Verification through Adaptivity

Brian Carnes
Bart van Bloemen Waanders
Kevin Copps
bcarnes@sandia.gov

Sandia National Laboratories
Albuquerque, NM

August 18-20, 2009
NCAR Frontiers of Geophysical Simulation Workshop
SAND09-5712C

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Contents

- Motivation
- Verification
- Error Estimation
- Adaptivity
- Applications

Motivation

- Geophysical events take place over vast time and length scales
- They also include phenomena at highly disparate scales
- The computational modeler must decide the relevant physics/scales/coupling to include and what to ignore
- Predictions from computational simulations must be high quality in order to defend against criticism
- The issue is to define quantitative measures of simulation quality

Predictive Simulation

- We need to keep in mind an overall process:
  1. Verification (Code and Solution)
  2. Calibration
  3. Validation
  4. Predication + Error/Uncertainty Quantification
- Verification is solving the equations correctly (numerical accuracy)
- Calibration is tuning parameters to agree with appropriate experimental data
- Validation is solving the correct equations (model accuracy)
- Then we are ready to make predictions with their associated errors and uncertainties
- Uncertainty quantification requires a large number of simulations

Overall Uncertainty Budget

- Sources of uncertainty in experiment and prediction (from Brian Adams, SNL):
  - parametric uncertainty (random fields and processes)
    - ★ physical parameters
    - ★ statistical variation, inherent randomness
    - ★ operating environment, interference
    - ★ initial, boundary conditions, forcing data
    - ★ geometry, structure, connectivity
    - ★ material properties
    - ★ manufacturing quality
  - model form (e.g., equation of state)
  - programmatic (policy decisions, requirements)
  - human reliability, subjective judgment
  - experimental error (measurement error, bias)
  - numerical accuracy (mesh, solver, approximations, etc.)

Code Verification

- Code verification is a process of determining that the numerical algorithms in the code converge correctly.
- Adopted by many orgs (AIAA, ASME, ANS, DoD, DOE/NNSA)
- “We are solving the equations right”
- Can be very useful for determining coding errors
- Pass/fail based on comparing observed and theoretical convergence rates
- Requires test problems that satisfy conditions for theoretical rates
  - Example: IC = 300 K, BC = 500 K - non-smooth jump at boundary
- Cannot detect errors resulting in inefficient algorithms
- Code is “verified” by the accumulation of verification test evidence

Grid Refinement Studies

- Measuring convergence requires sequence of grids / time steps
- Standard approach is Method of Manufactured Solutions (MMS)
- Began in CFD community, but long used in numerical analysis
- Requires (arbitrary) analytical solution $u^{EX}(x,t)$
- PDE forcing data and BCs computed by substitution of $u^{EX}(x,t)$
- Errors can be measured using response quantities

$$\mathcal{Q}(u^{EX}) - \mathcal{Q}(U)$$

  or norms

$$\|u^{EX} - U\|$$

- Norms are specific to PDE and numerical method

Estimating Error Rates

- Uniform grids: can relate mesh size (h) to DoFs (N):

$$h \approx C N^{-1/d} \quad d \text{ is spatial dimension}$$

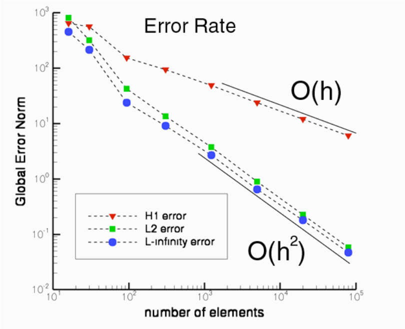
- Since the exact solution is known, any two grids can be used
- Spatial error rate for two grids:

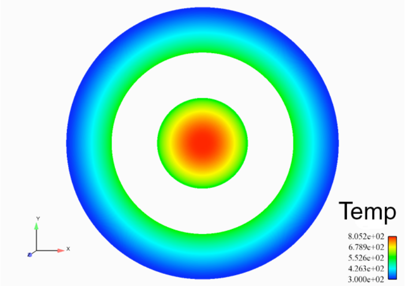
$$p \equiv \frac{\log\left(\frac{\|u^{EX} - U_1\|}{\|u^{EX} - U_2\|}\right)}{\log(h_1/h_2)} = -d \, \frac{\log\left(\frac{\|u^{EX} - U_1\|}{\|u^{EX} - U_2\|}\right)}{\log(N_1/N_2)}$$

- The second form is more general - can easily apply to highly graded or adaptive grids
- Time error can be included with appropriate modifications.

Example: Thermal Verification

- Heat conduction in solids, enclosure radiation within void space
- Analytic solution is piecewise radial function
- Finite element norms were used (H^1, L^2, L^∞)
- Observed rates agreed with theoretical rates





Barriers to Code Verification

- Codes must support subroutines for ICs, BCs, source terms
- Grids must conform to geometry as mesh size $\to 0$.
- Subgrid physics modules
  - May be calibrated to a specific mesh
  - May not converge as the mesh is refined
- Modules not based on ODE/PDEs
- Inability to reach the asymptotic regime (where observed convergence rates stabilize)
- Non-monotone response quantities
- Adaptive time stepping and mesh refinement

Solution Verification (Numerical Error Estimation)

- Code Verification applies to the general purpose accuracy of a code (e.g., code is second order)
- Solution or Calculation Verification applies to specific problems approximated by the code
  - PDE model
  - geometry
  - material/constitutive models
  - grid or sequence of grids
  - initial, boundary conditions; forcing data
  - parameter ranges of interest
- In Code Verification these are suitably varied to cover the intended code usage
- Most general purpose Solution Verification technique is Richardson Extrapolation

Generalized Richardson Extrapolation: Response Quantities

- Assumed error model: $Q_i = Q + C\, h_i^{\alpha}$
- We do not assume that the convergence rate α is known
- Consider three uniform grids with mesh size $h_1 > h_2 > h_3$
- Constant mesh size ratio $\sigma = h_1/h_2 = h_2/h_3 > 1$.
- The Richardson Extrapolation estimate of (Q, α, C) are

$$\alpha \approx \frac{\log\left(\frac{Q_1 - Q_2}{Q_2 - Q_3}\right)}{\log \sigma}, \quad C \approx \frac{Q_1 - Q_2}{(h_1^{\alpha} - h_2^{\alpha})}, \quad Q \approx Q_1 - C\, h_1^{\alpha}$$

- Can use as extrapolated value or as error estimate $(Q - Q_i)$

Richardson Extrapolation: Limitations

- Requires monotone values of the response quantity:

$$(Q_1 - Q_2)/(Q_2 - Q_3) > 0$$

- Requires response quantity to be in asymptotic regime:

$$(Q_i - Q)/h_i^{\alpha} \approx C$$

- For large models three uniform grid refinements may be too expensive
- Similar issues as code verification: subgrid models, non-ODE/PDE models, etc.

Example: Well-Behaved Problem

- 2D Poisson equation on unit square with MMS solution:

$$u(x,y) = \exp(-((x-x_0)^2 + (y-y_0)^2)/\sigma_0)$$

- Linear finite elements on triangles
- Response function is average value: $Q(u) = \frac{1}{|\Omega|}\int_\Omega u\,dx$

N	Q	α	C	Q^{EX}
16	3.52262e-01	-	-	-
49	3.76728e-01	1.97800	-3.54810e-01	3.85052e-01
169	3.82939e-01	1.99072	-3.62391e-01	3.85027e-01
625	3.84502e-01	1.99651	-3.67375e-01	3.85024e-01
2401	3.84893e-01	-	-	-

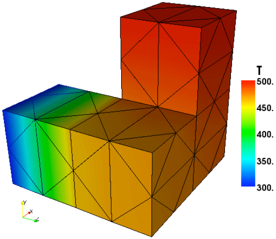
- Exact value is 3.85024405e-01.

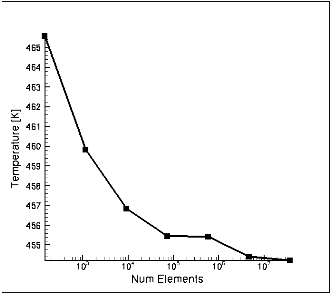
Example: Ill-Behaved Problem

- Stationary heat conduction; four blocks with different properties.
- Geometric and material singularities.
- Response function is point value: $Q(u) = u(x_0, y_0)$

N	Q	α	Q^{EX}
1.46e+02	465.56	-	-
1.16e+03	459.82	0.9482	453.64
9.34e+03	456.84	1.0888	454.20
7.47e+04	455.46	5.8664	455.42
5.98e+05	455.42	-5.4123	455.44
4.78e+06	454.40	2.5375	454.18
3.82e+07	454.22	-	-

- Non-smooth problems and localized quantities (point values) can complicate RE.
- Validation/prediction must account for this numerical error.





Richardson Extrapolation: Further Details

- Not assuming rate allows use in case of non-smooth data
- Fixed mesh ratio not required
- Can be extended to time-dependent problems, anisotropic meshes, pointwise errors in solution variables (Kamm and Hemez, LANL)

$$Q_i = C\, h_i^{\alpha} + D\, k_i^{\beta} + E\, h_i^{\gamma}\, k_i^{\delta} + \dots$$

- Can be extended to adaptive meshes (using number of unknowns instead of mesh size)

A Posteriori Error Estimation

- Mesh convergence studies performed for nominal (fixed) parameters can be impractical or unreliable:
  - parametric studies
  - optimization
  - uncertainty quantification
- A posteriori error estimation: estimating the numerical error using a single mesh in the absence of an exact solution.
- Goal: provide quantitative estimate of numerical error
- Requires analysis that can be specific to the mathematical model
- Dependent on the numerical method: finite element, finite volume, finite difference.

Basic Types of Error Estimators

- Averaging/Recovery/Reconstruction operators
  - compute a smoother numerical solution using averaging.
  - the error estimator is the norm of the difference.
- Residual based
  - Explicit: compute PDE equation residuals scaled by mesh size.
  - Implicit: project PDE residuals against higher order basis
- These are based on estimating the error in norms.
- May not be robust across mathematical models.
  - elliptic -> parabolic requires small time step
- Goal oriented
  - Based on quantities of interest
  - General purpose approach - applicable to many PDEs

Errors and Residuals

- Consider a linear PDE with operator L and data f:

$$L\,u = f$$

- If U is an approximate solution, the error $e \equiv u - U$ solves

$$L\,e = f - L\,U \equiv R(U) \qquad \text{the residual}$$

- The error is equivalent to the residual up to conditioning of L

$$\|L\|^{-1}\,\|R\| \le \|e\| \le \|L^{-1}\|\,\|R\|$$

- For nonlinear, transient problems the problem needs to be locally stable near u and U.

Residuals and Adjoints

- Given response function Q, solve adjoint problem:

$$L^* \phi = Q$$

- The error in the quantity is then

$$\begin{aligned} Q(u) - Q(U) &= Q(e) \\ &= \langle L^* \phi, e \rangle = \langle L e, \phi \rangle \\ &= \langle R(U), \phi \rangle \\ &= \langle R(U), \phi - I_h \phi \rangle \end{aligned}$$

- The last equation uses finite element error orthogonality.
- The residual is weighted by the error in the adjoint.

Goal-oriented Error Estimation: Transient Case

- Example: transient convection-diffusion

$$u_t + V \cdot \nabla u - k \triangle u = f$$

- The adjoint is linearized around U and runs backwards in time
- Quantity of interest can be initial/forcing/flux BC data

$$-\phi_t - \nabla \cdot (\phi V) - k \triangle u = Q_f, \quad \phi(\cdot, T) = Q_T$$

- Form of error estimate (Φ is space/time adjoint approximation):

$$\begin{aligned} Q(u) - Q(U) = & \sum_n \int_{I_n} (f - U_t - V \cdot \nabla U, \phi - \Phi)\, dt \\ & - \sum_n \int_{I_n} (k \nabla U, \nabla(\phi - \Phi))\, dt \\ & + (u_0 - U_0, \phi_0) \end{aligned}$$

Local Error Contributions

- The residual is broken into space/time contributions

$$\|R_{K,n}(U)\| = \|LU - f\|_{K,n} + \dots$$

- The adjoint weights are also calculated locally

$$\|\omega_{K,n}\| = \|\phi - I_h\,\phi\|_{K,n}$$

- Local residuals are weighted by their influence on the error in the global output.

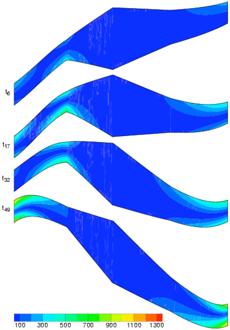
$$\text{Error Estimator} = \sum_{n=1}^{N} \sum_{K} \|R_{K,n}(U)\| \, \|\omega_{K,n}\|$$

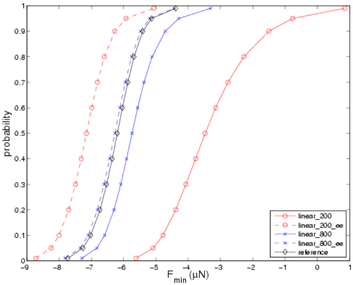
Goal-oriented Error Estimation Challenges

- Approximation to adjoint:
  - Solve adjoint using a higher order method
  - Solve adjoint using same order method + postprocessing
- Computational cost - balance increased accuracy against cost
- Requires more access to code internals, more developer time.
- Loosely coupled physics can hide error propagation effects
- Time dependent problems introduce backward time integration, storage of forward solution.

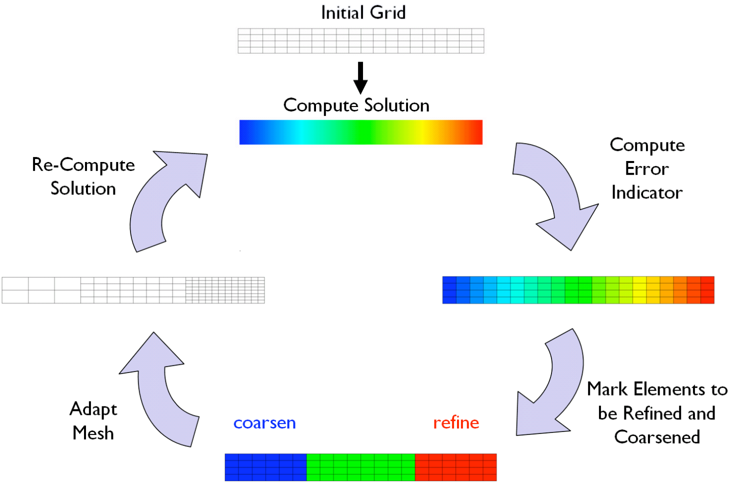
Example: Bistable MEMS Beam

- The response is the minimum force (as function of displacement)
- Several parameters are varied in a UQ study.
- The goal-oriented error estimator can correct for numerical error.





Adaptivity


Initial Grid
Compute Solution
Compute
Error
Indicator
Mark Elements to
be Refined and
Coarsened
coarsen
refine
Adapt
Mesh
Re-Compute
Solution

Adaptivity Implementation

- Types of adaptive mesh refinement (AMR)
  - Block structured AMR
  - Unstructured AMR
  - External remeshing using mesh size function
- Components of error estimation
  - Compute error indicators (local contributions)
  - Mark elements for refinement/coarsening
  - Refine mesh by creating/deleting elements
  - Prolong/restrict solution fields to new mesh
- Modifications to numerical algorithms from hanging nodes
  - Finite element: hanging node constraint enforcement
  - Finite volume: flux computation on refined faces

Adaptivity and Solution Verification

- Recall the error rate for uniform refinement

$$\text{Error} = C\,h^{\alpha} = C\,N^{-\alpha/d}$$

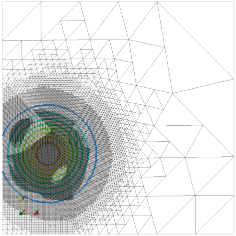
- For realistic problems we have $\alpha < \alpha^{OPT}$
  - discontinuous material properties
  - irregular forcing data, BCs, ICs
  - geometric singularities (e.g., reentrant corners)
- Adaptivity can restore the optimal rate in terms of N by considering the larger class of adapted meshes

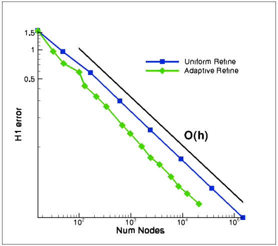
$$\text{Error} = C\,N^{-\alpha^{OPT}/d}$$

  This leads to exponential gains in efficiency for same accuracy
- For problems that are smooth but highly variable, adaptivity can only reduce the constant C

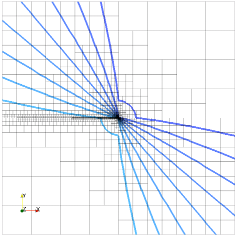
Example: Adaptivity and Error in Norms

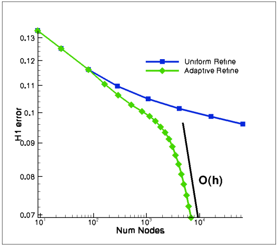
- Smooth localized exponential solution





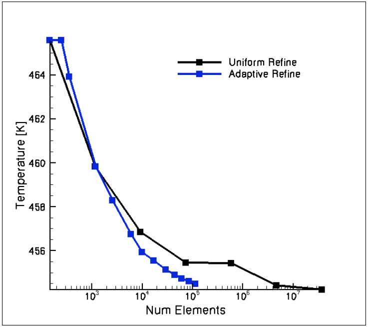
- Solution with discontinuous material properties

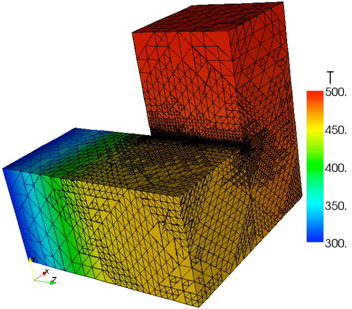




Example: Adaptivity and Point Value

- Stationary heat conduction; four blocks with singularities.
- Response function is point value: $Q(u) = u(x_0, y_0)$
- Adaptivity based on gradient recovery (H^1 norm error estimator)





- Result not typical: error estimator not based on error in Q

Adaptivity and Goal-oriented Error Estimation

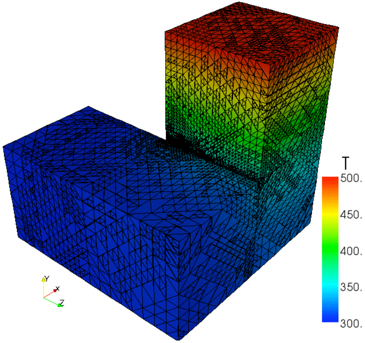
- Typically element error contributions are nonnegative.
- Most accurate form of local contribution is signed: but can bound contribution by absolute value

$$\eta_{K,n} = \langle R_{K,n}(U), \phi - I_h\,\phi \rangle \leq \|R_{K,n}(U)\|\,\|\phi - I_h\,\phi\|$$

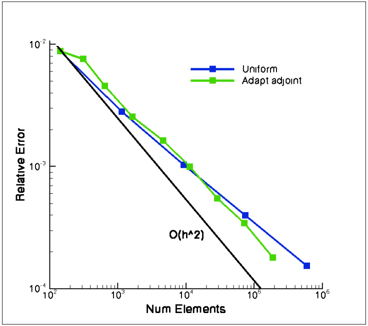
- Using the signed values requires new approaches to marking elements for adaptivity
  - Statistical: mark elements for refinement if outliers: $|\eta_{K,n} - \mu| > \gamma_R\,\sigma$
  - Cancellation: use cancellation of positive and negative contributions to reduce error
  - Correlations: refine/coarsen where similar contributions cluster

Application: Transient Thermal

- Response is average temperature in one block at final time
- Spatial error dominated, transient error controlled by IBDF
- Adaptivity as outer loop around transient solve



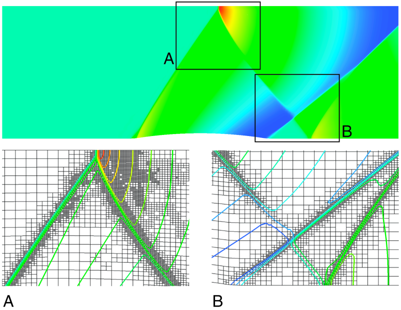
(a) Temperature (t=15)



(b) Convergence rates

Application: High Speed Flow

- Adaptivity can resolve shocks in high speed compressible flow
- Stabilized finite element Euler equations in SNL code Aria
- Here the error indicator is jumps in density gradients



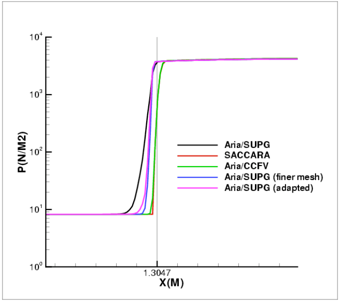
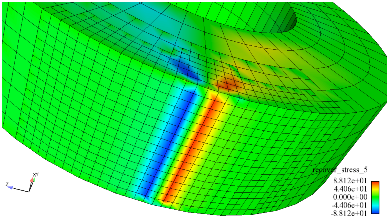


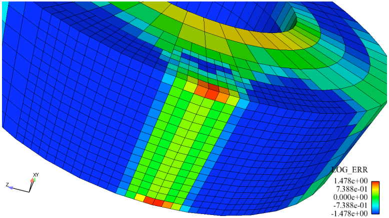
Figure: (left) pressure and density contours (right) comparison of stagnation line profile

Application: Solid Mechanics

- Modeling of rolling tire contact with road surface
- Adaptivity based on prescribed refinement box
- Error indicator computed to contrast with feature-based refinement



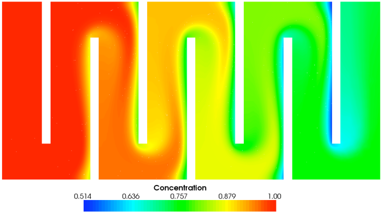
Recovered σ_{yz}



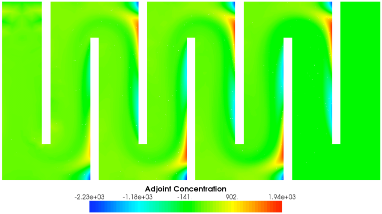
Log of Indicator

Application: Optimal Control

- Contaminant transport and surface reaction in contact tank reactor
- Surface reaction rates (six) are control parameters
- Objective function is least squares misfit to prescribed surface concentration (linear in x-direction)
- Goal-oriented error estimator applied to objective functional



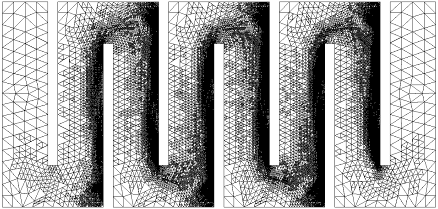
Optimized concentration



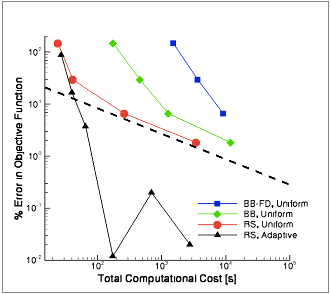
Adjoint concentration

Application: Optimal Control and Adaptivity

- Adaptivity loop around optimization solver (Trilinos/MOOCHO)
- Acceleration of optimization using re-use of reduced Hessian between adaptive meshes
- Compared black box & embedded optimizers
- Adaptivity and embedded optimizer most efficient combination



Final adapted mesh



Accuracy versus CPU time

Conclusions

- Overview of verification, error estimation and adaptivity
- Described barriers and limitations of verification and error estimation
- Demonstrated effectiveness of goal oriented error estimation (transient, optimization)

Future Work

- Optimal balance of efficient and accurate goal oriented error estimation (how much higher order do we need?)
- Higher order adjoint for various time integration schemes (relate methods to Galerkin + quadrature)
- Optimal adaptivity for transient problems (adaptive meshes for large time blocks)