Title title title title title title title title ‽250 characters max

R.M.P. Morillo ^{1*}, Tim David ^{2}, Pierre A. Gremaud ^{1}

1 Department of Mathematics, North Carolina State University, Raleigh, North Carolina, United States of America
2 Department of Mechanical Engineering, University of Canterbury, New Zealand

* rmmorill@ncsu.edu

Todo list

‽Notes that came in the template

Please do not include colors or graphics in the text.

The manuscript LaTeX source should be contained within a single file (do not use `\input`, `\externaldocument`, or similar commands).

DO NOT INCLUDE GRAPHICS IN YOUR MANUSCRIPT (more details commented out)

Please use ”sentence case” for title and headings (capitalize only the first word in a title (or heading), the first word in a subtitle (or subheading), and any proper nouns).

Use ”Eq” instead of ”Equation” for equation citations.

For figure citations, please use ”Fig” instead of ”Figure”.

Place figure captions after the first paragraph in which they are cited. (example figure commented out)

Place tables after the first paragraph in which they are cited. (example table commented out)

PLOS does not support heading levels beyond the 3rd (no 4th level headings).

Abstract

write abstract

‽max 300 words
‽no citations, minimal abbreviations if any
The abstract. Summary of what we did, what we got, etc.

Introduction

Add in stuff very similar to the previous introduction: talk about the error propagation V model discrepancy curve, (briefly) about what our model is, why we want to “improve” the model, and define notation for a generic model.

introduce following abbreviations: QoI, GSA

Let us now set up some notation. We denote $\vec{\theta} = [\theta_1, \theta_2, \ldots, \theta_N]$ as the parameters that are input into the model. For each of our parameters θ_j we have a corresponding nominal value $\theta_j^\star$ that was obtained through previous studies. If we wish to have all parameters set to their nominal values we use $\vec{\theta}^\star = [\theta_1^\star, \ldots, \theta_N^\star]$. These parameters are uncertain and therefore also have an associated distribution. When we drawing samples from the distributions we use the superscript to indicate each sample, i.e. θ_j^m would be the mth sample for parameter j and $\vec{\theta}^m$ would be the vector containing the mth sample of each parameter.

For our QoI we denote

$$q = g(\vec{\theta}). \tag{1}$$

The function g involves taking the parameter values in $\vec{\theta}$, solving the ordinary differential equations within the model using said parameter values, and then using the resulting time series data of the state variables within the model to compute the desired QoI.

Our first use of GSA is a linear screening step. While this is a very crude form of GSA, we use it to identify and exclude the non-influential parameters. As a result, when we move on to a more computationally expensive GSA step our number of parameters will be reduced to a more manageable level. The linear screening step starts by repeatedly sampling values for the parameters and computing the QoI produced by each sample. Then using these values a linear approximation to the model is constructed:

$$q = g(\vec{\theta}) = g(\theta_1, \ldots, \theta_N) \approx \beta_0 + \sum_{j=1}^{N} \beta_j \theta_j. \tag{2}$$

Once we have values for β_0 through β_N that provide an accurate enough approximation we define a linear importance measure L_j for each parameter:

$$L_j = \frac{|\beta_j}{\sum_{j=1}^{N} |\beta_j|}. \tag{3}$$

By construction the L_j’s are non-negative and sum to 1; therefore the closer L_j is to 1 the more influential θ_j. The calculated L_j’s can then be used to determine which parameters can be removed at this step. We decided to retain only the parameters that contribute to the top 95% of the total linear importance. The parameters that are not retained, i.e the non-influential ones, are set to their nominal values.

We create two new vectors of parameters: $\hat{\vec{\theta}}$ and $\bar{\vec{\theta}}$. The vectors $\vec{\theta}$ and $\hat{\vec{\theta}}$ both have dimension N but in $\hat{\vec{\theta}}$ the non-influential parameters, as determined by the linear screening, as fixed at their nominal values. The vector $\bar{\vec{\theta}}$ has a dimension less than N as it only contains the non-fixed entires of $\hat{\vec{\theta}}$. These two new vectors of parameters allow us to construct the following approximation:

$$q = g(\vec{\theta}) \approx g(\hat{\vec{\theta}}) = h(\bar{\vec{\theta}}). \tag{4}$$

With the number of non-fixed parameters reduced to a reasonable number we move on to surrogate modeling and Sobol analysis.

To reduce computational costs all of our remaining GSA work is not done on the original model but rather upon a high-fidelity surrogate model. Specifically we create a Polynomial Chaos (PC) approximation of $h(\bar{\vec{\theta}})$:

$$h(\bar{\vec{\theta}}) \approx H(\bar{\vec{\theta}}) = \sum_{i=0}^{N} c_i \psi_i(\bar{\vec{\theta}}). \tag{5}$$

In this the ψ_i terms are pre-determined multivariate polynomials of the parameters within $\bar{\vec{\theta}}$. The c_i terms are scalar coefficients that we solve for using least squares minimization. Once we are satisfied with the fidelity of the PC approximation, we turn to preforming Sobol analysis upon $H(\bar{\vec{\theta}})$. Due to use using a PC approximation as our surrogate model, preforming the Sobol analysis requires almost no additional computational cost.

Find better place for Sobol explanation, doesn't flow right now. Appendix? Remove?

Sobol Analysis is based upon the ANOVA decomposition; any function $Y = F(\vec{X})$ that takes in a vector of uncertain parameters of dimension d can be broken down as follows:

$$Y = f_0 + \sum_{i=1}^{d} f_i(X_i) + \sum_{i<j}^{d} f_{i,j}(X_i, X_j) + \ldots + f_{1,2,\ldots,d}(\vec{X}). \tag{6}$$

In this decomposition the subscripts on the functions f determine which of the uncertain parameters it is a function of: f_0 is a constant, f_i is a function of just X_i, $f_{i,j}$ is a function of just X_i and X_j, and so on. As long as these functions are all orthogonal and square-integrable we obtain the following:

$$\begin{aligned} Var(Y) &= \int F(\vec{x}) d\vec{X} - f_0^2 && (7) \\ &= \sum_{i=1}^{d} \int f_i(X_i) dX_i + \sum_{i<j}^{d} \int f_{i,j}(X_i, X_j) dX_i dX_j + \ldots + \int f_{1,2,\ldots,d}(\vec{X}) d\vec{X} && (8) \\ &= \sum_{i=1}^{d} V_i + \sum_{i<j}^{d} V_{i,j} + \ldots + V_{1,2,\ldots,d}, && (9) \end{aligned}$$

enabling us to decompose the variance of the function output into terms associated with each uncertain parameter or interactions between multiple parameters. From this decomposition two indexes can be calculated, the first-order Sobol index and the total

Sobol index:

$$S_i = \frac{V_i}{Var(Y)} \tag{10}$$

$$S_{Ti} = S_i + \sum_{j \neq i} S_{i,j} + \ldots + S_{1,2,\ldots,d}. \tag{11}$$

The first-order Sobol index S_i can be interpreted as how much of the output variance comes from varying just the uncertain parameter X_i without taking into account any interactions it may have with other parameters being varied; the total Sobol index S_{Ti} takes into account not just the effect of varying X_i but also all of the interactions between X_i and other parameters.

Once the Sobol analysis is done our number of parameters should have decreased noticeably. The linear screening step will remove a large number of parameters that, while important within the model, have little to no effect upon the specific QoI we are investigating. Then the Sobol analysis will tell us of the parameters that are influential upon the QoI's behavior which ones have the greatest influence. Having reduced the number of parameters to a more reasonable level we can move on to optimizing the nominal values of those highly influential parameters. The specific cost function and optimization approach taken will depend upon the specific problem being examined.

‽Majority of citations here

‽Note any relevant controversies or disagreements in the field?

Conclude with a brief statement of the overall aim of the work and a comment about whether that aim was achieved

Materials and methods

Introduce specific information about the model

Abbreviations to introduce: HbO, HbR, CBF

The experimental data we used was obtained by the Berwick Group in Sheffield, UK. This data contains time series for HbO and HbR obtained from murine experiments consisting of a short duration sensory stimulation while the animal breathed oxygen (100%). In each experiment there were 30 successive 25 second long trials wherein the first 5 seconds were used for sensor calibration followed by 2 seconds of whisker stimulation. For all stimulation experiments, the whiskers were mechanically deflected using aplastic T-bar attached to a stepper motor under computer control. Whiskers were deflected ~ 1 cm in the rostro-caudal direction at 5Hz. 5 different animals were used across all of the experiments. The experiment was run 19 times, each one consisting of 30 trials, and was recorded by 4 different sensors. This provides us with 2280 sets of HbO and HbR time series to analyze. A visualization of this data can be seen in Fig.

Add Figure of Raw data

. One thing to note about this data is that around the -2.5 second mark there is a distinct decrease in the standard deviation. This is a result of how the sensors use the first 5 seconds to calibrate. The sensors do not report the raw values of HbO and HbR but rather the percentile change in HbO or HbR from a baseline value. This baseline value is set as the average value over the first 5 seconds. As a result if the values of HbO or HbR has an increasing or decreasing trend during the first 5 seconds, the value at -2.5 seconds is very likely to be same as the baseline value. When looking at all 2280 sets of data this leads to a the standard deviation decreasing around -2.5 seconds.

Comparing the experimental values of HbO to the numerical values is extremely simple as HbO is one of the state variables within the numerical model. By simply interpolating the numerical results to match the times at which the measurements were taken experimentally (which were identical in all of the trials) and normalizing by the baseline value, the numerical and experimental results can be compared. Dealing with HbR is slightly more complicated as it is not a state variable within the numerical model. HbO has to be computed at post-processing using the numerical state variables HbR and CBF. Once this is done HbO is again interpolated to be evaluated at the time time measurements as the experimental results and normalized by its baseline value. When comparing the numerical results to the experimental we are dealing with four vectors; the time series produced by the numerical model HbR_{mod} and HbO_{mod} as well as the experimental time series HbR_{exp} and HbO_{exp}. In order to compare these we use the data misfit as our QoI,

$$q = g(\vec{\theta}) = \sum_{n} \left\{ \left(\mathrm{HbR}_{mod}(t_n) - \mathrm{HbR}_{exp}(t_n)\right)^2 + \left(\mathrm{HbO}_{mod}(t_n) - \mathrm{HbO}_{exp}(t_n)\right)^2 \right\}, \quad (12)$$

where the t values are the times at which the sensors recorded data.

For our specific problem the process of sampling our parameters in oder to build a linear approximation wasn't trivial. While we could easily draw samples for each of our parameters, we used $\theta_i \sim U(\theta_i - 0.1|\theta_i^\star|, \theta_i + 0.1|\theta_i^\star|)$ for all of our parameters, the resulting parameter vector $\vec{\theta}^j$ was not guaranteed to produce feasible model behavior. Quite often the sampled parameter values would cause state variables to blow up or behave in ways that, while mathematically reasonable, were physiologically impossible. We used the following process to iteratively update the joint distribution for the parameters until it was producing samples that lead to feasible solution at least 85% of the time:

- Sample the current distribution and evaluate the model using each sample
- Reject samples that led to infeasible solutions
- Update the joint distribution using the remaining samples through kernel density estimation.

add specific conditions for what was/wasn't feasible?

We used the MATLAB command `fitdist` for kernel density estimation. This takes the accepted parameter values and fits them to a kernel distribution through the use of maximum likelihood estimation; the kernel distribution is constrained to have the same bounds as the original distribution, i.e. $[0.9\theta_i^\star, 1.1\theta_i^\star]$. If the accepted parameters ever indicated that some of the parameters were correlated copulas were used instead via the MATLAB command `copulafit`.

When preforming the screening step on our model using the QoI given in Eq. (12) we were able to reduce our number of parameters by over 25%, from 234 to 171. When preforming Sobol Analysis upon the PC surrogate model produced with these 171 parameters we found that the values of the total Sobol indexes decreased rapidly, which indicates that the majority of the variance in the QoI comes from only a few of the parameters. After examining the rate of decay in the total Sobol indexes we decided to place the cutoff for important parameters after the 15th parameter. A visualization of the parameter importances in the linear screening and in the Sobol analysis can be seen in Fig.

Add Figure of importances

When attempting to optimize the parameter values using the information from the GSA work, we were presented with several challenges. First of all, we could not reliably produce approximations of the gradient of our QoI. This was a result of the complexity of the model as well as a result of its stiffness. We remedied this issue by using a gradient-free optimization method: the simplex method via the MATLAB command `fminsearch`. We note that using this method there is no guarantee of convergence, even to a local minimum. The second issue we ran into was that the optimization process was not bounded by the support of the joint distribution used in the GSA work. Thus, without us stepping in, the optimization routine could produce value for the important parameters outside of the region that we had analyzed. We remedied this by introducing a penalty term $P(\vec{\theta})$,

$$P(\vec{\theta}) = \begin{cases} 2g(\vec{\theta}^\star) & \text{if any } \theta_j \text{ lies outside its appropriate support} \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

and using the cost function $G(\vec{\theta}) = P(\vec{\theta}) + g(\vec{\theta})$ in our optimization routine. By using $\vec{\theta}^\star$ as our initial guess we ensure that the $\vec{\theta}$ output by `fminsearch` will lie within the appropriate support. And finally, there is no guarantee that the model will produce a feasible solution for *every* sampled parameter value from the joint distribution.

The final step of optimization we did was to compare the parameter values obtained from running `fmincon` to the parameter samples we created during the GSA work. Because all of the parameters were being varied during the GSA work, there is the possibility that one of the random samples we used to build the distributions would result in a lower cost than the result obtained by `fmincon`, where only some of the parameter values were being altered.

Due to the complexity of our model we decide to run the entire process, starting with building parameter distributions and ending with optimizing parameter values, iteratively. After each run we adjust the nominal values of the parameters based upon the optimization results, and therefore are in a completely different point in the parameter space. As a result a parameter that was previously important may no longer be, or vice versa. This also allows each individual parameter to be adjusted further - because the nominal value is changed at each iteration the initial distribution of $U(\theta_i^\star - 0.1|\theta_i^\star|, \theta_i^\star + 0.1|\theta_i^\star|)$ will change as well. For our specific problem we found that two iterations were of the GSA-optimization process was sufficient. The decrease in G produced by the second iteration was significantly smaller than the decrease in G produced by the first iteration; smaller enough that we did not expect a third iteration to provide noticeable improvement.

Results

Table.

insert table of optimized parameter values

lists all of the parameters that were changed throughout the two iterations of the GSA-optimization process. Only 25 out of the 234 parameter values were altered. Note that the changes in nominal values to are not drastic, with the maximal absolute change being ≈ 15.7% and the mean absolute change being ≈ 7.3%. Despite the moderate changes the GSA-optimization process provided significant improvement in model behavior. Fig.

insert figure showing results of optimization

shows the mean and standard deviation of the data (solid black line and shaded green region), the model output using the original nominal values (dash-dotted blue line), and the model output using the optimized parameter values (dotted magenta line). When looking at HbO we can see a dramatic improvement in model behavior with the data misfit decreasing from 1.87×10^{-2} to 9.3×10^{-4}. While the optimization didn't improve HbR's performance as drastically, there is still significant reduction in data misfit from 5.7×10^{-3} using the original values to 3.2×10^{-3} using the optimized parameter values. Overall this corresponds to an 83% reduction in error.

Possibly talk about convergence of parameter distributions? Possibly talk about intermediate optimization results?

Discussion

Same as before - talk about what this helps us know about the model and how doing OAT optimization would have moved some parameters in the wrong way

Conclusion

Its a conclusion. Wrap everything up, re state key things obtained

Supporting information

S1 Fig. Bold the title sentence. Add descriptive text after the title of the item (optional).

S2 Fig. Lorem ipsum. Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque.

S1 File. Lorem ipsum. Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque.

S1 Video. Lorem ipsum. Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque.

S1 Appendix. Lorem ipsum. Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque.

S1 Table. Lorem ipsum. Maecenas convallis mauris sit amet sem ultrices gravida. Etiam eget sapien nibh. Sed ac ipsum eget enim egestas ullamcorper nec euismod ligula. Curabitur fringilla pulvinar lectus consectetur pellentesque.

Acknowledgments

Author Contributions

‽Not all needed, these are all the categories listed on the website
Conceptualization
Data Curation
Formal Analysis

Funding Acquisition
Investigation
Methodology
Project Administration
Resources
Software
Supervision
Validation
Visualization
Writing – Original Draft Preparation
Writing – Review & Editing

References

1. Conant GC, Wolfe KH. Turning a hobby into a job: how duplicated genes find new functions. Nat Rev Genet. 2008 Dec;9(12):938–950.

2. Ohno S. Evolution by gene duplication. London: George Alien & Unwin Ltd. Berlin, Heidelberg and New York: Springer-Verlag.; 1970.

3. Magwire MM, Bayer F, Webster CL, Cao C, Jiggins FM. Successive increases in the resistance of Drosophila to viral infection through a transposon insertion followed by a Duplication. PLoS Genet. 2011 Oct;7(10):e1002337.