Response by author to interactive comment by RC1 on "Fast sensitivity analysis methods for computationally expensive models with multidimensional output" by Edmund Ryan et al.

Reviewer's comment #1:
The authors present several estimators for the Sobol' indices. They may consider [1] where the "most efficient formulas available today..." for Sobol' index estimation is described.

Author's response:
As stated before, I am happy to include [1] in the revised manuscript. However, I am having problems accessing it. I have asked for my university library to purchase the 'Handbook of Uncertainty Quantification' book, but at the time of resubmission it has not yet arrived.

Changes made in manuscript:
I am unable to include this reference in the revised manuscript for the reasons listed above.

Reviewer's comment #2:
What is being plotted in figures 3 and 4? Based on the magnitudes, I assume this is the numerator of the Sobol' index, i.e. Var(E[f(X)|Xi]). Did you have to rescale anything to compare results from the different methods?

Author's response:
These are the sensitivity indices (SIs) calculated using the five different methods (a-e in both figures). Using the variance based (not 'variational' as I incorrectly stated before) methods (Sobol, eFAST and GAM) the SIs are computed using Si = Var(E[f(X)|Xi]) / Var(f(X)), so no rescaling needed.

Changes made in manuscript:
I have made changes to the parts of the methods section that describe these different sensitivity methods: these changes include (i) adding more detail and (ii) adding more equations. These changes can be seen on pages 11, 12, 15, 16. I have amended the captions for figures 3 and 4 to make it clear what formulae and by explicitly referring to the equations used to compute the sensitivity indices.

Reviewer's comment #3:
How did you reconstruct the spatially distributed sensitivity indices in figures 3 and 4 from the PCA? Based on comparing the methods you clearly did it correctly; it would be nice to be a bit more explicit about this.

Author's response:

Yes, this is a good point. In non-statistical terms, using PCA for this purpose is a little bit like zipping a file to make it smaller in size and then unzipping it when you want to use the file again. Here, we go from an output dimension of 2000 (e.g. 2000 modelled ozone values at different latitudes / longitudes) to a dimension of say 5 (the first 5 principal components). When we compute the SIs using the eFAST method we need to run the emulator 5000 times for each of the 2000 model outputs. Using PCA, we run the emulator only 5000 times for each of the 5 transformed outputs. For each of the 5000 emulator runs, we can reconstruct the map of 2000 model outputs from the 5 transformed outputs. It's a little long to explain here, but in the methods I have added further detail to make this clear.

Changes to be made in manuscript:
In the PCA part of the methods section (page 16/17) I have included extra detail about how the 2000 model outputs are recovered from the first Npc PCs where Npc=5 for one of the chemistry models and Npc=40 for the other one.

Reviewer's comment #4:
As mentioned in [2], it is frequently useful to have a scalar sensitivity instead of a spatially distributed one as in figures 3 and 4. How can your results be "averaged" in space to provide one scalar sensitivity for each parameter?

Author's response:
In a paper currently in preparation, Oliver Wild (my collaborator and line manager) will be presenting results to a sensitivity analysis where one of the outputs is global methane lifetime (i.e. the methane lifetimes presented here but just as one number). The sensitivity analysis in that paper has the same inputs and the same training runs as this study, so to avoid repetition of results I have not shown these sensitivity analysis results in this paper. I have however referred to his paper in the revised manuscript.

Changes to be made in manuscript:
In the revised manuscript, I have added a new subsection (4.3) to the discussion section which talks in detail about Wild et al. (in prep.). You can find this on page 24 of the revised manuscript.

Reviewer's comment #5:
Two approaches are considered for constructing a meta-model for a spatial dependent output. One is based upon constructing a meta-model for each point in space, and the other is based upon constructing a meta model for each PCA mode. Would it be possible to construct a meta-model which is learned to predict all points in space simultaneously? In this case, it would be a function from Rn to Rm where n is the number of parameters and m is the number of model grid points. I could imagine training a neural network to learn this function. How would this approach compare with the methods of this article?

Author's response:

Yes, this is possible and a neural network approach would work. The problem with neural networks is that they need a lot of training data, of the order of 1000s. The main reason for using a Gaussian process emulator is that it even with not many model runs to train it (80 in this study but in general < 200 for a high number of inputs), it has been shown in lots of settings how it can robustly and accurately approximate the input-output relationship of the computationally expensive model.

Changes to be made in manuscript:
In the revised manuscript, I have added new text to page 5 of the revised manuscript to address this point.

References
[1] Clementine Prieur and Stefano Tarantola. Variance-based sensitivity analysis: Theory and estimation algorithms. In Roger Ghanem ,David Higdon ,and Houman Owhadi, editors, Handbook of Uncertainty Quantification. Springer, 2017.
[2] Amandine Marrel, Nathalie Saint-Geours, and Matthias De Lozzo. Sensitivity analysis of spatial and/or temporal phenomena. In Roger Ghanem, David Higdon, and Houman Owhadi, editors, Handbook of Uncertainty Quantification. Springer, 2017.

Response by author to interactive comment by RC2 on "Fast sensitivity analysis methods for computationally expensive models with multidimensional output" by Edmund Ryan et al.

Reviewer's comment #1:
The PCA which brings the most novelty and potentials should be developed.

Author's response:
Thank-you for this comment. I have included further details in the discussion about the PCA method.

Changes to be made in manuscript:
New text has been added to page 23 / 24 of the revised manuscript.

Reviewer's comment #2:
page 9: the authors should precise X are inputs and Y outputs in Eq (1) as well as their dimension in the context

Author's response:
Thank-you for spotting this omission. I have included these details in the revised manuscript

Changes to be made in manuscript:
The terms included in equation 1 have now been properly defined. For clarity, I also now refer to Y_j (instead of Y) where j is a point in the output space. Please see page 9 of the revised manuscript.

Reviewer's comment #3:
page 9: "the method operates by first generating N*2p matrix ..." this first step of the method is not clear for unfamiliar readers

Author's response:
Good point. I have explained what I mean here in the revised manuscript.

Changes to be made in manuscript:
See page 10 of the revised manuscript for improved clarity of this part of the methods.

Reviewer's comment #4:
page 13: the difference the GP and the GAM methods should be clarified

Author's response:
A fair point. Text has been modified to improve clarity.

Changes to be made in manuscript:
See the bottom of page 14 for the changes to the text following this comment. In particular, I have removed all mention of 'GP' to avoid any confusion by the reader.

Reviewer's comment #5:
page 14: the authors say that the p inputs x are independent variables, the independence is an assumption required to apply the proposed method. To extent this hypothesis is realistic? The authors could have precised this assumption earlier in the text to my opinion

Author's response:
I do not understand your question. However, in response to your second sentence this is a fair point and changes to the manuscript have been made (see below).

Changes to be made in manuscript:
Please see my full response to this comment at the bottom of page 15 of the revised manuscript. In the interests of keeping things clear and simple, I have removed all reference to 'independent inptus' vs 'correlated inputs'. Our inputs are independent. Correlated inputs would be situations where the inputs consisted of a time series or varied spatially. This is not the case here. On page 9 of the revised manuscript I have also added text to make it clear that our input variables are independent.

Reviewer's comment #6:
page 15: further details should be given on how the sensitivities with the hybrid methods. Moreover, one benefit of using PCA in more general setups is to work with independent variables (PCs). Could the authors justify the threshold of 99% in the PCs selection? This seems a high value compared to some common use of PCA

Author's response:
Yes, I agree with the first point. I realised that my description of the PCA approach, I did not explain how the multi-dimensional model output is reconstructed from the PCAs, and thus how the sensitivity indices are computed. I will add this detail in. The 99% threshold seems valid for this study at least. If you look at figure 2d, you can see that even with 99% of the variance of the model output explained, the median absolute different between the emulator versus chemistry model outputs from the validation runs is okay but could be better. Setting the PCA threshold lower (e.g. 95%) may be sufficient for other studies but for this study it would resulted in a poorer performance by the emulator (i.e. it would have resulted in a lower R^2 value and a higher MAD value in figure 2d).

Changes to be made in manuscript:
On pages 16 and 17 of the revised manuscript I have added detail to the PCA part of the methods section to make sure that it is completely clear how the method works and how the sensitivity indices are computed. On page 16 I have also added two sentences to defend the threshold of 99% for this study, but also explain that lower thresholds may be okay for different modelling setups.

Reviewer's comment:
How do the authors determine the number of needed runs from the emulators in this study

Author's response:
The rule of thumb is 10 * p where p = no. of input factors.

Changes to be made in manuscript:
I have a sentence at the bottom of page 18 of the revised manuscript addressing this point.

Response by author to interactive comment by RC3 on "Fast sensitivity analysis methods for computationally expensive models with multidimensional output" by Edmund Ryan et al.

Reviewer's comment #1:
Page 4, lines 1-6: I think local sensitivity can be described better here. In this case the sensitivity is only analyzed along the nonlinear trajectories (locality). So the model is still nonlinear, the linearity is assumed for the perturbation.

Author's response:
This is a good point. To avoid confusion, I have decided to remove all reference of 'local' sensitivity analysis given the differing definitions of this for different scientific disciplines.

Changes to be made in manuscript:
At the bottom of page 3 / top of page 4, the sentences that mentioned 'local' sensitivity analysis have been removed. Then at the top of page 4, I have slightly modified how I introduce global sensitivity analysis.

Reviewer's comment #2:
Page 4, lines 7-12: I am not sure how thorough this analysis can be conceived. Depending on the community, people call sensitivity analysis either global or local implicitly. I would argue that this should be discussed in the analysis of the results as well.

Author's response:
I understand your point, however the purpose of the mentioning of local sensitivity analysis was to lead into talking about global sensitivity analysis. The purpose of this paper is not to discuss the merits of local versus global. As with my response to comment #1, I have thus decided to remove all reference of 'local' sensitivity analysis given the differing definitions of this for different scientific disciplines.

Changes to be made in manuscript:
At the bottom of page 3 / top of page 4, the sentences that mentioned 'local' sensitivity analysis have been removed. Then at the top of page 4, I have slightly modified how I introduce global sensitivity analysis.

Reviewer's comment #3:
Page 6, lines 10-15: Calibrating the emulator may not be a trivial task especially for a global search. This is an element that needs to be discussed in any case.

Author's response:

The page numbers and line numbers don't seem to match with you comment. I take your point that calibrating the emulator may not seem trivial, but with a small number of inputs and a scalar output (remember, we build a separate emulator for every point in the output space) it is actually quite simple. In the DICE-Kriging R package, maximum likelihood is used. This a common approach.

Changes to be made in manuscript:
On page 13 of the revised manuscript I have added text to address this comment.

Reviewer's comment #4:
Page 16, line 17: an R^2 of 0.97 to 0.99 is quite high. This may be an indication of a mostly linear system (necessary but not sufficient). Have the authors looked at this aspect is some detail?

Author's response:
I'm not sure I understand this comment. The R^2 value is a measure of how well the emulator outputs agree with the chemistry model outputs corresponding to the validation inputs. There are other metrics than can be used (e.g. AIC). Using R^2 on its own is not wise, which is why I also give the median absolute difference value and I also graphically show the differences (Fig. 2).

Changes to be made in manuscript:
Since I don't entirely understand the comment, I don't think there's anything to change.

Reviewer's comment:
Figure S3 could benefit from adjusting the colormap

Author's response:
Can you be more specific here? Do you mean that the colormap scale should not go as high as 65%? I purposely wanted the colormap scale to be consistent for all maps of the sensitivity indices to make it easier to compare different maps. For example, if I want to compare figure S3 with figure S4 I don't need to look at the scale to do this – I can just compare the amount of yellow versus blue in each figure. If the colormap scale went up to for example 20% in figure S3, yellow areas in figure S3 would not correspond to yellow areas in figure S4 so it's a bit of hassle to do hassle. It's far easier and less hassle if the colormap scales have the same min and max.

Changes to be made in manuscript:
I don't think there's anything to change.

Fast sensitivity analysis methods for computationally expensive models with multi-dimensional output

Edmund Ryan[1]*, Oliver Wild[1], Fiona O'Connor[2], Apostolos Voulgarakis[3], and Lindsay Lee[4]

[1]Lancaster Environment Centre, Lancaster University, Lancaster. UK

[2]UK Met Office Hadley Centre, Exeter. UK

[3]Department of Physics, Imperial College London, London. UK

[4]School of Earth and Environment, University of Leeds. UK

*Corresponding author:

Lancaster Environment Centre, Lancaster University,

Bailrigg, Lancaster.

LA1 4YQ.

UK

Tel: +44 (0)1524 594009

edmund.ryan@lancaster.ac.uk

Keywords: *global sensitivity analysis, emulator, meta-model, multi-dimensional output, principal component analysis, FAST, Sobol, partial least squares, generalized additive model, atmospheric chemical transport models, tropospheric methane lifetime.*

For submission to: *Geoscientific Model Development*

Abstract

Global sensitivity analysis (GSA) is a critical approach in identifying which inputs or parameters of a model most affect model output. This determines which inputs to include when performing model calibration or uncertainty analysis. GSA allows quantification of the sensitivity index (SI) of a particular input – the percentage of the total variability in the output attributed to the changes in that input – by averaging over the other inputs rather than fixing them at specific values. Traditional methods of computing the SIs using the Sobol and extended FAST (eFAST) methods involve running a model thousands of times, but this may not be feasible for computationally expensive earth system models. GSA methods that use a statistical emulator in place of the expensive model are popular as they require far fewer model runs. We performed an eight-input GSA, using the Sobol and eFAST methods, on two computationally expensive atmospheric chemical transport models using emulators that were trained with 80 runs of the models. We considered two methods to further reduce the computational cost of GSA: (1) a dimension reduction approach and (2) an emulator-free approach. When the output of a model is multi-dimensional, it is common practice to build a separate emulator for each dimension of the output space. Here, we used principal component analysis (PCA) to reduce the output dimension, built an emulator for each of the transformed outputs, and then computed SIs of the reconstructed output using the Sobol method. We considered the global distribution of the annual column mean lifetime of atmospheric methane, which requires ~2000 emulators without PCA, but only 5-40 emulators with PCA. We also applied an emulator-free method using a generalised additive model (GAM) to estimate the SIs using only the training runs. Compared to the emulator-only methods, the PCA/emulator and GAM methods accurately estimated the SIs of the ~2000 methane lifetime outputs but were on average 24 and 37 times faster, respectively.

Commented [RE1]: I've modified this text to improve clarity.

1. Introduction

Sensitivity analysis is a powerful tool for understanding the behaviour of a numerical model. It allows quantification of the sensitivity in the model outputs to changes in each of the model inputs. If the inputs are fixed values such as model parameters, then sensitivity analysis allows study of how the uncertainty in the model outputs can be attributed to the uncertainty in these inputs. Sensitivity analysis is important for a number of reasons: (i) to identify which parameters contribute the largest uncertainty to the model outputs; (ii) to prioritise estimation of model parameters from observational data, and (iii) to understand the potential of observations as a model constraint, and (iv) to diagnose differences in behaviour between different models;

1.1 Different approaches for sensitivity analysis

By far, the most common types of sensitivity analysis are those performed one-at-a-time (OAT) and locally. OAT sensitivity analysis involves running a model a number of times, varying each input in turn whilst fixing other inputs at their nominal values. For example, Wild (2007) showed that the tropospheric ozone budget was highly sensitive to differences in global NOx emissions from lightning. The observation-based range of 3-8 TgN/yr in the magnitude of these emissions could result in a 10% difference in predicted tropospheric ozone burden. OAT sensitivity analysis is used in a variety of research fields including environmental science (Bailis et al., 2005; Campbell et al., 2008; de Gee et al., 2008; Saltelli and Annoni, 2010), medicine (Coggan et al., 2005; Stites et al., 2007; Wu et al., 2013), economics (Ahtikoski et al., 2008) and physics (Hill et al., 2012). While the ease of implementing OAT sensitivity analysis is appealing, a major drawback of this approach is that it assumes that the model response to different inputs is independent, which in most cases is unjustified (Saltelli and Annoni, 2010) and can result in biased results (Carslaw et al., 2013).

Commented [RE2]: RC3 comments #1 and #2. The sentences that were here and which referred to local sensitivity analysis have been deleted. Different scientific disciplines have differing understandings of what 'local' means in this sense. I don't want the reader to be distracted by this, so I have removed any reference of 'local sensitivity analysis'.

Global sensitivity analysis (GSA) overcomes this OAT issue by quantifying the sensitivity of each input variable by averaging over the other inputs rather than fixing them at nominal values. However, the number of sensitivity analysis studies using this global method has been very small. Ferretti et al. (2016) found that out of around 1.75 million research articles surveyed up to 2014, only 1 in 20 of studies mentioning 'sensitivity analysis' also use or refer to 'global sensitivity analysis'. A common type of GSA is the variance based method, which operates by apportioning the variance of the model's output into different sources of variation in the inputs. More specifically, it quantifies the sensitivity of a particular input – the percentage of the total variability in the output attributed to the changes in that input – by averaging over the other inputs rather than fixing them at specific values. The Fourier Amplitude Sensitivity Test (FAST) was one of the first of these variance based methods (Cukier et al., 1973). The classical FAST method uses spectral analysis to apportion the variance, after first exploring the input space using sinusoidal functions of different frequencies for each input factor or dimension (Saltelli et al., 2012). Modified versions of FAST include the extended FAST method which improves its computational efficiency (Saltelli et al., 1999) and the random-based-design (RBD) FAST method which samples from the input space more efficiently (Tarantola et al., 2006). Another widely used GSA method is the Sobol method (Homma and Saltelli, 1996; Saltelli, 2002; Sobol, 1990), which has been found to outperform FAST (Saltelli, 2002). Most applications of the Sobol and FAST methods involve a small number of input factors. However, Mara and Tarantola (2008) carried out a 100-input sensitivity analysis using the RBD version of FAST and a modified version of the Sobol method and found that both methods gave estimates of the SIs that were close to the known analytical solutions. A downside to the Sobol method is that a large number of runs of the model typically need to be carried out. For the model used in

Commented [RE3]: RC3 comments #1 and #2. This sentence has been changed.

Mara and Tarantola (2008), 10,000 runs were required for the Sobol method but only 1000 were needed for FAST.

1.2 Emulators and meta-models

If a model is computationally expensive, carrying out 1000 simulations may not be feasible. A solution is to use a surrogate function for the model called a meta-model that maps the same set of inputs to the same set of outputs, but is computationally much faster. Thus, much less time is required to perform GSA using the meta-model than using the slow-running model. A meta-model can be any function that maps the inputs of a model to its outputs, e.g. linear or quadratic functions, splines, neural networks, etc. A neural network, for example, works well if there are discontinuities in the input-output mapping, but such a method can require thousands of runs of the computationally expensive chemistry model to train it (particularly if the output is highly multi-dimensional) which will likely be too time-consuming. Here, we use a statistical emulator because it requires far fewer training runs and it has two useful properties. First, an emulator is an interpolating function which means that at inputs of the model that are used to train the emulator, the resulting outputs of the emulator must exactly match those of the model (Iooss and Lemaître, 2015). Secondly, for inputs that the emulator is not trained at, the probability distribution of the outputs represents their uncertainty (O'Hagan, 2006). The vast majority of emulators are based on Gaussian process (GP) theory due to its attractive properties (Kennedy and O'Hagan, 2000; O'Hagan, 2006; Oakley and O'Hagan, 2004), which make GP emulators easy to implement while providing accurate representations of the computationally-expensive model (e.g. Chang et al., 2015; Gómez-Dans et al., 2016; Kennedy et al., 2008; Lee et al., 2013). A GP is a multivariate Normal distribution applied to a function rather than a set of variables. The original GP emulator in a Bayesian setting was developed by Currin et al. (1991) (for basic

Commented [RE4]: RC1, comment #5. New text to describe why a neural network approach wouldn't work for this particular modelling setup.

overview see also O'Hagan, 2006) and is mathematically equivalent to the Kriging interpolation methods used in geostatistics (E.g. Cressie, 1990; Ripley, 2005). Kriging regression has been used as an emulator method since the 1990s (Koehler and Owen, 1996; Welch et al., 1992). More recently there has been considerable interest in using this Kriging emulator approach for practical purposes such as GSA or inverse modelling (Marrel et al., 2009; Roustant et al., 2012). Examples of its application can be found in atmospheric modelling (Carslaw et al., 2013; Lee et al., 2013), medicine (Degroote et al., 2012) and electrical engineering (Pistone and Vicario, 2013).

For GSA studies involving multi-dimensional output, a traditional approach is to apply a separate GP emulator for each dimension of the output space. However, if the output consists of many thousands of points on a spatial map or time-series (Lee et al., 2013) then the need to use thousands of emulators can impose substantial computational constraints even using the FAST methods. A solution is to adopt a GSA method that does not rely on an emulator, but is based on generalized additive modelling (Mara and Tarantola, 2008; Strong et al., 2014; Strong et al., 2015b) or on a partial least squares approach (Chang et al., 2015; Sobie, 2009). A separate generalized additive model (GAM) can be built for each input against the output of the expensive model, and the sensitivity of the output to changes in each input are then computed using these individual GAM models. Partial least squares (PLS) is an extension of the more traditional multivariate linear regression where the number of samples (i.e. model runs in this context) can be small, and may even be less that the number of inputs (Sobie, 2009).

An alternative way of reducing the computational constraints is to use principal component analysis (PCA) to reduce the dimensionality of the output. This means that we require far fewer emulators to represent the outputs, reducing the GSA calculations by a large

margin, although there is some loss of detail. This PCA-emulator hybrid approach has been successfully used in radiative transfer models (Gómez-Dans et al., 2016), a very simple chemical reaction model (Saltelli et al., 2012) and general circulation models (Sexton et al., 2012). While we hypothesize that both emulator-free and PCA-based methods are suited to large-scale GSA problems (e.g. those involving more than 20 input factors), a focus of our work is to determine the accuracy of these methods for a smaller scale GSA study.

1.3 Aims of this study

Recent research comparing different GSA methods based on Gaussian Process emulators has been limited in application to relatively simple models and low-dimensional output (Mara and Tarantola, 2008). Using two computationally expensive models of global atmospheric chemistry and transport – namely FRSGC/UCI and GISS – we compare the accuracy and efficiency of global sensitivity analysis using emulators and emulator-free methods, and we investigate the benefits of using PCA to reduce the number of emulators needed. We compare and contrast a number of ways of computing the first order sensitivity indices for the expensive atmospheric models: (i) the Sobol method using an emulator; (ii) the extended FAST method using an emulator; (iii) generalised additive modelling; (iv) a partial least squares approach; (v) an emulator-PCA hybrid approach. Hereafter, we refer to (i) and (ii) as emulator-based GSA methods and (iii) and (iv) as emulator-free GSA methods.

2. Materials and methods

2.1 Atmospheric chemistry models

Global atmospheric chemistry and transport models simulate the composition of trace gases in the atmosphere (e.g. O_3, CH_4, CO, SO_x) at a given spatial resolution (latitude × longitude ×

altitude). The evolution in atmospheric composition over time is controlled by a range of different dynamical and chemical processes, our understanding of which remains incomplete. Trace gases are emitted from anthropogenic sources (e.g., NO from traffic and industry) and from natural sources (e.g. isoprene from vegetation, NO from lightning), they may undergo chemical transformation (e.g., formation of O_3) and transport (e.g., convection or boundary layer mixing), and may be removed through wet or dry deposition. Global sensitivity analysis is needed to understand the sensitivity of our simulations of atmospheric composition and its evolution to assumptions about these governing processes.

In this study, we performed global sensitivity analysis (GSA) on two such atmospheric models. We used the Frontier Research System for Global Change version of the University of California, Irvine chemistry transport model, the FRSGC/UCI CTM (Wild et al., 2004; Wild and Prather, 2000), and the Goddard Institute for Space Studies general circulation model, the GISS GCM (Schmidt et al., 2014; Shindell et al., 2006). We used results from 104 model runs carried out with both of these models from a comparative GSA study (Wild et al., in prep.). This involved varying eight inputs or parameters over specified ranges using a maximin Latin hypercube design: global surface NOx emissions (30-50 TgN/year), global lightning NOx emissions (2-8 TgN/year), global isoprene emissions (200-800 TgC/year), dry deposition rates (model value ± 80%), wet deposition rates (model value ± 80%), humidity (model value ± 50%), cloud optical depth (model value × 0.1–10) and boundary layer mixing (model value × 0.01–100). For this study, we focus on a single model output, the global distribution of annual tropospheric column of mean lifetime of methane (CH_4). The CH_4 lifetime is an important indicator of the amount of highly reactive hydroxyl radical in the troposphere (Voulgarakis et al., 2013), and we choose this output because of its contrasting behaviour in the two models. The

native spatial resolution of the models is 2.8°×2.8° for FRSGC and 2.5°×2.0° for GISS, but we combine neighbouring grid points so that both models have a comparable resolution of 5-6°, giving a total of 2048 grid points for FRSGC/UCI and 2160 grid points for GISS.

2.2 Global sensitivity analysis using the Sobol and extended FAST methods

For brevity and generality, we hereafter refer to each of the atmospheric chemical transport models as a simulator. A common way of conducting global sensitivity analysis for each point in the output space of the simulator – where the output consists of for example a spatial map or a time-series – is to compute the first order sensitivity indices (SIs) using variance based decomposition; this apportions the variance in simulator output (a scalar) to different sources of variation in the different model inputs. Assuming the input variables are independent – which they are for this study – of one another then the first-order SI, corresponding to the *i*th input variable (*i*=1, 2, …, p) and the *j*th point in the output space, is given by:

$$S_{i,j} = \frac{Var\left[E\left(Y_j|X_i\right)\right]}{Var(Y_j)} \qquad (1)$$

where X_i is the *i*th column of the $n \times p$ matrix (i.e. a matrix with n rows and p columns) which stores the n samples of p-dimensional inputs and Y_j is the *j*th column of the $n \times m$ matrix which stores the corresponding n sets of m-dimensional outputs (table 1). The notation given by $Var(\cdot)$ and $E(\cdot)$ denote the mathematical operations that compute the variance and expectation. The simplest way of computing $S_{i,j}$ is by brute force, but this is also the most computationally intensive (Saltelli et al., 2008).

2.2.1 The Sobol Method

The Sobol method, developed in the 1990s, is much faster than brute force at computing the terms in equation (1), in part because it requires fewer executions of the simulator (Homma and

Commented [RE5]: New text.

Commented [E6]: RC2, comment #5. New text to make it absolutely explicit that the eight input variables are independent.

Commented [E7]: RC2, comment #2. New text to address comment.

Saltelli, 1996; Saltelli, 2002; Saltelli et al., 2008; Sobol, 1990). The method operates by first generating a $n \times 2p$ matrix (i.e. a matrix with n rows and $2p$ columns) of random numbers from a space filling sampling design (e.g. a maximin Latin hypercube design), where n is the number of sets of inputs and p is the number of input variables. The inputs are on the normalised scale so that each element of a p-dimensional input lies between 0 and 1. Typical values for n are 1000-10,000. The matrix is split in half to form two new matrices, **A** and **B**, each of size $n \times p$. To compute the ith SI ($1 \leq i \leq p$), we define two new matrices **Ci**, and **Di**, where **Ci** is formed by taking the ith column from **A** and the remaining columns from **B**, and **Di** is formed by taking the ith column from **B** and the remaining columns from **A**. We then execute the simulator – denoted by f – at each set of inputs given by the rows of matrices **A**, **B, Ci** and **Di**. This gives vectors $\mathbf{Y_A}=f(\mathbf{A})$, $\mathbf{Y_B}=f(\mathbf{B})$, $\mathbf{Y_{Ci}}=f(\mathbf{Ci})$ and $\mathbf{Y_{Di}}=f(\mathbf{Di})$. Vectors $\mathbf{Y_A}$ and $\mathbf{Y_{Ci}}$ are then substituted into eqn (2):

$$\widehat{S_i} = \frac{\widehat{Var}\left[\hat{E}\left(Y_j|X_i\right)\right]}{\widehat{Var}(Y)} = \frac{\mathbf{Y_A} \cdot \mathbf{Y_{Ci}} - \left(\frac{1}{N}\sum_{j=1}^{N} \mathbf{Y_A^{(j)}}\right)^2}{\mathbf{Y_A} \cdot \mathbf{Y_A} - \left(\frac{1}{N}\sum_{j=1}^{N} \mathbf{Y_A^{(j)}}\right)^2} \quad (2)$$

where $\mathbf{Y_A} \cdot \mathbf{Y_{Ci}} = \left(\frac{1}{N}\sum_{j=1}^{N} \mathbf{Y_A^{(j)}} \mathbf{Y_{Ci}^{(j)}}\right)$ and $\mathbf{Y_A^{(j)}}$ and $\mathbf{Y_{Ci}^{(j)}}$ are the jth elements of $\mathbf{Y_A}$ and $\mathbf{Y_{Ci}}$. For all p input variables, the total number of simulator runs is $12 \times n \times p$. Saltelli (2002) and Tarantola et al. (2006) suggested using eight variants of equation (2), using different combinations of $\mathbf{Y_A}$, $\mathbf{Y_B}$, $\mathbf{Y_{Ci}}$ and $\mathbf{Y_{Di}}$ (Appendix A). Lilburne et al. (2009) proposed using the average of these eight SI estimates as they deemed this to be more accurate than a single estimate.

2.2.2 The Extended FAST (eFAST) Method

An alternative and even faster way of estimating the terms in equation (1) is to use the extended-FAST method, first developed by Saltelli *et al.* (1999) and widely used since (Carslaw et al., 2013; Koehler and Owen, 1996; Queipo et al., 2005; Saltelli et al., 2008; Vanuytrecht et al.,

Commented [E8]: RC2, comment #3. The text here is either new or modified to address the comment.

2014; Vu-Bac et al., 2015). A multi-dimensional Fourier transformation of the simulator f allows a variance-based decomposition that samples the input space along a curve defined by:

$$x_i(s) = G_i\big(sin(\omega_i s)\big), \tag{3}$$

where x=(x_1, …, x_p) refers to a general point in the input space that has been sampled, $s \in \mathbb{R}$ is a variable over the range (-∞,∞), G_i is the ith transformation function (Appendix A), and ω_i is the ith user-specified frequency corresponding to each input. Varying s allows a multi-dimensional exploration of the input space due to x_is being simultaneously varied. Depending on the simulator, we typically require n = 1000-10000 samples from the input space. After applying the simulator f the resulting scalar output – denoted generally by y – produces different periodic functions based on different ω_i. If the output y is sensitive to changes in the ith input factor, the periodic function of y corresponding to frequency ω_i will have a high amplitude.

More specifically, we express the model $y = f(s) = f\left(x_1(s), x_2(s), \dots, x_p(s)\right)$ as a Fourier series:

$$y = f(s) = \sum_{j=-\infty}^{\infty} A_j co\, s(js) + B_j\, sin(js) \tag{4}$$

Using a domain of frequencies given by $j \in \mathbb{Z} = \{-\infty, \dots, -1, 0, 1, \dots, \infty\}$, the Fourier coefficients A_j and B_j are defined by:

$$A_j = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(s)\cos(js)\,.\,ds$$
$$B_j = \frac{1}{2\pi}\int_{-\pi}^{\pi} f(s)\sin(js)\,.\,ds \tag{5}$$

With ω_i stated in equation (3), the variance of model output attributed to changes in the ith input variable for the jth point in the output space (numerator of equation 1) is defined as:

$$\widehat{Var}\left[\hat{E}\left(Y_j|X_i\right)\right] = \sum_{q \in \mathbb{Z}^0} A^2_{q\omega_i} + B^2_{q\omega_i} \quad (6a)$$

where $\mathbb{Z}^0$ is the set of all integers except zero. The total variance (denominator of equation 1) is:

$$\widehat{Var}(Y_j) = \sum_{k \in \mathbb{Z}^0} A^2_k + B^2_k \quad (6b)$$

Further details of extended-FAST is given in Saltelli et al. (1999). The difference between the original and the extended versions of the FAST method are given in Appendix A.

Commented [E9]: RC1 comment #2, I have included this new text. In figures 3 and 4 the captions now refer to equations 6a-b for the eFAST method.

2.3 Gaussian Process Emulators

When the simulator is expensive to run – like the atmospheric chemical transport models used here – we substitute it with an emulator which is a surrogate of the expensive simulator but much faster to run. If we are confident that the emulator is accurate, then we can compute the first order SIs from the Sobol and eFAST methods using the outputs of the emulator rather than the simulator. Mathematically, an emulator is a statistical model that mimics the input-output relationship of a simulator. As stated in the *Introduction*, an emulator is an interpolating function at model outputs it is trained at and gives a probability distribution and other outputs (O'Hagan, 2006).

An emulator is trained using N sets of p-dimensional inputs denoted by $\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_N$, and N sets of 1-dimensional outputs from the simulator given by $\mathbf{y}_1=f(\mathbf{x}_1), \mathbf{y}_2=f(\mathbf{x}_2),\ldots, \mathbf{y}_N=f(\mathbf{x}_N)$, f represents the simulator and for our study $N = 80$ (see §2.6). The most common form of an emulator is a Gaussian Process (GP) since it has attractive mathematical properties that allow an analytical derivation of the mean and variance of the emulated output (given by $\hat{f}(x)$ for a general input x). A notable exception is Goldstein and Rougier (2006) who used a non-GP emulator based on a Bayes linear approach. More formally, a GP is an extension of the

multivariate Gaussian distribution to infinitely many variables (Rasmussen, 2006). The multivariate Gaussian distribution is specified by a mean vector μ and covariance matrix $\mathbf{\Sigma}$. A GP has a mean function which is typically given by $m(x) = E(f(x))$ and covariance function given by $c(x,x') = \mathrm{cov}(f(x), f(x'))$ where x and x' are two different p-dimensional inputs. For the latter we used a Matern(5/2) function (Roustant et al., 2012), which is given by:

$$c(x,x') = s^2 + \left(1 + \sqrt{5}\left(\frac{|x-x'|}{\theta}\right) + \frac{5}{3}\left(\frac{|x-x'|}{\theta}\right)^2\right) \times exp\left(-\sqrt{5}\left(\frac{|x-x'|}{\theta}\right)\right), \quad (7)$$

where s denotes the standard deviation and θ is the vector of range parameters (sometimes called *length-scales*). These emulator parameters are normally estimated using maximum likelihood (see Bastos and O'Hagan, 2009, for details). GP emulators for uncertainty quantification were originally developed within a Bayesian framework (Currin et al., 1991; Kennedy and O'Hagan, 2000; O'Hagan, 2006; Oakley and O'Hagan, 2004).

Commented [RE10]: RC3 comment #3. New text to address this comment.

Developed around the same time, the Kriging interpolation methods used in geostatistics are mathematically equivalent to the GP methods developed by Currin et al. (E.g. Cressie, 1990; Ripley, 2005). Kriging based emulators have been used for 25 years (Koehler and Owen, 1996; Welch et al., 1992), with recent implementations including the DICE-Kriging R packages used for GSA and inverse modelling (Marrel et al., 2009; Roustant et al., 2012). Since the latter approach is computationally faster, we adopted the DICE-Kriging version of the GP emulator for this study. For the statistical theory behind both emulator versions and descriptions of related R packages, see Hankin (2005) and Roustant et al. (2012).

2.4 Emulator-free global sensitivity analysis

For GSA studies involving highly multi-dimensional output, the time to compute the SIs can be significantly reduced by employing an emulator-free GSA approach. In this study, we consider two such methods using: (i) generalised additive modelling (GAM) and (ii) a partial least squares

(PLS) regression approach. For both the GAM and PLS methods we used $n = N$ simulator runs to compute the sensitivity indices (Table 1), and for our study these were the same $N = 80$ runs that were used to train the emulators described in §2.3. In the descriptions of these two sensitivity analysis methods (§2.4.1 and §2.4.2), we thus use $\mathbf{X}=[\mathbf{X_1},\mathbf{X_2},\ldots,\mathbf{X_p}]$ and $\mathbf{Y}$ to denote the matrices that store N sets of p-dimensional inputs and m-dimensional outputs.

Commented [RE11]: This text was in section 2.4.1 but it applies to both 2.4.1 and 2.4.2 so this seems a more appropriate place in order to avoid repetition.

2.4.1 The Generalised Additive Modelling method

A generalized additive model (GAM) is a generalized linear model where the predictor variables are represented by smooth functions (Wood, 2017). The general form of a GAM is:

$$\mathbf{Y_j} = g(\mathbf{X}) + \varepsilon \tag{8a}$$

$$g(\mathbf{X}) = s(\mathbf{X_1}) + s(\mathbf{X_2}) + \cdots + s(\mathbf{X}_p) \tag{8b}$$

where: $\mathbf{X_i}$ is the ith column of input matrix $\mathbf{X}$ (i = 1, 2, …, p); $\mathbf{Y_j}$ is the jth column of output matrix $\mathbf{Y}$ (j = 1, 2, …, m) since we construct a separate GAM for each point in the output space (i.e. for each latitude/longitude point in our case); $s(.)$ is the smoothing function such as a cubic spline; and ε is a zero-mean Normally distributed error term with constant variance. If we wish to include second order terms in $g(\mathbf{X})$, we would add $s(\mathbf{X_1},\mathbf{X_2}) + s(\mathbf{X_1},\mathbf{X_3}) + \cdots + s(\mathbf{X_{p-1}},\mathbf{X_p})$ to the right-hand side of equation (8b). A GAM it is not an emulator as defined by O'Hagan (2006) because the fitted values of the GAM are not exactly equal to the outputs of the training data (Wood, S.N, personal communication). It is still a meta-model and we could use it as a surrogate of the expensive model in order to perform variance based sensitivity analysis using for example the Sobol or extended FAST method. However, we have found that the number of runs of the simulator to train it in order for it to be an accurate surrogate for the model are too many (i.e. too computationally burdensome). Instead, it is possible to obtain accurate estimates of the

Commented [E12]: RC2, comment #4. The reviewer only indicated the page number of the submitted manuscript. Hence, I was unsure of the exact line or lines being referred to. I think the comment referred to the sentence that mentioned 'GP', so I have removed this reference to GP and also amended the sentence so that it's clearer.

first order SIs by using a GAM to estimate the components of equation (1) directly (Stanfill et al., 2015; Strong et al., 2014; Strong et al., 2015b). To compute the *i*th first order SI ($1 \le i \le p$), we first recognise that taking the expectation of equation (8a) leads to $E(\mathbf{Y_j}) = g(\mathbf{X})$. The expression for $E(\mathbf{Y_j}|\mathbf{X_i})$ is thus the marginal distribution of $E(\mathbf{Y_j})$. We could fit the full model and then compute this marginal distribution following Stanfill et al. (2014). However, an easier and quicker way is to fit a GAM to the ($\mathbf{X_i}$, $\mathbf{Y_j}$) "data" where $\mathbf{X_i}$ and $\mathbf{Y_j}$ are defined above. Then, $E(\mathbf{Y_j}|\mathbf{X_i})$ consists of the fitted values of this reduced model (Strong et al., 2015b). Thus, $Var[E(\mathbf{Y_j}|\mathbf{X_i})]$ (numerator of equation 1) is determined by computing the variance of the *n* points from this fitted GAM model. In other words,

$$\widehat{Var}[\hat{E}(\mathbf{Y_j}|\mathbf{X_i})] = var\left(s(\mathbf{x_{1,i}}), s(\mathbf{x_{2,i}}), \dots, s(\mathbf{x_{n,i}})\right) \quad (9)$$

where $\mathbf{x_{k,i}}$ is the element from the *k*th row and *i*th column of matrix **X**. Finally, the denominator term of equation 1 is computed by taking the variance of the *n* samples of the outputs from the computationally expensive model that are stored in $\mathbf{Y_j}$.

Commented [RE13]: RC1 comment #2, I have included this new text. In figures 3 and 4 the captions now refer to equations 9 for the GAM method.

2.4.2 The Partial Least Squares (PLS) method

The partial least squares (PLS) method is the only one of the four GSA methods considered here that is not variance-based (Chang et al., 2015). Multivariate linear regression (MLR) is a commonly used tool to represent a set of outputs or response variables (**Y**) based on a set of inputs or predictor variables (**X**), where **X** and **Y** are matrices (table 1). MLR is only appropriate to use when the different inputs (columns in **X**) are independent and not excessive in number. In many situations, such as GSA studies, there can be a large number of input variable and/or they could be highly correlated with each other (Sobie, 2009). PLS is an extension of MLR which is able to deal with these more challenging multivariate modelling problems (Wold et al., 2001).

Commented [E14]: RC2, comment #5. The sentence that was here has been removed. I thought that this was the easier thing to do to address this comment since I didn't want the reader to be confused. The reviewer asked that the 'input variables being independent' should have been mentioned earlier in the text. It was already mentioned in the text prior to equation 1, but I have made this more explicit there. The purpose of the paper isn't to get into a discussion about independent versus correlated input variables – that debate is a whole paper in itself. The main point is that our input variables are independent (which was the only point I was making in the sentence that I deleted). Correlated input variables are (for example) a time series of some type of driving data which is definitely not the case with this paper.

The main reason for choosing PLS over other applicable regression approaches is that it has been shown to give similar estimates of the sensitivity indices to a variance based GSA approach (Chang et al., 2015). Thus, for sensitivity analysis problems when the inputs are correlated, this PLS method could be considered an alternative to the variance based GAM method which assumes that the inputs are independent. Mathematically, PLS operates by projecting $\mathbf{X}$ and $\mathbf{Y}$ into new spaces, determined by maximising the covariance between the projections of $\mathbf{X}$ and $\mathbf{Y}$ (see section S1, supplemental information for details). PLS regression is then performed where the regression coefficients represent the sensitivity indices. When $n > p$, it is standard to estimate the PLS regression coefficients using the traditional multivariate linear regression. Thus, the $p \times m$ matrix of sensitivity indices (S) can be computed using the formula:

$$S = (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{Y} \quad (10)$$

2.5 Principal Component Analysis

As an alternative approach for speeding up the sensitivity analysis calculations, we computed the SIs from the Sobol GSA method using a hybrid approach involving principal component analysis (PCA) to reduce the dimensionality of the output space, and then use separate Gaussian Process emulators for each of the transformed outputs (Gómez-Dans et al., 2016; Saltelli et al., 2012; Sexton et al., 2012). After performing the emulator runs, we then reconstruct the emulator output on the original output space, from which we can compute the sensitivity indices.

PCA transforms the outputs onto a projected space with maximal variance. Mathematically, we obtain the matrix of transformed outputs $\mathbf{Y}_{\mathbf{PC}}$ by

$$\mathbf{Y}^{(\mathbf{PC})} = \mathbf{Y}\mathbf{A}^{*} \quad (11)$$

where $\mathbf{Y}$ is the $N \times m$ matrix of training outputs from the simulator (see § 2.3), and $\mathbf{A}^{*}$ is a matrix whose columns are orthogonal to one another and whose ith column ($\mathbf{A}_{\mathbf{i}}^{*}$) is chosen such that

Commented [RE15]: RC1 comment #2, I have included this new text. In figures 3 and 4 the captions now refer to equation 10 for the PLS method.

Commented [RE16]: RC1 comment #3 and RC2 comment #6. New text.

Commented [RE17]: RC1, comment #3. New text. For equation (12) to make sense it's necessary to put this as an equation too.

$var(\mathbf{YA_i^*})$ is maximised subject to the constraint $(\mathbf{A_i^*})^T\mathbf{A_i^*} = 1$. The vector $\mathbf{A_1^*}$ is called the first principal component (PC1), and we define λ_1 to be the principle eigenvalue of $\mathbf{S}=var(\mathbf{Y})$ which is the largest variance of the outputs **Y** with respect to PC1. The second, third, fourth, etc columns of A are referred to as PC2, PC3, PC4, etc with λ_2, λ_3, λ_4, etc representing the second, third, fourth, etc largest variance of **Y**, respectively. PC1 contains the most information in the output, followed by PC2, then PC3, etc. The number of principal components required is commonly determined by plotting the following points: $(1, \lambda_1)$, $(2, \lambda_1+\lambda_2)$, $(3, \lambda_1+\lambda_2+\lambda_3)$, …, and identifying the point where the line begins to flatten out. This is equivalent to choosing a cut off when most of the variance is explained. In this study, we included the first N_{pc} principal components such that 99% of the variance is explained. The 99% threshold was also necessary for this study to ensure that the reconstructed emulator output accurately approximated the simulator output for the validation runs (Fig. 2). While we found the 99% threshold was necessary, other studies may find that a lower threshold (e.g. 95%) is sufficient.

This technique of reducing the dimension of the output space from m = ~2000 spatially varying points to the first N_{pc} principal components (e.g. N_{pc} = 5 for the FRSGC model; see § 2.6) means that the number of required emulator runs to compute the sensitivity indices from the Sobol method is reduced by a factor of m / N_{pc} (=~400 using above m and N_{pc} values). However, after having generated the N_{pc} sets of output vectors for the Sobol method ($\mathbf{Y_A^{(PC)}}$, $\mathbf{Y_B^{(PC)}}$, $\mathbf{Y_{Ci}^{(PC)}}$, $\mathbf{Y_{Di}^{(PC)}}$; see § 2.2) we need to reconstruct the m sets of output vectors which are required to compute the sensitivity indices for each of the m points in the output space. To do this we first set the elements of the *(N_{pc} +1)th*, *(N_{pc} +2)th, …, m* columns of the matrix $\mathbf{A^*}$ (eq. 11) to zero and call this new matrix $\mathbf{A^*_{sample}}$. We also form a $n \times m$ matrix $\mathbf{Y^{(PC)}_{sample}}$ whose first N_{pc} columns are vectors storing the emulator outputs corresponding to the first N_{pc} principal components,

Commented [RE18]: RC2 comment #6. New text to address this comment.

while the elements of the remaining columns are set to zero. Recall that $\mathbf{Y}_{\mathbf{sample}}^{(\mathbf{PC})}$ is different to $\mathbf{Y}^{(\mathbf{PC})}$ where the latter has N rows (80 for this study) which correspond to the number of simulator runs required to train the emulators; whereas the number of samples n (n = 10000 for this study) refer to the number of emulator runs needed to estimate the sensitivity indices. The $n \times m$ matrix $\mathbf{Y}_{\mathbf{sample}}$ of the reconstructed m-dimensional outputs is computed using

$$\mathbf{Y}_{\mathbf{sample}} = \mathbf{Y}_{\mathbf{sample}}^{(\mathbf{PC})} \left(\mathbf{A}_{\mathbf{sample}}^{*}\right)^{\boldsymbol{T}} \quad (12)$$

We use this formula to compute the $\mathbf{Y}_{\mathbf{A}}$, $\mathbf{Y}_{\mathbf{B}}$, $\mathbf{Y}_{\mathbf{Ci}}$ and $\mathbf{Y}_{\mathbf{Di}}$ vectors from § 2.2 and the resulting sensitivity indices using equation 2 from the Sobol method (§ 2.2).

Commented [RE19]: RC1 comment #3 and RC2 comment #6. New text to explain how the PCA reconstruction occurred.

2.6 Experimental setup

The sequence of tasks to complete when performing global sensitivity analysis is shown schematically in figure 1. The choice of inputs (e.g. parameters) to include in the sensitivity analysis will depend upon which have the greatest effects, based on expert knowledge of the model and field of study. Expert judgement is also needed to define the ranges of these inputs. A space-filling design such as maximin Latin hypercube sampling (see Section S2 from the supplemental material for R code) or sliced Latin hypercube sampling (Ba et al., 2015) is required in order to sample from the input space with the minimum sufficient number of model runs. We used n = 10000 for the Sobol method and n = 5000 for the eFAST method, but $n = N$ = 80 for the GAM and PLS methods. The third stage is to run the model at the set of input points specified by the space-filling sampling design.

Commented [E20]: New text.

If we are employing an emulator, the next stage is build the emulator using the training runs. The number of training runs (N) is determined by $N = 10 \times p$, where p is the number of input variables (Loeppky et al., 2009). We also need to perform runs of the computationally

Commented [RE21]: RC2 comment # 7. New text to address comment.

expensive model to validate the emulators. For this study, we ran the models with an additional set of inputs for validation. A simple comparison like this is usually sufficient, but more sophisticated diagnostics can also be carried out if needed (Bastos and O'Hagan, 2009). If employing the emulator-free approach, validation is also needed to do because we are using a statistical model to infer the SIs. Such a validation is not a central part of our results but is included in the supplemental material (Fig. S2). For the emulator-PCA hybrid approach (Figure 1), we found that the first 5 (for FRSGC) and 40 (for GISS) principal components were required to account for 99% of the variance. This means that only 5-40 emulators are required to generate a global map in place of ~2000 needed if each grid point is emulated separately, which provides a large computational saving.

The final stage is to compute the first-order SIs for all the inputs; these quantify the sensitivity of the output to changes in each input. The SIs are also known as the main effects. The eFAST, Sobol and GAM approaches can also be used to compute the total effects, defined as the sum of the sensitivities of the output to changes in input *i* on its own and interacting with other inputs. For this study, we do not consider total effects as the sum of the main effects was close to 100% in each case.

3. Results

3.1 Validation of the emulators

Since the emulators we employed are based on a scalar output, we built a separate emulator for each of the ~2000 model grid points to represent the spatial distribution of the CH_4 lifetimes. At the 24 sets of inputs set aside for emulator validation, the predicted outputs from the emulators compared extremely well with the corresponding outputs from both chemistry models (Figure 2a,b, R^2=0.9996-0.9999, median absolute difference = 0.1-0.18 years). When PCA is used to

reduce the output dimension from ~2000 to 5-40 (depending on the chemistry model), the accuracy of the predicted outputs was not as good (Figure 2c,d, R^2=0.9759-0.9991, median absolute difference = 0.94-3.44) but was still sufficient for this study.

3.2 Comparison of sensitivity indices

As expected, the two emulator-based global sensitivity analysis approaches (eFAST and Sobol) produced almost identical global maps of first order sensitivity indices (SIs, %) of CH_4 lifetime, see Figures 3 and 4. The statistics (mean, 95th percentile and 99th percentile) of the differences in SIs between the two GSA methods over all 8 inputs at 2000 output points for the FRSGC and GISS models are shown in Figure 5, M1 vs M2.

Our results show that the GAM emulator-free GSA method produces very similar estimates of the SIs to the emulator-based methods (Figures 3-4; (a) vs (c)). The 95th and 99th percentiles of differences of the emulator-based method (eFAST or Sobol) versus GAM are 5% and 9% for FRSGC, and 7% and 10% for GISS (Figure 5; M1 vs M3). For both models, the PLS non-emulator-based method produced SIs that were significantly different from those using the eFAST and Sobol methods (Figures 3-4; (a) vs (d)). For FRSGC, the mean and 95th percentile of the differences in SIs for the emulator based method versus PLS was around 21% and 31%, while for GISS the corresponding values were around 14% and 23% (Figure 5; M1 vs M4). Thus, our results indicate that the PLS method is not suitable for use as an emulator-free approach to estimating the SIs.

The global map of SIs using the emulator-PCA hybrid approach compared well to those from the emulator-only approach (Figures 3-4; (a) vs (e)). The 95th and 99th percentiles of differences between the two approaches were 6% and 10%, respectively for FRSGC (Figure 5a, M1 vs M5) and 3% and 5%, respectively for GISS (Figure 5b, M1 vs M5). These are both

higher than the corresponding values for the emulator-only methods (Figure 5, M1 vs M2; <2% and <3%, respectively). These higher values for the emulator-PCA hybrid approach is also reflected in the poorer estimates of the validation outputs using this approach versus the emulator-only approach (Figure 2). Such poorer estimates are expected because the PCA transformed outputs only explain 99% of the variance of the untransformed outputs used in the emulator-only approach.

4. Discussion

4.1 Comparison of sensitivity indices

Our results align with the consensus that the eFAST method or other modified versions of the FAST method (e.g. RBD-FAST) produce very similar SIs to the Sobol method. Mathematically, the two methods are equivalent (Saltelli et al., 2012) and when the analytical (true) values of the SIs can be computed, both methods are able to accurately estimate these values (Iooss and Lemaître, 2015; Mara and Tarantola, 2008). However, many studies have noted that the Sobol method requires more model (or emulator) runs to compute the SIs. Saltelli et al. (2012) states that $\frac{2}{k} \times 100$ (%) more model runs are required for the Sobol method compared to eFAST, where k is the number of input factors (e.g. if $k=8$, then 25% more runs are needed for Sobol). Mara and Tarantola (2008) found that the Sobol method required ~10,000 runs of their model to achieve the same level of aggregated absolute error to that of FAST, which only needed 1000 runs. This is comparable to our analysis where the Sobol method required 18,000 runs of the emulator but only 1000 runs were needed for the eFAST method.

Given recent interest in applying generalized additive models (GAMs) to perform GSA (Strong et al., 2015a, 2014; Strong et al., 2015b), only Stanfill et al. (2015) has compared how

they perform against other variance based approaches. The authors found that first order SIs estimated from the original FAST method were very close to the true values using 600 executions of the model, whereas the GAM approach only required 90-150 model runs. This is roughly consistent with our results, as we estimated the SIs using 80 runs of the chemistry models for GAM and 1000 runs of the emulator for the eFAST method.

There are a limited number of studies comparing the accuracy of the SIs of the GAM method amongst different models, as in our study. Stanfill et al. (2015) found that the GAM method was accurate at estimating SIs based on a simple model (3-4 parameters) as well as a more complex one (10 parameters). However, if more models of varying complexity and type (e.g. process versus empirical) were to apply the GAM approach, we expect that while GAM would work well for some models, but for others the resulting SIs may be substantially different to that produced using the more traditional Sobol or eFAST methods. Saltelli et al. (1993) suggests that the performance of a GSA method can be model dependent, especially when the model is linear versus non-linear, monotonic versus non-monotonic, or if transformations are applied on the output (e.g. logarithms) or not. This is particularly true for GSA methods based on correlation or regression coefficients (Saltelli et al., 1999), which might explain why the SIs calculated from the PLS method in our analysis also disagreed with those of the eFAST/Sobol methods for the FRSGC versus GISS models. Not all GSA methods are model dependent; for example the eFAST method is not (Saltelli et al., 1999).

4.2 Principal Component Analysis

For both models, using principal component analysis (PCA) to significantly reduce the number of emulators needed resulted in SIs very similar to those calculated using an emulator-only approach. For the GISS model, this was encouraging given that the spread of points and their

bias in the emulator against the model was noticeably larger than those of the FRSGC model (Figure 2c,d). If we had increased the number of principle components so that 99.9% of the variance in the output was captured rather than 99% , following Verrelst et al. (2016), then we would expect less bias in the validation plot for GISS. However, the poor validation plots did not translate into poorly estimated SIs for the emulator-PCA approach. On the contrary, the estimated SIs for GISS are consistent with the estimated SIs using the emulator-only approach (Fig. 5).

The use of PCA in variance based global sensitivity analysis studies is relatively new but has great potential for application in other settings. De Lozzo and Marrel (2017) used an atmospheric gas dispersion model to simulate the evolution and spatial distribution of a radioactive gas into the atmosphere following a chemical leak. The authors used principal component analysis to reduce the dimension of the spatio-temporal output map of gas concentrations to speed up the computation of the Sobol sensitivity indices for each of the ~19,000 points in the output space. This PCA-emulator hybrid approach was also used to estimate the Sobol sensitivity indices corresponding to a flood forecasting model that simulates the water level of the a river at 14 different points along its length (Roy et al., 2017). Using a crop model to simulate a variable related to nitrogen content of a crop over a growing season of 170 days, Lamboni et al. (2011) using PCA to reduce the dimension of the output space. However, unlike other comparable studies, the computed the sensitivity indices corresponded to the principal components, i.e. to a linear combination of the 170 output values. This is permissible to do if the principal components can be interpreted in some physical sense. For Lamboni et al. (2011), the first principal component (PC) approximately corresponded to mean

nitrogen content over the whole growing season while the second PC was the difference in Nitrogen content between the first and second halves of the growing season.

4.3 Scientific context of this study

Our work extends the work of Wild et al. (in prep.) who used the same training inputs and the same atmospheric chemical transport models (FRSGC and GISS), but different outputs. Instead of using highly multidimensional output of tropospheric methane lifetime values at different spatial locations, Wild et al. (in prep.) used a one-dimensional output of global tropospheric methane lifetime. Using the eFAST method, the authors found that global methane lifetime was most sensitive to change in the humidity input for the FRSGC model, while for the GISS model the surface NOx and the lightning NOx inputs were most important for predicting methane lifetime at the global scale, followed by the isoprene and the boundary layer mixing inputs Wild et al. (in prep.). As expected, our results indicated that these same inputs explained most of the variance in the outputs for the different spatial locations. However, while the humidity sensitivity index (SI) for GISS was very low at the global scale (SI = 5%) out study found that the SIs for humidity were very high (50-60%) for the higher latitude regions (Fig. 4).

4.4 Implications for large scale sensitivity analysis studies

GSA studies for expensive models involving a small number of inputs (e.g. <10) are useful and straightforward to implement (Lee et al., 2012). However, the inferences made are limited due to the large number of parameters on which these models depend and the number of processes that they simulate. Hence, interest is growing in carrying out large scale GSA studies involving a high number of inputs to improve understanding of an individual model (e.g. Lee et al., 2013) or to diagnose differences between models (Wild et al., in prep.). For GSA studies when the number of inputs is small, our study has demonstrated that the GAM approach is a good

Commented [E22]: RC2, comment #1. New comment to address this comment.

Commented [E23]: RC1, comment #4. New text in response to comment.

candidate for carrying out emulator-free GSA since it calculates very similar SIs without the computational demands of emulation. A caveat is that the performance of GAM may depend on the behaviour of the model; although we have found it is a good GSA method for our models (FRSGC and GISS) and output (CH_4 lifetimes) its suitability may not be as good in all situations.

5. Conclusion

Global sensitivity analysis (GSA) is a powerful tool for understanding model behaviour, for diagnosing differences between models and for determining which parameters to choose for model calibration. In this study, we compared different methods for computing first order sensitivity indices for computationally expensive models based on modelled spatial distributions of CH_4 lifetimes. We have demonstrated that the more established emulator-based methods (eFAST and Sobol) can be used to efficiently derive meaningful sensitivity indices for multi-dimensional output from atmospheric chemistry transport models. We have shown that an emulator-free method based on a generalised additive model (GAM) and an emulator-PCA hybrid method produce first order sensitivity indices that are consistent with the emulator-only methods. For a reasonably smooth system with few parameters, as investigated here, the GAM and PCA methods are viable and effective options for GSA, and are robust over models that exhibit distinctly different responses. Moreover, the computational benefit of these alternative methods is apparent, with the GAM approach allowing calculation of variance based sensitivity indices 22-56 times faster (or 37 times faster on average) compared to the eFAST or Sobol methods. Using the Sobol method, the PCA-emulator hybrid approach is 19-28 times faster (or 24 times faster on average) compared computing SIs compared to using an emulator-only

approach depending on which chemistry model is used. Finally, we have provided guidance on how to implement these methods in a reproducible way.

Code Availability

The R code to carry out global sensitivity analysis using the methods described in this paper are available in the sections S2-S7 of the supplemental material. This R code as well as the R code used to validate the emulators is also be found via http://doi.org/10.5281/zenodo.1038667.

Data Availability

The inputs and outputs of the FRSGC chemistry model that was used to train the emulators in this paper can be found via http://doi.org/10.5281/zenodo.1038670.

Appendix A: Further details of the Sobol and eFAST global sensitivity analysis methods

Sobol method: Saltelli (2002) and Tarantola et al. (2006) suggest using eight variants of equation (2), using different combinations of $\mathbf{y_A}$, $\mathbf{y_B}$, $\mathbf{y_{Ci}}$ and $\mathbf{y_{Di}}$:

$$\widehat{S}_i^{\,I} = \frac{\mathbf{Y_A}\cdot\mathbf{Y_{C_i}} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{B}}^{(j)}\right)}{\mathbf{Y_A}\cdot\mathbf{Y_A} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{B}}^{(j)}\right)} \qquad \widehat{S}_i^{\,II} = \frac{\mathbf{Y_A}\cdot\mathbf{Y_{D_i}} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{B}}^{(j)}\right)}{\mathbf{Y_B}\cdot\mathbf{Y_B} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{B}}^{(j)}\right)}$$

$$\widehat{S}_i^{\,III} = \frac{\mathbf{Y_A}\cdot\mathbf{Y_{C_i}} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{B}}^{(j)}\right)}{\mathbf{Y_B}\cdot\mathbf{Y_B} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{B}}^{(j)}\right)} \qquad \widehat{S}_i^{\,IV} = \frac{\mathbf{Y_A}\cdot\mathbf{Y_{C_i}} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{C_i}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{D_i}}^{(j)}\right)}{\mathbf{Y_{C_i}}\cdot\mathbf{Y_{C_i}} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{C_i}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{D_i}}^{(j)}\right)}$$

$$\widehat{S}_i^{\,V} = \frac{\mathbf{Y_A}\cdot\mathbf{Y_{C_i}} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{C_i}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{D_i}}^{(j)}\right)}{\mathbf{Y_{D_i}}\cdot\mathbf{Y_{D_i}} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{C_i}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{D_i}}^{(j)}\right)} \qquad \widehat{S}_i^{\,VI} = \frac{\mathbf{Y_B}\cdot\mathbf{Y_{D_i}} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{B}}^{(j)}\right)}{\mathbf{Y_A}\cdot\mathbf{Y_A} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{B}}^{(j)}\right)}$$

$$\widehat{S}_i^{\,VII} = \frac{\mathbf{Y_B}\cdot\mathbf{Y_{D_i}} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{C_i}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{D_i}}^{(j)}\right)}{\mathbf{Y_{C_i}}\cdot\mathbf{Y_{C_i}} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{C_i}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{D_i}}^{(j)}\right)} \qquad \widehat{S}_i^{\,VIII} = \frac{\mathbf{Y_B}\cdot\mathbf{Y_{D_i}} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{C_i}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{D_i}}^{(j)}\right)}{\mathbf{Y_{D_i}}\cdot\mathbf{Y_{D_i}} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{C_i}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathbf{D_i}}^{(j)}\right)}$$

Thus, the *i*th first order Sobol SI estimate is:

$$\widehat{S_i} = \frac{1}{8}\left(\widehat{S_i}^{I} + \widehat{S_i}^{II} + \widehat{S_i}^{III} + \widehat{S_i}^{IV} + \widehat{S_i}^{V} + \widehat{S_i}^{VI} + \widehat{S_i}^{VII} + \widehat{S_i}^{VIII}\right)$$

The extended FAST (eFAST) method: The main difference between classical FAST (Cukier et al., 1973), and extended FAST (Saltelli et al., 1999) when computing first order SIs is the choice of transformation function G_i:

Classical FAST: $G_i(z) = \bar{x}_i e^{\bar{v}_s z}$, ($\bar{x}_i$, $\bar{v}_s$ are user-specified) (A1a)

Extended FAST: $G_i(z) = \frac{1}{2} + \frac{1}{\pi}\arcsin(z)$ (A1b)

Using equation (A1b), equation (3) now becomes a straight line equation:

$$x_i(s) = \frac{1}{2} + \frac{1}{\pi}\omega_i s$$

Author contributions

ER and OW designed the study. ER conducted the analysis and wrote the manuscript and OW gave feedback during the analysis and writing up phases. OW, FO and AW provided output from the global atmospheric model runs needed to carry out the analysis. LL advised on statistical aspects of the analysis. All coauthors gave feedback on drafts of the manuscript.

Acknowledgements

This work was supported by the Natural Environment Research Council [grant number NE/N003411/1].

References

Ahtikoski, A., Heikkilä, J., Alenius, V., and Siren, M.: Economic viability of utilizing biomass energy from young stands—the case of Finland, Biomass and Bioenergy, 32, 988-996, 2008.

Ba, S., Myers, W. R., and Brenneman, W. A.: Optimal sliced Latin hypercube designs, Technometrics, 57, 479-487, 2015.

Bailis, R., Ezzati, M., and Kammen, D. M.: Mortality and greenhouse gas impacts of biomass and petroleum energy futures in Africa, Science, 308, 98-103, 2005.

Bastos, L. S. and O'Hagan, A.: Diagnostics for Gaussian process emulators, Technometrics, 51, 425-438, 2009.

Campbell, J. E., Carmichael, G. R., Chai, T., Mena-Carrasco, M., Tang, Y., Blake, D., Blake, N., Vay, S. A., Collatz, G. J., and Baker, I.: Photosynthetic control of atmospheric carbonyl sulfide during the growing season, Science, 322, 1085-1088, 2008.

Carslaw, K., Lee, L., Reddington, C., Pringle, K., Rap, A., Forster, P., Mann, G., Spracklen, D., Woodhouse, M., and Regayre, L.: Large contribution of natural aerosols to uncertainty in indirect forcing, Nature, 503, 67-71, 2013.

Chang, E. T., Strong, M., and Clayton, R. H.: Bayesian sensitivity analysis of a cardiac cell model using a Gaussian process emulator, PloS one, 10, e0130252, 2015.

Coggan, J. S., Bartol, T. M., Esquenazi, E., Stiles, J. R., Lamont, S., Martone, M. E., Berg, D. K., Ellisman, M. H., and Sejnowski, T. J.: Evidence for ectopic neurotransmission at a neuronal synapse, Science, 309, 446-451, 2005.

Cressie, N.: The origins of kriging, Mathematical geology, 22, 239-252, 1990.

Cukier, R., Fortuin, C., Shuler, K. E., Petschek, A., and Schaibly, J.: Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. I Theory, The Journal of chemical physics, 59, 3873-3878, 1973.

Currin, C., Mitchell, T., Morris, M., and Ylvisaker, D.: Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments, Journal of the American Statistical Association, 86, 953-963, 1991.

de Gee, M., Lof, M. E., and Hemerik, L.: The effect of chemical information on the spatial distribution of fruit flies: II parameterization, calibration, and sensitivity, Bulletin of mathematical biology, 70, 1850, 2008.

De Lozzo, M. and Marrel, A.: Sensitivity analysis with dependence and variance-based measures for spatio-temporal numerical simulators, Stochastic Environmental Research and Risk Assessment, 31, 1437-1453, 2017.

Degroote, J., Couckuyt, I., Vierendeels, J., Segers, P., and Dhaene, T.: Inverse modelling of an aneurysm's stiffness using surrogate-based optimization and fluid-structure interaction simulations, Structural and Multidisciplinary Optimization, 46, 457-469, 2012.

Ferretti, F., Saltelli, A., and Tarantola, S.: Trends in sensitivity analysis practice in the last decade, Science of The Total Environment, 2016. 2016.

Goldstein, M. and Rougier, J.: Bayes linear calibrated prediction for complex systems, Journal of the American Statistical Association, 101, 1132-1143, 2006.

Gómez-Dans, J. L., Lewis, P. E., and Disney, M.: Efficient Emulation of Radiative Transfer Codes Using Gaussian Processes and Application to Land Surface Parameter Inferences, Remote Sensing, 8, 119, 2016.

Hankin, R. K.: Introducing BACCO, an R package for Bayesian analysis of computer code output, Journal of Statistical Software, 14, 1-21, 2005.

Hill, T. C., Ryan, E., and Williams, M.: The use of CO2 flux time series for parameter and carbon stock estimation in carbon cycle research, Global Change Biology, 18, 179-193, 2012.

Homma, T. and Saltelli, A.: Importance measures in global sensitivity analysis of nonlinear models, Reliability Engineering & System Safety, 52, 1-17, 1996.

Iooss, B. and Lemaître, P.: A review on global sensitivity analysis methods. In: Uncertainty Management in Simulation-Optimization of Complex Systems, Springer, 2015.

Kennedy, M., Anderson, C., O'Hagan, A., Lomas, M., Woodward, I., Gosling, J. P., and Heinemeyer, A.: Quantifying uncertainty in the biospheric carbon flux for England and Wales, Journal of the Royal Statistical Society: Series A (Statistics in Society), 171, 109-135, 2008.

Kennedy, M. C. and O'Hagan, A.: Predicting the output from a complex computer code when fast approximations are available, Biometrika, 87, 1-13, 2000.

Koehler, J. and Owen, A.: 9 Computer experiments, Handbook of statistics, 13, 261-308, 1996.

Lamboni, M., Monod, H., and Makowski, D.: Multivariate sensitivity analysis to measure global contribution of input factors in dynamic models, Reliability Engineering & System Safety, 96, 450-459, 2011.

Lee, L., Carslaw, K., Pringle, K., and Mann, G.: Mapping the uncertainty in global CCN using emulation, Atmospheric Chemistry and Physics, 12, 9739-9751, 2012.

Lee, L., Pringle, K., Reddington, C., Mann, G., Stier, P., Spracklen, D., Pierce, J., and Carslaw, K.: The magnitude and causes of uncertainty in global model simulations of cloud condensation nuclei, Atmos. Chem. Phys, 13, 8879-8914, 2013.

Lilburne, L. and Tarantola, S.: Sensitivity analysis of spatial models, International Journal of Geographical Information Science, 23, 151-168, 2009.

Loeppky, J. L., Sacks, J., and Welch, W. J.: Choosing the sample size of a computer experiment: A practical guide, Technometrics, 51, 366-376, 2009.

Mara, T. A. and Tarantola, S.: Application of global sensitivity analysis of model output to building thermal simulations, 2008, 290-302.

Marrel, A., Iooss, B., Laurent, B., and Roustant, O.: Calculations of sobol indices for the gaussian process metamodel, Reliability Engineering & System Safety, 94, 742-751, 2009.

O'Hagan, A.: Bayesian analysis of computer code outputs: a tutorial, Reliability Engineering & System Safety, 91, 1290-1300, 2006.

Oakley, J. E. and O'Hagan, A.: Probabilistic sensitivity analysis of complex models: a Bayesian approach, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 66, 751-769, 2004.

Pistone, G. and Vicario, G.: Kriging prediction from a circular grid: application to wafer diffusion, Applied Stochastic Models in Business and Industry, 29, 350-361, 2013.

Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K.: Surrogate-based analysis and optimization, Progress in aerospace sciences, 41, 1-28, 2005.

Rasmussen, C. E.: Gaussian processes for machine learning, 2006. 2006.

Ripley, B. D.: Spatial statistics, John Wiley & Sons, 2005.

Roustant, O., Ginsbourger, D., and Deville, Y.: DiceKriging, DiceOptim: Two R packages for the analysis of computer experiments by kriging-based metamodeling and optimization, 2012. 2012.

Roy, P. T., El Moçayd, N., Ricci, S., Jouhaud, J.-C., Goutal, N., De Lozzo, M., and Rochoux, M. C.: Comparison of Polynomial Chaos and Gaussian Process surrogates for uncertainty

quantification and correlation estimation of spatially distributed open-channel steady flows, Stochastic Environmental Research and Risk Assessment, 2017. 1-19, 2017.

Saltelli, A.: Making best use of model evaluations to compute sensitivity indices, Computer Physics Communications, 145, 280-297, 2002.

Saltelli, A., Andres, T., and Homma, T.: Sensitivity analysis of model output: an investigation of new techniques, Computational statistics & data analysis, 15, 211-238, 1993.

Saltelli, A. and Annoni, P.: How to avoid a perfunctory sensitivity analysis, Environmental Modelling & Software, 25, 1508-1517, 2010.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S.: Global sensitivity analysis: the primer, John Wiley & Sons, 2008.

Saltelli, A., Ratto, M., Tarantola, S., and Campolongo, F.: Update 1 of: Sensitivity analysis for chemical models, Chemical reviews, 112, PR1-PR21, 2012.

Saltelli, A., Tarantola, S., and Chan, K.-S.: A quantitative model-independent method for global sensitivity analysis of model output, Technometrics, 41, 39-56, 1999.

Schmidt, G. A., Kelley, M., Nazarenko, L., Ruedy, R., Russell, G. L., Aleinov, I., Bauer, M., Bauer, S. E., Bhat, M. K., and Bleck, R.: Configuration and assessment of the GISS ModelE2 contributions to the CMIP5 archive, Journal of Advances in Modeling Earth Systems, 6, 141-184, 2014.

Sexton, D. M., Murphy, J. M., Collins, M., and Webb, M. J.: Multivariate probabilistic projections using imperfect climate models part I: outline of methodology, Climate dynamics, 38, 2513-2542, 2012.

Shindell, D., Faluvegi, G., Unger, N., Aguilar, E., Schmidt, G., Koch, D., Bauer, S. E., and Miller, R. L.: Simulations of preindustrial, present-day, and 2100 conditions in the NASA GISS composition and climate model G-PUCCINI, Atmospheric Chemistry and Physics, 6, 4427-4459, 2006.

Sobie, E. A.: Parameter sensitivity analysis in electrophysiological models using multivariable regression, Biophysical journal, 96, 1264-1274, 2009.

Sobol, I. y. M.: On sensitivity estimation for nonlinear mathematical models, Matematicheskoe Modelirovanie, 2, 112-118, 1990.

Stanfill, B., Mielenz, H., Clifford, D., and Thorburn, P.: Simple approach to emulating complex computer models for global sensitivity analysis, Environmental Modelling & Software, 74, 140-155, 2015.

Stites, E. C., Trampont, P. C., Ma, Z., and Ravichandran, K. S.: Network analysis of oncogenic Ras activation in cancer, Science, 318, 463-467, 2007.

Strong, M., Oakley, J. E., and Brennan, A.: An efficient method for computing the Expected Value of Sample Information. A non-parametric regression approach. 2015a.

Strong, M., Oakley, J. E., and Brennan, A.: Estimating multiparameter partial expected value of perfect information from a probabilistic sensitivity analysis sample a nonparametric regression approach, Medical Decision Making, 34, 311-326, 2014.

Strong, M., Oakley, J. E., Brennan, A., and Breeze, P.: Estimating the expected value of sample information using the probabilistic sensitivity analysis sample a fast nonparametric regression-based method, Medical Decision Making, 2015b. 0272989X15575286, 2015b.

Tarantola, S., Gatelli, D., and Mara, T. A.: Random balance designs for the estimation of first order global sensitivity indices, Reliability Engineering & System Safety, 91, 717-727, 2006.

Vanuytrecht, E., Raes, D., and Willems, P.: Global sensitivity analysis of yield output from the water productivity model, Environmental Modelling & Software, 51, 323-332, 2014.

Verrelst, J., Sabater, N., Rivera, J. P., Muñoz-Marí, J., Vicent, J., Camps-Valls, G., and Moreno, J.: Emulation of Leaf, Canopy and Atmosphere Radiative Transfer Models for Fast Global Sensitivity Analysis, Remote Sensing, 8, 673, 2016.

Voulgarakis, A., Naik, V., Lamarque, J.-F., Shindell, D. T., Young, P., Prather, M. J., Wild, O., Field, R., Bergmann, D., and Cameron-Smith, P.: Analysis of present day and future OH and methane lifetime in the ACCMIP simulations, Atmospheric Chemistry and Physics, 13, 2563-2587, 2013.

Vu-Bac, N., Rafiee, R., Zhuang, X., Lahmer, T., and Rabczuk, T.: Uncertainty quantification for multiscale modeling of polymer nanocomposites with correlated parameters, Composites Part B: Engineering, 68, 446-464, 2015.

Welch, W. J., Buck, R. J., Sacks, J., Wynn, H. P., Mitchell, T. J., and Morris, M. D.: Screening, predicting, and computer experiments, Technometrics, 34, 15-25, 1992.

Wild, O.: Modelling the global tropospheric ozone budget: exploring the variability in current models, Atmospheric Chemistry and Physics, 7, 2643-2660, 2007.

Wild, O., Pochanart, P., and Akimoto, H.: Trans-Eurasian transport of ozone and its precursors, Journal of Geophysical Research: Atmospheres, 109, 2004.

Wild, O. and Prather, M. J.: Excitation of the primary tropospheric chemical mode in a global three-dimensional model, Journal of geophysical research, 105, 2000.

Wild, O., Ryan, E., O'Connor, F., Vougarakis, A., and Lee, L.: Reducing Uncertainty in Model Budgets of Tropospheric Ozone and OH., Intended for submission to Atmospheric Chemistry and Physics, in prep., in prep.

Wold, S., Sjöström, M., and Eriksson, L.: PLS-regression: a basic tool of chemometrics, Chemometrics and intelligent laboratory systems, 58, 109-130, 2001.

Wood, S. N.: Generalized additive models: an introduction with R, CRC press, 2017.

Wu, J., Dhingra, R., Gambhir, M., and Remais, J. V.: Sensitivity analysis of infectious disease models: methods, advances and their application, Journal of The Royal Society Interface, 10, 20121018, 2013.

Figure and Tables

Table 1 Summary of algebraic terms used in this study that are common to all of most of the statistical methods described in this study. For brevity, the terms that are specific to a particular method are not listed here.

Commented [E24]: RC2, comment #2. I decided to add this table so that it was clear what the different terms mean, however I only include terms that are used throughout the manuscript. For brevity, the variables that are distinct to a particular statistical method are not listed here.

Symbol	Description	Eqn(s).
$S_{i,j}$	The first order sensitivity index corresponding to the ith input variable (i=1, 2, …, p) and the jth point in the output space.	
X	The $n \times p$ matrix which stores the n sets of p-dimensional inputs that used to as part of the calculation to compute the sensitivity indices.	
X_i	The ith column of matrix X.	
Y	The $n \times m$ matrix which stores the n sets of m-dimensional outputs (corresponding to the n sets of inputs stored in X) that used to as part of the calculation to compute the sensitivity indices.	
Y_j	The jth column of matrix Y.	
n	In general, n is the number of executions of the simulator required to compute the sensitivity indices. For this study, n is the number of executions of the 'emulator' required to compute the sensitivity indices since the simulator is computationally too slow to run. For the Sobol and eFAST methods, n = 1000-10,000 (for this study we used n = 10,000 for Sobol and n = 5000 for eFAST). For the GAM and PLS methods, we believe $n < 100$ is sufficient (for this study we used $n = N = 80$).	
p	The number of input variables / the dimension of the input space.	
m	The number of output variables / the dimension of the output space.	
N	The number of executions of the simulator required to train an emulator (for this study N=80).	
$\mathbf{X}$	The $N \times p$ matrix which stores the N sets of p-dimensional inputs that used for two purposes: (i) in the calculations to train the emulators that are used to replace the simulator (see §2.3); (ii) in the calculation of the sensitivity indices using the sensitivity analysis methods that do not require an emulator (namely GAM and PLS).	
$\mathbf{X_i}$	A column vector represented by the ith column of matrix $\mathbf{X}$ (i=1,2,…,p).	
$\mathbf{x_i}$	The row vector represented by the ith row of matrix $\mathbf{X}$ (i=1,2,…,N)	
$\mathbf{Y}$	The $n \times m$ matrix which stores the n sets of m-dimensional simulator outputs (corresponding to the n sets of inputs stored in $\mathbf{X}$) that used to as part of the calculation to compute the sensitivity indices.	
$\mathbf{Y_j}$	The jth column of matrix $\mathbf{Y}$ (j=1,2,…,m)	
$\mathbf{y_i}$	The simulator output after the simulator has been run at the p-dimensional	

	input given by $\mathbf{x}_i$ (i=1,2,…,N)	

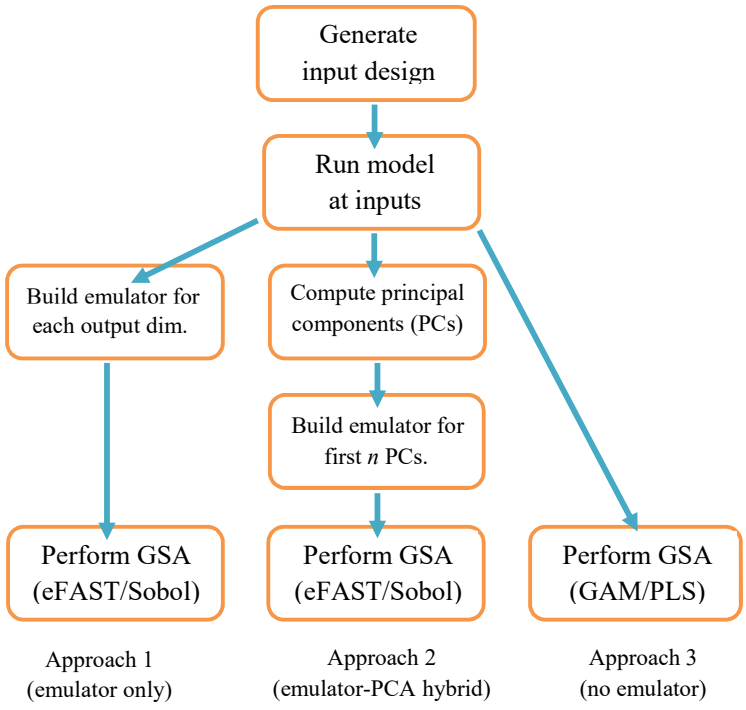

Figure 1. Flow-chart for order of tasks to complete in order to perform global sensitivity analysis (GSA) on a computationally expensive model. The ranges on the inputs, from which its design is based, are determined by expert elicitation. For approach 1, the dimensions of the output consist of different spatial or temporal points of the same output variable (CH_4 lifetime for this study). For approach 2, a principal component (PCs) is a linear combination of the different dimensions of the output, where n is chosen such that the first n PCs explain 99% of the variance of the output.

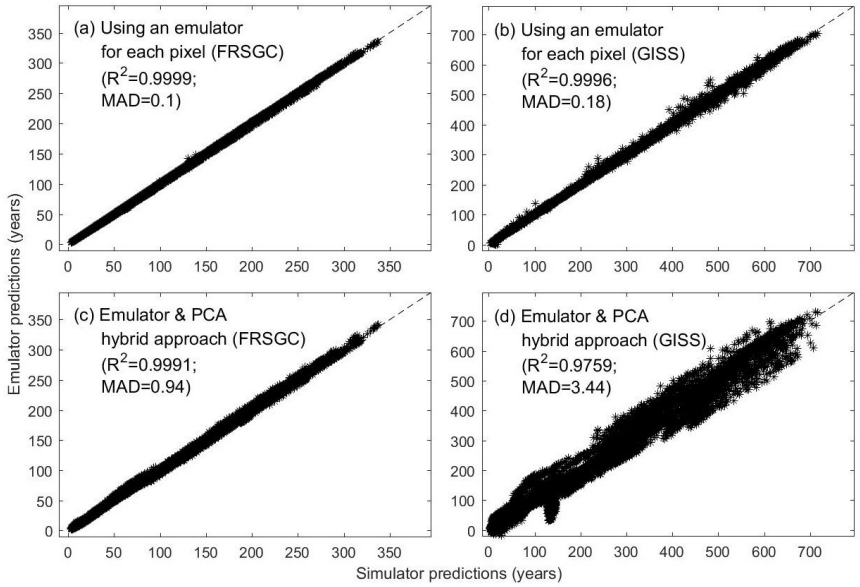

Figure 2. Annual column mean CH_4 lifetime calculated by the FRSGC and GISS chemistry models from each of 24 validation runs (x-axis) versus that predicted by the emulator (y-axis). In each plot, the R^2 and median absolute difference (MAD) are given as metrics for the accuracy of the emulator predictions. Each validation run contains ~2000 different output values, corresponding to different latitude-longitude grid squares.

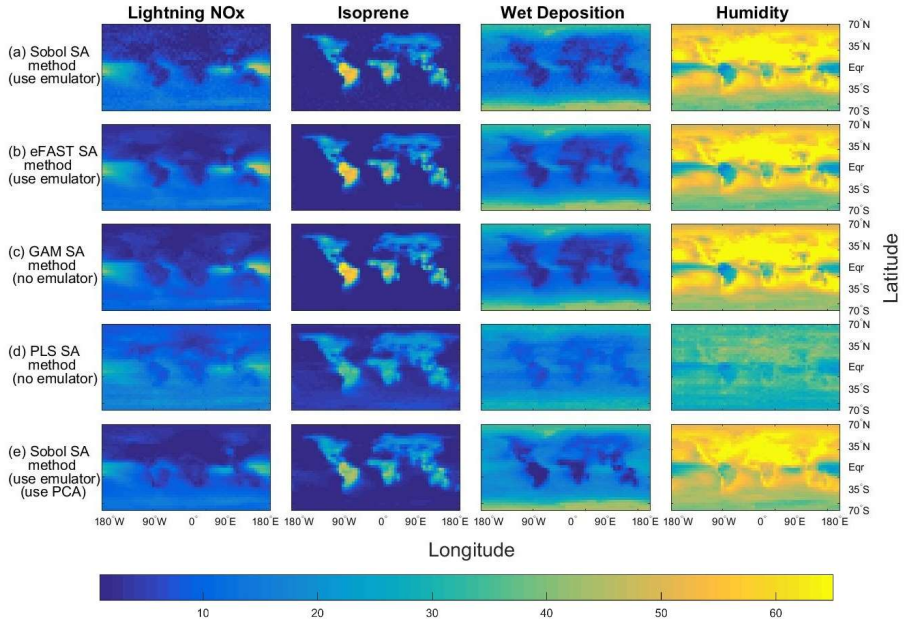

Figure 3. The sensitivity indices (percentage of the total variance in a given output) for the four dominant inputs, for annual column mean CH_4 lifetime in the FRSGC chemistry transport model. The rows show the results from five different methods for performing sensitivity analysis (SA), whose formulae for computing the SIs are given by Eqs. 1,2 and §2.3 (Sobol method & emulator), Eqs. 1, 6a-b, §2.3 (eFAST method & emulator), Eqs 1, 9 (GAM method), Eq 10 (PLS method), Eqs 1,2, §2.3 and §2.5 (Sobol method & emulator & PCA).

Commented [RE25]: RC1 comment #2. New text to address comment.

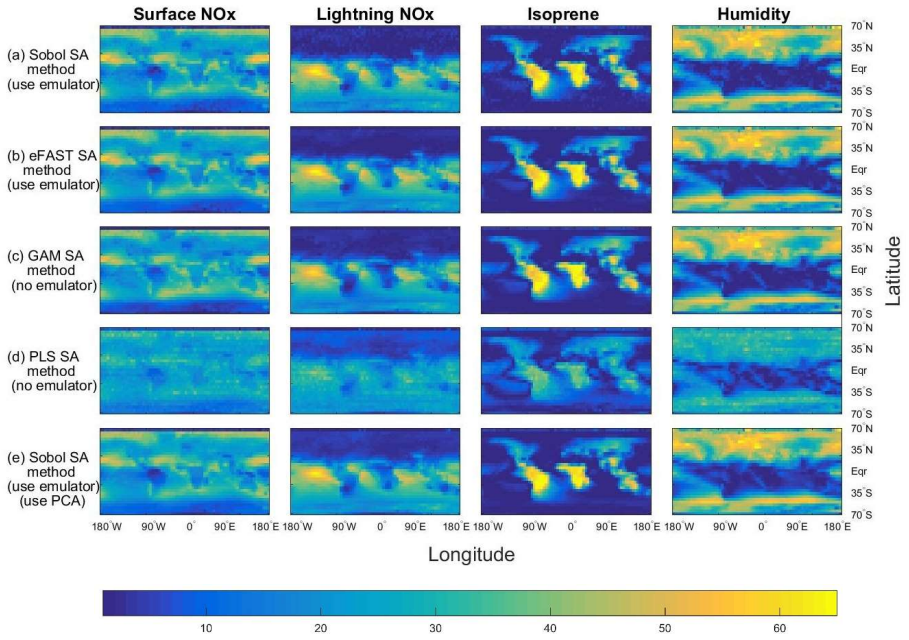

Figure 4. The sensitivity indices (percentage of the total variance in a given output) for the four dominant inputs, for annual column mean CH_4 lifetime in the GISS chemistry transport model. See caption for figure 3 for further details about the five methods used.

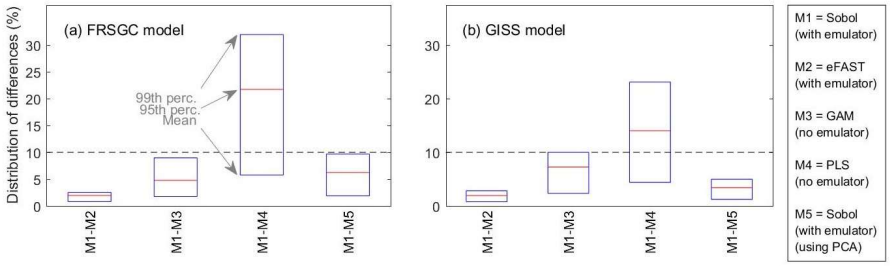

Figure 5. Statistics (mean, 95% percentile and maximum) of the distribution of differences in sensitivity indices (SIs) between pairs of methods. For each comparison, the 16,000 pairs of SIs are made up of ~2000 pairs of SIs for each of the 8 inputs.