International Journal for Uncertainty Quantification, 2 (4): 341–361 (2012)

REDUCED ORDER MODELING FOR NONLINEAR MULTI-COMPONENT MODELS

Hany S. Abdel-Khalik, [*] ***Youngsuk Bang, Christopher Kennedy, & Jason Hite***

Department of Nuclear Engineering, North Carolina State University, Raleigh, North Carolina 27695-7909, USA

Original Manuscript Submitted: 5/26/2011; Final Draft Received: 2/7/2012

Reduced order modeling plays an indispensible role in most real-world complex models. A hybrid application of order reduction methods, introduced previously, has been shown to effectively reduce the computational cost required to find a reduced order model with quantifiable bounds on the reduction errors, which is achieved by hybridizing the application of local variational and global sampling methods for order reduction. The method requires the evaluation of first-order derivatives of pseudo-responses with respect to input parameters and the ability to perturb input parameters within their user-specified ranges of variations. The derivatives are employed to find a subspace that captures all possible response variations resulting from all possible parameter variations with quantifiable accuracy. This paper extends the applicability of this methodology to multi-component models. This is achieved by employing a hybrid methodology to enable the transfer of sensitivity information between the various components in an efficient manner precluding the need for a global sensitivity analysis procedure, which is often envisaged to be computationally intractable. Finally, we introduce a new measure of conditioning for the subspace employed for order reduction. Although, the developments are general, they are applied here to smoothly behaving functions only. Extension to non-smooth functions will be addressed in a future article. In addition to introducing these new developments, this manuscript is intended to provide a pedagogical overview of our current developments in the area of reduced order modeling to real-world engineering models.

KEY WORDS: *nonlinear sensitivity analysis, reduced order modeling, subspace methods*

1. INTRODUCTION

Reduced order modeling (ROM) [1] is an essential analytical tool when repeated execution of a computational model is required for engineering-oriented studies such as uncertainty quantification, sensitivity analysis (SA), identification studies, and model inversion. For most engineering models exhibiting sufficient real-world complexity, the repeated execution of the forward model becomes computationally intractable. This follows as most real-world complex models (e.g., nuclear reactors) require long execution times, are associated with nonlinear models tightly coupling different physics and scales, and are manipulating millions of input parameters and output responses. Although many ROM techniques have been developed over the years to address some of the pertinent challenges, application to real-world engineering models is still considered computationally challenging.

Recently, a new hybrid application of several state-of-the-art ROM techniques has been developed in a manner that renders efficient their execution for complex engineering models [2, 3]. Both local variational methods [4] and global sensitivity methods [4–8] are hybridized to generate an ROM model that determines the variations (linear and nonlinear) in all responses due to all possible variations in input parameters in a computationally efficient manner with quantifiable accuracy. These developments were demonstrated in radiation transport models employed in typical nuclear reactor design calculations [9]. Reactor design calculations present excellent test beds for exercising reduced

[*]Correspond to Hany S. Abdel-Khalik, E-mail: abdelkhalik@ncsu.edu

2152–5080/12/$35.00

order methods as they present many challenges that are collectively shared by many engineering fields. Some of these challenges include the curse of dimensionality represented by the explosion in the number of dimensions of both input parameters and output response space, strong nonlinear behavior, and the multi-scale and multi-physics nature of the associated models.

The objective of this manuscript is twofold. First, we further develop this hybrid methodology by (a) extending it to multi-component models and (b) presenting a new conditioning criterion for the active subspace generated to achieve order reduction. Second, in order to motivate the discussion for the new development, we present a pedagogical overview of our previous developments in the area of hybridized ROM.

Given the hybrid nature of our developments, we will focus primarily on the new developments rather than providing extensive reviews for each of the areas employed in the hybridization process. In doing so, we recognize that several excellent review articles have been developed by others such as proper orthogonal decomposition, response surface methodologies, randomized algorithms for low rank matrix approximation, etc. However, for the sake of a complete presentation, we will provide a brief overview of the state-of-the-art methods supported by references to several comprehensive review articles. The interested reader is encouraged to consult these review articles and the references therein.

The manuscript is organized as follows. First a brief overview of the building blocks of the ROM hybridization process is given. This includes an overview of existing ROM techniques, variational and sampling-based methods, and randomized algorithms for low rank matrix approximation. Second, an overview of our past ROM developments is presented, followed by this manuscript's new developments. Finally, illustrative case studies are given, followed by conclusions and directions for future work.

Before we begin our review, we would like to introduce some nomenclature. As mentioned before, the hybrid ROM (h-ROM) approach is based on hybridization of local and global SA methods. To extend its applicability to multi-component models, a recently developed SA approach will be employed. As will be discussed later, this approach hybridizes another set of SA methods for multi-component models, namely, the amalgamated and brute force methods. This approach will be referred to as hybrid multi-component SA (h-MCSA) approach.

2. REDUCED ORDER MODELING

Computational scientists have invested in ROM techniques to build surrogates whose predictions are sufficiently close to the original model. Response surface methods (RSMs) [10] represent the most common approach for building surrogates. Methods like Gaussian process models, stochastic collocation methods, polynomial chaos methods (see Refs. [5, 7, 8] for an overview), and generalized Fourier expansion methods [11] have a common feature shared by all RSM methods—that is, the functional form of the response surface is pre-selected with some undetermined features; features may represent coefficients, function ranges, splines types, knots placement [12], etc. The original model is then executed at a number of points in the input parameter space and a minimization problem is employed to determine the unknown features that minimize the discrepancies between the surrogate's and the original model's predictions at the sampled points.

Regardless of the strategy developed to pick the random samples for RSM methods, an unavoidable challenge is when the number of afforded samples is much less than the dimension of the parameter space, which may be described as follows: the structure of the surrogate is unlikely to match that of the original model unless some insight is available a priori. Consequently, one would expect the surrogate to miss some of the influential parameter dependencies, which may have a noticeable impact on the responses of interest. Therefore, it becomes difficult to assess the accuracy of the surrogate outside the range of the samples employed for its construction.

To combat some of the challenges of RSM methods, projection methods have proven effective for the construction of ROMs. Projection methods recast the model in terms of a reduced basis. The computational cost needed to construct the ROM is proportional to the size of the reduced basis. Many methods have been employed to compute the basis; e.g., Krylov subspace methods [13], approximate balanced truncation methods [14], and proper orthogonal decomposition (POD) methods [15]. The POD methods have proven to be the most effective for large scale linear models and, more recently, for nonlinear models (see, for example, Refs. [16] and [17]). A typical POD method expands the state variables in terms of pre-selected basis functions, which are also employed to develop the so-called weak form of

the equations. The weak form is derived by projecting the original model's equations onto a subspace spanned by the selected basis. Several choices of the basis functions could be made. Notably, the method of snapshots has proven to be one of the most effective for both linear and nonlinear models [18]. It achieves its effectiveness by tailoring the basis functions to the unique features of the model. This is done by collecting "snapshots" of the solutions to the governing equations at different times.

Although POD methods are distinctively different from RSM methods, they share an important characteristic; that is, the subspace employed for ROM is also pre-selected. Therefore, like RSM methods, the subspace choice does not guarantee the structure of the original model will be properly captured by the ROM model for all possible parameter–response variations.

It is worth mentioning that the control community has made significant contributions to the area of ROM. Although a decade old, an excellent overview of ROM methods may be found in Ref. [19].

3. SENSITIVITY ANALYSIS METHODS

Since the h-ROM hybridizes the use of global and local methods for sensitivity analysis, it is instructive to first give a short overview highlighting the distinctive features of these two classes of methods. See Ref. [4] for a comprehensive comparison. The discussion in this section will be focused on single component models only. Section 5 will discuss SA methods for multi-component models before introducing the h-MCSA approach.

Both global and local SA methods are employed to propagate parameter uncertainties and extract dominant parameter sensitivities on the estimated response uncertainties. Two different classes of global methods have appeared in the literature: global sampling-based methods [6], and global variational-based methods [4]. Global sampling-based methods employ the forward model to randomly sample all input parameters from their prior probability density functions (PDFs). After each run, response deviations from reference values are recorded and the procedure is repeated with different random samples until a reliable estimate of response PDFs is obtained. This method is advantageous because of its simplicity and ability to obtain detailed (i.e., all moments) PDFs for all responses.[1] This is primarily important for a general nonlinear model and general input parameter PDFs where the response PDFs are expected to deviate considerably from the Gaussian shape (often employed to describe PDFs when only first and second moments are available). The disadvantage is that sensitivity information is more difficult to infer (often sought via a response surface or an analysis of variance approach), and the number of model executions can be too large to render the approach practical for high-dimensional models. When the number of input parameters is sufficiently small, one could exhaustively sample the parameter space to identify important sensitivities, especially when the models are highly nonlinear. With the number of input parameters numbering in the tens of thousands and larger, the search algorithm becomes computationally prohibitive. Computational scientists have devised numerous strategies to limit the number of samples with considerable success [5, 7, 8].

Global variational-based methods employ a rigorous mathematical framework to determine all the unique features of the model (e.g., response maxima and minima and bifurcation points) in the combined phase space formed by the model parameters, state dependent variables, and adjoint variables [4]. The identified critical points are then further analyzed using local SA methods. The framework employs a functional analysis approach to form a Lagrange functional of system response of interest, the set of linear or nonlinear equations relating the model's input parameters to system-dependent variables, and the set of equality and/or inequality constraints used to delimit parameter values. The Lagrange functional is then minimized using Kuhn–Tucker necessary conditions to identify the model's critical points. While being genuinely effective in identifying all unique features of the model, the computational requirements to solve the minimization problem is expensive; thereby limiting its applicability to models with few parameters.

Alternatively, the local variational methods trade obtaining detailed response PDFs for achieving computational efficiency and obtaining more sensitivity information. In their most practical implementation, the model is linearized and only first-order derivatives of responses with respect to input parameters are determined. With this information only, the response PDFs are approximated as linear combinations of the input parameter PDFs. And, if the input parameter uncertainties are described by Gaussian PDFs, the responses will also have Gaussian PDFs. However,

[1] This is only true in the limit as the number of samples approaches infinity.

if the model is nonlinear, response PDFs are expected to deviate from the Gaussian shape. The advantage is that sensitivity information can help identify influential input parameters responsible for propagated uncertainties. The disadvantage is that if the model deviates considerably from being linear, this approach will poorly approximate the response PDFs. In realistic engineering applications, this is often the case and, therefore, higher-order effects must be quantified to properly capture tail probabilities. To identify higher-order effects, the mathematical framework for variational methods has been extended to nonlinear models [20]. For example, to characterize second-order effects, an extra execution of the adjoint model is required for each input parameter in order to characterize its second-order derivatives that capture its interaction with other parameters. Clearly, for a high-dimensional model, this approach becomes computationally prohibitive. For an overview of the state-of-the-art in high-order local variational methods, the reader is encouraged to consult a previous publication [21].

4. HYBRID APPLICATION OF SINGLE-COMPONENT SA METHODS

Past work has demonstrated that a hybrid method could be devised to combine the benefits of both local and global methods and circumvent some of their deficiencies [2]. This section describes in detail this hybridization process in order to set the stage for the development of the h-ROM model.

The hybridization of local and global sensitivity methods consists of two steps. In the first step, local methods are employed to identify a subspace (denoted as the active subspace) that captures simultaneously *all* direct and indirect parameter cross-interaction effects including all higher orders that are inherent in the original model. This is done based on a numerical tolerance specified by the user that sets an upper bound on the maximum allowed discrepancy between the original and reduced model. This implies all parameter cross interactions that are below the specified tolerance will be considered non-influential and, hence, not included in the active subspace. In the second step, global methods are used to build a surrogate that searches only the active subspace for detailed sensitivity information. The surrogate could also be employed to efficiently propagate parameter uncertainties.

The first step employs a snapshot-like method to identify the basis functions used to describe the active subspace. There are three marked differences with the existing snapshot algorithm. First, each snapshot represents a single execution of the adjoint model as opposed to the forward model. Second, the forward model does not need to be a time-marching model as opposed to the existing snapshot algorithm, which requires different time steps to collect the snapshots. Third, in every execution, all input parameters are randomly perturbed within their user-defined ranges of variations as opposed to keeping them at their reference values. This process is repeated until the active subspace in the parameter space is properly spanned. The forward model is then executed with all input parameter perturbations confined to the active subspace.

This idea first appeared in Ref. [22] to approximate the sensitivity matrix for a large-scale computational model. In this context, a sensitivity matrix is a rectangular matrix that contains all first-order derivatives of all responses with respect to all input parameters. The goal was to reduce the computational burden required when the number of input parameters and output responses number in the millions. The idea was to rely on low rank matrix approximations that are well established in the mathematical community [23]. The result of this work was an algorithm that constructs a low rank approximation to a matrix whose elements are not explicitly available; however, they can be accessed via matrix–vector products only. Unfortunately, the error bounds were not rigorously constructed at the time; only engineering judgment regarding the desired level of accuracy and numerical experiments were employed to validate that the size of the subspace is adequate for user-defined accuracy requirements. Recently, however, applied mathematicians were able to determine the minimum rank required to satisfy an error bound that quantifies the discrepancy between the original matrix and its low rank approximation [24]. We employ these combined developments in conjunction with a tensor-free infinite series described later to determine with confidence an active subspace that meets user-defined error bounds for a general nonlinear model. This represents one of the pressing challenges facing the state-of-the-art in ROM methods.

Previous work has extended the developments in Refs. [22] and [24] to allow the determination of the active subspace that captures higher-order response variations; i.e., nonlinear effects [2, 3, 9]. Higher-order effects are often represented by tensors. It was shown that by employing a modified representation of an infinite series expansion—denoted here as tensor-free expansion—the higher-order variations of responses could be represented by a matrix

operator amenable to low rank approximation. This allows a natural extension of the above developments to nonlinear models.

The first subsection gives a brief overview of the algorithm employed to generate a low rank approximation as described in Ref. [22] and the construction of the error norms in Ref. [24]. The following subsection introduces the tensor-free infinite series expansion and describes how it is used to identify the active subspace, first for a single-valued model, then for a multi-response model.

4.1 Construction of Low Rank Matrix Approximation

Consider a matrix operator $\mathbf{A} \in \mathbb{R}^{m \times n}$ whose elements are not explicitly available; but assume that the following two operations are available:

1. Matrix–vector product: $\vec{y} = \mathbf{A}\vec{x}$; and
2. Matrix–transpose–vector product: $\vec{z} = \mathbf{A}^T \vec{w}$.

A low rank approximation of matrix **A** may be constructed as follows [this is denoted by the subspace identification algorithm (SIA)]:

1. Generate k random realizations of parameters $\left\{ \vec{x}_i \right\}_{i=1}^{k}$.
2. Calculate $\vec{y}_i = \mathbf{A}\vec{x}_i$ for $i = 1, ..., k$.
3. Calculate the decomposition: $\left[\begin{array}{cccc} \vec{y}_1 & ... & \vec{y}_k \end{array} \right] = \mathbf{Y} = \mathbf{QR} = \left[\begin{array}{cccc} \vec{q}_1 & ... & \vec{q}_k \end{array} \right] \mathbf{R}$.
4. Determine the effective rank r (see notes below).
5. If $r < k$ stop, otherwise go back to step 1 and add more realizations of $\vec{x}$.
6. Calculate $\vec{p}_i = \mathbf{A}^T \vec{q}_i$ for $i = 1, ..., r$.
7. Using $\left\{ \vec{q}_i \right\}_{i=1}^{r}$ and $\left\{ \vec{p}_i \right\}_{i=1}^{r}$, calculate a low rank approximation $\mathbf{A} = \mathbf{USV}^T$, where $\mathbf{U} \in \mathbb{R}^{m \times r}$, $\mathbf{V} \in \mathbb{R}^{n \times r}$, and $\mathbf{S} \in \mathbb{R}^{r \times r}$. The ranges of matrices **U** and **V** are defined such that $\mathrm{R}(\mathbf{U}) = \mathrm{span}\{\vec{q}_1, ..., \vec{q}_r\}$ and $\mathrm{R}(\mathbf{V}) = \mathrm{span}\{\vec{p}_1, ..., \vec{p}_r\}$. We denote the $\mathrm{R}(\mathbf{U})$ and $\mathrm{R}(\mathbf{V})$ by the active subspaces in $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively. For details on how these matrices are calculated, see Appendix B. For a more comprehensive treatment, the full details may be found in the Appendix of Ref. [22].

Note that the above algorithm identifies both the range of the matrix and its transpose. Depending on the application, one may need both or only one of them. Notice that k matrix–vector products are employed initially to identify one of the two subspaces where $k > r$; this follows as one does not know the rank a priori. Moreover, as described later, a few additional matrix–vector products (called oversamples) are need to ensure the user-defined tolerance is satisfied. Once the rank is identified, one needs only r matrix–vector products to determine the second subspace. An example is given in Appendix B.

From a practical viewpoint, the rank could be determined by trial and error until an acceptable accuracy for the reduced model is achieved. However, from a mathematical viewpoint some upper bound on the error must be established. In Ref. [24], it is shown that the rank could be identified as follows (this is denoted as the rank identification algorithm (RIA)]:

1. Pick a small integer s; $s = 10$ is often very conservative.
2. Pick a sequence of s random Gaussian vectors $\left\{ \vec{w}_i \right\}_{i=1}^{s}$.

3. Calculate $\vec{y}_i = \left(\mathbf{I} - \mathbf{Q}\mathbf{Q}^T\right)\mathbf{A}\vec{x}_i$ for $i = 1, ..., s$, where $\mathbf{Q}$ is identified by the SIA algorithm and has dimensions $n \times r$.
4. Calculate $\varepsilon = \max_{i=1,..,s} \left\|\vec{y}_i\right\|_2$.

One can show that with probability at least $1 - 10^{-s}$, the effective rank r can be determined by testing the following criterion:

$$\left\|\left(\mathbf{I} - \mathbf{Q}\mathbf{Q}^T\right)\mathbf{A}\right\|_2 \leq \varepsilon \Big/ \left(10\sqrt{2/\pi}\right) \tag{1}$$

where ε is the user-defined error tolerance. This statement assures that the norm of the truncated part of the matrix (i.e., not represented by the low rank approximation) is bounded by some value that can be made as small as desired by the user. If the criterion in Eq. (1) is not reached, the estimate of the rank from the SIA algorithm is increased and the RIA algorithm is repeated. The power of the RIA algorithm is that once the criterion in Eq. (1) is satisfied, no more expansion of the active subspace is necessary, and the SIA algorithm could be terminated.

4.2 Construction of Active Subspace

In this section we discuss how the previous low rank approximation algorithms could be applied to a given model to identify the active subspace. Let the original model be described by the following equation:

$$\vec{y} = \vec{\Theta}\left(\vec{x}\right) \tag{2}$$

where $\vec{x} \in \mathbb{R}^n$ is a n-component vector representing the n-independent input parameters to the model, and the m model's responses are represented by the vector $\vec{y} \in \mathbb{R}^m$.

4.2.1 Active Subspace of Single-Valued Function

Consider first the case when $m = 1$; i.e., a single-valued model. A general nonlinear function $\Theta\left(\vec{x}\right)$ of n input parameters represented by a vector $\vec{x} \in \mathbb{R}^n$ may be expanded around a reference point $\vec{x}_0$ as follows [without loss of generality, we assume $\vec{x}_0 = \vec{0}$ and $\Theta\left(\vec{x}_0\right) = 0$ to simplify the expressions]:

$$\Theta\left(\vec{x}\right) = \sum_{k=1}^{\infty} \sum_{j_1,..j_l..,j_k=1}^{n} \psi_1\left(\vec{\beta}_{j_1}^{(k)T}\vec{x}\right) .. \psi_l\left(\vec{\beta}_{j_l}^{(k)T}\vec{x}\right) .. \psi_k\left(\vec{\beta}_{j_k}^{(k)T}\vec{x}\right) \tag{3}$$

where $\{\psi_l\}_{l=1}^{\infty}$ can be any kind of scalar functions. The outer summation over the variable k goes from 1 to infinity. Each term represents one order of variation: e.g., $k = 1$ represents the first-order term; $k = 2$, the second-order term, etc. For the case of $\psi_l(\theta) = \theta$, the kth term reduces to the kth term in a multi-variable Taylor series expansion. The inside summation for the kth term consists of k single valued functions $\{\psi_l\}_{l=1}^{k}$ that are multiplying each other. The arguments for the $\{\psi_l\}_{l=1}^{k}$ functions are scalar quantities representing the inner products between the vector $\vec{x}$ and the n vectors $\left\{\vec{\beta}_{j_l}^{(k)}\right\}_{j_l=1}^{n}$ that span the parameter space. The superscript (k) implies that a different basis is used for each order; i.e., one basis is used for the first-order term, another for the second-order term, and so on. This expression provides an exact value for the function $\Theta\left(\vec{x}\right)$ at every point in the parameter space. This could be easily verified by reducing it to a Taylor series expansion by assuming: $\psi_l(\theta) = \theta$. See Appendix A for a formal proof. We give here a few examples of this expansion:

1. Linear model: $\Theta\left(\vec{x}\right) = \vec{a}^T\vec{x}$. Observe that any variations in the input parameter space that are orthogonal to the vector $\vec{a}$ will not lead to changes in the function's value. This model can be reduced to one with a single effective input parameter.

2. Second-order model: $\Theta\left(\vec{x}\right) = \vec{a}^{\,T}\vec{x} + \sum_{i=1}^{n} s_i \left(\vec{v}_i^{\,T}\vec{x}\right)^2$. The term comprising the second-order summation represents a singular value decomposition (SVD) expansion of the Hessian term, often used to represent the second-order term. If all the terms in the summation are appreciably large, this model cannot be reduced since all directions in the input parameter space induce changes in the function. Notice that, if the summation only goes up to r that is smaller than n then a reduction is possible.

3. Third degree polynomial model: $\Theta\left(\vec{x}\right) = \vec{a}^{\,T}\vec{x} + \left(\vec{b}^{\,T}\vec{x}\right)^2 + \left(\vec{c}^{\,T}\vec{x}\right)\left(\vec{d}^{\,T}\vec{x}\right)\left(\vec{e}^{\,T}\vec{x}\right)$. Similarly, any variations in the parameters that satisfy the following condition: $\vec{x} \perp \text{span}\left\{\vec{a}, \vec{b}, \vec{c}, \vec{d}, \vec{e}\right\}$ will not change the function's value.

The above ideas can be generalized such that any input parameter variations that are orthogonal to the range formed by all the $\left\{\vec{\beta}_{j_l}^{(k)}\right\}$ vectors will not lead to changes in the function's value. If these vectors span the entire input parameter space (i.e., span $\left\{\text{all } \vec{\beta}_{j_i}^{(k)}\right\} = \mathrm{R}^n$), then all directions are important and no reduction is possible. However, if they span a subspace of size r, then one can reduce the effective number of input parameters from n to r. We show next that the mathematical range of these vectors could be identified by relying only on first-order derivatives.

Differentiating Eq. (3) with respect to $\vec{x}$ gives:

$$\vec{\nabla}\Theta\left(\vec{x}\right) = \sum_{k=1}^{\infty} \sum_{j_1,..j_l..,j_k=1}^{n} \psi_1\left(\vec{\beta}_{j_1}^{(k)T}\vec{x}\right)..\psi_{l-1}\left(\vec{\beta}_{j_{l-1}}^{(k)T}\vec{x}\right)\psi_l'\left(\vec{\beta}_{j_l}^{(k)T}\vec{x}\right)\psi_{l+1}\left(\vec{\beta}_{j_{l+1}}^{(k)T}\vec{x}\right)..\psi_k\left(\vec{\beta}_{j_k}^{T}\vec{x}\right)\vec{\beta}_{j_l}^{(k)} \quad (4)$$

where $\psi_l'\left(\vec{\beta}_{j_l}^{(k)T}\vec{x}\right)\vec{\beta}_{j_l}^{(k)}$ is the derivative of the term $\psi_l\left(\vec{\beta}_{j_l}^{(k)T}\vec{x}\right)$. Note that each term of this summation represents a vector in the parameter space pointing in the direction of $\vec{\beta}_{j_l}^{(k)}$. Accordingly, the gradient of the response function $\vec{\nabla}\Theta\left(\vec{x}\right)$ may be viewed as a linear combination of all the $\left\{\vec{\beta}_{j_l}^{(k)}\right\}$ vectors:

$$\vec{\nabla}\Theta\left(\vec{x}\right) = \sum_{k=1}^{\infty} \sum_{j_1,..j_l..,j_k=1}^{n} \chi_{j_l}^{(k)}\vec{\beta}_{j_l}^{(k)} = \left[\begin{array}{cccc} \vec{\beta}_1^{(1)} & \cdots & \vec{\beta}_{j_l}^{(k)} & \cdots \end{array}\right] \left[\begin{array}{c} \alpha_1^{(1)} \\ \vdots \\ \alpha_{j_l}^{(k)} \\ \vdots \end{array}\right] = \mathbf{B}\vec{\chi} \quad (5)$$

where

$$\chi_{j_l}^{(k)} = \psi_1\left(\vec{\beta}_{j_1}^{(k)T}\vec{x}\right)..\psi_{l-1}\left(\vec{\beta}_{j_{l-1}}^{(k)T}\vec{x}\right)\psi_l'\left(\vec{\beta}_{j_l}^{(k)T}\vec{x}\right)\psi_{l+1}\left(\vec{\beta}_{j_{l+1}}^{(k)T}\vec{x}\right)..\psi_k\left(\vec{\beta}_{j_k}^{T}\vec{x}\right)$$

One can make several observations about Eq. (5). First, note that the **B** matrix is unknown a priori. As mentioned before, to render a ROM model, one needs to approximate the mathematical range of the matrix **B**, which can be constructed using the SIA algorithm described earlier using the first five steps only, since only the R (**B**) is required. Moreover, the size of this subspace can be determined with quantifiable accuracy using the RIA algorithm. The implication is that the function (although nonlinear) depends only on r effective dimensions and, hence, can be reduced. Second, note that the expansion employed to represent the function and its derivative has infinite radius of convergence since the expansion goes up to infinity, implying that it is accurate anywhere in the parameter space. This is important as no approximations were made about the function shape when constructing the range of matrix **B**. This follows as matrix **B** contains all the beta vectors responsible for all the high-order terms that may be present in the function. Third, note that to generate the ROM, one only requires access to the subspace that represents the mathematical range

of matrix **B**. Mathematically, there is an infinite number of bases that can be used to span the same subspace. Therefore, it is important to emphasize that our goal is not to identify the particular beta vectors that appear in the expansion in Eq. (3); however, the main goal is to find the associated subspace. Fourth, note that in order to implement the SIA and RIA algorithms, one needs the ability to calculate random matrix–vector products of the form $\mathbf{B}\vec{\chi}$. To achieve that, note that vector $\vec{\chi}$ in Eq. (5) is a function $\vec{x}$; implying that if the derivatives are evaluated at random points in the parameter space, one could emulate the effect of multiplying matrix **B** with a random vector $\vec{\chi}$. In Ref. [24], it is shown that the performance of the SIA and RIA algorithms is very insensitive to the choice of the random vector distribution. Therefore, any simple random strategy should be sufficient to determine the perturbed parameter values at which the derivatives are calculated.

To summarize, the algorithm to determine the active subspace for a single-valued function is given here:

1. Generate k random realizations of the parameters $\left\{\vec{x}_i\right\}$.
2. Calculate the derivative of the function at each of the points $\left\{\vec{x}_i\right\}$, and aggregate the resulting derivatives in a matrix:
$$\mathbf{G} = \left[\frac{d\Theta}{d\vec{x}}\,|_{\vec{x}_1} \quad \ldots \quad \frac{d\Theta}{d\vec{x}}\,|_{\vec{x}_k}\right] \in \mathbb{R}^{n\times k}$$
3. Calculate the QR decomposition of $\mathbf{G} = \mathbf{QR}$.
4. Apply the RIA algorithm to determine the rank and measure error norm in Eq. (1).
5. If the error norm does not meet user-specified tolerance, go back to step 1 and add more realizations for $\vec{x}$; otherwise stop.
6. The basis of the active subspace is given by the first r columns of matrix **Q**.

4.2.2 Active Subspace of Multi-Response Model

The previous development is now extended to models with multiple responses. Define the qth response's function and gradient as:

$$\Theta_q\left(\vec{x}\right) = \sum_{k=1}^{\infty}\sum_{j_1,..j_l..,j_k=1}^{n} \psi_1\left(\vec{\beta}_{j_1,q}^{(k)T}\vec{x}\right) .. \psi_l\left(\vec{\beta}_{j_l,q}^{(k)T}\vec{x}\right) .. \psi_k\left(\vec{\beta}_{j_k,q}^{(k)T}\vec{x}\right) \tag{6}$$

$$\vec{\nabla}\Theta_q\left(\vec{x}\right) = \sum_{k=1}^{\infty}\sum_{j_1,..j_l..,j_k=1}^{n} \psi_1\left(\vec{\beta}_{j_1,q}^{(k)T}\vec{x}\right) ..\psi_l'\left(\vec{\beta}_{j_l,q}^{(k)T}\vec{x}\right) ..\psi_k\left(\vec{\beta}_{j_k,q}^{T}\vec{x}\right)\vec{\beta}_{j_l,q}^{(k)} \tag{7}$$

Given that the active subspace will generally be different for different responses, the overall model's active subspace is selected to span the union of all single-response active subspaces. As done before, define a matrix **B** that contains all the $\left\{\vec{\beta}_{j_1,q}^{(k)}\right\}$ vectors associated with all orders and all responses. As before, this matrix will have n rows and infinite columns. In order to find a low rank approximation to this matrix, one needs the matrix–vector product operation. This could be achieved by defining a pseudo-response of the form:

$$\Theta^{\text{pseudo}}\left(\vec{x}\right) = \sum_{q=1}^{m}\gamma_q\sum_{k=1}^{\infty}\sum_{j_1,..j_l..,j_k=1}^{n} \psi_1\left(\vec{\beta}_{j_1,q}^{(k)T}\vec{x}\right) .. \psi_l\left(\vec{\beta}_{j_l,q}^{(k)T}\vec{x}\right) .. \psi_k\left(\vec{\beta}_{j_k,q}^{(k)T}\vec{x}\right) \tag{8}$$

where $\{\gamma_q\}$ are randomly generated. Notice that the pseudo-response is a random linear combination of the original m responses. The gradient of the pseudo-response is given by

$$\vec{\nabla}\Theta^{\text{pseudo}}\left(\vec{x}\right)=\sum_{q=1}^{m}\gamma_q\sum_{k=1}^{\infty}\sum_{i_1,...i_l,...i_k=1}^{n} \psi_1\left(\vec{\beta}_{i_1,q}^{(k)T}\vec{x}\right)..\psi_{l-1}\left(\vec{\beta}_{i_{l-1},q}^{(k)T}\vec{x}\right)\psi_{l+1}\left(\vec{\beta}_{i_{l+1},q}^{(k)T}\vec{x}\right)..\psi_l'\left(\vec{\beta}_{i_l,q}^{(k)T}\vec{x}\right)\vec{\beta}_{i_l,q}^{(k)} \tag{9}$$

By evaluating the derivative of the pseudo-response at multiple random points in the input parameter space, the active subspace corresponding to all responses could be identified.

In practical situations when the number of input parameters is too large, derivative information is often generated using a variational approach. In this case, one can calculate the derivatives of the pseudo-response directly without calculating the individual response derivatives. This proves to be important for models with many responses, where executing the adjoint model m times could be computationally prohibitive. See, for example, Ref. [25] for a demonstration on the construction of pseudo-responses using a variational approach applied for linear models.

The algorithm to identify the active subspace for a multi-response model proceeds as follows:

1. Generate m random numbers $\{\gamma_q\}_{q=1}^{m}$.
2. Formulate a pseudo-response of the form: $\Theta^{\text{pseudo}} = \sum_{q=1}^{m} \gamma_q \Theta_q$ for the sensitivity analysis procedure employed.
3. Generate k random realizations of the input parameters $\left\{\vec{x}_i\right\}$.
4. Calculate the derivative of the pseudo-function at each of the points $\left\{\vec{x}_i\right\}$, and aggregate the resulting derivatives in a matrix:
$$\mathbf{G} = \left[\left.\frac{d\Theta^{\text{pseudo}}}{d\vec{x}}\right|_{\vec{x}_1} \quad \ldots \quad \left.\frac{d\Theta^{\text{pseudo}}}{d\vec{x}}\right|_{\vec{x}_k} \right] \in \mathbb{R}^{n\times k}$$
5. Calculate the QR decomposition of $\mathbf{G} = \mathbf{QR}$.
6. Apply algorithm RIA to determine the rank and measure error norm in Eq. (1).
7. If the error norm does not meet user-specified tolerance, go back to step 1 and add more realizations for $\vec{x}$; otherwise stop.

4.3 Global Sensitivity Analysis

The previous section described how the active subspace in the input parameter space may be identified. This knowledge could be used to express the model as

$$\vec{y}^{\text{SUB}} = \vec{\Theta}\left(\vec{x}\right) = \vec{\Theta}\left(\mathbf{Q}\vec{\alpha}\right) \tag{10}$$

where $\vec{\alpha} \in \mathbb{R}^r$. The $\vec{y}^{\text{SUB}}$ denotes the responses calculated using the original model but with the parameters constrained to the active subspace. The error resulting from this is the error that can be constrained by the RIA algorithm as discussed earlier.

Note that the effective dimensionality of the problem has now been reduced from n to r. If the reduction is significant, it may now become possible to employ RSM to find surrogates as functions of the reduced variables $\vec{\alpha}$. Effectively, one tries to find a surrogate that approximates Eq. (10):

$$\vec{y}^{(\text{ROM})} = \vec{\Theta}^{(\text{surrogate})}\left(\vec{\alpha}\right) \tag{11}$$

The error resulting from the surrogate construction will depend on the specific method employed to build the surrogate. This is important in order to distinguish this error from the error resulting from the reduction in Eq. (10). With the surrogate constructed in terms of the reduced variables, it could be easily transformed back in terms of the original variables by employing the back substitution $\vec{\alpha} = \mathbf{Q}^T\vec{x}$ such that:

$$\vec{y}^{(\text{ROM})} = \vec{\Theta}^{(\text{surrogate})}\left(\mathbf{Q}^T\vec{x}\right) \tag{12}$$

Note that in this expression any input parameter variations that are orthogonal to the active subspace [i.e., the range of the matrix $\mathbf{Q}$, $\mathrm{R}(\mathbf{Q})$] will produce a zero change in the reduced variables $\vec{\alpha}$ and, hence, no change in the ROM model's predictions for the responses $\vec{y}^{(\mathrm{ROM})}$.

Before concluding this section, we would like to make a few important remarks. The methodology presented allows one to reduce the effective number of input parameters by identifying a subspace. However, it does not advise the user on how to build a surrogate, which must be constructed to reach the ultimate goal of the reduction. The computational cost for building a surrogate is highly dependent on the number of input parameters. If the method is applied successfully and the effective number of parameters can be reduced, then computational savings could be realized for the downstream calculations employed to build the surrogate. However, if the downstream calculations have outstanding issues such as rate of convergence, stability, radius of convergence, etc., the surrogate developed is expected to suffer from the very same issues whether the input parameters are reduced or not. We do not attempt to address any of these issues in this paper. Finally, the expression in Eq. (12) does not imply a deterministic approach to building a surrogate or a particular RSM approach; however, it is considered as a general approach that approximates the relationship between responses (quantities of interest) and the input parameters; e.g., a Monte Carlo model could be employed to capture this relationship.

5. EXTENSION TO MULTI-COMPONENT MODELS

This section extends the above hybrid methodology (h-ROM) to multi-component models. This is important since in most real-world engineering models the quantities of interest are calculated using a chain of codes. A typical example is the so-called multi-scale phenomena modeling strategy, which has found wide application in many engineering disciplines. In this modeling strategy, several models are employed to describe system behavior starting with detailed first-principles fine scale models and ending with coarse scale models to predict the system's macroscopic performance metrics. From a high level, this modeling strategy may be viewed as an assembly of numerous models coupled together in various manners to account for the different scales and physics that affect system behavior. The interconnectivity of the models complicates the manner in which sensitivity information is transferred between the models. Clearly, if a forward sensitivity approach is employed, one can run the overall model a number of times proportional to the number of input parameters to get the first-order derivatives using a finite-differencing approach. However, in reality the number of input parameters is very large and, hence, in our development it is assumed that the finite-differencing approach is not possible. We, therefore, focus on adjoint SA.

Although powerful adjoint SA tools may exist for the individual scales and/or physics models (often referred to as single-physics or single-scale adjoints and, hereinafter, denoted as single-component adjoints), there is often no generally accepted way to formulating a global[2] adjoint for the multi-scale multi-physics model (denoted, hereinafter, by the multi-component model). A global adjoint is often much more complicated to implement and must be planned in advance for the particular set of components. Given the dynamic nature of code development and the need to utilize and exchange models frequently, it is paramount to design SA algorithms that can generate sensitivity information for multi-component models from the single-component adjoint. This is a challenging task since a global adjoint for a multi-component model depends on the manner in which the single-component models are connected.

Recently, a new method has been developed to elucidate the coupling of adjoint sensitivity information between different components in a multi-component model [25]; this method is denoted by h-MCSA. The method hybridizes two existing methods for evaluating sensitivity information for a multi-component model: brute force methods[3] and amalgamated methods [4]. The brute force method is simple to implement but as will be shown in the next section

[2]Note that the term "global" has been used by various practitioners to mean different things. In the context of variational methods [4], global refers to the construction of an adjoint model that represents the *entire* model. In the sampling SA community [6], global is used to mean two different things: (a) the ability to generate sensitivity information over *all possible ranges of variations* for all parameters; and (b) the ability to generate sensitivity information for *all* parameters. Unfortunately, it is hard to avoid these inconsistencies even in the same community.

[3]In this context, *brute force* does not refer to finite-difference methods for evaluation of first-order derivatives. However, it refers to brute forcing the application of adjoint methods to all the components of a multi-component model. To our knowledge, this terminology has not been used before in the literature on global adjoint methods.

requires significant computational overhead. The amalgamated method minimizes the number of adjoint evaluations but requires great insight into the components models. This development is employed in the current work to calculate first-order derivatives of the responses calculated by the last component with respect to the parameter input to the first component; evaluating these derivatives represents a key factor in determining the active subspace for the ROM model.

The basic idea behind the h-MCSA method is now presented for a simple two-component linear model with a single response. Let Θ, as described previously in Eq. (2), denote the corresponding mathematical operator for the overall model. The two components can be described by:

$$y = \Pi\left(\vec{z}\right) \quad \text{and} \quad \vec{z} = \vec{\Omega}\left(\vec{x}\right) \tag{13}$$

where $\vec{z}$ is the output (assumed to consist of l elements, $\vec{z} \in \mathbb{R}^l$) of the first component, which is passed as input to the second component. Using the chain rule of differentiation, the first-order derivatives of output response y with respect to input data $\vec{x}$ are given by

$$\frac{dy}{d\vec{x}} = \frac{dy}{d\vec{z}} \times \frac{d\vec{z}}{d\vec{x}} \tag{14}$$

which in matrix form reduces to $\vec{g}^T = \vec{\pi}^T \mathbf{\Theta}$, where $\vec{\pi}^T = dy/d\vec{z}$ is a row vector of l elements (or a matrix of dimensions $1 \times l$); $\mathbf{\Theta} = d\vec{z}/d\vec{x}$ is a matrix of dimension $l \times n$; and $\vec{g} \in \mathbb{R}^n$ is an n component vector representing the derivatives of y with respect to the elements of $\vec{x}$.

Two types of methods may be utilized to determine the global sensitivity matrix. The first method, denoted as the brute force method, determines the full sensitivity matrix $\mathbf{\Theta}$ and the sensitivity vector $\vec{\pi}$ and then performs the matrix–vector product in Eq. (14). For a K component model, this corresponds to evaluating the full sensitivity matrix for each component and then performing a K-matrix product. The second method, referred to as the amalgamated method, recognizes that if the equations describing each component are available, one could combine all equations to form one overall model with one output response. Given the equations describing the overall model, one could formulate the associated global adjoint problem, which would require only one adjoint execution of the overall model to evaluate the global sensitivity matrix. If implemented carefully, the single adjoint execution of the overall model implies one adjoint execution for each component that is independent of the number of output responses of the first component. To achieve that, one needs to write down all the equations corresponding to the overall model, set up the associated adjoint problem, and break it down into *two* adjoint problems, each associated with a component as done by amalgamated-type methods [4].

The recent developments offer an alternative approach to reach the same goal but without explicit access to the equations comprising each of the two components [25]. To achieve that, one must avoid the explicit evaluation of matrix $\mathbf{\Theta}$, which (if well-conditioned) requires executing the first component l times in an adjoint mode. This can be avoided by re-writing Eq. (14) as follows:

$$\vec{g} = \mathbf{\Theta}^T \vec{\pi} = \mathbf{\Theta}^T \mathbf{Q}\mathbf{Q}^T \vec{\pi} \tag{15}$$

where $\mathbf{Q} \in \mathbb{R}^{l \times l}$ is an orthonormal matrix such that $\mathbf{Q}\mathbf{Q}^T = \mathbf{I} \in \mathbb{R}^{l \times l}$. Select $\mathbf{Q}$ such that

$$\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^{||}\mathbf{Q}^{||T} + \mathbf{Q}^{\perp}\mathbf{Q}^{\perp T} \tag{16}$$

where $\mathrm{R}\left(\mathbf{Q}^{||} \in \mathbb{R}^{l \times 1}\right) = \mathrm{R}\left(\vec{\pi}\right)$ and $\mathrm{R}\left(\mathbf{Q}^{\perp} \in \mathbb{R}^{l \times (l-1)}\right) \perp \mathrm{R}\left(\vec{\pi}\right)$. In this representation matrix $\mathbf{Q}^{||}$ has only a single column (i.e., a vector) and matrix $\mathbf{Q}^{\perp}$ has $l - 1$ columns. Re-write Eq. (15) using the decomposition in Eq. (16):

$$\vec{g} = \mathbf{\Theta}^T \left(\mathbf{Q}^{||}\mathbf{Q}^{||T} + \mathbf{Q}^{\perp}\mathbf{Q}^{\perp T}\right) \vec{\pi} = \hat{\mathbf{\Theta}}_1^T \vec{\pi} + \hat{\mathbf{\Theta}}_2^T \vec{\pi} = \hat{\mathbf{\Theta}}_1^T \vec{\pi} \tag{17}$$

where $\hat{\mathbf{\Theta}}_1^T = \mathbf{\Theta}^T \mathbf{Q}^{||}\mathbf{Q}^{||T} \in \mathrm{R}^{n \times l}$ and $\hat{\mathbf{\Theta}}_2^T = \mathbf{\Theta}^T \mathbf{Q}^{\perp}\mathbf{Q}^{\perp T} \in \mathrm{R}^{n \times l}$. This decomposition implies that matrix $\mathbf{\Theta}$ may be split into two different pieces: $\hat{\mathbf{\Theta}}_1$, whose range coincides with the vector $\vec{\pi}$; and $\hat{\mathbf{\Theta}}_2$, whose range is orthogonal

to it. This means matrix $\hat{\mathbf{\Theta}}_2$ does not contribute to the left-hand side $(\vec{g})$ and, hence, needs not be evaluated. The first piece, $\hat{\mathbf{\Theta}}_1$, could be determined by performing a single adjoint execution of the first component. To do that, notice that one can transpose Eq. (17) to give:

$$\vec{g}^T = \left(\vec{\pi}^T \mathbf{Q}^{||}\right)\left(\mathbf{Q}^{||T}\mathbf{\Theta}\right) \tag{18}$$

The term $\mathbf{Q}^{||T}\mathbf{\Theta}$ is a vector multiplying a matrix from the left. By definition, this produces a linear combination of the rows of the matrix, and the weights are given by the elements of the vector. This linear combination could be generated using a single adjoint model execution by defining a pseudo-response of the form [25]:

$$\tilde{z} = \sum_{i=1}^{l} q_i z_i \tag{19}$$

where $\{q_i\}$ are the elements of vector $\mathbf{Q}^{||}$. If the model has more than one response, $\mathbf{Q}^{||}$ becomes a matrix, and one needs to define an equal number of pseudo-responses, each associated with one column of matrix $\mathbf{Q}^{||}$. When applied to a K component model with m responses, one needs to proceed as follows:

1. Execute the last component in an adjoint mode m times, one for each response.
2. Using the m gradients, formulate m pseudo-responses for component $K-1$.
3. Execute component $K-1$ in an adjoint mode m times, one for each pseudo-response.
4. Repeat the same process for component $K-2$ until reaching the first component.

This presents the basic idea for the algorithm. Several other variations are possible when some of the components do not have an adjoint model, when some of the component sensitivity matrices are poorly conditioned, or when some of the components could be further reduced via the ROM model. The latter case is described next since it represents the goal of this work. Other variations may be found in Ref. [25].

Assume that one has K components that are serially connected; other types of connections will be considered in future work. Component k receives l_{k-1} inputs and generates l_k outputs such that: $l_0 = n$ and $l_K = m$. Let $\vec{z}_k = \left[\begin{array}{ccc} z_{k,1} & \ldots & z_{k,l} \end{array}\right]^T$ refer to the l outputs of component k. The algorithm in Section 4.2.2 may now be modified as follows to identify an active subspace at each component-to-component interface:

1. FOR $j = K, K-1, ..., 2, 1$.
2. Generate r_{j+1} random numbers $\vec{\Gamma}_j = \{\gamma_q\}_{q=1}^{r_{j+1}}$. If $j = K$, then $r_{j+1} = m$.
3. Formulate a pseudo-response of the form: $\tilde{z}_j^{\text{pseudo}} = \vec{\Gamma}_j^T \mathbf{Q}_{j+1}^T \vec{z}_j$. If $j = K$, then $\mathbf{Q}_{j+1} = \mathbf{I}$.
4. Generate k random realizations of the input parameters to component j $\left\{\vec{z}_{j,i}\right\}_{i=1}^{k}$.
5. Calculate the derivative of the pseudo-responses at each of the points $\left\{\vec{z}_{k,i}\right\}$, and aggregate the resulting derivatives in a matrix:

$$\mathbf{A}_j = \left[\left.\frac{d\tilde{z}_j^{\text{pseudo}}}{d\vec{z}_j}\right|_{\vec{z}_{j,1}} \quad \ldots \quad \left.\frac{d\tilde{z}_j^{\text{pseudo}}}{d\vec{z}_j}\right|_{\vec{z}_{j,1}}\right] \in \mathbb{R}^{l_{j-1}\times k}$$

6. Calculate the QR decomposition of $\mathbf{A}_j = \mathbf{Q}_j\mathbf{R}_j$.
7. Apply algorithm RIA to determine the rank r_j and measure the error norm in Eq. (1).

8. If the error norm does not meet user-specified accuracy requirements, go back to step 2 and add more realizations for $\left\{\vec{z}_j\right\}$; otherwise stop.
9. END FOR.

The distinct difference between this algorithm and the algorithm in Section 4.2.2 is in the way the pseudo-responses at each component-to-component interface are being formulated. The formulation in step 3 implies that the pseudo-response for component j is a linear combination of the r_{j+1} reduced outputs that were determined by component $j+1$. This follows as the derivatives of the remaining responses (i.e., $l_j - r_{j+1}$) will be in the null space of the sensitivity matrix for component $j+1$ and, hence, need not be evaluated. Finally notice that the derivatives are randomly sampled to account for nonlinear behavior as was illustrated in Section 4.2.

6. STOPPING CRITERION FOR ACTIVE SUBSPACE

The h-ROM approach hinges on the ability to identify an active subspace that is generated via sampling of first-order derivatives from Eq. (9). It is, therefore, important to have a criterion by which one can confidently decide when to terminate the expansion of the active subspace. In previous work, we relied on two approaches: (a) a pure engineering judgment-type approach in which numerical experiments are conducted to prove that an appropriate size for the active subspace has been reached [22]; and, more recently, on (b) a pure mathematical approach that rigorously derives an upper bound on the error using elements from random matrix theory [24]. In this paper, we develop another approach that employs a rigorous order statistics-type approach but has a physical meaning and, therefore, can be related to engineering quantities and tolerances that are relevant to the design.

In this approach the user needs to define a tolerance interval for the responses of interest and a probability for meeting this tolerance. The tolerance interval quantifies an upper bound on the discrepancy between the ROM and original model's predictions. An order-statistics-type approach is used to provide a measure of confidence for the accuracy of the ROM model. The Wilks' formula [26, 27] is used to determine the minimum number of samples required such that the discrepancies between the ROM and original model's predictions lie within the tolerance interval with a user-defined probability. For example, for a tolerance/probability pair of 5%/0.99, there is a 1% chance that the ROM predictions will be more than 5% discrepant from the original model's predictions. Therefore, in general for a given pair (ε, p), where $[-\varepsilon, \varepsilon]$ is the tolerance interval and p is the probability, there is a $1-p$ chance that the discrepancy between the ROM and original model's predictions will fall outside $[-\varepsilon, \varepsilon]$. In this context, a ROM model is judged inaccurate if it is executed **K** times (**K** determined from Wilks' formula) and in **M** of the executions, the discrepancies between the original model's and surrogate's predictions are outside the tolerance interval, such that $\mathbf{M}/\mathbf{K} > 1-p$.

Now assume that estimates for the active subspace $\mathbb{Z}_{\text{A}}^{\text{est}}$ and its dimension $\ell_{\text{A}}^{\text{est}}$ are available, where $\mathbb{Z}_{\text{A}}^{\text{est}}$ is given by $\mathbb{Z}_{\text{A}}^{\text{est}} = \text{R}(\mathbf{Q})$. To update these estimates, execute Eq. (9) **K** more times and determine the projection of the new **K** gradients $\left\{\vec{\nabla}_{\ell_{\text{A}}+q}\Theta\left(\vec{x}\right)\right\}_{q=1}^{\text{K}}$ onto $\mathbb{Z}_{\text{A}}^{\text{est}}$. The **K** residuals (i.e., the components of the **K** gradients that do not belong to $\mathbb{Z}_{\text{A}}^{\text{est}}$) are given by

$$\vec{\nabla}_q^{\text{res}} = \left(\mathbf{I} - \mathbf{Q}\mathbf{Q}^T\right)\vec{\nabla}_{\ell_{\text{A}}+q} f\left(\vec{x}\right) \quad q = 1, 2, ..., \mathbf{K}$$

Now, execute the forward model in Eq. (2) **K** times and evaluate the metric κ as follows:

$$\kappa_j(\ell_{\text{A}}^{\text{est}}) = \frac{\Theta\left(\vec{x}_0 + \sum_{q=1}^{\mathbf{K}} \eta_{q,j}\vec{\nabla}_q^{\text{res}}\right) - \Theta\left(\vec{x}_0\right)}{\Theta\left(\vec{x}_0 + \sum_{q=1}^{\mathbf{K}} \eta_{q,j}\vec{\nabla}_{\ell_{\text{A}}+q}\Theta\left(\vec{x}\right)\right) - \Theta\left(\vec{x}_0\right)} \tag{20}$$

where $\eta_{q,j}$ are randomly generated from standard normal distributions. The denominator is the change in the response due to a random linear combination of the additional $\left\{\vec{\nabla}_{\ell_{\text{A}}+q}\Theta\left(\vec{x}\right)\right\}_{q=1}^{\mathbf{K}}$ gradient vectors and the numerator is

the change in the response due to the components of the gradient that are orthogonal to the active subspace; these components are not accounted for by the ROM model.

After evaluating the κ metric in Eq. (20), record the number of times **M** in which the discrepancies between the exact and approximate model are greater than the user-defined tolerance. If $\mathbf{M}/\mathbf{K} > 1 - p$, increase $\ell_{\mathrm{A}}^{\mathrm{est}}$, and repeat the above procedure.

The κ metric can be calculated for an important design metric; e.g., maximum fuel temperature, critical heat flux, etc. Designers could set tolerances on these quantities easily because they have physical meanings.

7. CASE STUDIES

Several case studies were employed to demonstrate the various developments in this paper; some of them are replicated here from previous publications and some are new. The models employed in these case studies are based on neutron transport inside nuclear reactor cores. Ref. [28] may be consulted for an excellent overview of nuclear reactor calculations.

7.1 Case Study 1

This test case performs reduction into a realistic radiation transport model employed in nuclear reactor design calculations. The objective is to compare response variations calculated by the ROM model to those calculated by the original model. In addition, the computational cost required to perform the reduction is compared to both variational and forward sampling methods.

For this model, the input parameters are represented by the nuclear cross sections, which are experimentally evaluated and are used to characterize the probability of interaction between the radiation and the medium nuclei. For realistic calculations, the cross sections number in the thousands to millions depending on the application and the level of details required for the analysis. The radiation transport model is described by a generalized eigenvalue problem, where the first operator "**L**" characterizes the loss of neutrons via absorption and leakage and the other operator "**F**" describes the production of neutron via fission. In a critical nuclear reactor, the two terms must be equal, which is measured by the eigenvalue λ; i.e., it should be equal to 1. The model may be described as follows:

$$\mathbf{L}\vec{\varphi} = \lambda \mathbf{F}\vec{\varphi}$$

The responses include the critical eigenvalue or its reciprocal, referred to as $k_{\mathrm{eff}} = 1/\lambda$, and the various reaction rate densities, which are integral quantities involving the flux solution. For this case study, we consider both the k_{eff} and the flux $\vec{\varphi}$ as responses.

Employed for this case study is the TSUNAMI-2D [29], a SCALE control module that facilitates the application of sensitivity and uncertainty analysis to criticality safety models. It provides the relative sensitivity coefficients $((x_i/R)\ (\partial R/\partial x_i))$, where x_i denotes a cross section and R is a user-defined general response. In our notation, the input parameters are given by a vector of length n, such that: $\vec{x} = [x_1\ \ \ldots\ \ x_n]^T$. Cross sections are basic nuclear constants that characterize the probability of neutron interactions with matter. They have a very complicated dependence on the neutron energy. In engineering calculations, they are often represented in a piecewise format, referred to as a group format. An energy group is an energy interval over which the cross section is assumed constant. To properly average cross sections, a weighting function is used that approximates the neutron distribution in the system, which is the solution sought by the analysis. This represents one of the major dilemmas of reactor calculations. For the sake of our discussion, we assume cross sections are available in some energy group structure. The solution representing neutron density in energy and space could be used to further average (i.e., collapse) cross sections into a coarser energy structure. This process is referred to as homogenization or energy and spatial collapsing. This process is needed because the dimensionality of the neutron transport problem with all the cross-section energy and spatial details results in a problem that is computationally intractable. It is standard practice in reactor calculations to collapse cross sections over multiple steps; i.e., from the continuous format to a many-group format, then to an intermediate energy group format, then to a coarse energy group format. At each level, the reduced energy details enable one to analyze problems with more spatial heterogeneity.

The first case study employs a stand-alone single assembly mini-core model designed by OECD/NEA that is representative of light water reactor models (see Fig. 1) [30]. The 44 energy group library (v5-44) of SCALE is used and the reference (unperturbed) k_{eff} of the model is 1.09164721. The k_{eff} is the eigenvalue of the system; it is an important factor in determining criticality of the chain reaction.

Large cross-section perturbations are employed to demonstrate the nonlinear behavior. The fission cross sections for four isotopes (U-234, U-235, U-236, and U-238) in nine mixtures in the assembly model are selected as input parameters. Each cross section has 44 values corresponding to a 44 group energy structure. A total of 1584 cross sections are perturbed randomly (sampled from uniform distribution within the $\pm 10\%$ range) to generate the active subspace.

A second-order surrogate model is employed to approximate the relationship between responses and the input parameters as constrained to the active subspace; for more details see Ref. [31]. The algorithm in Section 4.2.2 is employed to construct the surrogate. First, the input parameters are perturbed and the derivatives are evaluated at random points in the input parameter space to construct matrix **G**, whose QR decomposition reveals the active subspace. As explained before, the rank is identified using the RIA algorithm, which requires a user-defined tolerance. Several tolerances are employed to investigate the effect on the accuracy of the ROM model. Once the active subspace is determined, a second-order model is assumed that relates the responses and the input parameters as confined to the active subspace. The second-order derivatives in this approximation are often described in a compact form using the so-called Hessian matrix. The ROM model may be described as follows for a general response:

$$\Theta\left(\vec{x}_0 + \Delta\vec{x}\right) \simeq \Theta\left(\vec{x}_0\right) + \vec{g}^T \Delta\vec{x} + \frac{1}{2}\Delta\vec{x}^T \mathbf{H}\Delta\vec{x} \tag{21}$$

where $\vec{x}_0$ denotes the reference cross sections; $\Delta\vec{x}$ is a general perturbation in the cross sections; $\vec{g}$ is the vector of the first-order derivatives evaluated at reference cross sections; and **H** is a matrix of the second-order derivatives, often referred to as the Hessian matrix. The reduced system of equations can be written as

$$\Theta\left(\vec{x}_0 + \Delta\vec{x}\right) \simeq \Theta\left(\vec{x}_0\right) + \left(\mathbf{Q}^T\vec{g}\right)^T \left(\mathbf{Q}^T\Delta\vec{x}\right) + \frac{1}{2}\left(\mathbf{Q}^T\Delta\vec{x}\right)^T \mathbf{Q}^T\mathbf{H}\mathbf{Q}\left(\mathbf{Q}^T\Delta\vec{x}\right) \tag{22}$$

or

$$\Theta\left(\vec{x}_0 + \mathbf{Q}\vec{\alpha}\right) \simeq \Theta\left(\vec{x}_0\right) + \vec{g}_\alpha^T\vec{\alpha} + \frac{1}{2}\vec{\alpha}^T\mathbf{H}_\alpha\vec{\alpha} \tag{23}$$

where $\vec{\alpha} \in \mathrm{R}^r$ is a vector of r reduced variables. Figure 2 shows the singular values of the **G** matrix (step 5 in the algorithm in Section 4.2.2) that describes the active subspace. The RIA algorithm was used to determine the

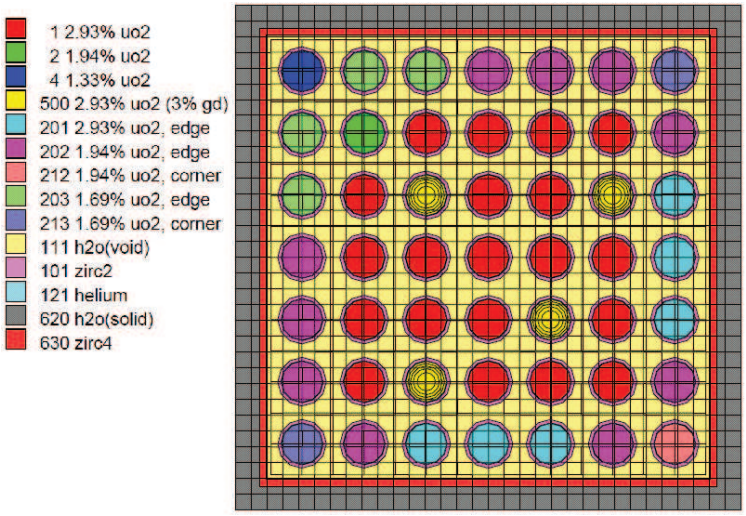


FIG. 1: A 7×7 BWR benchmark assembly model.

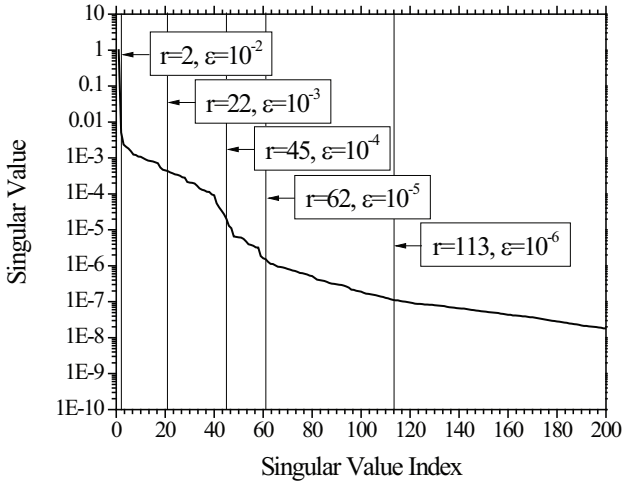


FIG. 2: Singular values of the **G** matrix.

corresponding rank required to meet specified user-defined tolerance. Figure 3 shows the ranks determined for a range of accuracy requirements. In all these graphs, the number of adjoint model executions is proportional to the rank of matrix **G**; i.e., the size of the active subspace.

To verify the accuracy of the predicted ROM, forward calculations are compared to those predicted by Eq. (21) for a random perturbation of all cross sections. This numerical experiment is repeated with varying degrees of perturbations to assess the quality of the ROM. Since all cross sections are perturbed simultaneously, a root-mean-square (RMS) average of all perturbations is employed in Fig. 3 to describe the size of cross-section perturbations.

7.2 Case Study 2

In this case study, we emulate a two-component model based on the previous assembly model. The overall model homogenizes the 44 group cross sections into five energy groups, in which the number of responses is 20; i.e., $\vec{y} \in \mathrm{R}^{20}$. In the first component, the many group cross sections in the 44 energy group format are collapsed into a 22 energy

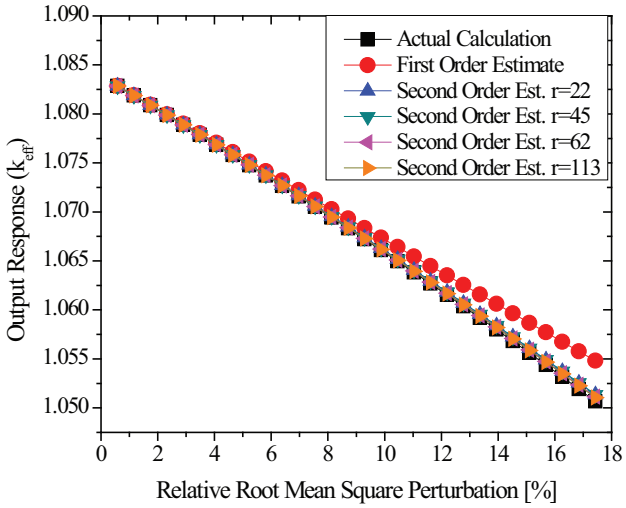


FIG. 3: Accuracy of the ROM model versus the size of the active subspace.

group format, and the second-component model further collapses the 22 group cross sections into a five energy group format. In each component, a radiation transport model is solved for the neutron flux, which is used to collapse the cross sections. Calculations are completed using the TSUNAMI-2D module of the SCALE 6.1 [29] and the sensitivity profiles for all responses calculated by the new multi-component SA approach are compared to those computed by the standard generalized perturbation theory (GPT) "adjoint" approach [3]. As representative results, the sensitivity profiles by the traditional GPT approach and new approach are presented in Fig. 4. Results indicate that the multi-component SA (h-MCSA) employing the pseudo-response approach accurately estimates the first-order derivatives.

7.3 Case Study 3

This case study is analyzed using the TSUNAMI-1D code [32]. The model is based on a fast unreflected critical sphere—JEZEBEL. The JEZEBEL model (PU-MET-FAST-001) is a critical homogenized unreflected metal sphere from the criticality safety benchmark experiments series [33]. Reaction rate densities are calculated in the sphere fuel volume. TSUNAMI-1D calculates the derivatives of the k_{eff} with respect to the cross sections. The active subspace for the cross sections has been determined using the algorithm in Section 4.2.2 and the κ metric for $\varepsilon = 0.95$ and $p = 0.95$ assuming different sizes for the active subspace. The graph in Fig. 5(a) shows that an estimate of $\ell_{\text{A}}^{\text{est}} = 50$

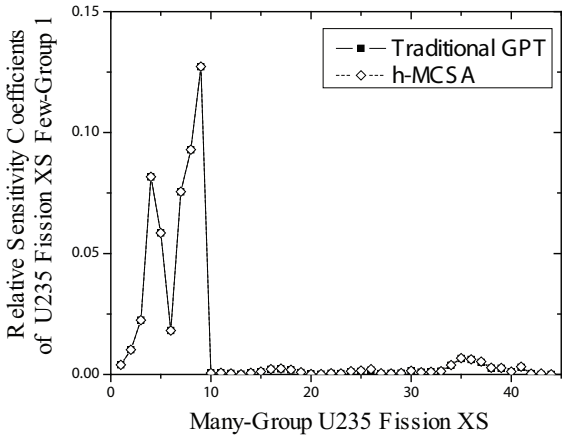
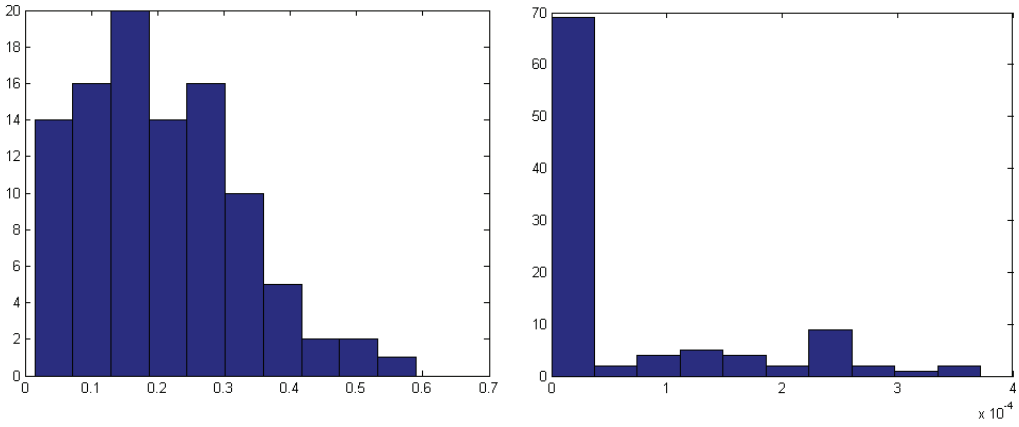


FIG. 4: Comparison of the traditional and new h-MCSA approaches.



(a) (b)

FIG. 5: Graphs of the κ-metric: (a) $\ell_{\text{A}}^{\text{est}} = 50$ and (b) $\ell_{\text{A}}^{\text{est}} = 200$.

is poor and an estimate of $\ell_{\mathrm{A}}^{\mathrm{est}} = 200$ is over-conservative. For more experiments, the reader is encouraged to consult Ref. [34].

8. CONCLUSIONS AND FUTURE WORK

A hybrid methodology has been developed for order reductions for nonlinear models with many input parameters and output responses with quantifiable error bounds. The method has been extended to multi-component models that are connected in a serial manner, where the output of one component is passed as input to the next. The methodology hybridizes global and local SA methods to search for an active subspace that can be sampled exhaustively using global sampling methods. The latest advances in matrix theory have been leveraged to establish quantifiable error bounds on the maximum discrepancy between the reduced and original models. Additionally, an order statistics error criterion is also developed, which is more relevant to engineering calculations than the pure mathematical error bounds. Moreover, via the introduction of pseudo-responses, the method is able to transfer efficiently sensitivity information between components. The result is a hybrid approach that combines the benefits of the state-of-the-art methods in order reduction, thereby enabling application to real-world problems.

Two challenges describe our future research focus: extending applicability of the approach to ill-behaved models and to multi-component models with complex connectivity patterns. For ill-behaved models such as discontinuous bifurcated models, the infinite series expression employed in our work will not work. It needs to be extended to describe non-smooth functions. We believe that via an appropriate choice of ψ functions, a wider class of non-smooth functions could be handled; however, this issue remains to be explored. Additionally, our current development has focused on demonstrating a proof-of-principle-type approach to a simple connectivity pattern for the components. The idea was to show that one needs to track information that belongs to the active subspaces only at each component-to-component interface. We believe more research is needed to extend this idea to models with complex connectivity.

9. ACKNOWLEDGMENTS

This work has been supported by the DOE Nuclear Energy Advanced Modeling and Simulation (NEAMS) program; and the Consortium for Advanced Simulation of Light Water Reactors (www.casl.gov), an Energy Innovation Hub (http://www.energy.gov/hubs) for Modeling and Simulation of Nuclear Reactors under U.S. Department of Energy Contract No. DE-AC05-00OR22725. Special thanks are due to Dr. Ralph Nelson for supporting this work under the NEAMS grant and to Dr. James Stewart and Dr. Brian Adams for hosting a visit for H. Abdel-Khalik at SNL.

REFERENCES

1. Schilders, W. H., van der Vorst, H. A., and Rommes, J., eds., *Model Order Reduction: Theory, Research Aspects and Applications*, Springer, Berlin, 2008.
2. Bang, Y., Hite, J. M., and Abdel-Khalik, H. S., Hybrid reduced order modeling applied to nonlinear models, *Int. J. Numer. Meth. Eng*., DOI: http://dx.doi.org/10.1002/nme.4298, 2012.
3. Abdel-Khalik, H. S., Nelson, R., and Adams, B. M., Hybrid uncertainty and sensitivity algorithms for high dimensional nonlinear models, *Trans. Am. Nucl. Soc*., 103:375–377, 2010.
4. Cacuci, D. G., *Sensitivity and Uncertainty Analysis*, Chapman & Hall/CRC, Boca Raton, FL, 2003.
5. Adams, B. M., Bohnhoff, W. J., Dalbey, K. R., Eddy, J. P., Eldred, M. S., Gay, D. M., Haskell, K., Hough, P. D., and Swiler, L. P., DAKOTA: A multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 5.0 user's manual, *Technical Report SAND2010-2183*, Sandia National Laboratories, 2009.
6. Saltelli, A., Chan, K., and Scott, E. M., *Sensitivity Analysis*, Wiley, New York, 2000.
7. Ghanem, R. G. and Spanos, P. D., *Stochastic Finite Elements: A Spectral Approach*, Courier Dover, New York, 1991.
8. Xiu, D., *Numerical Methods for Stochastic Computations: A Spectral Method Approach*, Princeton University Press, Princeton, NJ, 2010.

9. Abdel-Khalik, H. S., On nonlinear reduced order modeling, *Proc. of International Conference on Mathematics and Computations in Nuclear Science and Engineering*, Brazil, 2011.
10. Box, G. E. P. and Draper, N. R., *Response Surfaces, Mixtures, and Ridge Analyses*, 2nd ed., Wiley, Hoboken, NJ, 2007.
11. Lighthill, M. J., *Introduction to Fourier Analysis and Generalised Functions*, Cambridge University Press, Cambridge, UK, 1958.
12. Friedman, J., Multivariate adaptive regression splines, *Ann. Stat*., 19(1):1–67, 1991.
13. Gallivan, K., Grimme, E., and Van Dooren, P., Pade Approximation of large-scale dynamic systems with Lanczos methods, *Proc. of 33rd IEEE Conference on Decision and Control*, 1994.
14. Penzl, T, Algorithms for model reduction of large dynamical systems, *Linear Algebr. Appl*., 415(2-3):322–343, 2006.
15. Holmes, P., Lumley, J., and Berkooz, G., *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge Monographs on Mechanics, Cambridge University Press, Cambridge, UK, 1996.
16. Carlberg, K., Bou-Mosleh, C., and Farhat, C., Efficient non-linear model reduction via a least-squares Petrov–Galerkin projection and compressive tensor approximation, *Int. J. Numer. Methods Eng*., 86(2):155–181, 2011.
17. Chaturantabut, S., and Sorensen, D. C., Nonlinear model reduction via discrete empirical interpolation, *SIAM J. Sci. Comput*., 32(5):2737–2764, 2010.
18. Sirovich, L., Turbulence and the dynamics of coherent structure. Part 1: Coherent structures, *Q. Appl. Math*., 45(3):561–571, 1987.
19. Antoulas, A. C., Sorensen, D. C., and Gugercin, S., A survey of model reduction methods for large-scale systems, *J. Contemp. Math*., 280:193–219, 2001.
20. Gandini, A., Generalized perturbation theory for nonlinear systems for the importance conservation principle, *Nucl. Sci. Eng*., 77:316, 1981.
21. Wang, C. and Abdel-Khalik, H. S., Exact-to-precision perturbation theory for source-driven problems, *Nucl. Eng. Des*., 241(12):5104–5112, 2011.
22. Abdel-Khalik, H. S., Adaptive core simulation, *Ph.D. Dissertation*, North Carolina State University, Raleigh, NC, 2004.
23. Golub, G. H. and Van Loan, C. F., *Matrix Computations*, 3rd ed., John Hopkins, Baltimore, MD, 1996.
24. Liberty, E., Woolfe, F. F., Martinsson, P.-G., Rokhlin, V., and Tygert, M., Randomized algorithms for the low-rank approximation of matrices, *Proc. Natl. Acad. Sci. U.S.A*., 104(51):20167–20172, 2007.
25. Abdel-Khalik, H. S., Adjoint-based sensitivity analysis for multi-component models, *Nucl. Eng. Des*., http://dx.doi.org/10.1016/j.nucengdes.2012.01.017, 2012.
26. Wilks, S. S., Statistical prediction with special reference to the problem of tolerance limits, *Ann. Math. Stat*., 13(4):400–409, 1942.
27. Kolowrocki, K., ed., *Advances in Safety and Reliability: Proceedings of the European Safety and Reliability Conference (ESREL 2005)*, pp. 1692–1696, Taylor & Francis, London, UK, 2005.
28. Williams, M. M. R., ed., *Handbook of Nuclear Reactor Calculations*, CRC Press, Boca Raton, FL, 1988.
29. Rearden, B. T., TSUNAMI-2D: Control module for one-dimensional cross-section sensitivity and uncertainty analysis for criticality, version 6, *ORNL/TM-2005/39*, 2009.
30. Ivanov, K., Avramova, M., Kodeli, I. and Sartori, E., *Benchmark for Uncertainty Analysis in Modeling (UAM) for Design, Operation and Safety Analysis of LWRs*, vol. I, OECD Nuclear Energy Agency, Moulineaux, France, 2007.
31. Bang, Y. S. and Abdel-Khalik, H. S., An efficient reduced-order method for Hessian matrix construction, *Proc. of International Conference on Mathematics and Computational Methods Applied in Nuclear Science and Engineering*, Rio de Janeiro, Brazil, 2011.
32. Readen, R., TSUNAMI-1D: Control module for one-dimensional cross-section sensitivity and uncertainty analysis for criticality, version 6, ORNL/TM-2005/39, 2009.
33. International Criticality Safety Benchmark Evaluation Project (ICSBEP), *International Handbook of Evaluated Criticality Safety Benchmark Experiments*, September 2006 ed., OECD-NEA, 2006.
34. Kennedy, C. and Abdel-Khalik, H., GPT-free sensitivity analysis for eigenvalue problems, *Nucl. Technol*., in press, 2012.

APPENDIX A

We wish to show that $\prod_{i=-1}^{k}\left(\vec{\beta}_i^{\,T}\,\vec{x}\right)$ is a polynomial of the same form as the kth term of the Taylor series. First, we must introduce some notation. Throughout, we take $\vec{x},\vec{\beta}_i \in \mathbb{R}^n$ for $i = 1, ..., k$. Let $a \circ b$ denote the Schur product (also known as the Hadamard product) of a and b and $a \otimes b$ denote the Kronecker product. Also, denote:

$$B = \vec{\beta}_1 \otimes \vec{\beta}_2 \otimes \cdots \otimes \vec{\beta}_k$$

$$X = \vec{x} \otimes \vec{x} \otimes \cdots \otimes \vec{x} (k \text{ times})$$

We will adopt a notation common in the analysis community and define a *multi-index*, α, as an n-tuple of natural numbers. For a vector $\vec{x} \in \mathbb{R}^n$, we define multi-index operations as follows:

$$|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_n$$

$$\vec{x}^{\,\alpha} = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$$

$$\alpha! = \alpha_1!\alpha_2! \cdots \alpha_n!$$

$$D^{\alpha} = \left(\frac{\partial}{\partial x_1}\right)^{\alpha_1}\left(\frac{\partial}{\partial x_2}\right)^{\alpha_2} \cdots \left(\frac{\partial}{\partial x_n}\right)^{\alpha_n}$$

Using this notation, we can express Taylor's theorem for a single-valued function about a point $\vec{x}_0$ compactly:

$$f\left(\vec{x}\right) = \sum_{|x|\geq 0} \frac{D^{\alpha} f\left(\vec{x}_0\right)\left(\vec{x} - \vec{x}_0\right)^{\alpha}}{\alpha!} \tag{A1}$$

where the kth term of Eq. (A2) is

$$f\left(\vec{x}\right) = \sum_{|\alpha|=k} \frac{D^{\alpha} f\left(\vec{x}_0\right)\left(\vec{x} - \vec{x}_0\right)^{\alpha}}{\alpha!} \tag{A2}$$

The multinomial theorem can also be expressed as follows:

$$\left(x_1 + x_2 + \cdots + x_k\right)^m = \sum_{|\alpha|=m} \frac{m!}{\alpha!}\vec{x}^{\,\alpha} \tag{A3}$$

Note that $\left(\vec{\beta}_i^{\,T}\,\vec{x}\right) = \vec{1}_n^{\,T}\left(\vec{\beta}_i \circ \vec{x}\right)$, where $\vec{1}_n$ is a vector of dimension n where all entries are equal to 1. Applying the associative property of scalar multiplication, we can rewrite as follows:

$$\prod_{i=1}^{k}\left(\vec{\beta}_i^{\,T}\,\vec{x}\right) = \left\{\vec{1}_n^{\,T}\left(\vec{\beta}_1 \circ \vec{x}\right)\right\}\left\{\vec{1}_n^{\,T}\left(\vec{\beta}_2 \circ \vec{x}\right)\right\} \cdots \left\{\vec{1}_n^{\,T}\left(\vec{\beta}_n \circ \vec{x}\right)\right\} \; = \vec{1}_n \left(\mathbf{B} \circ \mathbf{X}\right) \vec{1}_{n^{k-1}} \tag{A4}$$

where $\mathbf{X}$ and $\mathbf{B}$ are $n^{k-1} \times n$ matrices that contain every combination of the entries in $\vec{x}$ and $\vec{\beta}_i$ such that the total order is n. Note that Eq. (A3) implies that Eq. (A2) is a polynomial with terms that are all possible combinations of entries in $\vec{x}$ that have total order n multiplied by constants and Eq. (A4) is simply the sum of entries in $(\mathbf{B} \circ \mathbf{X})$; thus, we conclude that Eqs. (A2) and (A4) are polynomials of the same form.

APPENDIX B

In this appendix, we explain how to construct a low rank approximation for a matrix using only matrix–vector products. Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ be a given matrix, available via matrix–vector products only; i.e., $\vec{q} = \mathbf{A}\vec{p}$ and $\vec{p} = \mathbf{A}^T\vec{q}$ where $\vec{p} \in \mathbb{R}^n$ and $\vec{q} \in \mathbb{R}^m$. In some applications, matrix $\mathbf{A}$ may not be available in a transpose form; however, another matrix $\mathbf{B}$ may be available such that $\mathrm{R}(\mathbf{B}^T) = \mathrm{R}(\mathbf{A}^T)$. This occurs when matrix $\mathbf{A}$ is a product of several matrices; say, $\mathbf{A} = \mathbf{CDB}$, where both $\mathbf{C}$ and $\mathbf{D}$ are full rank. Then one can show that $\mathrm{R}(\mathbf{B}^T) = \mathrm{R}(\mathbf{A}^T)$. In this case, one could use $\mathbf{B}^T$ instead of $\mathbf{A}^T$ in the algorithm below.

Let the sought low rank approximation be described by a singular value decomposition:

$$\mathbf{A} = \mathbf{USV}^T$$

where $\mathbf{U} \in \mathbb{R}^{m\times r}$ is an orthonormal matrix whose columns approximate the range of the matrix, $\mathrm{R}(\mathbf{A})$; the $\mathbf{V} \in \mathbb{R}^{n\times r}$ is also an orthonormal matrix that spans the range of the matrix–transpose, $\mathrm{R}(\mathbf{A}^T)$; and the $\mathbf{S} \in \mathbb{R}^{r\times r}$ is a diagonal matrix that approximates the singular values of matrix $\mathbf{A}$.

The algorithm proceeds as follows:

1. Pick an estimate for the rank r.
2. Generate a random matrix: $\mathbf{Z} \in \mathbb{R}^{m\times r}$.
3. Calculate $\mathbf{A}^T\mathbf{Z}$ and perform QR such that: $\mathbf{A}^T\mathbf{Z} = \mathbf{QR}$.
4. Employing the RIA algorithm, determine whether the rank r satisfies the user-defined error tolerance. If not, go back to step 1 and increase r. Note that the $\mathbf{Q}$ matrix has r columns that span $\mathrm{R}(\mathbf{A}^T)$.
5. Generate a matrix $\mathbf{X} = \mathbf{QP} \in \mathbb{R}^{n\times r}$ such that $\mathbf{P} \in \mathbb{R}^{r\times r}$ is randomly generated. This step guarantees that the columns of $\mathbf{X}$ live in $\mathrm{R}(\mathbf{A}^T)$.
6. Calculate $\mathbf{Y} = \mathbf{AX}$.
7. Calculate the SVD: $\mathbf{YP}^{-1} = \mathbf{USW}^T$. This step determines the $\mathbf{U}$ and $\mathbf{S}$ matrices.
8. Evaluate: $\mathbf{V} = \mathbf{QW}$.
9. END.

Several variations of this algorithm are possible and have appeared elsewhere; see Ref. [24], and the references cited within. The basic idea is to find the ranges for $\mathrm{R}(\mathbf{A}^T)$ and $\mathrm{R}(\mathbf{A})$ via random matrix–vector products, which are both expected to have a size r. Because random products are employed, one can show with high probability that the vectors resulting from the matrix–vector products are also random with high probability. This implies that one needs at most r matrix–vector products to find the respective spaces. In practice, one performs few additional matrix–vector products to ensure the user-defined tolerance is met as described earlier in the RIA algorithm.