Validation and Uncertainty Quantification of CFD Smooth Seal Models: ANSYS and Bulk-Flow

A Thesis

Presented to

the faculty of the School of Engineering and Applied Science

University of Virginia

in partial fulfillment
of the requirements for the degree

Master of Science

by

Madeline Carlisle Collins

May 2020

APPROVAL SHEET

This Thesis
is submitted in partial fulfillment of the requirements
for the degree of
Master of Science

Author Signature: Madeline G. C. Collins

This Thesis has been read and approved by the examining committee:

Advisor: Houston Wood

Committee Member: Roger Fittro

Committee Member: Haibo Dong

Committee Member: ______________

Committee Member: ______________

Committee Member: ______________

Accepted for the School of Engineering and Applied Science:

Craig H. Benson, School of Engineering and Applied Science

May 2020

Contents

Acknowledgements

A big thanks to my advisor, Houston Wood, and Cori Watson-Kassa for faithfully guiding me throughout this endeavor. Thank you to Neal Morgan for helping build the CFD model and an efficient sensitivity study.

Abstract

Turbomachinery faces rotordynamic issues such as imbalance, instability, and large vibrations. Such issues are costly in downtime and maintenance, while also introducing safety concerns. Careful design of components such as bearings and seals can reduce vibrations. Practitioners build computational fluid dynamics (CFD) models of these components to obtain performance predictions. The reliability of such CFD software is typically unknown, and performance studies have shown significant prediction disagreement between different models of the same scenario. Standard literature validation practice lacks rigor and formalization. Furthermore, sufficient experimental data is scarce and difficult to obtain. These constraints invoke the need for a standardized method of validating CFD models for turbomachinery components. In 2009, ASME published a standard for verification and validation of CFD models, V&V-20. This work intends to serve as (1) an assessment of the feasibility of implementing the V&V-20 procedure (2) a baseline in the refinement validation procedure for CFD models of turbomachinery component. Methods from V&V-20 are applied an ANSYS CFX model and a ROMAC model of a straight liquid seal model. One prediction from each model is evaluated in this work. In practice, this procedure would be repeated for other predictions at various input conditions. This study resulted in uncertainty quantification of flow out of the seal, or leakage: experimental uncertainty (5%), numerical uncertainty (0.1% and 0.5%), input uncertainty (16% and 7%), and model error interval (-21% to 13% and -19% to -1%). These quantifications involved a grid convergence study and perturbation study consisting of hundreds of simulations. The experiment used for validation was performed by Kaneko, et al (2003). The study found V&V-20 to be useful but tedious for practical application, especially on a complicated solver such as ANSYS CFX; it serves as a thorough and useful starting point for quantifying uncertainties and standardizing terminology.

KEYWORDS

Keywords: Validation, Uncertainty, Numerical Uncertainty, Input Uncertainty, Convergence Study, Sensitivity Analysis, Grid Refinement, Bulk Flow Model, ASME Standard, ANSYS, Computational Fluid Dynamics, Rotordynamics, Smooth Seal, Turbomachinery, Rotating Machinery, Meshing, Turbulence

Chapter 1: Overview

1.1. Motivation

Computational fluid dynamics (CFD) software predicts fluid behavior by numerically solving governing equations of fluid mechanics. The Navier Stokes equations are the best mathematical representation of fluid behavior. Typically, commercial CFD software such as ANSYS solve the Navier Stokes equations numerically. Considering the complexity of the Navier Stokes equations, the embedded algorithm can take a long time to converge to a solution. Additionally, this type of software has a steep learning curve. Together, their learning curve and delayed results render complex CFD software inefficient for engineers in many industries. However, if an engineer is modelling a specific family of fluid dynamic problems, assumptions may lead to a simplified version of the Navier Stokes equations. Simplified equations will converge to a solution more quickly. Such CFD software is termed lower fidelity or reduced because of its narrower applicability and capability.

The Rotating Machinery and Controls (ROMAC) Laboratory at the University of Virginia focuses on fluid flow in machine components such as bearings and seals. ROMAC develops software for designing and analyzing such fluid regions and their effects on the system. Since thin-film lubrication flow is a specific sector of fluid dynamics, ROMAC applies the appropriate assumptions to develop lower fidelity codes for different machine components and various geometries. ROMAC codes converge more quickly than commercial full Navier Stokes solvers such as ANSYS. This ROMAC software is much more feasible for rotordynamic engineers to use in practice.

Most people build models for one of two reasons: 1) imitating an existing physical system, 2) predicting behavior of a new, nonexistent system. Software developers build models for imitating when testing out their modeling software while end-users do so to troubleshoot observed problems. Prediction is usually for end-users for design. Either way, the model is intended to emulate reality. However, reality is too complex to be modeled perfectly, and reality includes unexplained randomness. As George Box puts it, “All models are wrong, but some are useful.” Even the complexity of Navier Stokes and commercial CFD cannot fully capture reality. Error in a model due to assumptions is termed model error. Lower-Fidelity models contain more simplifying assumptions and thus higher model error. A model should not be used for prediction until reasonable agreement with reality is demonstrated.

Validation, as defined in [1], is the determination of the degree to which a model agrees with reality. For models built to imitate, this can be done directly once a metric for agreement is established. If a model's agreement with reality is considered acceptable, then the model may be used for prediction.

Validation for model developers, then, is limited by the available experimental data or acquisition thereof. In rotordynamics, experimental data is sparse which limits the robustness and trustworthiness of validation studies. Researchers in ROMAC have begun building ANSYS models, validating them with the limited experimental data, and then using the ANSYS models' outputs to represent reality in validating or checking low-fidelity models. This ANSYS representation of reality is used to supplement experimental data, and maybe eventually replace it in full. This perspective is controversial, and its statistical validity must be questioned. However, if the ANSYS model is found to be a reasonable representation of reality, then this may be a defendable substitution. Regardless of a validation study's rigor, users will trust software if it has predicted well for them in personal experience. So, the software developers must balance theory with practicality. That is, the low-fidelity model should be made as accurate as possible by whatever means are available.

The goals of this project are to outline (1) the readily available progress made in CFD model validation, (2) where steps in the validation process may fit practically for developers and users, (3) reveal the gaps in validation efforts and standardization. This project will address the difficulties with coupling of ANSYS and sparse experimental data to assess the reliability of a low-fidelity model. The practical goal for software developers should be optimizing accuracy and reporting the accuracy in an understandable way to the end-user. For the end-user, higher accuracy and known accuracy means software can be trusted with more confidence, allowing engineers to push the limits in design. This can serve preventative measures such as avoiding failure, shut down, large losses, and injuries. This can also serve to improve efficiency and increase profit. Rotating machinery keeps progressing towards higher rotational speeds and fluid pressures. With improved modeling capabilities, these industrial advancements can be pursued sooner and with more confidence.

1.2. Background

1.2.1. NEED FOR VALIDATION

A user's confidence in model predictions is entirely dependent on the model's ability to predict reality well [2]. Validation, according to V&V-20 [1], is the determination of the degree at which a model agrees with reality. Without comprehensive validation, a model's applicability is limited [3]. With proper validation, users can optimize decisions, and models can lead research and development into uncharted territories [2], [3].

A numerical model is a numerical solver which predicts the outcome of a process for a given set of inputs [2]. Models are extremely helpful to modern day design and troubleshooting. Rotating machinery can experience very critical damage from instability, unbalance, and misalignment [3], [4]. A study presents seven problems encountered in such issues where modeling was used to resolve each scenario [4]. However, models often produce inaccurate predictions. For a study, multiple rotordynamics model developers were asked to predict dynamic coefficients (which reflect dynamic behavior of a system) with their models for a specified scenario.

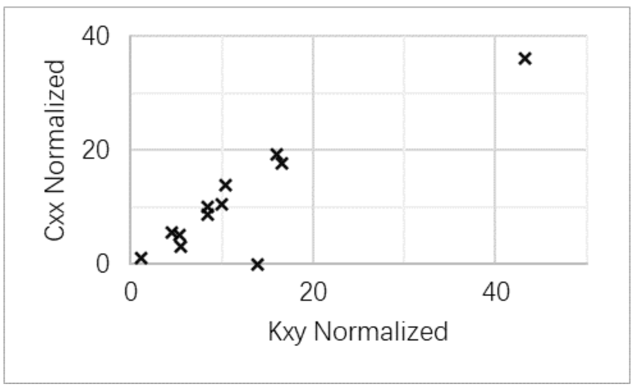

FIGURE 1
DISAGREEMENT BETWEEN MODELS, ADAPTED FROM [5]

A prediction output is on each axis in Figure 1 (stiffness on the horizontal axis, damping on the vertical axis), and each data point represents a particular model's predictions. Ideally, these datapoints should be overlapping. Instead, the spread of the reported outputs varied by an order of magnitude, posing concern for model accuracy and reliability [5]. Model prediction variability is also addressed by Stern et al [6]. These examples highlight the need for validation improvements in this field.

1.2.2. DATA LIMITATIONS

Current validation efforts of rotordynamic models are rudimentary. Often, a model is compared to one experiment graphically, only mentioning minimum and maximum percent difference between the experiment and the predictions [7]–[11]. Additionally, users are often modeling a component of a system, but validation of that component and the whole system may be needed. The component validation would then be propagated to whole system validation [12], so the validation study's reliability becomes more critical. Control theory may also be employed to validate on a system level [13].

Often, validation is done with only of one or two real-world datasets [2], [7]–[11]. Validation efforts have higher error when little data is used [14]. Model inaccuracy demonstrates the need for more experimental data and validation; mass data collaboration could mitigate this issue [5], [14]. A method for scoring completeness of experimental data from validation perspective has been developed by Oberkampf and Smith [15]. When the available data is not complex enough to represent the model's intended use, the validation opportunities is very limited [16]. However, Di Baldassarre and Montanari [17] suggest a method for combining data from a highly-related experiment with data from a tangentially-related experiment using a two-tier hierarchy. This could be a way to address the sparse data problem.

Higher fidelity models have fewer assumptions and more thorough physics theory embedded in their numerical solver. Lower fidelity models have more built-in assumptions and are simplified for efficient runtime. Lower fidelity models are sometimes referred to as surrogate models or reduced models. Comparing a low-fidelity model to a high-fidelity model could be useful in testing if the assumptions are justified for the intended application. This is especially useful when dealing with sparse or insufficient data. Methods have been developed for N-version validation or validation between different CFD models [6]. Terming this 'validation' conflicts with V&V-20 [1] which claims validation must include experimental data. Nonetheless, perhaps two sources of reliable data, experimental data and a validated CFD model's predictions, may be combined with a two-tier hierarchy as discussed above. This combined data could then be used to 'validate' a lower fidelity model.

1.2.3. CALIBRATION

Lower fidelity models often have more user inputs than higher fidelity models. Some inputs may be physically inherent (dependent on the physics) rather than driving the physics. A well understood example of such a parameter is the coefficient of friction. A lower fidelity model may be trained with higher fidelity models to find these inherent parameters [18]. Training refers to calibration or determination of parameters within a model. Calibration is most traditionally done with experimental data, but some models are calibrated with higher fidelity models rather than experimental data [19]. This is commonly used for calibration and validation when limited data is available [14], [20], [21]. Four calibration methods are demonstrated with examples and found to be reliable for small datasets [21]. In facing a small data problem, sampling with Monte Carlo or kNN nearest neighbor may be a useful approach to calibration [19]–[21]. To choose optimal next set of conditions to evaluate, Lewis et al [19] uses an information theory approach, and Perdikaris et al [18] uses Bayesian optimization with an improvement metric. Maximum likelihood estimators may be used to calibrate parameters [18].

Cross validation is a way to couple validation and calibration. One must not validate with the same data used for calibration [22]. Otherwise, a misleading, artificially high agreement will be found between the data and model. Cross validation involves

(1) splitting a dataset into two groups
(2) calibrating with the first group
(3) predicting the second group
(4) comparing the predictions to second group

1.2.4. UNCERTAINTIES

The computational solver and experimental setup have various sources of commonly recognized and anticipated error. V&V-20 [1] recognizes three sources of error in a CFD model: (1) model error – due to modeling assumptions (2) numerical error – due to numerical solver (3) input error – due to uncertain, sensitive input parameters. V&V-20 [1] combines error from experiment into one source of error: experimental error – due to measurement error and randomness.

When assessing the agreement between the model and reality, these errors should be accounted for; otherwise, agreement may appear better or worse than it truly is. Since one cannot know the exact truth, one cannot know the exact errors introduced. Estimation of these errors is termed uncertainty [1]. A study suggests that disagreement between models is likely due to modeling assumptions [5]. V&V-20 [1] considers model uncertainty to be the goal of the validation process or the assessment of model accuracy.

Though the experiment is representing the 'truth,' measurements are taken with manufactured devices, so the truth is blurred with experimental uncertainties. V&V-20 [1] requires a validation experiment for quantifying experimental uncertainty. A validation experiment involves repetition of measurements to observe variation. Data from a validation experiment is rare in literature, and experimental uncertainty of measurements is not common [2], [23], [24]. Sometimes, uncertainty is provided for a calculated value, dissolving the measurement uncertainty by data reduction [25], [26].

1.2.5. STANDARDIZATION OF VALIDATION

Software developers have not agreed upon a validation standard. Taxonomy of validation, verification, and uncertainty is also inconsistent in the literature and a topic of debate [2], [27]. A review of different validation approaches is demonstrated in Biondi, et al [2]. A universally accepted method of validation is needed, though it is difficult for one method to be applicable to a wide range of models [2]. Standardization can establish clear expectations of models for users [2], [28].

The end user is not a statistician and should not be expected to read validation studies and assess a model's trustworthiness. An assessment criterion is presented for certifying validation of CFD which involves the comparison of results from multiple models [6]. This method uses similar theory to that in V&V-20. V&V-20 compares model error and comparison error to assess a model's current state; that is, this method may highlight the driving error source in the model. We need standardized validation criteria so that users can know the reliability of the model for their application [28]. V&V-20 [1] suggests validation should only focus on the domain of intended use of the model. Perhaps, a range of input variables where the model is adequately validated should be presented to the user [17].

Chapter 2: Setup

2.1. ASME Standard: V&V 20-2009

The validation study in this thesis follows the methods of this ASME Standard, referred to as V&V-20.

2.1.1. OBJECTIVE

This standard aims to outline a procedure for validation of computational fluid dynamics and heat transfer models. Validation, as defined here, refers to determining the degree to which a model represents reality, with respect to its intended use. The deliverable of this process is an estimate of model form error or ‘model error’ [1].

2.1.2. SCOPE

This process is centered around comparing one simulation result to one experimental result. Thus, this validation may only be performed if experimental data is available. The procedure requires the experiment and simulation are under the same set of conditions, or at the same ‘validation point.’ Typically, validation is done at various points in the domain of interest, and thus, this procedure would be repeated for each one. Interpolation and extrapolation among validation points is outside of the scope of this standard and left to engineering judgement. The methods within may be applied to topics outside of fluid dynamics and heat transfer [1].

Note, in this thesis, that the standard’s procedure is performed once for each model. This results in evaluation of one prediction variable at one set of conditions. In practice, this procedure would be repeated for other predictions and at various input conditions.

2.1.3. CONTENTS

The standard highlights the importance of verification as a precursor for validation. Verification is considered two-fold: code verification and solution verification. Code verification is the process of checking that the software accurately solves the mathematical model [1]. In other words, code verification checks that the model is doing what the developer intends for it to do. This aligns with its colloquial use of “verification.” This process should reveal any typos or user-errors in the code. Then, of course, the code developer should amend any errors before moving forward to the solution verification and validation study.

Next, solution verification should be performed. Solution verification is the process of estimating the numerical uncertainty of the code. Solution verification is part of the validation process as it results in an estimate of contributing error [1]. This study focuses on the validation portion, assuming code verification has already been performed. However, since solution verification is part of the validation process, it is included in this study.

The model output chosen is called the quantity of interest, QOI. QOI may be flow rate, outlet pressure, outlet temperature, etc. The validation point is the conditions of the experiment. Figure 2 shows QOI plotted at a validation point. x may be thought of as Reynold's number, for example. Typically, validation point is defined by several quantities. At the validation point, there must be experimental data (D) and simulation output (S). Truth (T) is an inherent value that cannot be obtained directly. D is the direct measurement of the true value.

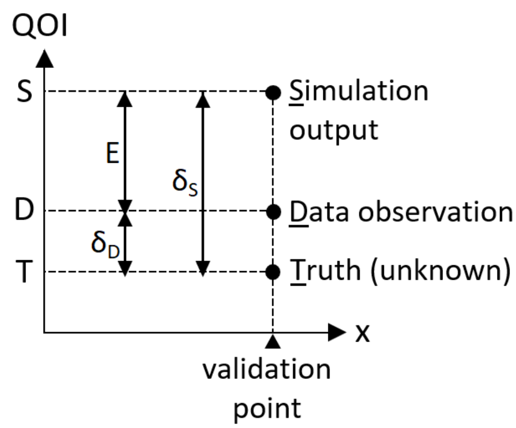

FIGURE 2
CONCEPTUAL DIAGRAM FOR ERROR, ADAPTED FROM [1]

Figure 2 demonstrates conceptual relationships between experiment, simulation, and truth. Error, denoted by δ, is defined as the difference between an observed value and the true value T, so error cannot be directly found. δ_{S} is error in the simulation output S, and δ_{D} is error in the experimental observation D. Without knowing true value T, error cannot be found, but the difference in errors, or comparison error E, is conceivable [1].

$$E = \delta_S - \delta_D = S - D \quad (1)$$

Thus, the sign and magnitude of the comparison error is known once the experiment and simulation outputs are obtained. The simulation error can be decomposed into three error sources.

$$\begin{cases} \delta_S = \delta_{model} + \delta_{num} + \delta_{input} \\ \delta_{model} = E - (\delta_S - \delta_D) = E - (\delta_{num} + \delta_{input} - \delta_D) \end{cases} \tag{2}$$

Model error is the simulation's error caused by its modeling assumptions. An estimate of model error is the deliverable result of the validation study [1].

Let uncertainty u be an estimate of unknown error δ. It is assumed that any known errors have already been corrected, so the sign and magnitude of error is unknown. So u estimates an interval for δ such that $\delta \in [-u, u]$. Uncertainty u is analogous to one standard deviation of the population from which error is realized [1]. Standard deviation is a familiar estimator for estimating uncertainty.

Recall model error is $E - (\delta_{num} + \delta_{input} - \delta_D)$. Each error δ is estimated with its corresponding uncertainty as an interval, $\delta \in [-u, u]$. Thus, an arithmetic sum of these errors is estimated by combining these uncertainty intervals into a term called validation uncertainty. Uncertainties are combined with the square root of the sum of the uncertainties squared.

$$u_{val} \equiv \pm\sqrt{u_{num}^2 + u_{input}^2 + u_D^2} \tag{3}$$

Throughout the standard's procedure, estimates for $u_{num}, u_{input}, u_D, \delta_{model}$ are determined, each following the ISO guidelines. Then model error may be estimated [1].

$$\delta_{model} \in E \pm u_{val} \tag{4}$$

Model error, also called model form error, is error due to built-in model assumptions and approximations.

2.1.3.i. Numerical Uncertainty (V&V Section 2)

Numerical uncertainty is the estimation of simulation error caused by the numerical solver. Estimation of numerical uncertainty requires a grid convergence study, also called mesh independence study. Three to six different grids must be generated to study the solution's dependence on grid resolution. The grids should be asymptotically converging for these calculations to be useful. The observed order of accuracy is calculated with the grids' simulation results. A factor of safety is required for calculation of GCI, grid convergence index. GCI is a metric for how converged the grids are. The factor of safety is lower for grids which were refined in a structured fashion. Structured grid refinement requires that a grid meshing is reduced by the same factor in all physical dimensions in the model. An expansion factor is applied to account for numerical error distribution assumptions. The expansion factors recommended depend on the convergence behavior: some problems are poorly behaved and oscillatory while others are well-behaved and smooth [1].

This process must precede the input uncertainty analysis. This process may reveal needs for model improvements or a finer grid. After this evaluation, an appropriately resolved grid should be chosen for input uncertainty analysis. If the coarsest grid used for grid convergence is used for input uncertainty analysis, the input uncertainty analysis may not be accurate. If input and numerical uncertainties are similar in magnitude, then the finest grid from the grid convergence study should be used for input uncertainty analysis. If the numerical uncertainty is significantly smaller than the input uncertainty, then a coarser grid is acceptable for input uncertainty analysis [1].

2.1.3.ii. Input Uncertainty (V&V Section 3)

Input uncertainty is the estimation of simulation error due to the uncertainty of input parameters. First, pertinent simulation input parameters must be identified, which is a somewhat arbitrary selection and must be done with engineering intuition and expertise. Inputs which aren't included in this set of "input parameters" are then considered hard-wired into the model and are thus accounted for in model (form) error. Inputs which may be left out of the set, for instance, would be turbulence model choice, meshing statistics, or convergence criteria. Since turbulence model choice is not a numerical input, it cannot be evaluated with this method. So this is becomes a property of the model. Meshing and convergence specifications should be captured in numerical uncertainty.

Next, the measurement or "standard" uncertainty of each simulation input parameter must be obtained. This value is found by either a manufacturer's specifications or with a validation experiment. In a validation experiment, a parameter is measured many times, and the standard deviation of the readings serves as an estimate of the parameter's measurement uncertainty. This means that the reading X should be considered $X \pm u_X$ due to the measurement instrument's inconsistency.

Then, these experimental parameter uncertainties are scaled by sensitivity coefficients, or the simulation's sensitivity to each input parameter. Sensitivity is an inherent property of the model's behavior with respect to that parameter. The sensitivity coefficients can be found through various methods. Local and global methods are presented in V&V-20 [1]. The global method involves sampling. The local method specifically suggested in V&V-20 [1] is the finite difference approximation, also called perturbation method, mean value method, etc. First and second order are considered sufficient. This study uses the local method and is thus focused on its explanation. Each parameter should be perturbed, one at a time, and the simulation result is stored. The size of the perturbation is the primary difficulty in executing this method. If the perturbation is too large then parameter discretization occurs (the approximation is inaccurate), and if the perturbation is too small, then subtractive cancellation occurs (round-off error). Multiple perturbations should be performed, and the user must select a perturbation from an interval where the sensitivity coefficient is somewhat constant. Then the first or second order finite difference estimation of the sensitivity coefficient is obtained. The input uncertainty should be compared to the numerical uncertainty to guarantee the grid choice was valid [1].

2.1.3.iii. Experimental Uncertainty (V&V Section 4)

Experimental uncertainty is the uncertainty of the experimental result. This process requires uncertainties of the experimental error sources. The random sources and the systematic sources are combined separately, and then all combined into the uncertainty of the experimental result. Due to the complexity of measurements needed, it is advised that the experimentalist and validation leader work concurrently. ASME PTC 19.1-2005 "Test Uncertainty" should be referred to for a more thorough explanation of this procedure [1].

For the present study, no details of experiment uncertainties were reported except the experimental result, itself. Therefore, the procedure in the standard for estimating experimental uncertainty was not utilized, and the experimentalist is trusted. This choice may be defended by the standard's claim that each section may be viewed independently [1].

2.1.3.iv. Model Error (V&V Section 5 & 6)

Model error is simulation error due to built-in model assumptions and approximations. Model error is somewhat a catchall for uncertainties which have not been estimated in the numerical and input uncertainty sections. Effects which are not captured in those sections are considered inherent to the model and thus fall under model error. Validation uncertainty is calculated with Equation (3) for situations where the validation variable QOI is directly measured. This is called the local method. For QOIs which are the result of data reduction of other outputs, a global method with sampling is required which combines input and experimental uncertainty [1].

E and u_{val} are validation metrics. If comparison error is much greater than validation uncertainty, then $\delta_{model} \approx u_{val}$. This reveals potential opportunities for improving the model. However, if $E \leq u_{val}$, then model error is within the noise of the model's error sources, and opportunities for improvement are less clear [1].

2.2. Case Study

The experiment used in this work was published by Kaneko et al [25]. Various annular pressure seal geometries were tested with water as the working fluid. In industry, these seals are used to contain high pressure fluids in high speed machinery such as compressors and pumps. Annular seals avoid contact between the rotating shaft and the stationary components around it, so perfect fluid containment is infeasible. Leakage is the measure of the flow out of the seal and has a direct impact on machine efficiency. Leakage is to be mitigated by designers without imposing too much friction on the rotor.

The leakage of the straight, or smooth, seal is the focused response of this validation study. A smooth seal is used for this demonstration due to its simple flow geometry, which allows for faster convergence, and the goal herein is to demonstrate a simulation-intensive procedure. Other quantities could be analyzed in a validation study such as stiffness or damping. Leakage was chosen because it is a straightforward model output for both models studied.

2.2.1. KANEKO EXPERIMENT

The experiment used for this validation, by Kaneko et al [25], involved the testing of various seals. Figure 3 is a simplification of the seal test rig diagram. The horizontal dashed line is the axial centerline of the test rig's rotor. The vertical dashed line represents a plane of symmetry: the rig tests two nominally identical seals at once, and they share an inlet.

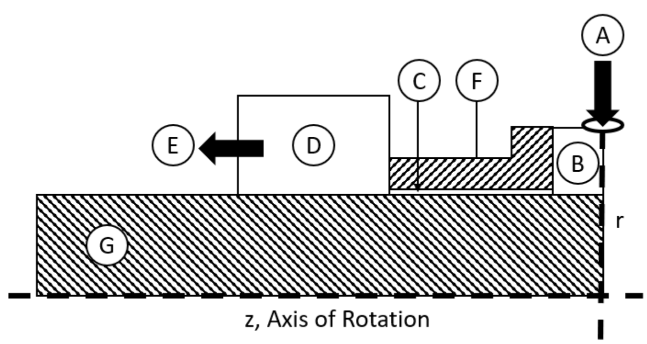

FIGURE 3
TEST RIG ADAPTED FROM [25]: (A) INLET (B) INLET REGION (C) CLEARANCE REGION (D) OUTLET REGION (E) OUTLET (F) SEAL (G) ROTOR

For the smooth seal, Kaneko, et al [25] report the rotor diameter to be 71.379 mm, the seal clearance to be 0.168 mm, and the seal length to be 60 mm. With these reported dimensions and assuming the figure representing the test rig is to scale, the inlet region is estimated to be 20.9 mm axially and 19.7 mm radially, and the outlet region is estimated to be a 34.1 mm in both directions. Similarly, each inlet is approximately 16.5 mm in diameter. There are two inlets, but they feed to two seals, so the area of one represents the area of the inlet flow to the seal. The actual inlets used appear to be pipes located at opposite sides of the cylindrical inlet region.

Leakage is simply the volumetric flow rate leaving the seal from the outlet, which is a standard quantity to express an annular seal's performance characteristics. No densities or temperatures are given for this published experiment. Figure 4 is the leakage at various speeds for the different seals tested by Kaneko et al [25].

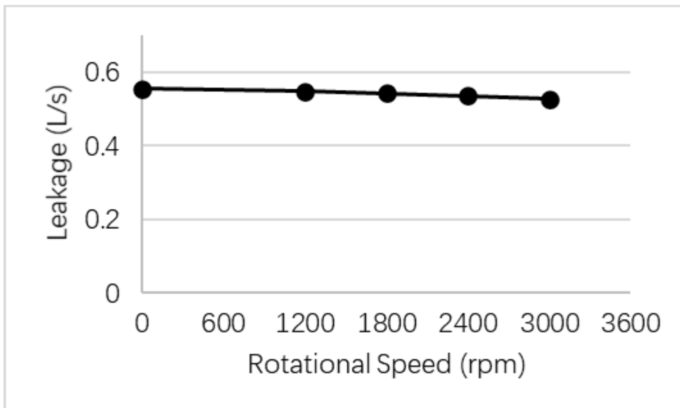

FIGURE 4
LEAKAGE FROM KANEKO EXPERIMENT, ADAPTED FROM [25]

The operating point for this study is 3000 rpm, and the leakage from Figure 4 is about $\mathrm{D} = 0.528\ \mathrm{L}/s$. The reported pressure differential across the seal length, 60 mm, is $\mathrm{P_{static}} = 784\ \mathrm{kPa}$ [25].

2.2.1.i. Experimental Uncertainty

V&V-20 [1] explains the experimental efforts required for calculating experimental uncertainty with validation experiment data. A validation experiment involves repeating measurements inputs and outputs in the experiment to find systematic and random uncertainties. Kaneko, et al [25] does not report such data. Ideally, the validation analyst and experimentalist are working together to design the experiment [1]. Since the sections of V&V-20 [1] may be considered independently, for experimental uncertainty, this validation study used the uncertainty reported from the experimentalists, which followed an ASME/ANSI method found in [25], [29]. The experiment reports 5% uncertainty in leakage [25]. The leakage at the validation point is D=0.528 L/s, so the dimensional experimental uncertainty is 0.0264 L/s.

2.2.2. ANSYS MODEL

Since the purpose of this study is to investigate V&V-20 [1] with respect to CFD modeling of turbomachinery components, a CFD model was built to emulate the Kaneko, et al [25] experiment.

2.2.2.i. Geometry

The seal flow region is assumed to be axisymmetric, and the seals are assumed identical. For this reason, a sector of the flow was modelled with symmetry boundary conditions on the faces. The sector angle was $1°$, which allowed for reasonable aspect ratios between the dimensions of mesh elements, and ANSYS CFX does not support two-dimensional geometry. The CFD model includes an inlet region, clearance region, and outlet region. The ANSYS model instead assumes an axisymmetric inlet ribbon with the same area as one inlet pipe. The width $\mathrm{w_{inlet}}$ of this equivalent, axisymmetric inlet ribbon was found with Equation (5).

$$2\pi(R_{rotor} + R_{inlet.region})w_{inlet} \equiv \pi \frac{d_{inlet}^2}{4} \tag{5}$$

where $\mathrm{R_{rotor}}$ is the radius of the rotor, $\mathrm{R_{inlet.region}}$ is the radial height of the inlet region, and $\mathrm{d_{inlet}}$ is the inlet feed hole diameter from the diagram. Then the width of the inlet ribbon is $\mathrm{w_{inlet}} = 0.61\ \mathrm{mm}$ wide.

The final geometry of the fluid flow region is shown in Figure 5. Relative to the clearance regions, the inlet and outlet regions are very large. Therefore, the exact sizes of the inlet and outlet regions typically have little effect on the flow through the clearance region.

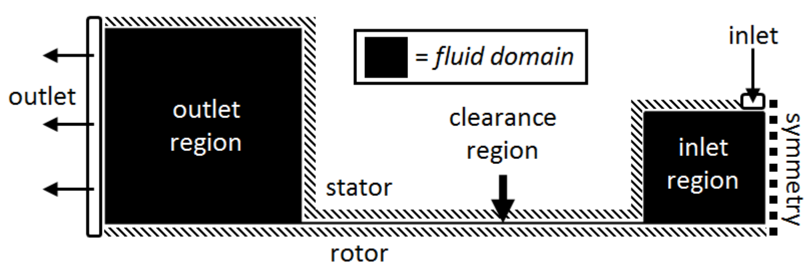

FIGURE 5

GEOMETRY OF CFD MODEL (INLET RIBBON NOT TO SCALE). VERTICAL= RADIAL DIRECTION, HORIZONTAL=AXIAL DIRECTION, NORMAL TO PAGE=CIRCUMFERENTIAL DIRECTION

2.2.2.ii. Boundary Conditions

The fluid behavior at any cross section of the annulus is assumed to be identical. Thus, these two faces are connected as a domain interface with rotational symmetry. Also, since the actual test rig is symmetrically testing two identical seals which share an inlet, the edge of the inlet region is a symmetry line, shown in Figure 5.

The hatched region above the fluid domain in Figure 5 is the stator, which is a stationary wall containing the fluid. On the actual test rig, the seal is the wall against the clearance region. Against the inlet and outlet regions are parts built into the test rig. The hatched region below the fluid domain in Figure 5 is the rotor, or a rotating wall containing the fluid from the bottom. The rotor is specified to rotate at 3000 rpm.

Dynamic pressure of the fluid is at the rotor surface is

$$P_{dynamic} = \frac{1}{2}\rho v_{axial}^2 = 97.5\ kPa \tag{6}$$

where v_{axial} is the axial velocity of the fluid through the seal clearance, and ρ is the density of the fluid. $v_{axial} = Q/A$ where Q is the reported leakage and A is the area of the annulus around the rotor. Therefore, the total pressure at the inlet is $P_{total} = P_{static} + P_{dynamic} = 881.5\ kPa$. This total pressure is the boundary condition applied to the inlet. The outlet has an average static pressure of 0 kPa.

2.2.2.iii. Turbulence Modeling

The Reynolds number at 3000 rpm is

$$Re = \frac{D_H v_{circ}}{\mu/\rho} \approx 4{,}225 \tag{7}$$

where D_H is the hydraulic diameter, v_{circ} is the circumferential velocity, μ is the dynamic viscosity, and ρ is density. For flow in a seal, hydraulic diameter $D_H = 2c$ is twice the clearance. Circumferential velocity $v_{circ} = (2\pi)R\Omega$ is the linear velocity of the flow around the rotor at the rotational speed where Ω is rotational speed and R is rotor radius. 3000 rpm rotational speed was selected to avoid the transition region at the lower speeds. According to [30], turbulence begins as early as $Re > 1800$. For fluid properties, since the temperature of the water is not reported, Standard Conditions for Temperature and Pressure (STP) are assumed, $25°C$.

Two common turbulence models are SST [31]–[33] and k-epsilon [34]–[36]. The turbulence model used imposes limitations on acceptable Y^+ values. Y^+ is a dimensionless distance from the wall to a point in the flow.

$$y^{+} = \frac{y\,u_{\tau}}{\nu} \tag{8}$$

where y is distance from the wall, ν is kinematic viscosity, and u_{τ} is the friction velocity (velocity close to the boundary) [37].

In numerical modeling, Y^{+} refers to the thickness of the first element of a mesh along a wall, for this model, the stator and rotor. Y^{+} values below 5 are compliant with the SST turbulence model and must be larger than 30 for the k-ε turbulence model [38]. The initial Y^{+} observed for this model was around 10, which is not compliant with SST or k-epsilon. Since this model was going to be run on finer meshes than the initial mesh generated, SST was chosen.

2.2.2.iv. Meshing

A structured mesh was chosen for this model because the geometry is logically rectangular. Three radial sweeps were used to build the mesh, shown in Figure 6. The inlet region and outlet region were each meshed with a sweep, both above the clearance region, and the clearance region was meshed with a sweep, extending into the inlet and outlet regions. Across the various meshes generated, the maximum mesh element size ranged from $0.2\ \mathrm{mm}$ to $0.9\ \mathrm{mm}$.

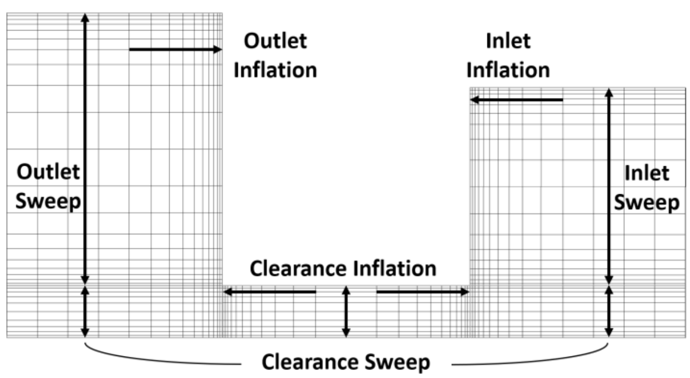

FIGURE 6
CONCEPTUAL DIAGRAM OF MESH (NOT TO SCALE): BIDIRECTIONAL RADIAL MESH SWEEPS AND UNIDIRECTIONAL AXIAL INFLATION LAYERS

To keep Y^{+} low, elements touching the stator and rotor need to be small. However, it would be a waste of computational efforts to assign the whole region to have an equally fine mesh. So, a bidirectional bias was imposed on each sweep to prefer smaller elements towards the connecting faces of the flow domain parts. These bias sweeps control the Y+ along horizontal edges of the rotor and stator. To maintain low Y+ values along vertical edges, a similar bias was imposed in the axial direction with a tool called inflation layers.

A growth rate is the ratio between two elements' thicknesses, larger to smaller. The growth rate was kept below 1.2 for a smooth transition. Thus, the thickness of elements along the interface between the clearance region sweep and the inlet and outlet sweeps must be within a factor of 1.2. For simplicity, all three sweeps have the same first layer height. For a swept section with n divisions, the bias is defined as

$$\begin{cases} B = \dfrac{xg^{n-1}}{x} = g^{n-1} \\ 1 \leq g \leq 1.2 \end{cases} \tag{9}$$

where g is the growth rate and x is the first layer height. Within one sweep, the growth rate is constant. If a region of height H is to be meshed with a bidirectional sweep, then H is the sum of all of the elements' thicknesses, which may be expressed as two identical geometric series.

$$H = 2\sum_{i=0}^{n-1} xg^{i} = x\frac{1-g^{n}}{1-g} \tag{10}$$

All of these mesh relations must be satisfied. In order to keep Y+ below 5 while changing the total number of elements, the growth rate, the number of divisions, and the first layer height were adjusted accordingly. The number of divisions and bias were directly specified for each region in ANSYS CFX, and the first layer height was driven by these two specifications. A total of 13 compliant meshes were used in this study which obey Equations (9) and (10). Details can be found in Appendix A: ANSYS Mesh Parameters.

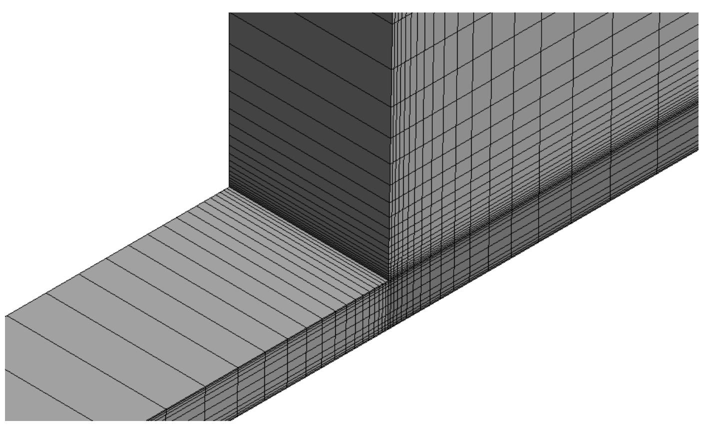

FIGURE 7

COARSEST MESH IN DETAIL: CLEARANCE & OUTLET REGION SWEEP, OUTLET REGION INFLATION

Figure 7 gives an example mesh with one element in the circumferential direction.

2.2.2.v. Solver

Water is the working fluid with constant fluid properties at Standard Temperature and Pressure (STP). The simulation was assumed to be isothermal with an SST turbulence model. The residual target for root mean square of all conservation equation residuals is 1e-5. A conservation target of 0.01 is enforced to ensure the differences between inflow and outflow of conservation properties through the boundaries is less than 1%.

2.2.3. BULK-FLOW MODEL

The bulk-flow smooth seal code described here calculates multiple outputs such as rotordynamic coefficients and leakage. However, the output of interest for this study is leakage, or flow out of a seal.

$$\begin{cases} \dot{m} = \rho u_{outlet} A_{outlet} \\ \dot{V} = u_{outlet} A_{outlet} \end{cases} \tag{11}$$

where ρ is fluid density, u_{outlet} is axial flow through the outlet, and A_{outlet} is the cross-sectional area of the outlet. The focus for leakage evaluation in practice is typically mass flow $\dot{m}$. Volumetric flow $\dot{V}$ will be used for this study because the experiment reports volumetric flow and does not report temperature or fluid properties.

2.2.3.i. Geometry & Meshing

Only the clearance region is modeled for this code. An inlet loss coefficient is assumed to account for the change in pressure from inlet to clearance region. This is a one-dimensional code, so the mesh elements or control-volumes are along the length of the seal. The user specifies a number of elements or control-volumes, and the clearance region is evenly divided into that number of regions.

2.2.3.ii. Boundary Conditions

The circumferential velocity of the rotor is 3000 rpm. This boundary condition is a user input to the code which drives the circumferential flow. Static pressure is a user input which drives the axial flow, given by Kaneko et al [25] to be 784 kPa.

2.2.3.iii. Solver

The bulk flow method breaks the fluid region into multiple discrete control-volumes, slicing along the axial direction. The smooth seal code evaluates the fluid flow in a smooth seal by solving the continuity equation and conservation of axial and circumferential momentum equations. Radial velocity is assumed to be negligible in this small clearance flow region. The continuity equation is

$$0 = \frac{\partial h}{\partial t} + \frac{\partial (uh)}{\partial z} + \frac{1}{r}\frac{\partial (wh)}{\partial \phi} \tag{12}$$

The conservation of axial momentum equation is

$$-h\frac{\partial P}{\partial z} - \tau_{sz} - \tau_{rz} = \rho h\left(\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial z} + \frac{w}{r}\frac{\partial u}{\partial \phi}\right) \tag{13}$$

The conservation of circumferential momentum equation is

$$-\frac{h}{r}\frac{\partial P}{\partial \phi} - \tau_{s\phi} - \tau_{r\phi} = \rho h\left(\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial z} + \frac{w}{r}\frac{\partial w}{\partial \phi}\right) \tag{14}$$

where u is axial velocity, w is circumferential velocity, h is fluid thickness, r is radius, P is pressure, τ is shear stress (subscripts r and s refer to rotor and stator). For solvability, the code assumes a linear perturbation form for components of the driving equations. Axial and circumferential velocities are assumed to be

$$\begin{cases} u = u_0 + \frac{\epsilon}{c}(u_{re} + iu_{im})e^{i(\Omega t+\phi)} \\ w = w_0 + \frac{\epsilon}{c}(w_{re} + iw_{im})e^{i(\Omega t+\phi)} \end{cases} \tag{15}$$

The fluid thickness is assumed to be

$$h = c - \epsilon e^{i(\Omega t+\phi)} \tag{16}$$

Pressure is assumed to be

$$P = P_0 + \frac{\epsilon}{c}(P_{re} + iP_{im})e^{i(\Omega t+\phi)} \tag{17}$$

Shear stress is assumed to be

$$\begin{aligned} \tau_{sz} &= \frac{1}{2}\rho u\lambda_s\sqrt{u_0^2 + w_0^2} \\ \tau_{rz} &= \frac{1}{2}\rho u\lambda_r\sqrt{u_0^2 + (w_0^2 - R\omega)^2} \end{aligned} \tag{18}$$

where ϵ is eccentricity, c is clearance, R is rotor radius, Ω is whirl speed, ω is rotor speed, and λ is the friction factor. Friction factors are typically estimated by empirical data or approximations. Blasius and Moody approximations for the friction factor is of the form

$$\lambda = aRe^{-b}$$

$$\lambda = 0.0055\left[1 + \left(2 \cdot 10^4 \frac{e}{c} + \frac{10^6}{Re}\right)^{\frac{1}{3}}\right] \tag{19}$$

$$\xRightarrow[b=0.217]{e=0} a = \lambda Re^b = 0.0055\left[1 + 100 \cdot Re^{-\frac{1}{3}}\right] Re^{0.217}$$

where Re is the Reynold’s number, e is eccentricity, c is clearance, and a and b are empirically found constants. For this experiment, eccentricity is zero for leakage measurements. Hirs found $a = 0.0674, b = 0.217$ describes the friction factor well for fluid flow in rotating machinery [39]. The friction factor coefficient a is an adjustable coefficient in the smooth seal code as this is the parameter most often adjusted to improve models. Removing the friction factor term through development of a 2D seal code is the subject of current research in our lab. More details on this smooth seal bulk-flow code can be found in references [40], [41].

Chapter 3: Numerical Uncertainty

3.1. Overview

Numerical uncertainty is an estimate of error in simulation output due to the numerical scheme of the CFD solver. V&V-20 [1] recommends the Grid Convergence Index method for estimating numerical uncertainty, which is represented below.

Three meshes, or grids, of sufficiently different sizes are needed to test convergence. Define a representative mesh size $\mathrm{h_i}$ for a 3D model as

$$h_i = \begin{cases} \left(\frac{V}{N_i}\right)^{1/3}, & 3D\ model \\ \left(\frac{A}{N_i}\right)^{1/2}, & 2D\ model \\ \left(\frac{L}{N_i}\right), & 1D\ model \end{cases} \tag{20}$$

where $\mathrm{V, A, L}$ are the volume, area, and length of the fluid domain and $\mathrm{N_i}$ is the number of elements in the $\mathrm{i^{th}}$ mesh. $\mathrm{h_i}$ physically represents the average size of an element from a mesh in one dimension [1]. Meshes 1,2, and 3 are ordered finest to coarsest mesh where $\mathrm{h_1 < h_2 < h_3}$. Define the ratio between one grid and a finer grid to be

$$r_{i+1,i} = \frac{h_{i+1}}{h_i} \tag{21}$$

The meshes must be sufficiently different in size to test convergence, so V&V-20 recommends the ratio between mesh sizes to be $\mathrm{r_{i+1,i}} \geq 1.3$ [1]. After at least three compliant meshes have been created, run the simulation to collect the quantity of interest S for each mesh. Relative error between simulations is

$$\varepsilon_{i+1,i} = S_{i+1} - S_i \tag{22}$$

Observed order of convergence p is the order of the error in a numerical scheme as it approaches zero with finer resolution. Observed order of convergence may be estimated empirically with the output of various mesh sizes. Solution error is proportional to h^p such that

$$S_i - S_{exact} = Ch^p + H.O.T. \tag{23}$$
$$\Rightarrow log(S_i - S_{exact}) \approx p\, log(h) + log(C)\,, \qquad H.O.T. \approx 0$$

This relationship can be approximated by fitting a line to $log(S_i - S_{exact})$ vs $log(h)$ and the slope approximates p. Since S_{exact} is not affecting the slope p, this value is not needed. If $r_{21} = r_{32}$, then the calculation of p is direct with three grid solutions, shown in Equation (24) with $q = 0$. Otherwise, Richardson's extrapolation is needed, which involves expressing S as a Taylor expansion of h. This results in the following nonlinear system of equations [42].

$$\begin{cases} p = \dfrac{ln|\varepsilon_{32}/\varepsilon_{21}| + q}{ln(r_{21})} \\ q = ln\left(\dfrac{r_{21}^p - s}{r_{32}^p - s}\right) \end{cases} \tag{24}$$

where $\mathrm{s} = 1 * \mathrm{sign}(\varepsilon_{32}/\varepsilon_{21})$ [1]. These equations are solved simultaneously. This (observed) order of convergence formulation can only estimate the convergence of three points at a time and therefore only assesses their convergence with respect to each other. The Grid Convergence Index GCI is

$$GCI = \frac{F_s|\varepsilon_{21}|}{r_{21}^p - 1}$$
$$F_s = \begin{cases} 1.25, & structured\ grid\ refinement \\ 3, & unstructured\ grid\ refinement \end{cases} \tag{25}$$

where F_s is a factor of safety whose values were empirically determined from several hundred case studies with a 95% confidence [1]. Structured grid refinement preserves the geometry of each element from grid to grid. In other words, all physical dimensions are altered by the same amount ($r_x \approx r_y \approx r_z$).

Recall from section 2.1.3. , u_{num} is an estimate of one standard deviation of the random error imposed by numerical error. To reach an estimate of standard deviation, assumptions must be made about the shape of the numerical error distribution. GCI is an estimate of length of a 95% confidence interval of numerical error, based on the empirical observations which led to Fs. An expansion factor k is applied to GCI to estimate the standard deviation, u_{num} [1].

$$u_{num} = \frac{GCI}{k}$$
$$k = \begin{cases} 1.15, & well\ behaved\ convergence \\ 2, & poorly\ behaved\ convergence \end{cases} \tag{26}$$

A poorly behaved problem (oscillatory convergence) is assumed to have a Gaussian error distribution. A well-behaved problem (smooth, monotonic convergence) is assumed to have a shifted Gaussian error distribution [1].

A MATLAB Code was developed for this analysis, found in Appendix C.III. Numerical Uncertainty Analysis (general).

3.1. ANSYS Model

13 meshes were generated and the simulation was run to collect the quantity of interest for each mesh. Figure 8 shows the simulation results (leakage) for all grids. The grids were generated with a structured grid refinement ($r_x \approx r_y \approx r_z$). Details of the structured grid refinement and mesh parameters can be found in Appendix A: ANSYS Mesh Parameters. Since the convergence is not monotonic in Figure 8, this problem is assumed to be poorly behaved.

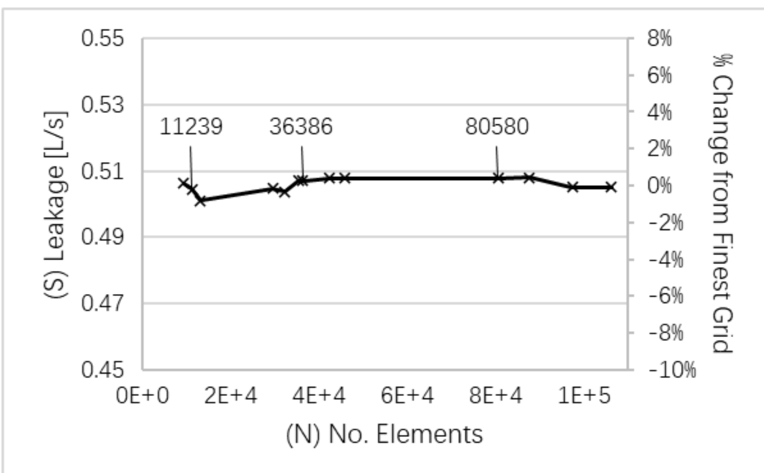

FIGURE 8
LEAKAGE ANSYS OUTPUT FOR 13 DIFFERENT GRIDS

The ANSYS model consistently estimates leakage to be near 0.505 L/s. In the ANSYS model, the volume of the seal sector's fluid region is 1410 mm^3, which is applied to Equation (20) to find representative mesh sizes, h_i. Since 13 different meshes were generated, there were 57 possible combinations of three meshes (grid triplets) of sufficiently different size ($r > 1.3$).

One grid triplet was chosen for the remainder demonstration; this selection is discussed in Special Considerations. These grids' results are shown in Table 1, following Equations (21) and (22).

TABLE 1
CONVERGENCE METRICS RESULTS

Grid	N	h	r	S	eps
1	80580	0.230		0.50776	
			1.30		-7.97E-4
2	36386	0.300		0.50697	
			1.48		-2.48E-3
3	11239	0.444		0.50449	

Table 2 shows the observed order of convergence p, the associated term q, and the grid convergence index GCI following Equations (24) and (25). The results in Table 2 are based on the grid triplet shown in Table 1. Since the grid refinement was structured, $F_s = 1.25$ is used for GCI in Equation (25).

TABLE 2
CONVERGENCE METRICS RESULTS

p	q	GCI
2.21	-5.47E-1	1.25E-3

The convergence in in Figure 8 is poorly behaved, so $k = 2$ is used for u_{num} in Equation (26). Finally, for the set of grids in Table 1, the numerical uncertainty is $u_{num} = 0.0006251\ L/s$, which estimates numerical error $\delta_{num} \in [-u_{num}, u_{num}]$.

3.1. Bulk-Flow Model

49 meshes were generated and the simulation was run to collect the quantity of interest for each mesh. Figure 9 shows the simulation results (leakage) for all grids. Since this model is one-dimensional, only the number of nodes is specified, which is naturally a structured grid refinement. Since the convergence is smooth and monotonic in Figure 9, this problem is assumed to be well behaved. A MATLAB code was developed for running a mesh independence study on this code, found in Appendix C.I. Mesh Independence Study (example).

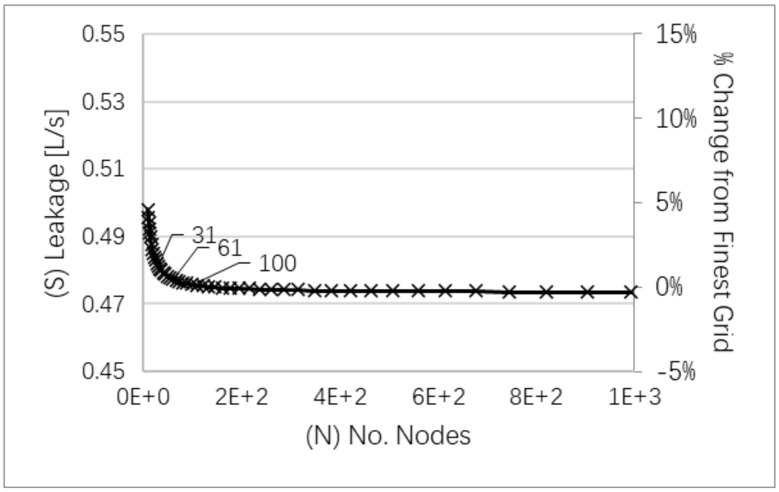

FIGURE 9
LEAKAGE OUTPUT FOR 49 DIFFERENT GRIDS

The bulk-flow model converges to a leakage of about 0.475 L/s. The length of the seal fluid region is the seal length, 60 mm, which is applied to Equation (20) to find representative mesh sizes, h_i. Since 49 different meshes were generated, there were 813 possible combinations of three meshes (grid triplets) of sufficiently different size ($r > 1.3$). One grid triplet was chosen for the remainder demonstration; this selection is discussed in Special Considerations. These grids' results are summarized in Table 3, following Equations (21) and (22).

Table 4 shows the observed order of convergence p, the associated term q, and the grid convergence index GCI following Equations (24) and (25). The results in Table 4 are based on the grid triplet shown in The length of the seal fluid region is the seal length, 60 mm, which is applied to Equation (20) to find representative mesh sizes, h_i.

TABLE 3
THREE COMPLIANT MESHES AND SIMULATION RESULTS

Grid	N	h	r	S	eps
1	100	0.006		0.47580	
			1.64		1.51E-3
2	61	0.010		0.47731	
			1.97		3.78E-3
3	31	0.019		0.48108	

Since 49 different meshes were generated, there were 813 possible combinations of three meshes (grid triplets) of sufficiently different size ($r > 1.3$). One grid triplet was chosen for the remainder demonstration; this selection is discussed in Special Considerations. Since the grid refinement was structured, $F_s = 1.25$ is used for GCI in Equation (25).

TABLE 4
THREE COMPLIANT MESHES AND SIMULATION RESULTS

p	q	GCI
1.01	-4.16E-1	2.91E-3

The convergence in Figure 9 is well behaved, so $k = 1.15$ is used for u_{num} in Equation (26). Finally, for the set of grids in Table 3, the numerical uncertainty is $u_{num} = 0.002528\ L/s$, which estimates numerical error $\delta_{num} \in [-u_{num}, u_{num}]$.

This evaluation is also done on the other simulation output variables, shown in Appendix B: Additional Bulk-Flow Results. To do this efficiently, each output variable was stored for all grid sizes. The analysis above was run for all outputs and for all grid triplets.

3.2. Special Considerations

3.2.1. DISCREPENCY BETWEEN MODELS

The bulk-flow model converges to a leakage of about 0.475 L/s while the ANSYS model consistently estimates leakage to be near 0.505 L/s. The experimental measurement was 0.528 L/s. There are many differences between these models which may account for the difference in their predictions.

The ANSYS model includes the inlet region and outlet region where the bulk-flow model does not. However, recirculation and radial flow in the inlet and outlet region should slow down the flow out of the seal, leakage. The higher leakage in the ANSYS model indicates that it loses less pressure than the Bulk-Flow model. This may be because total pressure is specified at the inlet so ANSYS determines the initial velocity. The Bulk-Flow model instead requires static pressure at the inlet. The inlet and outlet region geometries were estimated from a diagram. These dimensions could be off enough to affect the flow. Nonetheless, the ANSYS model is much more directly replicating the experiment by including the inlet and outlet regions. This could explain why its prediction is closer to the experimental value.

Another major difference between the models is ANSYS models turbulence with SST while Bulk-Flow models turbulence with the empirical Hirs model. This validation point is expected to be well into the turbulence regime, so the turbulence model choice plays a big role in predicting flow. For the ANSYS model, SST is used where the Bulk-Flow model uses the empirical estimates from the Hirs model. This could account for the Bulk-Flow's lower leakage, which is further from the experimental value.

3.2.2. SPREAD OF P

V&V-20 gives no recommendation on how to incorporate more than three grids in the numerical uncertainty quantification or how to deal with multiple uncertainty calculations. For each model, these calculations were done on all grid triplets. V&V-20 recommends running this calculation on more than one grid triplet to check if p is consistent. p should be close to the theoretical p. Comparing these is useful for assessing numerical error [1]. If p varies greatly across grid triplets, it may indicate

- The grid is not refined enough.
- There are unaddressed errors in the code.
- The boundary or initial conditions are unsatisfactory.
- The convergence is incomplete.
- The grid refinement is not structured enough.
- The simulations are not in the asymptotic region of convergence.

There are various suggestions for estimating p outside of the methods in V&V-20.

For the ANSYS model, the variance in p across grid triplets was unsatisfactory. Initially, the grid refinement was not structured, resulting in a huge spread of p, shown in Figure 10.

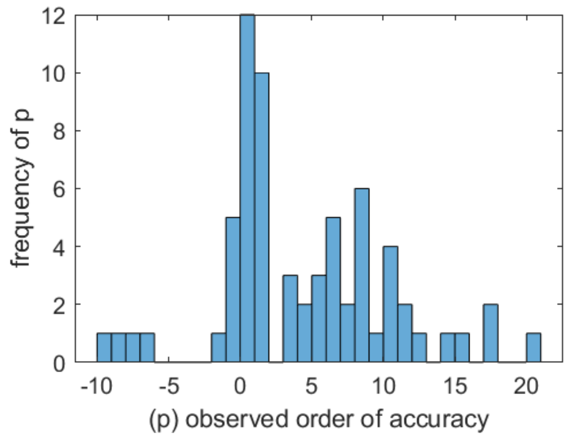

FIGURE 10
ANSYS: SPREAD OF P FOR UNSTRUCTURED GRID REFINEMENT

To address this issue, a structured grid refinement was developed, detailed in Appendix A: ANSYS Mesh Parameters. This reduced the spread of p, shown in Figure 11.

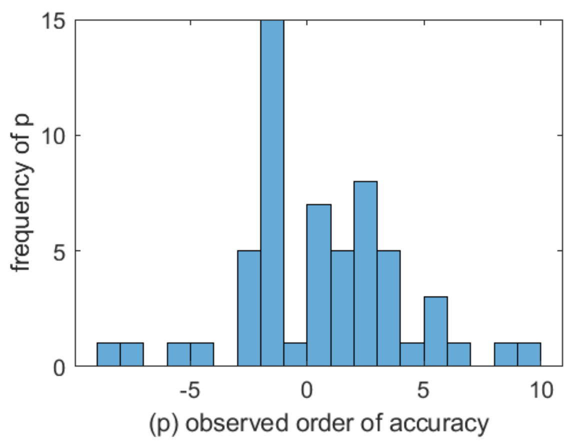

FIGURE 11
ANSYS: SPREAD OF P FOR STRUCTURED GRID REFINEMENT

The spread of p remained unsatisfactory but improved. ANSYS is a 2^{nd} order model, and the spread of p is bimodal with a sharp peak at $p \approx -1.5$ and a shallow peak at $p \approx 2.5$. The grid could not be easily refined further as finer meshes resulted in numerical overflow. The grid refinement was, in fact, structured. The simulation seems to be in the asymptotic region as its leakage predictions are consistently within 1% of one another.

In contrast, the spread of p was satisfactory for the bulk-flow model, shown in Figure 12. There was still a spread, but the spread was much smaller and the peak is at $p = 1$, which is the theoretical order of accuracy for this code.

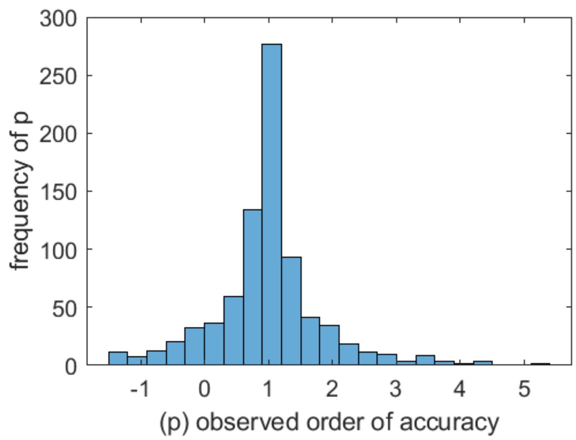

FIGURE 12
BULK-FLOW: SPREAD OF P

3.2.3. CHOOSING A GRID TRIPLET

The grid triplet used in the ANSYS numerical uncertainty calculations were chosen by first only considering grid triplets with the smallest N_1 grid ($\mathrm{N}_1 = 80580$ elements), allowing for efficient computations for input uncertainty. This was important for ANSYS due to the slow computations. Since ANSYS CFX is a second order scheme, the order of convergence should be near 2. So the grid triplet which best resembles the numerical scheme was chosen, with $\mathrm{p} \approx 2$.

The grid triplet used in the bulk-flow numerical uncertainty calculations were chosen by first only considering grid triplets with $\mathrm{N}_1 = 100$ nodes. This was chosen because $N = 100$ is at the beginning of the asymptotic region in Figure 9 and for the other Bulk-Flow output variables shown in Appendix B: Additional Bulk-Flow Results. This 100-node grid allowed for efficient but reliable output for input uncertainty quantification and running this procedure on other output variables. Since the ROMAC Smooth Seal code is a first order scheme, the order of convergence should be near 1. So the grid triplet which best resembles the numerical scheme was chosen, with $\mathrm{p} \approx 1$ for leakage and all output variables.

These two N_1 grids are deemed adequate for demonstration in the perturbation study for input uncertainty. V&V-20 remarks that its sections may be considered independently, so the following section is not tainted by the uncertain order of convergence of these grids. If $u_{input} \approx u_{num}$, then the finest grid from the grid triplet (N_1) should be used for estimating u_{input}. If $u_{input} \gg u_{num}$, then a coarser grid from the grid triplets may be used for estimating u_{input}. This cannot be checked until u_{input} is estimated, but we proceed using N_1 to be conservative, so no modifications will need to be made later on.

Chapter 4: Input Uncertainty

4.1. Overview

For a CFD model with m relevant input parameters, the input uncertainty $\mathrm{u_{input}}$ is

$$u_{input}^2 = \sum_{i=1}^{m} \left(u_{X_i} \cdot \frac{\partial S}{\partial X_i} \right)^2 \tag{27}$$

where $\mathrm{u_{X_i}}$ is standard uncertainty of input parameter $\mathrm{X_i}$, and $\frac{\partial \mathrm{S}}{\partial \mathrm{X_i}}$ is sensitivity coefficient of simulation output S with respect to input parameter $\mathrm{X_i}$ [1]. The sensitivity coefficient expresses how sensitive the simulation output is to the input parameter and is found by perturbing the input parameter. The input to the model should replicate the experiment, so uncertainties of experimental parameters are contributing to the uncertainty of the inputs to the model. To account for this, each input parameter's input uncertainty is the standard uncertainty from experiment scaled by the simulation's sensitivity to the parameter itself. The standard uncertainty should be found from the experiment, a database, or expert opinion [1]. Since uncertainties are combined as root sum of the squares, $\mathrm{u_{input}}$ is simply a combination of input uncertainty introduced by each input parameter.

To approximate the sensitivity coefficients $\frac{\partial S}{\partial X}$, second order finite difference is used

$$\frac{\partial S}{\partial X} \approx \frac{S(X + \Delta X) \; - S(X - \Delta X)}{2\, \Delta X} \tag{28}$$

The first order finite difference approximation may also be used here [1], but second order was used for a more robust estimation. Choosing an appropriate perturbation ΔX is a nontrivial task. If the perturbation is too small (ie, $0.1\ \mathrm{rpm}$ for a $3000\ \mathrm{rpm}$ input), then there will be no observable change in S due to the limitations of computer precision. This effect is called subtractive cancellation [1]. If the perturbation is too large (i.e., $1000\ \mathrm{rpm}$ for a $3000\ \mathrm{rpm}$ input) then the observed change in S is reflecting the change in the physics rather than the model's sensitivity; the value $\mathrm{S}(X + \Delta \mathrm{X})$ would no longer represent the same problem. This effect is called parameter discretization [1]. To determine an appropriate value for ΔX which avoids these effects, a wide range of values should be tested [1].

Normalizing metrics are useful in graphically assessing the perturbation study of all parameters. Let the order of the relative perturbation be O_X and the order of the relative sensitivity coefficient be O_S.

$$O_X = log\left(\frac{\Delta X}{X}\right)$$
$$O_S = log\left(X\frac{\Delta S}{\Delta X}\right) \tag{29}$$

These normalizations allow for a generalized, fair analysis of the parameters with graphs of the order of relative sensitivity coefficient against the order of each relative perturbation.

For a given model, let the precision of the output be $10^{-\mathrm{PS}}$, so when no change in output is observed, $\Delta \mathrm{S} = 10^{-(\mathrm{PS}+1)}$ is assumed, though it could be smaller. If no change in simulation output is observed,

$$O_S = log(\Delta S) - log\left(\frac{\Delta X}{X}\right) \tag{30}$$

to reflect model output precision, where $\Delta \mathrm{S} = 10^{-(\mathrm{PS}+1)}$ represents the precision of the results.

The simulations in this section should be on one of the grid triplets evaluated for numerical uncertainty. Choosing which grid to use is a somewhat iterative process. If $u_{input} \sim u_{num}$, then the finest grid should be used in this perturbation study. If $u_{input} \gg u_{num}$, then one of the coarser grids is adequate for the perturbation study.

A MATLAB Code was developed for this analysis, found in Appendix C.IV. Input Uncertainty Analysis (general).

4.2. ANSYS Model

This section quantifies input uncertainty for ANSYS Grid 1 shown in Table 1. This was chosen because u_{input} is not much larger than u_{num}, because of the noise in p, and to be conservative.

4.2.1. STANDARD UNCERTAINTY

The experiment used for this work did not report standard uncertainties; therefore, expert opinions were sought. Specifically, a ROMAC post-doc, graduate student, and practicing engineer were consulted; additionally, several publications were referenced [23],

[43], [44]. For this smooth seal sector model, there are twelve relevant input parameters. The three dimensions specified in the publication (seal length, seal clearance, and rotor diameter) are assigned standard uncertainty of $5\ \mu\mathrm{m}$. The five dimensions interpreted from the diagram in the publication (inlet and outlet regions' height and width along with the inlet's width) are assigned standard uncertainty of $5\ \mathrm{mm}$ (see Table 6 and [25]). The rotor rotational speed is estimated to have a standard uncertainty of $100\ \mathrm{rpm}$, and the inlet pressure has an estimated standard uncertainty of $5\ \mathrm{kPa}$. Since the temperature of the water was not given, the temperature was assumed to be between 10 and 50 degrees Celsius. The standard uncertainty of viscosity and density were approximated with the standard deviation of known values for water across this temperature range [45], [46]. Estimation of these standard uncertainties was somewhat subjective without the appropriate experimental data.

4.2.2. SENSITIVITY COEFFICIENT

The sensitivity coefficients are found with $\frac{\partial \mathrm{S}}{\partial \mathrm{X}} \approx \frac{\Delta \mathrm{S}}{\Delta X}$ by perturbing by ΔX. For this study, the perturbations were within $0.2 \geq \frac{\Delta X}{X} \geq 10^{-10}$. One realization of $\frac{\Delta \mathrm{S}}{\Delta X}$ for a particular $\frac{\Delta X}{X}$ is needed for input uncertainty estimation. The chosen value should be within an interval of $\frac{\Delta \mathrm{X}}{X}$ where $\frac{\Delta \mathrm{S}}{\Delta X}$ is stable or where the effect X has on S is constant. The leakage is reported with precision of 10^{-6}, so when no change in leakage output is observed, $\Delta \mathrm{S} = 10^{-7}$ is assumed, though it could be smaller.

Normalizing metrics from Equations (29) and (30) are useful in graphically assessing the perturbation study of all twelve parameters, again using the order of the relative perturbation be $\mathrm{O_X}$ and the order of the relative sensitivity coefficient be $\mathrm{O_S}$.

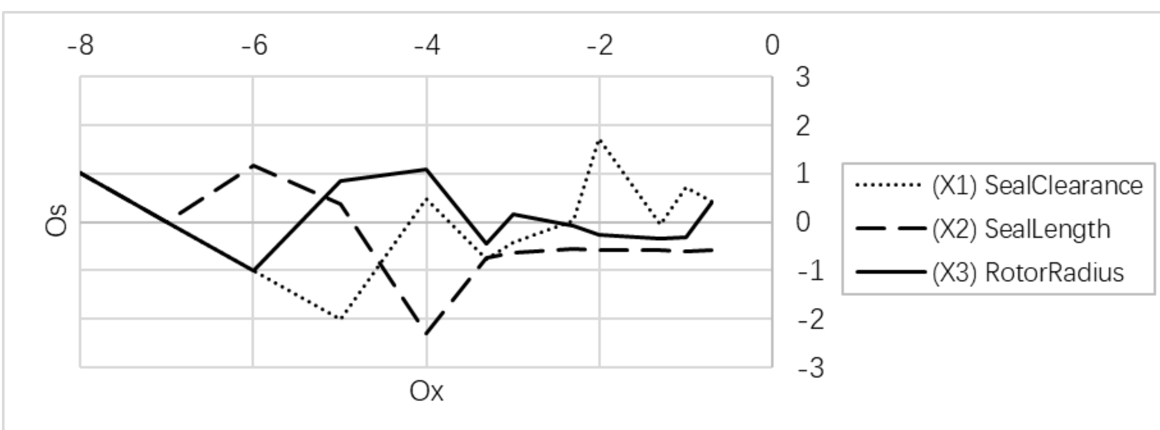

FIGURE 13
SENSITIVITY VS. PERTURBATION FOR X1-X3

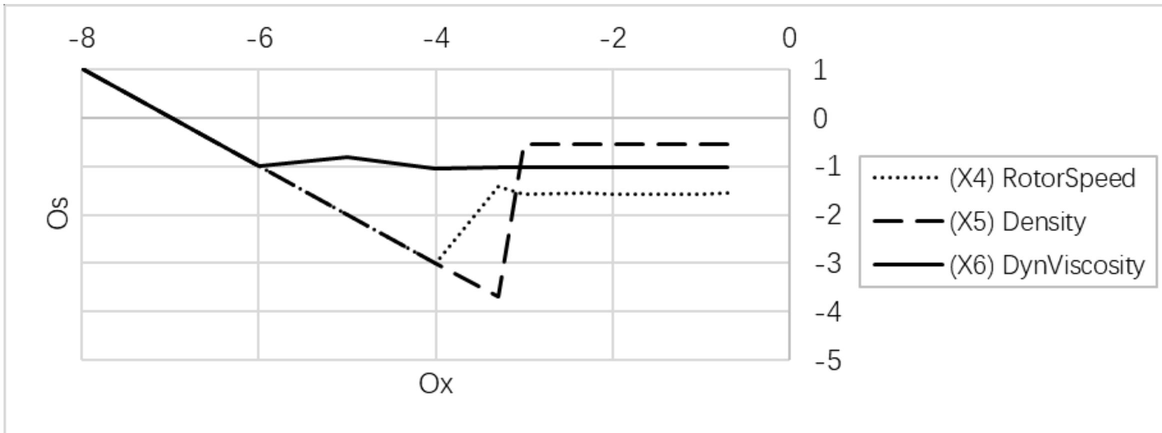

FIGURE 14
SENSITIVITY VS. PERTURBATION FOR X4-X6

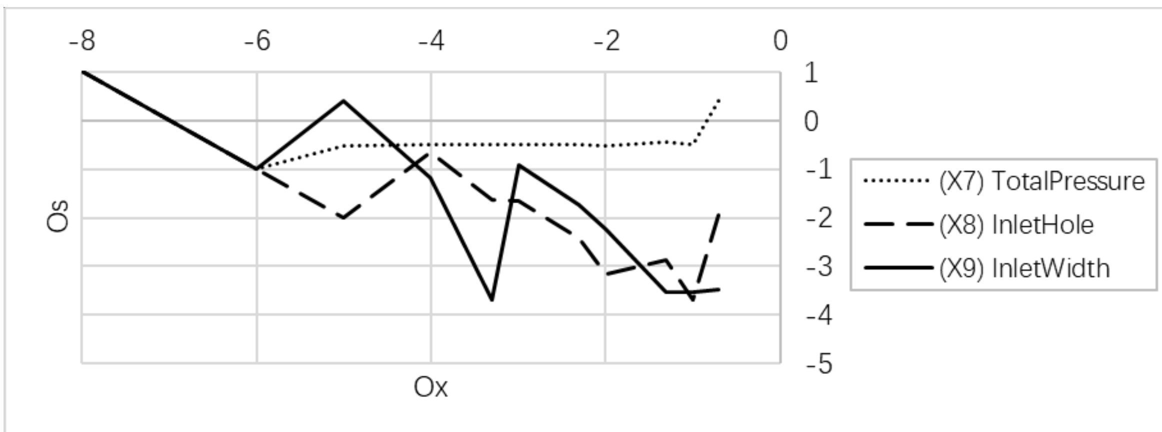

FIGURE 15
SENSITIVITY VS. PERTURBATION FOR X7-X9

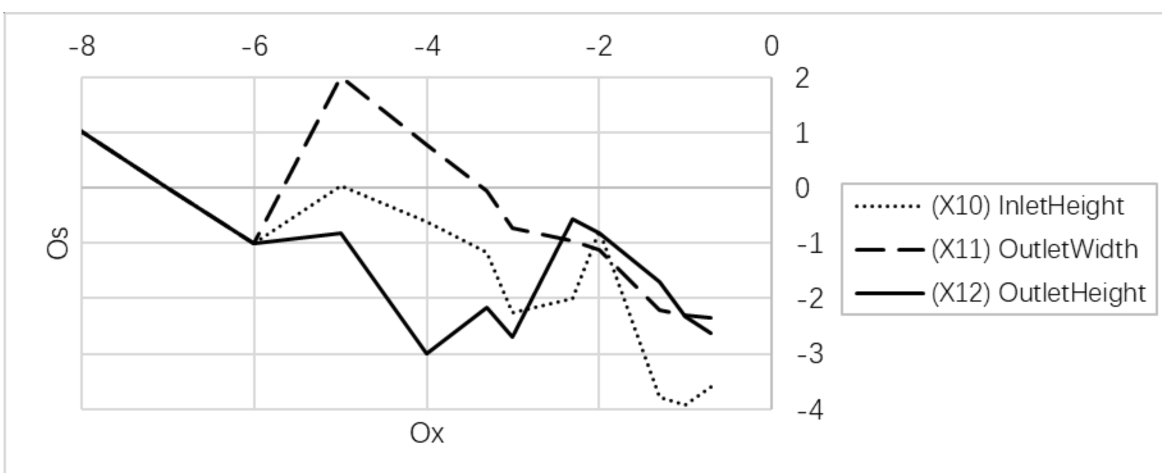

FIGURE 16
SENSITIVITY VS. PERTURBATION FOR X10-X12

Only one parameter was adjusted at a time, and each of the twelve parameters were adjusted bilaterally with 13 different perturbation sizes ΔX. This required 312 simulations, costing about half of a week of runtime. In order to do this efficiently, the relevant parameters were flagged in ANSYS so that they may be adjusted easily. A script was created in MATLAB which read an Excel sheet of input parameters; case by case, adjusted parameters in ANSYS; ran the simulations; and wrote the important results to a file. These results were used to approximate each derivative with Equation (28).

The results of the perturbation study are shown in Figure 13-Figure 16 for each parameter and each perturbation. The interval of $\mathrm{O_X}$ where the $\mathrm{O_S}$ is most stable is desired. This value will be trusted to represent the true estimation of the sensitivity coefficient, $\frac{\partial \mathrm{S}}{\partial \mathrm{X_i}}$. The O_x values are shown in Table 5.

TABLE 5
STABLE OX FOR EACH INPUT PARAMETER

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12
$\mathrm{O_X}$	-0.7	-2	-1.3	-2	-2	-2	-3	-1.3	-1	-2.3	-2.3	-5

For parameters X2, X4-X7, the stable regions are very distinct and wide, so any O_x along the flat part of each curve are acceptable. For parameters X1, X3, X8-X12, a stable region is not clear, but the above values were chosen as best estimates.

4.2.1. INPUT UNCERTAINTY

TABLE 6
INPUT UNCERTAINTY SUMMARY

Input Parameter		[u]	Xnom	Ux	dS/dX	Uinput
			[u]	[u]	L/s/[u]	L/s
Seal Clearance	**X1**	mm	0.168	0.005	1.53E+1	7.66E-2
Seal Length	**X2**	mm	60	0.005	4.32E-3	2.16E-5
Rotor Radius	**X3**	mm	35.6895	0.005	1.26E-2	6.30E-5
Rotor Speed	**X4**	rpm	3000	100	9.10E-6	9.10E-4
Density	**X5**	kg m-3	997	2.5	2.92E-4	7.30E-4
Dyn Visocity	**X6**	mPa s	0.8899	0.25	1.05E-1	2.63E-2
Total Pressure	**X7**	kPa	881.38	5	3.52E-4	1.76E-3
Inlet Hole	**X8**	mm	0.61	0.1	2.16E-3	2.16E-4
Inlet Width	**X9**	mm	20.9	5	1.44E-5	7.18E-5
Inlet Height	**X10**	mm	19.5	5	5.03E-4	2.51E-3
Outlet Width	**X11**	mm	34.1	5	3.15E-3	1.57E-2
Outlet Height	**X12**	mm	34.1	5	4.40E-3	2.20E-2

Table 6 summarizes each parameter's nominal value, standard uncertainty, sensitivity coefficient from analysis of Figure 13-Figure 16, and input uncertainty contribution. According to Table 6, seal clearance (X1), dynamic viscosity (X6), outlet width (X11), and outlet height (X11) are the dominating contributors to input uncertainty, which are on the order of 1E-2 L/s. All other contributions were on the order of 1E-3 L/s or smaller. As given by Equation (27), the uncertainty contributions of each parameter are combined into a total input uncertainty to be $u_{input} = 0.08550\ L/s$, which estimates input error $\delta_{input} \in [-u_{input}, u_{input}]$.

4.1. Bulk-Flow Model

This section quantifies input uncertainty with a perturbation study on Bulk-Flow Grid 1 shown in Table 3. This was chosen because u_{input} is not much larger than u_{num} and to be conservative.

4.1.1. STANDARD UNCERTAINTY

For this Bulk-Flow model, there are nine relevant input parameters. The first six bulk-flow parameters (X1-X6) correspond directly to the ANSYS model's first six parameters (X1-X6). The seventh parameter is static pressure rather than total pressure. The same resources and standard uncertainties were used as in the ANSYS analysis for these parameters. Pre-swirl standard uncertainty was estimated with values from Darden et al [26]. Since the necessary experimental data was unavailable, the standard uncertainty of friction factor coefficient was estimated by sampling with the friction factor expressions from Blasius and Moody in Equation (19). Assuming $b = 0.217$, the friction factor coefficient becomes a function of Reynold's number $a = a(Re)$ from Equation (19). Since $Re = Re(c, R, \Omega, \rho, \mu)$ from Equation (7), the friction factor coefficient is a function of these other input parameters $a = a(c, R, \Omega, \rho, \mu)$. For 500 trials, each of these input parameters $X = c, R, \Omega, \rho, \mu$ is sampled from as a random variable from a normal distribution $\mathrm{x} \sim N(X, u_X)$ with its nominal value as mean and its standard uncertainty as standard deviation. Each trial estimates friction factor coefficient a. From the 500 trials, a standard deviation of a was found to estimate u_X for friction factor coefficient, resulting in $u_{X=a}$ =0.0054.

4.1.2. SENSITIVITY COEFFICIENT

The sensitivity coefficients are found with $\frac{\partial \mathrm{S}}{\partial \mathrm{X_i}} \approx \frac{\Delta \mathrm{S}}{\Delta \mathrm{X_i}}$ by perturbing by $\Delta \mathrm{X_i}$. For this study, the perturbations were within $0.5 \geq \frac{\Delta \mathrm{X_i}}{\mathrm{X_i}} \geq 10^{-10}$. One realization of $\Delta \mathrm{S}/\Delta \mathrm{X_i}$ for a particular $\Delta \mathrm{X_i}/\mathrm{X_i}$ is needed for input uncertainty estimation. The chosen value should be within an interval of $\Delta X_i/X_i$ where $\Delta S/\Delta X_i$ is stable or where the effect X_i has on S is constant. The leakage is reported in MATLAB with precision of 10^{-16}, so when no change in leakage output is observed, $\Delta S = 10^{-17}$ is assumed, though it could be smaller.

Only one parameter should be adjusted at a time, and each of the nine parameters was adjusted bilaterally with 16 different perturbation sizes ΔX_i. This required 288 simulations, which took less than ten minutes of runtime. In order to do this efficiently, the relevant parameters were automatically adjusted in MATLAB with a loop. These results were used to approximate each derivative with Equation (28).

A MATLAB Code was developed for running a perturbation study on this code, found in Appendix C.II. Perturbation Study (example).

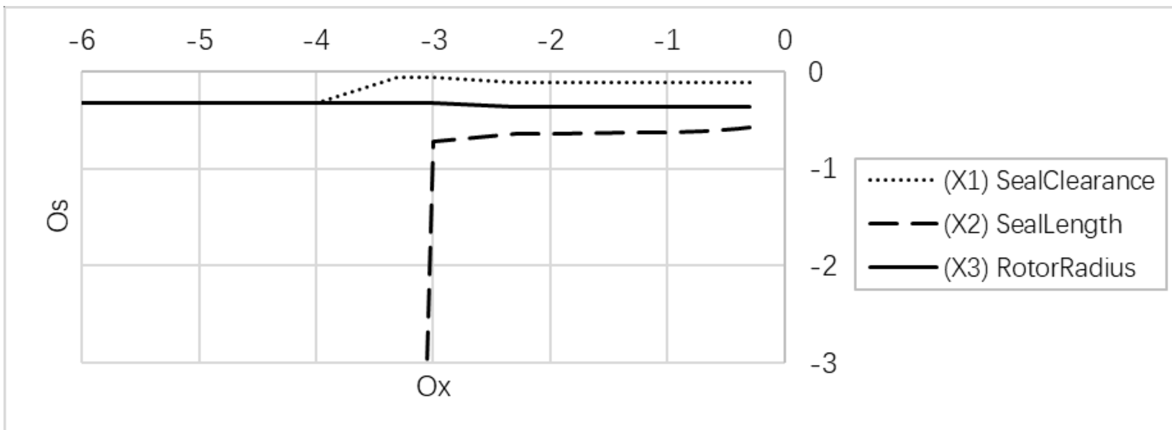

FIGURE 17
SENSITIVITY VS. PERTURBATION FOR X1-X3

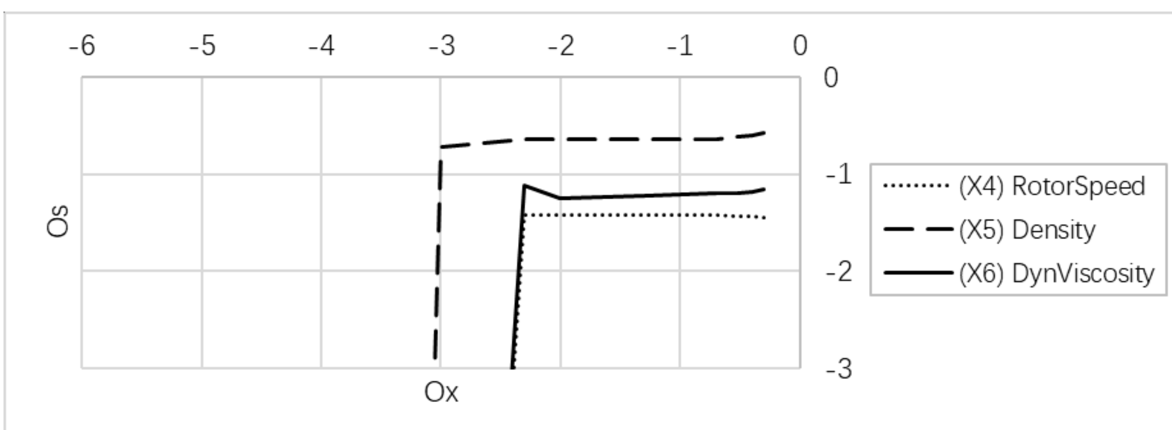

FIGURE 18
SENSITIVITY VS. PERTURBATION FOR X4-X6

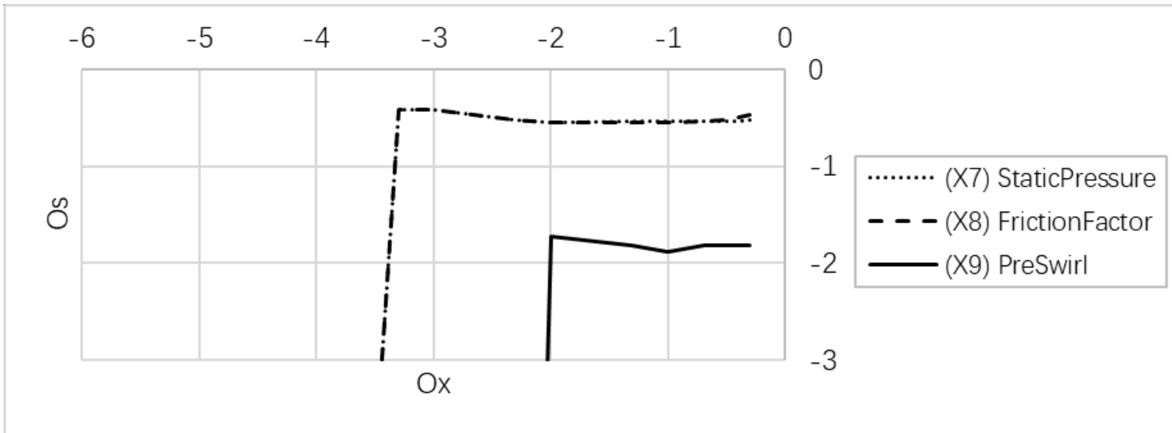

FIGURE 19
SENSITIVITY VS. PERTURBATION FOR X7-X9

Normalizing metrics from Equations (29) and (30) are useful in graphically assessing the perturbation study of all nine parameters, again using the order of the relative perturbation be $\mathrm{O}_{\mathrm{X}_\mathrm{i}}$ and the order of the relative sensitivity coefficient be $\mathrm{O}_{\mathrm{S}_\mathrm{i}}$.

For each of the parameter's graphs in Figure 17-Figure 19, the interval of O_X where the O_S is most stable is desired. This value will be trusted to represent the true estimation of the sensitivity coefficient, $\frac{\partial s}{\partial x_i}$. The stable region for these parameters is much clearer than for those in the ANSYS simulation. $\mathrm{O}_\mathrm{X} = -0.7$ is sufficient for each parameter. Results are shown in Table 7.

4.1.3. INPUT UNCERTAINTY

Table 7 summarizes each parameter's nominal value, standard uncertainty, sensitivity coefficient from analysis of Figure 17-Figure 19, and input uncertainty contribution.

According to Table 7, the seal clearance (X1), dynamic viscosity (X6), and friction factor coefficient (X8) are the dominating contributor to input uncertainty, on the order of 1E-2 L/s. All other contributions were on the order of 1E-3 L/s or smaller. As given by Equation (27), the uncertainty contributions of each parameter are combined into a total input uncertainty to be $u_{input} = 0.03741\ L/s$, which estimates input error $\delta_{input} \in [-u_{input}, u_{input}]$.

TABLE 7
INPUT UNCERTAINTY SUMMARY

Input Parameter		[u]	Xnom	Ux	dS/dX	Uinput
			[u]	[u]	L/s/[u]	L/s
Seal Clearance	**X1**	mm	0.168	0.005	4.60E+0	2.30E-2
Seal Length	**X2**	mm	60	0.005	4.03E-3	2.01E-5
Rotor Radius	**X3**	mm	35.6895	0.005	1.22E-2	6.11E-5
Rotor Speed	**X4**	rpm	3000	100	1.26E-5	1.26E-3
Density	**X5**	kg m-3	997	2.5	2.32E-4	5.80E-4
Dyn Visocity	**X6**	mPa s	0.89	0.25	7.11E-2	1.78E-2
Static Pressure	**X7**	kPa	784	5	3.68E-4	1.84E-3
Friction Factor	**X8**		0.0674	0.0054	4.33E+0	2.34E-2
PreSwirl	**X9**		0	0.1	1.51E-2	1.51E-3

This evaluation was also done on the other simulation output variables, shown in Appendix B: Additional Bulk-Flow Results. This was done with grid $N_1 = 100$ nodes for each output variable. For each perturbation, all output variables were stored, so the number of perturbation simulations was the same as for the study shown above. This was possible because in the previous section, the grid size with $N_1 = 100$ nodes was found to be reasonable for all output variables. If their converged regions did not overlap so nicely, these perturbation simulations would need to be done separately, for each output parameter's converged grid size. Practically, finding a grid size where all output variables are numerically converged is more useful with respect to using the model for prediction purposes. For each output variable, perturbation curves were plotted and analyzed, as shown here, and flat region were identified. The graphs were steady, as they are for leakage shown above.

4.2. Special Considerations

The process of determining the appropriate perturbation 0_X can be quite subjective, especially for ANSYS parameters X1, X3, X8-X12. This choice could misrepresent the input uncertainty. Such a situation calls for refinement and poses questions for future work. This may highlight needs for model improvement, more simulations, or a more detailed analysis of sensitivity study results.

Assigning a standard uncertainty without the appropriate experimental data is quite subjective. This can have a major impact on the input uncertainty predictions. For example, the friction factor coefficient has a standard uncertainty of 0.0054. This value was found by estimating the standard deviation of Moody and Blasius approximations with sampled input parameters, rather than by experimental perturbations. This estimate was used because it was the best available information. If the true value of standard uncertainty for friction factor coefficient was 10 times larger, then the input uncertainty of friction factor coefficient would be 10 large smaller as well. This would have a huge impact on the total input uncertainty of the bulk flow model. The same logic follows for the other standard uncertainties in this section. The subjectivity of standard uncertainty without the necessary experimental data is concerning.

As remarked earlier, if $u_{input} \gg u_{num}$, then a coarser grid from the grid triplets may be used for estimating u_{input}. Since this is the case for both models, each perturbation study could have been performed with grid N_2 or N_3 instead of grid N_1. This would reduce runtime.

Chapter 5: Model Error Estimate

5.1. Overview

The difference in errors, or comparison error E, is explicitly attainable.

$$E = \delta_S - \delta_D = S - D \tag{1}$$

The simulation error is the culmination of three error sources.

$$\begin{cases} \delta_S = \delta_{model} + \delta_{num} + \delta_{input} \\ \delta_{model} = E - (\delta_S - \delta_D) = E - \left(\delta_{num} + \delta_{input} - \delta_D\right) \end{cases} \tag{2}$$

Model error is error in simulation caused by modeling assumptions and the deliverable result of this validation study. Validation uncertainty is then the uncertainty of the estimate of model error.

$$u_{val} \equiv \pm\sqrt{u_{num}^2 + u_{input}^2 + u_D^2} \tag{3}$$

Throughout the standard's procedure, estimates for $u_{num}, u_{input}, u_D, \delta_{model}$ are determined, each following the ISO guidelines. Then model error is estimated to be within an interval.

$$\delta_{model} \in E \pm u_{val} \tag{4}$$

Here, the bounds for model error are found. Additionally, the magnitudes of E and u_{val} will be compared to evaluate the codes.

5.2. ANSYS Model

The goal of this section is to estimate model error. Comparison error is the difference in simulation output and experimental observation. From Equation (1), the comparison error for leakage is shown in Table 8.

TABLE 8
RESULTS SUMMARY

D	S	E
0.528	0.50776	-0.0202

The value E is an estimator of model error, and its magnitude and sign are certain and constant. u_{val} is the uncertainty of that estimator. The uncertainty values are shown in Table 9.

TABLE 9
UNCERTAINTY SUMMARY

	u_D	u_{num}	u_{input}	u_{val}
L/s	0.02640	0.00063	0.08549	0.08947
% of D	5%	0.1%	16%	17%

In Table 9, u_{val} is found from Equation (3). Since $|E| \leq u_{val}$, the modeling error is within the noise imposed by numerical, input, and experimental uncertainty. Thus, there is no clear opportunity for improvement of this model besides tending to each uncertainty source. According to Equation (4), the model error interval is shown in Table 10.

TABLE 10
MODEL ERROR INTERVAL ESTIMATE

	$\delta_{model,lo}$	$\delta_{model,hi}$
L/s	-0.10971	0.06924
% of D	-21%	13%

The model error estimates the uncertainty due to modeling assumptions.

5.3. Bulk-Flow Model

The goal of this section is to estimate model error. Comparison error is the difference in simulation output and experimental observation. From Equation (1), the comparison error for leakage is shown in Table 11.

TABLE 11
RESULTS SUMMARY

D	S	E
0.528	0.47580	-0.0522

The value E is an estimator of model error, and its magnitude and sign are certain and constant. u_{val} is the uncertainty of that estimator. The uncertainty values are shown in Table 12.

TABLE 12
UNCERTAINTY SUMMARY

	u_D	u_{num}	u_{input}	u_{val}
L/s	0.02640	0.00253	0.03741	0.04585
% of D	5%	0.5%	7%	9%

In Table 12, u_{val} is found from Equation (3). Since $|E| \approx u_{val}$, the modeling error is well within the noise imposed by numerical, input, and experimental uncertainty. Thus, there is no clear opportunity for improvement of this model besides tending to each uncertainty source. According to Equation (4), the model error interval is shown in Table 13.

TABLE 13
MODEL ERROR INTERVAL ESTIMATE

	$\delta_{model,lo}$	$\delta_{model,hi}$
L/s	-0.09806	-0.00635
% of D	-19%	-1%

The model error estimates the uncertainty due to modeling assumptions.

Chapter 6: Closing

6.1. Conclusion

Chapter 1 explained the importance of validation and uncertainty for both the software developer and end-user. The need for improvement and standardization of validation in practice was demonstrated. Due to the limited availability of valuable experimental data, ROMAC Laboratory at the University of Virginia is interested in developing a method of assessing low-fidelity models with high-fidelity models, such as ANSYS models. Various topics were discussed targeting this pursuit, such as N-version validation and calibration. Additionally, the weak areas in the validation community were discussed including assessment criteria. ASME V&V-20 was deemed to be the first step in ROMAC efforts to develop its own validation process.

In Chapter 2-5, ASME V&V-20 was thoroughly discussed. This included formally defining terms such as validation, verification, error, and uncertainty. The scope was also discussed: this procedure may only be applied to one simulation output at a time, under a specific set of conditions, and it requires experimental data. The methodology of quantifying numerical and input uncertainty was explained in great detail. This standard was found to be a very helpful start to ROMAC's goals as it exposed the intricacies of validation efforts within the CFD field.

Chapter 2 discusses the details of the case study. This section explained in great detail the development of the ANSYS model and the careful considerations of geometry, mesh, and solver settings. The bulk-flow smooth seal code is an actual development in ROMAC for quick assessment of smooth seal behavior. The experimental setup and apparatus of a typical rotordynamics experiment was presented as it guided the development of the models. The experiment included little uncertainty analyses details but did include experimental uncertainty of 5% for the quantity of interest, leakage. In rotordynamics, it's typical to use published experiments for validation, which often means insufficient information. In Chapter 3, experimental uncertainty is discussed, and the experimentalists' reported value is used for the validation study.

In Chapter 3, numerical uncertainty of leakage was estimated for each model; these uncertainties only pertain to the particular validation point. The numerical uncertainty was estimated to be 0.1% for the ANSYS model and 0.5% for the bulk-flow model. This was also done for other important output variables of the bulk flow model. This resulted in an estimate of the error caused by the numerical solver of each model, which required a grid convergence study. This process was very time-consuming for the ANSYS model as it was difficult to prevent numerical overflow for fine meshes and difficult to satisfy Y+ requirements for coarse meshes. Additionally, the V&V-20 method emphasizes preserving the geometry of each element between meshes. This additional constraint complicated and lengthened the process further. The two models resulted in similar numerical uncertainty values, but the calculation was much smoother and more reliable for the bulk flow model than the ANSYS model. The process was also smooth for the other bulk flow output variables, where numerical uncertainty was estimated for the same grid triplet. The odd behavior encountered with the ANSYS model may be due to poorly behaved convergence; however, its predictions were consistently within 1% of one another. Unfortunately, this is not always well-defined or easy to fix. However, this situation did lead to model improvements and highlights a need for improvement in the mesh or convergence settings. This information is invaluable as a graphical analysis of convergence rendered it sufficient. In this way, the numerical uncertainty analysis demonstrated an ability to significantly improve insight to a model's weaknesses.

In Chapter 4, input uncertainty of leakage was estimated for each model; these uncertainties only pertain to the particular validation point. This was also done for other important output variables of the bulk flow model. This resulted in an estimate of the error caused by the uncertainty of inputs to each model, which required a perturbation study. For the output variables of the bulk flow model, all of this was done with one perturbation study since they share a converged grid size. The input uncertainty was estimated to be 16% for the ANSYS model and 7% for the bulk-flow model. Standard uncertainties were estimated rather than determined from experimental data. For friction factor, this was directly estimated with friction factor approximation expressions and other input parameters. For both models, seal clearance was one of the top contributors to input uncertainty which is expected as this dimension has a very big impact on flow. Friction factor had a low impact on the uncertainty compared to expectations. The perturbation study was much smoother for the bulk flow model than the ANSYS model; for some input parameters, the ANSYS model had a variable response, rendering the choice of perturbation size quite subjective. The cause of this issue is not clear, but perhaps it may be due to inadequate convergence, discussed in the previous paragraph.

The ANSYS simulations took a very long time, each simulation lasting between 2 minutes and 6 hours. Considering about 300 simulations were needed for input uncertainty and 13 more simulations needed for numerical uncertainty, the total runtime was about half a week, not accounting for redoing the process when a mistake was found or improvements were added. In contrast, for the Bulk-Flow model, around 400 simulations were run in less than 10 minutes, not accounting for correcting mistakes. The post-processing time is very similar between models because they were both done with MATLAB.

In Chapter 5, each models' overall validation uncertainty and model error for leakage were estimated; these estimates only pertain to this particular validation point. The model error was estimated to be within [-21%, 13%] for the ANSYS model and within [-19%, -1%] for the bulk-flow model. The validation uncertainty combines experimental, numerical, and input uncertainty. Surprisingly, the bulk-flow model had a smaller validation uncertainty (9%) than the ANSYS model (17%). The validation uncertainty represents the uncertainty of the model error estimate, which is a comparison error. For both models, the validation uncertainty and comparison error were of similar magnitude, so no clear opportunities to improve model accuracy are apparent. Model precision may clearly be improved by reducing validation uncertainty. Model error is the error due to modeling assumptions and approximations. The estimate for ANSYS model error was -12% and 4%, and for the bulk-flow model, between - -19% and -1%. Since quantification of the models' agreement with reality was found, according to V&V-20, the leakage prediction of these models was validated at 3000 rpm.

In the Appendices, additional resources are included. A generalized MATLAB code was developed for analyzing input and numerical uncertainty, which may be used for future analysis. This procedure was also performed on other important outputs of the bulk-flow model, such as stiffness and damping, shown in the Appendices.

6.2. Contributions

This work raises awareness and knowledge about validation and the need for improved methods in rotordynamics. Literature highlights are presented on validation and other topics relating to ROMAC's end goal in validation. This work is the first formal step in building a standard for ROMAC validation and uncertainty quantification. There are clear opportunities to build upon this work. This thesis may serve as a guide and example for understanding the validation procedure in V&V-20 applied to rotordynamics problems. Due to the newness of V&V-20, this work also may contribute to the ASME V&V community's growth and refinement.

This procedure highlights where models may have weaknesses. Two formal grid convergence studies are demonstrated on a relatively basic problem, resulting in estimation of numerical uncertainty. Problems encountered in this process highlight the need for a more converged mesh. This method will tend to reveal such numerical weaknesses. Structured grid refinement was shown to be critical in finding the observed order of accuracy. Two formal perturbation studies were also demonstrated, resulting in input uncertainty of each input parameter. Selecting perturbation size carefully was shown to be difficult and important. This process highlights the impact of each parameter. This whole procedure may also be used to compare a low fidelity code to the ANSYS model of the same component, as was done here. The uncertainties may be compared to verify adequate robustness and justify assumptions. The low-fidelity model here, the bulk-flow model, was shown to have much higher model error than the high-fidelity model, the ANSYS model, as expected. This was primarily driven by input uncertainty.

MATLAB codes and documentation are provided for others to perform a numerical and input uncertainty analysis with grid convergence and perturbation study results. The validation study results for all output variables of the bulk-flow code are also included for the same validation point as in the previous chapters. Code developers can perform this procedure on a variety of conditions where experimental data is available and analyze the uncertainty trends.

Through the above points, this process can lead to code accuracy improvements. The demonstration of the validation process can lead users to be more careful with certain model inputs and resolution choices, can lead users to trust the codes more readily, and will also inform users of specific cases where the code has been validated so that they are aware of when they might be veering far from the validated conditions.

6.3. Recommendations

Considering the importance and benefits of validation, ROMAC should stay involved in developments in the verification and validation community and continue researching these topics. Methods of verification should be researched, standardized, and documented for ROMAC practice. More research is needed on the instability of the observed order of accuracy and sensitivity coefficients. In review of the validation results, assessment criteria would be valuable. Assessment criteria would present a clear standard that codes are expected to reach, providing consistent quality codes to members. An assessment criterion could be a review board's approval of a validation study's results.

There are various specific areas which may be useful to explore. One opportunity for growth may be trying global methods - sampling. This may be advantageous when applied to input parameter uncertainty analysis rather than the local method used in this work. Consider using design of experiment and sampling software for developing perturbation study inputs. Cross validation and other calibration-validation methods may be useful for allocating experimental data between calibration and validation. This would be useful for estimating parameters such as friction factor. Algorithms could be developed for analyzing an unstable observed order of accuracy and unstable regions of input uncertainty.

ROMAC has a unique goal to use ANSYS to assess in-house codes. To continue this endeavor, ROMAC's terminology should be defined and agreed upon; calibration and validation are sometimes confused in such situations, and ASME V&V would not term code-to-code comparison a validation since reality is not directly involved. However, ANSYS may certainly be used to calibrate empirical factors for in-house codes such as friction factor coefficient. Though not yet backed by V&V theory, a practical initiation of the end-goal would be: (1) build an Ansys model, (2) perform validation procedure thoroughly as per the most up-to-date protocol in ROMAC, (3) assess the validation results, (4) if passed assessment, check ROMAC code against Ansys output, perhaps following this V&V-20 procedure (considering ANSYS output as experimental output). This fourth step could use the simulation uncertainty of the ANSYS model (sum of numerical, input, and model error estimates) as the experimental uncertainty. These steps would be a reasonable starting point towards the end-goal. Before commencing this, it would be wise to first research the aforementioned topics and refine the methods in this work to be more time effective.

The ANSYS model's simulations were extremely time consuming, especially for the perturbation study. The ANSYS total runtime could be reduced by only using the first order finite difference approximation for the sensitivity coefficient. This would reduce the number of simulations by about 50% since perturbations would only be made in one direction. Since $u_{input} \gg u_{num}$, a coarser grid from the ANSYS grid triplet (N_2 or N_3 instead of N_1) could have been used for the perturbation study. This should reduce runtime. Of course, this could not be known until u_{input} is found. However, in the future, it may be wise to start with N_2 for an initial perturbation study. Another way to reduce runtime may be to explore sampling methods discussed in V&V-20.

The demonstration in this work only validates one output variable at one validation point. No formal procedure is available for extending this process to a whole model, but the agreed upon approach is that this procedure should be repeated on many different validation points (within the model's domain of interest) and with all pertinent output variables. Practically speaking, during the grid convergence study, it would be wise to identify a grid resolution where all output variables are converged so that the perturbation simulations do not need to be repeated at various grid resolutions. This would reduce runtime substantially and was done here for the Bulk Flow output variables in Appendix B: Additional Bulk-Flow Results. This is additionally useful for informing the end-user of a reliable grid resolution. After running this perturbation study, model form error and simulation error may be estimated for validation points. The resulting error estimates could be viewed as a function of their validation point (thus, input parameter values). This data may be viewed as points building an n-dimensional surface, where n is the number of input parameters. Building this surface would be extremely tedious. Thus, selecting which validation points (within the model's domain of interest) to evaluate at may be a research project itself, but it is one of the most critical steps moving forward. Sampling validation points from the domain of interest may expedite the process. Interpolation between validation points is not backed by the statistics, but it may be the best way to make use of the validation results. In this case, the uncertainty, or error estimates, across the domain of interest may be represented by a continuous, n-dimensional surface. In this case, uncertainty estimates may be reported to end-users.

Until more progress is made on ROMAC validation methods, code developers in ROMAC should follow this method at various experimental validation points and discuss with others if the resulting uncertainties seem reasonable. Throughout the validation process, the code developer should be prepared to find weaknesses in their code which should be addressed before moving forward. The results of a validation study may be reported similarly to this thesis and be available and approachable to end-users.

New experimental rigs should also be designed carefully, keeping in mind the validation needs. Simulation inputs should be measurable, and the uncertainty of measurement instruments should be well documented. Experimental data should be reported with measurement uncertainty and validation experiment results for use in other validation studies. Data should be generated and, if possible, dispersed for use in the community to share. Experimental data would also be useful for estimating standard uncertainties so that the subjectivity of input uncertainty is removed. In the meantime, estimating standard uncertainty with the available resources might be the best option.

These recommendations should lead the verification, validation, uncertainty quantification, and calibration in ROMAC to great improvements in code development, accuracy, and reliability. These steps may also further the research in the V&V community and raise the standard for rotating machinery model developers.

References

[1] V&V20 Standards Committee, "Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer." ASME, 2009.

[2] D. Biondi, G. Freni, V. Iacobellis, G. Mascaro, and A. Montanari, "Validation of hydrological models: Conceptual basis, methodological approaches and a proposal for a code of practice," *Phys. Chem. Earth*, vol. 42–44, pp. 70–76, 2012.

[3] R. Walker, S. Perinpanayagam, and I. K. Jennions, "Rotordynamics Faults: Recent Advances in Diagnosis and Prognosis," *Int. J. Rotating Mach.*, vol. 2013, p. 12, 2013.

[4] M. E. Leader, "Using Rotordynamics to Solve Serious Machinery Vibration Problems."

[5] J. A. Kocur and J. Nicholoas, "Surveying Tilting Pad Journal Bearing and Gas Labyrinth Seal Coefficients and Their Effect on Rotor Stability.pdf," *Proc. Thirty-Sixthturbomachinery Symp.*, p. 10, 2007.

[6] F. Stern, M. Diez, H. Sadat-Hosseini, H. Yoon, and F. Quadvlieg, "Statistical Approach for Computational Fluid Dynamics State-of-the-Art Assessment: N-Version Verification and Validation," *J. Verif. Valid. Uncertain. Quantif.*, vol. 2, no. 3, p. 031004, 2017.

[7] P. Jolly, A. Hassini, M. Arghir, and O. Bonneau, "Experimental and Theoretical Rotordynamic Coefficients of Smooth and Round-Hole Pattern Water Fed Annular Seals," in *ASME Tubo Expo*, 2014, no. June.

[8] W. T. Lindsey and D. W. Childs, "The Effects of Converging and Diverging Axial Taper on the Rotordynamic Coefficients of Liquid Annular Pressure Seals :," *J. Vib. Acoust.*, vol. 122, no. April, pp. 126–131, 2000.

[9] T. W. Ha and B. S. Choe, "Numerical simulation of rotordynamic coefficients for eccentric annular-type-plain-pump seal using CFD analysis †," *J. Mech. Sci. Technol.*, vol. 26, no. 4, pp. 1043–1048, 2012.

[10] J. Moore and A. Palazzolo, "Liquid Annular Seal CFD Analysis for Rotordynamic Force Prediction," vol. 1, pp. 295–337, 2006.

[11] M. Zhang, X. Wang, S. Xu, and W. Wang, "Numerical Simulation of the Flow Field in

Circumferential Grooved Liquid Seals," *Adv. Mech. Eng.*, pp. 1–10, 2013.

[12] M. Eek, H. Gavel, and J. Ölvander, "Definition and Implementation of a Method for Uncertainty Aggregation in Component-Based System Simulation Models," *J. Verif. Valid. Uncertain. Quantif.*, vol. 2, no. 1, p. 011006, 2017.

[13] Q. Wang and B. Pettinato, "IDENTIFICATION IN ROTORDYNAMICS : MODEL-BASED VS . DIRECT MEASUREMENTS," 2009.

[14] G. Varoquaux, "Cross-validation failure: Small sample sizes lead to large error bars," *Neuroimage*, no. April, pp. 1–10, 2017.

[15] W. L. Oberkampf and B. L. Smith, "Assessment Criteria for Computational Fluid Dynamics Model Validation Experiments," *J. Verif. Valid. Uncertain. Quantif.*, vol. 2, no. 3, p. 031002, 2017.

[16] D. Muschalla *et al.*, "The HSG Procedure for Modelling Integrated Urban Wastewater Systems," *Water Sci. Technol.*, vol. 60, no. 8, pp. 2065–2075, 2009.

[17] A. Jatale, P. J. Smith, J. N. Thornock, S. T. Smith, J. P. Spinti, and M. Hradisky, "Multiscale Validation and Uncertainty Quantification for Problems With Sparse Data," *J. Verif. Valid. Uncertain. Quantif.*, vol. 2, no. 1, p. 011001, 2017.

[18] P. Perdikaris *et al.*, "Model inversion via multi-fidelity Bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond.," *J. R. Soc. Interface*, vol. 13, no. 118, pp. 409–423, 2016.

[19] A. Lewis, R. Smith, and B. Williams, "An Information Theoretic Approach to Use High-Fidelity Codes to Calibrate Low-Fidelity Codes An Information Theoretic Approach to Use High-Fidelity Codes to Calibrate Low-Fidelity Codes," pp. 0–29, 2014.

[20] C. B. Storlie, W. A. Lane, E. M. Ryan, J. R. Gattiker, and D. M. Higdon, "Calibration of Computational Models With Categorical Parameters and Correlated Outputs via Bayesian Smoothing Spline ANOVA," *J. Am. Stat. Assoc.*, vol. 110, no. 509, pp. 68–82, 2015.

[21] H. A. Martens and P. Dardenne, "Validation and verification of regression in small data sets," 1998.

[22] V. Klemes, "Operational Testing of Hydrological Simulation Models," *Hydrol. Sci. J.*, vol. 31, pp. 13–24, 1986.

[23] D. W. Childs, L. A. Seals, and L. E. Rodriguez, "Influence of Groove Size on the Static

and Rotordynamic Characteristics of Short ," *J. Tribol.*, vol. 129, no. April, pp. 398–406, 2007.

[24] G. Di Baldassarre and A. Montanari, "Uncertainty in River Discharge Observations: A Quantitative Analysis," *Hydrol. Earth Syst. Sci.*, vol. 13, no. 6, pp. 913–921, 2009.

[25] S. Kaneko, T. Ikeda, T. Saito, and S. Ito, "Experimental Study on Static and Dynamic Characteristics of Liquid Annular Convergent-Tapered Damper Seals With Honeycomb," *J. Tribol.*, vol. 125, no. July, pp. 592–599, 2003.

[26] J. M. Darden, E. M. Earhart, and G. T. Flowers, "Comparison of the Dynamic Characteristics of Smooth Annular Seals and Damping Seals," *J. Eng. Gas Turbines Power*, vol. 123, no. October, pp. 857–863, 2001.

[27] M. G. Anderson and P. D. Bates, *Model Validation: Perspectives in Hydrological Science*. J. Wiley, 2001.

[28] T. Hug *et al.*, "Wastewater Treatment Models in Teaching and Training : The Mismatch Between Education and Requirements for Jobs," *Water Sci. Technol.*, vol. 59, no. 4, pp. 745–753, 2009.

[29] ANSI/ASME PTC 19.1, "Measurement Uncertainty," 1985.

[30] K. M. Becker and J. Kaye, "Measurements of Diabatic Flow in an Annulus With an Inner Rotating Cylinder," *J. Heat Transfer*, vol. 84, no. 2, pp. 97–104, May 1962.

[31] H. Chougule, D. Ramerth, and D. Ramachandran, "Low-Leakage Designs for Rotor Teeth and Honeycomb Lands in Labyrinth Seals," in *ASME Paper GT2008-51024*, 2008, vol. 4, pp. 1613–1620.

[32] A. Ivanov and A. Moskvicev, "Influence of Geometry on Vortex Configuration and Dimension in LRE Turbopump Labyrinth Seal," *Procedia Eng.*, vol. 106, pp. 126–131, 2015.

[33] W. Wróblewski, D. Fra, and K. Marugi, "Leakage Reduction by Optimisation of the Straight – Through Labyrinth Seal with a Honeycomb and Alternative Land Configurations," *Int. J. Heat Mass Transf.*, vol. 126, pp. 725–739, 2018.

[34] Y. Doğu, M. C. Sertçakan, A. S. Bahar, A. Pişkin, E. Arıcan, and M. Kocagül, "CFD Investigation of Labyrinth Seal Leakage Performance Depending on Mushroom Shaped Tooth Wear," in *Proceedings of ASME Turbo Expo*, 2015.

[35] J. J. Moore, "Three-Dimensional CFD Rotordynamic Analysis of Gas Labyrinth Seals ,"

J. Vib. Acoust., vol. 125, no. 4, pp. 427–433, Oct. 2003.

[36] S. Subramanian, A. S. Sekhar, and B. V. S. S. S. Prasad, "Rotordynamic Characterization of Rotating Labyrinth Gas Turbine Seals with Radial Growth: Combined Centrifugal and Thermal Effects," *Int. J. Mech. Sci.*, vol. 123, no. June 2016, pp. 1–9, Jan. 2017.

[37] D. C. Wilcox, *Turbulence Modeling for CFD*, 3rd ed. La Canada, California: DCW Industries, Inc., 2006.

[38] T. Cebeci, *Turbulence Models and Their Application*. Long Beach, CA: Horizons Publishing Inc., 2004.

[39] Y. Yamada, "Resistance of flow through an annulus with an inner rotating cylinder," *J. JSME*, vol. 5, no. 18, pp. 302–310, 1962.

[40] C. Watson-Kassa, N. Morgan, M. He, and H. Wood, "Theoretical and Computational Fluid Dynamics (CFD) Predictions of Annular Seal Inlet Loss and Exit Recovery Coefficients," 2020.

[41] L. Zhao and P. Allaire, "Manual for Computer Program Seal3," *Univ. Virginia ROMAC Rep.*, vol. 418, pp. 1–70, 1998.

[42] P. J. Roache, *Verification and Validation in Computational Science and Engineering*. Hermosa Publishers, 1998.

[43] M. P. Dawson and D. W. Childs, "Measurements Versus Predictions for the Dynamic Impedance of Annular Gas Seals — Part Ⅱ: Smooth and Honeycomb," *J. Eng. Gas Turbines Power*, vol. 124, no. October, pp. 963–970, 2002.

[44] T. W. Ha and D. W. Childs, "Friction-Factors for Flat-Plate Tests of Smooth and Honeycomb Surfaces," *J. Tribol.*, vol. 114, no. January, pp. 722–730, 1992.

[45] "The Engineering Toolbox," *Water - Dynamic and Kinematic Viscosity*, 2004. [Online]. Available: https://www.engineeringtoolbox.com/water-dynamic-kinematic-viscosity-d_596.html.

[46] UIUC, "Illinois Chemistry," *Temperature Effects on Density*. [Online]. Available: http://butane.chem.uiuc.edu/pshapley/GenChem1/L21/2.html.

Appendices

Appendix A: ANSYS Mesh Parameters

The inflation and sweeps are structured the same way. They both have a starting small element that increases as a geometric sequence in one dimension. If the smallest height is increased by a factor γ then all elements in the sweep or inflation are heightened by the same factor. To maintain a structured refinement, the all dimensions should be adjusted by the same factor. So, the smallest height in the inflations and sweeps are changed by the same amount. Additionally, the inflations share the same minimum height, and the sweeps share the same minimum height. The sweeps are in the radial directions and the inflations are in the radial and axial directions. The only control in the circumferential direction is the maximum overall element size. This is simply controlled by a setting which sets the maximum element size. This setting is adjusted by γ as well to maintain a structured grid refinement. With this method, $r_r \approx r_z \approx r_\theta$. Since this sector is so thin, these dimensions are effectively Cartesian.

TABLE 14

MESH PARAMETERS FOR MESH INDEPENDENCE STUDY

Factor		Minimum Heights, x			Sweep Parameters						Results	
Grid i	Ratio γ(i,1)	Inflation *r,z* (mm)	Overall *r,z,theta* (mm)	Sweep *r* (mm)	Clearance Region (# div)	(bias)	Outlet Region (# div)	(bias)	Inlet Region (# div)	(bias)	Grid Size (# el.)	Leakage Output (L/s)
1	1	9.09E-3	1.82E-1	9.09E-4	16	15	79	1693	73	956	106220	0.505029
2	1.1	1.00E-2	2.00E-1	1.00E-3	16	15	78	1539	72	869	97404	0.504997
3	1.2	1.10E-2	2.20E-1	1.10E-3	15	13	77	1399	71	790	87687	0.507896
4	1.3	1.21E-2	2.42E-1	1.21E-3	15	13	76	1272	70	718	80580	0.507763
5	1.5	1.33E-2	2.66E-1	1.33E-3	14	11	75	1156	69	653	45798	0.507717
6	1.6	1.46E-2	2.93E-1	1.46E-3	14	11	74	1051	68	593	42332	0.507693
7	1.8	1.61E-2	3.22E-1	1.61E-3	13	9	73	956	67	539	36386	0.506966
8	1.9	1.77E-2	3.54E-1	1.77E-3	13	9	72	869	66	490	35356	0.507038
9	2.1	1.95E-2	3.90E-1	1.95E-3	12	7	71	790	65	446	32188	0.503657
10	2.4	2.14E-2	4.29E-1	2.14E-3	12	7	70	718	64	405	29496	0.504669
11	2.6	2.36E-2	4.72E-1	2.36E-3	11	6	69	653	63	368	13036	0.501031
12	2.9	2.59E-2	5.19E-1	2.59E-3	11	6	68	593	62	335	11239	0.504491
13	3.5	3.14E-2	6.28E-1	3.14E-3	10	5	66	490	60	277	9366	0.506231

In Table 14, the factor shown in the first column is representing the ratio each minimum height to the first one. In other words, $\gamma(i,1) = x_i/x_1$.

Appendix B: Additional Bulk-Flow Results

Below is a table of the output variables for which input and numerical uncertainties are calculated in this appendix.

TABLE 15
OUTPUT VARIABLES AND INDICES

Y Index	Output Variable Name
1	Direct Stiffness (Kxx)
2	Cross-Coupled Stiffness (Kxy)
3	Direct Damping (Cxx)
4	Cross-Coupled Damping (Cxy)
5	Moment (Mxx)
6	Mass Flow
7	Volumetric Flow
8	Axial Velocity
9	Inlet Loss
10	Outlet Recovery
11	Perturbed Pressure

B.I. NUMERICAL UNCERTAINTY

Below are graphs of each output variable's convergence.

B.I.i. Convergence Plots

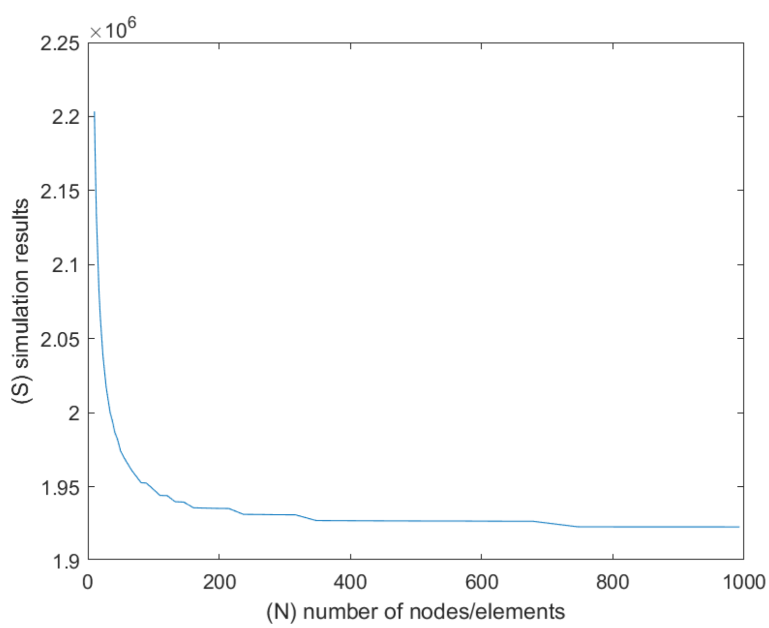

FIGURE 20
CONVERGENCE OF OUTPUT VARIABLE 1

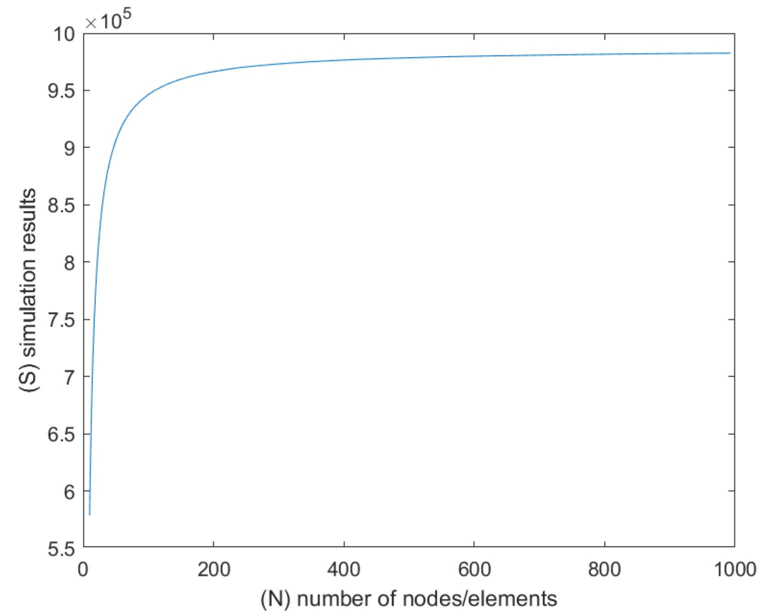

FIGURE 21
CONVERGENCE OF OUTPUT VARIABLE 2

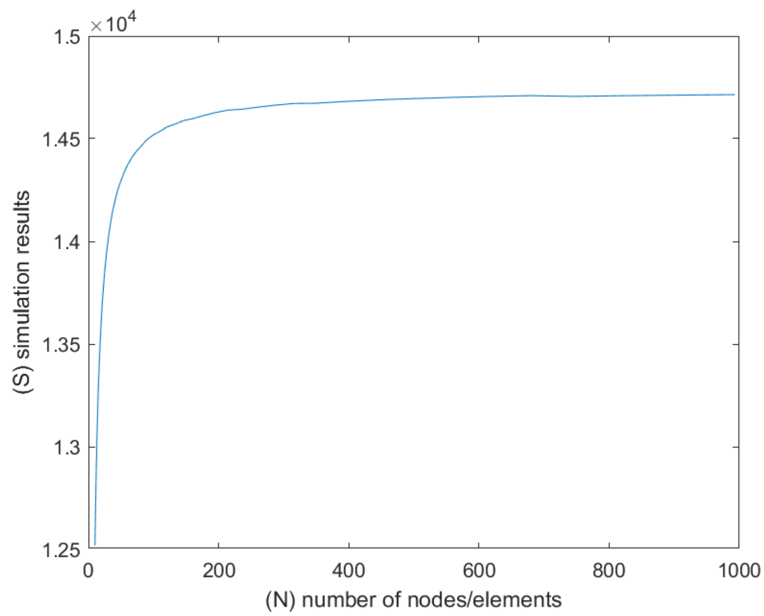

FIGURE 22
CONVERGENCE OF OUTPUT VARIABLE 3

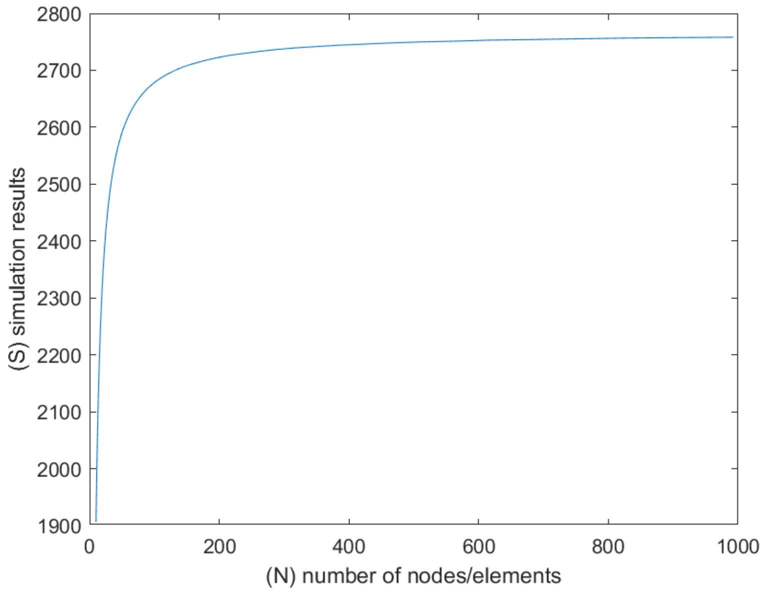

FIGURE 23
CONVERGENCE OF OUTPUT VARIABLE 4

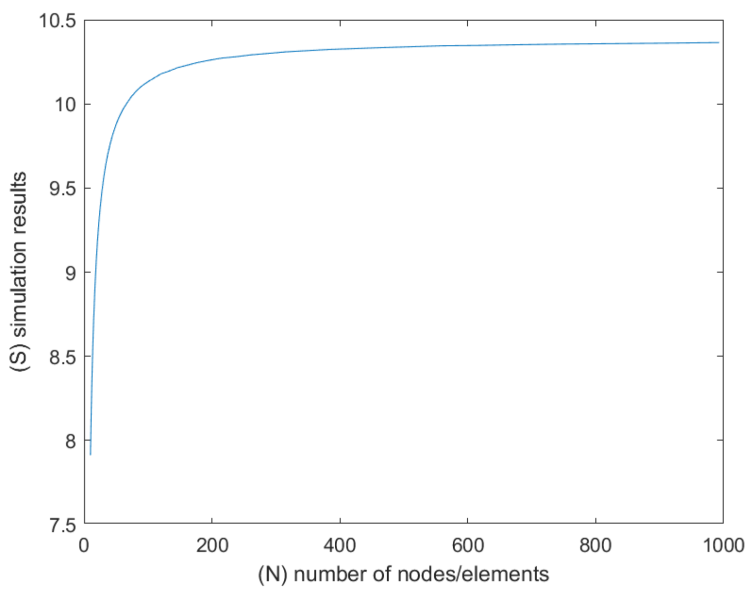

FIGURE 24
CONVERGENCE OF OUTPUT VARIABLE 5

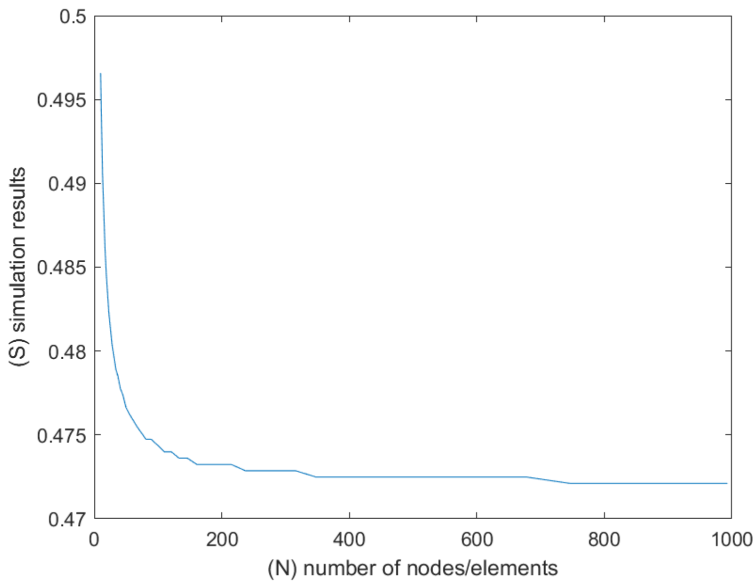

FIGURE 25
CONVERGENCE OF OUTPUT VARIABLE 6

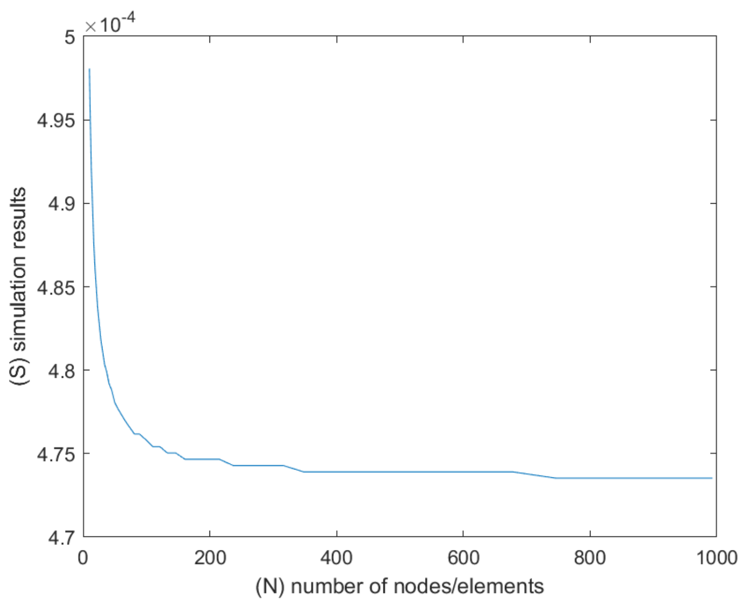

FIGURE 26
CONVERGENCE OF OUTPUT VARIABLE 7

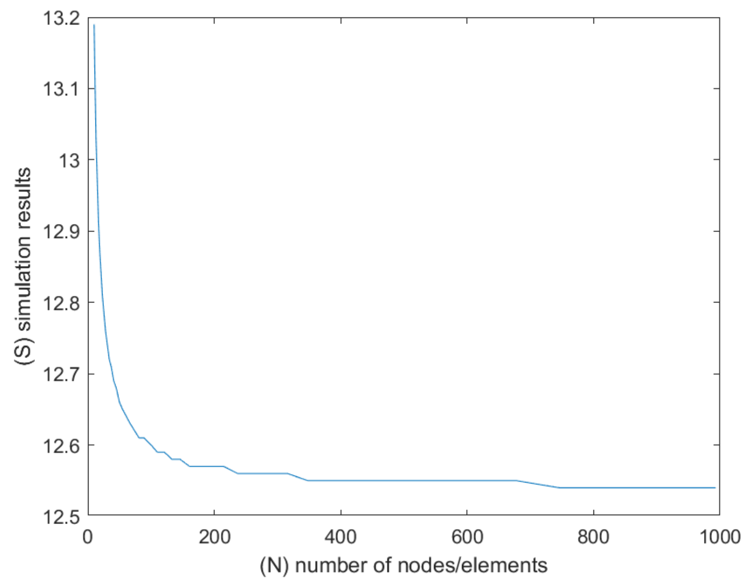

FIGURE 27
CONVERGENCE OF OUTPUT VARIABLE 8

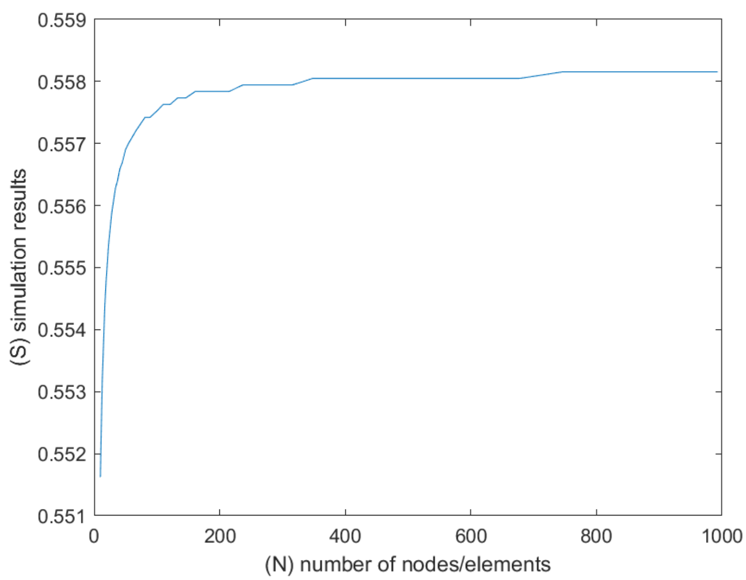

FIGURE 28
CONVERGENCE OF OUTPUT VARIABLE 9

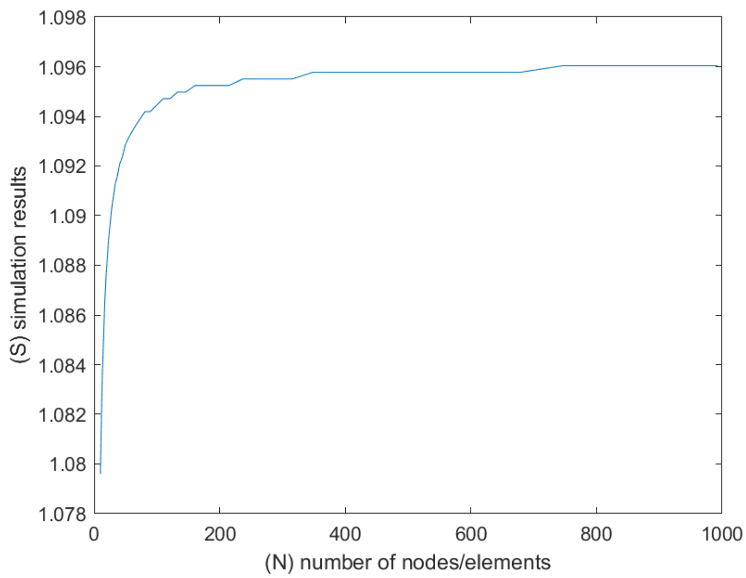

FIGURE 29
CONVERGENCE OF OUTPUT VARIABLE 10

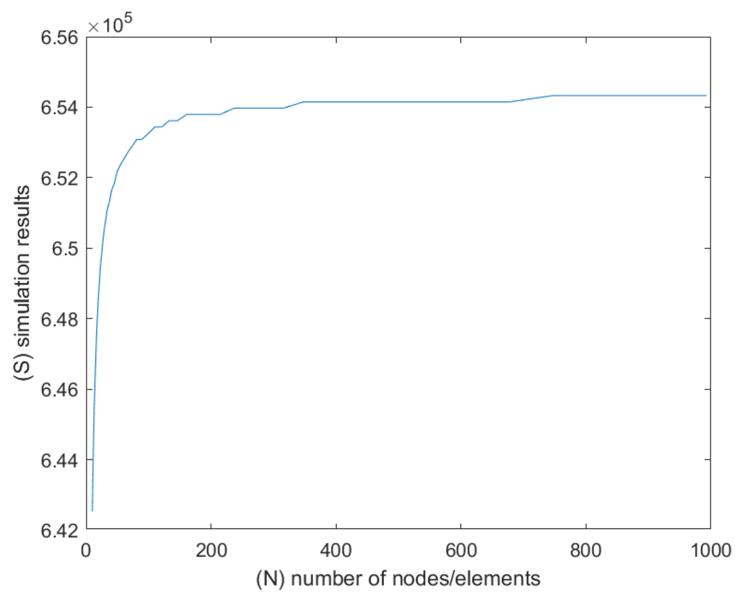

FIGURE 30
CONVERGENCE OF OUTPUT VARIABLE 11

B.I.ii. Observed Order of Convergence p

Below are histograms of the observed order of accuracy across all grid triplets. Each histogram corresponds to an output variable. Here, the spread can be observed. Clearly, they are all roughly centered around 1 which is the theoretical order of accuracy for this bulk-flow code. The spreads vary but are generally reasonable.

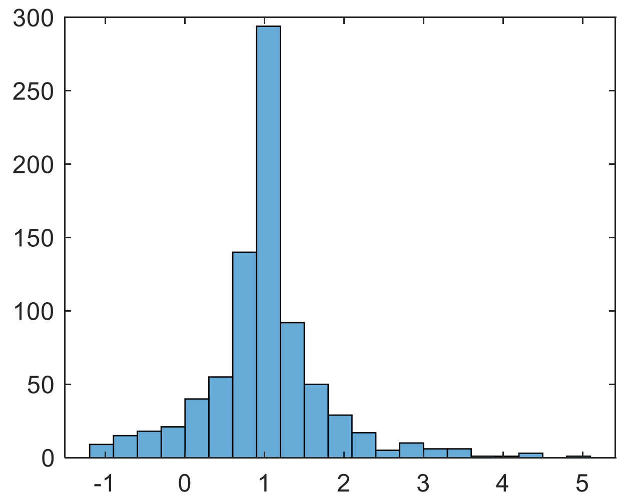

FIGURE 31

SPREAD OF OBSERVED ORDER OF ACCURACY P FOR OUTPUT VARIABLE 1

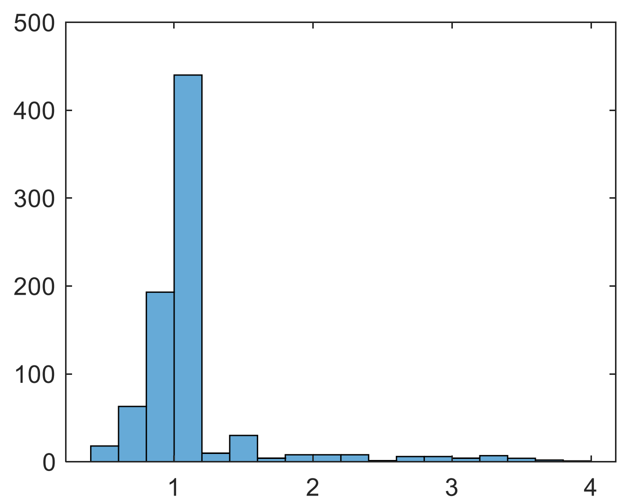

FIGURE 32

SPREAD OF OBSERVED ORDER OF ACCURACY P FOR OUTPUT VARIABLE 2

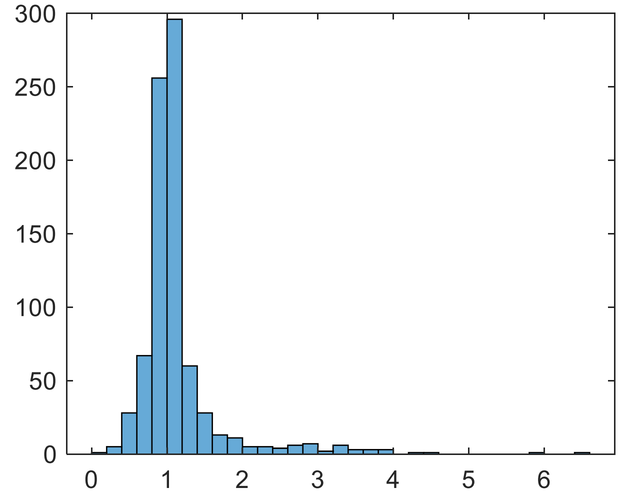

FIGURE 33
SPREAD OF OBSERVED ORDER OF ACCURACY P FOR OUTPUT VARIABLE 3

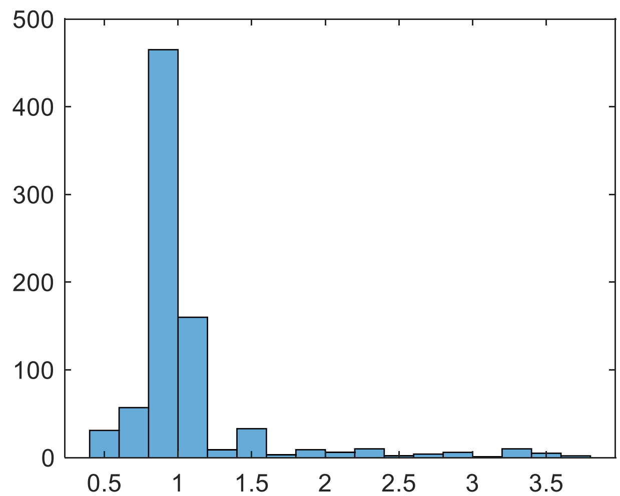

FIGURE 34
SPREAD OF OBSERVED ORDER OF ACCURACY P FOR OUTPUT VARIABLE 4

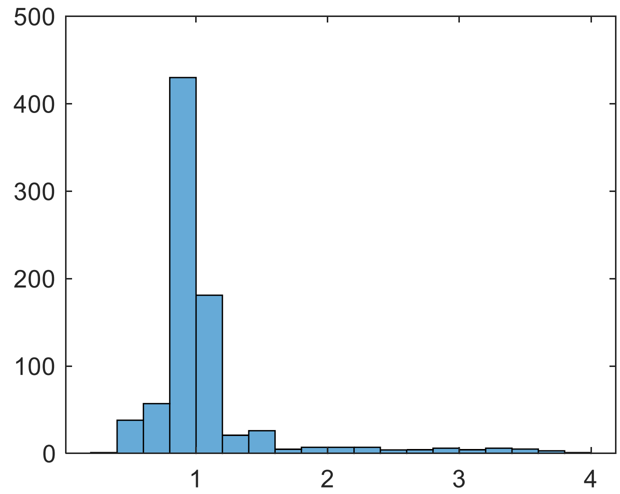

FIGURE 35
SPREAD OF OBSERVED ORDER OF ACCURACY P FOR OUTPUT VARIABLE 5

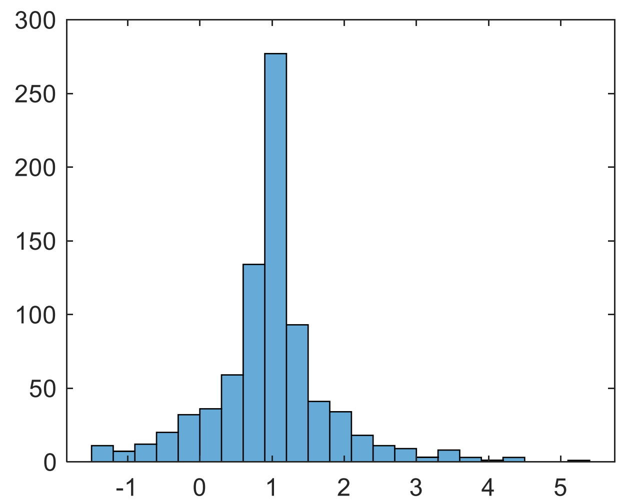

FIGURE 36
SPREAD OF OBSERVED ORDER OF ACCURACY P FOR OUTPUT VARIABLE 6

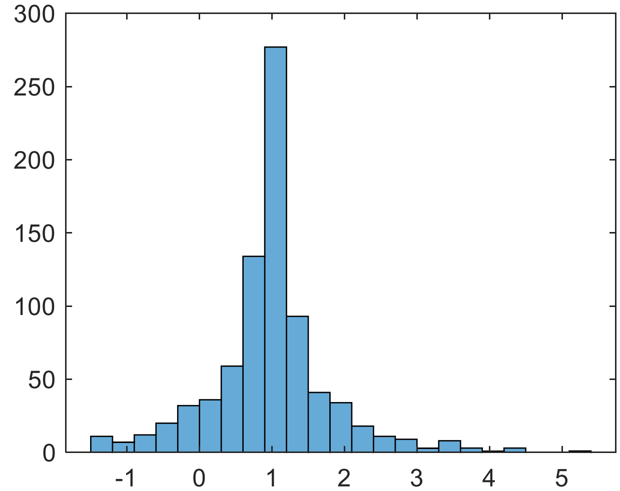

FIGURE 37
SPREAD OF OBSERVED ORDER OF ACCURACY P FOR OUTPUT VARIABLE 7

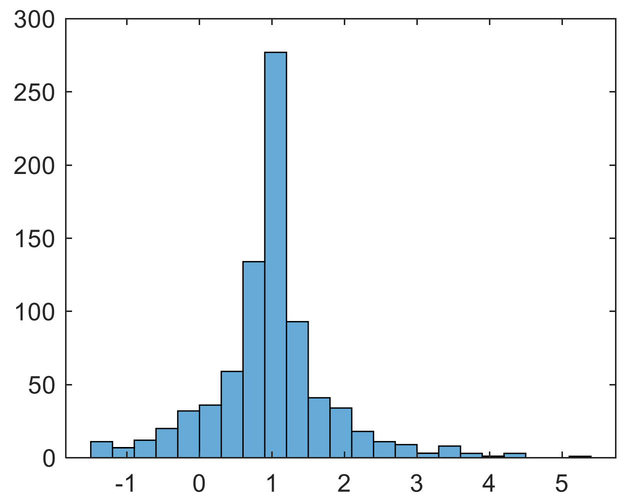

FIGURE 38
SPREAD OF OBSERVED ORDER OF ACCURACY P FOR OUTPUT VARIABLE 8

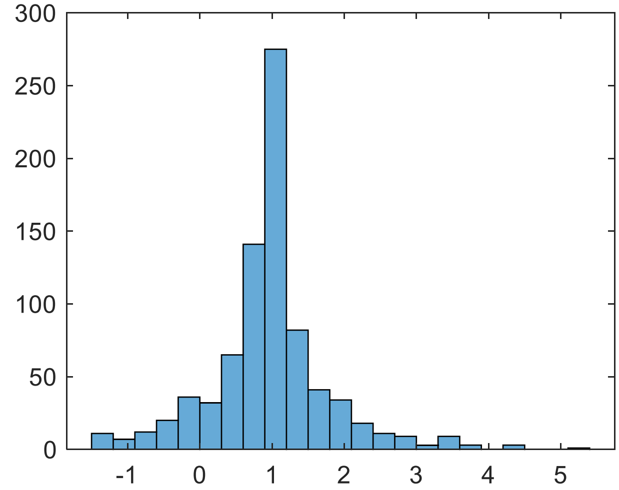

FIGURE 39
SPREAD OF OBSERVED ORDER OF ACCURACY P FOR OUTPUT VARIABLE 9

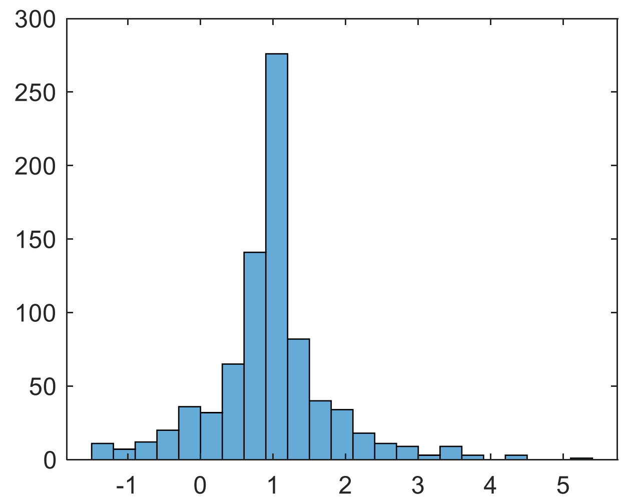

FIGURE 40
SPREAD OF OBSERVED ORDER OF ACCURACY P FOR OUTPUT VARIABLE 10

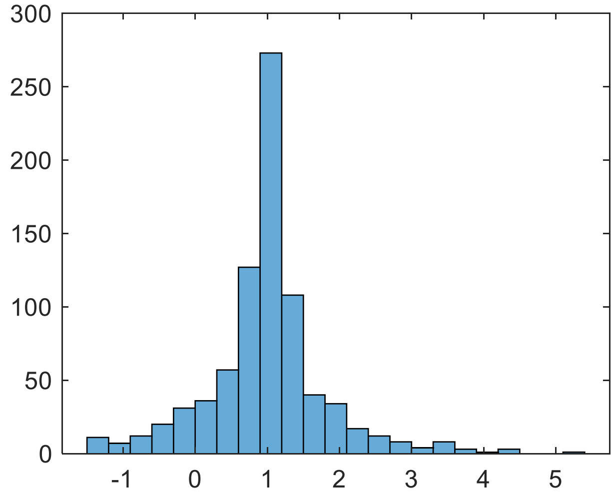

FIGURE 41
SPREAD OF OBSERVED ORDER OF ACCURACY P FOR OUTPUT VARIABLE 11

A grid triplet with N1=100 is chosen for the numerical uncertainty calculations and perturbation study for all output variables. This choice is reasonable because all output variables have p~1 where N1=100. However, other N1s also met this criterion. The output variables need not share the same grid triplet for validation, but this was done for simplicity. And practically, it makes most sense to make sure all output variables are reasonably converged for recommended use for an end-user.

B.I.iii. Grid Triplets and Simulation Results

Below is a zoomed in convergence plot of output variable 1 to show the grid triplet's location.

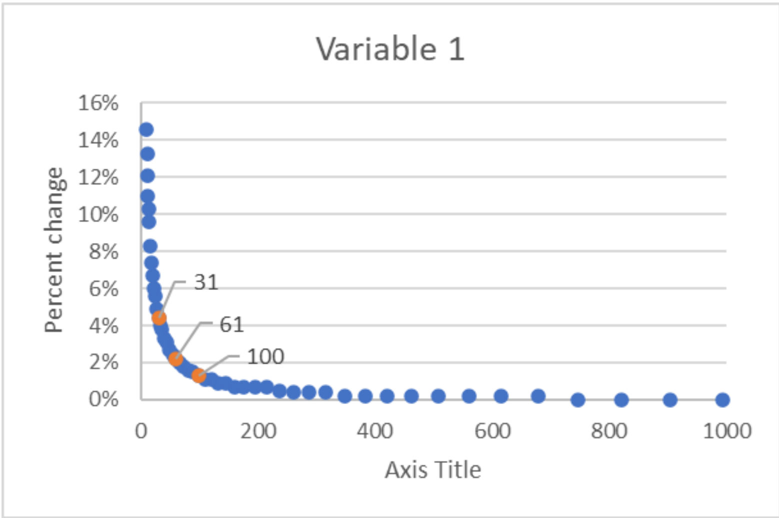

FIGURE 42
CONVERGENCE OF ERROR FOR OUTPUT VARIABLE 1

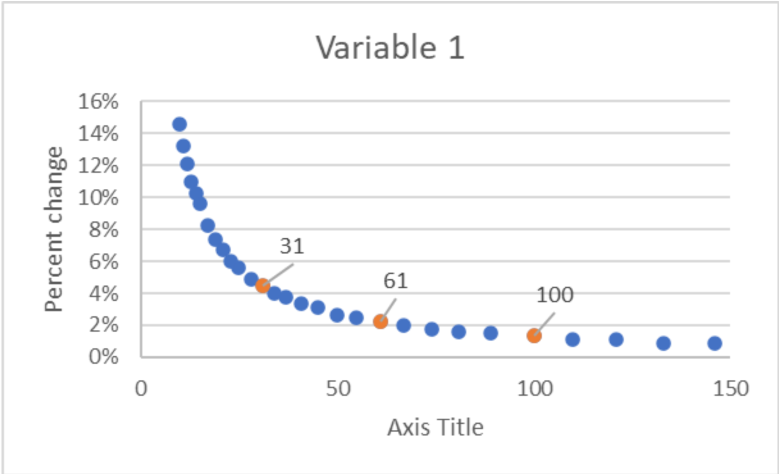

FIGURE 43
ZOOMED CONVERGENCE OF ERROR FOR OUTPUT VARIABLE 1

Below are summaries for each output variable's grid triplet simulation results.

TABLE 16

GRID TRIPLET RESULTS FOR OUTPUT VARIABLE 1

Grid	N	h	r	S	eps
1	100	0.006		1.0944	
			1.64		-1.05E-3
2	61	0.010		1.0934	
			1.97		-2.59E-3
3	31	0.019		1.0908	

TABLE 17

GRID TRIPLET RESULTS FOR OUTPUT VARIABLE 2

Grid	N	h	r	S	eps
1	100	0.006		946380	
			1.64		-2.57E+4
2	61	0.010		920640	
			1.97		-6.40E+4
3	31	0.019		856638	

TABLE 18

GRID TRIPLET RESULTS FOR OUTPUT VARIABLE 3

Grid	N	h	r	S	eps
1	100	0.006		14519	
			1.64		-1.41E+2
2	61	0.010		14378	
			1.97		-3.51E+2
3	31	0.019		14026	

TABLE 19

GRID TRIPLET RESULTS FOR OUTPUT VARIABLE 4

Grid	N	h	r	S	eps
1	100	0.006		2679	
			1.64		-5.65E+1
2	61	0.010		2622	
			1.97		-1.39E+2
3	31	0.019		2483	

TABLE 20

GRID TRIPLET RESULTS FOR OUTPUT VARIABLE 5

Grid	N	h	r	S	eps
1	100	0.006		10.1321	
			1.64		-1.63E-1
2	61	0.010		9.9688	
			1.97		-4.02E-1
3	31	0.019		9.5673	

TABLE 21

GRID TRIPLET RESULTS FOR OUTPUT VARIABLE 6

Grid	N	h	r	S	eps
1	100	0.006		0.47437	
			1.64		1.51E-3
2	61	0.010		0.47587	
			1.97		3.76E-3
3	31	0.019		0.47964	

TABLE 22

GRID TRIPLET RESULTS FOR OUTPUT VARIABLE 7

Grid	N	h	r	S	eps
1	100	0.006		4.758E-4	
			1.64		1.51E-6
2	61	0.010		4.773E-4	
			1.97		3.78E-6
3	31	0.019		4.811E-4	

TABLE 23

GRID TRIPLET RESULTS FOR OUTPUT VARIABLE 8

Grid	N	h	r	S	eps
1	100	0.006		12.600	
			1.64		4.00E-2
2	61	0.010		12.640	
			1.97		1.00E-1
3	31	0.019		12.740	

TABLE 24

GRID TRIPLET RESULTS FOR OUTPUT VARIABLE 9

Grid	N	h	r	S	eps
1	100	0.006		0.55753	
			1.64		-4.17E-4
2	61	0.010		0.55711	
			1.97		-1.03E-3
3	31	0.019		0.55608	

TABLE 25

GRID TRIPLET RESULTS FOR OUTPUT VARIABLE 10

Grid	N	h	r	S	eps
1	100	0.006		1.09445	
			1.64		-1.05E-3
2	61	0.010		1.09340	
			1.97		-2.59E-3
3	31	0.019		1.09081	

TABLE 26
GRID TRIPLET RESULTS FOR OUTPUT VARIABLE 11

Grid	N	h	r	S	eps
1	100	0.006		653260	
			1.64		-7.15E+2
2	61	0.010		652545	
			1.97		-1.79E+3
3	31	0.019		650750	

B.I.iv. Convergence Metrics

Below are each output variable's convergence metrics results.

TABLE 27
CONVERGENCE METRICS FOR OUTPUT VARIABLE 1

p	q	GCI
0.99	-4.14E-1	1.89E-3

TABLE 28
CONVERGENCE METRICS FOR OUTPUT VARIABLE 2

p	q	GCI
1.00	-4.15E-1	5.29E-2

TABLE 29
CONVERGENCE METRICS FOR OUTPUT VARIABLE 3

p	q	GCI
1.00	-4.15E-1	1.90E-2

TABLE 30
CONVERGENCE METRICS FOR OUTPUT VARIABLE 4

p	q	GCI
0.98	-4.13E-1	4.22E-2

TABLE 31
CONVERGENCE METRICS FOR OUTPUT VARIABLE 5

p	q	GCI
0.99	-4.13E-1	3.21E-2

TABLE 32
CONVERGENCE METRICS FOR OUTPUT VARIABLE 6

p	q	GCI
1.01	-4.16E-1	6.11E-3

TABLE 33
CONVERGENCE METRICS FOR OUTPUT VARIABLE 7

p	q	GCI
1.01	-4.16E-1	6.11E-3

TABLE 34
CONVERGENCE METRICS FOR OUTPUT VARIABLE 8

p	q	GCI
1.01	-4.16E-1	6.11E-3

TABLE 35
CONVERGENCE METRICS FOR OUTPUT VARIABLE 9

p	q	GCI
1.00	-4.14E-1	1.47E-3

TABLE 36
CONVERGENCE METRICS FOR OUTPUT VARIABLE 10

p	q	GCI
0.99	-4.14E-1	1.89E-3

TABLE 37
CONVERGENCE METRICS FOR OUTPUT VARIABLE 11

p	q	GCI
1.02	-4.17E-1	2.09E-3

B.I.v. Simulation Output Summary

Below is a summary of the accepted values of each output variable, unperturbed, at N=100.

TABLE 38
SIMULATION RESULTS FOR N=100

Y Index	Output Variable Name	Nominal Value (Y)	Units [U]
1	Direct Stiffness (Kxx)	1.95E+6	N/m
2	Cross-Coupled Stiffness (Kxy)	9.46E+5	N/m
3	Direct Damping (Cxx)	1.45E+4	Ns/m
4	Cross-Coupled Damping (Cxy)	2.68E+3	Ns/m
5	Moment (Mxx)	1.01E+1	Nm
6	Mass Flow	4.74E-1	kg/s
7	Volumetric Flow	4.76E-4	m3/s
8	Axial Velocity	1.26E+1	m/s
9	Inlet Loss	5.58E-1	
10	Outlet Recovery	1.09E+0	
11	Perturbed Pressure	6.53E+5	Pa

B.II. INPUT UNCERTAINTY

Below are the input parameters perturbed in the perturbation study. These are the same parameters used for the leakage study.

TABLE 39
INPUT PARAMETER INDICES AND NOMINAL VALUES

X Index	Input Parameter Name	Nominal Value (X)	Units [u]
1	Seal Radial Clearance	1.68E-4	m
2	Seal Length	6.00E-2	m
3	Rotor Radius	3.57E-2	m
4	Rotational Speed	3.00E+3	rpm
5	Density	9.97E+2	kg/m3
6	Dynamic Viscosity	8.90E-4	Pa s
7	Static Pressure Differential	7.84E+5	Pa
8	Friction Factor Coefficient	6.74E-2	
9	Preswirl	0.00E+0	

B.II.i. Perturbation Study Results

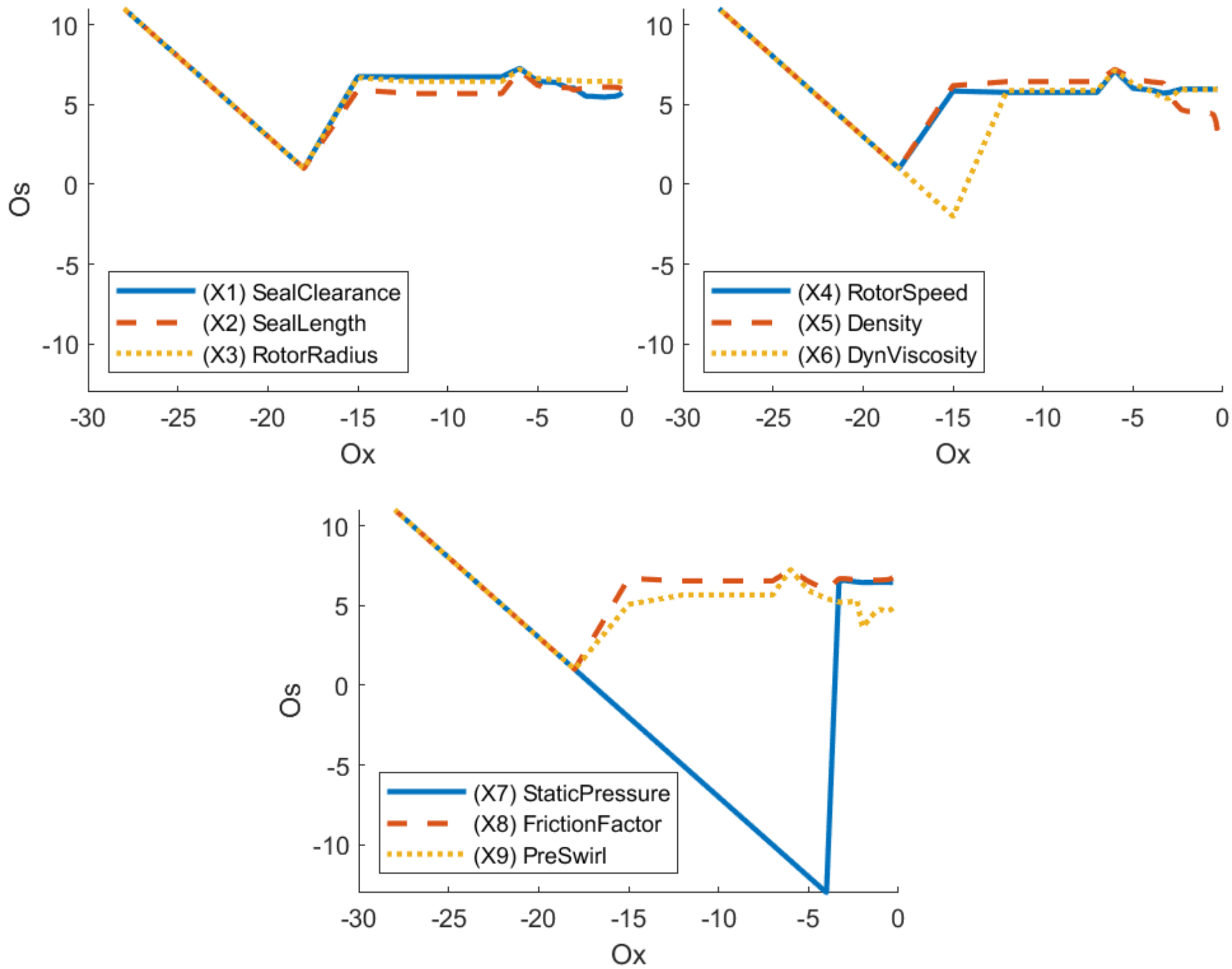

FIGURE 44

PERTRBATION RESULTS FOR OUTPUT VARIABLE 1

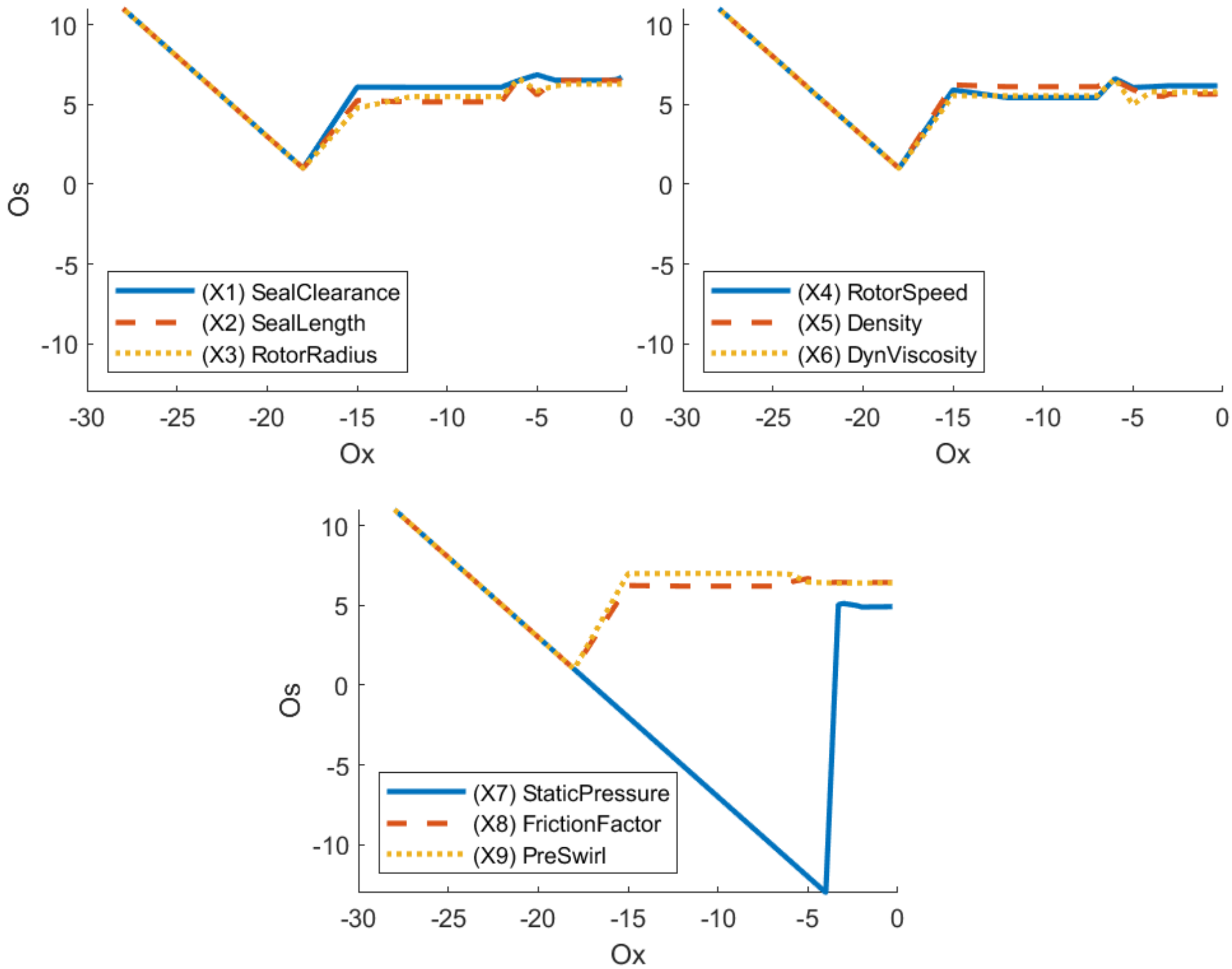

FIGURE 45
PERTRBATION RESULTS FOR OUTPUT VARIABLE 2

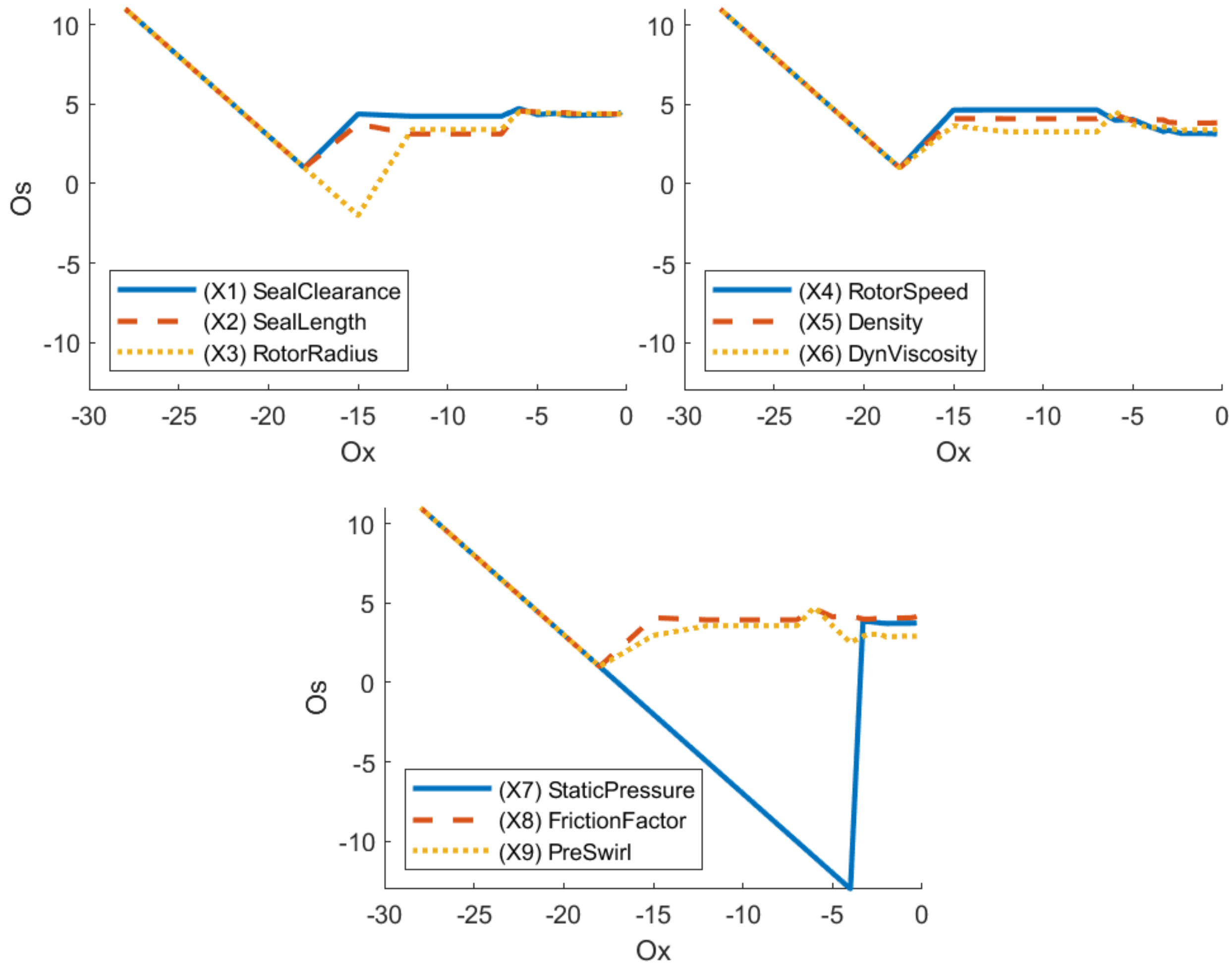

FIGURE 46
PERTRBATION RESULTS FOR OUTPUT VARIABLE 3

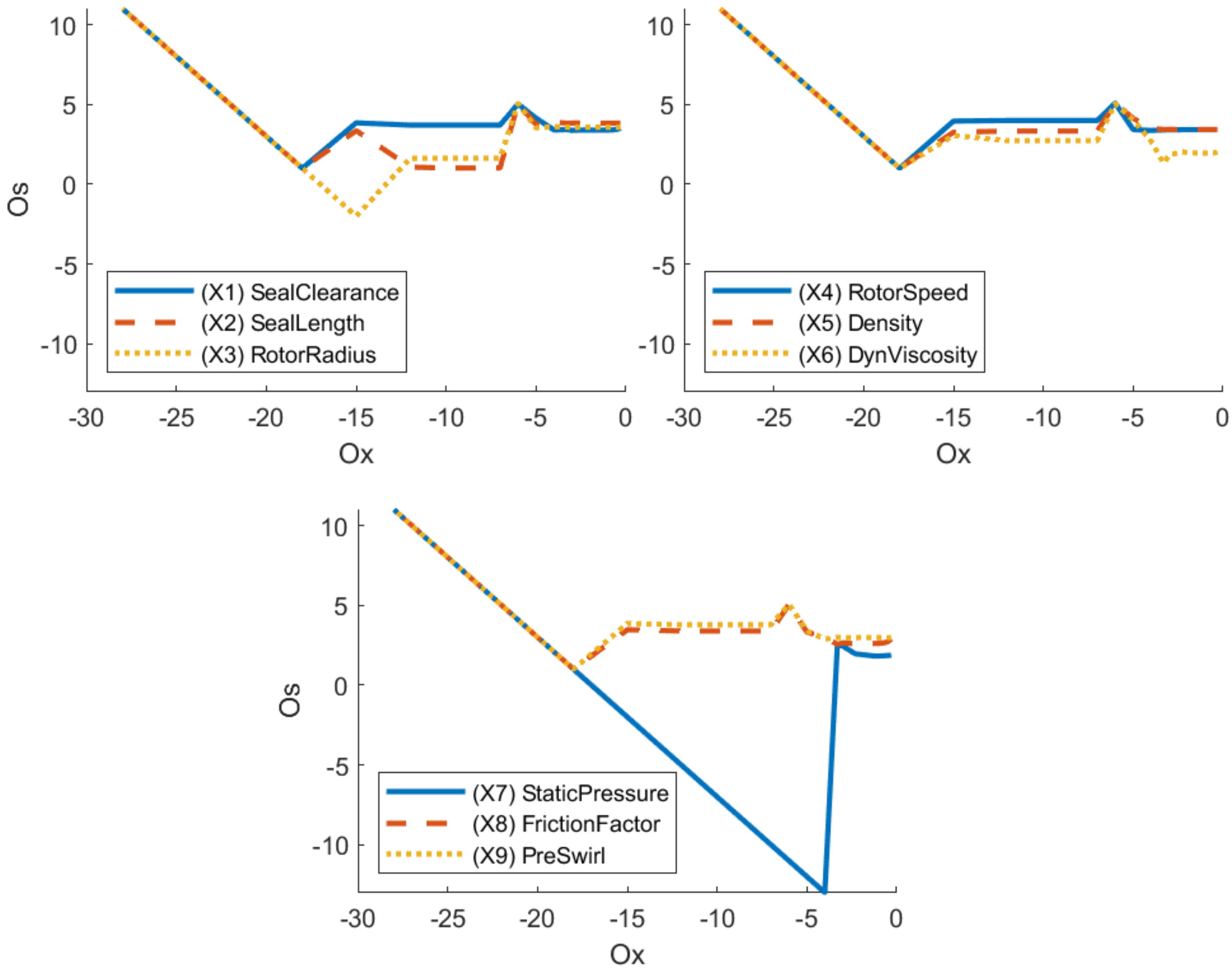

FIGURE 47
PERTRBATION RESULTS FOR OUTPUT VARIABLE 4

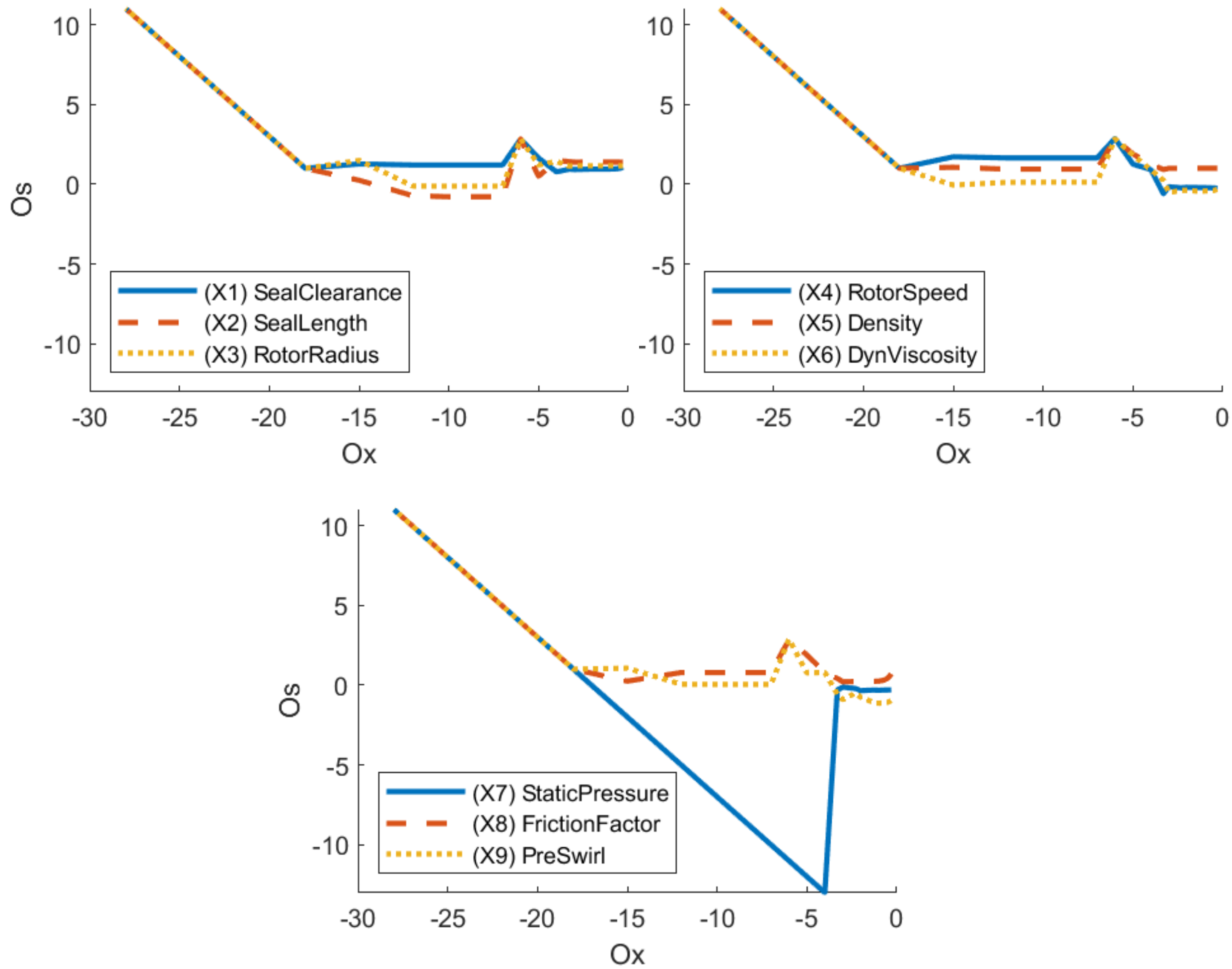

FIGURE 48
PERTRBATION RESULTS FOR OUTPUT VARIABLE 5

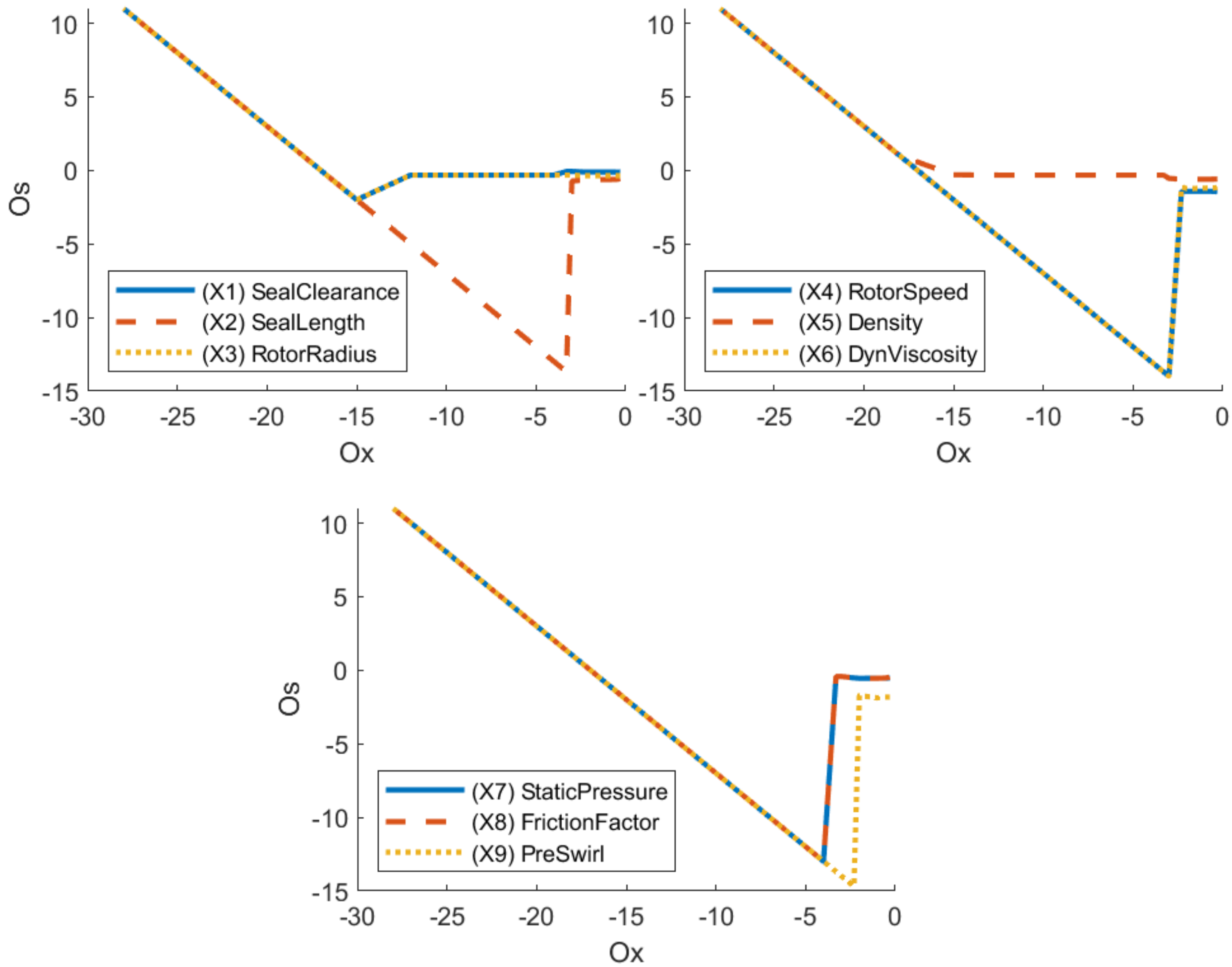

FIGURE 49
PERTRBATION RESULTS FOR OUTPUT VARIABLE 6

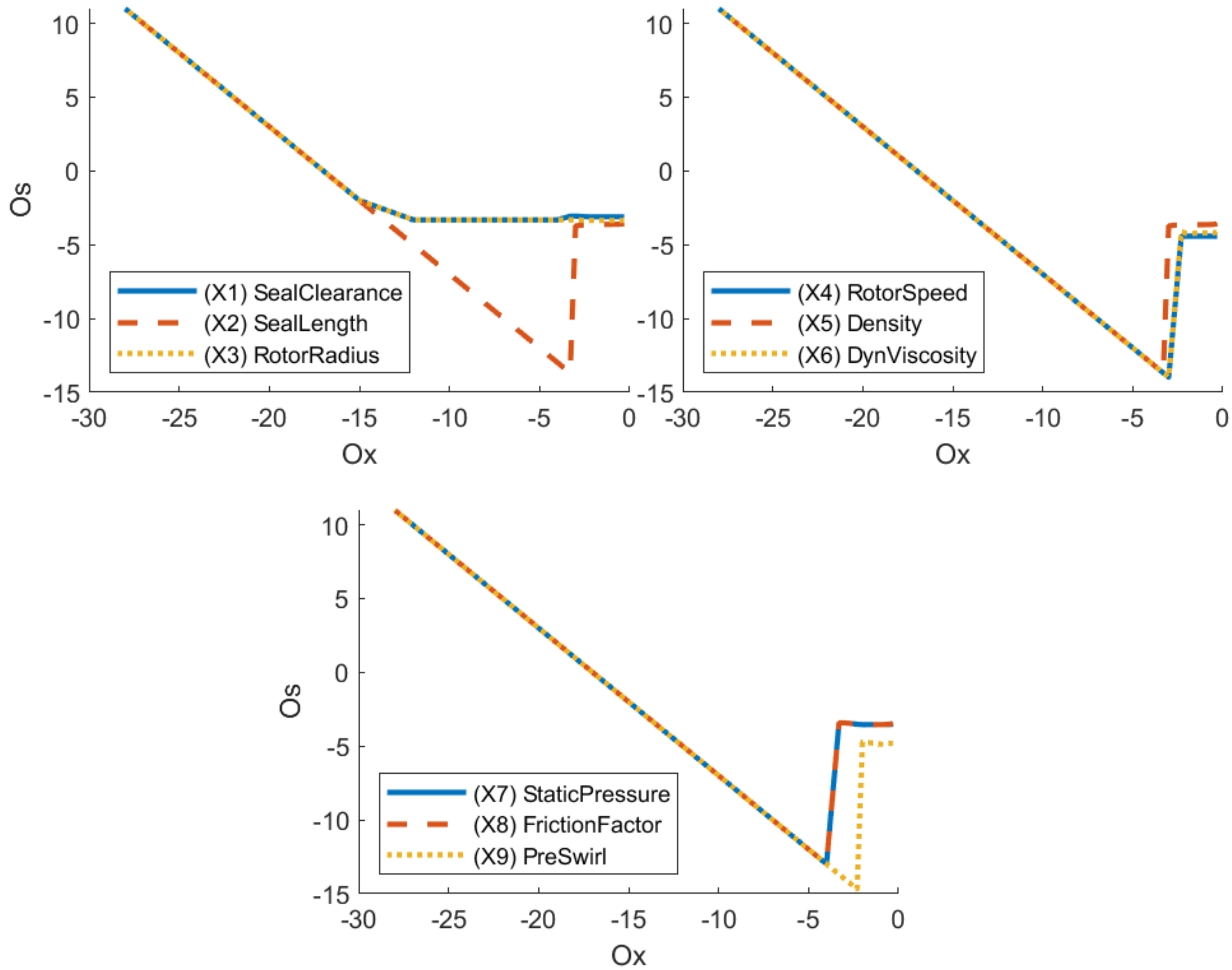

FIGURE 50
PERTRBATION RESULTS FOR OUTPUT VARIABLE 7

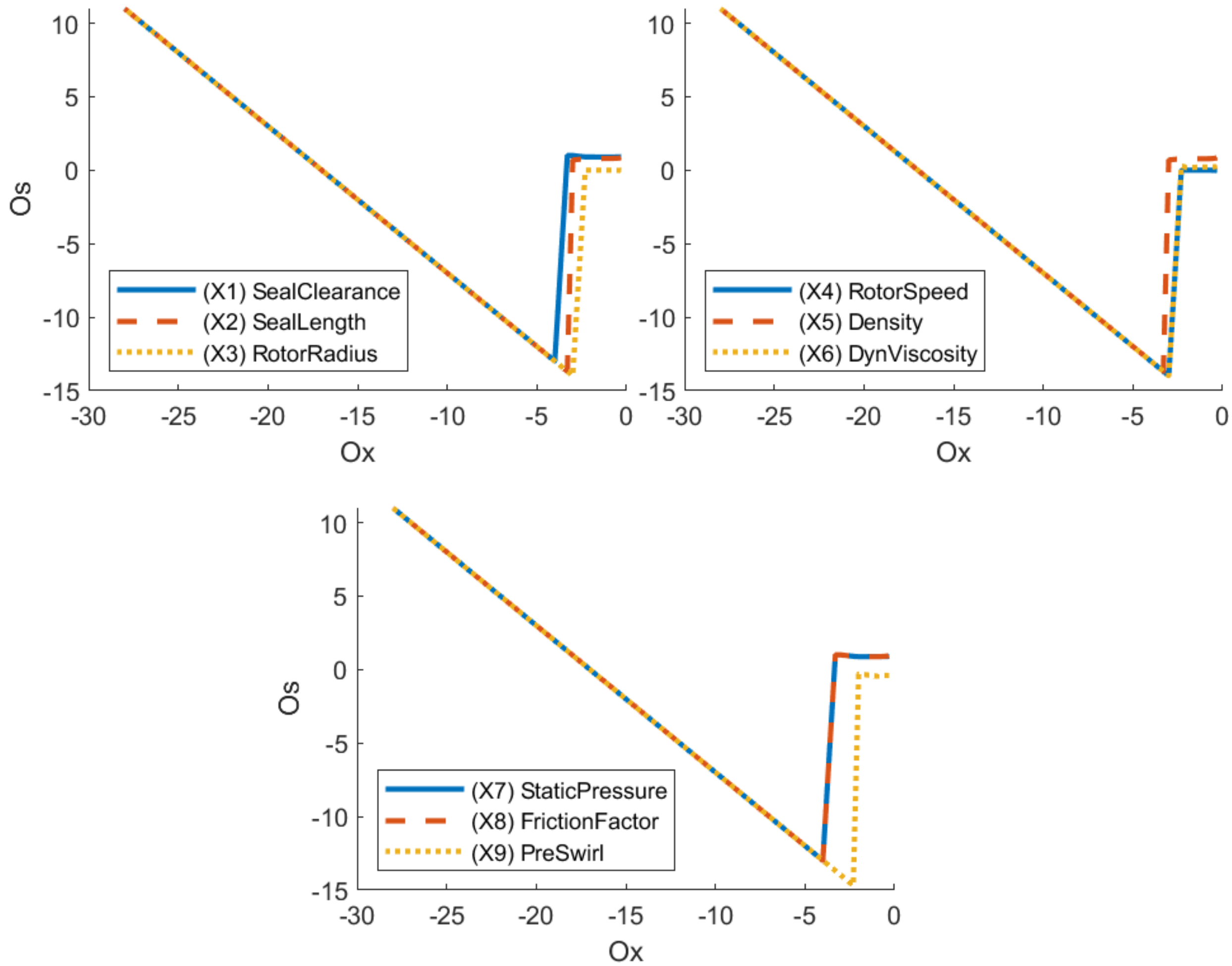

FIGURE 51
PERTRBATION RESULTS FOR OUTPUT VARIABLE 8

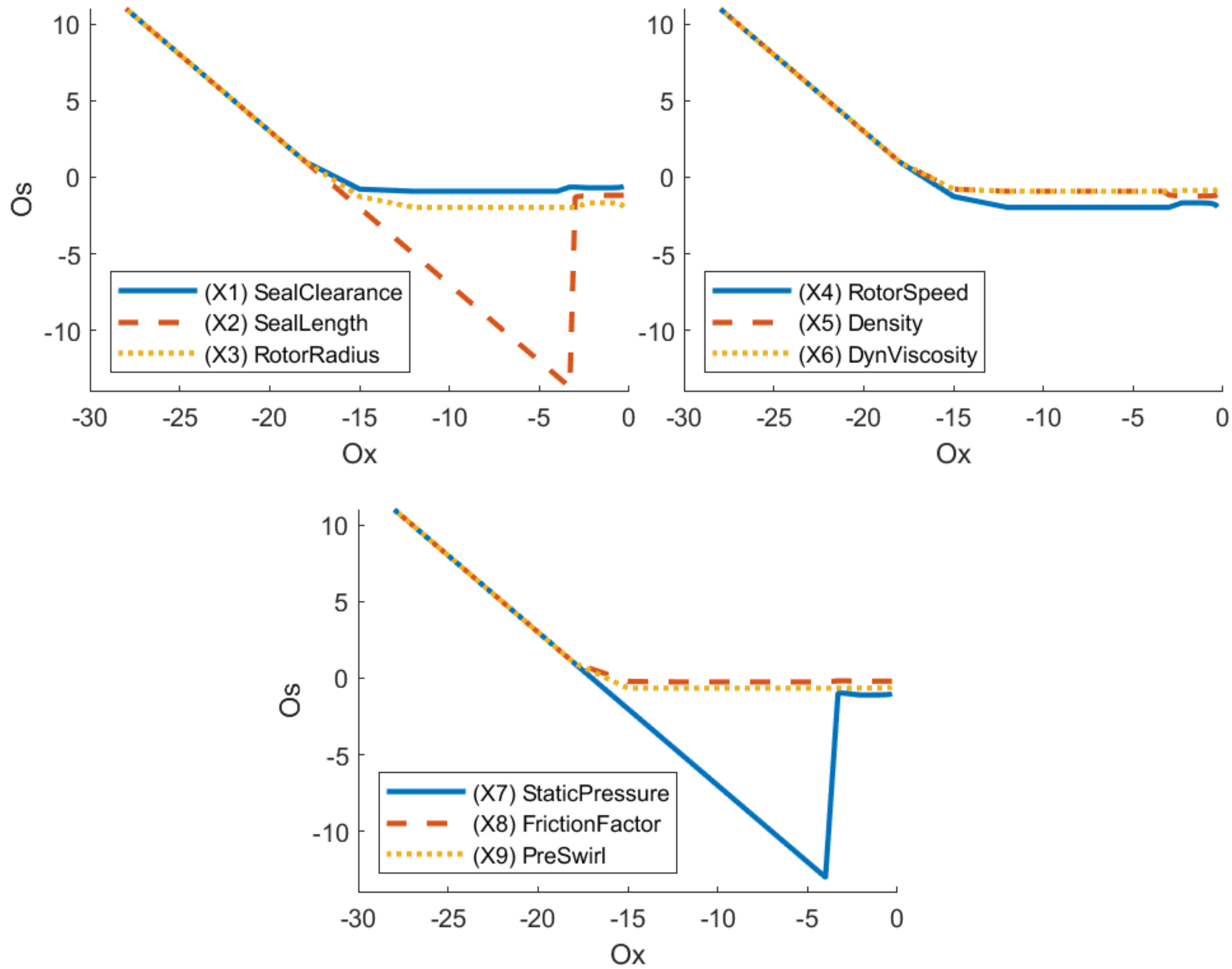

FIGURE 52
PERTRBATION RESULTS FOR OUTPUT VARIABLE 9

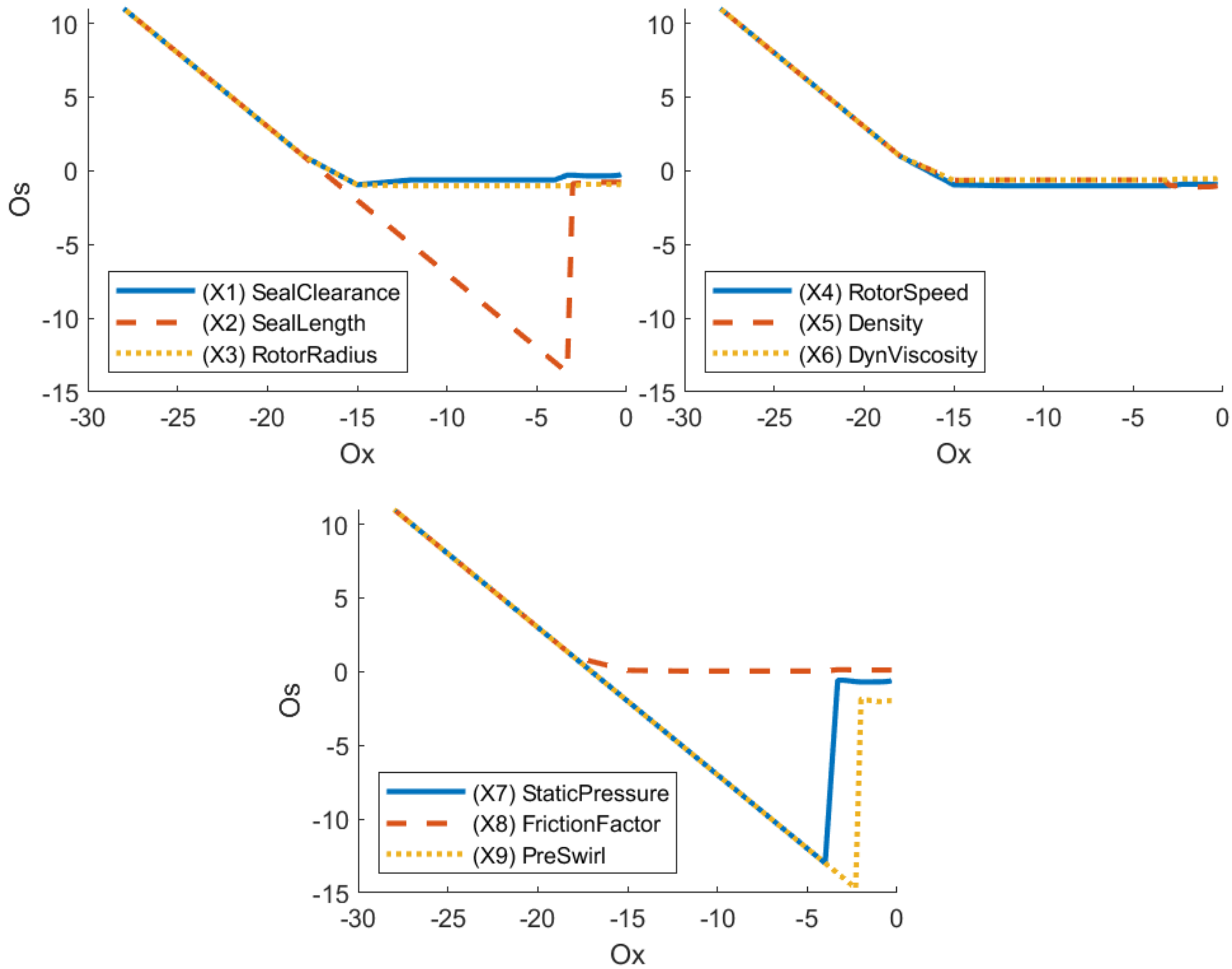

FIGURE 53
PERTRBATION RESULTS FOR OUTPUT VARIABLE 10

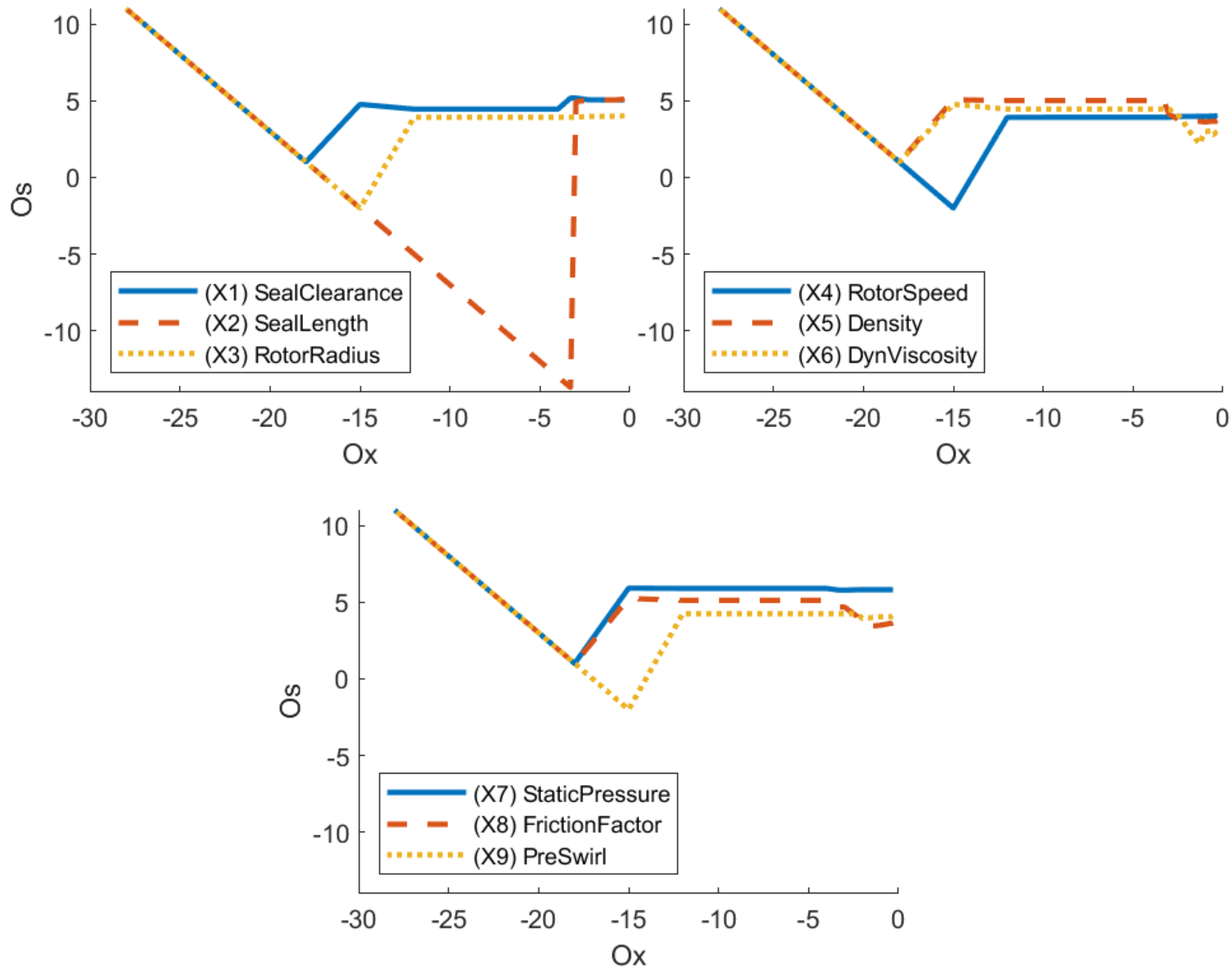

FIGURE 54
PERTRBATION RESULTS FOR OUTPUT VARIABLE 11

B.II.ii. Input Uncertainty Summaries

Below are tables for each output variable's perturbation study. The effects of each input parameter are detailed in each table.

TABLE 40

INPUT UNCERTAINTY SUMMARY FOR VARIABLE 1

	[u]	Xnom	Ux	dS/dX	Uinput
		[u]	[u]	[U]/[u]	[U]
X1	mm	0.168	0.005	1.41E+0	7.07E-3
X2	mm	60	0.005	2.63E-3	1.32E-5
X3	mm	35.6895	0.005	2.63E-3	1.32E-5
X4	rpm	3000	100	3.13E-5	3.13E-3
X5	kg m-3	997	2.5	2.38E-4	5.96E-4
X6	mPa s	0.89	0.25	2.67E-1	6.67E-2
X7	kPa	784	5	2.57E-4	1.28E-3
X8		0.0674	0.06	1.62E+1	9.74E-1
X9		0	0.1	9.22E-3	9.22E-4

TABLE 41

INPUT UNCERTAINTY SUMMARY FOR VARIABLE 2

	[u]	Xnom	Ux	dS/dX	Uinput
		[u]	[u]	[U]/[u]	[U]
X1	mm	0.168	0.005	7.08E+6	3.54E+4
X2	mm	60	0.005	2.38E+3	1.19E+1
X3	mm	35.6895	0.005	8.64E+3	4.32E+1
X4	rpm	3000	100	8.99E+1	8.99E+3
X5	kg m-3	997	2.5	1.30E+3	3.24E+3
X6	mPa s	0.89	0.25	3.91E+5	9.76E+4
X7	kPa	784	5	9.96E+1	4.98E+2
X8		0.0674	0.06	2.38E+7	1.43E+6
X9		0	0.1	9.96E+6	9.96E+5

TABLE 42

INPUT UNCERTAINTY SUMMARY FOR VARIABLE 3

	[u]	Xnom	Ux	dS/dX	Uinput
		[u]	[u]	[U]/[u]	[U]
X1	mm	0.168	0.005	1.02E+5	5.11E+2
X2	mm	60	0.005	2.20E+1	1.10E-1
X3	mm	35.6895	0.005	7.01E+1	3.50E-1
X4	rpm	3000	100	1.52E+1	1.52E+3
X5	kg m-3	997	2.5	1.27E+1	3.17E+1
X6	mPa s	0.89	0.25	2.11E+3	5.28E+2
X7	kPa	784	5	6.62E+0	3.31E+1
X8		0.0674	0.06	1.29E+5	7.72E+3
X9		0	0.1	3.69E+3	3.69E+2

TABLE 43

INPUT UNCERTAINTY SUMMARY FOR VARIABLE 4

	[u]	Xnom	Ux	dS/dX	Uinput
		[u]	[u]	[U]/[u]	[U]
X1	mm	0.168	0.005	3.01E+4	1.51E+2
X2	mm	60	0.005	1.74E-1	8.72E-4
X3	mm	35.6895	0.005	1.20E+0	5.98E-3
X4	rpm	3000	100	3.28E+0	3.28E+2
X5	kg m-3	997	2.5	2.15E+0	5.37E+0
X6	mPa s	0.89	0.25	6.03E+2	1.51E+2
X7	kPa	784	5	1.07E-1	5.34E-1
X8		0.0674	0.06	3.67E+4	2.20E+3
X9		0	0.1	6.07E+3	6.07E+2

TABLE 44

INPUT UNCERTAINTY SUMMARY FOR VARIABLE 5

	[u]	Xnom	Ux	dS/dX	Uinput
		[u]	[u]	[U]/[u]	[U]
X1	mm	0.168	0.005	2.83E+0	1.41E-2
X2	mm	60	0.005	3.76E-3	1.88E-5
X3	mm	35.6895	0.005	1.33E-2	6.63E-5
X4	rpm	3000	100	1.25E-5	1.25E-3
X5	kg m-3	997	2.5	4.76E-4	1.19E-3
X6	mPa s	0.89	0.25	6.35E-2	1.59E-2
X7	kPa	784	5	3.65E-4	1.82E-3
X8		0.0674	0.06	4.22E+0	2.53E-1
X9		0	0.1	1.32E-2	1.32E-3

TABLE 45

INPUT UNCERTAINTY SUMMARY FOR VARIABLE 6

	[u]	Xnom	Ux	dS/dX	Uinput
		[u]	[u]	[U]/[u]	[U]
X1	mm	0.168	0.005	9.59E+1	4.80E-1
X2	mm	60	0.005	2.74E-3	1.37E-5
X3	mm	35.6895	0.005	2.12E-2	1.06E-4
X4	rpm	3000	100	1.49E-2	1.49E+0
X5	kg m-3	997	2.5	8.82E-3	2.21E-2
X6	mPa s	0.89	0.25	1.50E+0	3.75E-1
X7	kPa	784	5	5.78E-4	2.89E-3
X8		0.0674	0.06	9.13E+1	5.48E+0
X9		0	0.1	1.11E+0	1.11E-1

TABLE 46
INPUT UNCERTAINTY SUMMARY FOR VARIABLE 7

	[u]	Xnom	Ux	dS/dX	Uinput
		[u]	[u]	[U]/[u]	[U]
X1	mm	0.168	0.005	4.60E-3	2.30E-5
X2	mm	60	0.005	4.03E-6	2.01E-8
X3	mm	35.6895	0.005	1.22E-5	6.11E-8
X4	rpm	3000	100	1.26E-8	1.26E-6
X5	kg m-3	997	2.5	2.32E-7	5.80E-7
X6	mPa s	0.89	0.25	7.11E-5	1.78E-5
X7	kPa	784	5	3.68E-7	1.84E-6
X8		0.0674	0.06	4.33E-3	2.60E-4
X9		0	0.1	1.51E-5	1.51E-6

TABLE 47
INPUT UNCERTAINTY SUMMARY FOR VARIABLE 8

	[u]	Xnom	Ux	dS/dX	Uinput
		[u]	[u]	[U]/[u]	[U]
X1	mm	0.168	0.005	4.73E+1	2.37E-1
X2	mm	60	0.005	1.05E-1	5.25E-4
X3	mm	35.6895	0.005	2.80E-2	1.40E-4
X4	rpm	3000	100	3.33E-4	3.33E-2
X5	kg m-3	997	2.5	6.02E-3	1.50E-2
X6	mPa s	0.89	0.25	1.85E+0	4.63E-1
X7	kPa	784	5	9.69E-3	4.85E-2
X8		0.0674	0.06	1.12E+2	6.72E+0
X9		0	0.1	3.50E-1	3.50E-2

TABLE 48
INPUT UNCERTAINTY SUMMARY FOR VARIABLE 9

	[u]	Xnom	Ux	dS/dX	Uinput
		[u]	[u]	[U]/[u]	[U]
X1	mm	0.168	0.005	7.20E-1	3.60E-3
X2	mm	60	0.005	1.05E-3	5.23E-6
X3	mm	35.6895	0.005	3.00E-4	1.50E-6
X4	rpm	3000	100	3.57E-6	3.57E-4
X5	kg m-3	997	2.5	1.21E-4	3.03E-4
X6	mPa s	0.89	0.25	1.36E-1	3.40E-2
X7	kPa	784	5	1.00E-4	5.00E-4
X8		0.0674	0.06	8.27E+0	4.96E-1
X9		0	0.1	2.21E-1	2.21E-2

TABLE 49

INPUT UNCERTAINTY SUMMARY FOR VARIABLE 10

	[u]	Xnom	Ux	dS/dX	Uinput
		[u]	[u]	[U]/[u]	[U]
X1	mm	0.168	0.005	1.41E+0	7.07E-3
X2	mm	60	0.005	2.63E-3	1.32E-5
X3	mm	35.6895	0.005	2.63E-3	1.32E-5
X4	rpm	3000	100	3.13E-5	3.13E-3
X5	kg m-3	997	2.5	2.38E-4	5.96E-4
X6	mPa s	0.89	0.25	2.67E-1	6.67E-2
X7	kPa	784	5	2.57E-4	1.28E-3
X8		0.0674	0.06	1.62E+1	9.74E-1
X9		0	0.1	9.22E-3	9.22E-4

TABLE 50

INPUT UNCERTAINTY SUMMARY FOR VARIABLE 11

	[u]	Xnom	Ux	dS/dX	Uinput
		[u]	[u]	[U]/[u]	[U]
X1	mm	0.168	0.005	1.69E+5	8.44E+2
X2	mm	60	0.005	1.88E+3	9.39E+0
X3	mm	35.6895	0.005	2.32E+2	1.16E+0
X4	rpm	3000	100	2.76E+0	2.76E+2
X5	kg m-3	997	2.5	1.03E+2	2.57E+2
X6	mPa s	0.89	0.25	3.19E+4	7.97E+3
X7	kPa	784	5	1.00E+3	5.00E+3
X8		0.0674	0.06	1.94E+6	1.16E+5
X9		0	0.1	1.75E+4	1.75E+3

B.III. SUMMARY

Below is a summary of the results of this study. For each output variable, the numerical and input uncertainties are tabulated.

TABLE 51
NUMERICAL AND INPUT UNCERTAINTY FOR ALL OUTPUT VARIABLES

Y Index	Output Variable Name	Nominal Value (Y)	Units [U]	Unum		Uinput	
				[U]	%Y	[U]	%Y
1	Direct Stiffness (Kxx)	1.95E+6	N/m	1.79E-3	9E-10	9.77E-1	5E-07
2	Cross-Coupled Stiffness (Kxy)	9.46E+5	N/m	4.24E+4	4%	1.74E+6	184%
3	Direct Damping (Cxx)	1.45E+4	Ns/m	2.37E+2	2%	7.91E+3	54%
4	Cross-Coupled Damping (Cxy)	2.68E+3	Ns/m	9.61E+1	4%	2.32E+3	87%
5	Moment (Mxx)	1.01E+1	Nm	2.78E-1	3%	5.71E+0	56%
6	Mass Flow	4.74E-1	kg/s	2.53E-3	1%	2.54E-1	54%
7	Volumetric Flow	4.76E-4	m3/s	2.54E-6	1%	2.61E-4	55%
8	Axial Velocity	1.26E+1	m/s	6.72E-2	1%	6.74E+0	54%
9	Inlet Loss	5.58E-1		7.13E-4	0.1%	4.98E-1	89%
10	Outlet Recovery	1.09E+0		1.79E-3	0.2%	9.77E-1	89%
11	Perturbed Pressure	6.53E+5	Pa	1.18E+3	0.2%	1.17E+5	18%

Appendix C: MATLAB Scripts

C.I. MESH INDEPENDENCE STUDY (EXAMPLE)

C.I.i. Code

```
%% SETUP

close all
clear all
clc

Nmin=10;
Nmax=1000;
Nfactor=1.1;
location=' --- ';

%% nominal inputs
rotationspeed=3000;         %speed of the rotor in rpm
R = (71.379/2)/10^3;                  %radius of rotor in m
viscosity=8.9*10^-4;              %viscosity in Pa*s
C = .168/10^3;                 %seal radial clearance in m
L = 60/10^3;                  %seal length in m
rho = 997;                  %density of liquid in kg/m^3
deltaP = 784*10^3;          %pressure differential in Pa %+DYN=881.5kPa

%advanced inputs
preswirl=       0;  %ratio of inlet circumferential velocity to rotor surface
speed
a=          0.0674;       %.0674;    %friction factor -- this value is
approximately from Blasius

whirlratio=    1;   %maximum whirl speed simulated as fraction of rotor speed
-- should not impact the results unless extreme

inputvector=[C           L           R           rotationspeed   rho ...
             viscosity   deltaP      a           preswirl                ];

%%

n=1;
N=Nmin;
L=.6;
while N<=Nmax
    if N==98
        N=100;
    end
    hv(n)=(L/N);
    [outvec]=fBulk(inputvector,N);
    for i=1:length(outvec)
        dat(i).val(n,:)=[N outvec(i) hv(n)];        %store leakage (row=dX,
col=Xi)
    end
    n=n+1;
```

```
    N=round(N*(Nfactor));
end

%%
for i=1:length(outvec)
    num=flip(dat(i).val); %order h small to large
    array2table(num,'VariableNames',{'N' 'S' 'h'});
    num=num2cell(num)
    row1={'N','S','h'};
    row2={'nodes','U','mm'};
    txt=[row1;row2];
    D=[txt;num];
    filename=strcat(location,'MISRes_Code_Bulk_',num2str(i),'.xlsx');
    writecell(D,filename);
end

%% END
```

C.II. PERTURBATION STUDY (EXAMPLE)

C.II.i. Code

```
function [Ydat] = fPertBulk(location,dX_ND)


%% BEGIN

%% INPUT PARAMETERS

%nominal inputs
rotationspeed=3000;          %speed of the rotor in rpm
R = (71.379/2)/10^3;                   %radius of rotor in m
viscosity=8.9*10^-4;               %viscosity in Pa*s
C = .168/10^3;                  %seal radial clearance in m
L = 60/10^3;                   %seal length in m
rho = 997;                 %density of liquid in kg/m^3
deltaP = 784*10^3;           %pressure differential in Pa %+DYN=881.5kPa

%advanced inputs
preswirl=       0;  %ratio of inlet circumferential velocity to rotor surface
speed
a=          0.0674;       %.0674;    %friction factor -- this value is
approximately from Blasius

%setup vector of nominal input parameter values
Xnom=[  C           L           R           rotationspeed   rho ...
        viscosity   deltaP      a           preswirl            ];
ux=[    0.005       0.005       0.005       100             2.5 ...
        0.25        5           0.06        0.1                 ];
Xnames={'SealClearance','SealLength','RotorRadius','RotorSpeed', ...
    'Density','DynViscosity','StaticPressure','FrictionFactor','PreSwirl'};
Xunit={ 'm',        'm',        'm',        'rpm',          'kg m-3',...
        'Pa s',     'Pa',       ' ',        ' '};
Xunit2={'mm',       'mm',       'mm',       'rpm',          'kg m-3',...
        'mPa s',    'kPa',      ' ',        ' '                     };
Xnom2=[ 1000        1000        1000        1               1 ...
        1000        1/1000      1           1
].*Xnom;

data_vars=[Xnames,'Leakage', 'Power', 'Parameter']; %columns of dat

%% PERTURB AND STORE OUTPUT

dX_ND_abs=dX_ND(dX_ND>0);
for ind_Xi=1:length(Xnom) %step through parameters to perturb
    for ind_dX=1:length(dX_ND) %step through dX for Xi
        %INPUT PARAMETERS
            X=Xnom;   %initially set to nominal parameter values
            X(ind_Xi)=X(ind_Xi)*(1+dX_ND(ind_dX)); %adjust the Xi in X
            if X(ind_Xi)==0
                X(ind_Xi)=dX_ND(ind_dX);
            end
        %OUTPUT VALUES
```

```
            [outvec] = fBulk(X,100);
            for i=1:length(outvec)
               Ydat(i).val(ind_dX,ind_Xi)=outvec(i);        %store leakage
(row=dX, col=Xi)
            end
    end
end

for i=1:length(outvec)
    num=[ux;Xnom2;Ydat(i).val];
    num=num2cell(num);
    v=[{[ ]};{[ ]};num2cell(dX_ND')];
    mat=[v num];
    txt=[{'dX_ND'} Xnames;{' '} Xunit2];
    D=[txt;mat];
    filename=strcat(location,'PertRes_Code_Bulk_',num2str(i),'.xlsx');
    writecell(D,filename);
end

%% END

end
```

C.III. NUMERICAL UNCERTAINTY ANALYSIS (GENERAL)

C.III.i. Documentation

```
%% NUMERICAL UNCERTAINTY GENERAL CODE
%
% Written by Madeline Carlisle Collins
% 2019-2020
%
% Purpose:
% Takes in raw data from mesh independence study. Mesh indenpendence study
% should include various mesh sizes. The elements should change in all
% directions. Structured refinement requires that the grid change by the
% same factor in all directions. This code calculates the observed order of
% accuracy, the grid convergence index, and the numerical uncertainty for
% all combinations of 3 meshes allowable. The user chooses a specific grid
% triplet as an input to this function. That choice specifies the formal
% table output of this code. However, all grid triplet outcomes are output.
%
%
%EXAMPLE EXCEL SHEET
% N             S           h
% elements      L/s         mm
% 1000          0.507       0.216
% 900           0.506       0.220
% 750           0.509       0.252
% 500           0.511       0.273

% The first row simply labels the columns.
% The second row is the units for the columns.
% The third row is where the numerical data begins.
% The first column is the number of elements or nodes.
% The second column is the simulation output at that number of nodes.
% The third column is the representative mesh element size in one
% dimension. This is calculated from N according to V&V20-2009 formulas.

%excelfile: text string, location of excel file and file itself
%location: text string, location path for output files to be saved
%   (includes tables, plots, graphs)
%nameform: text string, base name of output files
%SimSetCho_ind: grid triplet index #, chosen from AllSets output sheet,
%   this grid triplet is then printed out with its results
%chotable: binary 1 or 0, 1 exports the chosen triplets results and
%   overwrites previous one, 0 does nothing
%options: binary 1 or 0, 1 exports the all grid triplets results onto a
%   sheet and overwrites previous one, 0 does nothing
%doplot: binary 1 or 0, 1 plots S vs N, 0 does nothing
```

C.III.ii. Code

```
function [] =
fNumericalVV(excelfile,location,nameform,SimSetCho_ind,chotable,options,doplo
t ,wellBehaved,structured)

%% SETUP

[num,text,raw]=xlsread(excelfile);
var_names=raw(1,1:end);
var_units=raw(2,1:end);
dat=num;
Nv=dat(1:end,1);
Sv=dat(1:end,2);
hv=dat(1:end,3);

%% CALCULATIONS

r_lo=1.3;
r_hi=2;

n=1;
for i=2:(length(hv)-1)
    for j=1:(i-1)
        if (hv(i)/hv(j)>=r_lo) && (hv(i)/hv(j)<=r_hi)
            for k=(i+1):length(hv)
                if (hv(k)/hv(i)>=r_lo) && (hv(k)/hv(i)<=r_hi)
                    loc=[j i k];
                    h=hv(loc);
                    S=Sv(loc);
                    N=Nv(loc);
                    r21=h(2)/h(1);
                    r32=h(3)/h(2);
                    e21=S(2)-S(1);
                    e32=S(3)-S(2);

                    sg=sign(e32/e21);
                    q=0.1;
                    p=(q+log(abs(e32/e21)))/log(r21);
                    error=1;
                    while error>0.001
                        q=log((r21^p-sg)/(r32^p-sg));
                        pnew=(q+log(abs(e32/e21)))/log(r21);
                        error=abs(pnew-p)/p;
                        p=pnew;
                    end
                    if isnan(p)==0

                        if structured==1
                            Fs=1.25; %structured grid refinement
                        elseif structured==0
                            Fs=3; % UN-structured
                        end
                        ea21=abs(e21); %changed 2/27/20  %ea21=abs(e21/S(1));
                        GCI=Fs*ea21/(r21^p-1);
```

```
                            if wellBehaved==1
                                k_expansion=1.15;
                            % POORLY behaved
                            elseif wellBehaved==0
%                                     S21ext=(r21^p*S(1)-S(2))/(r21^p-1);
%                                     ea21ext=abs((S21ext-S(1)));
%                                     GCI=Fs*ea21ext/(r21^p-1);
                                k_expansion=2;
                            end

                        u_num=abs(GCI/k_expansion); %u_numND=abs(GCI/k);
%u_num=u_numND*Sv(i); %changed 2/27/20

                        results(n,:)= [N(1) N(2) N(3) h(1) h(2) h(3) S(1)
S(2) S(3) ...
                            r21 r32 e21 e32 ea21 p q GCI u_num];
                        n=n+1;
                    end
                end
            end
        end
    end
end

%% NUMERICAL UNCERTAINTY CALCULATION RESULTS

%SimSetCho_res=1:(n-1);
AllSets_ind=1:(n-1);

if options==1
    AllSets_labs={'N1' 'N2' 'N3' 'h1' 'h2' 'h3' 'S1' 'S2' 'S3' 'r21' 'r32'
...
        'e21' 'e32' 'ea21' 'p' 'q' 'GCI' 'u_num'};

    %TABLE WITH EVERYTHING
    AllSets_ind=1:(n-1);
    AllSets_mat=[AllSets_ind' results];

AllSets_table=array2table(AllSets_mat,'VariableNames',['set',AllSets_labs]);

    %PRINT TO TEXT FILE
    filename=strcat(location,nameform,'_AllSets.txt');
    writetable(AllSets_table, filename);

    %PRINT TO EXCEL FILE
    filename=strcat(location,nameform,'_AllSets.xlsx');
    AllSets_indiv=num2cell(AllSets_mat);
    AllSets_xlsx=[['set',AllSets_labs]; AllSets_indiv];
    writecell(AllSets_xlsx,filename);
end

%% PLOT
if doplot==1
    all_ps=results(:,15);
    figure
    histogram(all_ps,'BinWidth',1)
        x0=100;
```

```
        y0=100;
        width=350;
        height=250;
        set(gcf,'position',[x0,y0,width,height])
    xlabel('(p) observed order of accuracy');
    ylabel('frequency of p');
    filename=strcat(location,nameform,'_SpreadOfP.png');
    saveas(gcf,filename);

    figure
    plot(Nv,Sv);
    xlabel('(N) number of nodes/elements');
    ylabel('(S) simulation results');
    filename=strcat(location,nameform,'_SimResPlot.png');
    saveas(gcf,filename);
end


%% CHOSEN SET DETAILS
%  N, h, r, S, eps

    %BUILD MATRIX
    SimSetCho_res=results(SimSetCho_ind,:);
    SimSetCho_indiv=cell(5,6);
    SimSetCho_indiv(1:5,1:6)=[{' '}];
    SimSetCho_indiv(:,1)=[{1}, {' '},{2},{' '},{3}];
    SimSetCho_indiv(:,2)=[{SimSetCho_res(1)},{' '},{SimSetCho_res(2)},{'
'},{SimSetCho_res(3)}];
    SimSetCho_indiv(:,3)=[{SimSetCho_res(4)},{' '},{SimSetCho_res(5)},{'
'},{SimSetCho_res(6)}];
    SimSetCho_indiv(:,4)=[{' '},{SimSetCho_res(10)},{'
'},{SimSetCho_res(11)},{' '}];
    SimSetCho_indiv(:,5)=[{SimSetCho_res(7)},{' '},{SimSetCho_res(8)},{'
'},{SimSetCho_res(9)}];
    SimSetCho_indiv(:,6)=[{' '},{SimSetCho_res(12)},{'
'},{SimSetCho_res(13)},{' '}];

    %PRINT IN MATLAB
    SimSetCho_labs={'Grid','N','h','r','S','eps'};
    array2table(SimSetCho_indiv,'VariableNames',SimSetCho_labs)

    if chotable==1
        %PRINT TO EXCEL FILE
        SimSetCho_xlsx(1:7,1:7)={' '};
        SimSetCho_xlsx(2,2:end)=SimSetCho_labs;
        SimSetCho_xlsx(3:end,2:end)=SimSetCho_indiv;
        filename=strcat(location,nameform,'_SimSetChosen.xlsx');
        writecell(SimSetCho_xlsx,filename);
    end

    %BUILD MATRIX
    pqGCI_num=[SimSetCho_res(15:17)];
    pqGCI_labs={'p','q','GCI'};
    pqGCI_indiv=num2cell(pqGCI_num);
    pqGCI_table=[pqGCI_labs;pqGCI_indiv];

    %PRINT IN MATLAB
```

```
    array2table(pqGCI_num,'VariableNames',pqGCI_labs)

    if chotable==1
        %PRINT TO FILE
        pqGCI_xlsx(1:3,1:4)={' '};
        pqGCI_xlsx(2:end,2:end)=pqGCI_table;
        filename=strcat(location,nameform,'_p,q,GCI.xlsx');
        writecell(pqGCI_xlsx,filename);
    end

%%
k_expansion
Unum=SimSetCho_res(18)
filename=strcat(location,nameform,'_NumericalUncertainty.xls');
xlswrite(filename, Unum)

end

%% END
```

C.IV. INPUT UNCERTAINTY ANALYSIS (GENERAL)

C.IV.i. Documentation

```
%% INPUT UNCERTAINTY GENERAL CODE
%
% Written by Madeline Carlisle Collins
% 2019-2020
%
% Purpose:
% Takes in raw data from perturbation study. Perturbations study should
% include various perturbations on various parameters in two directions.
% This code calculates sensitivity coefficients for all perturbations and
% parameters. Then graphs the log-log plot of normalized perturbation
% versus sensitivity coefficient. At this point, the user should select a
% perturbation which reflects the most stable sensitivity coefficient.
% Input this information. Then the code will calculate the input
% uncertainties for each parameter along with the cummulative input
% uncertainty.

%EXAMPLE EXCEL SHEET
%
% ---------------------------------------------------------
% dX_ND          var_name_1      var_name_2      var_name_3
%                mm              kg              mm
%                2               0.1             1
%                40              5               25
%  0.1           800             750             775
%  0.01          750             710             725
%  0.001         705             701             704
% -0.1           600             655             630
% -0.01          645             670             660
% -0.001         670             690             680
% ---------------------------------------------------------
%
% First row of text should be headers: dX_ND (nondimensional perturbation)
% variable names, respectively. The second row of text should contain the
% units, and the third row of text should contain unperturbed output.
% First row of values should be the standard uncertainty. The second row of
% values should be the nominal input parameter value. For example:
%       Variable 1 is 40 mm when unperturbed.
% The first column of values represents the nondimensional perturbation of
% each parameter.
%       Row dX_ND=0 shows the unperturbed outputs. The cell on row
%       dX_ND=0.1 and column "var_name_1" shows the simulation output for
%       perturbing 40mm by 0.1 (10%) or changing 40mm to 44mm while keeping
%       all other variables constant.
% The row should be ordered as: dX_ND=0, positive perturbations (largest to
% smallest), negative perturbations (largest to smallest), respectively.
%
%excelfile: text string, location of excel file and file itself
%precision: integer, precision of results, ie for MATLAB, 16
%nameform: text string, base name of output files
%plot_one: binary 1 or 0, 1 plots each parameter's results plots
%   individually and saves. 0 makes them not plotted
```

```
%groupgraph: binary 1 or 0, 1 plots multiple parameters' results plots and
%   saves. 0 makes them not plotted
%numpergraph: number between 1 and # of parameters, drives number of
%   parameters per plot for groupgraph=1
%userOx: row vector, length=# parameters, start off with an initial guess
%   Ox for each parameter. When Os-Ox plots are shown, choose an Ox for
%   each parameter which represents where Os is stable. This will be used
%   to calculate uncertainty.
%table: binary 1 or 0, 1 prints out and saves tables of results. 0 does
%   not. Overall input uncertainty will always be printed.
%location: text string, location path for output files to be saved
%   (includes tables, plots, graphs)
```

C.IV.ii. Code

```
function [] =
fInputVV(excelfile,precision,nameform,plot_one,groupgraph,numpergraph,userOx,
table,location)

%% SETUP

[num,text,raw]=xlsread(excelfile);

%TEXT
Xnames=raw(1,2:end);        %PARAMETER NAMES (Nx1)  [""]
Xunits=raw(2,2:end);        %PARAMETER UNITS (DEFINING [u]) (Nx1)  [""]
N=length(Xnames);           %NUMBER OF PARAMETERS

%NUMERIC
Xnom=num(2,2:end);          %NOMINAL INPUT PARAMETER VALUES (1xN)  [u]
ux=num(1,2:end);            %NOMINAL INPUT PARAMETER VALUES (1xN)  [u]
dX_ND=num(3:end,1);         %NON-DIMENSIONAL INPUT PERTURBATION (2Mx1) [-]
Y=num(3:end,2:end);         %PERTURBED OUTPUT VALUES (2MxN) [U]
dX_ND_abs=dX_ND(dX_ND>0);   %ABS NON-DIMENSIONAL INPUT PERTURBATION (Mx1)  [-
]
M=length(dX_ND_abs);        %NUMBER OF SENSITIVITY COEFFICIENTS (1x1)  [-]

%PARAMETER NAMES VERSIONS FOR PLOTS, FIGURES, TABLES
for i=1:N
    Xnames2{i}=strcat('X',num2str(i),'(',Xunits{i},')'); %X1 (mm)
    Xnames3{i}=char(strcat({'(X'},num2str(i),{') '},Xnames{i})); %(X1)
Density
    Xnames4{i}=strcat('X',num2str(i)); %X1
end

%% FIND SENSITIVITY COEFFICIENT

Y_up=Y(1:M,:);              %Y(X+dX) positive perturbation (MxN)  [U]
Y_lo=Y(M+1:end,:);          %Y(X-dX) negative perturbation (MxN)  [U]
dX=dX_ND_abs*Xnom;          %DIMENSIONAL PERTURBATION VALUES (MxN)  [u]

for i=1:N                   %IF XNOM IS 0
    if Xnom(i)==0
        dX(:,i)=dX_ND_abs';     %USE ND DX AS DX
    end
end

dY_dX=abs(Y_up-Y_lo)./dX/2; %SENSITIVITY COEFFICIENT (MxN)  [U/u]

if table==1
    %PRINT TO TEXT FILE
    dY_dX_table=array2table([dX_ND_abs dY_dX],'VariableNames',['dX_ND_abs'
Xnames])
    filename=strcat(location,nameform,'_Sens_Coef_table.txt');
    writetable(dY_dX_table, filename);
end

%% ORDER OF NORMALIZED SENSITIVITY COEFFICIENT
%ORDERS OF TEN FOR EFFECTIVE LOG-LOG PLOTS
```

```
Uinput_i=dY_dX.*ux;         %INPUT UNCERTAINTY (MxN)  [U]
dY_dX_N=dY_dX.*Xnom;        %NORMALIZED SENSITIVITY COEFFICIENT (MxN)  [U]
for i=1:N                   %IF XNOM IS 0
    if Xnom(i)==0
        dY_dX_N(:,i)=dY_dX(:,i);    %USE ITS D REGULAR VALUE
    end
end

for i=1:M                   %STEP THROUGH ROWS/PERTURBATIONS
    Ox(i)=log10(dX_ND_abs(i));  %ORDER OF ND DX OR PERTURBATION (1xM)
    for j=1:N                       %STEP THROUGH COLUMNS/PARAMETERS
        PS=10^(-(precision+1));
    %OS - LOG OF ORDER OF NORMALIZED SENSITIVITY COEFFICIENT
        %Os=log(X*dS/dX)=log(dS)+log(X/dX)=log(PS)-log(dX/X)=log(PS)-Ox
        Os(i,j)=log10(PS)-Ox(i);   %SET ORDER OF SC TO PRECISION (MxN)
        if dY_dX_N(i,j) ~= 0            %IF SENS COEF ISN'T ZERO
            Os(i,j)=log10(abs(dY_dX_N(i,j)));   %FIND SC ORDER (MxN)
        end
    %OU - LOG OF ORDER OF INPUT UNCERTAINTY
        if Uinput_i(i,j) ~= 0           %IF U_IN ISN'T ZERO
            Ou(i,j)=log10(abs(Uinput_i(i,j)));              %FIND ORDER(MxN)
        else                            %OTHERWISE
            Ou(i,j)=log10(ux(j))+log10(PS)-log10(dX(i,j));  %FIND ORDER WITH
PRECISION LIMIT (MxN)
        end
    end
end

%OS - TABLE
    Ox_Os=[Ox' Os];             %COMBINE INTO ONE MATRIX(MxN+1)
    if table==1
        %PRINT TO TEXT FILE
        Ovars(1)={'Ox'};
        Ovars(2:N+1)=Xnames(1:N);
        O_table=array2table(Ox_Os,'VariableNames',Ovars)
        filename=strcat(location,nameform,'_O_table.txt');
        writetable(O_table, filename);

        %PRINT TO EXCEL FILE
        colNames=Xnames3;
        num=Ox_Os;
        num=num2cell(num);
        Os_xlsx=[[{' '} colNames];num];
        filename=strcat(location,nameform,'_O_table.xlsx');
        writecell(Os_xlsx,filename);
    end


%OU - TABLE
    Ox_Ou=[Ox' Ou];             %COMBINE INTO ONE MATRIX (MxN+1)
    if table==1
        %PRINT TO TEXT FILE
        Ovars(1)={'Ox'};
        Ovars(2:N+1)=Xnames(1:N);
        Osc_table=array2table(Ox_Ou,'VariableNames',Ovars);
        filename=strcat(location,nameform,'_Ou_table.txt');
        writetable(Osc_table, filename);
```

```
    end

%% PLOT EACH PARAMETER'S OS-OX GRAPH

if plot_one==1
    for j=1:length(Xnames)
        figure
        plot(Ox,Os(:,j));
        hold on
        xlabel('Ox')
        ylabel('Os')
        title([Xnames3{j}, ', Parameter',num2str(j)])
        hold off
        filename=strcat(location,nameform,'_Os_X',num2str(j),'.png');
        saveas(gcf,filename);
    end
end

%% PLOT IN GROUPS
% PLOT OX AND OU FOR ALL PARAMETERS, MULTIPLE IN ONE GRAPH

if groupgraph==1
    maxOs=ceil(max(max(Os)));%ceil(min(max(max(Os)),5));
    minOs=floor(min(min(Os)));%floor(max(min(min(Os)),-5));
    maxOu=ceil(max(max(Ou)));%ceil(min(max(max(Ou)),5));
    minOu=floor(min(min(Ou)));%floor(max(min(min(Ou)),-5));
    endi=ceil(N/numpergraph);
    for i=1:endi
        startnum=(i-1)*numpergraph+1;
        endj=numpergraph;
        if i==endi
            endj=mod(N,numpergraph);
            if endj==0
                endj=numpergraph;
            end
        end
    %OS - LOG OF ORDER OF NORMALIZED SENSITIVITY COEFFICIENT
        figure
        hold on
        for j=1:endj
            plot(Ox,Os(:,startnum+j-1))
            ylim([minOs,maxOs])
            xlabel('Ox')
            ylabel('Os')
        end
        legend(Xnames3{startnum:startnum+endj-1},'Location','southwest')
        hold off
        filename=strcat(location,nameform,'_Os_group',num2str(i),'.png');
        saveas(gcf,filename);
    %OU - LOG OF ORDER OF INPUT UNCERTAINTY
        figure
        hold on
        for j=1:endj
            plot(Ox,Ou(:,startnum+j-1))
            ylim([minOu,maxOu])
            xlabel('Ox')
            ylabel('Ou')
```

```
        end
        legend(Xnames3{startnum:startnum+endj-1},'Location','southwest')
        hold off
        filename=strcat(location,nameform,'_Ou_group',num2str(i),'.png');
        saveas(gcf,filename);
    end
end


%% USER CHOOSE OX VALUE FOR STABILITY

%FIND VALUES FOR EACH PARAMETER
for i=1:N
    [~,OxIND]=min(abs(userOx(i)-Ox));
    Uinput_cho(i)=Uinput_i(OxIND,i);
    dX_cho(i)=dX(OxIND,i);
    dX_ND_cho(i)=dX_ND_abs(OxIND);
    dY_dX_cho(i)=dY_dX(OxIND,i);
end

%% COMPREHENSIVE OUTPUT OF CHOSEN

    %BUILD TABLE
    rowNames={'Xnom','ux','dX_dN', 'dX', 'SC','u_input'};
    colNames=Xnames;
    Uinput_cho_matrix=[Xnom; ux; dX_ND_cho; dX_cho; dY_dX_cho; Uinput_cho];

    %PRINT TO MATLAB

Uinput_cho_table=array2table(Uinput_cho_matrix,'RowNames',rowNames,'VariableN
ames',colNames)

if table==1
    %PRINT TO TEXT FILE
    filename=strcat(location,nameform,'_Uinput_cho.txt');
    writetable(Uinput_cho_table, filename);
end

%% CREATE DISPLAY TABLE

    colNames={'Xnom','Ux','dSdX', 'Uinput'};
    rowNames=Xnames2;

    %PRINT TO MATLAB
    Uinput_disp_table=array2table([Xnom', ux', dY_dX_cho',
Uinput_cho'],'RowNames',rowNames,'VariableNames',colNames)

if table==1

    %PRINT TO TEXT FILE
    filename=strcat(location,nameform,'_Uinput_disp.txt');
    writetable(Uinput_disp_table, filename);

    %BUILD MATRIX
    colNames={'Xnom','Ux','dS/dX', 'Uinput'};
    Uinput_disp_num=[Xnom', ux', dY_dX_cho', Uinput_cho'];
    Uinput_disp_num=num2cell(Uinput_disp_num);
```

```
    colUnits={'[u]','[u]','[U]/[u]','[U]'};
    Uinput_disp_colNames=[colNames;colUnits];
    Uinput_disp_partial=[Uinput_disp_colNames;Uinput_disp_num];
    Uinput_disp_rowNames=[[{' '};{' '};Xnames4'] [{'[u]'};{' '};Xunits']];
    Uinput_disp_indiv=[Uinput_disp_rowNames Uinput_disp_partial];
    for i=size(Uinput_disp_indiv,1)+1
        Uinput_xlsx(i,1)={' '};
    end
    for i=size(Uinput_disp_indiv,2)+1
        Uinput_xlsx(1,i)={' '};
    end

    %PRINT TO EXCEL FILE
    Uinput_xlsx(2:end,2:end)=Uinput_disp_indiv;
    filename=strcat(location,nameform,'_Uinput_disp.xlsx');
    writecell(Uinput_xlsx,filename)
end

Uinput_tot=sqrt(Uinput_cho*Uinput_cho')

end
```