The Need for Verifiable Visualization

Robert M. Kirby and Cláudio T. Silva
University of Utah

The use of simulation science as a means of scientific inquiry is increasing at a tremendous rate. The process of mathematically modeling physical phenomena, experimentally estimating important key modeling parameters, numerically approximating the solution of the mathematical model, and computationally solving the resulting algorithm has inundated the scientific and engineering worlds. As increasingly more science and engineering practitioners advocate the use of computer simulation for analyzing and predicting physical and biological phenomena, the computational science and engineering community has started asking introspective questions, such as[1]

- Can computer-based predictions be used as a reliable basis for making crucial decisions?
- How can you assess a computer-based prediction's accuracy or validity?
- What confidence (or error measures) can be assigned to a computer-based prediction of a complex event?

Those researchers outside traditional computational engineering and science areas (traditional areas such as computational fluid dynamics [CFD] and computational solid mechanics [CSM]) are sometimes shocked to hear these questions being asked, as they often have assumed that these types of issues had been settled long ago—at the inception of computing and computational modeling. A study of the computational science and engineering (CS&E) literature from the past 40 years clearly shows that these questions have not been ignored.

Scientists who employ computing for solving problems have always been concerned with accuracy, reliability, and robustness. It was not until the past 10 years, however, that the CS&E community has joined together in an attempt to generate a unified perspective from which to evaluate these questions. The consequence of these efforts has led to what some are calling a new CS&E discipline—validation and verification, or V&V, which seeks to articulate processes by which we can obtain answers to these questions. Let us take a closer look.

Validation and verification

Figure 1 shows the common simulation science pipeline consisting of the physical phenomena of interest, mathematical modeling of the phenomena, simulation, and evaluation (often through a combination of postprocessing and visualization). It also identifies where validation and verification fit into this process (we will further explain these terms later in the article).

Scientists frequently use visualization techniques to help them assess their simulation results. Visualization is the lens through which scientists often view modeling and discretization interactions—hence, visualization itself must be explicitly considered as part of the V&V process. Simulation researchers often create their own visualization tools, claiming that they "don't trust" visualization techniques that they themselves have not implemented. CFD researchers creating visualizations of their own data joke that they are experts in the presentation of their own brand of CFD: colorful faulty dynamics. Such a statement can only be truly understood in the light of the V&V process; it is the means by which simulation scientists gain confidence in their algorithms and implementations as well as those by others within their community. To gain the simulation community's confidence, the visualization process must come under this process's umbrella.

Visualization techniques have lagged behind in error and uncertainty analysis of the methodology as a component of a larger scientific pipeline. Little systematic research efforts have addressed quantifying and minimizing the visualization error budget (a concept we will discuss later in the article). Furthermore, there is a real need to look at this visualization error budget in the context of the error that the rest of the computational pipeline generated and how it impacts visualization algorithms (note that this is distinct from the area of "error and uncertainty visualization," which is concerned with visualizing errors and uncertainties).

Current Procedure
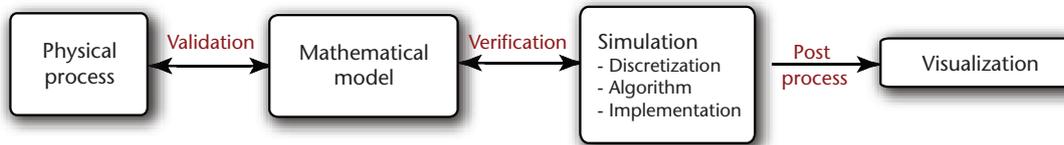
Verifiable Visualization
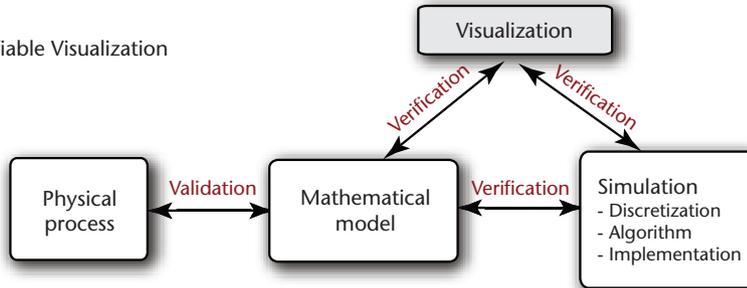
We argue that there is a need for creating a focus area of visualization research that we tentatively call "verifiable visualizations" that will consider both the errors of the individual visualization component within the scientific pipeline and the interaction between and interpretation of the accumulated errors generated in the computational pipeline, including the visualization component. We came to this conclusion on the basis of our realization that although strict error analysis and verification has been the norm in simulation science, visualization has lagged behind other computational techniques in this regard.

V&V within computational science and engineering

The first goal of the current V&V emphasis within the CS&E community is to arrive at a common nomenclature and semantic understanding of commonly used terms. This is imperative to establish V&V as the common base or process by which researchers build scientific trust in simulation codes and results. We will rely on other work[1] as a guide and refer interested readers to it for a more complete philosophical and technical discussion of V&V. Let us look at a brief overview of several key V&V concepts.

As Figure 1 illustrated, the simulation science scientific method consists of the following stages:

- *Scientific problem of interest ("physical process").* This stage states the scientific or engineering problem of interest. Scientists should develop questions so that they can evaluate quantifiable metrics for determining the simulation's level of success.
- *Modeling ("mathematical model").* This stage concerns developing a model that abstracts the problem of interest's salient features so that ex-

ploring and evaluating the model lets researchers find an answer to the questions specified concerning the problem of interest. Modeling techniques include, but aren't limited to, deterministic or probabilistic, discrete, or continuous mathematical models.
- *Computation ("simulation").* This stage concerns generating algorithms and implementations that accurately and efficiently evaluate the model over the range of data needed to answer the questions of interest.
- *Evaluation ("visualization").* This stage concerns the data's distillation and evaluation produced through computational simulation to answer the questions of interest and provide quantifiable determination of the experiment's success.

With this view of the simulation pipeline in place, we can now address the definitions of validation and verification (definitions are taken directly from other research[1]):

- *validation*—the process of determining whether a mathematical model of a physical event represents the actual physical event with sufficient accuracy; and
- *verification*—the process of determining whether a computational model obtained by discretizing a physical event's mathematical model and the code implementing the computational model can be used to represent the model with sufficient accuracy.

From these two definitions, we can see that the idea of the error budget is fundamental to simulation science. An error budget comprises those assumptions and approximations that introduce error (or approximations) into the simulation process and their corresponding impact (or cost)

on the scientific pipeline. Three types of errors are critical components of the scientific process and must be quantified and eliminated:

- *modeling*—errors introduced through the choice of a mathematical model to describe observable data,
- *approximation*—errors introduced in the numerical computation of solutions of the model, and
- *uncertainty*—errors due to variations in model parameters

Recognizing these errors lets scientists judiciously evaluate which component of the process (for example, modeling or numerical approximations) requires refinement in light of comparison with the real phenomenon of interest.

The verification process is commonly partitioned into two areas that most visualization researchers will recognize: *solution verification* and *code verification*. In solution verification, effort is directed toward assuring the input data's accuracy, estimating the numerical approximation error due to discretization, and assuring the resulting simulation output data's accuracy. In code verification, effort is directed toward finding and removing source-code mistakes and (numerical) algorithmic errors. Once these two forms of debugging are accomplished, researchers can not only correct and refine their scientific tools, but also build confidence in the design and handling of the scientific tool and the corresponding results it produces.

When these results are used in the scientific setting, we can examine differences between computational and experimental results in light of the assumptions used when generating the model and simulation. If visualization is the lens through which simulation scientists view their data, is that lens free of flaws? Is it possible that visual discrepancies between simulation and experimental results could be due to assumptions and approximations built into the visualization method? Are the visualization techniques designed based on (and in particular, to respect) properties of the model and the simulation used to generate the visualized data? To place visualization firmly within the scientific process, it must undergo the same level of rigorous analysis.

The state of visualization V&V

To gauge the extent to which verification already exists in the visualization community, we reviewed papers from the past five years (2003 to 2007) of the IEEE Visualization conference (http://vis.computer.org). To establish the extent to which the nomen-clature is already used, we searched the texts of these papers for occurrences of "verify," "validate," and other forms of these words. Less than half of the papers ever used any of these words, let alone in the context of validation or verification. To check whether these concepts exist, we reviewed a subset of the papers in greater detail. While many include sections presenting a logical and mathematically sound derivation of their technique and evaluation of their algorithms' performance, only a handful discussed concepts related to either validation or verification. Those that did tended not to adhere to any standard organization or systematic methodology.

However, just as with computational engineering science practitioners prior to the current V&V movement, the visualization community has not been summarily negligent or ignored verification considerations when designing and implementing visualization techniques. As early as 1995, Al Globus and Sam Uselton[2] specifically entreated the visualization community to adopt verification techniques (in terms of error analysis and software verification). They even proposed a methodology for evaluating visualization software. However, because there is no standard framework for verification and validation in visualization, the community employs a variety of techniques to defend their work. These range from visual comparisons between the results and the observed physical phenomena to measurements based on a well-defined set of benchmarks. Furthermore, given that resulting visualizations are intended to be viewed by humans, some researchers have investigated human perception to measure whether visualizations are adequately interpreted.[3,4] In most of these works, the focus is not so much on verifying the visualization techniques, but on the closely related problem of evaluating visualization methods for effectiveness.

In the related field of computer graphics, a major goal for graphics practitioners is rendering scenes that are either indistinguishable from a photograph of a real scene or evoke the same visual response as that real scene. Perhaps the most straightforward way of determining whether such criterion is met is to do a side-by-side comparison of a photograph and a rendering of the same scene. Given that certain types of visualization also attempt to present data as it would appear in the physical world, a similar type of argument has been applied in our field. For example, physicists investigating 3D flows might wish to visually compare their visualizations with physical observations of flows created under the same initial conditions.

Because many visualization techniques capture variables that are not visible in a physical setting, comparing a visualization with physical observations is not often possible. However, we can still make side-by-side comparisons between different visualization techniques. One type of verification leveraging the graphics processing unit is a comparison between GPU-accelerated techniques and software-based reference implementations. For example, Dan Laney and colleagues present a major speedup in understanding radiograph simulations that they verified by comparing the hardware-accelerated technique with the existing software-based implementation.[5]

In concert with comparison-based verification, some visualization techniques use benchmark data sets. While some of the data sets have become the standard because they were used in the original papers, whereas others were designed to expose interesting cases. For example, the Marschner-Lobb test volume is an example of an artificially generated data set representing a variety of pathological properties for a range of visualization and processing techniques.[6] Researchers have proposed test data sets and benchmarks for a variety of topics including streamlines[7] and topology.[8]

Another method that is often employed in the visualization community is an evaluation by users and domain experts. For example, in Diffusion Tensor—Magnetic Resonance Imaging (DT-MRI), researchers can rely on an expert's domain knowledge to check correct clustering or segmentation of specific regions of the brain.[9] The medical community has relied on manual segmentation as ground truth; medical practitioners verify various automated segmentation methods by comparing their results against those generated manually by experts. Other subfields use user studies as well; for example, some work[10] compares different 2D vector field visualization methods.

While comparing the results can be instructive, verifying the accuracy of visualization techniques requires mathematical error determination. Many papers in our field provide a derivation of the visualization given existing mathematical models, but not all quantify the error involved in evaluating the displayed quantities. Recent work details a case in which a misleading volume visualization led to unnecessary surgery for a patient.[11] Uncertainty visualization is concerned with helping viewers understand such error visually,[12] but it has also reminded visualization researchers of the importance of determining error.

Another example where error is being considered is error-control in streamline integration. The goal of most streamline integration techniques is maximizing accuracy while minimizing error and computational effort. Both Runge-Kutta and extrapolation methods are common in the literature—with the choice of which integration technique to use chosen on the basis of a multitude of mathematical and computational factors, such as error per computational cost and availability (and strength) of error estimators, among others. The lack of smoothness at element or grid boundaries can cause large errors during integration, leading some to use adaptive error control techniques such as Runge-Kutta with error estimation and adaptive step size control. Verifying streamline accuracy is relegated to what has been accomplished for the numerical technique and ordinary differential equation (ODE) solver employed.

Because viewers come to their own conclusions, some have argued that visualization researchers are in the business of applied perception.[13] It is not enough to ensure that each important aspect of a data set can be seen, but rather we must guarantee that a wide audience will understand it. Because of perceptive differences, visualization has been called a subjective field that relies on an expert's domain knowledge to ensure correct interpretation.[9] While these points are important, we argue that validation and verification are concerned with ensuring that the visualization accurately displays the information, not whether viewers interpret them correctly.

The need for verifiable visualization

We might naturally ask whether this is much ado about nothing—that is, is it the case that there exist concrete examples of visualization and data analysis methods that are currently used but do not necessarily respect some set of the constraints or properties that interest the user community? Are there methods (or methodologies) that could benefit from being included in the verification approach? To illustrate the issues, we use the excellent work of Guo-Shi Li, Xavier Tricoche, and Chuck Hansen as an example; most of the discussion in this section is based on their work.[14]

Consider the following computational science example. We are given a vector field $\mathbf{u}(x, y) = (u, v)^T$ over a domain Cartesian lattice on Ω, and we wish to simulate for visualization purposes dye advection through this field. A model of dye diffusion concentration C written in an Eulerian-frame is

$$\frac{\partial C}{\partial t} = (\mathbf{u} \cdot \nabla)C = \nu \nabla^2 C$$
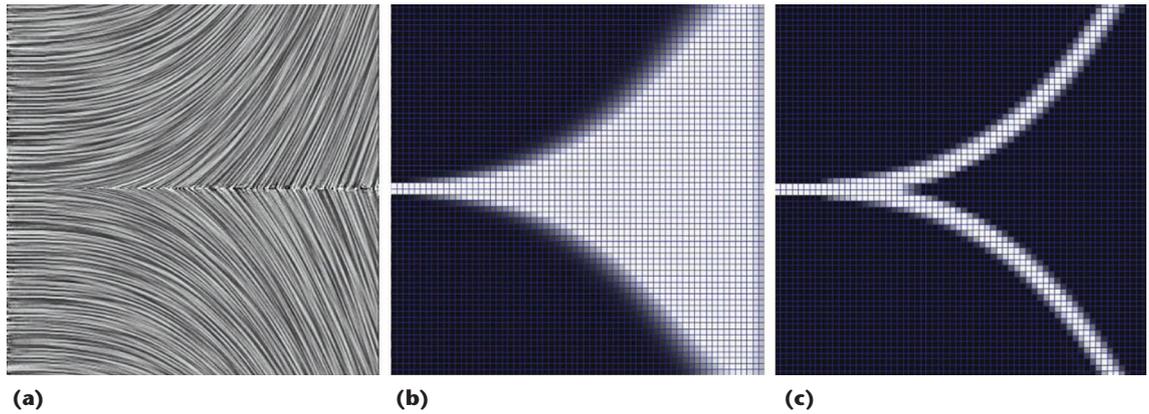
(a) **(b)** **(c)**

Figure 2. Visualization of a synthetic flow of zero divergence exhibiting a splitting behavior. (a) The properties of the flow using a line integral convolution algorithm that shows the behavior of the field. (b) Current dye advection algorithms exhibit interpolation errors that make the field appear divergent and diffusive. (c) Physically-based dye advection would result in the correct splitting behavior. Images used with permission.[14]

where the left-hand side denotes the Eulerian-frame equivalent of the total derivative of the time rate-of-change of concentration and the right-hand side denotes second-order diffusion with parameter v. Explicit time discretization of this equation can be quite expensive due to CFL (Courant-Friedrich-Levy) and diffusion number limit considerations. A semi-Lagrangian technique can solve equations of this form. We can use a splitting scheme in cases in which Lagrangian upwinding discretizes the advection operator. The implicit Euler-Backward or Crank-Nicolson approaches handle is used for time-discretizing the diffusion term. The spatial and temporal discretization of the diffusion term is fairly straightforward using standard finite-difference techniques; the challenge is in the advection term's discretization.

To solve the advection term in a Lagrangian fashion, we must evaluate the vector field $\mathbf{u}(x, y)$ at nonlattice coordinates, and hence some form of approximation of the vector field is necessary. A common approach is to use bilinear interpolation for evaluating the velocity field. Although this approach is unconditionally stable even with relatively long time-stepping, it is subject to numerical diffusion due to interpolation.

The result is a *smearing* artifact that is entirely due to the algorithm and not related to the physical process. In addition, the algorithm does not conserve the system's invariant properties (such as mass). The interpolant can overshoot or undershoot the samples within the local stencil, causing an artificial increase or decrease of mass in the system. Figure 2 shows the errors/ that can accumulate due to a methodology that does not explicitly respect the system's invariant properties. Although visualization researchers have developed several methods to suppress the numerical diffusion that is added due to the handling of the advection terms, they are not based on any physical properties of the underlying problem. With current dye advection schemes, the data under analysis could be incorrectly interpreted due to inaccuracies in the interpolation methods. Without proper verification techniques, these errors could continue to compound with each improvement to existing algorithms.

With the advent of programmable graphics hardware, advection techniques for visualization have increased substantially in interactivity. However, verifying these techniques is a difficult task that is frequently accomplished by comparing images generated with existing algorithms. We need a more formal verification process to guarantee that the recent speed-ups are not sacrificing the analysis' integrity for the ability to see changes in real time.

In other disciplines, researchers have proposed physically-based numerical schemes for dye advection that avoid numerical diffusion and conserve mass, based on an algorithm that uses quasiparabolic functions to represent the distribution of dye material within control volumes in a piecewise fashion. These schemes address the interpolation issue, but from a different perspective than is normally taken in the visualization community. The schemes do not just attempt to eliminate interpolation errors by going to higher-order interpolation (in isolation), but rather step back and investigate which model and discretization properties are required to solve the problem of interest. As a consequence of this investigation, we not only get a higher-order interpolation scheme (and hence less diffusion), but also a particular high-order scheme that preserves a fundamental property of the differential operator you were trying to simulate. Li and colleagues[14] took the first step at incorporating this method into texture advection, and thus achieved visualizations that are closer to the ground-truth.

There is a need for creating a focus area of visualization research that we tentatively call "verifiable visualizations" that will consider both the errors of the individual visualization component within the scientific pipeline (as you would accomplish verification of the simulation component) and also the interaction between and interpretation of the accumulated errors generated in the computational pipeline, including the visualization component.

To begin this process, we must first clearly understand what V&V encompasses and what it attempts to accomplish. Most V&V efforts of this form set out to "... [provide the] community with a common language, a conceptual framework, and general guidance for implementing the processes of computational model V&V" (adapted from the abstract of the *American Society of Mechanical Engineering-ASME 2006 Guide for Verification and Validation in Computational Solid Mechanics* (http://catalog.asme.org/Codes/PrintBook/VV_10_2006_Guide_Verification.cfm)

From this, we see that we need three components for successful V&V efforts: a common language, a conceptual framework, and guidance of implementation.

We advocate that the visualization community put into place a common visualization verification language, a framework consisting of data and visualization pipelines, and examples that individual researchers can mimic or extend while verifying their visualizations, and in doing so initiate a scientific culture of verifiable visualization. Successfully completing this vision will provide the visualization community with the framework for entering the V&V world—that is, in the same way that computational scientists seek verifiable simulations, they will now be able to view their simulation data through the lens of verifiable visualization.

The process of creating such a framework is likely to have other benefits. The data, code, and workflows that will be generated will serve not only for V&V, but should also be useful for issues related to benchmarking and debugging in general. Also, by providing a platform for people to contribute their own workflows, we will encourage greater reproducibility of scientific results. ◾

References

1. I. Babuska and J.T. Oden, "Verification and Validation in Computational Engineering and Science: Basic Concepts," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, nos. 36–38, 2004, pp. 4057–4066.
2. A. Globus and S. Uselton, *Evaluation of Visualization Software*, tech. report, NASA Report NAS-95-005, 1995.
3. R. Kosara et al., "User Studies: Why, How, and When," *IEEE Computer Graphics and Applications*, vol. 23, no. 4, 2003, pp. 20–25.
4. M. Tory and T. Moeller, "Human Factors in Visualization Research," *IEEE Trans. Visualization and Computer Graphics*, vol. 10, no. 1, 2004, pp. 72–84.
5. D. Laney et al., "Hardware-Accelerated Simulated Radiography," *Proc. IEEE Visualization*, IEEE CS Press ,2005, , pp. 343–350.
6. S.R. Marschner and R.J. Lobb, "An Evaluation of Reconstruction Filters for Volume Rendering," *Proc. IEEE Visualization*, IEEE CS Press, 1994, pp. 100–107.
7. X. Ye, D. Kao, and A. Pang, "Strategy for Seeding 3D Streamlines," *Proc. IEEE Visualization*, IEEE CS Press, 2005, pp. 471–478.
8. X. Zheng, B. Parlett, and A. Pang, "Topological Structures of 3D Tensor Fields," *Proc. IEEE Visualization*, IEEE CS Press, 2005, pp. 551–558.
9. J. van Wijk, "The Value of Visualization," *Proc. IEEE Visualization*, IEEE CS Press, 2005, pp. 79–86.
10. D.H. Laidlaw et al., "Comparing 2D Vector Field Visualization Methods: A User Study," *IEEE Trans. Visualization and Computer Graphics*, vol. 11, no. 1, 2005, pp. 59–70.
11. C. Lundström et al., "Uncertainty Visualization in Medical Volume Rendering using Probabilistic Animation," *IEEE Trans. Visualization and Computer Graphics*, vol. 13, no. 6, 2007, pp. 1648–1655.
12. C. Johnson and A. Sanderson, "A Next Step: Visualizing Errors and Uncertainty," *IEEE Computer Graphics and Applications*, vol. 23, no. 5, 2003, pp. 6–10.
13. D. House et al. "Design and Evaluation in Visualization Research," *Proc. IEEE Visualization*, IEEE CS Press, 2005, pp. 705–708.
14. G.-S. Li, X. Tricoche, and C. Hansen, "Physically-Based Dye Advection for Flow Visualization," *Proc. EuroVis 2008*, IEEE CS Press, 2008.

Contact author Robert (Mike) Kirby at Kirby@cs.utah.edu.

Contact editor Theresa-Marie Rhyne at tmrhyne@nscu.edu.