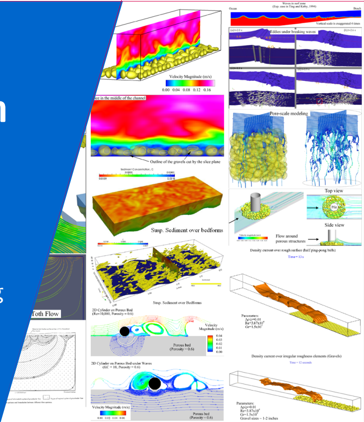
CE597D Special Topics, Fall, 2014

Chapter 10: Verification and Validation



Xiaofeng Liu, Ph.D., P.E.
Assistant Professor
Department of Civil and Environmental Engineering
Pennsylvania State University
xliu@engr.psu.edu

Outline

General introduction

Why we need verification and validation?

- The main concern is to assess the accuracy of computational results.
- Verification: deals with mathematics and numerics
  - code verification (correctness): accurately solve the mathematical model (free of mistakes)
  - solution verification (accuracy): estimate the numerical accuracy
- Validation: the process of determining the degree to which a model represents the real physical process

General introduction

Why we need verification and validation?

- The main concern is to assess the accuracy of computational results.
- Verification: deals with mathematics and numerics
  - code verification (correctness): accurately solve the mathematical model (free of mistakes)
  - solution verification (accuracy): estimate the numerical accuracy
- Validation: the process of determining the degree to which a model represents the real physical process

In more plain language:

- Verification: *Solve the equation right*.
- Validation: *Solve the right equation*.

General introduction

Verification: deals with mathematics and numerics

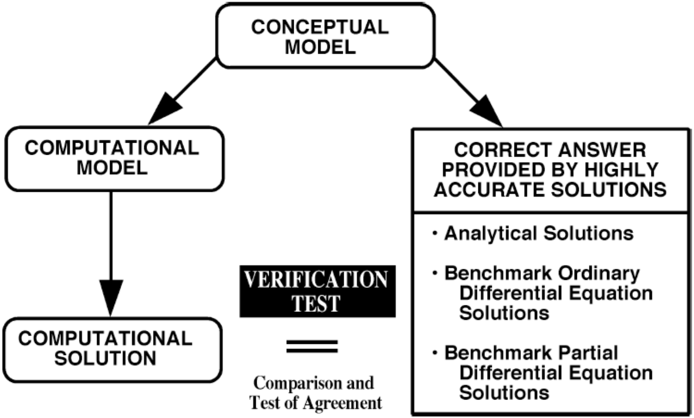


Figure: Scheme for verification (Oberkampf and Roy (2010))

Validation: deals with physics

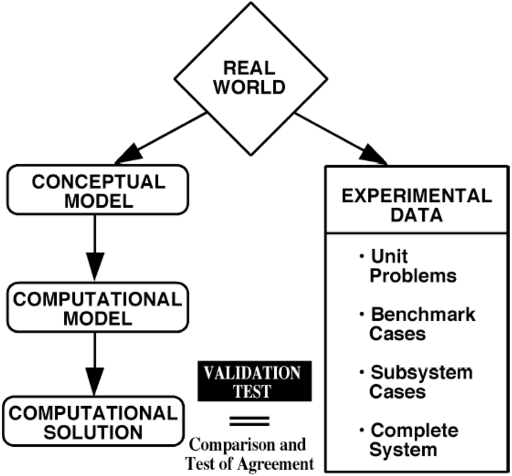


Figure: Scheme for validation (Oberkampf and Roy (2010)

General introduction

Both verification and validation provide:

- process: how to provide proof or evidence
- standard: a reference standard
  - Verification: conceptual model
  - Validation: real physical process

General introduction

Before we go over details of V&V, how to quantify the errors?
Answer: using norm.
Assume a vector $\mathbf{x} = (x_1, x_2, ..., x_n)$, the p-norm or L^p-norm is defined as

$$\|\mathbf{x}\|_p = \left(|x_1|^p + |x_2|^p + ... + |x_n|^p\right)^{\frac{1}{p}} \tag{1}$$

Commonly used measures for error vector $\mathbf{x}$:

- L^1 norm: sum of absolute error values
- L^2 norm: RMS of the error vector
- L^∞ norm: maximum of the absolute error values

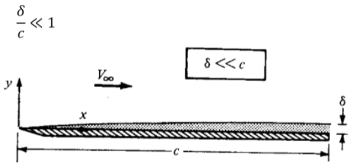
L^1 and L^2 norms are indicators for global (averaged) error, while L^∞ measures the local error (extremes).

Verification methodologies

There are several options to carry out code verification:

- Compare with analytical solution
- Methods of manufactured solutions (MMS)
- code-to-code comparison
- Grid convergence and Grid Convergence Index (GCI)

code verification: Compare with analytical solution



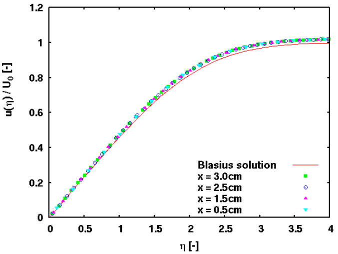


Figure: Comparison between OpenFOAM® result and Blasius similarity solution for flat-plat flow (source: openfoamwiki)

Verification methodologies

code verification: Methods of manufactured solutions (MMS)

- Most of time, given a PDE, B.C./I.C., analytical solution is hard to find.
- MMS starts at the end: it assumes an analytical solution form. When plug into the PDE, it will result a new equation with extra terms

Let the governing equation to be solved have the form

$$L(u) = 0, \tag{2}$$

where L is a general differential operator (advection, diffusion, etc.) Substitute a manufactured solution ϕ into the equation, we get

$$L(\phi) = F, \tag{3}$$

where F should be non-trivial since ϕ is not a solution of Equation 2.

- We can solve the new Equation 3 numerically using the tested code.
- The numerical solution should converge to the manufactured solution ϕ.

code verification: Code-to-code comparison

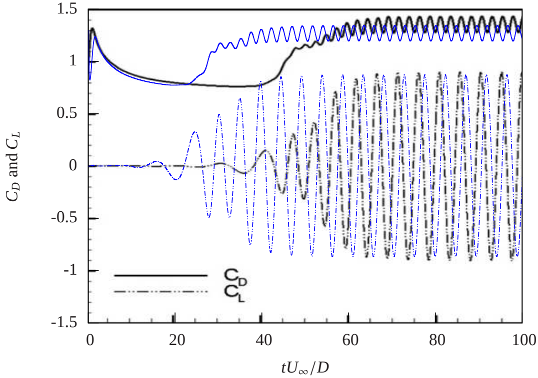


Figure: 2D flow around cylinder: Comparison between my simulation and Mittal et al., (2008)

Verification methodologies

Grid convergence:

- Numerical solution has to be independent of the grid used.
- The result should have nothing related to a particular grid.
- Usually, the more refined the grid, the better (reduce numerical errors)
- Further refinement of the mesh should not change the result: Grid convergency
- But how to measure the grid convergency?
  - qualitatively: for example, plot the velocity profiles from different grids and see if there is significant change
  - quantitatively: use the norm of error.

Verification methodologies

Grid convergence: quantification of error using norm.

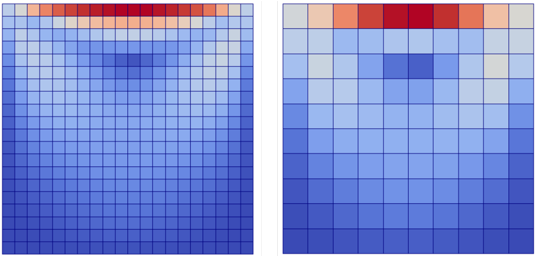


Figure: Velocity of the driven cavity cases with two grids

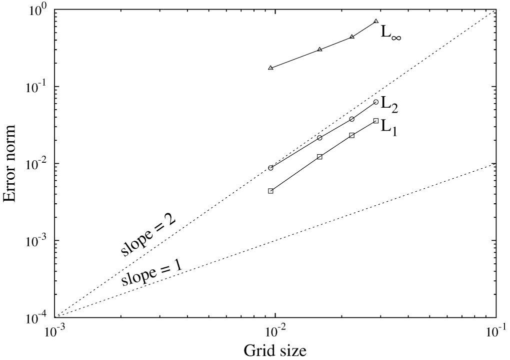
Steps to calculate the error norm:

1. sample the coarse mesh result with the fine mesh cell centers
2. calculate the error (difference) field ϵ between the fine mesh result and the sampled field
3. calculate the error norm of ϵ

Verification methodologies

Grid convergence:

- Sometime the grid convergence study also can prove the order to the numerical scheme
- Just recursively refine the mesh and perform the same calculation to get a sequence of error norm
- Plot the error norm and grid size on a log-log plot.
- The slope of the line shows the order of the accuracy.



Verification methodologies

Richardson extrapolation (RE):

- closely related to the convergence study
- RE is a method to improve the convergence rate of a sequence.
- In the V&V context, it can be used to analyze the convergence rate if we have the sequence.

Let f be the numerical solution on a grid with size h. Do a Tayor series expansion around the exact solution f_{exact}:

$$f = f_{exact} + a_1 h + a_2 h^2 + a_3 h^3 + ... \tag{4}$$

where a_i, $i = 1, 2, ...$, are independent of grid size h.

Verification methodologies

Take a second-order numerical code for example, i.e., $a_1 = 0$. Simulate on two grids with size h_1 and h_2. So we have

$$f_1 = f_{exact} + a_2 h_1^2 + a_3 h_1^3 + ... \tag{5}$$

$$f_2 = f_{exact} + a_2 h_2^2 + a_3 h_2^3 + ... \tag{6}$$

Verification methodologies

Take a second-order numerical code for example, i.e., $a_1 = 0$. Simulate on two grids with size h_1 and h_2. So we have

$$f_1 = f_{exact} + a_2 h_1^2 + a_3 h_1^3 + \dots \tag{5}$$

$$f_2 = f_{exact} + a_2 h_2^2 + a_3 h_2^3 + \dots \tag{6}$$

Eliminating a_2 and solving for f_{exact} gives

$$f_{exact} = \frac{f_1 h_2^2 - f_2 h_1^2}{h_2^2 - h_1^2} + O(h^3) \tag{7}$$

Verification methodologies

Take a second-order numerical code for example, i.e., $a_1 = 0$. Simulate on two grids with size h_1 and h_2. So we have

$$f_1 = f_{exact} + a_2 h_1^2 + a_3 h_1^3 + \ldots \tag{5}$$

$$f_2 = f_{exact} + a_2 h_2^2 + a_3 h_2^3 + \ldots \tag{6}$$

Eliminating a_2 and solving for f_{exact} gives

$$f_{exact} = \frac{f_1 h_2^2 - f_2 h_1^2}{h_2^2 - h_1^2} + O(h^3) \tag{7}$$

Note: if we solve the problem twice on two grids with size h_1 and h_2 using a second-order spatial scheme, the previous RE formula increases the accuracy to third-order!

If we double the grid number from Grid 2 to Grid 1($h_2 = 2h_1$), the formula is simply

$$f_{exact} = \frac{4}{3} f_1 - \frac{1}{3} f_2 + O(h_2^3) \tag{8}$$

Verification methodologies

Grid convergence index (GCI): proposed in Roache (1998)

- A uniform method for grid convergence study and quantify the error in computation

Assume

$$
\begin{aligned}
p &= \text{nominal order of numerical scheme} \quad (9)\\
r &= \text{grid refinement ratio} = h_2/h_1 \quad (10)
\end{aligned}
$$

Equation 7 on the previous slide can be written as

$$f_{exact} = f_1 + \frac{f_1 - f_2}{r^p - 1} + O(h^{p+1}) \quad (11)$$

Verification methodologies

We can define Grid Convergence Index (GCI) using the relative error:

$$GCI = \frac{\epsilon}{r^p - 1}, \text{ where } \epsilon = \left| \frac{f_1 - f_2}{f_1} \right| \tag{12}$$

Equivalently, we can combine Equations 11 and 14 and have

$$GCI = \left| \frac{f_{exact} - f_1}{f_1} \right| + O(h^{p+1}) \tag{13}$$

which clearly indicates the meaning of GGI: the relative error.

Verification methodologies

We can define Grid Convergence Index (GCI) using the relative error:

$$GCI = \frac{\epsilon}{r^p - 1}, \text{ where } \epsilon = \left| \frac{f_1 - f_2}{f_1} \right| \tag{12}$$

Equivalently, we can combine Equations 11 and 14 and have

$$GCI = \left| \frac{f_{exact} - f_1}{f_1} \right| + O(h^{p+1}) \tag{13}$$

which clearly indicates the meaning of GGI: the relative error.
In practice, a conservative Factor of Safety $F_s = 3$ can be used because there are factors affect nominal order p: a second-order scheme might not attain second-order due to various reasons.

$$GCI = F_s \frac{\epsilon}{r^p - 1} \tag{14}$$

PENNSTATE

Verification methodologies

Estimate uncertainty: Draw error bars

- The following content is from:
  Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications J. Fluids Eng. 130, 078001 (2008)
- General steps:
  - Define representative grid size, h. For example cubic (square) root of average cell volume (area):

$$h = \left[\frac{1}{N} \sum_{i=1}^{N} (V_i) \right]^{1/3} \text{ for 3D,} \qquad h = \left[\frac{1}{N} \sum_{i=1}^{N} (A_i) \right]^{1/2} \text{ for 2D} \tag{15}$$

  - select three significantly different sets of grids and run the simulation. Record the values of ϕ critical to the physical process. In general, the transition of grid resolution $r = h_{coarse}/h_{fine}$ should be greater than 1.3 based on experience.

Verification methodologies

Estimate uncertainty: Draw error bars

- General steps:
  - Let $h_1 < h_2 < h_3$, and $r_{21} = h_2/h_1$, $r_{32} = h_3/h_2$, calculate the apparent order p of the code using

$$p = \frac{1}{\ln(r_{21})} \left| \ln \left| \frac{\epsilon_{32}}{\epsilon_{21}} \right| + q(p) \right| \quad (16)$$

$$q(p) = \ln \left(\frac{r_{21}^p - s}{r_{32}^p - s} \right) \quad (17)$$

$$s = sgn \left(\frac{\epsilon_{32}}{\epsilon_{21}} \right) \quad (18)$$

  where $\epsilon_{32} = \phi_3 - \phi_2$, $\epsilon_{21} = \phi_2 - \phi_1$

- Calculate the extrapolated values

$$\phi_{ext}^{21} = \frac{r_{21}^p \phi_1 - \phi_2}{r_{21}^p - 1} \quad (19)$$

  similarly for ϕ_{ext}^{32}.

Estimate uncertainty: Draw error bars

- General steps:
    - Calculate the following error estimates:

$$\text{approximated relative error: } e_a^{21} = \left| \frac{\phi_1 - \phi_2}{\phi_1} \right| \tag{20}$$

$$\text{extrapolatged relative error: } e_{ext}^{21} = \left| \frac{\phi_{ext}^{12} - \phi_1}{\phi_{ext}^{12}} \right| \tag{21}$$

$$\text{fine-grid convergence index: } GCI_{fine}^{21} = \frac{1.25 e_a^{21}}{r_{21}^p - 1} \tag{22}$$

Verification

Estimate uncertainty: Draw error bars

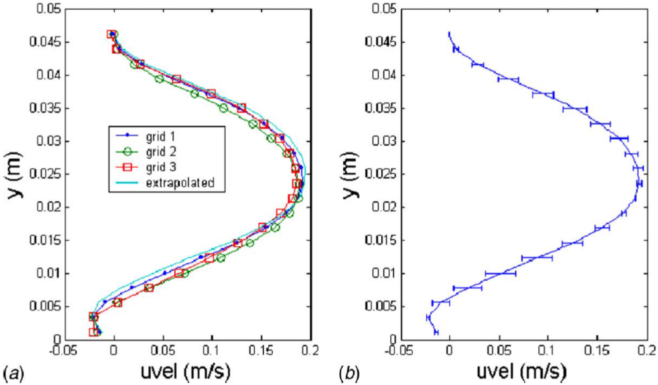


Fig. 2 (*a*) **Axial velocity profiles for a two-dimensional laminar backward-facing-step flow calculation [16]; (*b*) Fine-grid solution, with discretization error bars computed using Eq. (7)**

Figure: Example error estimation (Journal of Fluid Engineering)

Validation

- The goals of the validation:
  - minimize the uncertainties and errors in the computational model
  - increase and evaluate the confidence level of the predictions from the computational model
- Validation usually needs experimental data
- However, experimental data also has uncertainty and errors
- Notations: For a physical variable T_0
  - S: predicted value from computational model
  - D: value determined from experiment
  - T: TRUE value (usually unknown)
- Definition of errors:
  - Validation comparison error: $E = S - D$
  - True error in the solution value: $\delta_S = S - T$
  - True error in the experimental value: $\delta_D = D - T$
- It is easy to see: $E = \delta_S - \delta_D$

Validation

- Source of errors:
  - error due to modeling assumption and approximations
  - error due to the numerical solution of governing equations (discretization, linear system solver, etc.)
  - error due to the input parameter and coefficients

$$\delta_S = \delta_{model} + \delta_{num} + \delta_{input} \tag{23}$$

- The objective of validation is to estimate and quantify δ_{model}
- Comparison between computational results and experiments

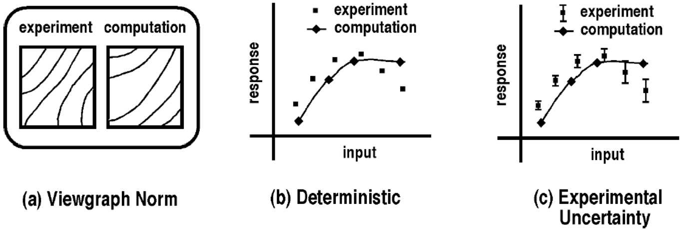
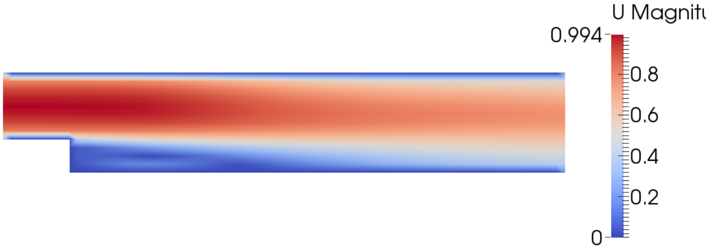


Figure: Qualitative validation comparison (Oberkampf and Roy, 2010)

Validation

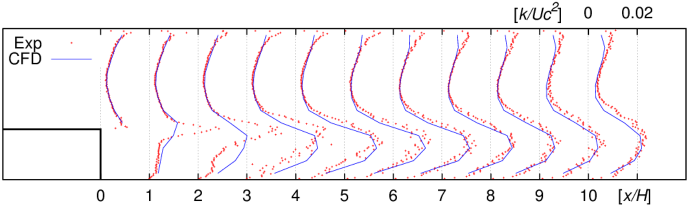
Example: Turbulent backstep flow

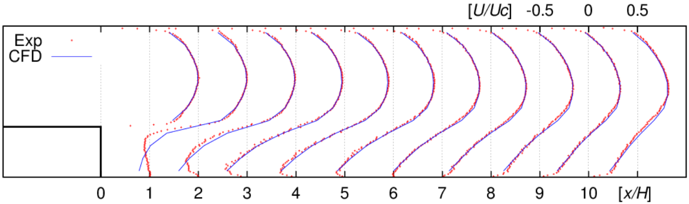
- Reference: Nobuhide Kasagi, Akio Matsunaga: Three-dimensional particle-tracking velocimetry measurement of turbulence statistics and energy budget in a backward-facing step flow, International Journal of Heat and Fluid Flow, Vol.16, No.6, pp.477-485, 1995
- Case setup from `http://www.opencae.jp/svn/OpenFOAM-VandV-SIG/Misc/trunk/turbulentBackstep/`
- It also provides experimental data and plotting scripts



Validation

Example: Turbulent backstep flow





Calibration

What about calibration?

- Definition: the process of adjusting numerical or physical modeling parameters in the computational model to improve the agreement between model results and (trustable) data
- Calibration is usually conducted before validation
- Experimental data used for calibration should not be used again for validation.

Further readings

Further readings on V&V:

- Oberkampf, W.L. and Roy, C.J., Verification and Validation in Scientific Computing, Cambridge: Cambridge University Press (2010)
- Roache , P.J. Verification and Validation in Computational Science and Engineering, Hermosa Publishers, Albuquerque, 1998
- ASME V&V 20-2009, Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer
- http://www.grc.nasa.gov/WWW/wind/valid/homepage.html

Questions?