SANDIA REPORT

SAND2015-8107
Unlimited Release
Printed ??? 2014

Addressing Model Form Error for Time-Dependent Conservation Equations

Richard G. Hills

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation,
a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's
National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND 2014-????
Unlimited Release
Printed ??? 2014

Addressing Model Form Error for Time-Dependent Conservation Equations

Richard G. Hills
V&V, UQ, Credibility Processes
Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87195-0828

Abstract

Model form error of the type considered here is error due to an approximate or incorrect representation of physics by a computational model. Typical approaches to adjust a model based on observed differences between experiment and prediction are to calibrate the model parameters utilizing the observed discrepancies and to develop parameterized additive corrections to the model output. These approaches are generally not suitable if significant physics is missing from the model and the desired quantities of interest for an application are different than those used for calibration. The approach developed here is to build a corrected surrogate solver through a multi-step process: 1) Sampled simulation results are used to develop a surrogate computational solver that maintains the overall conservative principles of the unmodified governing equations, 2) the surrogate solver is applied to candidate linear and non-linear corrector terms to develop corrections that are consistent with the original conservative principles, 3) constant multipliers on the these terms are calibrated using the experimental observations, and 4) the resulting surrogate solver is used to predict application response for the quantity of interest. This approach and several other calibration-based approaches were applied to an example problem based on the diffusive Burgers' equation. While all the approaches provided some model correction when the measure/calibration quantity was the same as that for an application, only the present approach was able to adequately correct the CompSim results when the prediction quantity was different from the calibration quantity.

CONTENTS

FIGURES

TABLE

EXECUTIVE SUMMARY

Validation experiments are used to evaluate whether computational models can represent the behavior of the physics of the experiments. Differences between experimental measurements and the corresponding CompSim predictions are expected due to uncertainty in both in the measurements and in the model predictions. Differences with magnitudes or coherent trends that are larger than that expected due to the experimental or prediction uncertainty, suggests that the CompSim contains model form error of the type associated with an incomplete or incorrect representation of the physics of the experiments.

Several approaches have been used to correct the CompSim model given such experimental evidence of model form error. The ideal approach is to develop a better understanding of the missing or incompletely represented physics, and to use this understanding to improve the physics model used by the CompSim. This approach may not be practical for the intended use due to excessive computational resources, and time and personnel constraints associated with code revision. Other but less ideal approaches to compensating for model form error are to use subsystem or system experimental data 1) to modified and calibrated constitutive models; 2) to develop and calibrate regression based model error terms that attempt to represent model form error over the conditions of the experimental observations and the application; or 3) to simply calibrate model parameters, such as those that appear in the existing constitutive models. Note that once this data has been used for these calibration-based approaches, the data can no longer be considered independent validation data and care must be used in interpreting the validity of the resulting 'calibrated' CompSim model.

The work presented here focuses on model form error associated with physics represented by conservation-based transient partial differential equations (e.g. conservation of mass, momentum, energy), with a specific focus on flux terms associated with the conserved quantities. Because of this focus, one can incorporate the requirements of conservation in the development of a model correction. Note that for this class of problems, model form errors are often due to miss-representation of the constitutive relations (i.e. effective conductivity through a high temperature porous media), or miss-representation of the driving potentials that lead to transport of the conserved quantities. The approach presented here utilizes sampling of the non-corrected CompSim model to develop approximate linear divergence operators that are then used to operate on parameterized correction terms (linear or non-linear) supplied by the analyst. This approach is similar to the three approaches listed in the previous paragraph in that this approach uses subsystem or system level data to calibrate the CompSim model. This approach attempts to improve the physics representation of the flux terms, rather than to simply recalibrating the parameters in the existing flux terms or utilizing non-physics based model correction terms, both of which may be less dependable when extrapolation away from the conditions of the experimental data. The approach does require that analysts have a sufficient understanding of the physics to postulate why model form error is present (i.e. postulate that there may be a non-linearity in the advection term, or a dependence of a constitutive relationship on a dependent variable).

The methodology is applied to an application that is correctly represented by the diffusive Burgers' equation, but modeled by the linear convective-diffusive equation. This example is challenging in that the incorrectly modeled physics represents a dominant transport mechanism, and as such, cannot be adequately corrected by methods that simply calibrate the existing model parameters, or develop additive calibrated model correction terms based on measurements of quantities other than those being predicted. In fact, experience by the present author in addressing model form error for this example served as direct motivation for the present approach. The application to the present problem does require that the analyst propose the general form of the correction. For the present case, the proposed form corresponded to a simple non-linear dependence of flux on the quantity being conserved. The results of the approach was compared to those using simple calibration of the convective-diffusion model parameters, and to the more general Partial Least Squares regression approach recently developed by the present author (Hills, 2013). The present approach was found to superior to these other calibration approaches, but did require some insight from the analyst to define the form of the correction.

1. INTRODUCTION

1.1. Scope

The present work utilizes observed transient experimental data from systems represented by conservation-based partial differential equations (PDEs) to compensation for model form error in the flux terms. Note that the potentials driving flux and the constitutive models associated with the flux terms are often the source of model form error for such systems. The approach, as demonstrated here, is non-invasive in the sense that sampled results from a computational model are used to develop a grid-based surrogate PDE solver in the form of one or more matrix operators. The surrogate is used with experimental data to develop candidate model form corrections that accommodate non-linear dependencies of flux on the dependent variables. However, because this approach represents an approximate correction for model form error, the approach should be consider as a temporary measure until the computational code can be formally revised to incorporate these non-linear dependencies into the computational physics model. The approach does provide a mechanism to test various nonlinear corrections to a computation model, which can provide insight into the adequacy of potential computation model revisions.

The surrogate solver can be used to ‘correct’ the computation model with two levels of approximation. The first, the one demonstrated here, is to develop a linear surrogate operator for the divergence of flux, use the surrogate operator to develop a source term that can include non-linear dependencies, utilize the experimental data to calibrate the parameters introduced by the correction process, and apply the corrected surrogate solver to the application without re-solving the original governing equations.

A second approach is to develop the surrogate operator for the divergence of the flux term; use the operator to develop a source term that is added to the original PDE solver; calibrate the parameters introduced by the source term utilizing the experimental data; and then apply the original algorithm with the addition of the source term correction to the application. We expect that this second approach will be superior to the first (however, we have not tested this) since the surrogate solver is utilized to develop a corrective source term, but the original solver, which may have non-linear features such as flux correction, is used to solve the system of PDEs that now includes the correction source term. This approach does require that the source corrected PDE solver be solved multiple times to support the required calibration to the experimental data.

The first approach is developed and demonstrated here. The method will be applied to a simple non-linear example problem, as a demonstration of concept. The “true physics” for the example is represented by the diffusive Burgers’ equation. The linear convective-diffusive equation will be use to represent the “modeled” behavior. This example was chosen because previous attempts by the author to adequately address model form error using other model correction methodology for this application were not fully successful (Hills, 2013) due to the dominant effect of the missing physics associated with the non-linear convection term.

As will be shown, these approaches are conceptually similar to correcting model form error by modifying the constitutive relations to represent nonlinear dependence on the quantities being conserved. The approaches presented are different from those that develop additive error correction terms on the measured quantities, or attempt to correct for model form error by calibration existing model parameters (Higdon et al., 2008, Mahadevan, 2011, Romero, 2006, 2007, 2008) as the present approach attempts to develop a modified solver to better represent the physics as the analyst understands it.

2. APPROACH

2.1. Time-dependent conservative equations

The conservative equations considered here are of the form

$$\frac{\partial u}{\partial t} + \nabla \cdot F(u) = 0 \tag{1}$$

where **u** is the quantity conserved and **F** is the flux of this quantity. Examples include

Linear convective-diffusive equation in 1-D:

$$F(u) = au - k\frac{\partial u}{\partial x} \tag{2}$$

Non-linear diffusive Burgers' equation in 1-D:

$$F(u) = au^2 - k(u)\frac{\partial u}{\partial x} \tag{3}$$

MHD equations:

$$u = \begin{bmatrix} \rho \\ \rho v \\ B \\ e \end{bmatrix};\ F = F(u) \tag{4}$$

Wave equation in 1-d:

$$\frac{\partial^2 y}{\partial t^2} = a^2 \frac{\partial^2 y}{\partial x^2} \tag{5}$$

let $p = \frac{\partial y}{\partial t}$, $q = a\frac{\partial y}{\partial x}$, $u = \begin{bmatrix} p \\ q \end{bmatrix}$

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\begin{bmatrix} 0 & -a \\ -a & 0 \end{bmatrix} u \tag{6}$$

2.2. Addressing model form error in flux

The flux **F** of Eq. (1) is generally a function of **u**, its spatial derivatives, constitutive relationships (which may also be a function of **u**), and the corresponding model parameters. Differences between experimental results and the corresponding model predictions, as observed from validation experiments, may be due to miss-specification of the model parameters (can be addressed through calibration), model form error in the constitutive relationships, or model form error in the non-constitutive components (e.g. spatial derivatives, or operation on non-linear functions of u) of the model for flux. For example, consider the case of Eqs. (2) and (3). Utilizing Eq. (2) to represent an application that should be modeled by Eq. (3) results in model form error in the convective component of flux (should be u^2 rather than u), and in the u dependence of the constitutive relationship $k(u)$. Note that Eq. (2) can be used to represent Eq. (3) if we transform the quantity operated on as follows:

$$F(u) = au^2 - \frac{\partial g(u)}{\partial x} \tag{7}$$

where $g(u)$ is chosen to satisfy

$$\frac{\partial g}{\partial u} = k(u) \tag{8}$$

For example, if the thermal conductivity possess a linear temperature dependence:

$$k(u) = k_0 + k_1 u \tag{9}$$

where k_0 and k_1 are constant, then

$$g(u) = k_0 u + k_1 u^2/2 \tag{10}$$

We see that Eq. (3) follows from Eq. (2) if we operate on au^2 for the convective term and on $g(u)$ for the diffusive term. Note that Eq. (9) represents a constitutive relationship where-as Eqs. (7) and (10) represents an operation on a non-linear function of u. The use of either Eq. (9) or (10) will give equivalent results. Here we develop a linear approximation to the divergence operator, and apply the approximate operator to non-linear functions of the dependent variables. We see from Eq. (8) that this is equivalent to adding an u dependence to the constitutive model.

2.3. The matrix operators

Finite difference/volume/element methods for solving Eq. (1) can be explicit, implicit, or mixed implicit/explicit. Implicit schemes require the solution of linear or non-linear algebraic equations. A common class of solvers for non-linear problems is those that solve a linearized problem over one time step. Such one-step implicit and explicit solvers can posses the following forms:

$$u^{k+1} = Au^k \tag{11}$$

$$Bu^{k+1} = u^k, \text{ so } u^{k+1} = B^{-1}u^k \tag{12}$$

$$Eu^{k+1} = Fu^k, \text{ so } u^{k+1} = E^{-1}Fu^k \tag{13}$$

Other solvers, such as flux-corrected solvers, cannot be written in these forms explicitly, but, as will be shown in the example problem, can sometimes be approximated by solvers of the above forms. Note that the coefficient matrices can be updated each time step if they show a nonlinear dependence on the dependent variables. In general, Eq. (11), (12) or (13) can be written as

$$u^{k+1} - u^k = (A - I)u^k = Hu^k \tag{14}$$

where **H** is a sparse matrix for explicit schemes, or a non-sparse matrix for implicit schemes. However, **H** for an implicit scheme may be well approximated by a somewhat sparse matrix for small time increments as **u** at some location is most heavily influenced by behavior in a neighborhood of this location.

For the present application, we wish to use the conservative principles imbedded in **H** for the specific mesh to develop a correction term. Because **H** is generally not accessible, we will use observations (samples) of the evolution of a solution on the mesh to develop an equivalent linearized approximation of **H**. Specifically, we estimate **H** using observations of u^k from the computational algorithm with source terms set to zero so that we can sample the flux terms. As will be shown below, the process for estimating **H** can be accomplished row by row (each row corresponding to a nodal location). The process can be made even more efficient by defining a neighborhood of influence for a time step, and solving for the best estimate of only those elements in a row of **H** that correspond to nodes in this neighbor. For the case of explicit schemes, the effective neighbor is small. For the case of implicit schemes, the effective neighborhood for which the values of **u** have a significant impact is larger.

For the case of a flux possessing both diffusive and convective components, one can separate **H** into symmetric and skew-symmetric parts. Note that the diffusive operator is symmetric and the convective operator is skew-symmetric, and that the meshed based solution approximates these operators in some sense. This separation is useful when the non-linear correction to the convective term (i.e. u to u^2 in the previous discussion) is different than the non-linear correction to the diffusive term (i.e., u to $g(u)$ in the previous discussion). For example, the matrix operators used to approximately solve Eq. (2) can be written as

$$u^{k+1} = u^k + Cu^k + Du^k \tag{15}$$

where **u** is a vector of mesh values, k represents the time step, and **C** and **D** represent the equivalent convective and diffusive terms. **C** is skew-symmetric and **D** is symmetric. Utilizing the **C** and **D** obtained from the linear problem, Eq. (2), the solution to the non-linear problem, Eq. (3), can be approximately solved by

$$u^{k+1} = u^k + Cv^k + Dg(u^k) \tag{16}$$

where the components in $\mathbf{v}^k$ are the square of the components of $\mathbf{u}^k$. Application of this approach to solve the non-linear Eq. (3), utilizing the linear-equation solver corresponding to Eq. (2), will be illustrated in the following chapter. We will also illustrate the 'calibration' of this type of model form error, given experimental observations of transient behavior at a few measurement locations.

2.4. Procedure

The basic procedure used here consists of the following steps:

1. Sample the computation model to build the linearized matrix operator for the divergence of the flux. If both diffusion and convection is present, it can be advantageous to separate the matrix operator into its symmetric and skew-symmetric components to approximately represent the diffusive and convective components separately. If the basic computational model possess source terms, set these to zero during the sampling process so that only the effect of the transient and divergence terms are sampled.
2. Use the conservation principles inherent in the approximate matrix operator for the divergence term to operate on parameterized non-linear functions of u to develop the correction term. The choice of these non-linear functions requires judgment.
3. Estimate the correction term parameters and their uncertainty to best match the experimental response data in some optimum sense. Non-linear least squares regression is used here.
4. Apply corrected solver to the QoI for the application and estimate the uncertainty in the resulting QoI due to correction uncertainty (step 3) and due to model parameter uncertainty and other forms of model uncertainty (only parameter uncertainty is considered here).

2.4.1. Step 1: Building an approximate matrix operator

The first step is to estimate **H** in Eq. (14). Note that if there is a source term in the original system of equations, this source term should be set to zero so that the sampling only samples the effect of the time derivative and the divergence terms. Here we estimate an effective **H** for the divergence term that is constant over time.

$$u^{k+1} - u^k = Hu^k \tag{14}$$

Note that the time step size has been absorbed into **H**. If one is utilizing samples over variable time step size, it would be appropriate to not absorb the time step into **H** and to leave this step size in the denominator of the right hand side of Eq. (14). We will use the estimate of **H** to develop a model correction that generalizes to

$$u^{k+1} - u^k = H\,f(u^k) \tag{17}$$

where **H** is invariant with time. Note that differential operators by themselves are constant with time (the gradient or divergence operators), but the quantities operated on are **u** dependent, and hence time and spatially dependent. Hence, our approach is to estimate the grid based approximation of the operator, and then use this approximate operator on linear and non-linear functions of **u** to represent the non-linearity of the system.

In concept, the sampling appears straightforward for the case of **H** invariant with time. One can sample the nodal values of **u** across the mesh as a function of time, and then use these **u** in Eq. (14) to estimate an equivalent **H**. Since we are approximating **H** as invariant over time, we can solve for **H** row by row as follows. For spatial node *i* and time *k*, Eq. (14) can be written as

$$u_i^{k+1} - u_i^k = h_i u^k \tag{18}$$

Since **H** is approximated as invariant with time, we can write

$$\begin{matrix} u_i^2 - u_i^1 \\ u_i^3 - u_i^2 \\ \vdots \\ u_i^m - u_i^{m-1} \end{matrix} = \begin{bmatrix} u^{1T} \\ u^{2T} \\ \vdots \\ u^{m-1T} \end{bmatrix} h_i^T \tag{19}$$

Note that Eq. (19) represents a matrix equation with the values of **u** known from sampling the transient behavior of **u** over the mesh, with the transpose of the i^{th} row of **H**, h_i, as the unknown vector. Evaluation of Eq. (19), node by node, leads to the full matrix **H**.

Multiple attempts were made to estimate **H** utilizing the numerical solutions of the equations for the example problem provide in the following chapter. We quickly realized that the sampling of these solutions do not insure that the different physics is exercised over the full domain. For example, if a front (see Figure 1) does not pass through the full domain, then the *u* for those regions for which the front does not reach does not contain sufficient information to obtain a good estimate of the corresponding rows of **H**. A method that was found to activate all regions to the physics was to randomly generate a nodal set of **u**, use this set as the initial conditions, march through a limited number of time-steps, collecting the nodal values of **u**. Note that due the random variation of the initial *u* over the mesh, strong diffusive behavior will be present at early times throughout the mesh, with the diffusive behavior becoming less dominant at later times. This process was repeated using different random generations of the initial conditions to obtain sets of results. In this case, Eq. (19) can be written as

$$\text{Set 1:} \quad \begin{matrix} u_i^2 - u_i^1 \\ u_i^3 - u_i^2 \\ \vdots \\ u_i^m - u_i^{m-1} \end{matrix} = \begin{bmatrix} u^{1T} \\ u^{2T} \\ \vdots \\ u^{m-1T} \end{bmatrix} h_i^T$$

Set 2: $$\begin{bmatrix} u_i^2 - u_i^1 \\ u_i^3 - u_i^2 \\ \vdots \\ u_i^m - u_i^{m-1} \end{bmatrix} = \begin{bmatrix} u^{1T} \\ u^{2T} \\ \vdots \\ u^{m-1T} \end{bmatrix} h_i^T \tag{20}$$

Set 3: $$\begin{bmatrix} u_i^2 - u_i^1 \\ u_i^3 - u_i^2 \\ \vdots \\ u_i^m - u_i^{m-1} \end{bmatrix} = \begin{bmatrix} u^{1T} \\ u^{2T} \\ \vdots \\ u^{m-1T} \end{bmatrix} h_i^T$$

etc.

Note that $\mathbf{h}_i$ is assumed invariant over all of these sets. Eq. (20) represents $k \cdot (m-1)$ equations with n unknowns where k is the number of sets, and m is the number of time steps.

A second issue that arose was the matrices expected for simple problems and algorithms, such as a finite difference approximation of the convective diffusion equation, were not consistent were with those found from the above procedure. An approach that resulted in more consistent estimates was to base the estimates on differences in the u from neighboring values. Consider time step k+1 for node i.

$$u_i^{k+1} - u_i^k = h_i u^k \tag{21}$$

Now consider the following series of algebraic manipulations.

$$u_i^{k+1} - u_i^k = h_i \left(u^k - u_i^k O\right) + h_i \left(u_i^k O\right) \tag{22}$$

where **O** is a n by 1 column vector of ones. Since u_i^k is a scalar, we can write

$$u_i^{k+1} - u_i^k = h_i \left(u^k - u_i^k O\right) + u_i^k \sum_{j=1}^{n} h_{i,j} \tag{23}$$

where $h_{i,j}$ is the j^{th} element of row i of **H**. For applications that spatial differences in u drive fluxes (typically of our problems), we should expect that for the case of a uniform $u = u_c$ across the mesh, there should be no change in u over a time step. For this case, Eq. (23) gives

$$u_c^{k+1} - u_c^k = 0 = h_i \left(u_c^k O - u_c^k O\right) + u_c^k \sum_{j=1}^{n} h_{i,j} \tag{24}$$

So the condition of conservation requires (note that we have excluded source terms from the present formulation)

$$\sum_{j=1}^{n} h_{i,j} = 0 \tag{25}$$

Using (25) in (23) gives

$$u^{k+1}_{i} - u^{k}_{i} = h_i \left(u^k - u^k_i O\right) \tag{26}$$

We found that by using this added piece of information, Eq. (25), resulted in a more robust estimate of **h**. But this condition also reduced the degrees of freedom of the system of equations by one, resulting in a rank deficient problem (rank deficient by one). We see this effect by recognizing that node i of vector $\mathbf{h}_i$ cannot be determined from Eq. (26) because

$$u^k - u^k_i O \tag{27}$$

will always contain a zero for the i^{th} element of u^k. The solution for rank deficient systems is not unique. Some matrix solver algorithms address rank deficient problems by setting one of the elements in $\mathbf{h}_i$ to zero. Others, such as those based on singular value decomposition, result in a different estimate of $\mathbf{h}_i$ (i.e. the minimum length solution) with the i^{th} element of $\mathbf{h}_i$ non-zero. To eliminate the non-uniqueness issues associated with rank deficient problem, we simply remove the i^{th} element of $\mathbf{h}_i$ from the solution process, solve for the remaining elements of $\mathbf{h}_i$, and then use Eq. (25) to solve for the i^{th} element of $\mathbf{h}_i$. Reformulating the problem in-terms of these modifications can be summarized as follows:

Define

$$\Delta v^k_i = \begin{bmatrix} u^k_1 - u^k_i \\ \vdots \\ u^{\ k}_{i-1} - u^k_i \\ u^{\ k}_{i+1} - u^k_i \\ \vdots \\ u^k_n - u^k_i \end{bmatrix};\ g_i^{\ T} = \begin{bmatrix} h_{i,1} \\ \vdots \\ h_{i,i-1} \\ h_{i,i+1} \\ \vdots \\ h_{i,n} \end{bmatrix} \tag{28a, b}$$

The system of equations becomes

$$\text{Set 1: } \begin{matrix} u^2_i - u^1_i \\ u^3_i - u^2_i \\ \vdots \\ u^m_i - u^{m-1}_{\ i} \end{matrix} = \begin{bmatrix} \Delta v^{1T}_i \\ \Delta v^{2T}_i \\ \vdots \\ \Delta v^{m-1T}_{\ i} \end{bmatrix} g_i^{\ T}$$

$$\text{Set 2: } \begin{matrix} u_i^2 - u_i^1 \\ u_i^3 - u_i^2 \\ \vdots \\ u_i^m - u_i^{m-1} \end{matrix} = \begin{bmatrix} \Delta v_i^{1T} \\ \Delta v_i^{2T} \\ \vdots \\ \Delta v_i^{m-1T} \end{bmatrix} g_i^T \tag{29}$$

$$\text{Set 3: } \begin{matrix} u_i^2 - u_i^1 \\ u_i^3 - u_i^2 \\ \vdots \\ u_i^m - u_i^{m-1} \end{matrix} = \begin{bmatrix} \Delta v_i^{1T} \\ \Delta v_i^{2T} \\ \vdots \\ \Delta v_i^{m-1T} \end{bmatrix} g_i^T$$

etc.

Eq. (29) is not rank deficient (rank equal to the number of unknowns) and can be solved using normal least squares procedures. For example, the least squares solution of

$$Ax = b \tag{30}$$

is

$$x = \left(A^T A\right)^{-1} A^T b \tag{31}$$

Given a solution for g_i, the corresponding elements of h_i follow directly from Eq. (28b). The remaining element of h_i follows from Eq. (25).

$$h_{i,i} = -\sum_{\substack{j=1 \\ j \neq i}}^{n} h_{i,j} \tag{32}$$

The process is repeated for each h_i (i.e. each row of **H**) leading to the desired estimate of **H** for use in Eqs. (14), (17). To emphasize that this is a linearized approximation to the original **H**, we denote the approximate **H** by $\tilde{H}$. Equation (14) becomes

$$u^{k+1} - u^k \approx \tilde{H} u^k \tag{33}$$

One other refinement to estimating **H** is to define Eqs. (28a, b) only in terms of some neighborhood of the i^{th} node. Note that discretized approximations to flux operators depend most heavily on nodes in a neighborhood of the node for which conservation is being applied, with the size of this neighborhood larger for implicit schemes than for explicit schemes. The use of a neighborhood will be demonstrated for the example presented in the following chapter.

2.4.2. Step 2: Specify a Correction

In concept, there are several approaches that can be taken to develop the mathematical form of the correction. Consider the generalized discretized version of a system of equations.

$$u^{k+1} - u^k = H(u^k,u^{k+1}) + S(u^k,u^{k+1}) \tag{34}$$

The u^{k+1} on the RHS of Eq. (34) indicates that the discretized system may have implicit and explicit components. $H(u^k,u^{k+1})$ represents some operation (matrix or non-linear operation) on the nodal values **u** and $S(u^k,u^{k+1})$ are source terms. By sampling output from Eq. (34) with the source term **S** set to zero, one can develop an approximate matrix operator $\tilde{H}$ that operates on u^k to represent a correction to the divergence of the flux term. Once $\tilde{H}$ is estimated, this source term is re-activated and a correction is added to Eq. (34) as follows:

$$u^{k+1} - u^k = H(u^k,u^{k+1}) + S(u^k,u^{k+1}) + \tilde{H}\, f(u^k,\gamma) \tag{35}$$

where $f(u^k,\gamma)$ is some parameterized non-linear function of u^k specified by the modeler. The values for the parameters γ are estimated using observed response data from experiment (i.e. data that provides evidence of model form error), and non-linear parameter estimation. This choice of $f(u^k,\gamma)$ requires judgment as is illustrated in the following section. A second approach also uses samples to develop $\tilde{H}$, but then uses $\tilde{H}$ to approximately solve the system of equations and to develop the correction based on this surrogate operator. In this case, the revised system of equations becomes

$$u^{k+1} - u^k = \tilde{H}\, u^k + S(u^k,u^{k+1}) + \tilde{H}\, f(u^k,\gamma) \tag{36}$$

Due to the process used to estimate $\tilde{H}$, Eq. (36) is appropriate even if the original system of equations utilized an implicit scheme. In this case $\tilde{H}$ will be a full matrix rather than a sparse matrix.

The disadvantage of the second approach, Eq. (36), is that the approach cannot fully capture features of non-matrix type numerical operations, such as flux correction. The advantage of this second approach is that the parameter estimation process may be more computational efficient. We have not applied the first approach, and as such, do not know the issues that may arise with its application.

2.4.3. Step 3: Estimate the Parameters in the Correction Term and their Covariance

Standard non-linear parameter estimation techniques can be used to estimate the parameters γ in the correction term. One can also simultaneous estimate the model parameters that appeared in the original equation (if using Eq. (35)), but this does add complexity to the process.

Two sources of uncertainty are considered here. The first contribution is due to the uncertainty in the parameters in the correction term due to the estimation process and will be characterized by their covariance matrix. The second contribution is due the uncertainty in the model parameters used for the CompSim model in prediction of the application conditions and is addressed in the following section. The methodology used to estimate of the covariance matrix for the correction parameters depends on the estimation methodology. Here we use a gradient-based non-linear least squares algorithm (the MatLab (2010) routine *lsqnonlin*), which provides a Jacobian matrix **J** and the residuals during at the completion of the iterative estimation process. This information can be used to estimate the covariance for the correction parameters as follows (Beck and Arnold, 1977):

$$V_\gamma = \sigma_r^2 (J^T J)^{-1} \tag{37}$$

where σ_r^2 is the estimate of the variance of the residuals for the regressed fit of the corrected model predictions to the data.

2.4.4. Step 4: Evaluate the QoI for the Application Conditions and its Uncertainty

The corrected model can now be applied to the application conditions. Here we use sampling to evaluate the associated uncertainty where the samples are based on the mean or nominal values and the covariance matrices for each of these parameters, as demonstrated in the following chapter. Note that in the case of Eq. (36), the linearized model is used to represent the solution, and the CPU expense of sampling output from this linearized solver is generally not significant.

3. EXAMPLE: BURGERS' EQUATION

3.1. Specification of the Problem

To illustrate this approach, consider the following equation and the initial and boundary conditions:

$$\frac{\partial u}{\partial t} + c\frac{\partial u^p}{\partial x} = d\frac{\partial^2 u}{\partial x^2} \tag{38}$$

$$u(x,0) = \begin{cases} 0.5; & x < 1 \\ x - 0.5; & 1 \le x < 2 \\ 1.5; & 2 \le x < 3 \\ 4.5 - x; & 3 \le x < 4 \\ 0.5; & 4 \le x \end{cases} \tag{39}$$

$$u(0) = u(20) \tag{40}$$

Note that periodic boundary conditions are used for u. When $d \neq 0$, $c = 0$, Eq. (38) represents the diffusion equation; $d = 0$, $c \neq 0$, $p = 1$, the convective equation; $d \neq 0$, $c \neq 0$, $p = 1$, the linear convective diffusion equation; $d = 0$, $c \neq 0$, $p = 2$, Burgers' equation, and $d \neq 0$, $c \neq 0$, $p = 2$, the diffusive Burgers' equation. For the present example, the true physics of the target application is correctly represented by the diffusive Burgers' equation. The CompSim model used for illustration is the linear convective diffusive equation and thus contains model form error. The model parameter values used for the example problem are specified in Table 1. This problem is challenging because the non-linear term ($p = 2$) dominates for d << c and its effect is not captured by a linear convective term ($p = 1$). The experimental measurements are time history observations (break-through curves) of $u(x_{meas}, t)$ for independent experiments at two different measurement locations x_{meas}. The desired Quantities of Interest (QoI) for the application conditions are u and flux at a third location as a function of time. The measurement and application locations are

$$x_{meas} = 4, 6;\ \ x_{appl} = 8 \tag{41}$$

The flux is given by

$$flux = cu^p - d\frac{\partial u}{\partial x} \tag{42}$$

The uncertainties in the model parameters, as characterized by standard deviations, are listed in Table 1. The phrase 'truth model' indicates that we have the correct physics represented by the model, but does not imply that we know the true values for the parameters in the model. The parameter p is considered a model specification parameter and takes on the values 1 or 2 with no uncertainty. Because this parameter does not appear in the convective-diffusion equation, there would be no reason to consider the variability in this parameter.

Table 1. Nominal Parameter Values and Uncertainties for Model and Experiment

d	σ_d	c	σ_c	Model p	Experiment p	σ_{meas}
0.05	0.05d	0.8	0.05c	$p = 1$	$p = 2$	0.02

All computations were performed using operator splitting, with a super-bee flux corrected algorithm for the convective term and an implicit scheme for the diffusive term. (see Hills et. al., 1994, for a version of the algorithm that uses a different flux limiter).

The experimental data was simulated by randomly generating values for *d* and *c* for use in the true physic model for the experiments, and adding uncorrelated measurements noise. The uncertainty in the parameters and the measurements were characterized as normally distributed with the standard deviations tabulated in Table 1.

Figure 1 illustrates the simulated experimental data and the impact of the non-linearity of the diffusive Burgers' equation. The model results shown are for the nominal values of the parameters listed in Table 1. Clearly the use of the nominal values for *d* and *c* in the convective-diffusive model provides very poor agreement between experiment and prediction. The results show a serious under-prediction of the front velocity, resulting in an arrival time significantly longer than that of the data.

3.2. Simple Calibration

Figure 2 illustrates the results from the convective-diffusive model using the calibrated values of the parameters. The calibrated values were obtained utilizing non-linear least squares estimate (the MatLab (2010) routine *lsqnonlin*), the CompSim model, and the experimented data. The parameters were estimated to be

$$d = 0.229; \quad c = 1.395 \tag{43}$$

Note that the calibrated values are approximately 72 and 15 standard deviations away from the nominal values listed in Table 1. This indicates that there is significant model form error and that the calibration is basically a force fit of an incorrect model to the data.

The top two plots of Figure 2 indicates that calibrated model does match the data significantly better than the un-calibrated model (Figure 1), and also provides superior predictions of *u* at the application location (lower left plot). Even so, the calibrated model cannot represent the steep front induced by the non-linearity of Burgers' equation and does a very poor job at predicting flux (lower right plot). This illustrates the dangers of calibrating a bad model to data of one type (*u*) and then predicting a QoI of another type (flux). In this case, the relation of u to flux is simply incorrect and one cannot expect a model calibrated using experimental observations of u to somehow correct for the incorrect dependence of flux on u.

3.3. Partial Least Squares Regression (PLSR)

Figure 3 illustrates the results obtained using the Partial Least Squares Regression (PLSR) methodology developed by Hills (2013) and applied to Burgers' equation. The results are slightly different from those presented by Hills due to the values of the standard deviations used. Hills utilized standard deviations for the application parameters d and c that where twice those listed in Table 1, but the same standard deviations for the experimental conditions. The same number of latent variables (16) was used for the present application as was used by Hills. Note from Figure 3 that the PLSR can capture the data well, but there is some higher frequency oscillation in the fit. Reducing the number of latent variables reduces the higher frequency oscillations, but also introduces lower frequencies oscillations for this example problem. The choice of the number of latent variables to use is ultimately based on judgment, as one must decide between high frequency and low frequency oscillations for this case. The bottom two plots of Figure 2 shows the predicted response for the application location and ± 2 sigma prediction intervals on the predictions using the methodology presented by Hills. Note that the prediction intervals capture some of 'truth' for the prediction of u, but less of 'truth' for the prediction of flux. Comparison of Figure 2 and 3 suggests that the PLSR outperforms the simple calibration approach for both u and flux, but that the results for flux are significantly different from 'truth'. When there is sufficient physics missing from the model (in this case the dominant physics represented by non-linear convection), the projection of observed model validation differences to the application conditions using a regression based on an invalid CompSim model can lead to poor results.

3.4. Application of the Operator Correction Method to Burgers' Equation

3.4.1. Step 1: Building the matrix operator

The first step was to build the approximate matrix $\tilde{H}$ for Eq. (36). Note that the present problem does not contain a source term $S(u^k,u^{k+1})$. The training sets used to estimate $\tilde{H}$ were developed by randomly generating 100 vectors (no correlation between randomly generated elements in a vector or between vectors) to represent the initial u over the nodes. The random values of each node for each of the 100 initial condition vectors were sampled from a uniform distribution, which ranged from 0.5 to 1.5. The computational model (linear convection-diffusion equation) was then used to evaluate and record the nodal values of u for only the first 10 steps. Operator splitting is used for the solution of the convective-diffusive equation with the diffusive component evaluated implicitly and the convective component evaluated explicitly using a flux corrector (Hills et. al., 1994). This implicit nature of the diffusive algorithm results in a full $\tilde{H}$ matrix. However, the values of the elements in $\tilde{H}$ for nodal values far from the neighborhood of node i are small. As a result, we choose to utilize only the nodal values from nodes *i*-5 to *i*+5 in Eqs. (28) to (32) to develop $\tilde{H}$. Specifically, we used only samples from the neighborhood nodes to define the matrix equation, Eq. (28), and then solved for $\tilde{H}$. As a result, $\tilde{H}$ is a banded matrix with a bandwidth of 11. Changing the bandwidth from 11 nodes to a larger number resulting in no perceptible difference in the results, as the elements ±5 nodes away from the diagonal are very small.

3.4.2. Step 2: Specify a Correction

Inspection of the upper right plot of Figure 1 illustrates that the true front moves about twice the speed of the front based on the linear convective-diffusive model. This is reflected by the earlier arrival times for the front. Because the response is not symmetric, the front motion, as indicated by the break-through curves, suggests a non-linear effect is present in the true physics, which is not reflected in the model. The width of the break-through curves for the modeled and measured fronts appear to be similar, suggesting that the model form error is due to a non-linear dependence in the convective term. The development of a non-linear correction to the convective term requires a separation of the convective and diffusive parts of $\tilde{H}$. This separation can be approximately accomplished by recognizing that the diffusive operator is symmetric and the convective operator is skew-symmetric. We thus expect that the matrix equivalents of these operators would also be symmetric and skew-symmetric. The separation of $\tilde{H}$ into symmetric and skew-symmetric components is straightforward:

$$\tilde{H} = C + D \tag{44}$$

where

$$D = \frac{\tilde{H} + \tilde{H}^T}{2} \quad ; \quad C = \tilde{H} - D \tag{45a, b}$$

D is symmetric and **C** is skew-symmetric.

In developing the correction term, we note that since we expect the convective term to contain a non-linear component, we may need to remove the linear approximation that is used for the linear convection-diffusion equation. We use the following form for the corrected solver (see Eq. (36)).

$$u^{k+1} - u^k = \gamma_1 C u^k + \gamma_2 D u^k + \gamma_3 C (u^k)^2 + \gamma_4 D (u^k)^2 \tag{46}$$

where $(u^k)^2$ represents the square of u^k element by element:

$$(u^k)^2 = [u_1^2, u_2^2, \ldots, u_n^2]^{\mathrm{T}} \tag{47}$$

Note that the terms are parameterized by the four parameters γ_1□□□□□□□□□γ_4. We have also added a non-linear term for the diffusion term, which would account for a linear temperature dependence of the diffusion coefficient d on u. For the case of Eq. (46), the flux, Eq. (42), becomes

$$flux = \gamma_1 cu - \gamma_2 d\frac{\partial u}{\partial x} + \gamma_3 cu^2 - \gamma_4 d\frac{\partial u^2}{\partial x} \tag{48}$$

Nodal values of u were used to fit a spline at a given time, which was then used to estimate flux at the measurement locations for this time.

3.4.3. Step 3: Estimate the Parameters for the Revised Model and their Uncertainty

Equation (46) is now used to simulate the experimental data. The MatLab (2010) routine *lsqnonlin* was used to provide a non-linear least squared fit of the surrogate model, Eq. (46), to the experimental date. The resulting estimated values for the four correction parameters were found to be

$$\gamma = [0.2783, \ -0.0047, 0.3904, 0.9322] \tag{49}$$

These results indicate that the linear convective term (the second term) is essentially eliminated and replaced with the non-linear convective term (the fourth term). This result is expected when one considers the form of Burgers' equation that was used to generate the experimental data. The results for the linear and non-linear diffusive terms (first and third term) are more difficult to interpret. One would expected that $\gamma_1 = 1, \gamma_3 = 0$. While the reason for this unexpected result is not clear, we note that a flux corrector was used to evaluate the convective term (i.e. not linear in u) while the linear approximation to the operators does not use flux correction. The combination of linear and non-linear diffusive terms may be a result of this effect. In addition, the data was not generated using the nominal values of the parameters in the diffusive Burgers' equation. We expect that there will be some compensation for this in estimating γ_1, γ_3. The covariance matrix of these parameters is estimated from Eq. (37) using the Jacobean and residuals provide by *lsqnonlin*. The corresponding covariance matrix was estimated to be

$$V_\gamma = 10^{-3}\begin{bmatrix} 0.9131 & 0.6133 & -0.5564 & -0.3239 \\ 0.6133 & 0.6660 & -0.1862 & -0.3507 \\ -0.5564 & -0.1862 & 0.6780 & 0.1066 \\ -0.3239 & -0.3507 & 0.1066 & 0.1898 \end{bmatrix} \tag{50}$$

3.4.4. Step 4: Applied to Application Conditions and Estimation of Uncertainty

The resulting fit of the calibrated solver, Eq. (46), to the data is shown in the upper plots of Figure 4. Overall, the results are superior to those of the previous methods. Note that non-symmetric shape of the experimental measured break-through curves was well approximated, which indicates that much of the non-linear behavior is being captured.

There is a minor inflection in the modeled result for the second measurement location at the left edge of the break-through curve. This inflection is not surprising as this result was obtained using linear operators operating on linear and non-linear functions of u with no flux correction. The resulting prediction of both u and flux are shown in the lower plots of this figure. The results agree quite well for both of these cases, compared to the earlier results. The uncertainty in these predictions due to model and correction parameter uncertainty was found from sampling. The covariance matrix for the model parameters (see Table 1) for the application is

$$V_\alpha = \begin{bmatrix} 0.005^2 & 0 \\ 0 & 0.08^2 \end{bmatrix} \tag{51}$$

The uncertainty in the QoI for the application conditions was estimated by randomly sampling model parameters and correction parameters from multivariate normal distributions with the means given by the nominal and estimated values in Table 1 and Eq. (49), and the covariance matrices given by Eqs. (50) and (51). Given each sampled set of these 6 parameters, the corrected approximated linear solver, Eq. (46), was used to predict u and flux at the application conditions. This process was repeated 1000 times to generate the associated distributions in the predicted application u and flux. The 0.025 and 0.0975 quintiles of the resulting predictions as a function of time are shown in the lower plots of Figure 4. Note that these quintiles do capture the data for both the prediction of u and the prediction of flux.

Overall, the results of the corrected approximate linear solver were found to be superior to the results of simple model parameter calibration and those for Partial Least Squares Regression. The primary reason for this is ‘expert knowledge’ was successfully used to specify the functional form of flux for the corrected solver. A comparison of the results from Figures 1 through 4 is provided in Figure 5.

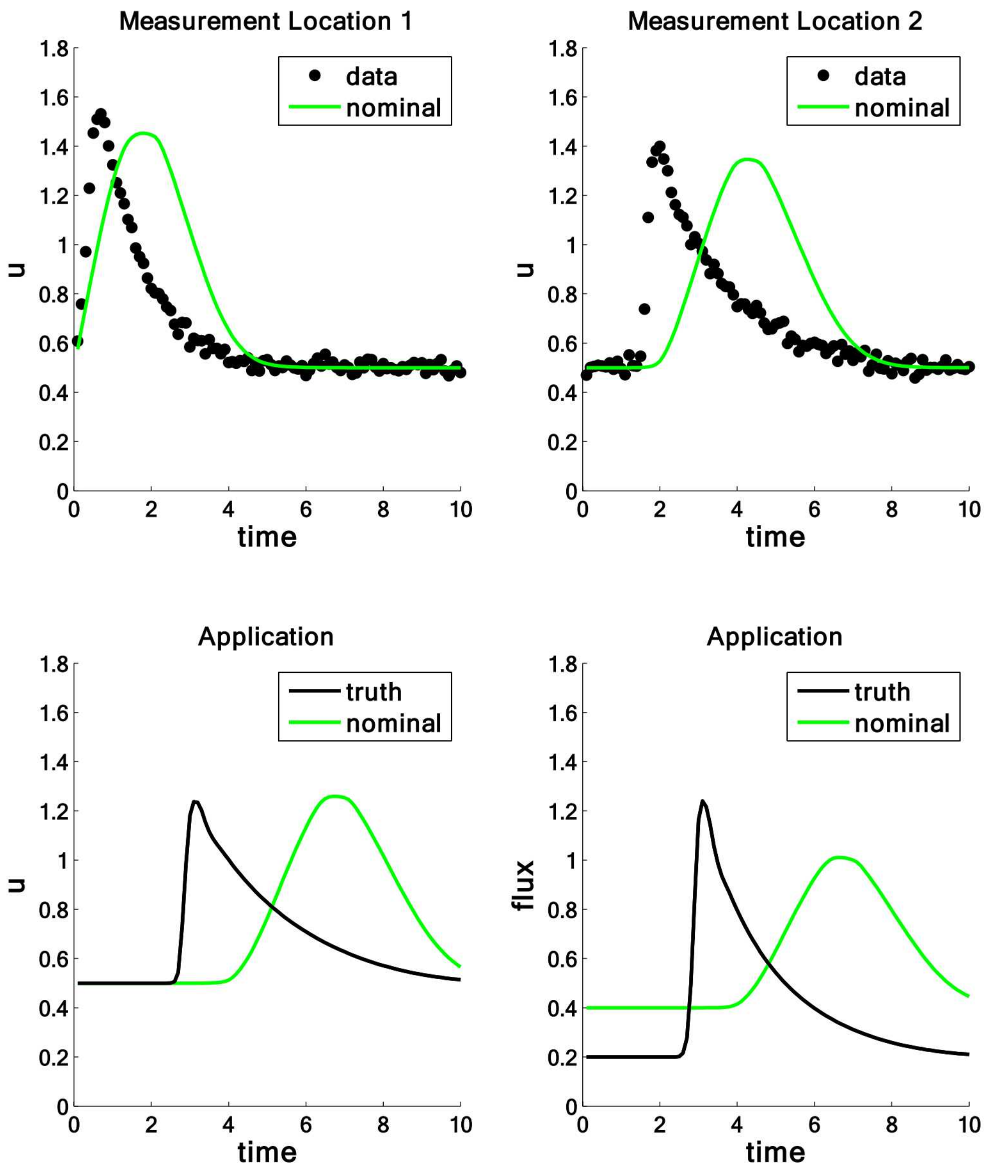

Figure 1. Results using Nominal Parameters

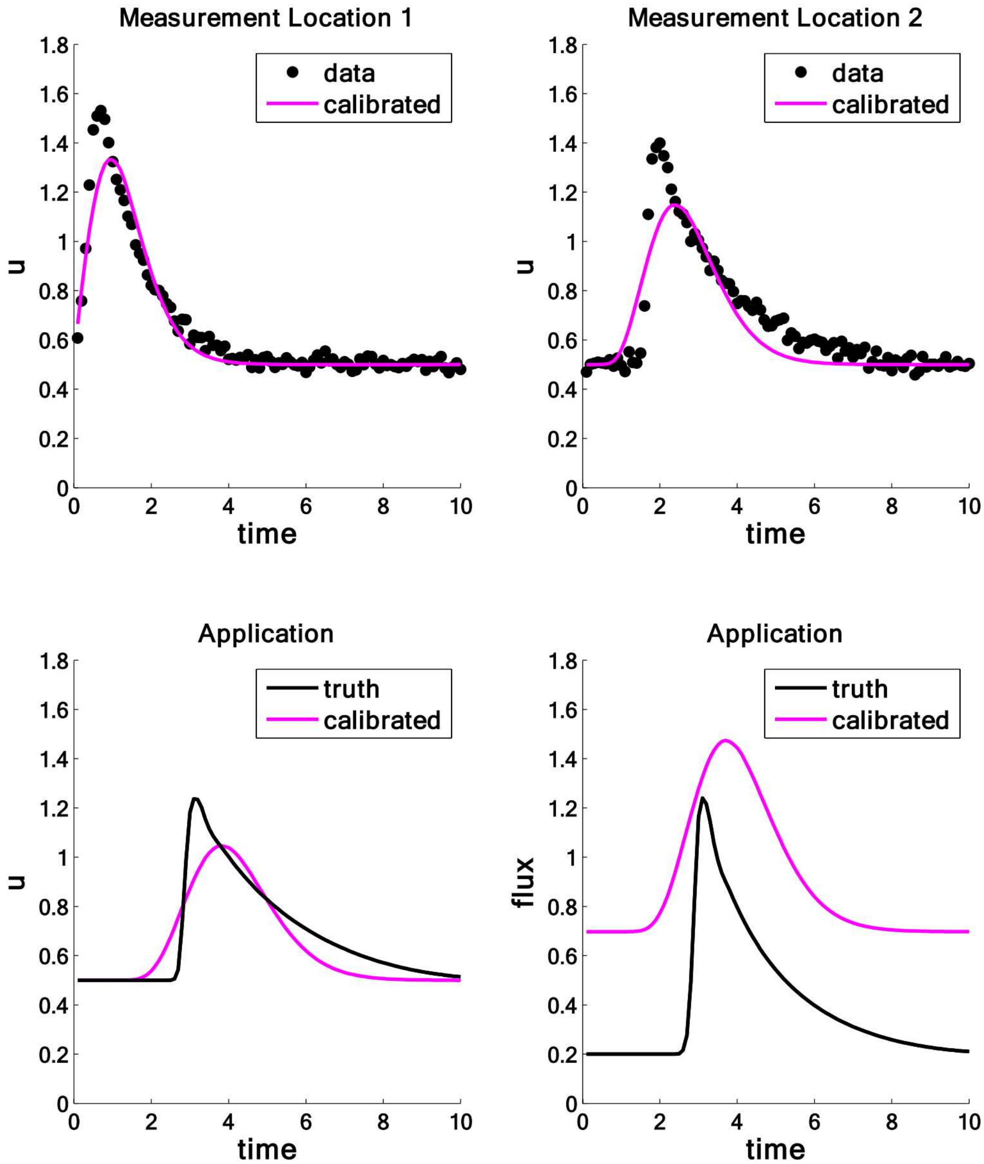

Figure 2. Results using Calibrated Parameters

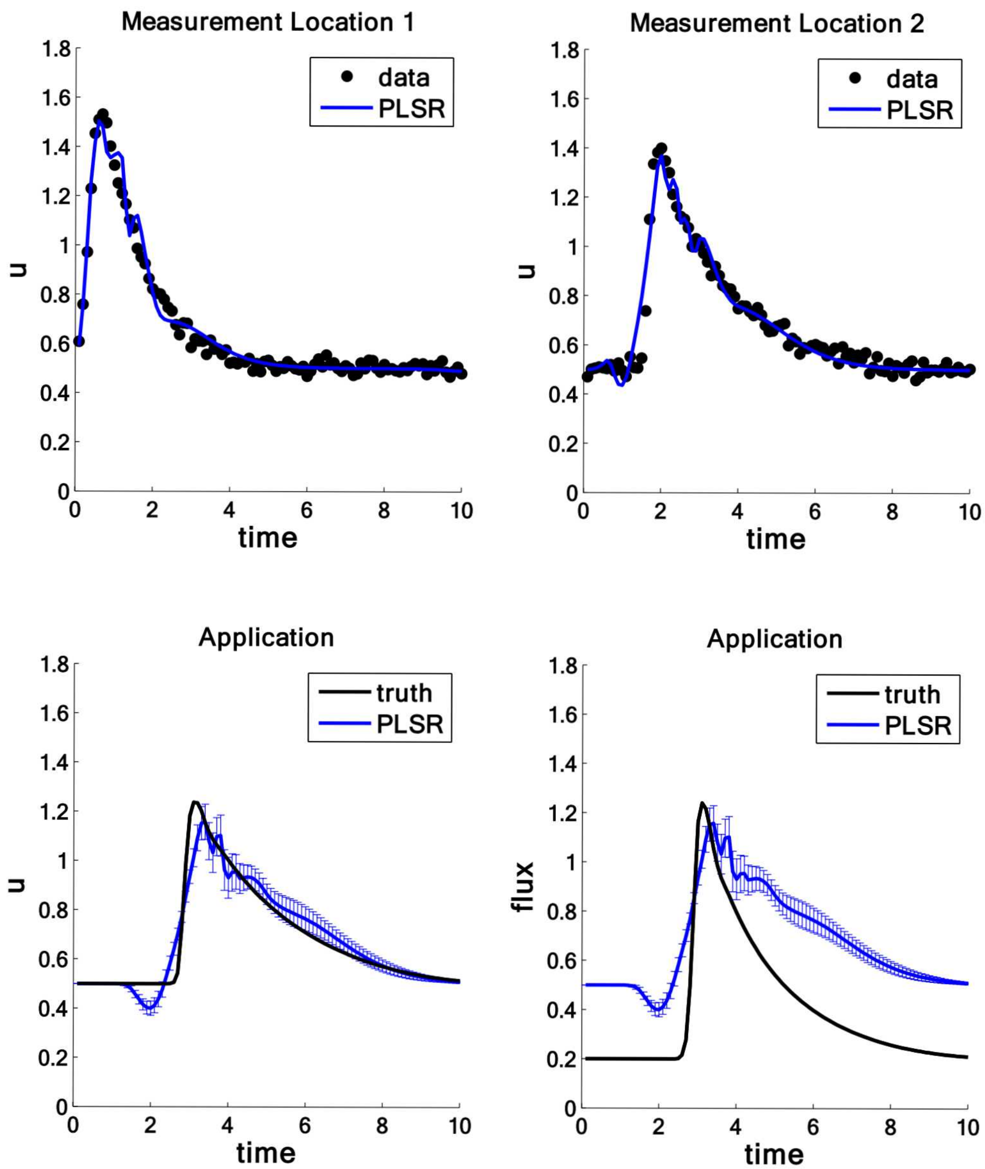

Figure 3. Results using PLSR Correction

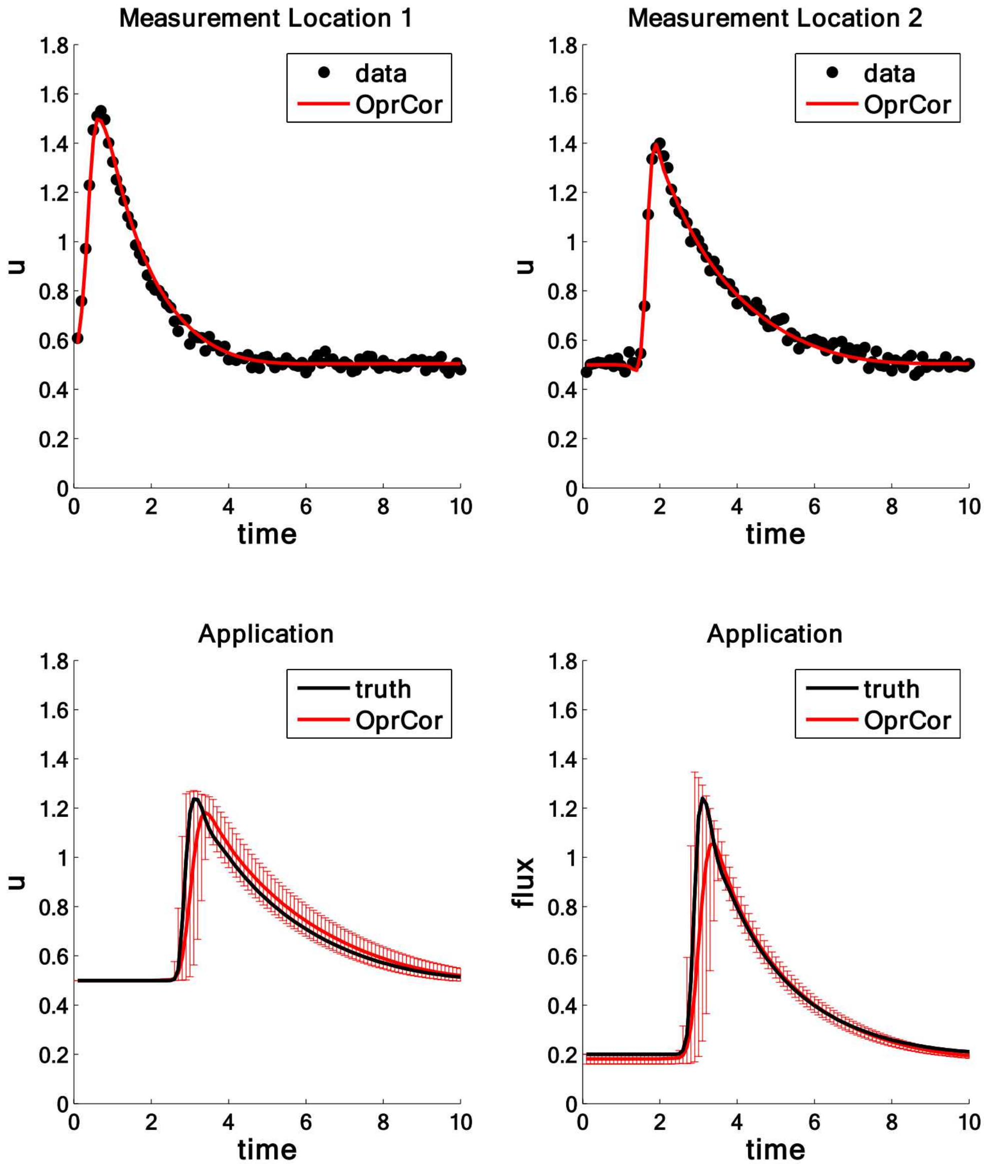

Figure 4. Results using the Corrected Solver (denoted OprCor)

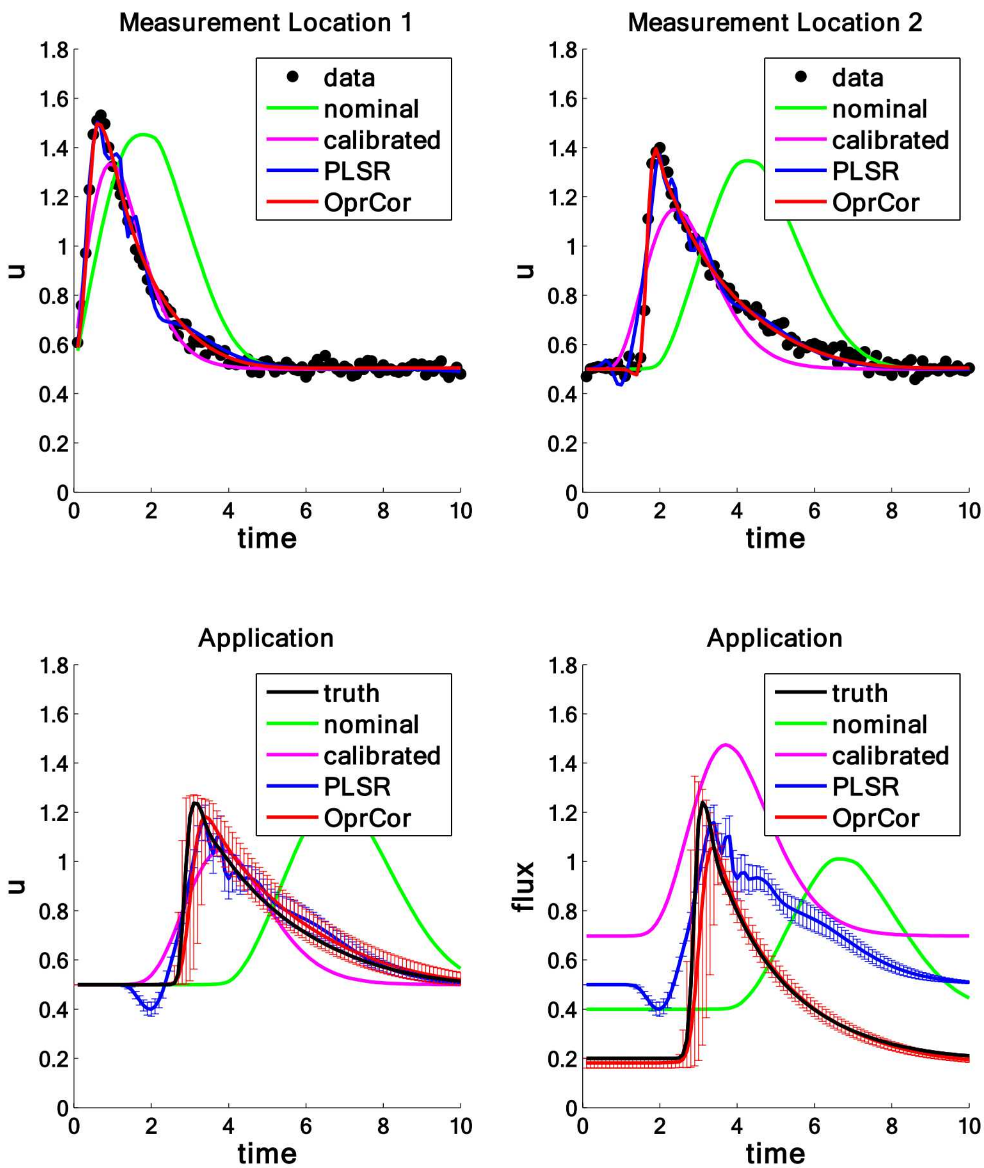

Figure 5. Comparison of All Results

4. DISCUSSION

The comparison between the results (see Figure 5) is rather striking, with the corrected solver method clearly performing best. However, the example problem presented represents a very simplified case in that the first attempt to define a non-linear term was found to be suitable. In principle, the methodology can be expanded to multidimensional and multi-equation problems but one must have a basic concept of why the original model is not performing well. In addition, significant judgment is required to choose the non-linear terms that are operated on by the approximate linear operator. For the present analysis, there was evidence that the front motion was nonlinear in u, leading to the addition of the operation on u^2. For more complex problems, the choice of the non-linear functions of u may be more difficult and somewhat exploratory. Overall, the characterization of the usefulness of this approach awaits application to more complex problems.

Note that two correction schemes were proposed; 1) the addition of a source term on the original solver, and 2) the use of an approximate surrogate solver. Here we demonstrated the second approach. It is not clear what issues may arise with the application of the first approach, but this approach may be more suitable for complex applications. We recommend that one apply this first approach to the present problem, prior to its application to a more complex problem, to gain more insight into the usefulness of the methodology.

Note that the model for the example problem possessed both diffusive and convective term (albeit the wrong convective term). There may be applications where some of these terms are missing from the model. For example, the model may include the convective terms, but not a diffusive term. If a diffusive term is important for the application, but not present in the model, then the sampling approach presented above will not capture the effect of this term. In this case, one will either need to revise the underlying computational model, or develop a source term that represents this diffusive term.

Overall, the methodology developed here brings a conceptually new approach to developing model correction that is based on conservation rather than simple regression. Generalization to more complex applications will clearly require additional development, and the practicality of this approach for such applications remains to be seen.

5. REFERENCES

Beck, J.V. and Arnold, K.J. (1977), Parameter Estimation in Engineering Science, John Wiley & Sons, New York, p 379.

Higdon, D., Nakhleh, C., Gattiker, J., and Ginzburg L. (2008), A Bayesian Calibration Approach to the Thermal Problem, Computer Methods in Applied Mechanics and Engineering, 197: 29-32, pp. 2457-2466.

Hills, R.G., K.A. Fisher, M.R. Kirkland, and P.J. Wierenga (1994), Application of Flux Corrected Transport to the Las Cruces Trench Site, Water Resources Research, 30: 8, pp. 2377-2385.

Hills, R. G. (2013), Roll-up of Validation Results to a Target Application, Sandia Report SAND2013-7424, Sandia National Laboratories.

Mahadevan, S. (2011), Roll-Up of Multi-Level UQ Activities towards System Level QMU, Presented at Sandia National Laboratories, Albuquerque, NM, Oct. 4.

MatLab (2010), MATLAB ® Mathematics, Version 7, R2010b, The MathWorks, Inc., Natick, MA.

Romero V.J. (2006), Issues and Needs in Quantification of Margins and Uncertainty (QMU) for Phenomenologically Complex Coupled Systems, paper AIAA-2006-1989, 8th AIAA Non-Deterministic Analysis Conference, Newport, RI, May 1-4.

Romero, V.J. (2007), Validated Model? Not So Fast. The Need for Model 'Conditioning' as an Essential Addendum to Model Validation, paper AIAA-2007-1953, 9th AIAA Non-Deterministic Methods Conference, April 23 - 26, Honolulu, HA.

Romero, V.J. (2008), Type X and Y Errors and Data & Model Conditioning for Systematic Uncertainty in Model Calibration, Validation, and Extrapolation, SAE paper 2008-01-1368 for Society of Automotive Engineers 2008 World Congress, April 14-17, Detroit, MI.

DISTRIBUTION

Sandia Internal Distribution (electronic)

1	MS0828	R. G. Hills	1544
1	MS0828	W. R. Witkowski	1544
1	MS0845	S. A. Hutchinson	1540
1	MS0110	D. E. Womble	1220
1	MS1318	T. G. Trucano	1440
1	MS0899	Technical Library	9536

Sandia National Laboratories