TU Delft

Delft University of Technology

Addressing Ambiguities in Constrained Sensitivity Analysis for Reactor Physics Problems

Seo, Jeongwon; Abdel-Khalik, Hany; Perko, Zoltan

DOI
10.1080/00295450.2020.1721407

Publication date
2020

Document Version
Final published version

Published in
Nuclear Technology

Citation (APA)
Seo, J., Abdel-Khalik, H., & Perko, Z. (2020). Addressing Ambiguities in Constrained Sensitivity Analysis for Reactor Physics Problems. *Nuclear Technology*, *206*(12), 1827-1839. https://doi.org/10.1080/00295450.2020.1721407

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' – Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Nuclear Technology

ISSN: 0029-5450 (Print) 1943-7471 (Online) Journal homepage: https://ans.tandfonline.com/loi/unct20

Addressing Ambiguities in Constrained Sensitivity Analysis for Reactor Physics Problems

Jeongwon Seo, Hany Abdel-Khalik & Zoltan Perko

To cite this article: Jeongwon Seo, Hany Abdel-Khalik & Zoltan Perko (2020): Addressing Ambiguities in Constrained Sensitivity Analysis for Reactor Physics Problems, Nuclear Technology, DOI: 10.1080/00295450.2020.1721407

To link to this article: https://doi.org/10.1080/00295450.2020.1721407

Published online: 12 Mar 2020.

Submit your article to this journal

Article views: 25

View related articles

View Crossmark data

Full Terms & Conditions of access and use can be found at
https://ans.tandfonline.com/action/journalInformation?journalCode=unct20

NUCLEAR TECHNOLOGY
© 2020 American Nuclear Society
DOI: https://doi.org/10.1080/00295450.2020.1721407

Check for updates

Addressing Ambiguities in Constrained Sensitivity Analysis for Reactor Physics Problems

Jeongwon Seo,[a] Hany Abdel-Khalik,[a]* and Zoltan Perko[b]

[a]*Purdue University, School of Nuclear Engineering, Flex lab, 205 Gates Rd, West Lafayette, Indiana 47096*
[b]*Delft University of Technology, Department of Radiation, Science and Technology, Section Nuclear Energy and Radiation Applications, Mekelweg 15, Delft, Netherlands*

Received October 30, 2019
Accepted for Publication January 22, 2020

Abstract — *This paper presents an algorithm for completing sensitivity analysis that respects linear constraints placed on the associated model's input parameters. Any sensitivity analysis (linear or nonlinear, local or global) focuses on measuring the impact of input parameter variations on model responses of interest, which may require the analyst to execute the model numerous times with different model parameter perturbations. With the constraints present, the degrees of freedom available for input parameter variations are reduced, and hence any analysis that changes model parameters must respect these constraints. Focusing here on linear constraints, earlier work has shown that constraints may be respected in many ways, causing ambiguities, i.e., nonuniqueness, in the results of a sensitivity analysis, forcing the analyst to introduce dependencies with downstream analyses, e.g., uncertainty quantification, that employ the sensitivity analysis results. This paper develops the theoretical details for a new algorithm to select model parameter variations that automatically satisfy linear constraints resulting in unique results for the sensitivity analysis, thereby removing any custom dependencies with downstream analyses. To demonstrate the performance of the algorithm, it is applied to solve the multigroup eigenvalue problem for the multiplication factor in a representative CANDU core-wide model. The model parameters analyzed are the group prompt neutron fractions, whose summation must be equal to one over all energy groups. The results indicate that the new algorithm identifies the gradient direction uniquely which represents the direction of maximum change while satisfying the constraints, thus removing any ambiguities resulting from the constraints as identified by earlier work.*

Keywords — *Sensitivity analysis, linear constraint, multigroup eigenvalue problem, CANDU core model.*

Note — *Some figures may be in color only in the electronic version.*

I. INTRODUCTION

Sensitivity analysis is an important analysis in many engineering disciplines as it allows analysts to quantify as well as prioritize the impact of various model parameters on a model's responses of interest.[1] The results of sensitivity analyses are important to a wide range of engineering-oriented studies such as model optimization, uncertainty propagation, inference analysis, and experimental design, etc. The results of a sensitivity analysis are a set of scalar measures, each associated with a given parameter (or a group of parameters) that describes the parameter(s) impact on the response of interest. Depending on the sensitivity analysis approach, the associated measures can be grouped in various formats, e.g., vectors, matrices, tensors, multidimensional tables, etc.

The literature is replete with many approaches for sensitivity analyses that can be classified in many different ways, e.g., deterministic versus stochastic, local versus global, forward versus adjoint, etc.[2–6] Common to all approaches is the

*E-mail: abdelkhalik@purdue.edu

need to prepare multiple sets of model parameter perturbations, e.g., whereby each set perturbs a single parameter, a group of parameters, or all parameters using deterministic or randomized strategies. For forward-based sensitivity analyses,[1] the perturbed sets of model parameters are employed to execute the model multiple times with the response variations recorded for later extraction of sensitivity measures. For adjoint-based sensitivity analyses,[5,6] the perturbations are employed in inner-product-like formulas—thereby precluding the need for model re-execution—containing both the forward and adjoint solutions to estimate the sensitivity measures.

Mathematically, a general model that reads n input parameters and calculates a given response y may be described by

$$y = f(x) = f\left([x]_1, [x]_2, \ldots, [x]_n\right) , \tag{1}$$

where $x \in \mathbb{R}^n$ is a vector that lives in an n dimensional space, such that its i'th component $[x]_i$ represents the i'th model input parameter, i.e.,

$$x = [\,[x]_1 \quad [x]_2 \quad \ldots \quad [x]_n\,]^T ,$$

where the y represents the model response of interest, which can be a vector as well, however, the discussion here is limited to a single response with the generalization being a straightforward step.

The distribution of the inputs, fully described by their joint probability density function, is crucial for accurate sensitivity and uncertainty analysis since the output variability directly depends on it. In most practical applications, however, only very limited information is available about the input uncertainties. Very often complete independence is assumed and all variables are considered Gaussian. For certain parameters in reactor physics, this is a reasonable assumption, e.g., fuel rod dimension uncertainties due to manufacturing tolerances can safely be assumed to be independent from the uncertainties of thermal conductivity values. For other parameters, however, independence is obviously not true. It is well established, for example, that group cross sections are heavily correlated, thus in any accurate uncertainty analysis these correlations must be accounted for. To accommodate this, cross-section data libraries typically contain cross-section values along with their covariance matrix, which then can be used with standard sampling techniques to generate random samples of the cross sections respecting their correlations, to be used subsequently in uncertainty propagation.

The marginal distribution of inputs and their correlations, however, do not provide all information about the joint probability density function. Moreover, while correlations are available for certain inputs, for several clearly correlated parameters they are not. Constrained inputs often fall into this category, where the constraint itself is well known, but the correlation is not. In reactor physics several such situations arise. The most well-known and studied example is the energy spectrum of fission and delayed neutrons, where the constraint is that they have to sum up to unity, but their exact correlations are unavailable.[7] Isotope abundances of a certain element[7,8] represent the same issue, as do most distribution-type quantities, such as energy and angle distributions, fuel-channelwise power distribution in a core,[9] or the volume, mass, and density of a certain region.

These problems have three main issues in common. First, for accurate uncertainty analysis and safety assessment it is clearly critical to correctly take these correlations into account. Second, most often only the marginal distributions are known, but not the correlations. Last, contrary to the correlations, the constraints are typically well known and are available as relatively simple analytical expressions. Since the constraints induce correlations between the inputs that must be taken into account, a unified, unambiguous methodology to handle such situations is clearly of great interest, providing the main motivation for our research.

The constraints are described by a set of linear equations as follows:

$$\mathbf{A}x = b\,, \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{m \times n}$ is a rectangular-wide matrix, such that m, the number of constraints, is less than n. The goal is to design model parameter perturbations Δx that can be used to complete the sensitivity analysis while satisfying the constraints, i.e.,

$$\mathbf{A}(x + \Delta x) = b \rightarrow \mathbf{A}\Delta x = 0\,. \tag{3}$$

Equation (3) implies that all model parameter perturbations that satisfy the constraints must be in the null space of the matrix $\mathbf{A}$. In other words, all perturbations must be orthogonal to the row space of the matrix $\mathbf{A}$. The row space of the matrix $\mathbf{A}$ is spanned by its m rows, meaning that the null space of $\mathbf{A}$ has n-m degrees of freedom. This further implies that the parameter perturbations can only have up to n-m degrees of freedom. Thus, to best cover this space with parameter perturbations in support of a sensitivity analysis, this work proposes to perturb the parameters along these n-m degrees of freedom. This can be seamlessly achieved by first finding a basis for the null space of $\mathbf{A}$, followed by perturbing the parameters along each vector in this basis.

To limit the scope of this paper, the presentation below will focus on a local deterministic sensitivity analysis. Nevertheless, the basic idea of the proposed algorithm is applicable to other sensitivity analysis approaches as long as they involve the perturbation of model parameters in search of sensitivity information. First, consider a linear sensitivity analysis where the sensitivity measures are represented by a vector of n numbers, each corresponding to one of n input model parameters and representing an estimate of the first-order partial derivative of the model response of interest with respect to the respective parameter. In a linear local sensitivity analysis, the goal is to estimate the so-called gradient vector, described by

$$g = \nabla y = \left[\frac{\partial y}{\partial [x]_1} \quad \frac{\partial y}{\partial [x]_2} \quad \cdots \quad \frac{\partial y}{\partial [x]_n} \right]^T , \tag{4}$$

where $\frac{\partial y}{\partial [x]_1}$ is the partial derivative of the response y with respect to the i'th model parameter. In this context, linearity does not imply the model is linear. Instead, it means that the estimate of the sensitivity measure is based on the linear portion of the model response variations with respect to the model parameters. In higher-order sensitivity analyses, one includes higher-order derivatives as additional measures for sensitivity. For example, in a second-order sensitivity analysis, one generally calculates, in addition to first-order derivatives, all second-order derivatives of the form:

$$\frac{\partial^2 y}{\partial [x]_i [x]_j} ,$$

where both i and j go from 1 to n, bringing the total number of sensitivity measures to

$$n + \frac{n(n+1)}{2} ,$$

with the first term representing the n first-order derivatives with respect to the n model parameters, and the second term counts all the second-order terms.

In a forward-based sensitivity analysis employing one-at-a-time perturbations, these derivatives can be calculated using finite differencing as follows[1]:

$$\frac{\partial y}{\partial [x]_i} = \frac{f([x]_1, [x]_2, .., [x]_i + \Delta_i, .., [x]_n) - f([x]_1, [x]_2, .., [x]_i, .., [x]_n)}{\Delta_i} , \quad \text{with} \quad i = 1, \ldots, n . \tag{5}$$

This implies the need for n model executions, wherein each execution perturbs a single model parameter by a preset amount Δ_i. Methods exist to select the optimum size of this perturbation to ensure it is not too small to avoid contaminating the results by numerical noise or not too large to ensure local function variations are captured.

In applying these perturbations, one must ensure that the constraints in Eqs. (2) and (3) are satisfied. This leads to an ambiguous situation because say in the first perturbation, $i = 1$, where the first parameter is perturbed and one has a single constraint as in Eq. (2), one would need to adjust the other n-1 variables to ensure the single constraint equation is satisfied.[7] This can be achieved in an infinite number of ways because a single equation with n-1 unknowns is an underdetermined problem with an infinite number of solutions.

Earlier work has identified this problem of ambiguity and has proposed a set of custom recipes to adjust the remaining parameters in order to satisfy the constraints, while also demonstrating some of the limitations of these recipes.[7] The challenge with a recipe-based approach is that each recipe will result in a different estimate of the sensitivity measures, which is likely to impact the results of any subsequent downstream analysis employing these measures. Thus, there exists a need to select parameter perturbations in some way that leads to unique results for the sensitivity measures representing the output of the sensitivity analysis.

Before discussing the details of this proposed algorithm, a general overview of sensitivity analysis approaches is given in Sec. II, highlighting their common need for a set of model parameter perturbations. Next, Sec. III explains the proposed idea in the context of a locally linear sensitivity analysis. Section IV demonstrates applicability to a representative core-wide problem. Finally, concluding remarks and future work are provided in Sec. V.

II. BACKGROUND ON SENSITIVITY ANALYSIS

Sensitivity analysis is a key mathematical analysis that supports a wide range of engineering activities, including the identification of main sources of uncertainties, selection of best strategies for minimizing uncertainties, determining the

best approaches for optimizing the system economy, and supporting safety analysis calculations.[1,3] At its core, sensitivity analysis is a mathematical approach that allows the analyst to determine how a model's calculated output results are impacted by its input parameters. When sensitivity analysis results are combined with parameter uncertainties, the analyst can determine the uncertainties of the model output results and identify key sources of uncertainties. When these results are subsequently combined with real measurements for system responses, the analyst can devise inference algorithms to improve the predictive ability of the model by adjusting model parameters to correct for the observed discrepancies between model predictions and measurements.[10]

The last two decades have witnessed a resurgence in the interest of performing sensitivity analyses and other dependent analyses,[11–14] e.g., uncertainty analysis and inference, for a wide range of existing as well as advanced reactor systems. There is a strong consensus in the nuclear community that sensitivity analyses and other dependent analyses can support many activities, including scoping-type design activities, optimization of experimental programs in support of model validation, optimizing operation leading to better system economy, and supporting the regulatory process by identifying key sources of uncertainties. Given its value to many engineering analyses, sensitivity analysis results must be calculated with care to ensure the reliability of downstream calculations.

There are two main approaches for representing the results of sensitivity analysis: derivative-based and importance-based approaches, sometimes also referred to as local and global sensitivity analysis. The local derivative-based approach employs the calculus definition of derivatives as a measure for the impact of each model parameter on the response of interest.[15] The approach is denoted "local" because the derivatives are approximated around a reference point for the parameters. The disadvantage of this approach is that if the model is nonlinear, knowing the derivatives at a given point may not provide adequate information on the behavior of the model for the range of expected parameter variations.

On the other hand, the global importance-based approach measures a parameter impact in terms of its contribution to the total variations in the response of interest.[1] The idea here is to perturb the model parameters within ranges defined by their prior uncertainties, and then to execute the model many times to estimate the variance in the responses of interest. From that, one can estimate the portion of that variation that is associated with each parameter or group of parameters. These measures do not allow one to capture the local variations in the model responses due to variations in the parameters. Instead they provide average behavior over the range of variations as determined by the prior uncertainties. The calculated sensitivity measures are generally different for both the local and global approaches. Nevertheless, they can be shown to be equal for special cases where the model is linear and all parameters are independent with equal ranges of variations.

The local sensitivity analysis approach employs two key implementation strategies, i.e., forward- and adjoint-based, whereas the global sensitivity analysis relies only the forward-based implementation; see Ref. 3 for a brief tutorial. In a forward implementation, the model is employed directly after perturbing its input parameters and re-executing the model and recording the variations in the responses of interest from some reference execution employing reference values for the model parameters. In an adjoint implementation, the analyst must develop a new model representing the mathematical dual of the forward model equations, which produces a new state variable, called the adjoint solution, as opposed to the forward solution. Both the adjoint and forward solutions can be combined together via inner product operation to estimate the derivatives of select model response with respect to its model parameters. In this inner product formulation, one has to change one parameter at a time to estimate the derivative of the response with respect to the given parameter. Thus, essentially a forward-based approach is employed to calculate the derivatives, but applied on the inner product formulation rather than the original model. The cost of re-executing the inner product formulation is far less expensive than re-execution of the original model, which renders the adjoint-based methods superior to forward-based methods for the evaluation of first-order derivatives when the number of model parameters is very high. The adjoint implementation, however, is challenged by the fact that one needs an adjoint model execution for each response of interest, which renders the forward-based method superior for models with many responses. Hybrid methods have also been developed to combine the advantages of the forward and adjoint methods and circumvent their limitations.[16]

III. MATHEMATICAL DESCRIPTION OF PROPOSED ALGORITHM

The idea behind the proposed algorithm starts with the definition of the gradient given in Eq. (4). This definition implies that one can write the gradient as follows:

$$g = \alpha_1 e_1 + \alpha_2 e_2 + \ldots + \alpha_n e_n \ ,$$

where $e_i = [0, 0, \ldots, 1, \ldots, 0] \in \mathbb{R}^n$ is a standard basis vector of all zeros, except 1 in the i'th component, and

$$\alpha_i = \frac{\partial y}{\partial [x]_i} \ ,$$

which may be rewritten as

$$\alpha_i = \frac{\partial y}{\partial [x]_i} = \frac{\partial y}{\partial e_i} = \frac{f(x + \Delta_i e_i) - f(x)}{\Delta_i} \ .$$

In this representation, the n standard vectors represent an orthonormal basis for the model parameter space, which is an n-dimensional space, and the α_i is the derivative of the function along the i'th vector of this basis, hence the notation of $\partial y / \partial e_i$, which highlights the notion of calculating the derivative along a direction. Interestingly, this represents a special case of a more general definition of the gradient of a general function[16] given by

$$g = \alpha_{q_1} q_1 + \alpha_{q_2} q_2 + \ldots + \alpha_{q_n} q_n \ , \tag{6}$$

where $\{q_1, \ldots, q_n\}$ is any set of n orthonormal vectors, and the coefficients are the directional derivatives of the function along these vectors. A directional derivative, say along the q_i direction, is described by

$$\alpha_{q_i} = \frac{\partial y}{\partial q_i} = \frac{f(x + \Delta_i q_i) - f(x)}{\Delta_{q_i}} \ . \tag{7}$$

This representation implies that the gradient, which is a unique direction in the space representing the direction of maximum change, can be calculated using any basis, as long as it is an orthonormal basis and it covers the allowable degrees of freedom for parameter variations.

To illustrate how this observation represents the basis for the proposed algorithm, a simple case is considered first. Consider a single constraint of the form: $[x]_1 = b$. This constraint implies that the first component of x is set to a fixed value b. This constraint may be written as $e_1^T x = b$, which may be interpreted as the component of x along e_1 must be fixed to the b value. In this simple case, a forward linear sensitivity analysis would simply avoid perturbing the first parameter in order to satisfy the constraint, and the resulting gradient would be

$$g_c = \nabla y = \left[0, \quad \frac{\partial y}{\partial [x]_2}, \quad \ldots \quad , \frac{\partial y}{\partial [x]_n} \right]^T ,$$

where the subscript c denotes the gradient is constrained. This result means the function is not sensitive to the first parameter, which is simply because of the constraint that the function is not allowed to change along the direction e_1. This observation leads naturally to the proposed algorithm where first one needs to identify a basis for the vectors representing the row space of the constraint equations. Next, satisfying the constraints reduces to first assigning zero derivatives to each vector in this basis. Second, the remaining directions, i.e., orthogonal to the identified basis, are searched for sensitivity information using a directional sensitivity analysis approach as implied by Eqs. (6) and (7). Effectively, this implies recasting the sensitivity analysis to look for sensitivity information along an effective subspace with less degrees of freedom than the total number of parameters. These degrees of freedom are free to change in a sensitivity analysis since their variations do not violate the constraints.

Consider the set of constraints in Eq. (2), a basis for the row space of $\mathbf{A}$ is given by a QR decomposition of

$$\mathbf{A}^T = \mathbf{Q}_A \mathbf{R}_A \ , \tag{8}$$

where $\mathbf{Q}_A$ has m columns spanning the row space of $\mathbf{A}$:

$$\boldsymbol{Q}_A = [\, q_{A,1} \quad q_{A,2} \quad \ldots \quad q_{A,m} \,] \in R^{n \times m} \ .$$

From Eq. (3), considering that R_A^T is an $m \times m$ nonsingular we gain:

$$\begin{aligned} &\mathbf{A}\Delta x = 0 \rightarrow \mathbf{Q}_A^T \Delta x = 0; \\ &q_{A,i}^T \Delta x = 0, i = 1, \ldots, m \ . \end{aligned} \tag{9}$$

Equation (9) means that allowable parameter perturbations, i.e., the effective sensitivity analysis degrees of freedom, must all be orthogonal to the m columns of the matrix $\mathbf{Q}_A$, i.e., the row space of the matrix $\mathbf{A}$. Since the whole space has n dimensions and the row subspace of $\mathbf{A}$ has m dimensions, the remaining orthogonal subspace has n-m dimensions, which may be spanned by an orthonormal basis of the form:

$$\{q_1, \ldots, q_{n-m}\} \ ,$$

where

$$\mathbf{Q}_A^T q_i = 0 \text{ for } i = 1, \ldots, n - m \ ,$$

and the resulting constrained gradient would be given by

$$g_c = \alpha_{q_1} q_1 + \ldots + \alpha_{q_{n-m}} q_{n-m} \ .$$

Notice that this gradient does not have any components along the columns of the $\mathbf{Q}_A$ matrix in order to satisfy the constraint. Thus, to estimate the gradient, one needs to find a set of *n-m* directions (referred to as the sensitivity analysis effective degrees of freedom) that are orthonormal and all orthogonal to the row space of **A**. This is a straightforward linear algebra exercise that can be done using a simple projection operation as follows. First, create an orthogonal projector **P**:

$$\mathbf{P} = \mathbf{I} - \mathbf{Q}_A \mathbf{Q}_A^T \ .$$

This is a standard orthogonal projection operator that takes any vector x and extracts the component that's orthogonal to the subspace $R(\mathbf{Q}_A)$ spanned by the columns of $\mathbf{Q}_A$, i.e., $\mathbf{P}x \perp R(\mathbf{Q}_A)$, as shown in Fig. 1. Figure 1 shows a three-dimensional parameter space, with the black arrow representing the direction of a single constraint denoted by the constraint subspace, in this case of single dimension, corresponding to $m = 1$. The blue outlined subspace represents the remaining directions over which sensitivity information can be collected, denoted by the sensitivity subspace. To find the sensitivity subspace, one throws a random direction x and removes the component along the constraint subspace using the projector **P**. This ensures the constraint is satisfied because the remaining component $\mathbf{P}x$ becomes orthogonal to the constraint subspace. If this process is repeated *n-m* times, i.e., two times in this $n = 3$ dimensional example, one can find *n-m* random directions that span the sensitivity subspace. An application of the QR algorithm to these random directions can identify an orthonormal set of *n-m* vectors suitable to identify the gradient as implied by Eq. (6).

Hence the overall algorithm is summarized as follows:

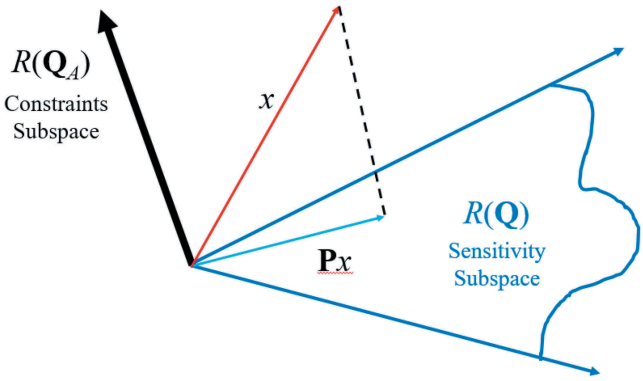


Fig. 1. Relationship between sensitivity and constraint subspaces.

1. Starting with the linear constraints defined by matrix **A**, calculate $\mathbf{Q}_A$ such that: $\mathbf{A}^T = \mathbf{Q}_A \mathbf{R}_A$. Each column of the $\mathbf{Q}_A$ matrix refers to a direction along which a zero derivative is assigned since it describes a direction along which parameter perturbations are not allowed.

2. Generate a random matrix $\mathbf{X} \in R^{n \times (n-m)}$, with each entry being normally distributed with unity variance and zero mean, i.e., *n-m* random and unconstrained realizations of the *n* input parameters.

3. Calculate **PX**, i.e., the projection of the randomly generated inputs to the active subspace containing *n-m* realizations of input perturbations each satisfying the constraints.

4. Calculate **Q** such that **QR** = **PX** (QR decomposition of the projected inputs).

5. Let $\mathbf{Q} = [\, q_1 \quad q_2 \quad .... \quad q_{n-m} \,]$. Each one of these directions refers to one of the *n-m* effective degrees of freedom that are to be explored by the sensitivity analysis.

6. Calculate the directional derivative along q_i with step size Δ_{q_i} as

$$\alpha_{q_i} = \frac{\partial y}{\partial q_i} = \frac{f(x + \Delta_{q_i} q_i) - f(x)}{\Delta_{q_i}}, i = 1, \ldots, n-m \ .$$

7. Calculate the gradient as

$$g_c = \sum_{i=1}^{n-m} \alpha_{q_i} q_i \ .$$

This algorithm can be verified by repeated execution, each time starting with different random matrix **X**. The success criteria are two part. First, the algorithm must produce the same gradient regardless of the starting random matrix **X**. Second, the resulting direction must be the direction of maximum change. These success criteria are employed in Sec. IV to test the proposed algorithm.

Next, notice that the algorithm and explanation have been written based on absolute perturbation so far. The sensitivity analysis, however, can be implemented based on either absolute or relative parameter perturbations. Absolute parameter perturbations are implemented by adjusting the parameters according to $[x]_i^p = [x]_i + \Delta_i$, whereas relative perturbations are introduced via $[x]_i^p = [x]_i(1 + \Delta_i)$, where the superscript p denotes the perturbed value for the i'th input parameter. Relative perturbations allow the analyst to calculate derivatives in unit-less form, whereas absolute perturbations tie the derivatives to the parameter units. The relative

approach is typically preferred for two reasons. First, when the parameters are markedly different in scale, e.g., differing in value by several orders of magnitude, it makes the comparison of the derivatives difficult. Second, when developing automated recipes for perturbing parameters it is simpler to work with relative perturbations, which allows for a generic perturbation approach that is agnostic to the parameter units and hence can be applied to any set of parameters.

IV. NUMERICAL EXPERIMENT AND MODEL DESCRIPTION

The beginning of this section describes the model employed to demonstrate the application of the proposed sensitivity analysis. The rest of this section describes the numerical results. The CANDU-6 core model is employed as a test case study. The lattice physics calculations are performed using the NEWT sequence in the SCALE suite,[17] and the core-wide calculations are done using the NESTLE-C core simulator.[18] The focus is on performing sensitivity analysis of the k_{eff} of the core-wide model with respect to the prompt neutron fractions as condensed in the few-group format that is input to the core simulator. Two different group structures, i.e., 4G and 8G, are employed to test the proposed algorithm.[19]

The CANDU-6 core is a single lattice–type with 37 fuel elements containing natural uranium in Zircaloy[20] with heavy water serving as both coolant and moderator. The heavy water inside the pressure tubes serves as the coolant at a temperature of 563 K, and the heavy water outside the pressure tubes acts as moderator at a temperature of 341 K. The detailed fuel bundle dimensions are given in Table I.

In the pressure tube, the fuel pins are surrounded by the cladding, and beyond the cladding there are three rings. The pressure tube is surrounded by a calandria tube, but there is a small gap, which is modeled as vacuum for present purposes, between the pressure and calandria tubes. The SCALE code is employed to generate the few-group cross sections for the CANDU-6 lattice. A representative schematic of the lattice as generated by SCALE is shown in Fig. 2. The cross-section group structure employed is shown in Table II.

TABLE I

Fuel Bundle Dimensions*

Fuel pin outer radius	0.6103
Clad outer radius	0.6522
Rod length	49.5260
Ring 1 radius	1.4480
Ring 2 radius	2.8755
Ring 3 radius	4.3305
Pressure tube inner radius	5.1689
Pressure tube outer radius	5.6032
Calandria tube inner radius	6.4478
Calandria tube outer radius	6.5875
Radial (x-y) lattice pitch	28.5750
Axial (z) lattice pitch	49.5300

*All units in centimeters.

The CANDU-6 core consists of 190 horizontal fuel channels, each comprised of 12 fuel bundles.[21] The adjuster rods are perpendicular to the fuel channels in three rows of seven at z = 217.49, 297.18, and 377.18 cm, where z = 0 is the exterior boundary of the first axial plane. These rods extend six lattice pitches from the plane of symmetry and contain a solid stainless steel cylinder (shim) that is located in the center of a stainless steel tube. The shim and tube have two vertical segments, each 87.725 and 171.45 cm long from the midplane. The detailed dimensions are given in Table III. The shim and tube assembly are surrounded by a zirconium guide tube. The gaps between the shim and the tube are filled with a heavy water. The reference channel burnup and power distributions are shown, respectively, in Figs. 3 and 4.

The numerical results are designed to demonstrate the application of the proposed algorithm using a forward-based implementation of the sensitivity analysis, relying on the use of a finite differencing operator as implied by Eq. (7). To achieve this, the optimum finite differencing step is first calculated, as required by finite differencing techniques. Next, the algorithm is tested with two different group structures to calculate the derivative of the critical eigenvalue, i.e., multiplication factor, with respect to the parameters characterizing the prompt fission spectrum. The prompt fission spectrum is described in a continuous form using the $\chi(E)$ function such that $\chi(E)dE$ estimates the fraction of fission neutrons released within dE. In energy-group representation, the fission spectrum is described using an n dimensional vector, where the i'th component represents the fraction of neutrons emitted in the i'th group. The models employed in this study use the following values for the fission spectra in the 4G and 8G groups, respectively:

$$\chi^{4G} = [0.96598\ 0.02402\ 0.00500\ 0.00500]^T$$

and

$$\chi^{8G} = [0.3599\ 0.4062\ 0.1998\ 0.0187\ 0.0053\ 0.0050\ 0.0025\ 0.0025]^T .$$

Regarding the finite differencing step, the goal is to determine the optimum step size Δ_{q_i} for each of the q_i directions. The optimality attempts to find a tradeoff

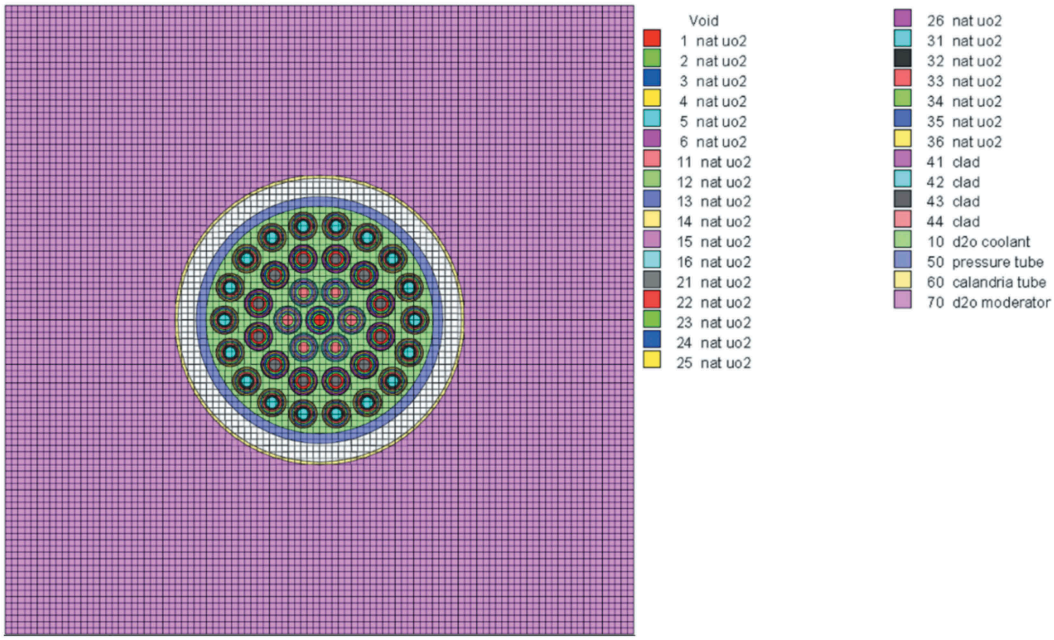


Fig. 2. CANDU-6 lattice model.

TABLE II

Energy Group Boundaries for Core Model

Few-Group Structure	Boundaries (MeV)
Four group (4G)	6.25E−7
	1.22E−4
	2.00E−1
Eight group (8G)	1.00E−7
	6.26E−7
	1.22E−4
	9.50E−3
	2.00E−1
	9.50E−1
	2.3540

between a step size too small that may cause numerical errors and too big that may not be able to capture local variations. This can be achieved by executing the code with the following perturbations:

$$\frac{\partial y}{\partial q_i} = \frac{f(x+\varepsilon q_i) - f(x)}{\varepsilon} \quad .$$

This expression implies calculating the directional directive of the function along the direction q_i, with the size of the perturbation being equal to ε. Different values for ε can be employed and a plot of the derivative versus ε can reveal the optimum value Δ_{q_i}. This process should be repeated for each direction q_i since the ideal finite differencing step may in general be different.

TABLE III

Adjuster Rod Dimensions*

	Inner Element (0.000 to 85.725 cm)	Outer Element (85.725 to 171.450 cm)
Shim outer radius	0.650	0.710
Steel tube inner radius	3.607	3.607
Steel tube outer radius	3.725	3.690
Guide tube inner radius	4.519	4.519
Guide tube outer radius	4.572	4.572

*All units in centimeters.

Figure 5 shows a representative result for this analysis for the q_1 direction in the 8G model. Results indicate that $0.1 < \varepsilon < 10$ is an acceptable step size for a sensitivity analysis based on relative parameter perturbations $0.01 < \varepsilon < 1.0$ for absolute perturbations. Similar step sizes for all other directions have been identified in the same way.

To test the algorithm, it is repeated with different random matrix $\mathbf{X}$ for both the 4G and 8G models to ensure the same gradient is calculated. The algorithm is demonstrated below using absolute perturbations of the fission spectrum parameters for the 4G case.

Random generate a matrix of candidate perturbation $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} -0.7905 & 0.0819 & -1.3873 \\ -1.1625 & 0.0839 & -0.1486 \\ 0.3457 & 0.3475 & -2.0856 \\ -0.6463 & 0.1356 & 0.7771 \end{bmatrix} .$$

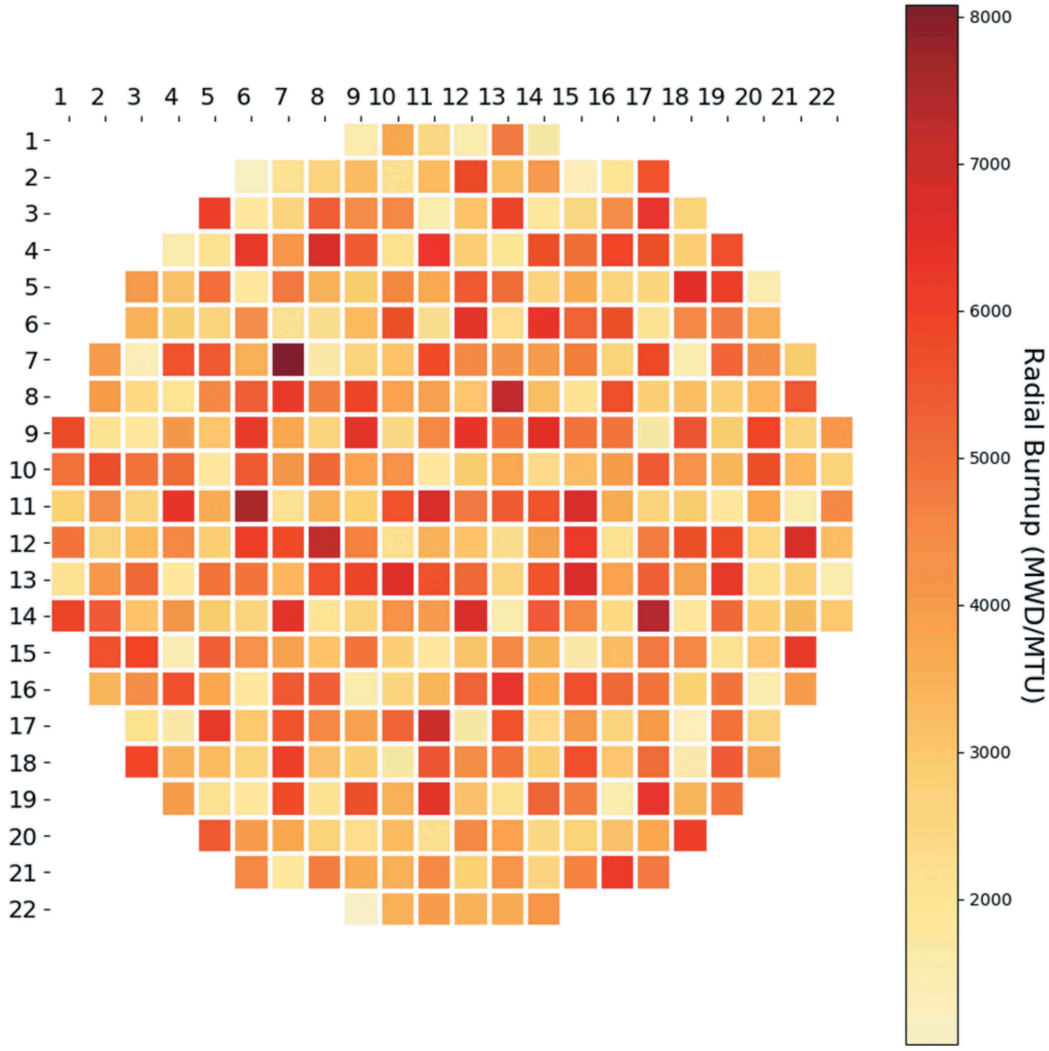


Fig. 3. Reference channel burnup distribution.

These candidate perturbations do not satisfy the constraint given by (after normalization to have unit norm)

$$\mathbf{Q}_A = [0.5 \quad 0.5 \quad 0.5 \quad 0.5]^T \rightarrow \|\mathbf{Q}_A\| = 1.0 \ ,$$

that is to say $\mathbf{Q}_A^T\mathbf{X} \neq 0$ as required by Eq. (3).

Per the proposed algorithm, the associated projection matrix for the perturbations is given by

$$\mathbf{P} = \begin{bmatrix} 0.7500 & -0.2500 & -0.2500 & -0.2500 \\ -0.2500 & 0.7500 & -0.2500 & -0.2500 \\ -0.2500 & -0.2500 & 0.7500 & -0.2500 \\ -0.2500 & -0.2500 & -0.2500 & 0.7500 \end{bmatrix} .$$

The projected perturbations are given by

$$\mathbf{PX} = \begin{bmatrix} -0.2271 & -0.0803 & -0.6762 \\ -0.5991 & -0.0784 & 0.5625 \\ 0.9091 & 0.1853 & -1.3745 \\ -0.0829 & -0.0266 & 1.4882 \end{bmatrix} .$$

Note that each column in this matrix represents perturbations that satisfy the constraint. Next, these columns are orthonormalized using a QR decomposition yielding three orthonormal directions that are orthogonal to the constraint:

$$\mathbf{Q} = \begin{bmatrix} -0.2036 & 0.6901 & -0.4820 \\ -0.5372 & -0.6526 & -0.1887 \\ 0.8151 & -0.2392 & -0.1682 \\ -0.0743 & 0.2017 & 0.8389 \end{bmatrix} .$$

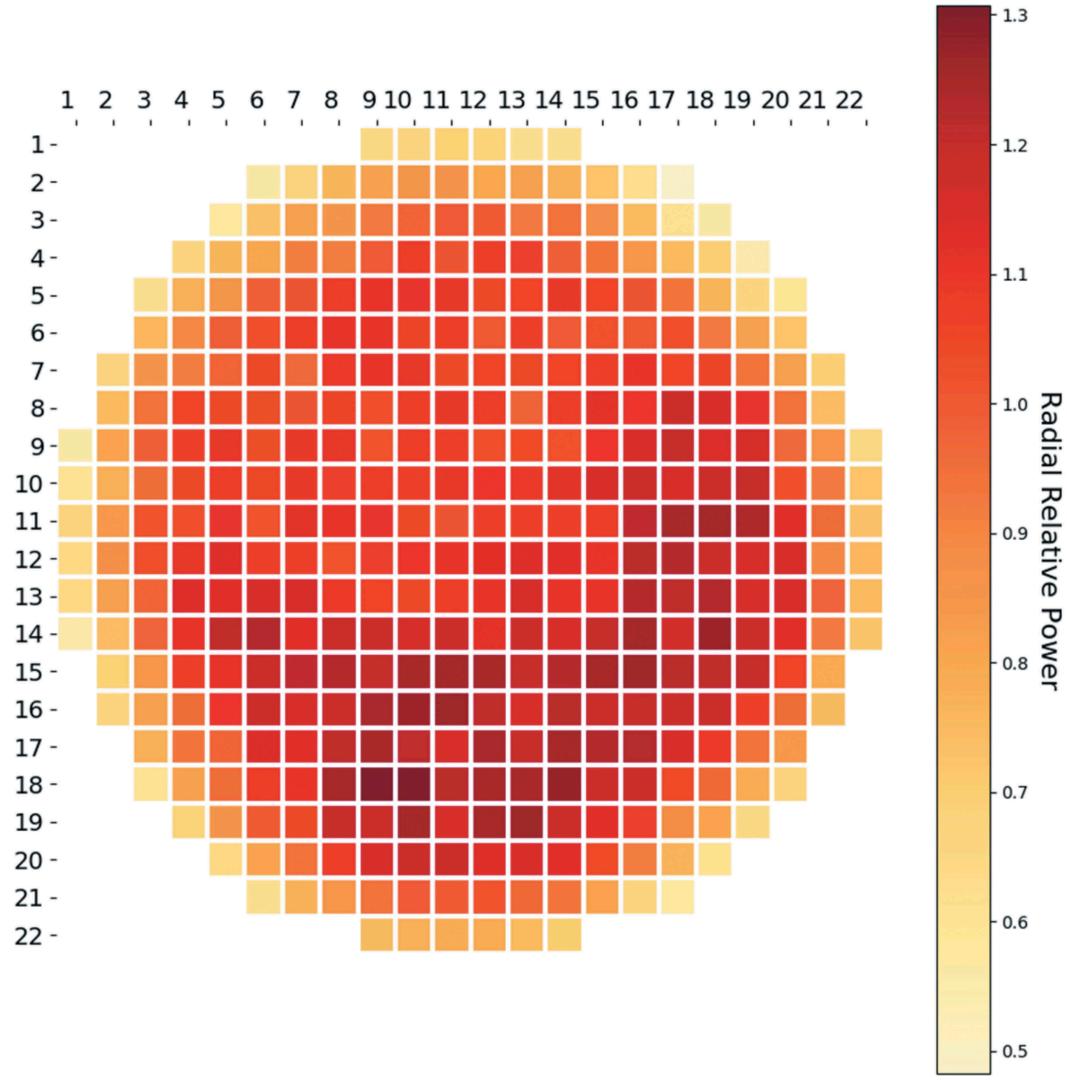


Fig. 4. Reference channel power distribution.

Next the model is executed three additional times, where each time the four parameters are perturbed by the individual components of each of the three columns in the **Q** matrix after scaling them by the optimum scaling parameters identified before. For the 4G case with absolute perturbations, the optimum scaling parameter is given by 0.1.

The resulting derivatives of k_{eff} along each of the three directions are given by

$$\frac{\partial y}{\partial q} = [\,0.3278 \quad 0.3163 \quad 0.1023\,] \text{ pcm} .$$

Finally, the reconstructed gradient in terms of the original four parameters is given by

$$g_c = [\,-0.0341 \quad -0.0575 \quad 0.0019 \quad 0.0897\,] .$$

To test the stability of our methodology, the same process was repeated several times starting with different random **X** matrices, producing the following representative results:

$$g_{c1} = [\,-0.0344 \quad -0.0576 \quad 0.0020 \quad 0.0900\,] ,$$

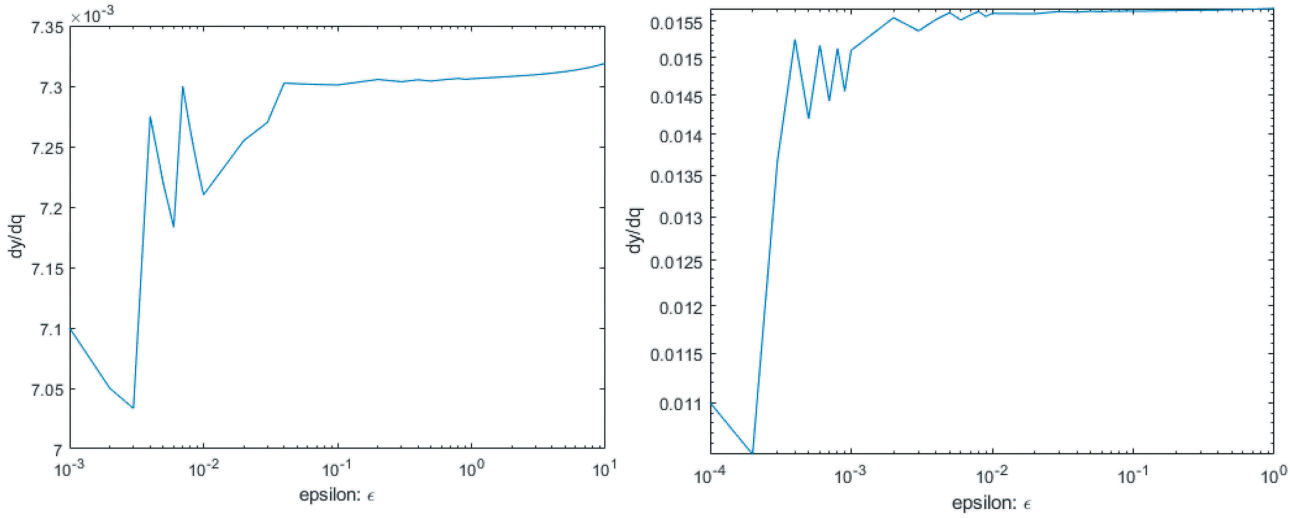
$$g_{c2} = [\,-0.0343 \quad -0.0577 \quad 0.0020 \quad 0.0900\,] ,$$

$$g_{c3} = [\,-0.0341 \quad -0.0578 \quad 0.0019 \quad 0.0900\,] ,$$

and

$$g_{c4} = [\,-0.0339 \quad -0.0575 \quad 0.0018 \quad 0.0896\,] .$$

Next, a representative set of results for the 8G case is given below:



(a) Relative Perturbations

(b) Absolute Perturbations

Fig. 5. Derivative plots with different ε values.

$$\mathbf{X} = \begin{bmatrix} -0.0482 & 1.1818 & 1.0181 & -1.8606 & 0.0699 & 1.4146 & 1.0237 \\ 1.6049 & -0.5244 & -1.0753 & 0.8695 & 0.7061 & -1.4778 & 0.1162 \\ 0.0754 & 1.1611 & -0.8535 & -1.1365 & 0.3430 & -2.0364 & 0.0470 \\ -0.3883 & -0.1582 & 0.0285 & -0.8927 & 0.0157 & 0.8700 & -0.3062 \\ -0.7077 & 1.0138 & -1.1207 & -0.1158 & -1.6396 & 0.5390 & -0.7065 \\ 0.4684 & -0.6067 & -0.9064 & 0.5320 & -0.2310 & -0.4603 & 0.3251 \\ -0.0223 & 0.0190 & 1.1880 & -0.1675 & 0.1010 & 1.6377 & -1.2673 \\ 0.8288 & 0.3675 & -1.2313 & -0.1572 & 0.8329 & 0.0985 & 0.1534 \end{bmatrix},$$

$$\mathbf{Q_A} = [0.3536 \quad 0.3536 \quad 0.3536 \quad 0.3536 \quad 0.3536 \quad 0.3536 \quad 0.3536 \quad 0.3536],$$

$$\mathbf{P} = \begin{bmatrix} 0.8750 & -0.1250 & -0.1250 & -0.1250 & -0.1250 & -0.1250 & -0.1250 & -0.1250 \\ -0.1250 & 0.8750 & -0.1250 & -0.1250 & -0.1250 & -0.1250 & -0.1250 & -0.1250 \\ -0.1250 & -0.1250 & 0.8750 & -0.1250 & -0.1250 & -0.1250 & -0.1250 & -0.1250 \\ -0.1250 & -0.1250 & -0.1250 & 0.8750 & -0.1250 & -0.1250 & -0.1250 & -0.1250 \\ -0.1250 & -0.1250 & -0.1250 & -0.1250 & 0.8750 & -0.1250 & -0.1250 & -0.1250 \\ -0.1250 & -0.1250 & -0.1250 & -0.1250 & -0.1250 & 0.8750 & -0.1250 & -0.1250 \\ -0.1250 & -0.1250 & -0.1250 & -0.1250 & -0.1250 & -0.1250 & 0.8750 & -0.1250 \\ -0.1250 & -0.1250 & -0.1250 & -0.1250 & -0.1250 & -0.1250 & -0.1250 & 0.8750 \end{bmatrix},$$

$$\mathbf{PX} = \begin{bmatrix} -0.2746 & 0.8626 & 1.3872 & -1.4945 & 0.0451 & 1.3415 & 1.1005 \\ 1.3785 & -0.7437 & -0.7062 & 1.2356 & 0.6813 & -1.5510 & 0.1931 \\ -0.1510 & 0.8419 & -0.4845 & -0.7704 & 0.3183 & -2.1096 & 0.1238 \\ -0.6147 & -0.4775 & 0.3975 & -0.5266 & -0.0090 & 0.7968 & -0.2294 \\ -0.9341 & 0.6945 & -0.7516 & 0.2503 & -1.6643 & 0.4659 & -0.6297 \\ 0.2420 & -0.9259 & -0.5374 & 0.8981 & -0.2558 & -0.5334 & 0.4019 \\ -0.2487 & -0.3002 & 1.5571 & 0.1986 & 0.0762 & 1.5646 & -1.1905 \\ 0.6024 & 0.0483 & -0.8622 & 0.2089 & 0.8082 & 0.0253 & 0.2303 \end{bmatrix},$$

$$\mathbf{Q} = \begin{bmatrix} -0.1421 & -0.4387 & 0.5533 & -0.2124 & -0.5142 & 0.1794 & 0.1207 \\ 0.7136 & 0.0322 & -0.0107 & 0.1856 & -0.2482 & -0.1827 & -0.4849 \\ -0.0781 & -0.4636 & -0.1918 & -0.2381 & 0.3450 & -0.6576 & 0.0953 \\ -0.3182 & 0.4745 & -0.0025 & -0.5693 & 0.0757 & 0.0731 & -0.4621 \\ -0.4835 & -0.1378 & -0.4908 & 0.5166 & -0.2396 & 0.1479 & -0.1876 \\ 0.1253 & 0.4868 & -0.2132 & -0.0522 & -0.3089 & -0.1511 & 0.6752 \\ -0.1287 & 0.2567 & 0.5699 & 0.5025 & 0.4532 & -0.0769 & 0.0627 \\ 0.3118 & -0.2112 & -0.2141 & -0.1327 & 0.4370 & 0.6678 & 0.1818 \end{bmatrix},$$

$$\frac{dy}{dq} = [0.225 \quad 0.118 \quad 0.633 \quad 0.516 \quad 0.723 \quad 0.910 \quad 0.743] \text{ pcm},$$

$$g_c = [0.0049 \quad -0.0453 \quad -0.0599 \quad -0.0533 \quad -0.0357 \quad 0.0060 \quad 0.0937 \quad 0.0895],$$

$$g_{c1} = [0.0049 \quad -0.0452 \quad -0.0599 \quad -0.0534 \quad -0.0358 \quad 0.0060 \quad 0.0938 \quad 0.0895],$$

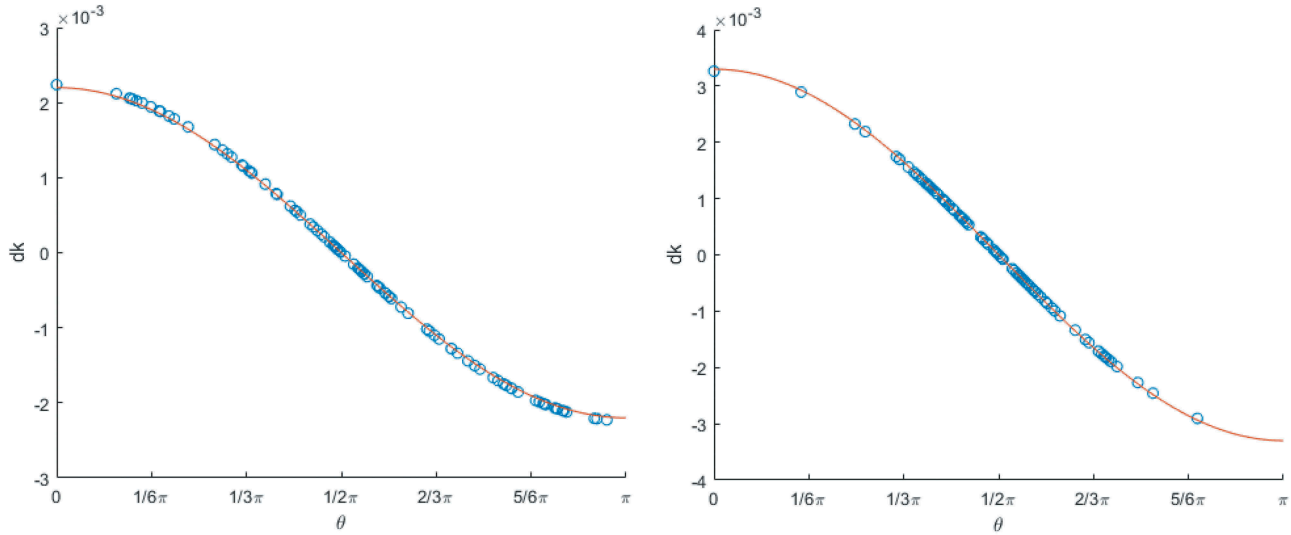
$$g_{c2} = [0.0049 \quad -0.0453 \quad -0.0599 \quad -0.0534 \quad -0.0358 \quad 0.0061 \quad 0.0938 \quad 0.0896],$$

$$g_{c3} = [0.0049 \quad -0.0452 \quad -0.0599 \quad -0.0534 \quad -0.0358 \quad 0.0061 \quad 0.0938 \quad 0.0896],$$

and

$$g_{c4} = [0.0049 \quad -0.0453 \quad -0.0599 \quad -0.0534 \quad -0.0358 \quad 0.0061 \quad 0.0938 \quad 0.0896].$$

Another test for the calculated gradient is conducted next. By definition, the gradient is expected to coincide with the direction that results in the maximum change. This basic criterion can be used to devise the next test. The idea is to generate many candidate perturbations and force them to satisfy the constraints using one of the recipes proposed in earlier work, for example, the "full normalization" approach is selected herein. The code is executed with this perturbation, and the angle between the selected perturbation and the calculated gradient is calculated. This angle is plotted against the change in k_{eff} scaled down by the Euclidean norm of the perturbation for a number of randomly generated perturbations as shown in Fig. 6. A cosine function, in orange, is superimposed on the plot for illustration. The results are consistent with one's expectation about the definition of the gradient, which produces the maximum change in the quantity of interest for all perturbations of the same Euclidean norm.



(a) Four group model

(b) Eight group model

Fig. 6. k_{eff} variations along randomized parameter directions versus angle with calculated gradient.

V. CONCLUSION

This paper presented an algorithm to support the conduction of constrained sensitivity analysis when the constraints are described using linear equations. This algorithm allows one to generate perturbations for model parameters, as required by sensitivity analysis, that do not have any components that violate the constraint equations. The idea is to calculate a projection operator that retains only components that satisfy the constraints. Theoretical developments have demonstrated that a problem with n model parameters and m constraints can be recast into a problem with only n-m effective model parameters, referred to as degrees of freedom, that can be perturbed freely as required by sensitivity analysis. The derivatives are calculated with respect to these n-m degrees of freedom, and via a simple transformation, the derivatives can be represented back in terms of the original parameters. Numerical results were presented to support the proposed algorithm. Future work will focus on employing the proposed constraint sensitivity analysis to complete other supporting engineering analyses, e.g., uncertainty and inference analyses.

References

1. A. SALTELLI, K. CHAN, and E. M. SCOTT, *Sensitivity Analysis*, John Wiley (2000).
2. J. C. HELTON et al., "Survey of Sampling Based Methods for Uncertainty and Sensitivity Analysis," *Rel. Eng. Syst. Saf.*, **91**, 1175 (2006); https://doi.org/10.1016/j.ress.2005.11.017.
3. H. S. ABDEL-KHALIK, Y. BANK, and C. WANG, "Overview of Hybrid Subspace Methods for Uncertainty Quantification, Sensitivity Analysis," *Ann. Nucl. Energy*, **52**, 28 (2013); https://doi.org/10.1016/j.anucene.2012.07.020.
4. J. MORIO, "Global and Local Sensitivity Analysis Methods for a Physical System," *Eur. J. Phys.*, **32**, 1577 (2011); https://doi.org/10.1088/0143-0807/32/6/011.
5. W. M. STACEY, *Variational Methods in Nuclear Reactor Physics*, Academic Press, New York (1974).
6. M. L. WILLIAMS, *Perturbation Theory for Reactor Analysis. CRC Handbook of Nuclear Reactor Calculations*, pp. 63–188, CRC Press (1986).
7. Z. PERKÓ et al., "Ambiguities in the Sensitivity and Uncertainty Analysis of Reactor Physics Problems Involving Constrained Quantities," *Nucl. Sci. Eng.*, **180**, 3 (2015); https://doi.org/10.13182/NSE14-17.
8. J. A. FAVORITE et al., "Adjoint-Based Sensitivity and Uncertainty Analysis for Density and Composition: A User's Guide," *Nucl. Sci. Eng.*, **185**, *3*, 384 (2017); https://doi.org/10.1080/00295639.2016.1272990.
9. Z. PERKÓ. et al., "Large Scale Applicability of a Fully Adaptive Non-Intrusive Spectral Projection Technique: Sensitivity and Uncertainty Analysis of a Transient," *Ann. Nucl. Energy*, **71**, 272 (2014); https://doi.org/10.1016/j.anucene.2014.03.035.
10. B. BROADHEAD et al., "Sensitivity- and Uncertainty-Based Criticality Safety Validation Techniques," *Nucl. Sci. Eng.*, **146**, 340 (2004); https://doi.org/10.13182/NSE03-2.
11. "Assessment of Sensitivity/Uncertainty Analysis Capabilities Applicable for the Nuclear Fuel Cycle," ACE Workshop, North Carolina State University (May 2006).
12. "Management of Numerical Errors in Nuclear Systems Modeling," ACE Workshop, North Carolina State University (Sep. 2007).
13. "Simulation and Modeling for Advanced Nuclear Energy Systems," Department of Energy Workshop, Department of Energy Offices of Nuclear Energy and Advanced Scientific Computing Research (Aug. 2006).
14. "Nuclear Energy Advanced Modeling and Simulation (NEAMS), Department of Energy Workshop, Argonne National Laboratory (May 2008).
15. T. APOSTOL, *Calculus, Vol. 2, Multi-Variable Calculus and Linear Algebra with Applications*, Wiley (1969).
16. Y. S. BANG, H. S. ABDEL-KHALIK, and J. M. HITE, "Hybrid Reduced Order Modeling Applied to Nonlinear Models," *Int. J. Numer. Meth. Eng.*, **91**, *9*, 929 (2012); https://doi.org/10.1002/nme.4298.
17. "SCALE Code System," ORNL/TM-2005/39, Version 6.2.2, B. REARDEN and M. JESSEE Eds., UT-Battelle, LLC, Oak Ridge National Laboratory (2017).
18. P. TURINSKY and H. SARSOUR, "NESTLE-C: Few-Group Neutron Diffusion Equation Solver Utilizing the Nodal Expansion Method for Eigenvalue, Adjoint, Fixed-Source Steady-State and Transient Problems: CANDU Version," North Carolina State University (2003).
19. D. HUANG and H. S. ABDEL-KHALIK, "Application of Cross Sections Uncertainty Propagation Framework to Light and Heavy Water Reactor Systems," *ASME J. Nuclear Rad. Sci.*, NERS-18-1134, **6**, *1*, 011104 (2020); https://doi.org/10.1115/1.4045032.
20. J. M. POUNDERS et al., "A 3D Stylized Half-cCore CANDU Benchmark Problem," *Ann. Nucl. Energy*, **38**, *4*, 876 (2011); https://doi.org/10.1016/j.anucene.2010.10.018.
21. "Enhanced CANDU 6 Technical Summary," SNC Lavalin (2015); https://www.snclavalin.com/~/media/Files/S/SNC-Lavalin/documents/enhanced-candu-6-technical-summary-en.pdf.