

Overview of NEAMS VU Verification Plan

Sept. 15, 2009

Brian Carnes
Validation and Uncertainty Quantification Processes
Sandia National Laboratories



Overview

- **What is covered**
 - **Software quality practices**
 - **Code verification**
 - **Solution verification**
- **What is not covered**
 - **Uncertainty quantification**
 - **Validation**
 - **Sensitivity analysis**



Source Material

- **DOE/ASC guides**
- **AIAA**
- **ASME**
- **ANS**
- **Technical reports (SNL, LANL)**
- **Journal articles**
- **Books (Roache, Knupp & Salari)**



Errors and Uncertainties

- AIAA defines for **uncertainty** as:
 - a potential deficiency in any phase or activity of the modeling process that is due to the lack of knowledge.
- **error** as:
 - a recognizable deficiency in any phase or activity of modeling and simulation that is not due to lack of knowledge
- This report is concerned only with **errors** and not uncertainty



Motivation for Software Quality and Verification

- **Reduce risk of high consequence code errors**
- **Quantify numerical errors as part of validation**
- **Reduce development and maintenance costs by finding errors sooner**
- **Reduce numerical errors through mesh adaptivity**
- **Assist in NRC licensing process by providing evidence of code quality**



Software Quality

- **Quality is traditionally about satisfying functional requirements set by customers.**
- **Quality according to ASC: “Conformance to customer requirements and expectations”**
 - **Project management, software engineering, software verification**
- **External quality characteristics – users (McConnell)**
 - **correctness, usability, efficiency, reliability, integrity, adaptability, accuracy, and robustness**
- **Internal characteristics - developers**
 - **maintainability, flexibility, portability, reusability, readability, testability, and understandability**



Software Quality Practices

- **Define project-specific software quality goals**
- **Use a version control system for all source code, tests, and supporting documentation**
- **Document code usage and theory in a User Manual**
- **Run all test suites regularly and maintain a clean test suite**
- **Track all support requests (code errors, feature requests)**
- **Adopt practices to promote better code construction and documentation (code reviews, self-documenting code, training, etc.)**
- **Define procedures for checkin, release, resolution of build/test failures**



Code Verification Practices

- **Document all mathematical model equations and numerical models**
- **Implement support for Method of Manufactured Solutions (MMS), including user-defined source terms, initial/boundary conditions, and material properties.**
- **Construct and regularly run a suite of order of verification tests using the MMS approach**

Code Verification Example

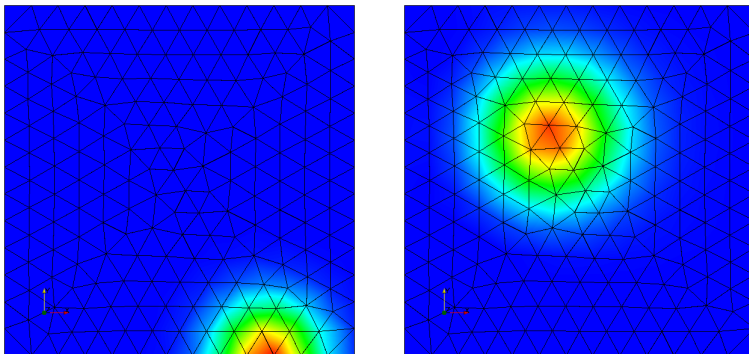
- Goal: verify theoretical convergence rates on test problem
- Compare numerical solution with reference (analytic) solution using error metrics

$$\|u - U\| \quad Q(u) - Q(U)$$

- Fit error model (steady/transient) to errors on different grids

$$E(h, k) = Ch^\alpha + Dk^\beta,$$

Transient order of converge parameters for error in gradients. Linear finite elements and second order time integration.



Transient diffusion problem with source term chosen to produce advecting exponential solution (left is $t=0$, right is $t=1$)

Level	C	D	alpha	beta
1	4.42	-0.752	0.618	0.566
2	6.22	0.137	0.976	2.48
3	8.01	0.531	1.07	2.82
4	5.68	0.691	0.970	2.81
5	6.99	0.857	1.02	2.58

Space-time nonlinear least squares fitting using two grids and two time steps



Solution Verification Practices

- **Insure at least two-three grid resolutions are available for any problem.**
- **Quantify non grid-related numerical error (iterative solver controls, contact tolerances)**
- **Quantify grid-related errors in response functions using Richardson extrapolation**
- **Evaluate implementation of a posteriori error estimation and adaptivity**

Solution Verification Example

- **Goal: estimate actual convergence rates on real problems**
- **Richardson extrapolation: compare numerical response quantities computed on three different grids/time steps**

$$\alpha = \frac{\log\left(\frac{Q_1 - Q_2}{Q_2 - Q_3}\right)}{\log\sigma},$$

$$C = \frac{Q_1 - Q_2}{(h_1^\alpha - h_2^\alpha)},$$

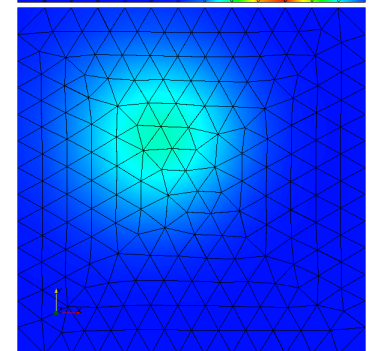
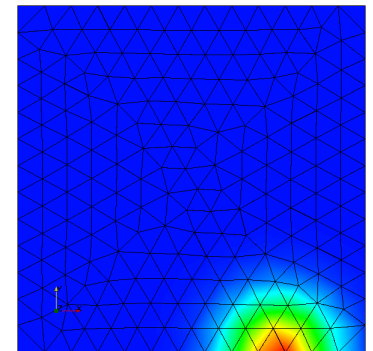
$$Q = Q_1 - Ch_1^\alpha$$

$$Q_{ij} = Q(h_i, k_j)$$

$$\begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ & Q_{22} & \\ & & Q_{33} \end{bmatrix}$$

Steady state requires 3 grids

Transient requires 3 grids/time steps



Transient advection-diffusion problem with exponential IC. Exact solution unknown.

Rates are estimated as 2nd order in space and time.

Level	C	D	alpha	beta	Q
1	-0.15919	1.5491	1.7424	10.212	0.055312
2	-0.32215	-1.3289	2.8858	2.2447	0.056667
3	-0.16251	-2.0058	2.7479	2.4672	0.056614
4	-0.074378	-1.0142	2.4408	2.1968	0.056621



Limitations of Richardson Extrapolation

- **Running on multiple grid scales is costly**
- **Errors can change significantly as model parameters are varied**
- **May not reach asymptotic regime**
- **Quantities may be non-monotone as grid is refined**
- **Realistic problems often do not exhibit theoretical converge rates**

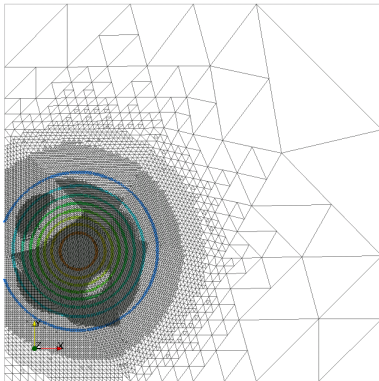


Error Estimation and Adaptivity

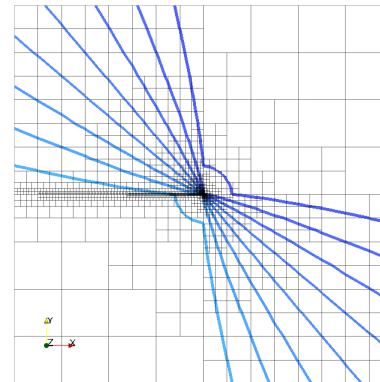
- **Goals:**
 - quantitative error estimates for a given grid
 - reduction of numerical error through adaptivity
- **Goal-oriented approach**
 - direct estimates of error in quantities of interest
 - requires intrusive implementation (solver, PDE residuals, numerical method)
 - rough cost: 20-50% of original simulation
- **Adaptivity**
 - For non-smooth problems efficiency improves by orders of magnitudes
 - Implementation can benefit from existing libraries

Examples of Error Estimation and Adaptivity

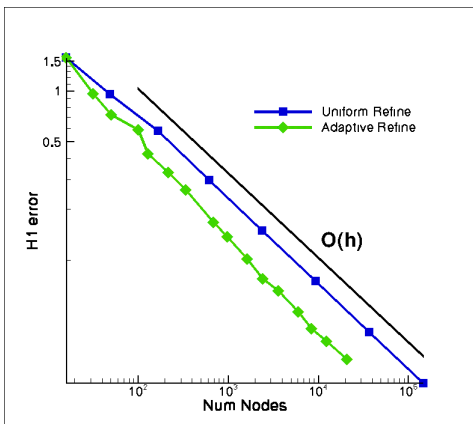
- Test problems with exact solutions: (1) smooth exponential (2) non-smooth multi-material problem



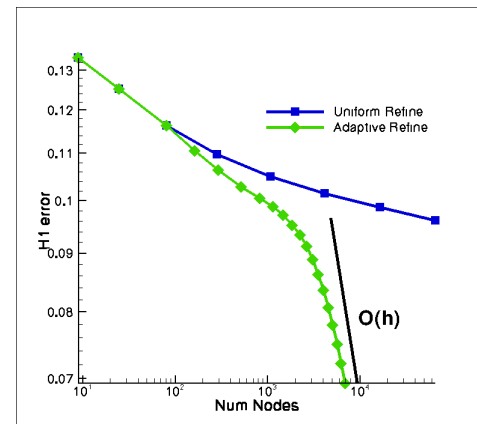
(1) Adaptivity based on error in gradients effectively resolves the solution



(2) Adaptivity is dramatic – six orders of magnitude in local mesh size.



(1) For smooth problems we get fractional increase in efficiency



(2) For non-smooth problems we get orders of magnitude increase in efficiency



Summary

- **FY09: NEAMS plan for Verification to be completed by 10/1/09**
- **FY10: interaction with NEAMS teams for assistance with**
 - **Software quality**
 - **Code verification**
 - **Solution verification**
 - **Error estimation and adaptivity**