A Multilevel Method for Solution Verification

Marc Garbey and Christophe Picard

University of Houston, Computer Science (http://www.cs.uh.edu/~garbey/)

Summary. This paper addresses the challenge of solution verification and accuracy assessment for computing complex Partial Differential Equation (PDE) model. Our main target applications are bio-heat transfer and blood flow simulation problems. However our long term goal is to provide a postprocessing package that can be attached to any existing numerical simulation package, for example widely used commercial codes such as ADINA, Ansys, Fluent, Star-CD etc., and provide an a posteriori error estimate to their simulation.

1 Introduction and Motivation

The problem of accuracy assessment is a necessary step that follows the code verification step and precedes the code validation step, completing the global task of providing a reliable virtual experiment tool [5].

Our major goal in this paper is to pursue our work on the design of a new multilevel method that offer a general framework to do Solution Verification (SV) efficiently. The standard approach in applied mathematics to handle the problem of SV is to work on the approximation theory of the PDE. For each specific PDE problem, the right Finite Element (FE) approximation may provide the correct a posteriori error estimate. Unfortunately this approach may require a complete rewriting of an existing CFD code based on Finite Volume (FV) for example and lack generality.

Our method relies on four main ideas that are (1) the embedding of the problem of error estimation into an optimum design framework that can extract the best information from a set of two or three existing numerical results, (2) the resolution of the problem as much as possible as a (non)linear set of discrete equations to produce a general tool, and renounce on using the specific approximation theory used the compute the PDE solution. Since we usually have no access to the detailed knowledge of the internal structure of the code that produces the numerical solution, (3) the formulation of a framework that can reuse any a posteriori estimator if they are available (4) the use of distributed computing (or grid computing) to get a cost effective SV.

2 Method

From the applied mathematics point of view, a posteriori estimates have been around for many years [1, 8]. There is a vast literature on this subject. The main challenge is still *to estimate numerical accuracy on under-resolved grids* [5]. As a matter of fact, in complex modeling, as described in the ASCI project, best grid solutions provided by our best computing resources are fairly under-resolved at least locally.

We present in this paper an entirely different framework to construct reliable a posteriori estimates for *general* PDEs or system of PDEs. Let us first describe the general concept of our method [2, 3, 7].

2.1 General Concept

We consider a boundary value problem (Ω is a polygonal domain and $n = 2$ or 3) :

$$L[u(x)] = f(x),\ x \in \Omega \subset \mathbb{R}^n,\ u = g \text{ on } \partial\Omega. \tag{1}$$

We assume that the PDE problem is well posed and has a unique smooth solution. We consider a finite volume approximation of (1) on a family of meshes $M(h)$ parametrized by $h > 0$ a small parameter. The smaller h the finer should be the discretization. We denote symbolically the corresponding family of linear systems

$$A_h U_h = F_h. \tag{2}$$

Let p_h denotes the projection of the continuous solution u onto the mesh $M(h)$. We assume a priori that ($||.||$ is a given discrete norm):

$$||U_h - p_h(u)|| \ \to\ 0, \text{ as } h\ \to\ 0, \tag{3}$$

Let $M(h_1)$ and $M(h_2)$ be two different meshes used to build two approximations U_1 and U_2 of the PDE problem (1). A consistent linear extrapolation formula should have the form

$$\alpha U_1 + (1-\alpha) U_2,$$

where α is a weight function. In classical Richardson Extrapolation (RE) the α function is a constant. In our optimized extrapolation method α is an unknown space dependent function solution of the following optimization problem, where G is an objective function to be defined:

P_α: *Find* $\alpha \in \Lambda(\Omega) \ \subset\ L_\infty$ *such that* $G(\alpha\, U_1 \ +\ (1-\alpha)\, U_2)$ *is minimum.*

The Optimized Extrapolated Solution (OES) if it exists, is denoted $V_e = \alpha U_1 + (1-\alpha)U_2$. For computational efficiency, $\Lambda(\Omega)$ should be a finite vector space of very small dimension compared to the size of matrix A_h defined in (2). The objective function G might be derived from any existing a posteriori error estimators if possible. For a number of fluid dynamic methods used in bioengineering such as the immersed boundary technique, or the chimera technique there is no solid theoretical framework that can provides such rigorous a posteriori estimators. For complex bioengineering problems, the fact that there exist a functional space framework to derive a posteriori estimate is more the exception than the generality. Our ambition is to provide a numerical estimate on $||U_j - U_\infty||$, $j = 1, 2$, without computing U_∞ effectively. The solution U_j can then be verified assuming (3). The fine mesh $M(h_\infty)$ should be set such that it captures all the scales of the continuous solution with the level of accuracy required by the application. We have a priori $h_\infty \ll h_1,\ h_2$. Both coarse grid solutions U_1 and U_2 must be projected onto $M(h_\infty)$. We will denote $\tilde{U}_1$ and $\tilde{U}_2$ the corresponding functions. We choose then to minimize the consistency error for the numerical approximation of (1) on a fine mesh $M(h_\infty)$. The objective function is then

$$G(U^\alpha) = ||A_{h_\infty}\, U^\alpha - F_{h_\infty}||, \text{ where } U^\alpha = \alpha\, \tilde{U}_1 \ + \ (1-\alpha)\, \tilde{U}_2. \tag{4}$$

The choice of the discrete norm should depend on the property of the solution. In the Least Square Extrapolation (LSE) method [2, 3] we chose the discrete L_2 norm. The choice of the L_1 or the L_∞ norm provides some useful additional information, for example, for stiff elliptic problems.

One of the difficulties encountered with a two-level extrapolation method is that there exists subsets of $M(h_\infty)$ where $\tilde{U}_1$ and $\tilde{U}_2$ are much closer to each other than what the expected order of accuracy based on local error analysis should provide. In such areas, the sensitivity of the extrapolation to the variation of α is very weak and the problem is ill posed. These subsets should be treated as outliers of the optimization computation procedure. A potentially more robust procedure consists of using three levels of grid solution. The optimization problem writes then

$P_{\alpha,\beta}$: *Find* $\alpha, \beta \in \Lambda(\Omega) \ \subset \ L_\infty$ *such that* $G(\alpha\, U_1 \ + \ \beta\, U_2 \ + \ (1-\alpha-\beta)\, U_3)$ *is minimum.*

We notice that if all $U_j,\ j = 1, \ldots, 3$, coincide at the same space location there is either no local convergence or all solutions U_j are exact. In such a situation, one cannot expect improved local accuracy from any OES. The robustness of OES should come from the fact that we do not suppose a priori any asymptotic on the convergence rate of the numerical method as opposed to RE.

Let us assume that the optimization problem P_α or $P_{\alpha,\beta}$ has been solved and that we have computed an optimum solution V_e either from the two levels or three levels method. We are going to discuss now its application to provide a posteriori error estimators.

Let us denote U_j to be one of the coarse grid approximations at our disposal. A global a posteriori estimate of the error $||U_j - p_h(u)||$ may come in two different ways. For the sake of simplicity we will assume that G is the L_2 norm of the residual (4).

• First is the recovery method based on the idea that the optimized extrapolated solution is more accurate than the coarse grid solution. Let us denote $\tilde{U}_j$ the coarse grid solution projected onto the fine grid $M(\infty)$ via a suitable interpolation procedure. Let us assume that the extrapolated solution is decisively more accurate than that based on interpolation from the coarse grid solution, namely,

$$||V_e - p_h(u)||_2 \ll ||\tilde{U}_j - p_h(u)||_2. \tag{5}$$

Then $||\tilde{U}_j - V_e||_2 \sim ||\tilde{U}_j - p_h(u)||_2$ and $||V_e - \tilde{U}_2||$ is a good error indicator to assess the accuracy on U_2.

We have seen in our numerical experiments with steady incompressible Navier-Stokes (NS) solutions that this method may give a good *lower* bound error estimate. But we do not know in general if the hypothesis (5) is correct. There is no guarantee that a smaller residual for V_e than for U_2 on the fine grid $M(h_\infty)$ leads to a smaller error.

• Second is a global *upper* bound that follows from a stability estimate with the discrete operator. We have

$$||V_e - U^0|| \; < \; \mu \, G(V_e), \text{ where } \mu \geq ||(A_{h_\infty})^{-1}|| \,,$$

where U^0 is the fine grid solution.

We conclude then

$$||\tilde{U}_2 - U^0||_2 \; < \; \mu \, G(V_e) + ||V_e - \tilde{U}_2||_2. \tag{6}$$

The procedure to derive an estimate for μ uses a combination of standard eigenvalue computation procedures applied to A_{h_j}, $j = 1, \ldots, 3$ and some extrapolation technique designed for scalar functions.

(6) is a good global a posteriori error estimator provided that

$$||U^0 - p_h(u)||_2 \; \ll \; ||U^0 - \tilde{U}_2||_2. \tag{7}$$

One way to test this hypothesis (7) is to measure the sensitivity of the upper bound (6) with respect to the choice of the fine grid $M(h_\infty)$. This is a feasible test because the fine grid solution is never computed in OES. Our verification procedure checks that $||U^0 - U_2||_2$ increases toward an asymptotic limit as $M(h_\infty)$ gets finer.

The algorithm procedure to construct V_e solution of P_α or $P_{\alpha,\beta}$ is straightforward when the operator is linear and the objective function is the discrete L_2 norm of the residual. Let e_i, $i = 1, \ldots, m$ be a set of basis function of $\Lambda(\Omega)$. The solution process can be decomposed into three steps.

• First, interpolation of the coarse grid solution from $M(h_j)$, $j = 1, \ldots, p$ to $M(h_\infty)$, with $p = 2$ for the two level method, respectively 3 for the three level method.

• Second, evaluate the residual $R[e_i \, (\tilde{U}_j - \tilde{U}_{j+1})]$, $i = 1, \ldots, m$, $j = 1, \ldots, p - 1$, and $R[\tilde{U}_p]$ on the fine grid $M(h_\infty)$.

• Third, the solution of the least square linear algebra problem that has m unknowns for each weight coefficient α and β used in the extrapolation procedure. In practice, m is much lower than the number of grid points on any coarse grid used.

We have generalized the LSE method to non-linear elliptic problems via a Newton like loop [2, 3]. We have also obtained preliminary results for unsteady parabolic problems [7]. Most of this work has been done on solutions produced by our own code on a fairly large variety of linear and nonlinear PDE problems on structured grids. To apply these techniques on solution produced by commercial code that have thousands of lines, and work with unstructured grids requires a more general and abstract approach, that we present in the next section.

2.2 Solution Verification of Off-the-Shelf CFD Code

We propose to generalize our method here to steady, CFD solutions produced by existing code. The challenge is that in most commercial codes, one cannot rely on the exact knowledge of the discretization method, neither have access to any information on the internal structure of the code. What we propose is fundamentally different than existing methods. We describe in the following the main ideas without seeking an exact formal mathematical description of a given specific PDE problem.

Let $(E, ||.||_E)$ and $(F, ||.||_F)$ be two normed linear space, $G \in L(E, F)$ be the operator corresponding to the CFD problem. Further let us denote $S \in F$ the input data of the CFD code and $U \in E$ the solution we are looking for.

In practice we look for an approximation of the accuracy of the solution U_h on the mesh $M(h)$ produced by the code $\mathcal{C}$ that operates on the data S_h: $\mathcal{C}: \; S_h \; \rightarrow U_h$. The objective is still to get an error estimate versus a very fine grid solution U_∞ that is never computed, because the cost is prohibitive. We will skip the index h when it is not essential. The space $E, \; F$ have (very large) finite dimensions indeed when they are for the discrete solutions on $M(h_\infty)$, and discrete data S_{h_∞}.

We assume that the code $\mathcal{C}$ has a procedure that provides the residual, i.e $V \rightarrow \rho = G(U_h) - G(V)$, where $V \in E, \; \rho \in F$. We note that this hypothesis is realistic, since most of the commercial code offer this feature or either provides a (first order explicit) time stepping procedure:

$$\frac{U_h^{n+1} - U_h^n}{dt} = G(U_h^n) - S \, . \tag{8}$$

The residual is then $\rho = \frac{U_h^1 - U_h}{dt}$. We assume that the following problem

$$G(u) \; = \; s, \; \forall s \in B(S, d)$$

is well posed for $s \in B(S, d)$, where B is a ball of center S and diameter d in $(F, ||.||_F)$. There should exist a unique solution for all data in $B(S, d)$ and the dependency of the solution on these data is supposed to be smooth enough to use a second order Taylor expansion.

Let us suppose that $G(U_h) \in B(S, d)$, that is

$$||\rho||_F = ||G(U_h) - S||_F < d. \tag{9}$$

We would like to get an error estimate on $e = U_h - U_\infty = G^{-1}(U_h) - G^{-1}(U_\infty)$. A Taylor expansion writes

$$G^{-1}(S) = G^{-1}(S + \rho) - (\rho \cdot \nabla_s) G^{-1}(S + \rho) + \frac{1}{2} \rho \cdot [\rho \cdot R(S)] \tag{10}$$

$$\text{where } ||R(S)||_E \leq K = \sup_{s \in B(S,d)} ||\nabla_s^2 G^{-1}(s)||_E. \tag{11}$$

Therefore

$$||e||_E \; \leq \; ||\rho||_F \; (||\nabla_s G^{-1}(S + \rho)||_E \; + \; \frac{K}{2} ||\rho||_F). \tag{12}$$

This completely general error estimate point out to two different tasks:

- Task 1: get an accurate upper bound on $||\nabla_S G^{-1}(S + \rho)||$
- Task 2: obtain a solution $U_\infty + e$ that gives a residual $||\rho||$ small enough to make the estimate useful, i.e. compatible with (9).

Task 2 is the purpose of the OES method, while Task 1 can be achieved by a perturbation method that can reuse the code.

2.3 Task 1: Stability Estimate

Let $\{b_i^E,\ i = 1,\dots,N\}$, (resp., $\{b_i^F,\ i = 1,\dots,N\}$) be a basis of E_h, (resp., F_h) and $\varepsilon \in \mathbb{R}$ such that $\varepsilon = o(1)$. Let $(V_i^{\mp})_{i=1,\dots,N}$, be the family of solutions of the following problems: $G(U_h \mp \varepsilon V_i) = S + \rho \mp \varepsilon b_i$. We get from finite differences the approximation

$$C_{h_\infty} = ||\nabla_S G^{-1}(S+\rho)|| \approx ||(\frac{1}{2}(V_j^+ - V_j^-))_{j=1,\dots,N}|| + O(\varepsilon^2).$$

We can get in as similar manner an approximation of the norm of the Hessian $\nabla_s^2 G^{-1}(S+\rho)$. For ρ small enough, we can verify that the upper bound is given essentially by:

$$||e||_E \ \preceq\ C_{h_\infty}||\rho||_F. \tag{13}$$

The column vectors $V_j^{\mp}$ can be computed with embarrassing parallelism. It is however unrealistic to compute these solutions that lies on the fine grid $M(h_\infty)$.

To make this task manageable, we have to reduce the dimension of the problem. We use the following two observations. While the solution of the CFD problem can be very much grid dependent, the conditioning number of the problem is in general much less sensitive to the grid. The idea is then to compute an approximation of C_{h_∞} by extrapolation from an estimate of two or three coarse grid computation of C_{h_j}. Further, let us assume that the fine grid $M(h_\infty)$ is a regular Cartesian grid. The number of terms to represent accurately the projected solution $\tilde{U}_j,\ j = 1,\dots,3$ with a spectral expansion or a wavelet approximation at a given accuracy is much less than the dimension of the coarse grid used in a Finite Element/Finite Volume computation. We propose to use preferably a grid M_{h_∞} that has enough regularity to allow a representation of the solution U_∞ with some form of compact representation, using either trigonometric expansion or wavelets.

The grid M_{h_∞} may have many more grid points than necessary, and therefore might not be computationally efficient for a true fine grid computation. But we do not have to do this computation anyway.

Let us denote $\hat{E}$ and $\hat{F}$ the spaces corresponding to one of these compact representation of the solution and residual. Let $(\hat{b}_j^{E/F}, j = 1,\dots,\hat{N})$, be the corresponding base with $\hat{N} \ll N$.Let $q_{E/F}$ be a mapping $E/F \to \hat{E}/\hat{F}$, respectively $q_{\hat{E}/\hat{F}}$ be a mapping $\hat{E}/\hat{F} \to E/F$ and let $\hat{C}:\ \hat{S}_h \to \hat{U}_h$. To summarize the procedure for Task 1, The estimate on C_{h_∞} will be applied to verify the code $\hat{C}$ based on the computation of $(\hat{V}_j^{\mp}, j = 1,\dots,\hat{N})$ vectors on the coarse grids $M(h_j),\ j = 1,\dots,3$ done by the code $\hat{C}$. We notice that the computation of the vector $\hat{V}_j^{\mp}$ can be done with embarrassing parallelism. Further because ε is small the code $\hat{C}$ can use as an initial guess in its iterative process the solution U_h that is hopefully very close to the unknown $\hat{U}_h \pm \hat{V}_j^{\mp}$.

2.4 Task 2: Optimized Extrapolation

We use here an optimized extrapolation method. To reduce the dimension of this problem we search for the unknown weight functions in a small space that can be described either by trigonometric expansion, or wavelet expansion, or possibly spectral elements. If Ω is the physical domain for the CFD solution, the unknown weight function can be search in a square domain $(0,1)^2$ modulo a change of variables. As a matter of fact no boundary conditions are required on the unknown weight functions. Let $\{\theta_j,\ j = 1, \ldots, m\}$ be the set of basis function of $\Lambda(\Omega)$.

We look for the solution of the optimization problem in the two level case

$$\text{Find } (\alpha_j) \in \mathbb{R}^m, \text{such that}$$
$$||G([\sum_{j=1,\ldots,m} \alpha_j \Theta_j]\tilde{U_1} + [1 - \sum_{j=1,\ldots,m} \alpha_j \Theta_j]\tilde{U_2})||_F \text{ is minimum.} \quad (14)$$

We have a similar formulation for the three level OES. Following the same argument than before we will rather look for this minimum in $\hat{F}$. As shown in [2, 3], we need a filtering process of the solution to have this minimization process numerically efficient. The postprocessing q_F is then useful. We can obtain easily the result when the weight function is a scalar function. To make this computation robust we use a response surface methodology [4] that is rather trivial in the scalar case. This procedure consist to compute a lower order polynomial best fit of the function $||G(\alpha\tilde{U_1} + (1-\alpha)\tilde{U_2})||$ by sampling α according to the expected convergence order range of the code. The minimization on α is then done with this polynomial approximation by a standard method. The sampling process is a cumbersome embarrassing parallel process that can take advantage of a computational grid [6].

3 A Numerical Example

To illustrate the pertinence of our methodology, let us present a Navier-Stokes back step flow example. The computation is done with ADINA. The ADINA system is a comprehensive finite element software that enable analysis of structures, fluid simulations, and fluid flows simulations with structural interactions.

Figure 1 shows an example of an unstructured mesh calculation of the back flow step problem at Reynolds number 500.

In this simulation, the number of elements are respectively 10347 on the fine grid G^∞, 1260 on the coarse grid G_1, and 2630 on the coarse grid G_2.

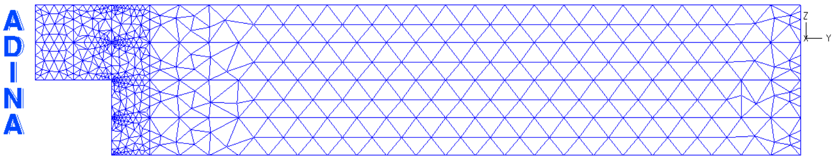

Fig. 1. Coarse mesh for the backstep

The steady solutions are obtained using a transient scheme for the incompressible Navier-Stokes equation.

For this test case OES outperforms the accuracy of the RE method by one order of magnitude - see Figure 2. An accurate error estimate is obtained for a representation of the solution on a 20 × 20 trigonometric expansion - see Figure 3. Let us conclude this paper with the design of the software that we are developing as a solution verification system independent of the CFD code.

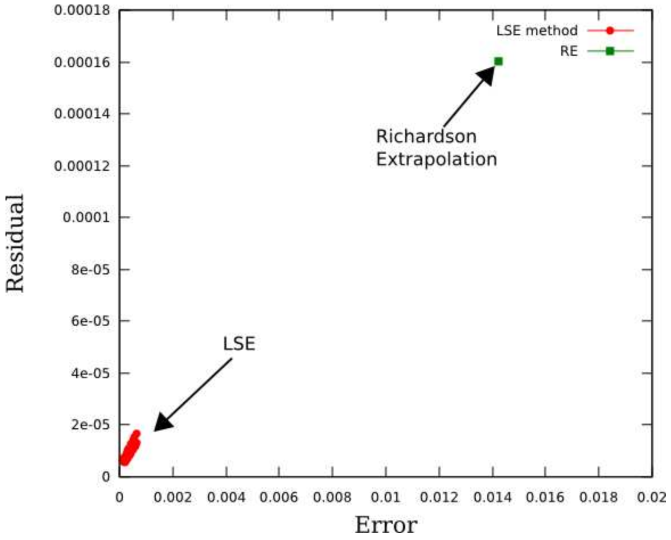

Fig. 2. Performance of LSE and Richardson Extrapolation.

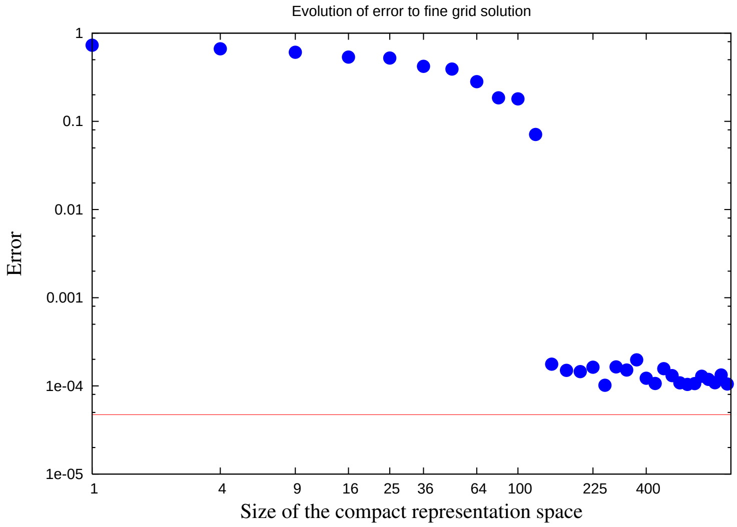

Fig. 3. Error estimate

4 Scientific Software Design and Conclusion

Our algorithm gives rise to a large set of cumbersome computations that can be done in parallel with a minimum of synchronization. This is a key feature to make our SV cost effective. We are developing a network oriented interface that allow our SV method to be executed remotely on several processing units, using the following methodology:

(i) a three-tier client server model architecture: it allows the system to be transparent, the user should not have to worry about technical details, to be open, each subsystem is open to interaction with the others, and to be scalable, the system should be easy to modify as the number of resources, users, softwares evolved.

(ii) Portability: to be able to run on UNIX/Linux/Windows platform

(iii) Security in data transfer, because industrial applications as well as computation on clinical data require that the data be protected.

(iv) Friendly user interface.

Some preliminary result on the performance of our distributed computing system for SV are reported in [6].

References

[1] M. Ainsworth. Identification and a posteriori estimation of transported errors in finite element analysis. *Comput. Methods Appl. Mech. Engrg.*, 176(1):3 – 18, 1999.

[2] M. Garbey and W. Shyy. A least square extrapolation method for improving solution accuracy of PDE computations. *J. Comput. Phys.*, 186:1 – 23, 2003.

[3] M. Garbey and W. Shyy. A least square extrapolation method for the a posteriori error estimate of the incompressible Navier-Stokes problem. *Internat. J. Numer. Methods Fluids*, 48:43 – 59, 2005.

[4] R.H. Myers and D.C. Montgomery. *Response Surface Methodology : Process and Product Optimization Using Designed Experiments.* Wiley, 2002.

[5] W.L. Oberkampf and T.G. Trucano. Verification and validation in computational fluid dynamics. Technical Report 2002-0529, Sandia, March 2002.

[6] C. Picard, M. Garbey, and V. Subramanian. Mapping LSE on a grid: Software architeture and performance gains. In *Proceedings of the International Conference on Parallel Computational Fluid Dynamic*, 2005.

[7] C. Soize, editor. *A Least Square Extrapolation Method for the A Posteriori Error Estimate of CFD and Heat Transfer Problem*, 2005.

[8] R. Verfurth. *A Review of A Posteriori Estimation and Adaptive Mesh Refinement Techniques.* Wiley-Teubner, 1996.