ABSTRACT: The main goal of solution verification is to assess the convergence of numerical predictions as a function of discretization variables such as element size $\Delta x$ or time step $\Delta t$. The challenge is to verify that the approximate solutions of the discretized laws-of-conservation or equations-of-motion converge to the solution of the continuous equations. In the case of code verification where the continuous solution of a test problem is known, common practice is to obtain several discrete solutions from successively refined meshes or grids; calculate norms of the solution error; and verify the rate with which discrete solutions converge to the continuous solution. In the case of solution verification where the continuous solution is unknown, common practice is to obtain several discrete solutions from successively refined meshes or grids; estimate an extrapolation of the continuous solution; verify the rate-of-convergence; and estimate numerical uncertainty bounds. The formalism proposed to verify the convergence of discrete solutions derives from postulating how truncation error behaves in the asymptotic regime of convergence. The paper offers an overview of these techniques with illustrations such as the modal analysis of plates or propagation of waves in ideal gases. It is contributed to the proceedings of IMAC-XXV in an attempt to raise awareness in the Structural Dynamics community about convergence-related issues that ubiquitously present themselves in numerical simulations. (Publication approved for unlimited, public release, LA-UR-06-6002, Unclassified.)

1. INTRODUCTION

In computational engineering and physics, partial differential equations that govern the evolution of state variables, such as energy, pressure, velocity, or displacement, are discretized for implementation and resolution on finite-digit computer arithmetic. The challenge of code and solution verification is to assess the extent to which approximate solutions of the discretized equations converge to the exact solution of the continuous equations. In addition to assessing how “closely” discrete solutions match the continuous solution, it is often of great importance to verify that the observed rate-of-convergence matches the rate that should theoretically be provided by the numerical method or solver. Simply speaking, verification is the first “V” of the Verification and Validation (V&V) of predictions obtained from numerical simulations [1].

Solution verification can be defined as a scientifically rigorous and quantitative process for assessing the mathematical consistency between continuum and discrete variants of partial differential equations used to represent a reality of interest [2]. Verification involves comparing numerical solutions obtained from calculations performed on successively refined meshes or grids to a reference. The main difficulty is that the exact solution of the continuous equations is not always known and available to define this reference. Continuous solutions can be derived analytically in a few special cases only that either feature a simple geometry, smooth dynamics, linearized operators, or combinations of the above. Well-known examples in Structural Dynamics include the single degree-of-freedom mass-
spring-damper oscillator, bending of the two-dimensional Euler-Bernoulli beam, and vibration of membranes or shells. These are examples of problems that have been extensively studied and for which exact or highly-accurate solutions of the continuous equations can be obtained. Few other verification test problems, however, exist that offer closed-form solutions for practical applications.

In general one talks of code verification when the solution of the continuous equations can be derived in analytical, closed form, which provides an exact reference to which the discrete solutions are compared. If the system of equations and its initial and boundary conditions are complicated enough, then a continuous solution cannot be derived analytically and one talks of solution verification (or self-convergence). In this latter case no exact reference is available to calculate the solution error due to discretization, which makes verification very challenging.

Even though the case of code verification is briefly mentioned, the discussion presented in this publication focuses on solution verification. The dominant paradigm for solution verification postulates that the numerical method provides discrete solutions that converge to the (unknown) continuous solution with a specific rate-of-convergence. For example a finite element analysis featuring quadratic elements should provide discrete solutions that converge with a rate $p = 2$. Common practice is to obtain several discrete solutions from successively refined meshes or grids; estimate an extrapolation of the continuous solution; verify the rate-of-convergence; and estimate numerical uncertainty bounds. Techniques such as the Richardson extrapolation and Grid Convergence Index are available to extrapolate the numerical approximations and quantify solution error, respectively. Studying the convergence of solutions is generally restricted to scalar predictions to simplify the equations involved, as shown below, and provide closed-form derivations. In the discipline of Structural Dynamics this would mean verifying the convergence, for example, of peak stresses obtained at a given location on the structure instead of verifying convergence of the entire stress fields predicted by the finite element simulations.

Successes of code and solution verification include the development of a formalism to study the convergence of discrete solutions for a wide range of applications, from linear, non-dissipative, elliptic equations to non-linear, shocked, hyperbolic equations. It means that the convergence, for example, of solutions for the resonant frequencies of a modal analysis can be studied with the same tools as the convergence of solutions for a blast wave propagating in ideal gases. The failures of code and calculation verification are embodied by the many restrictions imposed on the way an analysis is typically carried out. They include restricting studies to scalar quantities; further restricting them to spatial-only convergence; and not always accounting for how discrete solutions are constructed by the numerical methods, and the specific properties that result.³

The publication offers an introduction to code and solution verification. It is restricted to well-established techniques and does not discuss recent developments that attempt to push the boundaries of the current state-of-the-practice. (See, for example, References [3-9].) This work is contributed to the proceedings of IMAC-XXV in an attempt to raise awareness in the Structural Dynamics community about convergence-related issues that ubiquitously present themselves in numerical simulations.

2. THE ASYMPTOTIC REGIME OF CONVERGENCE

Verifying the convergence of discrete solution hinges on the concept of asymptotic regime of convergence. The choice of element or cell size, denoted by $\Delta x$, in a computational model can be thought of as being a one-dimensional discretization domain. Different choices of $\Delta x$ values within the discretization domain induce different behaviors of the overall numerical error. By definition the asymptotic regime is the region of the discretization domain where truncation error dominates the overall production of numerical error.

³ It is common practice, for example, to study the convergence of finite element solutions in the sense of a generic norm, such as $L^1$ or $L^2$, or restrict the analysis to scalar responses defined point-wise. A finite element solution, however, never satisfies the equations-of-motion in a point-wise sense. It is defined in the sense of the internal energy norm consistent with the weak formulation of the equations. Assessing convergence of point-wise, scalar responses is inconsistent with the way finite element solutions are constructed by the approximation method, which may or may not lead to erroneous conclusions.
Figure 1 gives a simplified illustration of the main three regimes of discretization, color-coded according to the following:

- **Domain where the discretization is inappropriate;**
- **Domain of asymptotic convergence;**
- **Domain where round-off errors accumulate.**

Going from right (larger values of $\Delta x$) to left (smaller values of $\Delta x$), the first domain shown is color red is where the choice of element or cell size is not even appropriate to solve the discrete equations. This is, for example, the case when elements are too coarse to resolve important geometrical features of a contact between components; a constraint of numerical stability is violated; or the cell size of a computational fluid dynamics grid is too coarse to capture a characteristic scale over which mixing between two materials occurs. Although it could be argued that discrete solutions originating from this regime of discretization are useful to indicate overall trends, their numerical quality should be questioned with the highest degree of suspicion.

The second region shown in color green is where truncation error dominates the numerical error of discrete solutions: it is the asymptotic regime of convergence. Because truncation error dominates, the numerical error can be reduced (that is, solution accuracy can be improved) by performing a calculation with smaller element or cell sizes. We have just outlined the basic principle of conducting a mesh or grid refinement study.

Furthermore and because truncation dominates within the asymptotic regime of convergence, the behavior of numerical error can be modeled mathematically using an equation such as:

$$
\log ||y^{\text{Exact}} - y(\Delta x)||
$$

The second region shown in color green is where truncation error dominates the numerical error of discrete solutions: it is the asymptotic regime of convergence. Because truncation error dominates, the numerical error can be reduced (that is, solution accuracy can be improved) by performing a calculation with smaller element or cell sizes. We have just outlined the basic principle of conducting a mesh or grid refinement study.

Furthermore and because truncation dominates within the asymptotic regime of convergence, the behavior of numerical error can be modeled mathematically using an equation such as:

$$
\epsilon_y(\Delta x) = ||y^{\text{Exact}} - y(\Delta x)|| = \beta_1(\Delta x^p) + O(\Delta x^{p+1})
$$

where $\epsilon_y$ denotes the difference, estimated in the sense of a user-defined norm $||\cdot||$, between the exact solution $y^{\text{Exact}}$ of the continuous partial differential equations and the discrete solution $y(\Delta x)$ obtained with an element or cell size $\Delta x$. In equation (1), the pre-factor $\beta_1$ represents a regression coefficient. The exponent $p$ characterizes the rate with which the solution error $\epsilon_y$ is reduced when the cell size decreases, $\Delta x \to 0$. Once again, because the asymptotic regime of convergence is the region of
discretization where truncation dominates, the value of exponent $p$ should match the rate-of-convergence of the numerical method implemented. For example a finite element calculation that uses linear shells should exhibit a rate equal to $p = 1$. Likewise a second-order accurate Godunov finite difference scheme should exhibit a rate equal to $p = 2$.

Figure 2 suggests the meaning of the two parameters $(\beta_1; p)$ of equation (1). If the solution error $\varepsilon_Y$ is represented graphically versus cell size $\Delta x$ on a log-log scale, then the exponent $p$ is the slope of a straight line that best-fits the data and $\log(\beta_1)$ is the value of the intercept at $\Delta x = 0$.

The last region shown in color grey in Figure 1 is a limiting case for asymptotic convergence due to the fact that, with finite arithmetic implemented in a computer, round-off effects eventually start to accumulate as $\Delta x \to 0$. Round-off could then accumulate to the point where it supplants truncation error as the dominant mechanism that produces numerical error.

Understanding the asymptotic regime of convergence is important for two reasons. First the derivations given in this paper, and that provide the foundation of solution verification, are valid only within the asymptotic regime. Second verifying that the discretization $\Delta x$ used in the calculation leads to a discrete solution in the asymptotic regime provides a strategy to reduce the solution error. If the solution error is too large for the intended application and needs to be reduced, then performing a calculation with a smaller element size reduces it. This is, however, true only within the asymptotic regime of convergence.

3. STATE-OF-THE-PRACTICE TO VERIFY THE CONVERGENCE OF SOLUTIONS

This section provides a summary of the main equations written to assess the convergence of discrete solutions (or lack thereof). The discussion introduces the current state-of-the-practice in computational engineering and physics without paying tribute to some of the more advanced techniques or recent developments. The reader is referred to References [4] and [8-10], to list only a few, that discuss issues that extend beyond the limited scope of this section.

The starting point to assess the convergence of discrete solutions is to postulate an equation, usually referred to as the solution error Ansatz model that describes how the numerical error behaves in the asymptotic regime of convergence. We have explained in section 2 why equation (1) is an appropriate choice.

The distinction needs to be emphasized between code verification, where the exact solution $y^{\text{Exact}}_{\Delta x}$ of the continuous equations is known analytically (or provided by an approximate yet highly-accurate solution procedure), and solution verification where $y^{\text{Exact}}_{\Delta x}$ is unknown. In the former case the left-hand side of equation (1), that is, $\varepsilon_Y(\Delta x) = ||y^{\text{Exact}}_{\Delta x} - y(\Delta x)||$, can be computed given a discrete solution $y(\Delta x)$ obtained from a computational grid with element or cell size $\Delta x$. The exact and discrete solutions can be any scalar quantity, curve, or multi-dimensional field and the only unknowns are the two parameters $(\beta_1; p)$ of equation (1). Two discrete solutions, one obtained from a "coarse" grid of cell size $\Delta x_C$ and another one obtained from a "fine" grid of cell size $\Delta x_F$ (where $\Delta x_C = R.\Delta x_F$ and $R > 1$), suffice to estimate the pair of unknowns $(\beta_1; p)$. The two equations are:
\[
\begin{align*}
\varepsilon_y(\Delta x_c) &= \|y^{\text{Exact}}(\Delta x_c) - y(\Delta x_c)\| = \beta_1(\Delta x_c^0) \\
\varepsilon_y(\Delta x_f) &= \|y^{\text{Exact}}(\Delta x_f) - y(\Delta x_f)\| = \beta_1(\Delta x_f^0)
\end{align*}
\] (2)

and the solution for parameters \((\beta_1; p)\) is given by:

\[
\log \left( \frac{\varepsilon_y(\Delta x_c)}{\varepsilon_y(\Delta x_f)} \right) = p \log(R), \quad \beta_1 = \frac{\varepsilon_y(\Delta x_c)}{\Delta x_c^0} = \frac{\varepsilon_y(\Delta x_f)}{\Delta x_f^0} \tag{3}
\]

where \(R\) is the refinement ratio, that is, \(R = \Delta x_c / \Delta x_f\). Running code verification test problems is useful for two purposes. First it assesses the performance of the code on specific test problems for which an exact solution \(y^{\text{Exact}}\) is known, which allows code developers and analysts to make sure that no programming error is present. Second it can serve as a benchmarking exercise to assess the rate-of-convergence of the numerical method or solver implemented in the code. The main drawback, however, is that the exact solution of the continuous equations must be known. It seriously restricts the application of code verification to a few test problems that may not cover the full range of mechanics or physics implemented in the code.

We now proceed to discuss the general case of solution verification, that is, the assessment of convergence of discrete solutions as the mesh or cell size is refined, \(\Delta x \to 0\). The difficulty is that the exact solution \(y^{\text{Exact}}\) of the continuous equations is unknown. It means that the solution procedure described by equation (3) cannot be implemented since \(\varepsilon_y(\Delta x)\) cannot be computed.

To circumvent this difficulty, two assumptions are made. The first assumption is to replace the exact solution \(y^{\text{Exact}}\) by a “reference” solution denoted by \(y^{\text{Reference}}\) and selected by the analyst. A typical practice is to run a calculation with a very fine grid, obtain a highly-accurate solution of the discretized partial differential equations, and assume that it is close enough to the exact-but-unknown solution \(y^{\text{Exact}}\). The solution error is then defined as:

\[
\varepsilon_y(\Delta x) = \|y^{\text{Reference}}(\Delta x) - y(\Delta x)\| \tag{4}
\]

and the same solution procedure is implemented, that is, equation (3) using definition (4).

Situations may occur where this practice is not satisfactory because one does not know \textit{a priori} what may or may not constitute a good-quality reference, one that would be “close enough” to the exact-but-unknown solution \(y^{\text{Exact}}\). In these situations it is customary to augment the first assumption with a second one whereby equation (4) is specialized to scalar quantities. The standard error Ansatz equation takes the functional form of one of the following equations:

\[
\begin{align*}
\varepsilon_y(\Delta x) &= \|y^{\text{Reference}} - y(\Delta x)\| = \beta_1(\Delta x^0) \tag{5-a} \\
\varepsilon_y(\Delta x) &= |y^{\text{Reference}} - y(\Delta x)| = \beta_1(\Delta x^0) \tag{5-b}
\end{align*}
\]

where the difference between the two versions is that convergence of discrete solutions to the reference solution \(y^{\text{Reference}}\) must be monotonic in the case of equation (5-a). Equation (5-b) is somewhat more general because monotonic convergence is not a pre-requisite. Changing the definition of the solution error \(\varepsilon_y(\Delta x)\) from a norm between multi-dimensional fields, such as shown in equations (1) or (4), to a difference between scalars, such as shown in equations (5), is essential to obtain an analytical solution for the triplet of unknowns \((y^{\text{Reference}}; \beta_1; p)\).

The case of non-monotonic convergence is dealt with in References [8] and [11]. An illustration is given in Figure 3. Four levels of refinement are applied to the finite element mesh of a plate and the 5th resonant frequency \(\omega_5\) is extracted from the mass and stiffness matrices. Figure 3 shows \(\omega_5\) as a function of element size \(\Delta x\). Even though this calculation is linear, well-behaved, and \(\omega_5\) represents an integral quantity (ratio of internal energy to kinetic energy for the 5th mode shape), it can be observed that
convergence of $\omega_5$ is not monotonic. This is a situation where a solution error Ansatz model based on equation (5-a) would not be appropriate.\footnote{Note that, to analyze the data of Figure 3, proposing a solution error Ansatz model based on equation (5-b) may not be appropriate either. It depends on whether or not a reference solution $y_{\text{Reference}}$ can be found such that the error $|y_{\text{Reference}}(\Delta x) - y(\Delta x)|$ converges monotonically as a function of discretization size $\Delta x$. (See References [8], [11] for details.)}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure3}
\caption{Convergence of a resonant frequency versus element size.}
\end{figure}

For simplicity, and because it corresponds to practices most often encountered in computational engineering and physics, the discussion is now limited to the case of monotonic convergence of scalar-valued solutions, that is, equation (5-a). With the essential assumptions of restricting the analysis to scalar solutions, replacing the exact solution $y_{\text{Exact}}$ by a reference $y_{\text{Reference}}$, and considering the case of monotonic-only convergence, the solution error Ansatz equation is re-written without loss of generality as:

$$y(\Delta x) = y_{\text{Reference}} + \beta_1 (\Delta x^p) + O(\Delta x^{p+1})$$

Because equation (6) features three unknowns ($y_{\text{Reference}}$, $\beta_1$, $p$), a minimum of three equations are needed. These are provided by discrete solutions obtained from a coarse grid (with element or cell size $\Delta x_C$), medium grid (with $\Delta x_M$), and fine grid (with $\Delta x_F$). Neglecting the influence of higher-order terms leads to the following system of equations:

\begin{align}
\begin{cases}
y(\Delta x_C) & \approx y_{\text{Reference}} + \beta_1 (\Delta x_C^p) \\
y(\Delta x_M) & \approx y_{\text{Reference}} + \beta_1 (\Delta x_M^p) \\
y(\Delta x_F) & \approx y_{\text{Reference}} + \beta_1 (\Delta x_F^p)
\end{cases}
\end{align}

The system of equations (7) is solved for the triplet ($y_{\text{Reference}}$, $\beta_1$, $p$) by combining the equations to eliminate two unknowns, $y_{\text{Reference}}$ and $\beta_1$. The value of the rate-of-convergence is obtained by solving the following non-linear equation:

$$p(\log(R_{MF}) + \log(1 - R_{CM}^p) - \log(1 - R_{MF}^p)) = \log\left(\frac{y(\Delta x_M) - y(\Delta x_C)}{y(\Delta x_F) - y(\Delta x_M)}\right)$$

where $R_{CM}$ is the refinement ratio from coarse-to-medium grids and $R_{MF}$ is the refinement ratio from medium-to-fine grids, that is, $R_{CM} = \Delta x_C/\Delta x_M$ and $R_{MF} = \Delta x_M/\Delta x_F$ ($R_{CM} > 1$ and $R_{MF} > 1$). There is no closed-form solution to the non-linear equation (8) when the refinement ratios $R_{CM}$ and $R_{MF}$ are different, and the value of $p$ that satisfies the equation must be obtained through numerical optimization. With a constant refinement ratio, equation (8) yields the well-known solution:
where \( R \) denotes the constant refinement ratio, that is, \( R = R_{CM} = R_{MF} \). The next step is to extrapolate the three discrete solutions \( y(\Delta x_C) \), \( y(\Delta x_M) \), and \( y(\Delta x_F) \) to a reference \( y_{Reference} \) that serves the purpose of substitute to the exact-but-unknown solution \( y_{Exact} \) of the continuous partial differential equations. This is known as the Richardson extrapolation. It can be verified that any combination of two grids provides the same value for \( y_{Reference} \):

\[
y_{Reference} = y(\Delta x_F) + \frac{y(\Delta x_F) - y(\Delta x_M)}{R^p - 1}
\]

\[
= y(\Delta x_M) + \frac{y(\Delta x_M) - y(\Delta x_C)}{R^p - 1}
\]

\[
= y(\Delta x_C) + R^p \frac{y(\Delta x_M) - y(\Delta x_C)}{R^p - 1} = y(\Delta x_C) + R^{2p} \frac{y(\Delta x_F) - y(\Delta x_M)}{R^{2p} - 1}
\]

where the rate-of-convergence is the value calculated from equation (9). Finally the regression coefficient \( \beta_1 \) can be back-calculated from any one of equations (7):

\[
\beta_1 = \frac{y(\Delta x_C) - y(\Delta x_M)}{\Delta x_C^p (R^p - 1)} = \frac{y(\Delta x_M) - y(\Delta x_F)}{\Delta x_M^p (R^p - 1)} = \frac{y(\Delta x_C) - y(\Delta x_F)}{\Delta x_F^p (R^{2p} - 1)}
\]

An illustration is given by verifying the convergence of discrete solutions for the one-dimensional Burgers equation initialized with an arc-tangent (\( \tan^{-1} \)) initial condition at time \( t = 0 \):

\[
\frac{\partial}{\partial t} (y(x; t)) + \frac{\partial}{\partial x} \left( \frac{1}{2} y^2(x; t) \right) = 0, \quad \frac{L}{2} \leq x \leq \frac{L}{2}, \quad 0 \leq t \leq 1, \quad y(x; t = 0) = y_0(x)
\]

where \( y_0(x) = \left( \frac{L}{\pi} \right) \tan^{-1}(\omega x) \). Coefficients \( L = 3 \) cm and \( \omega = 5 \) cm\(^{-1} \) are used for the numerical application. Equation (12) is solved in conservation form using the Lax-Wendroff integration scheme [12]. The non-linear and hyperbolic nature of the Burgers equation evolves the initially smooth jump \( y_0(x) \) into a sharp discontinuity, therefore, reducing convergence to first-order [13].

Figure 4 illustrates the convergence of the discontinuity jump, denoted as \( [y]_S \) and defined as the value of the solution to the left of the discontinuity minus the value to the right, as a function of cell size \( \Delta x \). Four calculations are performed with grid sizes listed in Table 1 and a constant refinement ratio equal to \( R = 4 \). Also shown are values of the rate-of-convergence obtained from equation (9) and Richardson extrapolation obtained from equation (10). It is clear from Figure 4 that convergence of the discrete solutions is monotonic. The observed rates-of-convergence match, for all practical purpose, the theoretical value of \( p = 1 \) as one would expect to get for this shocked (discontinuous) prediction \( [y]_S \).

<table>
<thead>
<tr>
<th>Grid Resolution</th>
<th>Cell Size, ( \Delta x )</th>
<th>Density Jump, ( [y]_S )</th>
<th>Rates-of-convergence, ( p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-coarse (XC)</td>
<td>5.00 cm</td>
<td>0.9995 gm.cm(^{-3} )</td>
<td>N/A</td>
</tr>
<tr>
<td>Coarse (C)</td>
<td>1.25 cm</td>
<td>1.3083 gm.cm(^{-3} )</td>
<td>1.021</td>
</tr>
<tr>
<td>Medium (M)</td>
<td>0.3125 cm</td>
<td>1.3832 gm.cm(^{-3} )</td>
<td>1.008</td>
</tr>
<tr>
<td>Fine (F)</td>
<td>0.078125 cm</td>
<td>1.4018 gm.cm(^{-3} )</td>
<td>N/A</td>
</tr>
<tr>
<td>Richardson extrapolation of the solution based on grids (C; M; F)</td>
<td>1.4080 gm.cm(^{-3} )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In summary, equations (9-11) calculate the triplet of unknowns \((y^{\text{Reference}}; \beta_1; p)\) in the case of a constant mesh or grid refinement ratio. Clearly, this analysis technique is appropriate only to the extent that the assumptions upon which it relies are satisfied. There are four essential assumptions summarized to conclude this section. 1) The sizes used (such as \(\Delta x_C\), \(\Delta x_M\), and \(\Delta x_F\)) provide discrete solutions located in the asymptotic regime of convergence. 2) The analysis is restricted to scalar quantities. 3) Because it is unknown, the exact solution \(y^{\text{Exact}}\) of the continuous equations is replaced by a reference \(y^{\text{Reference}}\) to be determined during the analysis. 4) Convergence of the discrete solutions \(y(\Delta x)\) to the reference \(y^{\text{Reference}}\) is monotonic.

4. THE GRID CONVERGENCE INDEX

The Grid Convergence Index (GCI) was proposed by Patrick Roache in 1994 to report the results of grid convergence studies in computational fluid dynamics [14]. Its main advantage is to provide a simple, standardized metric independent of the order of the numerical method and refinement strategy. In other words, the GCI obtained from analyzing a first-order numerical scheme with grid “halving” (that is, \(p = 1\) and \(R = 2\)) can be compared to the GCI obtained from analyzing a second-order numerical method with non-constant mesh refinement.

In simple terms, the GCI is a scaled version of the relative difference between two discrete solutions \(y(\Delta x_M)\) and \(y(\Delta x_F)\) obtained from running the same calculation with two element or grid sizes \(\Delta x_M\) and \(\Delta x_F\) (where, as noted before, \(\Delta x_M = R \cdot \Delta x_F\)). Scaling by a factor \(F_S/(R^2 - 1)\) is applied to account for the specific values of the rate-of-convergence and grid refinement ratio:

\[
\text{GCI} = \frac{F_S}{R^2 - 1} \left| \frac{y(\Delta x_F) - y(\Delta x_M)}{y(\Delta x_F)} \right|
\]  

(13)

where \(F_S\) is the so-called “safety factor” whose value depends on the application, characteristics of the code used, and smoothness of the solution. Values \(F_S = 1.25\) and \(F_S = 3\) have been proposed for various
applications in computational fluid dynamics and solid mechanics. Small values of the GCI, such as 1% or less, indicate that the discrete solutions are converging to the continuous solution $y^{\text{Exact}}$ and the Richardson extrapolation $y^{\text{Reference}}$ accurately estimates $y^{\text{Exact}}$.

Figure 4 illustrates the bounds of solution error obtained for the Burgers equation (12). The GCI values calculated with $F_S = 1.25$ are listed in Table 2. The vertical, solid lines shown in Figure 4 in color red are solution error bounds defined somewhat arbitrarily as +/- one GCI value. The small GCI values listed in Table 2 provide evidence of asymptotic convergence.

<table>
<thead>
<tr>
<th>Grid Resolution</th>
<th>Cell Size, $\Delta x$</th>
<th>Density Jump, $[\gamma]_S$</th>
<th>GCI Values, GCI</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-coarse (XC)</td>
<td>5.00 cm</td>
<td>0.9995 gm.cm$^{-3}$</td>
<td>9.83%</td>
</tr>
<tr>
<td>Coarse (C)</td>
<td>1.25 cm</td>
<td>1.3083 gm.cm$^{-3}$</td>
<td>2.26%</td>
</tr>
<tr>
<td>Medium (M)</td>
<td>0.3125 cm</td>
<td>1.3832 gm.cm$^{-3}$</td>
<td>N/A</td>
</tr>
<tr>
<td>Fine (F)</td>
<td>0.078125 cm</td>
<td>1.4018 gm.cm$^{-3}$</td>
<td>0.55%</td>
</tr>
</tbody>
</table>

Another attractive property of the GCI, besides providing a standardized metric of solution error, is that the ratio between two successive GCI values should be equal to $R^p$ when the discrete solutions are located in the asymptotic regime of convergence, that is:

$$\frac{\text{GCI}_{C,M}}{\text{GCI}_{M,F}} \approx R^p$$

where $\text{GCI}_{C,M}$ denotes the value calculated from solutions $y(\Delta x_C)$ and $y(\Delta x_M)$ while $\text{GCI}_{M,F}$ is the value calculated from solutions $y(\Delta x_M)$ and $y(\Delta x_F)$. This diagnostic of asymptotic convergence does not depend on the safety factor $F_S$. In fact one only needs triplets of simulation results $(y(\Delta x_C); y(\Delta x_M); y(\Delta x_F))$, such as those given in Tables 1 and 2, to calculate the diagnostic:

$$\frac{\text{GCI}_{C,M}}{\text{GCI}_{M,F}} = \frac{|y(\Delta x_M) - y(\Delta x_C)|}{|y(\Delta x_F) - y(\Delta x_M)|}$$

The numerical application using the three finest grids for the Burgers equation (12) gives a ratio of GCI values equal to 4.08, which is in excellent agreement with the theoretical value of $R^p$ for $p = 1.008$ (see Table 1) and $R = 4$, further confirming the hypothesis of asymptotic convergence.

The selection of an appropriate safety factor $F_S$ is a topic that generates heated debate. Some practitioners advocate keeping it constant while others prefer to turn it into a variable to account for better or worse-than-expected convergence, such as proposed by Fred Stern [15]:

$$F_S = \frac{R^{p^*} - 1}{R^p - 1}$$

where $p^*$ and $p$ are the theoretical and observed rates-of-convergence, respectively. It is often claimed that defining solution error bounds as $|y^{\text{Exact}} - y(\Delta x_F)| \leq \text{GCI}$ where the safety factor is equal to $F_S = 3$ defines a “confidence interval of grid convergence uncertainty at the 95% confidence level.” We emphasize that there is no rigorous justification for such practice and, to the best of our understanding, the claim that the GCI defines a probability or confidence interval at any confidence level is simply wrong. Nevertheless GCI values can be used to define conservative bounds of the solution error, such as those shown graphically in Figure 4.

---

5 A justification of scaling in equation (13) is briefly presented. Note that for second-order accuracy and grid halving (that is, $p = 2$ and $R = 2$) the denominator is equal to $(R^2 - 1) = 3$. A safety factor $F_S = 3$ simplifies definition (13) to the relative difference between the discrete solutions $y(\Delta x_M)$ and $y(\Delta x_F)$. The utility of pre-factor $F_S/(R^2 - 1)$ is therefore to scale the relative solution error when $(p; R) \neq (2; 2)$ so that the resulting GCI value can be directly compared to one that would be obtained with $(p; R) = (2; 2)$.
5. SOLUTION VERIFICATION FOR FINITE ELEMENT ANALYSIS: AN EXAMPLE

This section gives an example of solution verification for a linear Hertz contact problem. The problem is solved with an axi-symmetric finite element model analyzed using the general-purpose software package HKS/ABAQUS version 6.4.3. Figure 5 illustrates the contact between a rigid hemisphere (shown as a solid, blue line) and deformable medium.

![Figure 5. Illustration of the Hertz contact problem.](image)

Figure 5. Illustration of the Hertz contact problem.

![Figure 6. Stress fields $\sigma_{YY}$ for the Hertz contact problem.](image)

(a) Extra-coarse mesh with 20 elements. (b) Fine mesh with 160 elements.

Dimensions of the deformable, elastic medium are one meter-by-one meter. Material properties are $E = 10^3 \text{ N.m}^{-2}$ (modulus of elasticity) and $\nu = 0.3$ (Poisson’s ratio). The hemisphere is moved down under
displacement control by 1 cm beyond the initial contact with the deformable medium shown as point C in Figure 5. The contact condition is the HKS/ABAQUS default hard contact model with small sliding allowed and no friction. For the purpose of this illustration it is assumed that one is interested predicting stress values $\sigma_{yy}$ in the vertical direction on the top surface of the deformable medium, that is, at the location of finite elements that could potentially come into contact with the rigid hemisphere. Stress values $\sigma_{yy}$ are given in units of N.m$^{-2}$ (Pa) at the final simulation time of $T = 1$ sec.

Four meshes are analyzed with 20, 40, 80, and 160 finite elements to discretize the deformable medium in each spatial direction. Figure 6 illustrates the stress fields $\sigma_{yy}$ predicted by two of the four meshes analyzed, the extra-coarse and fine meshes. The four solutions are referred to as the extra-coarse, coarse, medium, and fine solutions. Their respective element sizes are $\Delta x = 5, 2.5, 1.25,$ and $0.625$ cm, hence, providing a refinement ratio equal to $R = 2$. The elements are quadratic quadrilaterals and the aspect ratio is kept uniform, constant, and equal to one for all meshes. Simply put, $\Delta x = \Delta y$ as suggested by Figures 5 and 6.

Figure 7 pictures the stress values ($\sigma_{yy}$) vs. location ($x$) of top-row finite elements. Each curve is obtained from one of the four meshes. Such "view-graph norm" comparison of curves indicates visually that the prediction seems to converge as the discretization is refined. An analysis of solution convergence using quantitative metrics, such as those defined in the previous sections, offers the important advantage over the view-graph norm of assessing convergence rigorously.

![Traction Stress “S22” of Hertz Contact Simulations](image)

Figure 7. Stress values ($\sigma_{yy}$) vs. location ($x$) of top-row finite elements.

Note that an analytical solution can be found for the Hertz contact problem applied to a simple geometry such as the one used here. That is, an exact solution of the continuous equations can be derived and used for assessing the convergence of discrete solutions. It is not what is done here. It is, instead, assumed that the exact solution is not available to us. Consequently all that can be studied is self-convergence or the ability of numerical approximations to converge to an unknown solution as the element or grid size is refined, $\Delta x \to 0$.

Tables 3-5 assess the self-convergence of peak stress prediction (in absolute value) at point C. The peak stress values extracted from the four curves of Figure 7 are listed in Table 3. These are the only data needed to compute equations (9-11). Table 4 lists the rates-of-convergence calculated from equation (9) and Table 5 lists values of the GCI calculated from equation (13) with a safety factor equal to $F_S = 1.25$. 
Table 3. Stress values $|\sigma_{yy}|$ predicted at point C.

<table>
<thead>
<tr>
<th>Quantity</th>
<th>Extra-coarse</th>
<th>Coarse</th>
<th>Medium</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size, $\Delta x$</td>
<td>5.000 cm</td>
<td>2.500 cm</td>
<td>1.250 cm</td>
<td>0.625 cm</td>
</tr>
<tr>
<td>Stress $</td>
<td>\sigma_{yy}</td>
<td>$ at Point C</td>
<td>74.82 N/m²</td>
<td>74.00 N/m²</td>
</tr>
</tbody>
</table>

(Legend of Table 3: Refinement ratio, $R = 2$.)

Table 4. Rates-of-convergence observed using triplets of discrete solutions.

<table>
<thead>
<tr>
<th>Triplets Used</th>
<th>Extra-coarse</th>
<th>Coarse</th>
<th>Medium</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Triplet (XC; C; M)</td>
<td>$p = 0.15$</td>
<td>(Not used.)</td>
<td>(Not used.)</td>
<td>$p = 2.09$</td>
</tr>
<tr>
<td>Triplet (C; M; F)</td>
<td>(Not used.)</td>
<td>(Not used.)</td>
<td>(Not used.)</td>
<td>$GCI = 0.458%$</td>
</tr>
<tr>
<td>Triplet (C; C; M)</td>
<td>(Not used.)</td>
<td>$GCI = 0.416%$</td>
<td>(Not used.)</td>
<td>$GCI = 0.099%$</td>
</tr>
</tbody>
</table>

(Legend of Table 4: XC = Extra-coarse; C = Coarse; M = Medium; F = Fine. Theoretical value, $p^* = 2$.)

Table 5. Grid convergence indices observed using pairs of discrete solutions.

<table>
<thead>
<tr>
<th>Pairs Used</th>
<th>Extra-coarse</th>
<th>Coarse</th>
<th>Medium</th>
<th>Fine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pair (XC; C)</td>
<td>$GCI = 0.458%$</td>
<td>(Not used.)</td>
<td>(Not used.)</td>
<td>(Not used.)</td>
</tr>
<tr>
<td>Pair (C; M)</td>
<td>(Not used.)</td>
<td>$GCI = 0.416%$</td>
<td>(Not used.)</td>
<td>(Not used.)</td>
</tr>
<tr>
<td>Pair (M; F)</td>
<td>(Not used.)</td>
<td>(Not used.)</td>
<td>(Not used.)</td>
<td>$GCI = 0.099%$</td>
</tr>
</tbody>
</table>

(Legend of Table 5: XC = Extra-coarse; C = Coarse; M = Medium; F = Fine. Obtained with $F_s = 1.25$.)

It is observed from Table 4 that the rate-of-convergence obtained with the triplet of solutions (XC; C; M) does not match the theoretical value of $p^* = 2$. This result makes sense because, with the extra-coarse discretization ($\Delta x = 5$ cm), only two finite elements are in contact with the rigid hemisphere at the final simulation time of $T = 1$ sec., as shown in Figure 6-a. This is not enough to provide an accurate solution. Table 4 also indicates that the rate-of-convergence becomes quadratic when the triplet of solutions (C; M; F) is used instead of the previous one (XC; C; M). This result suggests that we are in the presence of self-convergence. Alternatively it can be stated that for this problem the asymptotic regime of convergence, where truncation error dominates the overall numerical quality of discrete solutions, is reached for finite elements whose sizes are no larger than $\Delta x = 2.5$ cm, or areas $\Delta x \Delta y \leq 6.25$ cm².

Values of the GCI listed in Table 5 are less than 1%, which is an indication of small numerical error. Of particular interest is the ratio between two successive GCI values such as obtained, for example, with the pairs of solutions (XC; C) and (C; M). From Table 5 the ratio of GCI values with the pairs of solutions (XC; C) and (C; M) is equal to $0.458\% / 0.416\% = 1.1$, which does not indicate asymptotic convergence. The next ratio is $0.416\% / 0.099\% = 4.2$, which matches $R^2 = 2^{0.05} = 4.3$ and confirms the hypothesis of asymptotic convergence.

CONCLUSION

Verifying that an analysis code is free of errors and the numerical methods it implements reach rates-of-convergence that match the theory are essential to demonstrate the numerical quality of discrete solutions. Simply speaking, it is the first “V” of the Verification and Validation (V&V) of predictions obtained from numerical simulations. The discipline of code and solution verification in computational engineering and physics is developing tools to assess the source and severity of numerical solution error. The publication outlines a simple procedure to evaluate the asymptotic regime of convergence; estimate a rate-of-convergence based on mesh or grid refinement; extrapolate the discrete solutions; and develop bounds of solution error due to the discretization. Much, however, remains to be done. A non-exhaustive list of topics that warrant further research includes: extending the state-of-the-practice to non-scalar quantities (curves, multiple-dimensional fields); studying the coupling between space and time discretizations; defining a reference mesh for the estimation of solution error; and developing technology to verify Adaptive Mesh Refinement (AMR) calculations in computational engineering and physics.

---

6 A rate-of-convergence equal to 2 is expected because quadratic shape functions are used to define the finite elements of the four meshes used in this illustration, and the solution predicted is "smooth" (continuous).
For many decades the Structural Dynamics community has been at the forefront of developing methods for test-analysis comparison and finite element model updating, which collectively address the second “V” or validation of predictions obtained from numerical simulations. It is our hope that the community, while increasingly relying on modeling and simulation, will understand the importance that numerical issues play in model validation and become an advocate for best practices and more research and development in the discipline of code and solution verification.

ACKNOWLEDGMENTS

This work is performed under the auspices of the Advanced Scientific Computing verification project at the Los Alamos National Laboratory. The authors are grateful to Jerry Brock, project leader, for his support and technical leadership. Los Alamos National Laboratory is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396.

REFERENCES


