than LES.

DNS is unique amongst the turbulence models because it does not employ averaging (no $\bar{u}$, no u'), no Boussinesq (or nonlocal, nonequilibrium approaches), no k, ε, ω, or ν_t, no wall functions, no curve fits, no *ad hoc* models, and so on and so forth. *DNS is purely Navier-Stokes calculated for all time and spatial scales.* DNS uniquely solves Navier-Stokes to calculate the actual, instantaneous, primitive-variable fluctuations (u', v', w', T', P', etc.).

Fortunately, as of 2018, DNS is no longer limited to low to moderate *Re* flows for reasonably-sized domains (this excludes very large systems such an entire nuclear plant, cruise ships, very large geophysical systems, etc.). Starting around 2005 or so, it has become more commonplace to see higher *Re* calculations for small- to mid-sized industrial applications [Huser and Biringen, 1993; Terentiev, 2006; Stein, 2009]. For example, excellent results were obtained for swirling jets at Re=5,000 and swirl number S=0.79 [Freitag and Klein, 2005]. A higher *Re* in the range of 12,000 ≤ Re ≤ 33,500 and S≤0.5 was achieved for swirling jets a year later [Facciolo, 2006]. Channel flow involving cubes as wall roughness have been simulated up to Re = 7,000 [Leonardi and Castro, 2010]. DNS has also been conducted on oscillating and stationary cylinders at Re = 10,000 [Dong and Karniadakis, 2005]. Using spectral methods, a total of 4096^3 (6.87×10^{10} grid points) were used to explore isotropic turbulence in atmospheric flows at Re_λ = 1,000 [Kaneda and Ishihara, 2006; Yeung et al., 2010]. And many more examples can be cited that show a favorable trend towards DNS. The case for DNS is further strengthened in light of faster computational systems, such as multi-core and manycore processors for increased computing performance [Alfonsi, 2011], as well as nano and quantum computers [Rudinger, 2017]. The novel hardware, combined with quantum algorithms, will surely continue the DNS computational growth trend for many decades to come, if not centuries. In any case, as computational power increases, DNS will not only be used for turbulence research and small systems, but for engineering design as well [Kim, Moin, and Moser, 1987].

Moreover, if done properly, DNS is extremely accurate—as good, if not better, than experimental data. For example, DNS can be better than experimental data because it can tract parameters that are difficult, and perhaps even impossible, to measure experimentally (e.g., *P')*. Furthermore, DNS provides much more detailed data than any experiment could ever achieve, e.g. "computational probes" in the millions to billions to trillions. Moreover, many recent studies can be cited in the literature whereby DNS calculations compare favorably with experimental data, to the point that many authors go as far as considering the output as good as experimental data [Moet *et al*., 2004; Freitag and Klein, 2005; Duraisamy and Lele, 2006; Afgan, 2007; Bonaldo, 2007; Busch, Ryan, and Sheard, 2007; Walther *et al*., 2007; Taub *et al*., 2010].

For the interested reader, many useful guidelines, too many to be cited here, can be found in the literature. A small sample includes [Moin and Mahesh, 1998; Modi, 1999; Wilcox, 2006; Coleman and Sandberg, 2010; Alfonsi, 2011; Joslin, 2012; Tryggvason and Buongiorno, 2013; Argyropoulos and Markatos, 2015; Joshi and Nayak, 2019].

5.5.1 DNS numerics

DNS *usually* requires higher-order numerical methods (e.g., fourth, fifth, and sixth order in space), basically with the goal of reducing numerical errors such as aliasing and associated instabilities [Rai and Moin, 1991; Huser and Biringen, 1993; Drikakis and Geurts, 2002; Sengupta and Bhaumik, 2019]. The aliasing error refers to the calculational error associated with the nonlinear convective term on the node-based mesh.

Generally, fourth order methods are used in DNS, especially during its earlier years [Coleman and Sandberg, 2010]. More recently, this trend has seen some changes, including the usage of both second and fourth order methods that can be used

successfully *if* error mitigation precautions are taken [Verstappen and Veldman, 1997; Wilcox, 2006].

Many DNS solvers are *explicit* due to large memory constraints required by this approach. But, more recently, implicit solvers have become more common [Rodriguez, 2000; Wilcox, 2006; Coleman and Sandberg, 2010; Alfonsi, 2011].

5.5.2 DNS BCs and ICs

BCs must be employed such that they do not generate spurious instabilities or have numerical errors that overshadow the eddy dynamics, especially the smaller eddies that are associated with very small velocity and length scales.

Inflow BCs pose issues because it is not possible to implement *a priori* inflow distribution. Therefore, the BC takes the DNS output from the computational domain, modifies the results to reflect the inlet conditions, and then uses the rescaled data for the next time step. An alternative is to let the flow reach turbulence, but that extends the domain to a prohibitive size. Another approach is to supply functions that randomly perturb the flow. Additionally, a relatively-coarser mesh calculation can be used as input, or perhaps even an LES simulation can be used as a starting point for the input; of course, these two would only be approximations.

Well-posed outflow BCs permit the eddies to exit the boundary seamlessly, without producing numerical errors, instabilities, and reflective waves. For this reason, researchers have developed non-reflecting, damping BCs, which have demonstrated their ability to suppress spurious waves [Thompson, 1987; Spalart, 1990; Nordstrom, Nordin, and Henningson, 1999].

Periodic BCs are extremely useful for DNS modeling. Recall that fully developed flow (FD) is relatively homogeneous along the perpendicular (spanwise) direction of the primary flow [Kim, Moin, and Moser, 1987; Moin and Mahesh, 1998; Leonardi and Castro, 2010]. Therefore, periodic BCs are ideally suited here. This is an important point, as having to include the entire system means the addition of tens of millions to billions more computational cells. Refer to Section 2.6 for estimating the entrance length of turbulent flows.

Wall BCs with no slip (provided *Kn* is in the appropriate range) are considered fairly safe for DNS applications [Coleman and Sandberg, 2010; Leonardi and Castro, 2010].

Regarding ICs, these can be obtained from coarser meshes, and then superimposed on the DNS grid [Coleman and Sandberg, 2010]. Then, the initial input is flushed out by allowing the calculation to run for several time periods, typically three or more flowthroughs [Dong and Karniadakis, 2005]. Alternatively, ICs can be chosen such that they start the simulation using reasonable values, such as having the initial velocity equal to zero throughout the domain. Then, the simulation proceeds until it reaches its stationary limit [Day et al., 2009]. At this point, the calculated turbulence data should be independent of the ICs.

5.5.3 DNS spatial domain

LES calculates the integral eddies and larger eddies up to some user-defined or mesh-defined minimum scale, such as the Taylor scale. On the other hand, DNS not only resolves all the LES eddies and the smaller Taylor eddies that were filtered out by LES, but it also resolves the Kolmogorov eddies. Said most concisely, DNS calculates all eddy scales. Because DNS includes the Kolmogorov eddies, it is not uncommon for the first computational node to be at small fractions of y^+, with $y^+ << 1.0$. For example, a channel flow at *Re*=3,300 included the first computational node at $y^+ = 0.05$, while its maximum spacing was set to $y^+ = 4$ [Kim, Moin, and Moser, 1987]. Of course, Kolmogorov eddy size is dependent on *Re*, and the larger *Re* is, the smaller y^+ ought to be. These are good guidelines, but to ensure more problem-specific node spacing, the user is encouraged to use the LIKE algorithm to determine node spacing; refer to Sections 3.4 and 3.5.

The Kolmogorov eddy is the smallest eddy scale that is sustainable by the flow, and is calculated as follows,

$$\eta = \left(\frac{\nu^3}{\varepsilon} \right)^{1/4} .$$

The distance between the computational nodes must not exceed this value. However, to fully capture the eddy's interior dynamics, a more rigorous restriction is imposed for the distance between the computational nodes,

$$\Delta x \leq \frac{\eta}{3} \text{ to } \frac{\eta}{2}.$$

As for the largest eddies, it is not appropriate to assume that the hydraulic diameter is the limiting scale. For example, because the larger integral eddies can stretch significantly (say at 45° from the main flow direction), the maximum bound is conservatively placed at

$$\Lambda = 2D_h .$$

This size restriction guides the minimum size that a mesh domain can be, and still have enough space to capture all eddies. As a check that the DNS computational mesh is sufficiently large, the eddy fluctuations must be *uncorrelated* up to half the distance of the domain for the largest eddies.

5.5.4 DNS time and stability criteria

If the time step is limited by the Courant limit, then

$$\frac{u' \Delta t_{Courant}}{\Delta x} = \frac{u' \Delta t_{Courant}}{\eta} < 1.0 .$$

On the other hand, the Kolmogorov eddy life time before collapsing into a laminar sheet is calculated as

$$\tau = \left(\frac{\nu}{\varepsilon} \right)^{1/2} .$$

Therefore, the smallest time step is the minimum of the two,

$$\Delta t_{DNS} = \min\left(\tau,\ \Delta t_{Courant}\right).$$

If an explicit DNS simulation show signs of instability, the time step can be restricted even further using a more restrictive criterion, which is based on the wall friction velocity, u_τ and the channel characteristic length, x_{char} [Kim, Moin, and Moser, 1987; Wilcox, 2006],

$$\Delta t \approx \frac{0.003}{\sqrt{Re_\tau}} \frac{x_{char}}{u_\tau},$$

where

$$Re_\tau = \frac{u_\tau x_{char}}{\nu}.$$

A stability criterion that combines the convective and diffusive limits is as follows [Coleman and Sandberg, 2010],

$$\left(1 - 4\frac{\nu \Delta t}{\Delta x^2}\right)^2 + \left(\frac{u \Delta t}{\Delta x}\right)^2 \leq 1,$$

which applies to the following expression for momentum conservation,

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu \frac{\partial^2 u}{\partial x^2}.$$

As discussed in Section 3.5, the number of nodes required in 1D DNS calculations is estimated as

$$N_{1D} \approx \mathrm{Re}_T^{3/4},$$

where

$$Re_T = \frac{\eta \sqrt{k}}{\nu}.$$

For 3D DNS calculations, the number is significantly larger [Afgan, 2007; Sodja, 2007; Stein, 2009; Taub *et al.*, 2010],

$$N_{3D} \approx Re_T^{9/4} \text{ to } Re_T^{11/4}.$$

A more precise relationship is as follows [Wilcox, 2006; Sodja, 2007],

$$N_{3D} = \left(110 Re_T\right)^{9/4}$$

As might be fully expected by now, the impressive 3D DNS calculations come at a high cost: the CPU needed to solve these problems is a strong function of the turbulent *Re*, so the required computational power only increases stratospherically to the third power as *Re* increases,

$$\text{CPU} \propto Re_T^3 .$$

5.6 Miscellaneous Do's and Don'ts of the Trade

5.6.1 Well-posed solutions

Certain criteria must be met to solve fluid dynamics PDEs. This includes the mathematical notion that the problem is "well-posed", meaning that the following three conditions are satisfied by the solution: 1) the solution exists (which sounds obvious, but not all problems have solutions!), 2) the solution is unique, and 3) the solution depends continuously on its boundary and initial conditions. For tough engineering problems that seem to exhibit more than their fair share as "code breakers", it is advisable to question if the problem at hand is well-posed. Are the boundaries consistent? Does a solution exist?

Example 5.8. A university wishes to perform wind tunnel experiments and CFD analysis on a dimpled airfoil experiencing a wide *Ma* range, from subsonic (*Ma* < 1) to supersonic (*Ma* ≥ 1.0) (but not hypersonic). To simplify the analysis, assume that the flow is SS, inviscid, irrotational, and compressible. In addition, simplify the problem further by considering a 2D Cartesian geometry, with no external heat sources, and let the air behave as an ideal gas. Because the flow is SS and inviscid, the transient and viscous terms readily drop, but the convective term remains. For this highly-simplified situation, the momentum and energy conservation equations reduce to $\left(1 - Ma^2\right)\frac{\partial u}{\partial x} + \frac{\partial \mathrm{v}}{\partial y} = 0$ after several pages of elegant mathematical procedures that involve the Prandtl-Glauert rule for linearizing compressible, isentropic flow [Hanson, 2012; Pritamashutosh, 2014]. For this situation, determine if this problem is well-posed for the *Ma* domain in question.

Solution. Many terms in the conservation equations are zero for this for this idealized situation. In particular, because the flow is SS,

$$\frac{\partial}{\partial t} = 0 .$$

Because the flow is inviscid,

$$\nabla^2 V = 0 .$$

Because the flow is compressible (*Ma* > 0.3),

$$\vec{\nabla} \cdot \vec{V} \neq 0 .$$

This implies that the conservation of mass is reduced to,

$$\cancel{\frac{\partial \rho}{\partial t}} = -\rho\left(\frac{\partial u}{\partial x} + \frac{\partial \mathrm{v}}{\partial y} + \cancel{\frac{\partial w}{\partial z}}\right) - \left(\cancel{u\frac{\partial \rho}{\partial x}} + \cancel{\mathrm{v}\frac{\partial \rho}{\partial y}} + \cancel{w\frac{\partial \rho}{\partial z}}\right),$$

$$\frac{\partial u}{\partial x} + \frac{\partial \mathrm{v}}{\partial y} = 0.$$

And because the flow is irrotational, the cross product of the velocity is zero. Namely, from the definition of an irrotational flow,

$$\vec{\nabla} \times \vec{V} \equiv \vec{0} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ u & \mathrm{v} & w \end{vmatrix} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & 0 \\ u & \mathrm{v} & 0 \end{vmatrix} = \frac{\partial \mathrm{v}}{\partial x}\vec{k} - \frac{\partial u}{\partial y}\vec{k}.$$

Now use the dot product to multiply by $\vec{k}$ (because $\vec{k} \bullet \vec{k} = 1$), thus reducing the PDE to

$$\frac{\partial \mathrm{v}}{\partial x} - \frac{\partial u}{\partial y} = 0.$$

The momentum and irrotational PDEs are very valuable, as they allow the elegant usage of the eigenvalue method, and this will be used to investigate the unruly behavior of this seemingly straightforward PDE system of equations. In particular, the above two PDEs conform to the following 2D PDE generic classification [DuChateau and Zachmann, 2011],

$$\begin{cases} a_1 \frac{\partial u}{\partial x} + b_1 \frac{\partial u}{\partial y} + c_1 \frac{\partial \mathrm{v}}{\partial x} + d_1 \frac{\partial \mathrm{v}}{\partial y} = e_1 \\ a_2 \frac{\partial u}{\partial x} + b_2 \frac{\partial u}{\partial y} + c_2 \frac{\partial \mathrm{v}}{\partial x} + d_2 \frac{\partial \mathrm{v}}{\partial y} = e_2 \end{cases}.$$

Therefore, the two PDEs can be put into matrix form, in anticipation of determining their eigenvalues, and thus pin down their behavior (e.g., it will be determined if the flow is hyperbolic, parabolic, elliptic, or mixed). The two PDEs can now be expressed in general matrix format as,

$$M\frac{\partial W}{\partial x} + N\frac{\partial W}{\partial y} = E,$$

such that the *M* and *N* matrices are associated with the *x* and *y* partial derivatives, respectively,

$$M = \begin{bmatrix} a_1 & c_1 \\ a_2 & c_2 \end{bmatrix} \text{ and } N = \begin{bmatrix} b_1 & d_1 \\ b_2 & d_2 \end{bmatrix}.$$

In this context, the two vectors are

$$W = \begin{Bmatrix} u \\ \mathrm{v} \end{Bmatrix}$$

and

$$E = \begin{Bmatrix} e_1 \\ e_2 \end{Bmatrix}.$$

The coefficients for *M and N are as follows*,

$$M = \begin{bmatrix} 1 - Ma^2 & 0 \\ 0 & -1 \end{bmatrix} \text{ and } N = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

Fortunately, *M* is a 2x2 matrix, so obtaining its inverse is straightforward [Kreyzig, 1979]; the inversion is performed in preparation to obtain the system eigenvalues,

$$M^{-1} = \begin{bmatrix} \dfrac{1}{1 - Ma^2} & 0 \\ 0 & -1 \end{bmatrix}.$$

For convenience, let

$\Phi \equiv M^{-1}N$. Then,

$$\Phi = \begin{bmatrix} \dfrac{1}{1 - Ma^2} & 0 \\ 0 & \text{-}1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & \dfrac{1}{1 - Ma^2} \\ \text{-}1 & 0 \end{bmatrix}.$$

Notice that in this particular case, *N* acts as a rotational operator on *M*; that is, it rotates each of the coefficients in a clockwise direction by one element. Finally, the λ eigenvalues are found through the determinant, $\left|\Phi \text{ - } \lambda I\right| = 0$ [Kreyzig, 1979]. That is,

$$\begin{vmatrix} \text{-}\lambda & \dfrac{1}{1 - Ma^2} \\ \text{-}1 & \text{-}\lambda \end{vmatrix} = 0 = \lambda^2 + \frac{1}{1 - Ma^2}.$$

Solving for λ, it is evident that the two distinct eigenvalues are solely a function of *Ma*,

$$\lambda = \pm\sqrt{\frac{1}{Ma^2 - 1}}.$$

From the rules of the eigenvalue method, if there are two real and distinct eigenvalues, then the solution is hyperbolic. If there is only a single real eigenvalue, the solution is parabolic. Finally, if the eigenvalues are imaginary, then the solution is elliptic. Therefore, this problem has a mixed behavior, being elliptic if *Ma* < 1 and hyperbolic for *Ma* > 1. But, what happens when the hapless CFD engineer is asked to solve the problem for *Ma*=1? Will a solution exist? Will the CFD calculation cease to abort if the timestep is reduced or more elements are added?

5.6.2 Time steps, stability, and CFL

Most modern commercial CFD tools are fully implicit. This means that their numerical method is unconditionally stable. That is, the method *should* be stable for all time steps. However, this does not mean that very large time steps are encouraged. In fact, too large a time step in an implicit algorithm will increase the numerical error as a result of truncation. By contrast, explicit and semi-explicit methods are easier to program, but should never use a time step larger than the Courant limit; failure to do so will result in numerical instability, large parameter oscillations per time step, nonsensical output, a severe cut in the time step if the coding attempts to adjust the time step, and code aborts. Thus, whether a numerical method is implicit or explicit, it is always a good idea to calculate the Courant number, either as a guide to limit truncation error, or to avoid instabilities, respectively. The Courant number is also referred as the CFL number (the last name initials of its developers) [Courant, Friedrichs, and Lewy, 1967].

To determine the CFL limit for a 1D PDE, consider the following,

$$\frac{\partial \phi}{\partial t} + c\frac{\partial \phi}{\partial x} = 0,$$

where

ϕ = a scalar (e.g., ρ, *T, u, v, w*) and

c = parameter for a given PDE (e.g., *u*, etc.)

Then, if *c*=*u*, meaning that the PDE is a 1D *laminar*, inviscid flow momentum equation, the 1D CFL limit can be expressed as,

$$CFL = c\frac{\Delta t}{\Delta x} = u\frac{\Delta t}{\Delta x} \le C_{\max},$$

where

Δt = time step,

Δx = distance between computational nodes in the *x* direction, and

$C_{\max}$ is the maximum size of the CFL number, depending on the computational situation, as explained earlier and later in this Section.

In 2D, the *laminar* CFL is expressed as

$$CFL = \frac{u\Delta t}{\Delta x} + \frac{\mathrm{v}\Delta t}{\Delta y} \le C_{\max},$$

while in 3D, the *laminar* CFL is

$$CFL = \frac{u\Delta t}{\Delta x} + \frac{\mathrm{v}\Delta t}{\Delta y} + \frac{w\Delta t}{\Delta z} \le C_{\max}.$$

CFL values greater than 1, up to 5 or so, are acceptable for implicit solvers [Andersson et al., 2012]. Some code developers push the envelope even further, using CFL as large as 10, usually as a quick turnaround for testing new code. (Though this likely results in large truncation error). Nevertheless, used with caution, CFL values greater than 1 for implicit codes are desirable, especially for those seeking faster numerical solutions. However, it is up to the analyst to show that temporal discretization is satisfied, that the solution converges as Δt is reduced. This can be shown as follows: once the solution converges *spatially* using a given time step Δt_1, a second simulation is run using the same mesh, but with a time step of $\Delta t_1/2$. If the solution does not change appreciably (say <1%), temporal discretization has been reached, at least reasonably so. For the truly obsessed (or diligent!), yet a third simulation can be performed, with the *spatial* distance being cut once more in half, using $\Delta t_1/4$, and plotted to show temporal convergence vs. a desired variable, e.g., *u*, *P*, etc.

In contrast with implicit solvers, a CFL value of 1.0 or less is necessary for explicit solvers, lest the solution becomes unstable. For an *explicit* application for the *laminar, compressible*, *inviscid* Navier-Stokes in a 2D Cartesian system [Anderson, Tannehill, and Pletcher, 1984], the maximum recommended time step is,

$$\Delta t_{max} \le \frac{1}{\frac{|u|}{\Delta x} + \frac{|\mathrm{v}|}{\Delta y} + u_s\sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2}}}$$

where

u_s = sound speed.

For convenience, the above expression will be referred as “ATP”, and can be extended onto a *laminar* 3D format as,

$$\Delta t_{max} \le \frac{1}{\frac{|u|}{\Delta x} + \frac{|\mathrm{v}|}{\Delta y} + \frac{|w|}{\Delta z} + u_s\sqrt{\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} + \frac{1}{(\Delta z)^2}}},$$

as well as simplified onto the *laminar* 1D expression,

$$\Delta t_{max} \le \frac{1}{\frac{|u|}{\Delta x} + u_s \sqrt{\frac{1}{(\Delta x)^2}}}.$$

Note that the ATP expressions are for explicit numerics (not implicit), so CFL ≤ 1.0. Note as well that such expressions do not depend on viscosity, as Δt_{max} is limited to situations where the viscous effects are negligible compared with the inertial term. As a further note of caution, stability analysis not only depends on the PDE in question, but also on the type of numerical discretization used [Courant, Friedrichs, and Lewy, 1967]. Thus, stability analysis tends to be *ad hoc*, for very specific applications, and not necessarily generalizable. Fortunately, there are exceptions, because the physical PDEs tend to follow similar expressions (see Problem 5.8), and that is the focus of much research [Courant, Friedrichs, and Lewy, 1967; Anderson, Tannehill, and Pletcher, 1984]. And one final cautionary note: the above expressions are not suitable for turbulent flows! This situation will be treated though simple approximations later in this Section.

Now, a close inspection of the CFL criterion and the ATP expressions shows that these limits are specific to convective limits. What happens if stability is dependent on the viscous dissipation, meaning that only so much viscous momentum can be transferred per unit time? Consider a system where Navier-Stokes has the following form,

$$\rho \frac{\partial u}{\partial t} + \rho \left(u \frac{\partial u}{\partial x} + \mathrm{v} \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) = \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) - \frac{\partial P}{\partial x} + \rho g .$$

Suppose that the viscous (i.e., molecular diffusion) term is larger in magnitude than the convective term, and that the pressure and body force are small. Then, for a 1D *laminar* system,

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} .$$

The above equation can be approximated using forward in time, centered in space (second order) finite differences. Then, the von Neumann stability analysis method (or some such method) can be applied to determine stability limits. In this case, the numerical error *E* in both space and time can be modeled as a Fourier series,

$$E(x,t) = \sum_{n=1}^{M} e^{\alpha t} e^{ikx} .$$

And, after about one page of elegant algebra, a stable time step is obtained if the following condition is satisfied,

$$\Delta t_{max} \le \frac{1}{2} \frac{\Delta x^2}{\nu} .$$

Again, note that the above stability expression applies for *laminar* flows.

The above expression can also be derived by using scaling analysis, whereby the transient viscous PDE is transformed according to its key parameters and variables. Namely the PDE scales (transforms) approximately (to within an order of magnitude) as follows:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial x^2} \;\leftrightarrow\; \frac{u}{t} \sim \nu \frac{u}{x^2}.$$

Note that scaling is generally palatable for engineers, but is known to give mathematicians extreme heartburn!

For small time steps and small changes in space, then $t \sim \Delta t$ and $x \sim \Delta x$, respectively; it is assumed that the error in these approximations is small because the changes are linear, i.e., taken near a well-known point and not deviated much from it. Therefore, substituting the approximations, scaling allows the following approximation,

$$\Delta t_{max,\,scale} \sim \frac{\Delta x^2}{\nu}.$$

Then, because diffusing more than half of the available mass per unit time step will lead to instabilities, the maximum allowable time step is cut in half, and imposed as a limiting bound,

$$\Delta t_{max,\,scale} \;\le\; \frac{1}{2}\frac{\Delta x^2}{\nu}.$$

Thus, the same bounding equation is derived for laminar flows, whether scaling arguments are used or a more rigorous method is applied (i.e., von Neumann stability analysis). Thus, the solution lends credibility to the scaling method, and it is therefore extended onto *turbulent* flows. That is, how can the maximum time step be estimated for a turbulent flow? Lacking much guidance in the literature, the analysis begins by noting that for turbulent flows, $\nu_t \gg \nu$, so the viscous term can be ignored (at least as a first order approximation). Now assume a situation where the convective term is sufficiently smaller than the turbulence diffusion term, and considering only the *x*-direction,

$$\frac{\partial \bar{u}}{\partial t} = 2\frac{\partial\left(\nu_t \dfrac{\partial \bar{u}}{\partial x}\right)}{\partial x}.$$

Using scaling analysis, the PDE is thereby transformed into,

$$\frac{\bar{u}}{t} \sim 2\nu_t \frac{\bar{u}}{x^2}.$$

Finally, in analogy with the laminar convective scaling completed earlier, the limiting bound for turbulent diffusion ought to permit at most half of the available mass to transfer per unit time step. Further, the approximation will consider only small time steps and spatial changes. Hence,

$$\Delta t_{turbulent} \leq \frac{\Delta x^2}{\nu_t}.$$

Finally, to ensure stability for explicit methods, the minimum time step based on the convective (CFL or ATP) and diffusive bounds should be used, where diffusive refers to either molecular or turbulent,

$$\Delta t_{min} \leq \min\left(\Delta t_{convective}, \Delta t_{diffusive}\right).$$

5.6.3 A few more tips

- Reducing computational time:
  - Where appropriate, use symmetry to reduce the element count.
  - Calculational time is substantially reduced if the problem can be reduced from 3D to 2D, or even 1D.
  - Certainly, turbulence is a 3D phenomenon, but it can still be simulated in a 2D domain when the flow tends to be symmetric about the primary flow direction. Furthermore, the largest turbulent fluctuations tend to occur along the primary flow direction.
  - For implicit calculations, let CFL > 1, to perhaps 2, and as high as 5. This is highly practical for exploratory calculations; but for the final calculation, reduce the CFL so that truncation error is minimized.
  - Seek to reduce the system size. For example, regions of interest may already be FD, so modeling the flow as it reaches FD is not necessary. If so, an FD flow BC will significantly reduce the computational domain.
  - Use node biasing to increase the distance between nodes in regions with small velocity gradients. But ensure that the expansion ratio is less than 1.5.
- In general, higher-order numerical methods have higher accuracy, and therefore require fewer computational nodes. However, higher-order methods tend to be more unstable because they have less numerical dissipation. In addition, fourth-order spatial methods require more complex boundaries, including so-called “ghost boundaries”.
- Use iterative solvers (e.g., Gauss-Siedel, Jacobi, and successive over-relaxation) for very large matrices (e.g., systems with a large number of computational nodes, say in the millions or billions).
- Use preconditioners to transform an unruly matrix so that it is more manageable, thereby making it more amenable towards numerical iteration or solution.
- Preconditioners can be useful under the following circumstances:
  - unstructured meshes (meshes with irregular patterns),

- ill-posed matrices, matrices with large Jacobians, or systems with large condition numbers,
- multiphase flows,
- flows with widely-varying *Ma,* and
- meshes with large span (range) in aspect ratio.

5.7 Natural Circulation Modeling Hints

Many systems undergo natural convection, aka free convection and natural circulation. This phenomenon is typically generated as a result of temperature gradients that induce density changes in fluids, thereby causing the fluid to flow. Other natural circulation-inducing gradients include species concentration gradients such as changes in salinity, and so forth. The key point is that under natural circulation, there is no forced circulation, such as occurs from pumps, injectors, and the likes. Natural circulation systems include passive cooling of nuclear reactors under normal and accident conditions, bodies of water (ponds, lakes, oceans), weather patterns (thunderstorms, wind, tornadoes, hurricanes), storage tanks, heat exchangers, micro devices, heat pipes, heat sinks, building circulation flow patterns, etc. Natural circulation flows can be laminar, transitional, or turbulent. And to add more complexity, these flows can be purely natural or involve a degree of extraneous forced flow, in which case the flow is considered as "mixed" circulation. Because natural circulation flows tend to have lower velocities than forced flows, they exhibit special numerical challenges.

5.7.1 Natural circulation modeling

Consider a Cartesian system under natural circulation, subject to conservation of mass, momentum, and energy, as shown in Figure 5.13. If the system is 2D (which is a reasonable approximation due to symmetry), then conservation of momentum under laminar natural circulation is,

$$\rho\frac{\partial u}{\partial t}+\rho\left(u\frac{\partial u}{\partial x}+\mathrm{v}\frac{\partial u}{\partial y}\right)=\mu\frac{\partial^2 u}{\partial y^2}-\frac{\partial P}{\partial x}+\rho g$$

where
u = fluid velocity in the x-direction,
v = fluid velocity in the y-direction,
ρ = fluid density,
μ = fluid dynamic viscosity, and
g = gravitational constant in the x-direction.

Note that the viscous term in the x-direction is negligible because the u velocity gradient WRT to x is a relatively-smaller quantity.

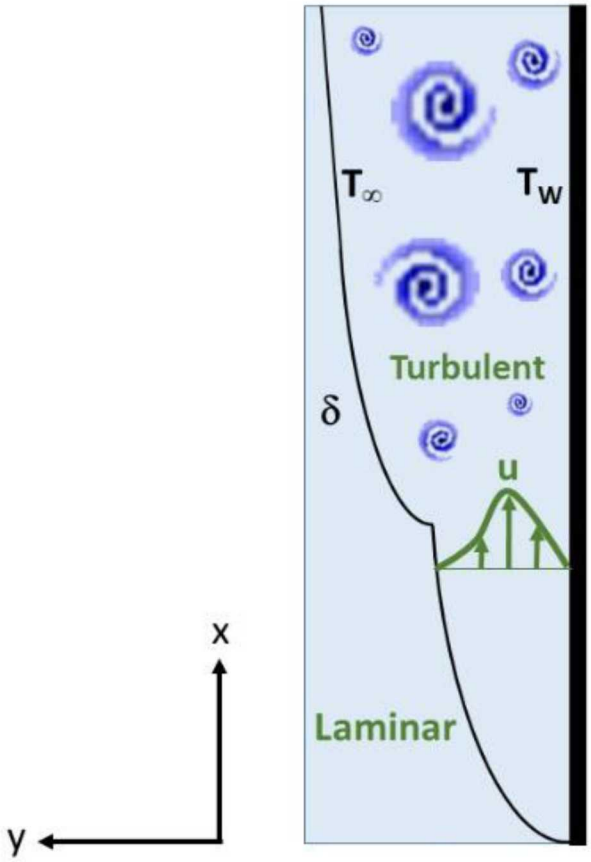

Figure 5.13. Thermal boundary layer generated by a heated wall on the LHS.

The conservation of energy equation is,

$$\frac{\partial T}{\partial t} + \vec{V} \bullet \vec{\nabla} T = \frac{\dot{Q}_0'''}{\rho C_p} + \alpha \nabla^2 T \,,$$

where

T = fluid temperature,

$\vec{V}$ = velocity vector,

$\dot{Q}_0'''$ = volumetric heat source,

t = time,

$\alpha = \frac{k}{\rho C_p}$ = thermal diffusivity.

k = fluid thermal conductivity,

ρ = fluid density, and

C_p = fluid heat capacity under constant pressure.

The energy equation simplifies under the assumption of a 2D system with no heat source,

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + \mathrm{v}\frac{\partial T}{\partial y} = \frac{k}{\rho C_p}\frac{\partial^2 T}{\partial y^2}.$$

At this point, the dimensionless Grashof number (*Gr*) can be defined, which is analogous to *Re*. In particular, *Re* is the degree of laminarity or turbulence under *forced* circulation, while *Gr* times the Prandtl number (*Pr)* is a measure of the degree of laminarity or turbulence under *natural* circulation (*GrPr* = Raleigh number=*Ra*). For example, for a vertical plate, G*rPr* < 10^9 implies laminar natural circulation, while *GrPr* > 10^9 implies turbulent natural circulation [Holman, 1990]. *Gr, Pr, and Re* are defined as,

$$Gr = \frac{\beta g h^3 \Delta T}{\nu^2},$$

$$Pr = \frac{C_p \mu}{k}, \text{ and}$$

$$Re = \frac{uh}{\nu}.$$

In this context,

$$\beta \equiv -\frac{1}{\rho}\left(\frac{\partial \rho}{\partial T}\right)_P \approx -\frac{1}{\rho}\frac{\left(\rho - \rho_\infty\right)}{\left(T - T_\infty\right)} = \frac{1}{\rho}\frac{\left(\rho_\infty - \rho\right)}{\left(T - T_\infty\right)},$$

or

$$\rho_\infty - \rho = \rho\beta\left(T - T_\infty\right),$$

$\Delta T = \left(T_w - T_\infty\right)$,

h = wall height,

T_w = temperature of the wall in contact with the fluid (heats-up the fluid),

T_∞ = fluid temperature far away from the wall, and

$\nu = \frac{\mu}{\rho}$ = fluid kinematic viscosity.

Note that there is a pressure gradient caused by the fluid's weight per unit length of the fluid,

$$\frac{\partial P}{\partial x} = -\rho_\infty g.$$

At this point, it is convenient to substitute the two expressions for pressure and density into the momentum equation, as follows,

$$\rho\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \mathrm{v}\frac{\partial u}{\partial y}\right) = \mu\frac{\partial^2 u}{\partial y^2} + g\left(\rho_\infty - \rho\right) = \mu\frac{\partial^2 u}{\partial y^2} + \rho g\left[\beta\left(T - T_\infty\right)\right].$$

Furthermore, the transient term can be dropped if SS is assumed, and dividing by ρ,

$$u\frac{\partial u}{\partial x}+\mathrm{v}\frac{\partial u}{\partial y}=\nu\frac{\partial^2 u}{\partial y^2}+\frac{g\left(\rho_\infty-\rho\right)}{\rho}=\nu\frac{\partial^2 u}{\partial y^2}+g\left[\beta\left(T-T_\infty\right)\right].$$

But the new PDE expression cannot be solved exactly. Fortunately, there are many ways to obtain approximate solutions for this intractable PDE, such as back-of-the envelope energy balances [White, 1991], numerical methods [Ostrach, 1953], and extended numerical solutions resulting in a curve-fit solution [Rodriguez and Ames, 2015].

A simplified method for estimating the natural circulation velocity begins by assuming that there is a macroscopic (lumped) balance between the potential and kinetic energies (*PE* and *KE*, respectively) for the fluid [White, 1991],

$$PE=\frac{1}{2}gh\Delta\rho$$

and

$$KE=\frac{1}{2}\rho V_{NC}^2,$$

where V_{NC} is the average natural convection velocity. If *PE* is approximately balanced by *KE*, then

$$\frac{1}{2}gh\Delta\rho\approx\frac{1}{2}\rho V_{NC}^2.$$

Solving for the characteristic fluid velocity,

$$V_{NC}\approx\left(gh\frac{\Delta\rho}{\rho}\right)^{1/2}\approx\left(gh\beta\Delta T\right)^{1/2},$$

where it is assumed that

$$\frac{\Delta\rho}{\rho}\approx\beta\Delta T.$$

Various researchers have employed polynomial techniques to approximate PDE solutions [Blasius, 1908; Holman, 1990; Haberman, 2004]. For example, Holman assumes a cubic polynomial with four unknown constants, and then solves the unknowns subject to the problem's BCs [Holman, 1990],

$$u(x,y)=U\left(a+by+cy^2+dy^3\right)=U\left(y\right).$$

The *laminar* velocity distribution *u* is assumed to be solely a function of *y*, with no *x* dependency. This is not a bad approximation, as most of the velocity changes occur in

y, as shown in Figure 5.13. Then, upon applying the BCs, and after a few pages of algebra, the desired velocity distribution is obtained for a laminar, natural circulation flow for Pr ≈ 1 [Holman, 1990],

$$U(y) = \frac{\beta\delta^2 \mathrm{g}\left(T_w - T_\infty\right)}{4\nu}\frac{y}{\delta}\left(1-\frac{y}{\delta}\right)^2.$$

However, the above laminar velocity can be further developed by taking the derivative of *u* WRT *y,* and setting it to 0. That is, the thermal boundary thickness, *y,* can be obtained at the location where *u* is maximum, as follows,

$$\frac{du}{dy} = 0 = \left[\frac{\beta\delta^2 g\left(T_w - T_\infty\right)}{4\nu}\right]\left(\frac{1}{\delta} - 4\frac{y}{\delta^2} + 3\frac{y^2}{\delta^3}\right).$$

After about one page of algebra,

$$y = \frac{\delta}{3}.$$

In other words, for any location *x*, there is a maximum peak velocity *u*, which is always located at $y=\delta/3$ (this is a direct consequence of the velocity distribution being parabolic; refer to Figure 5.13). Thus, the maximum velocity *u(x)* for laminar flows with *Pr* ≈ 1 is now derived as,

$$u_{\max}\left(y=\frac{\delta}{3}\right) = \frac{\beta\delta^2 \mathrm{g}\left(T_w - T_\infty\right)}{4\nu}\left(\frac{4}{27}\right) = \frac{\beta\delta^2 \mathrm{g}\left(T_w - T_\infty\right)}{27\nu}.$$

A more general laminar velocity expression, valid for 0.001 ≤ Pr ≤ 1,000 [Rodriguez and Ames, 2015], is based on an extension of Ostrach's work [Ostrach, 1953], and is cited as follows,

$$u_{max}(x) = 2\left(0.5Pr^{-0.11} - 0.24\right)\frac{\nu_\infty}{x}\sqrt{Gr_x}.$$

The thermal boundary layer for the laminar circulation was derived by Blasius over a century ago [Blasius, 1908],

$$\delta = \delta(x) = \frac{5x}{\sqrt{Re_x}}.$$

The thermal boundary layer thickness for the laminar natural circulation flow can also be expressed as [Holman, 1990],

$$\delta = \delta(x) = C_1 x^{1/4},$$

where C_1 is a function of the fluid physical properties α, β, and ν; moreover, the middle term is quite similar to *Gr*, and is in fact the so-called buoyancy parameter times the temperature difference [White, 1991],

$$C_1 = 3.93\left(\frac{20}{21}+\frac{\nu}{\alpha}\right)^{1/4}\left[\frac{\beta g\left(T_w - T_\infty\right)}{\nu^2}\right]^{-1/4}\left(\frac{\alpha}{\nu}\right)^{1/2}.$$

5.7.2 Additional natural circulation modeling guidelines

- Knowing the peak velocity allows the analyst to estimate a reasonable time step. The CFL criteria can therefore be estimated by using the peak velocity for laminar flows discussed in Section 5.7.1.
- *Pr* has a significant impact on the modeling of natural circulation flows [Ostrach, 1953; Holman, 1990; Yokomine et al., 2007; Rodriguez and Ames, 2015].
- Low *Pr* materials (e.g., liquid metals) cannot be adequately modeled using a fixed turbulent *Pr*, as the results can diverge from Nusselt number (*Nu*) experimental data by as much as 50%. This divergence trend increases as *Re* increases. Nevertheless, this situation can be fixed by using wall functions suitable for low-*Pr* fluids, along with low-*Pr* turbulence models such as the k-ω-k_t-ε_t [Bna et al., 2012]. Pr_t is generally a function of *Re* and *Pr* for low-*Pr* fluids [Jischa and Rieke, 1979; Chen et al., 2013],

$$Pr_t = 0.9 + \frac{182.4}{PrRe^{0.888}}.$$

- For large *Pr* materials (e.g., oils), the turbulent Pr_t is generally a function of both *Re* and *Pr* as well [Hasan, 2007; Yokomine, 2007],

$$Pr_t = 6.374Re - 0.238Pr - 0.161.$$

- In general, a reasonable time step for natural circulation flows is typically on the order of 10s to 100s of times larger than forced convection, because natural circulation flows tend to be that much slower.
- For codes that allow the user to use different (separate) solvers for mass, momentum, and energy (e.g., Fuego [Fuego, 2016B]), it is preferable to choose the same solver for all three PDEs, and to use the same convergence criteria. Failure to do so may result in inconsistent solutions and code aborts.
- Because the flow motion depends on density differences that occur near the wall, it is critically important that the mesh near the wall be sufficiently discretized and have good mesh metrics. For turbulent natural circulation flows, y^+=1 is an ideal starting point. The criteria for laminar, natural circulation flows is not as restrictive, and is resolvable with meshes having 10s to as much as 1,000s of fewer nodes than if the flow were turbulent and modeled with a RANS model.

- Natural circulation flows can involve many transitions, including chaotic transitions, bifurcations, and turbulence. Therefore, changes in key parameters may result in significantly divergent flows [Gleick, 1988; Strogatz, 1994].
    - Because of the above situation, small changes in initial conditions may result; under the right conditions, chaotic flows and bifurcations will occur as well [Strogatz, 1994].
    - In fact, the famous system of three coupled PDE equations presented by Lorenz that so beautifully capture the butterfly-like Lorenz attractor, were derived directly from conservation of energy and the Navier Stokes equations [Lorenz, 1963]. This confirms that the seemingly deterministic PDEs have lots of hidden chaotic structure!

5.8 Data Visualization Tips

As discussed in Chapter 1, an extreme advantage of CFD and Multiphysics over experiments is that millions of computational nodes can be used to model a system, with each node providing information as though it were a thermocouple, a pressure transducer, a flow meter, and so on. This is certainly not possible experimentally. Other issues include instrumentation that results in unintended changes for the experimental output, such as being a heat source or sink, or interfering with the flow by blocking it or diverting it, and so forth. Certainly, good experimentalists will take care of these and other issues, so this would never happen, right?

Needless to say, the human eye will not respond well to reams of numerical data, so what are some ways to most effectively summarize millions of computational data points? Because data visualization is a science and an art form that is embedded with diverse and conflicting human factors, there is no ultimate consensus as to how to generate excellent images, figures, charts, etc. [Sanders and McCormick, 1987]. But, speaking in general terms, computational output imagery should be focused, clear, legible, self-contained, and show a compelling point or story; having eye appeal ("eye-candy") is a definite bonus. That said, there are general, useful tips that blend various human factors, data display, and art forms.

- Use overlay as much as possible. Overlay refers to the superposition of two or more images, and in this context, the overlaying of key surfaces and some form of parameter color rendering. For example, the top image on the RHS of Figure 5.15 shows a velocity distribution for flow around six cylinders. However, by overlaying the six tubes with translucent coloring (refer to "volume" rendering in Paraview), the flow pattern comes alive and makes more intuitive sense, as shown in the lower LHS image. The overlay of streamlines and the tubes is shown in the lower RHS. More often than not, it is recommended that the system geometry (or a cut-out section) be included. If necessary for a clearer view of the velocity distribution, set the solid body's opacity to 5 to 20%.

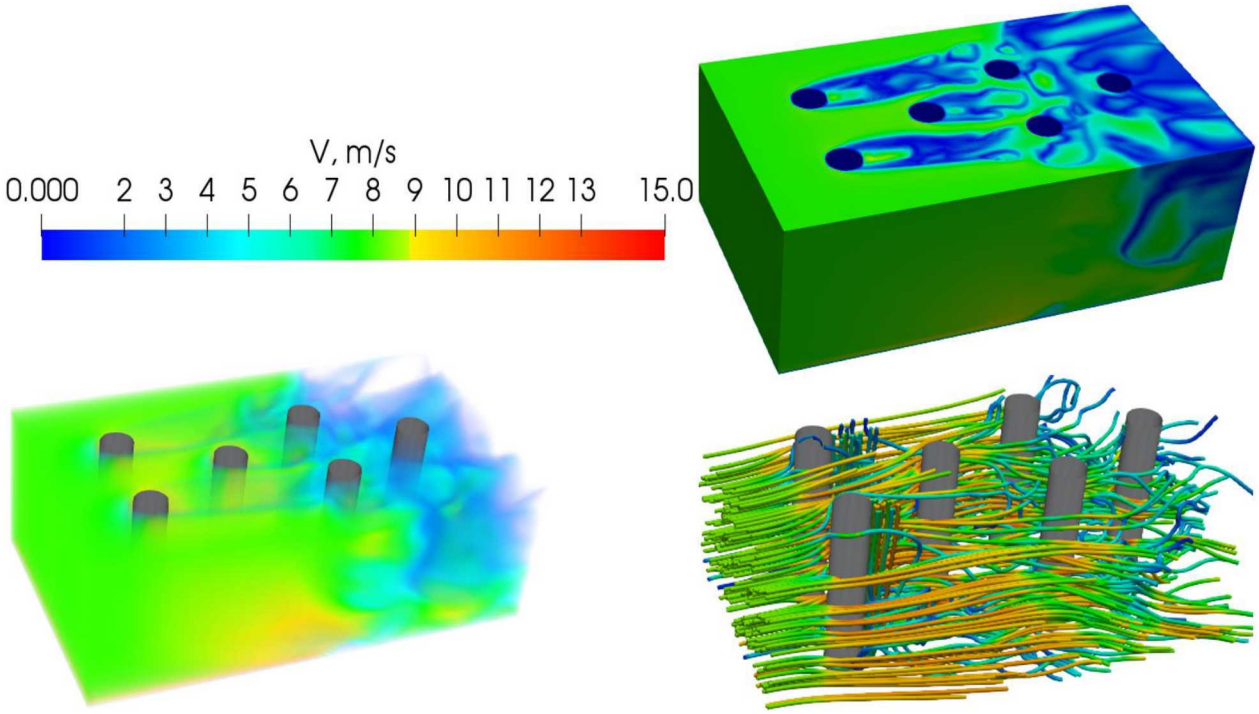

Figure 5.15. Velocity distribution around vertical cylinders using various overlay schemes.

- Visualization can be used to show how spatial convergence is coming along, and might even point to meshing issues. This is achieved by overlaying the computational mesh onto the parameter color display (whether it is *T, u, v, w,* etc.). In particular, meshes that are sufficiently discretized will show that the element coloring by parameter is independent of the mesh grid pattern—the parameter values (by color) must not follow (hug) the computational elements. For example, notice that Figure 5.16 (upper LHS) has various unusual vertical velocity streaks; this image provides few clues as to the source of the problem. However, once the mesh is overlaid onto the colored velocity distribution (upper RHS), it is immediately evident that the velocity distribution follows (is dependent upon) the element boundaries, and the culprit immediately follows. The overlay shows that the mesh has a large aspect ratio on the RHS of the domain and a large expansion ratio at the interface between the small and large elements (central region). Once better mesh metrics are applied to the model, the velocity distribution is shown to be independent of the element boundaries (lower RHS), and the mysterious velocity distribution vanishes.

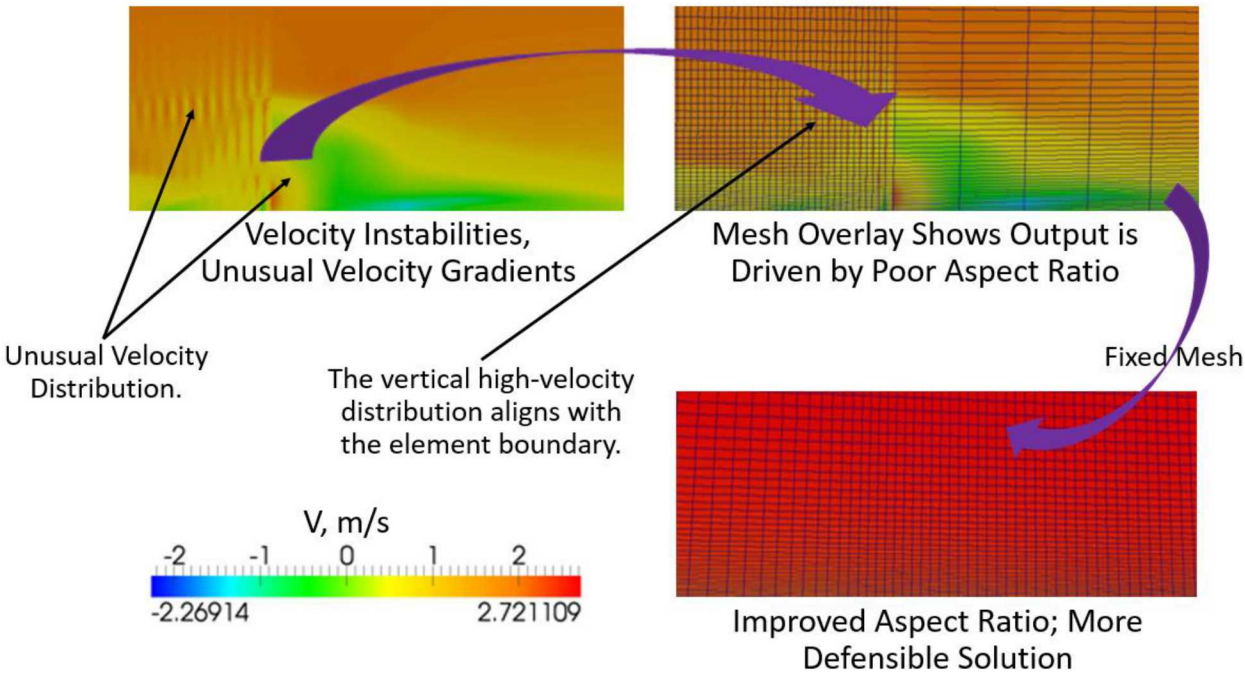

Figure 5.16. Velocity distribution in a rectangular domain. Upper two images: mesh has a large aspect ratio and large expansion ratio. Bottom: mesh with better mesh metrics.

- It is often useful to include multiple sets of parameters on a single image (e.g. single page). For example, if the image is split into two screens, then *T* can be shown on one side and perhaps *u* on the other, and so forth. This juxtaposition of computational parameters can provide many useful engineering insights.
- A significant portion of the population is color blind, so do not count on color to provide the entire exposition of data [Sanders and McCormick, 1987]. To avoid this issue, not only consider using different colors for the curves, but also use symbols (e.g., triangles, circles, etc.) and diverse types of curves (e.g., dashed, dotted, etc.). The curves and symbols should be much larger than size 1, and more likely should be on the order of 3 to 7.
- Consider using log scales (both for curves and coloring by parameter) when there is a broad range in parameter space.
- Use arrows and brief descriptions to identify key changes in parameters. Many programs can to do this, including Matlab and Paraview. However, the process can also be accomplished by copying the figure onto PowerPoint, overlaying arrows and comments on the slide, and then generating the final image through a screen shot.
- Certainly, too many arrows, comments, and “bells and whistles” can be distracting, and even detrimental. Again, this is an art and a science. The point is to *focus* on the narrative that a figure should convey. If too much information is required, consider using several figures instead of one.
- It is wise to spend on the order of five minutes per figure to check for errors and pesky, ubiquitous typos. This is also an area where spell checkers are not

available. Check the units. Make sure the font is still readable in the document (not just on the image that was created!). The same applies to the curves, are they distinguishable? If possible, have a colleague inspect your figure(s); do they make sense? Does your image convey the story you need to express?

- Do not use yellow on white! Images require sharp contrast.
- For extremely appealing, artful images, consider using opposing colors, such as they appear on color wheels. For example, blue and orange go well together on a side-by-side basis, and so does green and red; ditto cyan and light green; etc.
- In contrast to the previous point, avoid using, on a side by side basis, colors that are adjacent in the color wheel. For example, blue and purple do not go well near each other; ditto yellow and light orange; or light blue and green.
- Remember that data will likely be viewed by important people with diverse backgrounds, including engineers of many types, managers, administrators, students, lawyers, and financiers with deep pockets. Therefore, terms that make sense to a civil engineer may not mean much for a nuclear engineer, and so forth. A well-thought-out figure will cross these language barriers, and appeal to a wider audience.
- Avoid using ACRONYMS in figures. (As a side note, some acronyms have diverse meanings, depending on the field. They should always be defined in reports.)
- A figure should be "self-contained", meaning that in of itself, it must contain all the information necessary for the audience to understand its message. For example, do not assume that a deeply-buried paragraph in the report will clarify the information that is missing in a legend or that of a poorly-written figure title.
- Images should have font size that are at least 20 or higher. Do not ruin a great computational effort with legends that are unreadable (or not present, as is sometimes the case).
- The legend should clearly label all curves, and allow the reader to fully understand the parameter range under consideration.
- Indeed, "a picture is worth a thousand words". Make it count!

5.9 Problems

5.1 Is it ever acceptable to only use one mesh metric? Why or why not?

5.2 What is a minimum set of independent mesh metrics, and why?

5.3 Can a mesh metric replace the aspect ratio? Why or why not?

5.4 Explain the CFL number, and what are acceptable CFL values? Why is the CFL magnitude different for explicit and implicit solvers?

5.5 For an explicit calculation, is it sufficient to only check the CFL criteria? If not, what else should be checked?

5.6 Consider a smooth flat plate under isothermal boundary layer flow. The plate is 0.1 m long and 0.05 m wide, with air flowing parallel to the plate at 300 K and 1 atmosphere (ν=1.58x10^{-5} m^2/s and U_s = 347.3 m/s). The air flows from left to right along the 0.1 m plate at a constant velocity U_∞ = 15.8 m/s. Will a turbulence model be needed? Develop a mesh with aspect ratios ≥ 50 and skew ≥ 5. Use biasing near the wall, with an expansion ratio ≥ 15. Compare that CFD solution with another mesh that has aspect ratio ≤ 5, skew ≤ 0.5, and an expansion ratio ≤ 1.5. What happened here?

5.7 Show that for a laminar flow with the following velocity distribution,

$U(y) = \dfrac{\beta\delta^2 \mathrm{g}\left(T_w - T_\infty\right)}{4\nu}\dfrac{y}{\delta}\left(1-\dfrac{y}{\delta}\right)^2$, the peak velocity is

$u_{\max} = \dfrac{\beta\delta^2 \mathrm{g}\left(T_w - T_\infty\right)}{27\nu}$.

5.8 A system in 3D Cartesian space, where $V=V(x,y,z)$, $0 \le x \le L$, $0 \le y \le M$, and $0 \le z \le N$. The PDE is $\dfrac{\partial^2 u}{\partial x^2}+\dfrac{\partial^2 \mathrm{v}}{\partial y^2}+\dfrac{\partial^2 w}{\partial z^2}=0$. What BCs are needed to solve the equation?

5.9 A system in 3D Cartesian space, where $V=V(x,y,z)$, $0 \le x \le L$, $0 \le y \le M$, and $0 \le z \le N$. The mass conservation PDE is $\dfrac{\partial \rho}{\partial t} = -\rho\left(\dfrac{\partial u}{\partial x}+\dfrac{\partial \mathrm{v}}{\partial y}+\dfrac{\partial w}{\partial z}\right) - \left(u\dfrac{\partial \rho}{\partial x}+\mathrm{v}\dfrac{\partial \rho}{\partial y}+w\dfrac{\partial \rho}{\partial z}\right)$.
What BCs and ICs are needed to solve the equation?

5.10 Water at 400 K and 6 MPa is flowing inside a cylindrical duct at a mass flow rate of 50 kg/s with D=0.1 m. If an implicit solver is used with CFL=5 and the first computational node is at y^+ = 1, what is the expected time step?

5.11 Consider the following Cartesian 3D energy equation,

$$\frac{\partial T}{\partial t}+u\frac{\partial T}{\partial x}+\mathrm{v}\frac{\partial T}{\partial y}+w\frac{\partial T}{\partial z}=\frac{k}{\rho C_p}\left(\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}+\frac{\partial^2 T}{\partial z^2}\right)+\frac{\dot{Q}_0'''}{\rho C_p}.$$

Assume no heat source and SS. What is the heat convection stability criteria? Hint: note the similarity between the energy and the *u*-momentum conservation equation, with no external pressure source, no body force (e.g., no *g*), and the viscous term is much smaller than the convective term:

$$\rho\frac{\partial u}{\partial t}+\rho\left(u\frac{\partial u}{\partial x}+\mathrm{v}\frac{\partial u}{\partial y}+w\frac{\partial u}{\partial z}\right)=\mu\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}+\frac{\partial^2 u}{\partial z^2}\right)-\frac{\partial P}{\partial x}+\rho\mathrm{g}.$$

5.12 Consider the same situation for the energy equation as in the previous Problem, except that now, the diffusive term is much larger than the convective term. What is the diffusive stability limit?

5.10 References

Afgan, I., "Large Eddy Simulation of Flow Over Cylindrical Bodies using Unstructured Finite Volume Meshes", Ph.D. Diss., University of Manchester, July 2007.

Alfonsi, G., "Direct Numerical Simulation of Turbulent Flows", ASME, Applied Mechanics Reviews, Vol. 64, 2011.

Anderson, D. A., J. C. Tannehill, and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Corporation, 1984.

Andersson, B. et al., *Computational Fluid Dynamic for Engineers*, Cambridge University Press, 2012.

Blasius, H., "Grenzschichten in Flussigkeiten mit kleiner Reibung", Z. fur Math. und Phys., Vol 56, No., 1, 1908. Also available in English as, "The Boundary Layers in Fluids with Little Friction", National Advisory Committee for Aeronautics, Technical Memorandum 1256, 1950.

Bna, S. et al., "Heat Transfer Numerical Simulations with the Four Parameter k-ω-k_t-ε_t Model for Low-Prandtl Number Liquid Metals", XXX UIT Heat Transfer Conference, Bologna, June 2012.

Bolaños, S. J. and B. Vernescu, "Derivation of the Navier Slip and Slip Length for Viscous Flows over a Rough Boundary", Physics of Fluids, Vol. 29, 2017.

Bonaldo, A., "Experimental Characterisation of Swirl Stabilized Annular Stratified Flames", Ph. D. Diss., Cranfield University, 2007.

Busch, H., K. Ryan, and G. J. Sheard, "Strain-Rate Development between a Co-Rotating Lamb-Oseen Vortex Pair of Unequal Strength", 16th Australasian Fluid Mechanics Conference, Gold Coast, Australia, December 2-7, 2007.

Brewer, M. L. and D. Marcum, "Proceedings of the 16th International Meshing Roundtable", Springer, 2008.

CFD-Online, "RNG-LES Model", https://www.cfd-online.com/Wiki/RNG-LES_model. Accessed on July 13, 2018.

Chen, F. et al., "Investigation on the Applicability of Turbulent-Prandtl-Number Models for Liquid Lead-Bismuth Eutectic", Nuclear Engineering and Design, Vol. 257, 2013.

Clark, R. A., J. H. Ferziger, and W. C. Reynolds, "Evaluation of Subgrid-Scale Models Using an Accurately Simulated Turbulent Flow", J. Fluid Mech., Vol. 91, Part 1, 1979.

Coleman, G. N. and R. D. Sandberg, “A Primer on Direct Numerical Simulation of Turbulence – Methods, Procedures and Guidelines”, Aerodynamics & Flight Mechanics Research Group, University of Southampton, 2010.

Courant, R., K. Friedrichs, and H. Lewy, “On the Partial Difference Equations of Mathematical Physics”, IBM Journal, March 1967 (English version; the German version first appeared in Mathematische Annalen, Vol. 100, in 1928).

Cubit, “CUBIT 15.3 User Documentation”, SAND2017-6895 W, Sandia National Laboratories, 2017. Also available at https://cubit.sandia.gov/public/15.3/help_manual/WebHelp/cubithelp.htm. Accessed on June 27, 2018.

Day, M. et al., “Combined Computational and Experimental Characterization of Lean Premixed Turbulent Low Swirl Laboratory Flames”, Lawrence Berkeley National Laboratories, circa 2009.

Deardorff, J. W., “A Numerical Study of Three-dimensional Turbulent Channel Flow at Large Reynolds Numbers”, J. Fluid Mech., Vol. 41, Part 2, 1970.

Dong, S. and G. E. Karniadakis, “DNS of Flow Past a Stationary and Oscillating Cylinder at Re = 10000”, J. of Fluids and Structures, Vol. 20, 2005.

Drikakis, D. and B. J. Geurts, editors, *Turbulent Flow Computation*, Kluwer Academic Publishers, 2002.

DuChateau and Zachmann, *Partial Differential Equations*, Third Ed., Schaum’s Outlines, McGraw-Hill, 2011.

Duraisamy, K. and S. K. Lele, “DNS of Temporal Evolution of Isolated Vortices”, Center for Turbulence Research, Proceedings of the Summer Program, 2006.

Facciolo, L., “A Study on Axially Rotating Pipe and Swirling Jet Flows”, Royal Institute of Technology, Department of Mechanics, S-100 44 Stockholm, Sweden, Ph.D. Diss., 2006.

Ferziger, J. H., “Large Eddy Numerical Simulations of Turbulent Flows”, AIAA Journal, Vol. 15, No. 9, 1977.

FIDAP Version 8.52, Theory and User’s Manual, Fluent, Inc., 1999. (FIDAP uses the FIMESH meshing software.)

Fluent, “ANSYS FLUENT 12.0 User’s Guide”, 2009.

Fluent, “Best Practice Guidelines”, Lecture 10, ANSYS, 2012.

Fluent, “ANSYS FLUENT 16.2.3 User’s Guide”, 2018. Also available at https://www.sharcnet.ca/Software/Ansys/16.2.3. Accessed on June 27, 2018.

Freitag, M. and M. Klein, “Direct Numerical Simulation of a Recirculating Swirling Flow”, *Flow, Turbulence and Combustion*, Vol. 75, 2005.

Fuego, “SIERRA Low Mach Module: Fuego Theory Manual – Version 4.40”, Sandia National Laboratories, 2016A.

Fuego, “SIERRA Low Mach Module: Fuego User Manual – Version 4.40”, Sandia National Laboratories, 2016B.

Galperin, B. and S. A. Orszag (editors), *Large Eddy Simulation of Complex Engineering and Geophysical Flows*, Cambridge University Press, 1993.

Garnier, E., N. Adams, and P. Sagaut, *Large Eddy Simulation for Compressible Flows*, Springer, 2009.

Germano, M. et al., “A Dynamic Subgrid Scale Eddy Viscosity Model”, *Physics of Fluids A*, Vol. 3, No. 7, 1991.

Gleick, J., *Chaos*, Penguin Books, 1988.
Haberman, R., *Applied Partial Differential Equations with Fourier Series and Boundary Value Problems*, Pearson Prentice Hall, 4th Ed., 2004.
Hanson, R., "Aerodynamics", University of Toronto Institute for Aerospace Studies, AER307, 2012.
Hasan, B. O., "Turbulent Prandtl Number and its Use in Prediction of Heat Transfer Coefficient for Liquids", Nahrain University, College of Engineering Journal (NUCEJ), Vol. 10, No. 1, 2007.
Holman, J., *Heat Transfer*, 7th Edition, McGraw-Hill, Inc., 1990.
Huser, A. and S. Biringen, "Direct Numerical Simulation of Turbulent Flow in a Square Duct", J. Fluid Mech., Vol. 257, 1993.
HyperMesh, "Element Quality and Checks", https://altairhyperworks.com/product/HyperMesh. Can also access at https://www.scribd.com/doc/6675303/Hypermesh-Quality-Tutorials. Accessed on June 26, 2018.
Jischa, M. and H. B. Rieke, "About the Prediction of Turbulent Prandtl and Schmidt Numbers from Modeled Transport Equations", Int. J. Heat Mass Transfer, No. 22, 1979.
Joshi, J. and A. Nayak, *Fluid Dynamics in Nuclear Reactor Design and Safety Assessment*, Woodhead Publishing, 1st Ed., 2019.
Joslin, R. D., "Discussion of DNS: Past, Present, and Future", NASA, Langley Research Center, circa 2012.
Kaneda, Y. and T. Ishihara, "High-Resolution Direct Numerical Simulation of Turbulence", J. of Turbulence, Vol. 7, No. 20, 2006.
Kim, W. and S. Menon, "A New Dynamic One-Equation Subgrid-Scale Model for Large Eddy Simulation", 33rd Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 1995.
Kim, J., P. Moin, and R. Moser, "Turbulence Statistics in Fully Developed Channel Flow at Low Reynolds Number", J. Fluid Mech., Vol. 177, 1987.
Knupp, P. M., "Achieving Finite Element Mesh Quality via Optimization of the Jacobian Matrix Norm and Associated Quantities", Int. J. Numer. Meth. Engng., Vol. 48, 2000.
Knupp, P. M., "Algebraic Mesh Quality Metrics", SIAM J. Sci. Comput., Vol. 23, No. 1, 2001.
Knupp, P. M., "Algebraic Mesh Quality Metrics for Unstructured Initial Meshes", Finite Elements in Analysis and Design, Vol. 39, 2003.
Knupp, P. M., "Remarks on Mesh Quality", 45th AIAA Aerospace Sciences Meeting and Exhibit, 2007. Also as Sandia National Laboratories, SAND2007-8128C.
Kreyzig, E., *Advanced Engineering Mathematics*, John Wiley and Sons, 4th Ed., 1979.
Lawry, M. H., "I-DEAS Student Guide", Structural Dynamics Research Corporation, Structural Dynamics Research Corporation, 2000. Can also be found under https://www.scribd.com/doc/46303903/I-DEAS-Student-Guide.
Leonard, A., "Energy Cascade in Large-Eddy Simulations of Turbulent Fluid Flow", Advances in Geophysics, 1974.
Leonardi, S. and I. P. Castro, "Channel Flow Over Large Cube Roughness: A Direct Numerical Simulation Study", J. Fluid Mech., Vol. 651, 2010.

Lesieur, M., O. Metais, and P. Comte, *Large-Eddy Simulations of Turbulence*, Cambridge University Press, 2005.
Lilly, D. K., "A Proposed Modification of the Germano Subgrid-Scale Closure Method", Physics of Fluids A, Vol. 4, No. 3, 1992.
Lorenz, E., "Deterministic Nonperiodic Flow", J. of the Atmospheric Sciences, Vol. 20, 1963.
Modi, A., "Direct Numerical Simulation of Turbulent Flows", Penn State University, 1999.
Moet, H. *et al*., "Wave Propagation in Vortices and Vortex Bursting", *Phys. Fluids*, 1 – 55, 2004.
Moin, P. and K. Mahesh, "Direct Numerical Simulation: A Tool in Turbulence Research", Annu. Rev. Fluid Mech., Vol. 30, 1998.
Nicoud, F. and F. Ducros, "Subgrid-Scale Modeling Based on the Square of the Velocity Gradient Tensor", Flow, Turbulence and Combustion, Vol. 62, 1999.
Nordstrom, J., N. Nordin, and D. Henningson, "The Fringe Region Technique and the Fourier Method Used in the Direct Numerical Simulation of Spatially Evolving Viscous Flows", SIAM, J. Sci. Comput., Vol. 20, No. 4, 1999.
Ostrach, S., "An Analysis of Laminar Free-Convection Flow and Heat Transfer about a Flat Plate Parallel to the Direction of the Generating Body Force", Report 1111, National Advisory Committee for Aeronautics, NASA, 1953.
Pritamashutosh, "Differential Equation of Motion for Steady Compressible Flow", https://pritamashutosh.wordpress.com/2014/02/28/differential-equation-of-motion-for-steady-compressible-flow, 2014. Accessed on September 3, 2018.
Rai, M. M. and P. Moin, "Direct Simulation of Turbulent Flow Using Finite Difference Schemes", J. Comput. Phys., Vol. 96, No. 15, 1991. (Also available as Fluid Mechanics and Heat Transfer, AIAA Paper 89-0369, 1989.)
Rempfer, D., "On Boundary Conditions for Incompressible Navier-Stokes Problems", ASME, Applied Mechanics Reviews, Vol. 59, 2006.
Robinson, J., "CRE Method of Element Testing and the Jacobian Shape Parameters", Eng. Comput., Vol. 4, 1987.
Rodi, W. et al., "Status of the Large Eddy Simulation: Results of a Workshop", Transactions of the ASME, J. Fluids Engineering, Vol. 119, 1997.
Rodriguez, S., "A 4^{th} Order, Implicit, Adaptive Mesh Refinement Algorithm for Simulation of Flame-Vortex Interactions", Master Th., University of New Mexico, 2000.
Rodriguez, S., "Swirling Jets for the Mitigation of Hot Spots and Thermal Stratification in the VHTR Lower Plenum", PhD Diss., University of New Mexico, May 2011.
Rodriguez, S. and D. Ames, "Design Optimization for Miniature Nuclear Reactors", American Nuclear Society, Winter Meeting, November 2015.
Rodriguez, S. and D. Z. Turner, "Assessment of Existing Sierra/Fuego Capabilities Related to Grid-To-Rod-Fretting (GTRF)", Sandia National Laboratories, SAND2012-0530, 2012.
Rudinger, K., "Quantum Computing Is and Is Not Amazing", Sandia National Laboratories, SAND2017-7067C, 2017.
Sanders, M. S. and E. J. McCormick, Human Factors in Engineering and Design, McGraw-Hill Publishing Co., 6^{th} Ed., 1987.
Sengupta, T. K. and S. Bhaumik, *DNS of Wall-Bounded Turbulent Flows: A First Principle Approach*, Springer, 2019.

SDRC, “I-DEAS User's Guide”, Volumes 1 and 2, Structural Dynamics Research Corporation, 1988.
Shaanan, S., J. H. Ferziger, and W. C. Reynolds, “Numerical Simulation of Turbulence in the Presence of Shear”, Report No. TF-6, Dept. of Mechanical Engineering, Stanford University, 1975.
Smagorinsky, J., “General Circulation Experiments with the Primitive Equations, I. The Basic Experiment”, *Monthly Weather Report*, Vol. 91, No. 3, 1963.
Sodja, J., “Turbulence Models in CFD”, University of Ljubljana, 2007.
Spalart, P. R., “Direct Numerical Study of Crossflow Instability”, Laminar-Turbulent Transition, IUTAM Symposium, 1990.
Stefano, G. and O. V. Vasilyev, “Sharp Cutoff Versus Smooth Filtering in Large Eddy Simulation”, Physics of Fluids, Vol. 14, No. 1, 2002.
Stein, O., “Large Eddy Simulation of Combustion in Swirling and Opposed Jet Flows”, Ph.D. Diss., Imperial College London, March 2009.
Stimpson, C. J. et al., “The Verdict Geometric Quality Library”, Sandia National Laboratories, SAND2007-1751, 2007.
Strogatz, S. H., *Nonlinear Dynamics and Chaos*, Westview Press, 1994.
Tabeling, P., *Introduction to Microfluidics*, Oxford University Press, 2009.
Taub, G. *et al.*, “A Numerical Investigation of Swirling Turbulent Buoyant Jets at Transient Reynolds Numbers”, 48th AIAA Aerospace Sciences Meeting, AIAA 2010-1362, Orlando, Florida, January 4-7, 2010.
Terentiev, L., “The Turbulence Closure Model Based on Linear Anisotropy Invariant Analysis”, Universitat Erlangen-Nurnberg, PhD Diss., 2006.
Thompson, K. W., “Time Dependent Boundary Conditions for Hyperbolic Systems”, J. Comp. Phys., Vol. 68, 1987.
Tryggvason, G. and J. Buongiorno, “The Role of Direct Numerical Simulations in Validation and Verification”, University of Notre Dame and Massachusetts Institute of Technology, circa 2013.
Tutar, M. and A. E. Holdo, “Computational Modeling of Flow around a Circular Cylinder in Sub-Critical Flow Regime with Various Turbulence Models”, *Int. J. for Numerical Methods in Fluids*, Vol. 35, 2001.
Tyacke, J. et al., “Large Eddy Simulation for Turbines: Methodologies, Cost and Future Outlooks”, ASME, J. of Turbomachinery, Vol. 136, 2014.
Verstappen, R. W. C. P. and A. E. P. Veldman, “Direct Numerical Simulation of Turbulence at Lower Costs”, J. of Engineering Mathematics, Vol. 32, 1997.
Walther, J. H. *et al.*, “A Numerical Study of the Stability of Helical Vortices Using Vortex Methods”, *J. of Physics*, Conference Series, Vol. 75, 2007.
Wang, G. R., F. Yang, and W. Zhao, “There Can Be Turbulence in Microfluidics at Low Reynolds Number”, Lab on a Chip, 2014.
White, F., *Viscous Fluid Flow*, 2nd Ed., McGraw-Hill, Inc., 1991.
Wilcox, D. C., *Turbulence Modeling for CFD*, 3rd Ed., printed on 2006 and 2010.
Yeung, P. et al., “Turbulence Computations on a 4096^3 Periodic Domain: Passive Scalars at High Schmidt Number and Lagrangian Statistics Conditioned on Local Flow Structure”, 63rd Annual Meeting of the APS Division of Fluid Dynamics, Vol. 55, No. 16, Long Beach, California, 2010.

Yokomine, T. et al., “Experimental Investigation of Turbulent Heat Transfer of High Prandtl Number Fluid Flow under Strong Magnetic Field”, Fusion Science and Technology, Vol. 52, 2007.

You, D. and P. Moin, “A Dynamic Global-Coefficient Subgrid-Scale Eddy-Viscosity Model for Large-Eddy Simulation in Complex Geometries”, *Physics of Fluids*, Vol. 19, No. 6, 2007.

Zigh, G. and J. Solis, “Computational Fluid Dynamics Best Practice Guidelines for Dry Cask Application”, USNRC, NUREG-2152, 2013.