



Power uprates
and plant life extension

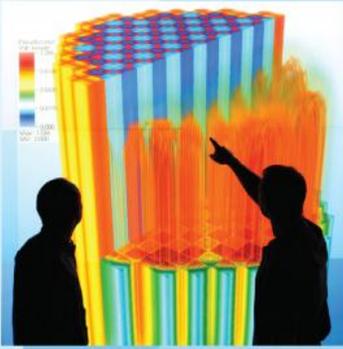


L3:VUQ.VVDA.P2-2.05

William Rider

SNL

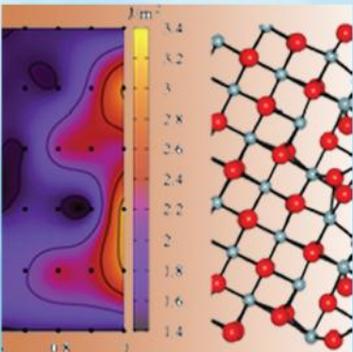
Completed: 7/31/11



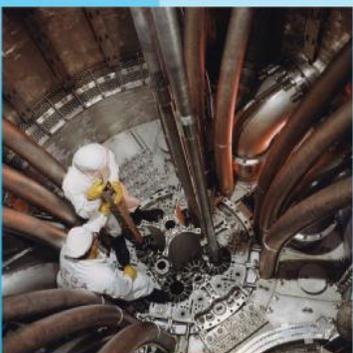
Engineering design
and analysis



Science-enabling
high performance
computing



Fundamental science



Plant operational data



U.S. DEPARTMENT OF
ENERGY

Nuclear Energy

Solution Verification Applied to Drekar and Fuego Calculations using of Grid-to-Rod-Fretting (GTRF)

William J. Rider

Sandia National Laboratories, Albuquerque

With contributions by *WEC* (CASL AMA): R. Lu, A. Mandour, *Fuego/Sierra* (CASL VRI/SNL): D. Turner, S. Rodriguez, R. Bond, R. Summers *Drekar* (CASL, THM/SNL): P. Weber (UNM student), T. Smith, J. Shadid, R. Pawlowski, E. Cyr, *Cubit*, R. Garcia (SNL)

Executive Summary

We have conducted an initial assessment of the numerical error associated with the computational fluid dynamics (CFD) solution of a 3x3 subregion flow through a span of the rod bundle for the purpose of estimating the vibrational excitation of the fuel rod. This phenomenon is known as “Grid-To-Rod-Fretting” or GTRF [Lu11,Wan10]. The flow field will then be utilized to compute the vibrational response of the fuel rod via structural calculations. This is a more high fidelity technique than current capability utilized industrially where substantially simpler models are used. The CFD solutions are computed with two Sandia National Laboratories’ codes, Fuego [Tur11] (VRI, part of the Sierra suite of codes) and Drekar (THM) [Sha11a,Sha11b]. Both codes use the large eddy simulation (LES) modeling technique to provide closure to the fluid equations for the effects of turbulence.

Our results indicate that the grids used for the current simulations provide convergent results, the solutions are currently quite poorly resolved with large 30-50% errors in the pressure drop through the simulated span of the rod bundle. Further, in order to reduce errors significantly relatively enormous grids would need to be utilized, the modeling approach would need to be modified. We also examined the initial lower level validation results achieved with Drekar for vortex shedding from a cylinder at several Reynolds numbers. This is important because the code is relatively immature. On the other hand, Fuego is a mature code with significant verification and validation work preceding this study [Fuego].

The solution verification and concomitant error analysis [Rid11a] shows that a number of issues remain outstanding with the GTRF calculations. These issues are the mesh resolution, mesh optimization, numerical method, and turbulence modeling approach. Another significant issue is the relatively short span of grid simulated, which may necessitate the simulation of significantly larger grid to get complete results. We also examine numerical uncertainty using the standard approach in the verification community, and a new method developed under the auspices of CASL utilizing the statistical uncertainty of the data itself.

Many of the descriptions here are cursory as other reference contain a much more detailed descriptions of the two codes and the GTRF physical problem.

Computational Fluid Dynamics (CFD)

CFD is being increasingly used by the nuclear industry to analyze their systems for a variety of operational scenarios [Ben09,Elm11,Abb10,Kar11,Lah05,Lak02]. The trend is increasing with many industrial users utilizing commercial CFD products, which have grown up through development in civil, aerospace and mechanical engineering applications [CD-Adaptco]. This is contrasted with the historical use of government driven simulation tools in the early years of the nuclear industry. We are seeking to fill this gap. While the CFD tools used are impressive, they are often lacking in the rigor of the quality control over simulations and the sophistication of the underlying algorithms. Furthermore, these tools are not freely available in the same manner as government developed codes.

The CFD codes are usually solving the incompressible Navier-Stokes equations often with turbulence and multiphase flow. The codes vary in approach, but are typically control volume or finite element for discretization with some sort of nonlinear stabilization technique for computing advective transport. Pressure-velocity coupling takes a number of approaches and the overall solution procedure is dominated by variants of the (linearized) SIMPLE algorithm. Overall, the codes are quite advanced, but their application especially in terms of uncertainty quantification is lacking. Numerical error analysis is rarely observed in practice although nothing about the current CFD codes precludes its regular application.

Fuego Description

Fuego is a 3D, incompressible computational fluid dynamics (CFD) code with state-of-the-art turbulence models [Fuego]. It utilizes generalized unstructured meshes using a control-volume-finite element formalism and runs well on massively parallel computers. It includes a variety of Reynolds-averaged Navier-Stokes (RANS) and large eddy simulation (LES) turbulence models. For anisotropic turbulent flow fields with complex mixing and swirling the dynamic Smagorinsky turbulence model [Smag63,Fis01] recommended [Rod10a,Rod10b,Rod10c] and chosen for the work here. The code is currently being developed at Sandia National Laboratories (SNL) as part of a set of strategic, comprehensive codes funded through the Advanced Simulation and Computing (ASC) program. Fuego is a part of the Sierra suite of computational tools.

For spatial discretization, Fuego handles advection in a hybrid fashion with varying mixing of upwind (user choice between first order upwind and second order upwind via a limited MUSCL scheme) and centered approaches. In this context, hybrid refers to the blending between the upwind and centered schemes. For LES simulations, ideally a skew-symmetric operator is commonly employed to preserve kinetic energy conservation in the absence of explicit dissipation. Higher order

upwind methods lack the formal kinetic energy conservation ability as the operator is not skew symmetric. Upwinded approximations produces numerical diffusion and as a consequence, secondary vortices generally do not develop properly during turbulent flow simulations. Thus, a first order upwind method can alleviate oscillations, but may do so at the expense of too much diffusion in the solution. Fuego has undergone extensive validation and verification (V&V) testing on flows that include: conventional jets, swirling jets, jets in cross flow, swirling jets in cross flow, flows around a vertical cylinder, staggered tubes in cross flows, and mixed convection heat transfer to heated horizontal cylinders in cross flow.

Drekar Description

Drekar is an unstructured, fully-implicit, finite element Navier-Stokes solver with the capability to include coupled conjugate heat and mass transfer effects [Sha11a,Sha11b]. It is based on the earlier Charon code, but in relative terms is immature and under considerable recent development. The spatial discretization is currently an equal-order stabilized finite element (SFE) approach that can employ linear or quadratic interpolation. Convection stabilization is achieved through the streamline-upwind-Petrov-Galerkin (SUPG) approach and discontinuity-capturing operators can also be included as part of the stabilized finite element formulation [Sha10b,Sha06]. The temporal discretization is based on a variable-order fully-implicit multi-step backward differentiation (BDF) method that can vary from first-order (a.k.a. Backward Euler) to a fifth-order BDF method (BDF5) in this work the second-order BDF2 is utilized. A parallel fully-coupled nonlinear solution procedure based on an scalable algebraic multilevel preconditioned Newton-Krylov method is used to solve the discretized system of equations [KK04].

The basic turbulence models are developed from a time-averaged Reynolds Averaged Navier- Stokes (RANS) procedure as well as a spatially-filtered large-eddy simulation (LES) approach. In the initial simulations reported on in this study the wall-adapted-large-eddy (WALE) model is used [Smi11]. The WALE model introduced by Nicoud and Ducros [Nic99] is an eddy viscosity model based on the square of the velocity gradient tensor that requires only local data. The results for this study were obtained on runs with 256 to 1024 cores of the SNL Redsky capacity computer. In preparation for this study, Drekar was also ported and run on up to 9600 cores of the ORNL Jaguar platform. Currently a sequence of production runs and scaling runs are planned for the Jaguar system.

Grid-to-Rod-Fretting (GTRF)

Solution of the challenge problems is crucial to the overall CASL effort. This particular problem requires the use of high-resolution temporal and spatial approximations to turbulent single- and multi-phase flow in complex geometries. The aspects of interest in this challenge problem are the high-resolution CFD simulation of turbulent flow to provide unsteady forcing via stresses for use in the GTRF structural dynamics simulations. This problem is characterized by flow

induced rod vibrations that cause erosion of the rod cladding material, and support grids, at points of contact. The vibrations are due to turbulent flow generated at the core inlet and substantially by rod-bundle support grid mixing vanes. Turbulence is desired to enhance heat transfer and prevent localized hot spots from occurring. Like most turbulence, this problem is inherently three-dimensional and unsteady. We hope that a more complete understanding of excitation phenomena through high fidelity CFD simulations can improve clad time-to-failure, and improve reactor core performance while reducing costs.

In the lead up to full-scale simulations, various sub-scale rod-bundle assemblies serving as prototypes will be simulated and analyzed. Separate fluid and structural dynamic simulations will be conducted where the rod excitation predicted by the fluid simulations will be transferred to the structural code through surface boundary conditions. Our initial prototype problem is a turbulent transient flow over the 3x3 rod assembly, with WEC V5H grid spacer, as defined by AMA [Wan10].

The image on the left of Figure 1 shows the center rod surface and the surrounding support grid spacers and mixing vanes. A portion of the geometry is presented in Figure 2 with a specified inflow, a free pressure (stress-free) outflow with periodic boundaries laterally to define an “infinite” lattice of fuel rods and flow channels. The domain spans 17.86 rod diameters with 3.11 diameters upstream of the spacer and 10.74 diameters downstream of the spacer. The original geometry was 54.94 diameters, however, this was shortened to reduce the mesh sizes. The location and orientation of the periodic boundaries are also labeled. The computational grids were generated using Sandia National Laboratories mesh generating software called CUBIT. The geometry was supplied by WEC in the form of a CAD file [Tur11]. Cross-sections of the grids are shown in Figure 3. The meshing strategy consisted of first generating a mesh comprising tetrahedral elements and then sub-dividing each tetrahedral into hexahedral elements. A smoothing algorithm was then applied to the meshes to improve element quality.

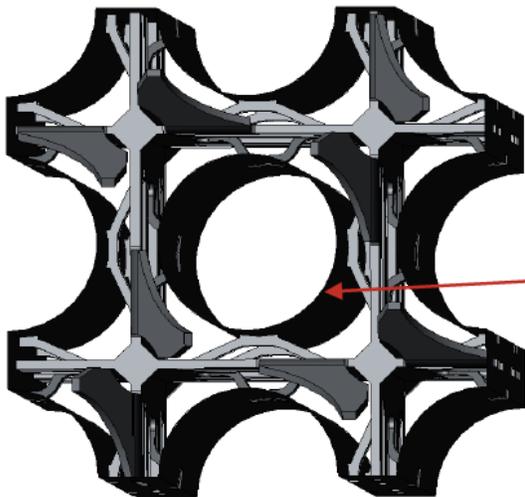


Figure 1. The metal boundaries for the fluid flow in the vicinity of the rod spacers and mixing vanes for the 3x3 geometry.

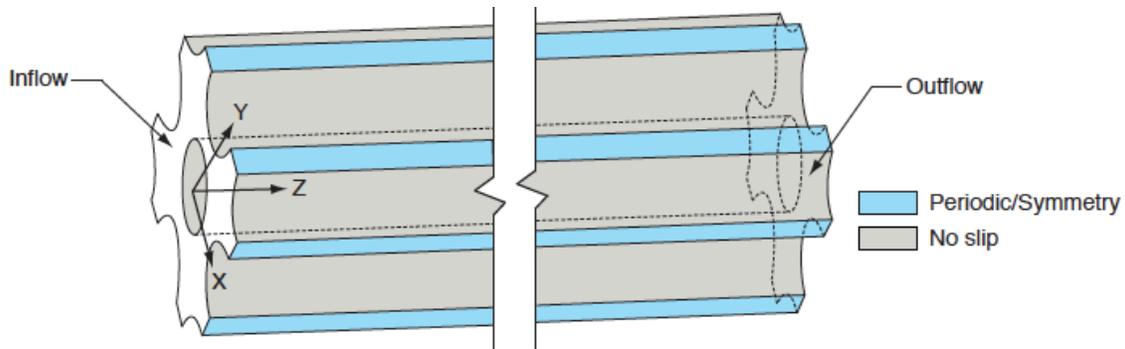


Figure 2. A diagram showing the boundary conditions used with the flow geometry. We note that the Fuego results are presently achieved with symmetry rather than periodic boundary conditions. Periodic boundary conditions will be conducted with Fuego later.

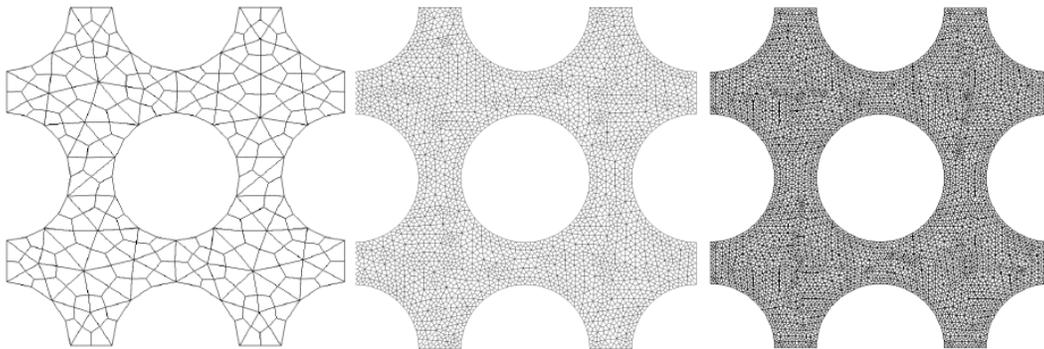


Figure 3. A sequence of coarse to fine meshes (left to right) showing the topology of mesh cells used in the Cubit derived mesh across the 3x3 channel. The coarse mesh corresponds to the 664K element mesh, the medium grid corresponds to the 1912K element mesh, and the fine mesh corresponds to the 5280K mesh.

The problem domain and boundary conditions are designed to model a subset of a rod bundle "ideally" far away from the reactor core walls such that significant cross-flow in the x-y directions can exist. A similar configuration was used by Benhamadouche et al., [Ben10] to study rod excitation using LES. In another study Abbasian et al., [Abb10] used a different rod configuration without the complication of a spacer grid to study rod excitation. At the inflow plane, a uniform velocity is assumed and at the outlet, a stress free condition is imposed. All other external flow surfaces are periodic and all solid surfaces are no-slip (note that the Fuego simulations are done with symmetric conditions laterally). The Reynolds number based on rod diameter and mean inlet velocity was $Re = 192,800$.

Solution Verification and Error Estimation

Calculation or solution verification is a process for the determination of estimates of the discretization or numerical error is estimated in simulations of problems of interest [Roa98,Roa09]. Calculation verification employs techniques of both error estimation and uncertainty quantification (UQ) [Rid11b]. We will apply defined

procedures to define numerical error estimates that are converted to numerical uncertainty estimates. *Code verification* is a related, but distinct process, where the correctness of a software implementation of a numerical algorithm is evaluated, typically by comparison against an exact solution [Rid10].

Numerical methods that are used to obtain approximate numerical solutions of continuum models unavoidably lead to errors in the computed results. These errors are associated with the numerical method *alone* and have nothing to do with any assumptions related to the form of the continuum models (e.g., model-form errors). The challenge of calculation verification is to help provide estimates of such numerical errors. These errors are of four general types: (1) round-off errors, (2) sampling errors, (3) iterative (linear and nonlinear) solver errors, and (4) discretization errors [Rid11a,Roy05,Roy10a,Roy10b].

Solution verification is an activity that is central to the quality of a calculation. The goal of solution verification is to assess the adequacy of a calculation and estimate the error associated with a numerical solution. It is often confused with the term mesh sensitivity study. The two approaches are similar, but the former lacks the quantitative edge offered by the more rigorous verification procedure.

Numerous error estimation techniques exist, but all are formally limited in applicability based on the setting of a problem in a functional setting. Despite these limitations, the process of performing a calculation on a properly posed sequence of discretizations remains a useful approach for estimating numerical error in the broadest set of circumstances. For its generality, we use this approach.

Let us first distinguish the approach of mesh sensitivity from solution verification. In mesh sensitivity, typically several grids are used (often only two total) and the solution is examined for the change in value. For example, “a second finer grid was used and the solution only changed by 1%”. We note that a convergent sequence cannot be established with fewer than three grid resolutions. Furthermore, our results will show that this approach to quality is nothing short of misleading and can lead to incorrect conclusions.

In this report we follow the standard approach to solution verification aside from a single important extension. Many of our results are quantities, which are unsteady and the result is an averaged value in the sense of a time series. This average is also reported with the fluctuations about that average in standard statistical language, that is normal (Gaussian) statistics. The variability is some value proportional to the standard deviation of the result. This is a common, but flawed assumption that will not be challenged here. Using the statistical “envelope” of solutions defined by normal statistics, we produce a concomitant envelope of error estimates defined by the lower and upper set of values reported. Each sequence is computed in the same manner as the convergence of the average values. The overall envelope of results provides a good characterization of the uncertainty in the numerical error.

The workhorse technique for estimating discretization error is systematic mesh refinement (or de-refinement, i.e., coarsening), while the method for estimating iterative error involves systematic changes in stopping criteria for the iteration. A fundamental expectation for a numerical method is the systematic reduction in solution error as, say, the characteristic length scale associated with the mesh is reduced. By the same token, iterative errors are assumed to be smaller as the stopping criterion is decreased in numerical value. For mesh refinement, in the asymptotic limit where the mesh length scale approaches zero, a correct implementation of a consistent method should approach a rate of convergence equal to that defined by numerical analysis (often obtained with the aid of the Taylor series expansion). In practice, this expectation is not typically met particularly as calculations become more realistic and relevant to engineered systems, i.e., calculations might not be in the asymptotic range. *This circumstance does not obviate the need for some estimate of the numerical error; in fact the necessity may be increased under these conditions.*

To conduct analysis using this approach, a sequence of grids with different intrinsic mesh scales is used to compute solutions and their associated errors. The combination of errors and mesh scales can then be used to evaluate the observed rate of convergence for the method in the code. In order to estimate the convergence rate, a minimum of two grids is necessary (giving two error estimates, one for each grid). The convergence tolerance for iterative solvers can be investigated by simple changes in the value of the stopping criteria. Assessing iterative convergence is complicated by the fact that the level of error is also related to the mesh through a bounding relation in which the error in the solution is proportional to the condition number of the iteration matrix. Most investigations of iterative solver error only consider the impact of the stopping criteria alone.

Ideal Asymptotic Convergence Analysis

In this section, we examine the case of ideal asymptotic convergence analysis. The axiomatic premise of asymptotic convergence analysis is that the computed difference between the reference and computed solutions can be expanded in a series based on some measure of the discretization of the underlying equations. Taking the spatial mesh as the obvious example, the ansatz for the error in a 1-D simulation is taken to be

$$\|g^f - g^c\| = A_0 + A_1(\Delta x)^\alpha + o((\Delta x)^\alpha) \quad (1)$$

In this relation, g^f is the reference solution, which for calculation verification is computed on a refined mesh, g^c is the computed solution, Δx is some measure of the mesh-cell size, A_0 is the zero-th order error, A_1 is the first order error, and the notation " $o((\Delta x)^\alpha)$ " denotes terms that approach zero faster than $(\Delta x)^\alpha$ as $\Delta x \rightarrow 0^+$. For consistent numerical solutions, A_0 should be identically zero; we assume this to be the case in the following discussion. For a consistent solution, the exponent α of Δx is the convergence rate: $\alpha = 1$ implies first-order convergence, $\alpha = 2$ implies second order convergence, etc.

Assume that the calculation has been run on a “coarse” mesh (subscript c), and a very course mesh (subscript vc) characterized by Δx_c (Δx_{vc}), which we hereafter also denote as Δx . The error ansatz implies:

$$\|g^f - g^c\| = A_0 + A_1 (\Delta x)^\alpha + \dots \quad (2)$$

Let us further assume that we have computational results on a “fine” mesh Δx_f (subscript f), where $0 < \Delta x_f < \Delta x_c$ with $\Delta x_c / \Delta x_f \equiv \sigma > 1$. In this case, the error ansatz implies:

$$\|g^f - g^c\| = \sigma^{-\alpha} A_1 (\Delta x)^\alpha + \dots \quad (3)$$

Manipulation of these two equations leads to the following explicit expressions for the quantities α and A_1 :

$$\alpha = \left[\log \|g^f - g^c\| - \log \|g^c - g^{vc}\| \right] / \log \sigma \quad (4)$$

$$A_1 = \|g^f - g^c\| / (\Delta x)^\alpha \quad (5)$$

These two equalities are the workhorse relations that provide a direct approach to convergence analysis as a means to evaluating the order of accuracy for code verification.

For quantities of interest (QOIs) or figures of merit (FOMs) the above development can be utilized without resorting to error norms. For our report here this is the approach taken here. We note that the lack of functional analytical setting weakens the expectations for formal convergence, but most QOIs or FOMs are closely related to some sort of normed quantity. The quantity, G , is defined without the use of a norm with the following related error model,

$$\tilde{G} = G^k + A_1 (\Delta x)_k^\alpha + \dots \quad (6)$$

with the remainder of the development proceeding as above provided the approach toward \tilde{G} , the mesh converged solution, is monotonic. In the case where a solution is not monotonically approached, the above error model can still be utilized as long as the error in absolute terms is diminishing monotonically. This recommendation is in some clear opposition to the existing literature although error norms themselves are typically positive definite quantities.

Non-Ideal Asymptotic Convergence Analysis

The term “non-ideal” refers to situations for which the error does not monotonically decrease with increasing mesh refinement in the simulation: the computed solutions may diverge under mesh refinement, in either a monotonic or non-monotonic fashion, or the computed values may have no overarching characteristics beyond simply being bounded. Some researchers contend that one cannot make further statements about calculations that do not exhibit the ideal asymptotically convergent behavior described in the previous section. Insofar as such non-ideal behavior is often seen in practical engineering simulations, however, this section contains a discussion of the analysis of such non-ideal cases.

The salient parameters in these analyses are the theoretical convergence rate of the underlying numerical scheme, α_{th} , and the observed convergence rate, based on Richardson Extrapolation (see, e.g., [Obe10]), corresponding to the actual simulation results, α_{RE} . A study of this approach is given by Xing and Stern [Xin10], who build upon earlier work of Eça and Hoekstra [Eça06].

For the estimation of convergence properties of scalar quantities, Eça and Hoekstra [Eça06] approach the problem of estimating the observed convergence rate as a least-squares solution to the governing error ansatz equation,

$$|G^f - G^c| = A_1 (\Delta x)^\alpha + \dots \quad (7)$$

Let p denote the convergence rate corresponding to the least-squares solution for this equation based on the *signed* differences of results computed with adjacent grid spacings, i.e., where both positive and negative values are allowed on the left-hand side of Eq. (9). Similarly, let α^* represent the convergence rate corresponding to the *absolute* differences of results computed with adjacent grid spacings, i.e., where absolute values of the left-hand side of Eq. (9) are used. Then, the behavior of the set of computations is determined from the following algorithm:

```
If  $\alpha > 0$  then monotonic convergence
Else if  $\alpha < 0$  then monotonic divergence
Else if  $\alpha^* < 0$  then oscillatory divergence
Else oscillatory convergence
```

A second way to characterize the nature of the convergence is to examine the ratio of signed error in the solution from one mesh refinement level to the next,

$$R = (G^f - G^c) / (G^c - G^{vc}) = \Delta^{cf} / \Delta^{vc}, \quad (10)$$

```
If  $R < 1$  then monotonic convergence
Else if  $R > 1$  then monotonic divergence
Else if  $R < -1$  then oscillatory divergence
Else  $(-1 < R < 0)$  oscillatory convergence
```

Estimation Of Associated Numerical Uncertainty

Once the nature of the solution has been properly categorized, the numerical uncertainty can then be estimated as part of the overall uncertainty estimate.¹ The Grid Convergence Index (GCI) of Roach (see [Roa98, Roa09]) is perhaps the original attempt to codify the numerical uncertainty associated with inferred convergence parameters. Roache [Roa98] claims that there is evidence for the numerical uncertainty based on the GCI method (with a safety factor of 1.25) to achieve a “95% confidence level.” This approach was extended to the Correction Factor (CF)

¹ The proceedings of the 1st, 2nd, and 3rd Workshops on CFD Uncertainty Analysis [Eça08]

method of Stern et al. [Ste01]. Xing and Stern [Xin10], however, take issue with both of these approaches, stating, “...there is no statistical evidence for what confidence level the GCI and CF methods actually achieve” and, more specifically, that their analyses “...suggest that the use of the GCI₁ method is closer to a 68% than a 95% confidence level.” As we describe below, Xing and Stern come to a different conclusion regarding an approach that does meet the 95% confidence level measured empirically on varied computational applications.

To evaluate the numerical uncertainty associated with these solution verification estimates, Xing and Stern performed a statistical analysis of 25 sets of computational data, covering a range of fluid, thermal, and structural simulations, to arrive at various parameters for their estimations of simulation uncertainty. The parameters obtained by Xing and Stern provide computational uncertainty estimates that demonstrably satisfy the 95% confidence level for the data sets upon which that analysis is based. They suggest that the formula below provides a safety factor with empirical, statistical support. We suggest following this approach whenever the grid sequence provides a convergent sequence.

$$U_{num} = FS|\delta_\alpha| = \begin{cases} (2.45 - 0.85P)|\delta_\alpha|, & \text{if } 0 < P \leq 1 \\ (16.4P - 14.8)|\delta_\alpha|, & \text{if } P > 1 \end{cases} \quad (9)$$

where $P = \alpha_{RE}/\alpha_{th}$ defines whether the solution is asymptotic in observed nature. The numerical error magnitude comes from the Richardson extrapolation toward the monotonically mesh converged solutions as

$$\delta_\alpha = \frac{G^f - G^c}{\sigma^\alpha - 1} \quad (10)$$

or the related error estimate for monotonically decreasing error as

$$\delta_\alpha = \frac{|G^f - G^c|}{\sigma^\alpha - 1} \quad (11)$$

In the case where the solution is not convergent, the numerical uncertainty should nonetheless be estimated, however rough those estimates may be. It is the authors' experience that users of codes will generally move forward with calculations and—without guidance to the contrary—may offer *no* numerical uncertainty estimates whatsoever. We maintain that this practice is potentially more dangerous than providing a weakly justified estimate. We offer the important caveat that this bound is not rigorously justified; instead, it is perhaps appropriately viewed as a heuristic estimate that can be produced given limited information. The simplest approach is to examine the range of solutions produced and multiply this quantity by a generous safety factor,

$$U_{num} = 3(\max G - \min G) \quad (12)$$

The safety factor, set to 3 in (14), might assume different values in different computational science applications. This heuristic approach is similar to that advocated by Eça and Hoekstra [Eça06].

The errors associated with iterative (linear and nonlinear) solution of systems of equations can be approached similarly. We note that many investigations have sought to merely render these errors too small to be significant. While such an approach can work, it may not be generally possible in practice. A more complete approach is to quantify these errors in terms of the controllable numerical parameters, usually the stopping criteria for such iterations.

Detailed Workflow

Here, we expand on the details of the proposed workflow. There are other workflow approaches (see, e.g., [Joh06]), and the steps described below are by no means exhaustive. The proposed steps do, however, standardize a calculation verification workflow that can be conducted by a code team (developers and testers) for the purpose of estimating numerical uncertainty. Ideally, the code verification process should be conducted regularly (as well as on demand) so that incorrect implementations impacting mathematical correctness are detected as soon as possible. The general consensus in software development is that the cost of bugs is minimized if they are detected as close as possible to their introduction.

This procedure assumes that the code team is using a well-defined software quality assurance (SQA) process, and the code verification is integrated with this activity. Such SQA includes source code control, regression testing, and documentation, together with other project management activities. For consistency and transparency, we recommend performing the code verification in the same manner and using the same type of tools as other SQA processes.

1. *Starting with an implementation (i.e., code) that has passed the appropriate level of SQA and code verification scrutiny, choose the executable to be examined.* Calculation verification can be a resource-intensive activity involving substantial effort to perform. Calculation verification should be applied to the same version of the code that analysts would use for any important application. The notion that verification and validation should be applied to the same code is important to keep in mind. This process should be applied to the specific version of the code used throughout the entire V&V UQ activity.
2. *Provide an analysis of the numerical method as implemented including accuracy and stability properties.* The analysis should be conducted using any one of a variety of standard approaches. Most commonly, the von Neumann-Fourier method could be employed. For nonlinear systems, the method of modified equation analysis can be used to define the expected rate and form of convergence. The form and nature of the solution being sought can also influence the expected behavior of the numerical solution. For example, if the solution is discontinuous, the numerical solution will not achieve the same order of accuracy as for a smooth solution. Finite element

methods can be analyzed via other methods to define the form and nature of the convergence (including the appropriate norm for comparison).

3. *Produce the code input to model the problem(s) for which the code verification will be performed.* Each problem is run using the code's standard modeling interface as for any physical problem that would be modeled. It can be a challenging task to generate code input that correctly specifies a particular problem²; e.g., special routines to generate particular initial or boundary conditions that drive the problem may be required, and these routines must be correctly interfaced to the code. It is advisable to consider the complexities and overhead associated with such considerations prior to undertaking such code verification analyses.
4. *Select the sequence of discretizations to be examined so each solution.* Verification necessarily involves convergence testing, which requires that the problem be solved on multiple discrete representations (i.e., grids or meshings). This is consistent with notions associated with h -refinement, although other sorts of discretization modification can be envisioned. The mathematical aspects of verification are typically most conveniently carried out if the discretizations are factors of two apart.
5. *Run the code and provide of means of producing appropriate metrics to compare the numerical solutions. The solutions to the problem are computed on the discretizations. Most commonly and as discussed above, these metrics take the form of norms (i.e., p -norms such as the L_2 or energy norm). The selection of metrics is inherently tied to the mathematics of the problem and its numerical solution. The metrics can be computed over the entire domain, in subsets of the domain, on surfaces, or at specific points. The domain over which the metrics are evaluated and the analysis to be conducted must be free of any spurious solution features (due, e.g., to numerical waves erroneously reflected from computational boundaries).*
6. *Use the comparison to determine the sequence of errors in the discretizations.* Using the well-defined metrics for each solution, the error can be computed for each discrete representation. Ideally, there will be a set of metrics available, providing a more complete characterization of the problem and its solution.
7. *The error sequence allows the determination of the rate-of-convergence for the method, which is compared to the theoretical rate.* With a sequence of errors in hand, the demonstrated convergence rate of the code for the problem is estimated. The theoretical convergence rate of a numerical method is a key property. Verification relies upon comparing this rate to the demonstrated rate of convergence. Evidence supporting verification is provided when the

demonstrated convergence rate is consistent with the theoretical rate of convergence. This can be a difficult inference to draw, because the theoretical rate of convergence is a limit reached in an asymptotic sense, which cannot be attained for any finite discretization. As a consequence, there are unavoidable deviations from the theoretical rate of convergence, to which judgment must be applied.

8. *Using the results, render an assessment of the method's implementation correctness.* Based on the discrete solutions, errors, and convergence rate, a decision on the correctness of a model can be rendered. This judgment is applied to a code across the full suite of verification test problems.
 - a. The assessment can be positive, that is, the convergence rate is consistent with the method's expected accuracy.
 - b. The assessment can be negative, that is, the convergence rate is inconsistent with the method's expected accuracy.
 - c. The assessment can be inconclusive, that is, one cannot defensibly demonstrate clearly uniform consistency or inconsistency with the method's expected accuracy. For example, the convergence rate is nearly the correct rate, but the differences between the expected rate and the observed rate is uncomfortably large, potentially indicating a problem.

Figures 1 show the entire process in diagrams that conceptually show how the work flows. This process should be repeatable and available on demand. As noted in the introduction to this section, having the code verification integrated with the ongoing SQA activity and tools can greatly facilitate this essential property. The solution verification process is not monolithic, but instead it is flexible and should meet the needs of the specific application.

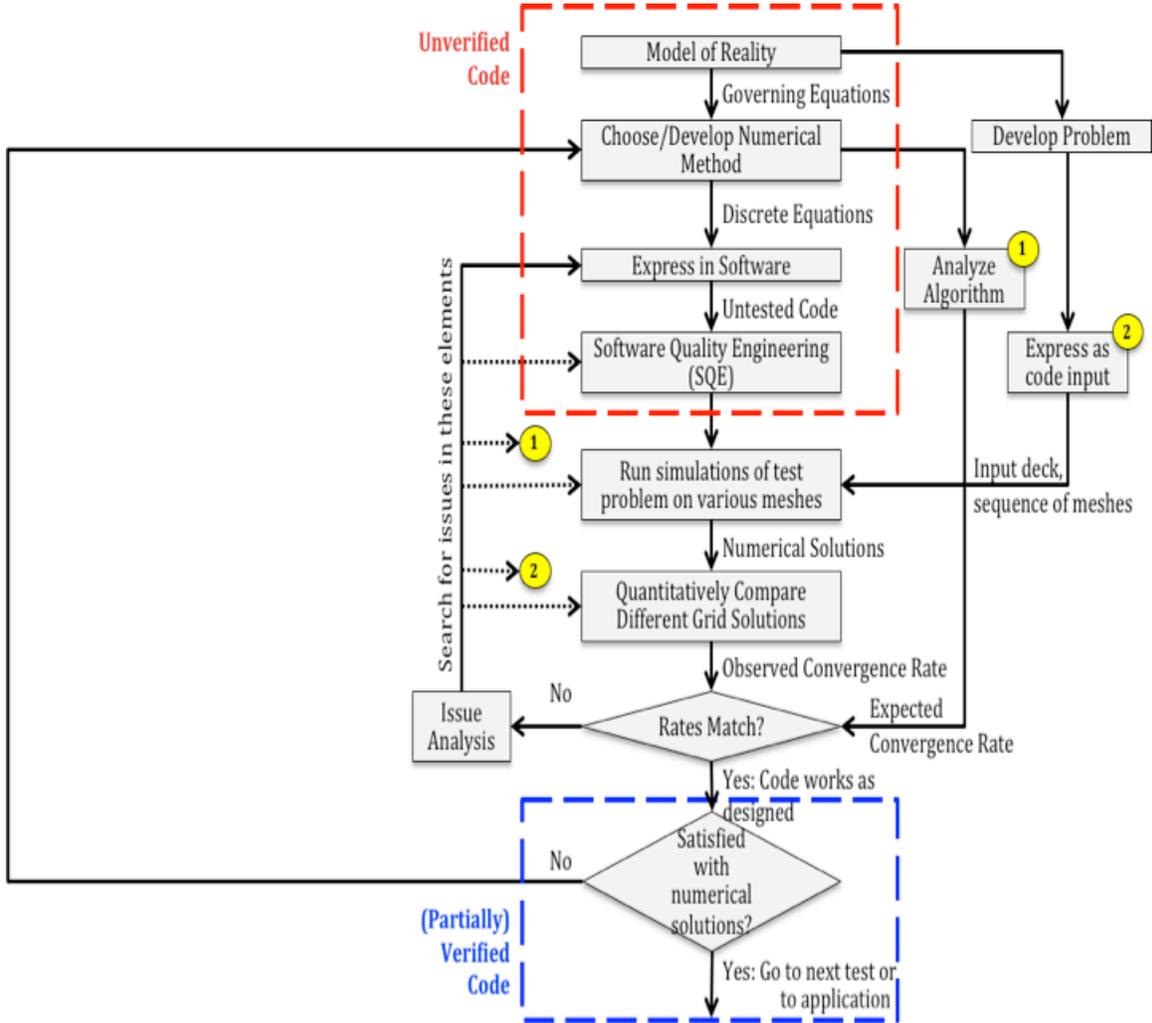


Figure 4. The flowchart version of the list of activities is shown for code verification, which can be interpreted as an expansion of the simple expression of this activity.

We will provide a detailed example using contrived data to illustrate this approach. In Table 1 we give some data to help illustrate our approach. The data shows a convergent sequence of calculations converging at first order along with a standard deviation of averages for several different cases. We hope to illustrate the sort of results that will be observed in practice.

Grid	Average	Error Case 1	Error Case 2	Error Case 3	Error Case 4	Error Case 5
4 h	8	±0	±1/2	±1	±1	±1/2
2 h	6	±0	±1/2	±1/2	±1/2	±1
h	5	±0	±1/2	±1/4	±1/16	±3/2

Table 1. The idealized data used in our demonstration example for convergence with statistically defined data.

We now set out to compute the convergence rate of three sequences for each case: the average (the same for all cases here), the lower boundary (the average minus error), and the upper boundary (the average plus the error). The error may be given in terms of a standard deviation, or a confidence interval, or simply an interval. We will give the results in terms of the estimated mesh converged solution, with lower and upper bounds and the level of numerical error associated with each on the finest grid, “h.” The average solution for the data in Table 1 behaves as,

$$G(\Delta x) = a + b(\Delta x)^c = 4 + \left(\frac{1}{h}\right)(\Delta x)^1 \tag{13}$$

for all cases. We summarize the results for the upper and lower bounds of the error in Table 2.

Case	Lower a	Lower b	Lower c	Upper a	Upper b	Upper c
1	4	1/h	1	4	1/h	1
2	3.5	1/h	1	4.5	1/h	1
3	4	0.75/h	1	4	1.25/h	1
4	3.42857	1.50893/h*	0.62148	4.16000	0.90251/h*	1.2115
5	1.25	2.25/h*	0.73697	6.25	0.25/h*	1.5850

Table 2. The results of the idealized error envelope example for the solutions defined by the data in Table 1. The “*” denotes that the exponent on “h” should equal the value of “c” in the adjacent column to assure dimensional consistency.

The first case is trivial, and the second case behaves as expected with the error envelope being constant across the meshes. When the error does not behave in the same manner as the average the results begin to become interesting. This is explored in the last three cases. In case 3 the error converges just as the average hence vanishes as expected, but the form of the error varies depending on the mesh. In case 4 the error converges faster than the average. This results in differing convergence behavior and uncertainty in the rate of convergence. Note that the errors are reported as symmetric about the average, but the lower and upper

bounds are unsymmetric. In case 5, the error becomes larger under mesh refinement and the upper and lower bounds expand the uncertainty tremendously (as it should). The constants in the convergence analysis become quite uncertain. This approach should provide an appropriate representation of the uncertainty given the reporting of statistically derived quantities (or other quantities with defined error, or fluctuations).

In case 4, the bounds of the solution can be described in a single equation as

$$A(h) = [3.42857, 4.16000] + [0.90251, 1.50893] h^{[0.62148, 1.2115]} \quad (13)$$

where the brackets, [a,b] show the upper and lower bounds of for the error equation.

Similarly, case 5 has a similar form,

$$A(h) = [1.25, 6.25] + [0.25, 2.25] h^{[0.73697, 1.5850]} \quad (14)$$

Drekar Validation and Error Estimates

Because Drekar is a relatively new code, we decided to proceed with a validation study on a phenomena related to GTRF. Specifically the vortex shedding from the mixing vanes and grid spacers is essential to simulate effectively. For this reason, the vortex shedding frequency associated with the classical flow over a right circular cylinder was examined. This classical problem has been examined at a large range of Reynolds number, and its solution should provide a confidence building exercise for Drekar. The vortex shedding is characterized by the Strouhal number, which is defined by $St = (d \lambda) / U$, where U is the flow velocity, d is the cylinder diameter, and λ is the frequency of vortex shedding (the Reynolds number $Re = d U / \nu$, where ν is the viscosity is the other dimensionless quantity of importance). The classical experimental studies can be found in Schlichtings text and we reproduce them in Table 3.

Flow	Re=60	Re=100	Re=400	Re=1000
St	0.136±0.006	0.164±0.005	0.205±0.005	0.212±0.006

Table 3. The relevant Strouhal numbers from classical experiments with error bars estimated from the plots of the historical data [Sch79].

Drekar was used with second-order time and space differencing (with linear interpolation and SUPG) on a sequence of grids. For all calculations shown here we should expect second-order results for mesh-converged global quantities. We measured the grid scale by the RMS length of the mesh. We note that this grid is far more regular and classical in structure than the highly irregular mesh used for the GTRF calculations.

$$St(h, \Delta t) = St_0 + C_h h^\alpha + C_{\Delta t} (\Delta t)^\beta$$

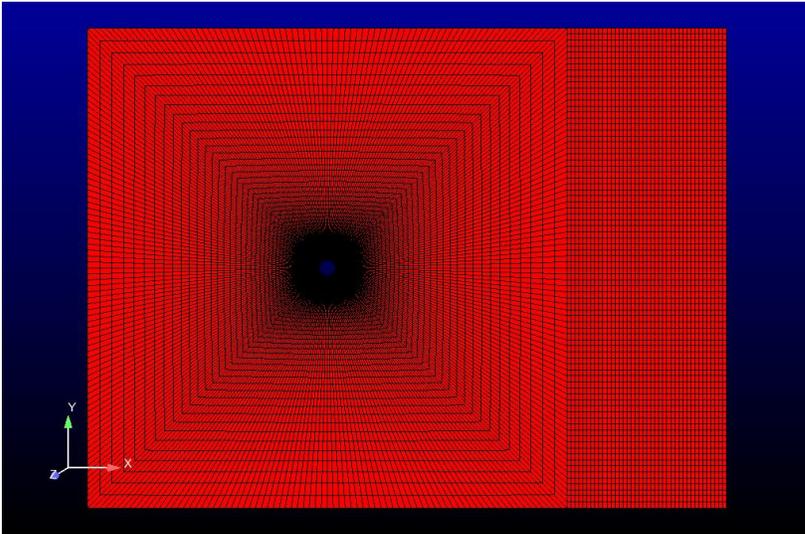


Figure 5: A snapshot of the grid used for the vortex shedding simulations.

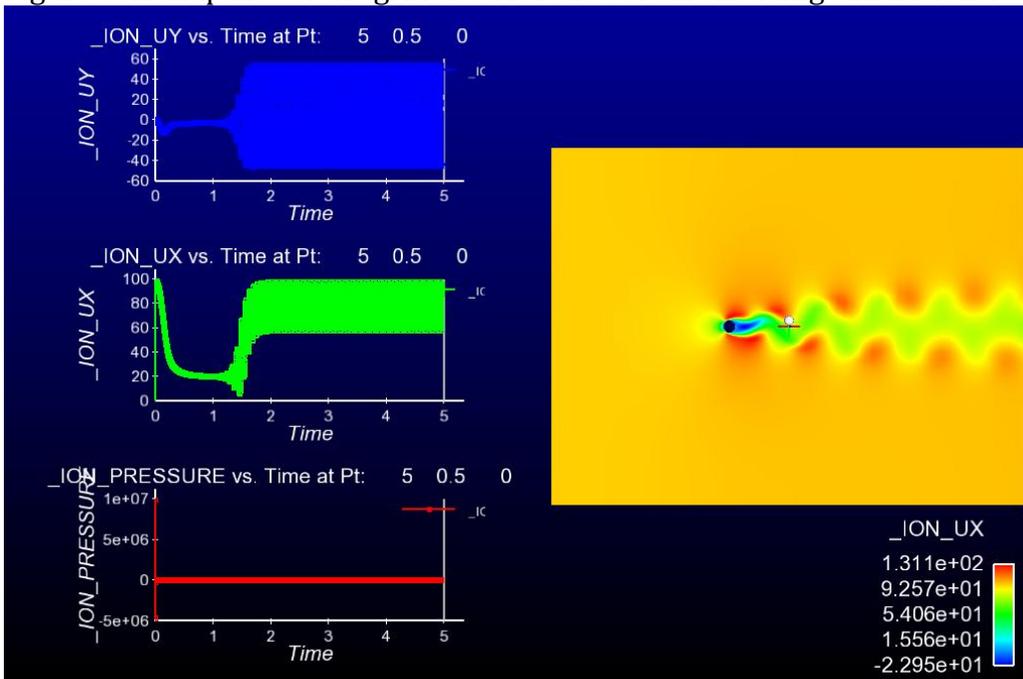


Figure 6: A snapshot of the vortex shedding simulation showing the flowfield, and several of the variables plotted in the form of a time series.

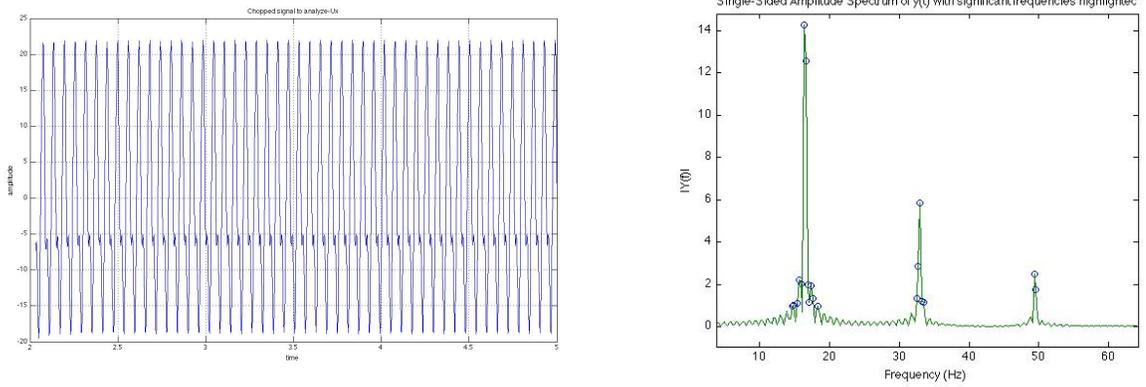


Figure 7: Data reduction is accomplished the FFT of the fluctuating velocity at a location in space and reporting the frequency where the FFT has its largest peak.

Case	Δt	RMS h	St	$\Delta(St)$ 95%
1	0.002	0.054111988	0.110474853	0.001219
2	0.002	0.023801688	0.152492294	0.004537
3	0.002	0.010786082	0.164777976	0.004979
4	0.002	0.005264375	0.165127187	0.004369
5	0.008	0.010786082	0.134265285	0.005306
6	0.004	0.010786082	0.154923058	0.00756
7	0.001	0.010786082	0.165999038	0.00491
8	0.0005	0.010786082	0.1661293	0.005012

Table 4. Drekar results for vortex shedding at $Re=100$.

The average Strouhal number estimate has the following structure

$$St(h, \Delta t) = 0.170935 - 7.94234h^{1.69637} - 47.434(\Delta t)^{1.50502} \quad (15a)$$

For the lower bounding estimate given statistically, the convergence is

$$St(h, \Delta t) = 0.167213 - 5.83905h^{1.61558} - 21.7375(\Delta t)^{1.33337} \quad (15b)$$

and the upper bounding estimate is

$$St(h, \Delta t) = 0.17470 - 10.3214h^{1.76271} - 164.206(\Delta t)^{1.77237} \quad (15c)$$

Overall, the solutions indicate that the code is performing well for this problem although the meshes are a bit under-resolved. Furthermore, the solution is converging to a solution that is 1.8 to 6.5% too high. This may be related to the size of the domain or three-dimensional effects neglected in the simulation.

In this case it make sense to recomputed the convergence and include only the five finest resolutions in time and space in the convergence rate. We will find that this produces a significantly lower level of numerical uncertainty. The results for this application of our procedure is

$$St(h, \Delta t) = 0.168561 - 190.70h^{2.5584} - 5.7963(\Delta t)^{1.2418}$$

for the average Strouhal number, and

$$St(h, \Delta t) = 0.162377 - 1726.87h^{3.1575} - 474.21(\Delta t)^{2.0380}$$

$$St(h, \Delta t) = 0.171947 - 15428h^{3.7435} - 1774.7(\Delta t)^{2.2061}$$

for the lower and upper bound respectively. The lower bounds value is almost identical to the experimental observed value, and range of asymptotic Strouhal numbers is smaller with higher rates of observed convergence. This would indicate that the coarser meshes and larger time steps are insufficient to resolve the problem. The UQ estimate computed using these values is quite small with a value of 0.36% of the Strouhal number. This is smaller than the range supported by the statistically bounded values. We suggest taking the larger reported bound as the uncertainty, which evaluates to 5.83% of the Strouhal number.

Case	Δt	RMS h	St	$\Delta(St)$ 95%
1	0.002	0.054111988	0.125325559	0.001116
2	0.002	0.023801688	0.19155126	0.00093
3	0.002	0.010786082	0.223538936	0.000943
4	0.002	0.005264375	0.230566339	0.003085
5	0.008	0.010786082	0.168450246	0.001905
6	0.004	0.010786082	0.204941286	0.002198
7	0.001	0.010786082	0.238016281	0.005241
8	0.0005	0.010786082	0.240079932	0.002972

Table 5. Drekar results for vortex shedding at Re=1000.

The average Strouhal number estimate has the following structure

$$St(h, \Delta t) = 0.266514 - 3.26418h^{1.12447} - 26.2295(\Delta t)^{0.805668} \quad (16a)$$

For the lower bounding estimate given statistically, the convergence is

$$St(h, \Delta t) = 0.259479 - 3.60563h^{1.16818} - 39.2129(\Delta t)^{0.866885} \quad (16b)$$

and the upper bounding estimate is

$$St(h, \Delta t) = 0.273924 - 2.97891h^{1.08355} - 18.1265(\Delta t)^{0.748689} \quad (16c)$$

Overall, the solutions indicate that the code is not performing well for this problem for the likely reason that the time and space mesh is quite under-resolved. This is a pre-cursor to the GTRF results shown next. To make that point, the solution is converging to a solution that is 22 to 29% too high. Additionally, this may be related to the size of the domain or three-dimensional effects neglected in the simulation.

We apply the same procedure as we did for the Re=100 results and remove the coarsest discretizations from the results and redo the convergence analysis. This results in the following results

$$St(h, \Delta t) = 0.252791 - 235.18h^{2.3204} - 12621(\Delta t)^{1.4870}$$

$$St(h, \Delta t) = 0.251719 - 220.12h^{2.3134} - 285.10(\Delta t)^{1.0562}$$

$$St(h, \Delta t) = 0.256687 - 1196.0h^{2.7508} - 1951.1(\Delta t)^{1.2663}$$

for the average, lower and upper bounds respectively. We then can establish our two estimates on the uncertainty, one using the convergence for mean Strouhal number and the second using the reported confidence interval's convergence. The reported interval gives an uncertainty of 1.97% of the value of the Strouhal number. The mean value's uncertainty estimate as given in the "Solution verification procedure" document [Rid11a] is 3.57%. Here, the two values reported are similar and our confidence is bolstered. Given these relatively small values we can express better confidence in the calculations. We note that the converged mean value is 19.2% from the experimental value, and the experimental value is not bounded by the converged lower and upper confidence intervals.

Based on the convergence analysis the space-time errors are almost identical in magnitude for both Reynolds number. The uncertainty estimates are different in character. For Re=100 the uncertainty estimates for space and time are nearly identical in size ($\Delta St = 0.00025$ and 0.00023 respectively). For Re=1000, the space uncertainty estimates are three times larger than the time estimate ($\Delta St = 0.00149$, 0.00058) indicating a problem in using the same grids for both Reynolds numbers. Applying these estimates to the solutions does not bring them into agreement with the experimentally observed values.

These calculations have only scratched the surface of the results although the results did secure a greater faith in Drekar's capabilities. The issues of mesh resolution and fundamental numerical methodology are not fully explored. Furthermore, the behavior of the vortex shedding for turbulent flows is unexamined at this point.

GTRF Error Estimation

The results from the initial Fuego and Drekar calculations are limited due to several issues although we expect further analysis to define more quantities to examine. For now, the single quantitative result that behaves reasonably for both codes is the pressure drop through the computational domain. It provides a good reference point for a comparison between the code and a starting point for validation to follow. Table 6 gives the data used for the analysis that follows.

For Fuego the analysis gives the following somewhat astonishing result. The convergence rate is absurdly high,

$$\Delta p(h) = 24.34 + 6.10 \times 10^{18} h^{15.85} \quad (17)$$

which is a warning sign given the methods utilized by the code, first-order in time and second(upwind)-order in space. It would imply that the solution is converged,

but a more accurate implication is significant doubt. Our uncertainty analysis below will indicate precisely that.

For Drekar the results look more reasonable in terms of the convergence rate,

$$\Delta p(h) = a + bh^c = 17.42 - 163.7h^{1.234} \quad (18)$$

indicating a solution requiring significantly greater resolution, but a convergence rate that is low, but reasonable for this challenging circumstance.

Mesh	Fuego	Drekar
Coarse	31.8 kPa	26.7 kPa
Medium	24.6 kPa	23.8 kPa
Fine	24.4 kPa	22.0 kPa

Table 6. Pressure drop through the grid span computed with both Fuego and Drekar for three grids. We note that the coarse, medium and fine grids are not the same for the two different codes.

Uncertainty estimation is a quantitative and more precise manner to give these conclusions. For Fuego, $\Delta p = -6.6$ kPa for the uncertainty or 27.1% of the asymptotic value. This results from the very high rate of convergence, which produces a larger multiplier on the asymptotic error estimate. For Drekar $\Delta p = -8.9$ kPa or 51.1% of the asymptotic value; however, its convergence rate is low, but reasonable given the methods used.

The calculations are similarly reliable as determined by the uncertainty analysis and quite under-resolved. Either far greater mesh density, or a more optimal mesh is called for. The results are too preliminary in nature to yield any conclusions on the differences between methods and models in the two codes. Greater mesh resolution and a more comprehensive set of metrics would provide a much clearer picture,

Summary

We have delivered a preliminary verification analysis of CFD calculations conducted with CFD codes Fuego, and Drekar. In both cases, the verification results indicate that the need for greatly refined meshes to reduce computational errors to an acceptable level. For GTRF the estimated error for the pressure drop is 30-50% of the mesh converged value. Comparison with experimental data is currently pending.

The validation case with Drekar for vortex shedding at $Re=100$ is adequately resolved with uncertainties of less than 5% of the correct experimentally observed shedding frequency. For the $Re=1000$ case the spatial mesh needed to be refined

considerably. The discrepancy between the mesh converged result and the experimental results need to be examined more completely to determine whether modeling details are ultimately responsible. Potential causes of the differences may be a truncated domain or suppressed three-dimensional features of the flow.

An assessment of our results here against our proposed workflow.

1. *Starting with an implementation (i.e., code) that has passed the appropriate level of SQA and code verification scrutiny, choose the executable to be examined.*
This is true for Fuego, but not strictly for Drekar. It is a direct reflection of the maturity of the respective codes.
2. *Provide an analysis of the numerical method as implemented including accuracy and stability properties.*
With the available evidence both codes appear to be stable and accurate as the developer assert.
3. *Produce the code input to model the problem(s) for which the code verification will be performed.*
This has been completed and documented in other reports on the CFD calculations themselves.
4. *Select the sequence of discretizations to be examined so each solution.*
A sequence of calculation using either three or four grids was computed for each code.
5. *Run the code and provide of means of producing appropriate metrics to compare the numerical solutions. The solutions to the problem are computed on the discretizations.*
The code has been run on each grid and preliminary metrics have been reported.
6. *Use the comparison to determine the sequence of errors in the discretizations.*
These errors were estimated and discussed in the previous section of this report.
7. *The error sequence allows the determination of the rate-of-convergence for the method, which is compared to the theoretical rate.*
The rate of convergence was estimated above and for Fuego was greater than the expected rate, and for Drekar was lower than the expected rate.
8. *Using the results, render an assessment of the method's implementation correctness.*
The codes appear to be correct insofar as the current results indicate.

Recommendations for Further Study

- More metrics produced from the calculations: velocity, TKE, dissipation, turbulent statistics
- Determine the mesh requirements

- Significant SA/UQ studies of modeling of GTRF, determine the relative importance of various modeling parameters
- Test URANS on GTRF. Our current meshes are too coarse for LES, we have no inertial range resolution
- LES needs significantly finer meshes, and/or a mesh that more biased towards the boundary layers
- Numerical impact of spatial and temporal integration method
- Impact of the LES model and its parameters
- Test the applicability of ILES

Acknowledgement

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000

References

- [Abb10] F. Abbasian, S. D. Yu, J. Cao, Large Eddy Simulation of Turbulent Axial Flow Along an Array of Rods, *Journal of Fluids Engineering* 132 (2010) 1–11.
- [Ben09] S. Benhamadouche, P. Moussou, and C. Le Maitre. CFD estimation of the flow-induced vibrations of a fuel rod downstream a mixing grid. *Proceedings of the ASME 2009 Pressure Vessels and Piping Division Conference*, PVP2009-78054, July, 2009.
- [CD-Adapco] CD-Adapco, STAR-CCM++, http://www.cd-adapco.com/products/star_ccm_plus/index.html.
- [Eça06] Eça, L., Hoekstra, M., “Discretization Uncertainty Estimation Based on a Least Squares Version of the Grid Convergence Index,” in *Proceedings of the Second Workshop on CFD Uncertainty Analysis*, Lisbon, Oct. 2006.
- [Eça08] Eça, L., Hoekstra, M., eds., *Proceedings of the 1st, 2nd, and 3rd Workshops on CFD Uncertainty Analysis*, http://maretec.ist.utl.pt/html_files/CFD_workshops/
- [Elm11] A. Elmahdi, R. Lu, M. Conner, Z. Karoutas, E. Baglietto, Flow induced vibration forces on a fuel rod by les cfd analysis, *The 14th International Topical Meeting on Nuclear Reactor ThermalHydraulics, NURETH-14*, Paper NURETH-14-365 (2011), Toronto, Ontario, Canada.
- [Fis01] Fischer, P.F. and Iliescu, T., “A 3D Channel Flow Simulation at Re-tau=180

Using a Rational LES Model,” 3rd AFOSR International Conference on Direct and Large Eddy Simulation TAICDL, 2001.

- [Fuego] Fuego. SIERRA/Fuego 2.7 User's Manual. Sandia National Laboratories, SAND 2006-6084P, 2008.
- [Joh06] Johnson, R. W., Schultz, R. R., Roache, P. J., Celik, I. B., Pointer, W. D., Hassan, Y. A., *Processes and Procedures for Application of CFD to Nuclear Reactor Safety Analysis*, Idaho National Laboratory report INL/EXT-06-11789 (2006).
- [Kar11] Z. Karoutas, J. Yan, C. M., M. A., Advanced thermal hydraulic method using 3x3 pin modeling, The 14th International Topical Meeting on Nuclear Reactor Thermal-Hydraulics, NURETH-14, Paper NURETH-14-338 (2011), Toronto, Ontario, Canada.
- [KK04] D. A. Knoll and D. E. Keyes, “Jacobian-free Newton-Krylov methods: a survey of approaches and applications”, JCP., 193, pp. 357 – 397, 2004.
- [La05] Lahey, R. T. Jr , “The simulation of multidimensional multiphase flows,” Nuclear Engineering and Design, vol. 235, pp. 1043—1060, 2005.
- [Lak02] Lakehal, D., Smith, B. L. and Milelli, M., “Large-eddy simulation of bubbly turbulent shear flows,” Journal of Turbulence, vol. 3, pp. 1—21, 2002.
- [Lu11] R. Y. Lu. Westinghouse Electric Company Memo. Subject: 17x17 V5H VIPER Test and Instrumented Fuel Rod General Mechanical Information Transmittal to CASL Project, Non-proprietary, April 13, 2011.
- [Nic99] F. Nicoud, F. Ducros, Subgrid-Scale Stress Modelling on the Square of the Velocity Gradient Tensor, Flow, Turbulence and Combustion 62 (1999) 183–200.
- [Obe10] Oberkampf, W. L., Roy, C. J., *Verification and Validation in Scientific Computing*, Cambridge University Press, New York (2010).
- [Rid10] Rider, W. J., Kamm, J. R., Weirs, V. G., *Code Verification Workflow in CASL*, Sandia National Laboratories report SAND2011-7060 (2010).
- [Rid11a] Rider, W. J., Kamm, J. R., Weirs, V. G., *Procedures for Calculation Verification*, Sandia National Laboratories report SAND2011-0233P (2011).
- [Rid11b] Rider, W. J., Kamm, J. R., Weirs, V. G., *Verification, Validation and Uncertainty Quantification in CASL*, Sandia National Laboratories report SAND2011-0234P (2011).
- [Roa98] Roache, P., *Verification and Validation in Computational Science and Engineering*, Hermosa Publishers, Albuquerque (1998).

- [Roa09] Roache, P., *Fundamentals of Verification and Validation*, Hermosa Publishers, Albuquerque (2009).
- [Rod10a] S. B. Rodriguez and M. S. El-Genk. Numerical investigation of potential elimination of hot streaking" and stratification in the VHTR lower plenum using helicoid inserts. *Nuclear Engineering and Design Journal*, 240:995{1004, 2010.
- [Rod10b] S. B. Rodriguez and M. S. El-Genk. Cooling of an isothermal plate using triangular array of swirling air jets. 14th International Heat Transfer Conference, Washington DC, 2010.
- [Rod10c] S. B. Rodriguez, S. Domino, and M. S. El-Genk. Safety analysis for the NGNP lower plenum using the Fuego CFD code. CFD4NRS-3 Workshop, Experimental Validation and Application of CFD and CMFD Codes to Nuclear Reactor Safety Issues, Washington, DC, September 14-16, 2010.
- [Roy05] Roy, C. J., "Review of Code and Solution Verification Procedures for Computational Simulation," *J. Comput. Phys.* **205**, pp. 131–156 (2005).
- [Roy10a] Roy, C. J., "Review of Discretization Error Estimators in Scientific Computing," 48th AIAA Aerospace Sciences Meeting, January 2010, Orlando, FL, AIAA 2010-126 (2010).
- [Roy10b] Roy, C. J., Oberkampf, W. L., "A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing," submitted to *Comp. Meth. Appl. Mech. Engrng.* (2010).
- [Sch79] Schlichting, H., "Boundary Layer Theory," McGraw-Hill, 1979.
- [Sha06] J. Shadid, A. G. Salinger, R. P. Pawlowski, P. T. Lin, G. L. Hennigan, R. Tuminaro, R. B. Lehoucq, Large-scale stabilized FE computational analysis of nonlinear steady-state transport / reaction systems, *Comp. Meth. App. Mechanics and Eng.* 195 (2006) 1846–1871.
- [Sha10] J. Shadid, R. Pawlowski, J. W. Banks, L. Chacon, P. Lin, R. Tuminaro, Towards a scalable fully-implicit fully-coupled resistive MHD formulation with stabilized FE methods, *Journal of Computational Physics* 229 (20) (2010) 7649–7671.
- [Sha11a] J. N. Shadid, T. M. Smith, R. P. Pawlowski, E. C. Cyr, P. D. Weber, and D. Z. Turner, "Initial Drekar LES CFD simulations for Grid-to-Rod Fretting (GTRF)." Sandia National Laboratories, 2011.
- [Sha11b] J. N. Shadid, T. M. Smith, R. P. Pawlowski, E. C. Cyr, "A Summary of SNL Drekar1 Large-scale Parallel CFD Code for use in the Demonstration and

Evaluation of Methods for VERA-CFD,” Sandia National Laboratories, 2011.

- [Sma63] Smagorinsky, J., “General Circulation Experiments with the Primitive Equations,” *Monthly Weather Review*, vol. 3, pp. 99—164, 1963.
- [Smi11] T. M. Smith, J. N. Shadid, R. P. Pawlowski, E. C. Cyr, Reactor core sub-assembly simulations using a stabilized finite element method, The 14th International Topical Meeting on Nuclear Reactor Thermal-Hydraulics, NURETH-14, Paper NURETH-14-500 (2011), Toronto, Ontario, Canada.
- [Ste01] Stern, F., Wilson, R. V., Coleman, H. W., Paterson, E. G., “Comprehensive Approach to Verification and Validation of CFD Simulations—Part 1: Methodology and Procedures,” *J. Fluids Engrng* **123**, pp. 793–802 (2001).
- [Tur11] D. Turner, S. Rodriguez., Assessment of existing Sierra/Fuego capabilities related to Grid-to-Rod Fretting (GTRF), VRI Report 2011.
- [Wan10] J. Wan. Westinghouse Electric Company Memo. Subject: V5H Fuel Assembly CAD Model and CFD Model Transmittal, December 10, 2010.
- [Xin10] Xing, T., Stern, F., “Factors of Safety for Richardson Extrapolation,” *J. Fluids Engrng* **132**, pp. 061403-1– 061403-13 (2010).