Geosci. Model Dev., 11, 3131–3146, 2018
https://doi.org/10.5194/gmd-11-3131-2018

Geoscientific Model Development
Open Access
EGU

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

Edmund Ryan[1], **Oliver Wild**[1], **Apostolos Voulgarakis**[2], **and Lindsay Lee**[3]

[1]Lancaster Environment Centre, Lancaster University, Lancaster, UK
[2]Department of Physics, Imperial College London, London, UK
[3]School of Earth and Environment, University of Leeds, Leeds, UK

Correspondence: Edmund Ryan (edmund.ryan@lancaster.ac.uk)

Received: 30 October 2017 – Discussion started: 13 November 2017
Revised: 30 May 2018 – Accepted: 6 June 2018 – Published: 3 August 2018

Abstract. Global sensitivity analysis (GSA) is a powerful approach in identifying which inputs or parameters most affect a model's 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 Fourier Amplitude Sensitivity Test (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 $\sim$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 emulator–PCA and GAM methods accurately estimated the SIs of the $\sim$2000 methane lifetime outputs but were on average 24 and 37 times faster, respectively.

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, (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 NO_x emissions from lightning. The observation-

based range of 3–8 TgN yr^{-1} 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).

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 (eFAST) 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 sensitivity indices (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 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. 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 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 expensive model that are used to train the emulator, the resulting outputs of the emulator must exactly match those of the expensive model (Iooss and Lemaître, 2015). Secondly, for inputs that the emulator is not trained at, a probability distribution of the outputs that represents their uncertainty is given (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 a basic 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 generalised additive modelling (Mara and Tarantola, 2008;

Strong et al., 2014, 2015b) or on a partial least squares approach (Chang et al., 2015; Sobie, 2009). A separate generalised 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 is 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 they 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 emulator–PCA 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 hypothesise 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 the Frontier Research System for Global Change/University of California at Irvine (FRSGC/UCI) and Goddard Institute for Space Studies (GISS) models – 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 and (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 $\times$ longitude $\times$ 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 they 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 GSA on two such atmospheric models. We used the FRSGC/UCI chemistry transport model (CTM) (Wild et al., 2004; Wild and Prather, 2000) and the GISS general circulation model (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., 2018). This involved varying eight inputs or parameters over specified ranges using a maximin Latin hypercube design: global surface NO_x emissions (30–50 TgN yr^{-1}), global lightning NO_x emissions (2–8 TgN yr^{-1}), global isoprene emissions (200–800 TgC yr^{-1}), dry deposition rates (model value $\pm 80\,\%$), wet deposition rates (model value $\pm 80\,\%$), humidity (model value $\pm 50\,\%$), cloud optical depth (model value $\times$ 0.1–10) and boundary layer mixing (model value $\times$ 0.01–100). For this study, we focus on a single model output, namely the global distribution of tropospheric columns of mean methane (CH_4) lifetime at the annual timescale. 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 $2.8^\circ \times 2.8^\circ$ for FRSGC and $2.5^\circ \times 2.0^\circ$ 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 of one another – which they are for this study – the first-order SI, corresponding to the ith input variable ($i = 1, 2, ..., p$) and the jth point in the output space, is

given by

$$S_{i,j} = \frac{\mathrm{Var}\left[E\left(\boldsymbol{Y}_j|\boldsymbol{X}_i\right)\right]}{\mathrm{Var}(\boldsymbol{Y}_j)} \times 100, \quad (1)$$

where $\boldsymbol{X}_i$ is the ith column of the $n \times p$ matrix $\mathbf{X}$ (i.e. a matrix with n rows and p columns) which stores the n samples of p-dimensional inputs, and $\boldsymbol{Y}_j$ is the jth column of the $n \times m$ matrix which stores the corresponding n sets of m-dimensional outputs (Table 1). We multiply by 100 so that the SI is given as a percentage. The notation given by $\mathrm{Var}(\cdot)$ and $E(\cdot)$ denotes 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 Eq. (1), in part because it requires fewer executions of the simulator (Homma and 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 $\boldsymbol{Y}_\mathrm{A} = f$ (**A**), $\boldsymbol{Y}_\mathrm{B} = f$ (**B**), $\boldsymbol{Y}_\mathrm{Ci} = f$ (**Ci**) and $\boldsymbol{Y}_\mathrm{Di} = f$ (**Di**). Vectors $\boldsymbol{Y}_\mathrm{A}$ and $\boldsymbol{Y}_\mathrm{Ci}$ are then substituted into Eq. (2):

$$\hat{S_{i,j}} = \frac{\hat{\mathrm{Var}}\left[\hat{E}\left(\boldsymbol{Y}_j|\boldsymbol{X}_i\right)\right]}{\hat{\mathrm{Var}}(\boldsymbol{Y}_j)} \times 100 = \frac{\boldsymbol{Y}_\mathrm{A} \cdot \boldsymbol{Y}_\mathrm{Ci} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_\mathrm{A}^{(j)}\right)^2}{\boldsymbol{Y}_\mathrm{A} \cdot \boldsymbol{Y}_\mathrm{A} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_\mathrm{A}^{(j)}\right)^2} \times 100, \quad (2)$$

where $\boldsymbol{Y}_\mathrm{A} \cdot \boldsymbol{Y}_\mathrm{Ci} = \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_\mathrm{A}^{(j)} \boldsymbol{Y}_\mathrm{Ci}^{(j)}\right)$, and $\boldsymbol{Y}_\mathrm{A}^{(j)}$ and $\boldsymbol{Y}_\mathrm{Ci}^{(j)}$ are the jth elements of $\boldsymbol{Y}_\mathrm{A}$ and $\boldsymbol{Y}_\mathrm{Ci}$ (equivalent formula for $\boldsymbol{Y}_\mathrm{A} \cdot \boldsymbol{Y}_\mathrm{A}$). 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 Eq. (2), using different combinations of $\boldsymbol{Y}_\mathrm{A}$, $\boldsymbol{Y}_\mathrm{B}$, $\boldsymbol{Y}_\mathrm{Ci}$ and $\boldsymbol{Y}_\mathrm{Di}$ (Appendix A). Lilburne and Tarantola (2009) proposed using the average of these eight SI estimates as they deemed this to be more accurate than a single estimate. We used this approach by Lilburne and Tarantola (2009) for this study.

2.2.2 The extended FAST method

An alternative and even faster way of estimating the terms in Eq. (1) is to use the eFAST 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., 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

$$x_i(s) = G_i\left(\sin\left(\omega_i s\right)\right), \quad (3)$$

where $\boldsymbol{x} = (x_1, ..., x_\mathrm{p})$ refers to a general point in the input space that has been sampled, $s \in R$ is a variable over the range $(-\infty, \infty)$, 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 the x_is being simultaneously varied. Depending on the simulator, we typically require $n = 1000$–$10\,000$ 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 $y = f(s) = f\left(x_1(s), x_2(s), \ldots, x_\mathrm{p}(s)\right)$ as a Fourier series:

$$y = f(s) = \sum_{j=-\infty}^{\infty} A_j \cos(js) + B_j \sin(js). \quad (4)$$

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

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

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

$$\hat{\mathrm{Var}}\left[\hat{E}\left(\boldsymbol{Y}_j|\boldsymbol{X}_i\right)\right] = \sum_{q \in Z^0} A_{q\omega_i}^2 + B_{q\omega_i}^2, \quad (6a)$$

where Z^0 is the set of all integers except zero. The total variance (denominator of Eq. 1) is

$$\hat{\mathrm{Var}}(\boldsymbol{Y}_j) = \sum_{k \in Z^0} A_k^2 + B_k^2. \quad (6b)$$

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.

Symbol	Description
$S_{i,j}$	The first-order sensitivity index corresponding to the ith input variable ($i = 1, 2, ..., p$) and the jth point in the output space
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, $N = 80$)
$\mathbf{X}$	Apart from Eq. (1), $\mathbf{X}$ refers to the $N \times p$ matrix which stores the N sets of p-dimensional inputs that are used for two purposes: (i) in the calculations to train the emulators that are used to replace the simulator (see Sect. 2.3) and (ii) in the calculation of the sensitivity indices using the sensitivity analysis methods that do not require an emulator (namely GAM and PLS). For Eq. (1), $\mathbf{X}$ also refers to the $n \times p$ matrix to compute the SIs if the simulator is computationally cheap to run.
$\boldsymbol{X}_i$	A column vector represented by the ith column of matrix $\mathbf{X}$ ($i = 1, 2, ..., p$)
$\boldsymbol{x}_i$	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 are used as part of the calculation to compute the sensitivity indices
$\boldsymbol{Y}_j$	The jth column of matrix $\mathbf{Y}$ ($j = 1, 2, ..., m$)
$\boldsymbol{y}_i$	The simulator output after the simulator has been run at the p-dimensional input given by $\boldsymbol{x}_i$ ($i = 1, 2, ..., N$)

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

2.3 Gaussian process emulators

When the simulator is computationally 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 $\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N$ and N sets of one-dimensional outputs from the simulator given by $y_1 = f(\boldsymbol{x}_1)$, $y_2 = f(\boldsymbol{x}_2)$, ..., $y_N = f(\boldsymbol{x}_N)$; f represents the simulator and for our study $N = 80$ (see Sect. 2.6). The most common form of an emulator is a GP since it has attractive mathematical properties that allow an analytical derivation of the mean and variance of the emulated output (given by $\hat{f}(\boldsymbol{x})$ for a general input $\boldsymbol{x}$). 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 $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. A GP has a mean function which is typically given by $m(\boldsymbol{x}) = E(f(\boldsymbol{x}))$ and covariance function given by $c(\boldsymbol{x}, \boldsymbol{x}') = \mathrm{cov}(f(\boldsymbol{x}), f(\boldsymbol{x}'))$, where $\boldsymbol{x}$ and $\boldsymbol{x}'$ are two different p-dimensional inputs. For the latter, we used a Matern (5/2) function (Roustant et al., 2012), which is given by

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

where s denotes the standard deviation and $\boldsymbol{\theta}$ 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).

Developed around the same time, the kriging interpolation methods used in geostatistics are mathematically equivalent to the GP methods developed by Currin et al. (1991) (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) GAM and (ii) a 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 Sect. 2.3. In the descriptions of these two sensitivity analysis methods (Sect. 2.4.1 and 2.4.2), we thus use $\mathbf{X} = [\boldsymbol{X}_1 \boldsymbol{X}_2, \ldots, \boldsymbol{X}_\mathrm{p}]$ and $\mathbf{Y}$ to denote the matrices that store N sets of p-dimensional inputs and m-dimensional outputs.

2.4.1 The generalised additive modelling method

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

$$\boldsymbol{Y}_j = g(\mathbf{X}) + \varepsilon \tag{8a}$$

$$g(\mathbf{X}) = s(\boldsymbol{X}_1) + s(\boldsymbol{X}_2) + \ldots + s\left(\boldsymbol{X}_\mathrm{p}\right), \tag{8b}$$

where $\boldsymbol{X}_\mathrm{i}$ is the ith column of input matrix $\mathbf{X}$ ($i = 1, 2, ..., p$); $\boldsymbol{Y}_j$ 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 $s(\boldsymbol{X}_1, \boldsymbol{X}_2) + s(\boldsymbol{X}_1, \boldsymbol{X}_3) + \ldots + s\left(\boldsymbol{X}_{\mathrm{p}-1}, \boldsymbol{X}_\mathrm{p}\right)$ to the right-hand side of Eq. (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 (Simon N. Wood, personal communication, 23 May 2017). It is still a meta-model and we could use it as a surrogate of the computationally expensive simulator 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 simulator is too many (i.e. too computationally burdensome). Instead, it is possible to obtain accurate estimates of the first-order SIs by using a GAM to estimate the components of Eq. (1) directly (Stanfill et al., 2015; Strong et al., 2014, 2015b). To compute the ith first-order SI ($1 \leq i \leq p$), we first recognise that taking the expectation of Eq. (8a) leads to $E\left(\boldsymbol{Y}_j\right) = g(\mathbf{X})$. The expression for $E\left(\boldsymbol{Y}_j|\boldsymbol{X}_i\right)$ is thus the marginal distribution of $E\left(\boldsymbol{Y}_j\right)$. We could fit the full model and then compute this marginal distribution following Stanfill et al. (2015). However, an easier and quicker way is to fit a GAM to the $(\boldsymbol{X}_i, \boldsymbol{Y}_j)$ "data" where $\boldsymbol{X}_i$ and $\boldsymbol{Y}_j$ are defined above. Then, $E\left(\boldsymbol{Y}_j|\boldsymbol{X}_i\right)$ consists of the fitted values of this reduced model (Strong et al., 2015b). Thus, $\mathrm{Var}\left[E\left(\boldsymbol{Y}_j|\boldsymbol{X}_i\right)\right]$ (numerator of equation 1) is determined by computing the variance of the n points from this fitted GAM model. In other words,

$$\hat{\mathrm{Var}}\left[\hat{E}\left(\boldsymbol{Y}_j|\boldsymbol{X}_i\right)\right] = \mathrm{var}\left(s\left(x_{1,i}\right), s\left(x_{2,i}\right), \ldots, s\left(x_{n,i}\right)\right), \tag{9}$$

where $x_{k,i}$ is the element from the kth row and ith column of matrix $\mathbf{X}$. Finally, the denominator term of Eq. (1) is computed by taking the variance of the n samples of the outputs from the computationally expensive simulator that are stored in $\boldsymbol{Y}_j$.

2.4.2 The partial least squares method

The 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 ($\mathbf{Y}$) based on a set of inputs or predictor variables ($\mathbf{X}$), where $\mathbf{X}$ and $\mathbf{Y}$ are matrices (Table 1). MLR is only appropriate to use when the different inputs (columns in $\mathbf{X}$) 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). 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 Sect. S1 in the Supplement for details). PLS regression is then performed where the regression coefficients represent the sensitivity indices (given as a percentage). 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 following formula:

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

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 PCA to reduce the dimensionality of the output space, and then used 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 compute the sensitivity indices.

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

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

where $\mathbf{Y}$ is the $N \times m$ matrix of training outputs from the simulator (see Sect. 2.3), and $\boldsymbol{A}^*$ is a matrix whose columns are orthogonal to one another and whose ith column ($\boldsymbol{A}_i^*$) is chosen such that var($\mathbf{Y}\boldsymbol{A}_i^*$) is maximised subject to the constraint $\left(\boldsymbol{A}_i^*\right)^T\boldsymbol{A}_i^* = 1$. The vector $\boldsymbol{A}_1^*$ is called the first principal component (PC1), and we define λ_1 to be the principle eigenvalue of $S = \mathrm{var}\,(Y)$ which is the largest variance of the outputs $\mathbf{Y}$ with respect to PC1. The second, third, fourth columns, etc. of $\mathbf{A}^*$ 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 $\mathbf{Y}$, 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 cutoff 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 $m = \sim 2000$ spatially varying points to the first N_{pc} principal components (e.g. $N_{\mathrm{pc}} = 5$ for the FRSGC model; see Sect. 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} ($= \sim 400$ using above m and N_{pc} values). However, after having generated

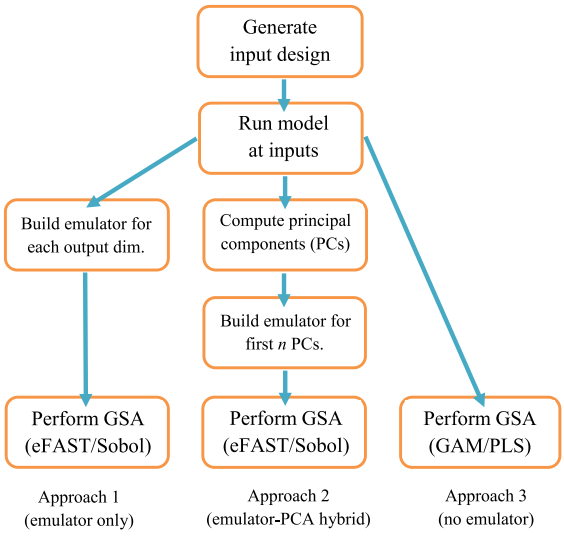

Figure 1. Flowchart for order of tasks to complete in order to perform GSA on a computationally expensive model. The ranges on the inputs, on which its design is based, are determined by expert elicitation. For approach 1, each dimension (dim.) of the output consists of a different spatial or temporal point of the same variable (CH_4 lifetime for this study). For approach 2, a PC 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.

the N_{pc} sets of output vectors for the Sobol method ($\boldsymbol{Y}_{\mathrm{A}}^{(\mathrm{PC})}$, $\boldsymbol{Y}_{\mathrm{B}}^{(\mathrm{PC})}$, $\boldsymbol{Y}_{\mathrm{Ci}}^{(\mathrm{PC})}$, $\boldsymbol{Y}_{\mathrm{Di}}^{(\mathrm{PC})}$; see Sect. 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 $(N_{\mathrm{pc}}+1)$th, $(N_{\mathrm{pc}}+2)$th, ..., m columns of the matrix $\mathbf{A}^*$ (Eq. 11) to zero and call this new matrix $\mathbf{A}^*_{\mathrm{sample}}$. We also form a $n \times m$ matrix $\mathbf{Y}^{(\mathrm{PC})}_{\mathrm{sample}}$ whose first N_{pc} columns are vectors storing the emulator outputs corresponding to the first N_{pc} principal components, while the elements of the remaining columns are set to zero. Recall that $\mathbf{Y}^{(\mathrm{PC})}_{\mathrm{sample}}$ is different from $\mathbf{Y}^{(\mathrm{PC})}$ 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 ($n = 10\,000$ for this study) refers to the number of emulator runs needed to estimate the sensitivity indices. The $n \times m$ matrix $\mathbf{Y}_{\mathrm{sample}}$ of the reconstructed m-dimensional outputs is computed using

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

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

2.6 Experimental setup

The sequence of tasks to complete when performing global sensitivity analysis is shown schematically in Fig. 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 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 $n = 10\,000$ 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.

If we are employing an emulator, the next stage is to 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 expensive simulator to validate the emulators. For this study, we ran the simulators with an additional set of inputs for validation. Comparing the emulator outputs with the simulator outputs using the validation inputs 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 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 Supplement (Fig. S2). For the emulator–PCA hybrid approach (Fig. 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, thus providing large computational savings.

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

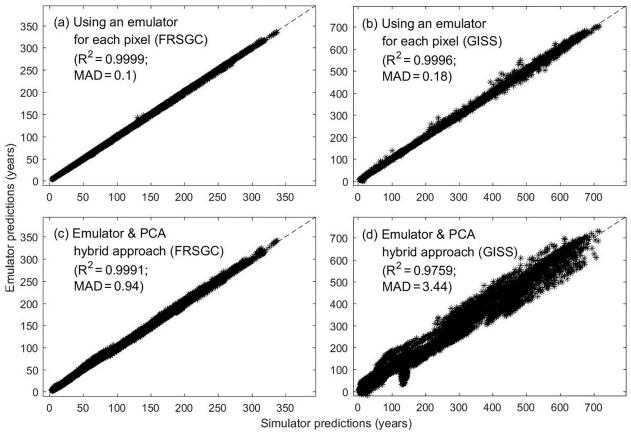

Figure 2. Annual column mean CH_4 lifetime calculated by the FRSGC and GISS chemistry models from each of 24 validation runs (x axis) versus that predicted by the emulator (y axis). In each plot, the R^2 and median absolute difference (MAD) are given as metrics for the accuracy of the emulator predictions.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

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 (Fig. 2a, b, $R^2 = 0.9996$–0.9999, median absolute difference of 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 (Fig. 2c, d, $R^2 = 0.9759$–0.9991, median absolute difference of 0.94–3.44 years) but was still sufficient for this study.

3.2 Comparison of sensitivity indices

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

Our results show that the GAM emulator-free GSA method produces very similar estimates of the SIs to the emulator-based methods (Figs. 3, 4; row a vs. c for Sobol versus GAM). The 95th and 99th percentiles of differences of the emulator-based methods (e.g. Sobol) versus GAM are 5 and 9 % for FRSGC, and 7 and 10 % for GISS (Fig. 5, M1 versus M3). For both models, the PLS non-emulator-based method produced SIs that were significantly different from those using the eFAST and Sobol methods (Figs. 3, 4; row a vs. d for Sobol vs. PLS). For FRSGC, the mean and 95th percentile of the differences in SIs for the Sobol versus PLS

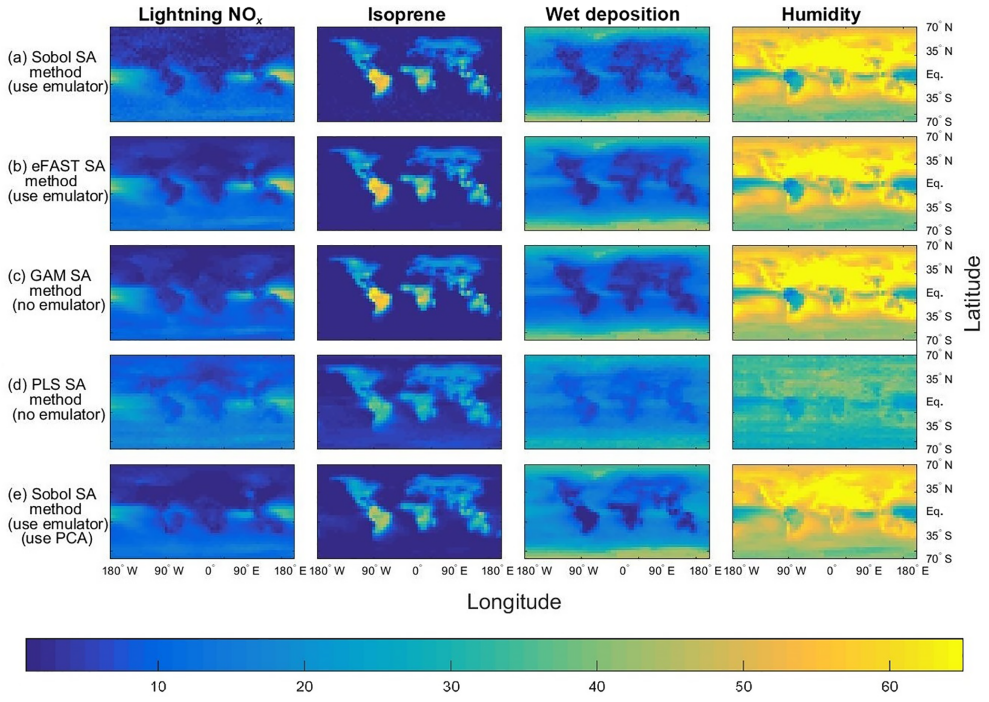

Figure 3. The sensitivity indices (percentage of the total variance in a given output) for the four dominant inputs, with the output given as the annual column mean CH_4 lifetime from the FRSGC chemistry transport model. The rows show the results from five different methods for performing sensitivity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

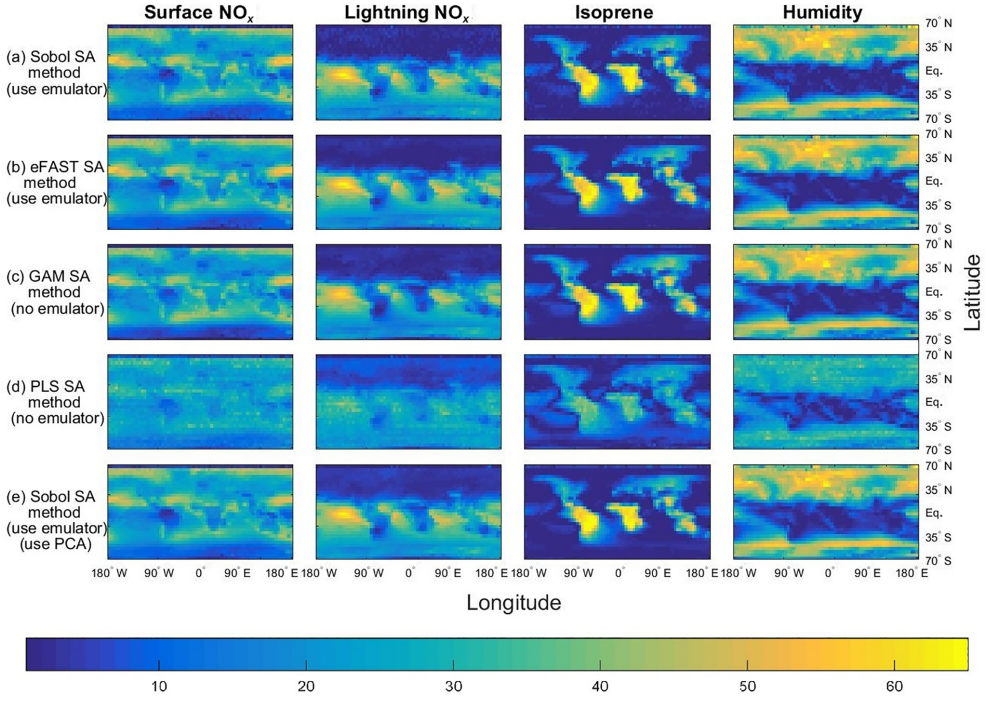

Figure 4. The sensitivity indices (percentage of the total variance in a given output) for the four dominant inputs, with the output given as the annual column mean CH_4 lifetime from the GISS chemistry model. See caption for Fig. 3 for further details about the five methods used.

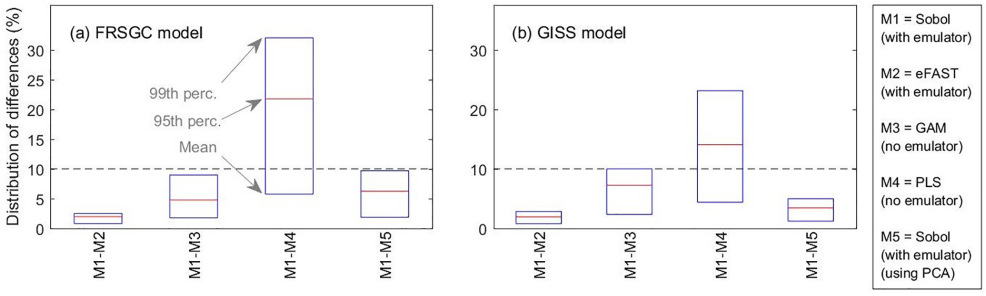

Figure 5. Statistics (mean, 95th percentile and 99th percentile) of the distribution of differences in sensitivity indices (SIs) between pairs of methods. For each comparison, the $\sim 16\,000$ pairs of SIs are made up of ~ 2000 pairs of SIs for each of the eight inputs.

methods are around 21 and 31 %, while for GISS the corresponding values are around 14 and 23 % (Fig. 5, M1 versus 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 (Figs. 3, 4; row a vs. e). The 95th and 99th percentiles of differences between the two approaches were 6 and 10 %, respectively, for FRSGC (Fig. 5a, M1 versus M5) and 3 and 5 %, respectively, for GISS (Fig. 5b, M1 versus M5). These are both higher than the corresponding values for the emulator-only methods (Fig. 5, M1 versus M2; < 2 and < 3 %, respectively). These higher values for the emulator–PCA hybrid approach are also reflected in the poorer estimates of the validation outputs using this approach versus the emulator-only approach (Fig. 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 simulator (or emulator) runs to compute the SIs. Saltelli et al. (2012) state 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 $\sim 10\,000$ 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 GAMs to perform GSA (Strong et al., 2015a, b, 2014), only Stanfill et al. (2015) have 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 (three to four 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, for others the resulting SIs may be substantially different from those 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 or 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 chemistry models, using 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 versus simulator scatter plot were noticeably larger than those of the FRSGC model (Fig. 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 either 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 $\sim 19\,000$ points in the output space. This emulator–PCA hybrid approach was also used to estimate the Sobol sensitivity indices corresponding to a flood forecasting model that simulates the water level of 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 sensitivity indices corresponded to the principal components, i.e. to a linear combination of the 170 output values. This is permissible if the principal components can be interpreted in some physical sense. For Lamboni et al. (2011), the first 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. (2018) who used the same training inputs and the same atmospheric chemical transport models (FRSGC and GISS) but different outputs. Instead of using highly multi-dimensional output of tropospheric methane lifetime values at different spatial locations, Wild et al. (2018) 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 NO_x and the lightning NO_x inputs were most important for predicting methane lifetime at the global scale, followed by the isoprene, the boundary layer mixing and the humidity inputs (Wild et al., 2018). 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 SI for GISS was very low at the global scale (SI of 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 computationally 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., 2018). For GSA studies when the number of inputs is small, our study has demonstrated that the GAM approach is a good 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 Conclusions

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 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 emulator–PCA hybrid approach is 19–28 times faster (or 24 times faster on aver-

age) at computing the sensitivity indices 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 and data availability. The R code to carry out global sensitivity analysis using the methods described in this paper is available in Sects. S2–S7 of the Supplement. This R code as well as the R code used to validate the emulators can also be found via https://doi.org/10.5281/zenodo.1038667 (Ryan, 2017).

The inputs and outputs of the FRSGC chemistry model that were used to train the emulators in this paper can be found via https://doi.org/10.5281/zenodo.1038670 (Ryan and Wild, 2017).

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

For the Sobol method, Saltelli (2002) and Tarantola et al. (2006) suggest using eight variants of Eq. (2), using different combinations of $\mathbf{y}_{\mathrm{A}}$, $\mathbf{y}_{\mathrm{B}}$, $\mathbf{y}_{\mathrm{Ci}}$ and $\mathbf{y}_{\mathrm{Di}}$:

$$\hat{S}_i^{\mathrm{I}} = \frac{\boldsymbol{Y}_{\mathrm{A}} \cdot \boldsymbol{Y}_{\mathrm{Ci}} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{B}}^{(j)}\right)}{\boldsymbol{Y}_{\mathrm{A}} \cdot \boldsymbol{Y}_{\mathrm{A}} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{B}}^{(j)}\right)}$$

$$\hat{S}_i^{II} = \frac{\boldsymbol{Y}_{\mathrm{B}} \cdot \boldsymbol{Y}_{\mathrm{Di}} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{B}}^{(j)}\right)}{\boldsymbol{Y}_{\mathrm{B}} \cdot \boldsymbol{Y}_{\mathrm{B}} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{B}}^{(j)}\right)}$$

$$\hat{S}_i^{\mathrm{III}} = \frac{\boldsymbol{Y}_{\mathrm{A}} \cdot \boldsymbol{Y}_{\mathrm{Ci}} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{B}}^{(j)}\right)}{\boldsymbol{Y}_{\mathrm{B}} \cdot \boldsymbol{Y}_{\mathrm{B}} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{B}}^{(j)}\right)}$$

$$\hat{S}_i^{\mathrm{IV}} = \frac{\boldsymbol{Y}_{\mathrm{A}} \cdot \boldsymbol{Y}_{\mathrm{Ci}} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{Ci}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{Di}}^{(j)}\right)}{\boldsymbol{Y}_{\mathrm{Ci}} \cdot \boldsymbol{Y}_{\mathrm{Ci}} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{Ci}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{Di}}^{(j)}\right)}$$

$$\hat{S}_i^{\mathrm{V}} = \frac{\boldsymbol{Y}_{\mathrm{A}} \cdot \boldsymbol{Y}_{\mathrm{Ci}} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathrm{Ci}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{Di}}^{(j)}\right)}{\boldsymbol{Y}_{\mathrm{Di}} \cdot \boldsymbol{Y}_{\mathrm{Di}} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{Ci}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{Di}}^{(j)}\right)}$$

$$\hat{S}_i^{\mathrm{VI}} = \frac{\boldsymbol{Y}_{\mathrm{B}} \cdot \boldsymbol{Y}_{\mathrm{Di}} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{B}}^{(j)}\right)}{\boldsymbol{Y}_{\mathrm{A}} \cdot \boldsymbol{Y}_{\mathrm{A}} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{A}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{B}}^{(j)}\right)}$$

$$\hat{S}_i^{\mathrm{VII}} = \frac{\boldsymbol{Y}_{\mathrm{B}} \cdot \boldsymbol{Y}_{\mathrm{Di}} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{Ci}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{Di}}^{(j)}\right)}{\boldsymbol{Y}_{\mathrm{Ci}} \cdot \boldsymbol{Y}_{\mathrm{Ci}} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{Ci}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{Di}}^{(j)}\right)}$$

$$\hat{S}_i^{\mathrm{VIII}} = \frac{\boldsymbol{Y}_{\mathrm{B}} \cdot \boldsymbol{Y}_{\mathrm{Di}} - \left(\frac{1}{N}\sum_{j=1}^{N} Y_{\mathrm{Ci}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{Di}}^{(j)}\right)}{\boldsymbol{Y}_{\mathrm{Di}} \cdot \boldsymbol{Y}_{\mathrm{Di}} - \left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{Ci}}^{(j)}\right)\left(\frac{1}{N}\sum_{j=1}^{N} \boldsymbol{Y}_{\mathrm{Di}}^{(j)}\right)}.$$

Thus, the ith first-order Sobol SI estimate is

$$\hat{S}_i = \frac{1}{8}\left(\hat{S}_i^{\mathrm{I}} + \hat{S}_i^{\mathrm{II}} + \hat{S}_i^{\mathrm{III}} + \hat{S}_i^{\mathrm{IV}} + \hat{S}_i^{\mathrm{V}} + \hat{S}_i^{\mathrm{VI}} + \hat{S}_i^{\mathrm{VII}} + \hat{S}_i^{\mathrm{VIII}}\right).$$

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:

$$\text{classical FAST} : G_i(z) = \overline{x}_i e^{\overline{v}_s z}, \quad (\overline{x}_i, \overline{v}_s \text{ are user-specified}) \tag{A1a}$$

$$\text{extended FAST} : G_i(z) = \frac{1}{2} + \frac{1}{\pi}\arcsin(z). \tag{A1b}$$

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

$$x_i(s) = \frac{1}{2} + \frac{1}{\pi}\omega_i s. \tag{A2}$$

Supplement. The supplement related to this article is available online at: https://doi.org/10.5194/gmd-11-3131-2018-supplement.

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 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.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. This work was supported by the Natural Environment Research Council (grant number NE/N003411/1).

Edited by: Andrea Stenke
Reviewed by: three anonymous referees

References

Ahtikoski, A., Heikkilä, J., Alenius, V., and Siren, M.: Economic viability of utilizing biomass energy from young stands – the case of Finland, Biomass Bioenerg., 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, https://doi.org/10.1371/journal.pone.0137004, 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, Math. Geol., 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 J. Chem. Phys., 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, J. Am. Stat. Assoc., 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, B. Math. Biol., 70, 1850–1868, 2008.

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, Struct. Multidis. Optim., 46, 457–469, 2012.

De Lozzo, M. and Marrel, A.: Sensitivity analysis with dependence and variance-based measures for spatio-temporal numerical simulators, Stoch. Environ. Res. Risk Assess., 31, 1437–1453, 2017.

Ferretti, F., Saltelli, A., and Tarantola, S.: Trends in sensitivity analysis practice in the last decade, Sci. Total Environ., 568, 666–670, https://doi.org/10.1016/j.scitotenv.2016.02.133, 2016.

Goldstein, M. and Rougier, J.: Bayes linear calibrated prediction for complex systems, J. Am. Stat. Assoc., 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 Sens., 8, 1–32, 2016.

Hankin, R. K.: Introducing BACCO, an R package for Bayesian analysis of computer code output, J. Stat. Softw., 14, 1–21, 2005.

Hill, T. C., Ryan, E., and Williams, M.: The use of CO_2 flux time series for parameter and carbon stock estimation in carbon cycle research, Global Change Biol., 18, 179–193, 2012.

Homma, T. and Saltelli, A.: Importance measures in global sensitivity analysis of nonlinear models, Reliab. Eng. Syst. Safe., 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, Operations Research/Computer Science Interfaces Series, Vol 59. Springer, Boston, MA, 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, J. Royal Stat. Soc. A , 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, Reliab. Eng. Syst. Safe., 96, 450–459, 2011.

Lee, L. A., Carslaw, K. S., Pringle, K. J., and Mann, G. W.: Mapping the uncertainty in global CCN using emulation, Atmos. Chem. Phys., 12, 9739–9751, https://doi.org/10.5194/acp-12-9739-2012, 2012.

Lee, L. A., Pringle, K. J., Reddington, C. L., Mann, G. W., Stier, P., Spracklen, D. V., Pierce, J. R., and Carslaw, K. S.: The magnitude and causes of uncertainty in global model simulations of cloud condensation nuclei, Atmos. Chem. Phys., 13, 8879–8914, https://doi.org/10.5194/acp-13-8879-2013, 2013.

Lilburne, L. and Tarantola, S.: Sensitivity analysis of spatial models, Int. J. Geogr. Inform. Sci., 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, Building Simulation, 1, 290–302, 2008.

Marrel, A., Iooss, B., Laurent, B., and Roustant, O.: Calculations of sobol indices for the gaussian process metamodel, Reliab. Eng. Syst. Safe., 94, 742–751, 2009.

Oakley, J. E. and O'Hagan, A.: Probabilistic sensitivity analysis of complex models: a Bayesian approach, J. Royal Stat. Soc. B, 66, 751–769, 2004.

O'Hagan, A.: Bayesian analysis of computer code outputs: a tutorial, Reliab. Eng. Syst. Safe., 91, 1290–1300, 2006.

Pistone, G. and Vicario, G.: Kriging prediction from a circular grid: application to wafer diffusion, Appl. Stoch. Models Business 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, Prog. Aerosp. Sci., 41, 1–28, 2005.

Rasmussen, C. E. and Williams, C. K. I.: Gaussian Processes for Machine Learning, 2006, the MIT Press, ISBN 026218253X, 2006.

Ripley, B. D.: Spatial statistics, John Wiley & Sons, Hoboken, New Jersey, 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, available at: https://hal.archives-ouvertes.fr/hal-00495766/document (last access: 15 June 2016), 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, Stoch. Environ. Res. Risk Assess., 2017, 1–19, 2017.

Ryan, E.: Fast sensitivity analysis methods for computationally expensive models with multi-dimensional output, https://doi.org/10.5281/zenodo.1038667, 2017.

Ryan, E. and Wild, O.: Data for the GSA methods paper by Ryan et al., https://doi.org/10.5281/zenodo.1038670, 2017.

Saltelli, A.: Making best use of model evaluations to compute sensitivity indices, Comput. Phys. Commun., 145, 280–297, 2002.

Saltelli, A., Andres, T., and Homma, T.: Sensitivity analysis of model output: an investigation of new techniques, Comput. Stat. Data Anal., 15, 211–238, 1993.

Saltelli, A. and Annoni, P.: How to avoid a perfunctory sensitivity analysis, Environ. Modell. Softw., 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, Chem. Rev., 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, J. Adv. Model. Earth Syst., 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, Clim. Dynam., 38, 2513–2542, 2012.

Shindell, D. T., Faluvegi, G., Unger, N., Aguilar, E., Schmidt, G. A., Koch, D. M., 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, Atmos. Chem. Phys., 6, 4427–4459, https://doi.org/10.5194/acp-6-4427-2006, 2006.

Sobie, E. A.: Parameter sensitivity analysis in electrophysiological models using multivariable regression, Biophys. J., 96, 1264–1274, 2009.

Sobol, I. Y. M.: On sensitivity estimation for nonlinear mathematical models, Matemat. Modeliro., 2, 112–118, 1990.

Stanfill, B., Mielenz, H., Clifford, D., and Thorburn, P.: Simple approach to emulating complex computer models for global sensitivity analysis, Environ. Modell. Softw., 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, ScHARR working paper, 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, Med. Decis. Mak., 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, Med. Decis. Mak., 35, 570–583, 2015b.

Tarantola, S., Gatelli, D., and Mara, T. A.: Random balance designs for the estimation of first order global sensitivity indices, Reliab. Eng. Syst. Safe., 91, 717–727, 2006.

Vanuytrecht, E., Raes, D., and Willems, P.: Global sensitivity analysis of yield output from the water productivity model, Environ. Modell. Softw., 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 Sens., 8, 673–699, 2016.

Voulgarakis, A., Naik, V., Lamarque, J.-F., Shindell, D. T., Young, P. J., Prather, M. J., Wild, O., Field, R. D., Bergmann, D., Cameron-Smith, P., Cionni, I., Collins, W. J., Dalsøren, S. B., Doherty, R. M., Eyring, V., Faluvegi, G., Folberth, G. A., Horowitz, L. W., Josse, B., MacKenzie, I. A., Nagashima, T., Plummer, D. A., Righi, M., Rumbold, S. T., Stevenson, D. S., Strode, S. A., Sudo, K., Szopa, S., and Zeng, G.: Analysis of present day and future OH and methane lifetime in the ACCMIP simulations, Atmos. Chem. Phys., 13, 2563–2587, https://doi.org/10.5194/acp-13-2563-2013, 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 B, 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, Atmos. Chem. Phys., 7, 2643–2660, https://doi.org/10.5194/acp-7-2643-2007, 2007.

Wild, O., Pochanart, P., and Akimoto, H.: Trans-Eurasian transport of ozone and its precursors, J. Geophys. Res.-Atmos., 109, D11302, https://doi.org/10.1029/2003JD004501, 2004.

Wild, O. and Prather, M. J.: Excitation of the primary tropospheric chemical mode in a global three-dimensional model, J. Geophys. Res., 105, 24647–24660, 2000.

Wild, O., Ryan, E., O'Connor, F., Vougarakis, A., and Lee, L.: Reducing Uncertainty in Model Budgets of Tropospheric Ozone and OH, Atmos.Chem. Phys., in preparation, 2018.

Wold, S., Sjöström, M., and Eriksson, L.: PLS-regression: a basic tool of chemometrics, Chemom. Intell. Labor. Syst., 58, 109–130, 2001.

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

Wu, J., Dhingra, R., Gambhir, M., and Remais, J. V.: Sensitivity analysis of infectious disease models: methods, advances and their application, J. Roy. Soc. Interf., 10, https://doi.org/10.1098/rsif.2012.1018, 2013.