The Impact of Uncertainty on Predictions of the CovidSim Epidemiological Code

Wouter Edeling[1], Hamid Arabnejad[2], Robbie Sinclair[3], Diana Suleimenova[2], Krishnakumar Gopalakrishnan[3], Bartosz Bosak[4], Derek Groen[2], Imran Mahmood[2], Daan Crommelin[1, 5], and Peter V Coveney[3, 6, *]

[1]Scientific Computing Group, Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
[2]Department of Computer Science, Brunel University London, London, UK
[3]Centre for Computational Science, University College London, London, UK
[4]Poznań Supercomputing and Networking Center, Poznań, Poland
[5]Korteweg-de Vries Institute for Mathematics, University of Amsterdam, Amsterdam, The Netherlands
[6]Informatics Institute, University of Amsterdam, Amsterdam, The Netherlands
[*]Corresponding author; email:p.v.coveney@ucl.ac.uk

January 19, 2021

Abstract

Epidemiological modelling has assisted in identifying interventions which reduce the impact of COVID-19. The UK government relied in part on the CovidSim model to guide its policy to contain the rapid spread of the COVID-19 pandemic during March and April 2020. However, CovidSim contains several sources of uncertainty which affect the quality of its predictions: parametric uncertainty, model structure uncertainty and scenario uncertainty. Here, we report on parametric sensitivity analysis and uncertainty quantification of the code. From the 940 parameters used as input to CovidSim, we find a subset of 19 to which the code output is most sensitive: imperfect knowledge in these inputs is magnified to the outputs by up to 300 %. The model displays significant bias with respect to observed data, failing to describe validation data well. Thus, quantifying parametric input uncertainty is not sufficient: the effect of model structure and scenario uncertainty must also be properly understood.

1 Introduction

CovidSim is an individual-based simulation code developed by the MRC Centre for Global Infectious Disease Analysis at Imperial College London. It is a modified version of an earlier model designed to support pandemic influenza planning [1], and has now been used to explore various non-pharmaceutical interventions (NPI) with the aim of reducing the transmission of the coronavirus, as documented in the key paper [2], denoted as 'Report 9'. CovidSim played an important role in the United Kingdom in reorienting UK Government policy from herd immunity to a strategy focused on suppression of the viral infection. It should be noted, however, that many competitor models exist. Notable examples include the work performed at the London School of Hygiene & Tropical Medicine; see for instance [3, 4], and especially [5], where the effect of different NPIs in the UK is modelled. Another noteworthy model is CovaSim [6], which is similar in structure to CovidSim, in the sense that it models a population of individuals via discrete 'agents'.

Likewise, CovidSim creates a network of individuals, located in areas defined by high-resolution population density data. In the model, contacts with other individuals can be made within four different place types, namely within the household, at schools, university and work places. It is possible to model a combination of different NPIs, namely general social distancing (SD), social distancing for those over 70 years of age (SDOL70), home isolation of suspected cases (CI), voluntary home quarantine (HQ) and place closure of universities and schools (PC); see Table 2 of Report 9 [2]. CovidSim contains over 900 input parameters, which are mainly located in two input files. In addition, a small number of parameters that define certain characteristics of the intervention scenario one wishes to study are supplied via the command line.

We have investigated the reproducibility of the code, as also done previously [7, 8]. That said, we especially focus on CovidSim's robustness under uncertainty in the input parameters. By 'robustness' in this context, we mean the extent to which the code amplifies uncertainties from the input to the output. Thus, our main aim is to take the model as given and examine the uncertainty in its predictions when its parameters are treated as random variables instead of deterministic inputs. We will use a dimension-adaptive sampling method for this purpose [9], in order to be able to handle the high-dimensional input space. This type of anisotropic sampling method adaptively exploits a possible low effective dimension, where only a subset of all inputs have a significant impact on the model output. A wide range of domains have seen the application of such dimension-adaptive samplers, e.g. computational electromagnetism [10], finance [11, 12] or natural convection problems [13], to name just a few. Here, we perform a validation study to examine the ability of the predicted output distribution to envelop the observed COVID-19 death count, conditional on a predefined intervention scenario.

Due to the large number of inputs, one cannot hope to obtain an accurate, data-informed value of all parameters in play. Moreover, considering CovidSim's influential status, and its likely use in future COVID-19 predictions, it is important to assess the impact of parametric uncertainty on the model output. We will argue the case for the prediction of uncertainty in high-impact decision making, after we first describe our results.

2 Results

We have performed an analysis on the original closed-source version of the code. However, the majority of our sensitivity analysis and uncertainty quantification efforts lie with the current updated open-source release of CovidSim.

With respect to the original version, we have been able to achieve exact reproducibility of the results in the influential Report 9 [2], although only when running within an Azure cloud environment. Other authors have also found CovidSim to be reproducible [8, 7]. Attempts to run the code on a Linux machine failed; we could not reproduce the same results here and since this version is no longer supported it was not investigated further.

2.1 Uncertainty in CovidSim

The predictions of most computational models are affected by uncertainty from a variety of different sources. We identify the following three sources of uncertainty in CovidSim, namely parametric uncertainty, model structure uncertainty and scenario uncertainty. This breakdown is not uncommon; see e.g. [14, 15, 16].

Parametric uncertainty arises due to imperfect knowledge of the model input parameters $\boldsymbol{\xi} \in \mathbb{R}^d$, described in Section 2.1.2. Model structure uncertainty is more fundamental, as it relates to uncertainty about the appropriate mathematical structure of the model, denoted by $\mathcal{M}$. One can think of missing epidemiological processes that are not implemented in CovidSim; see the discussion the Supplementary Section 6. Finally, a scenario $\mathcal{S}$ is the set of conditions under which a model $\mathcal{M}(\boldsymbol{\xi})$ is applied. In the case of CovidSim, $\mathcal{S}$ includes the choice of NPI scenarios, the initialisation of the model, and the well-known reproduction number R_0. Note that the actual implementation of $\mathcal{S}$ will be parameterised as well, and that we could technically lump these parameters in with $\boldsymbol{\xi}$. However, the scenario parameters are of a different nature than the 'internal' inputs $\boldsymbol{\xi}$, and treating $\mathcal{S}$ as a separate category mirrors the way in which the results were presented in Report 9, which showed results for different NPIs and R_0 values.

Thus, if we denote q as the predicted output quantity of interest, we have $q = q(\boldsymbol{\xi}, \mathcal{M}, \mathcal{S})$, where all three arguments are uncertain. As noted, our main goal is to quantify the impact of parametric uncertainty. By treating the inputs as random variables with probability density function $p(\boldsymbol{\xi})$, our mean prediction is given by

$$\mathbb{E}[q \mid \mathcal{M}, \mathcal{S}] := \int_{\Omega_{\boldsymbol{\xi}}} q(\boldsymbol{\xi}, \mathcal{M}, \mathcal{S})\, p(\boldsymbol{\xi})\, \mathrm{d}\boldsymbol{\xi}, \tag{1}$$

where $\Omega_{\boldsymbol{\xi}}$ is the support of $p(\boldsymbol{\xi})$. The uncertainty in the prediction (1) can be represented by either the corresponding variance or confidence intervals. It is important to note that our results are conditional on $\mathcal{M}$ and $\mathcal{S}$. We are not in a position to change the former, and we illustrate the importance of scenario uncertainty by repeating the parametric uncertainty analysis for two different scenarios.

2.1.1 Uncertainty propagation

We use EasyVVUQ [17, 18] from the Verified Exascale Computing for Multiscale Applications (VECMA) toolkit [19], for propagating the input uncertainties through CovidSim. To interface CovidSim with EasyVVUQ, templates from the CovidSim input files are generated. In the process, a single file is generated, which contains all inputs, with their types and default values specified. Simply counting the number of entries in this file allows us to exactly determine the number of parameters present in the code, which is how we arrived at a number of 940 inputs.

We will not vary all 940 parameters (see Section 2.1.2). Instead we will assign to d ($d \ll 940$) input parameters ξ_i an independent probability density function (PDF), i.e. $\xi_i \sim p(\xi_i)$. Next, a d-dimensional sampling plan from the joint PDF $\prod_i p(\xi_i)$ is created, after which CovidSim is evaluated at each input point. To handle the high-dimensional input space, we refine the sampling plan in a dimension-adaptive manner, the details of which can be found in Section 4.1.

2.1.2 CovidSim parameters

In this section we describe how we arrived at our selection of the input parameters that we vary as part of our uncertainty quantification study. We have divided the parameters present in the input files into three groups:

(i) Group 1: intervention parameters, these are parameters meant to slow down the viral infection, which can still be varied for a fixed $\mathcal{S}$, e.g. "Length of time households are quarantined.", when HQ is part of the selected NPI set.

(ii) Group 2: disease parameters, related to the characteristics of COVID-19, e.g. "Latent period."

(iii) Group 3: spatial / geographic parameters, parameters which apply to the properties of the network, e.g. "Relative transmission rate for place types."

The purpose of this classification was to direct initial, exploratory uncertainty quantification (UQ) and sensitivity analysis (SA) campaigns on a coherent subset of parameters. The final UQ campaign, contains parameters from all three groups. By a 'campaign' we mean a single forward propagation step of uncertainty from the input to the output.

Before starting the UQ analysis, we first performed a parameter study, using in part expert domain knowledge from the CovidSim team at Imperial College, to reduce the number of inputs. We focus on a scenario $\mathcal{S}$ based on the suppression release in the Report 9 folder on GitHub [20], using the intervention setting that combines PC, CI, HQ, and SD, as this class is the closest to actual NPIs that were implemented in the UK. Let us reiterate that these abbreviations correspond to place closure of schools and universities (PC), home isolation of suspected cases (CI), voluntary home quarantine (HQ) and general social distancing (SD). In this case, we have the aforementioned total of 940 parameters. Note that some input parameters are vectors, in which case we counted each entry as a separate input parameter. On top of our own initial selection, over the course of our analysis, we have received feedback from the developers of CovidSim as to the inclusion of given parameters in the UQ study.

Many of the parameters are (currently) not used in the case of COVID-19 simulation, such as numerous vaccination parameters. For more details, see the 'parameter list' Supplementary Data file for the full list with all input parameters, their default values and reasons for inclusion or exclusion from the Imperial College CovidSim team. In addition, this list contains a short description of the parameters.

Although we made our own considerations and decisions as to which parameters to include in the UQ study, the large number of parameters in play requires expert knowledge to make a suitable initial selection. A total of 60 of these parameters were included in a UQ campaign at some point. These are displayed separately in the parameter list Supplementary Data file. We choose uninformative uniform distributions to reflect our lack of knowledge in the most-likely values of these inputs, with bounds based on either data or expert knowledge.

Any input that was selected at least once for refinement during the dimension-adaptive sampling in one of the three exploratory UQ campaigns of Section 2.1.2, was included in the final, large-scale UQ campaign. This led to a total of 19 final parameter distributions; see Section 3 of the Supplementary Section.

Important scenario parameters are the reproduction number R_0, and two 'trigger' parameters, which are specified via the command line. In the case of modelling a suppression strategy, the SD and PC interventions are triggered when the weekly number of new intensive care unit (ICU) cases exceeds the value supplied by the first trigger. Likewise, they are suspended when this metric drops below the second specified trigger [2]. The results below are conditional on the selected NPI measures, as well as fixed values for R_0 and the ICU triggers.

2.2 Confidence intervals

Here, we consider two different PC_CI_HQ_SD suppression scenarios. The results that follow were obtained using a computational budget of 3000 CovidSim evaluations per scenario. Figure 1a shows the 68% and 95% confidence intervals of the cumulative death prediction, for scenario $\mathcal{S}_1$ with $R_0 = 2.4$, and on/off ICU triggers of 60/15. Remember the PC and SD interventions are turned on and off based on a specified number of new weekly ICU cases, i.e. 60 and 15 new cases here, which is one of the scenarios considered in Report 9. In addition, the PDF of the total death count after 800 days is plotted. The latter shows clear non-Gaussian behaviour, with a heavy tail towards a higher death count. The corresponding Report 9 total death count is 8700 [2]. The current version, which now supports averaging over stochastic realizations, predicts 9500 [21]. Our mean prediction from Eqn. (1) is almost double this amount. The Report 9 predictions are still captured by the distribution (at approximately the lower boundary of the 68% confidence interval), but the distribution also supports low-probability events which are about 5-6 times higher than those given in Report 9.

Note that the output distribution conditioned on $\mathcal{S}_1$ clearly under predicts the observed death count in the UK, which is also plotted in Figure 1a. We therefore selected $\mathcal{S}_2$ using parameters from Report 9 that gave the highest predicted mortality, i.e. $R_0 = 2.6$ and ICU on/off triggers of 400/300 cases. The results are displayed in Figure 1b. The deterministic Report 9 prediction is still located at the 68% confidence interval lower border. However, the total death count PDF is notably less skewed, although still not exactly symmetric.

Figure 1 clearly indicates that the results are very sensitive to $\mathcal{S}$. As noted, we also plot the observed death count validation data from [22] in both sub figures, which are clearly not captured well by the output distributions, although scenario $\mathcal{S}_2$ does perform better than $\mathcal{S}_1$. It is also clear from Figure 1 that, in both cases, the rate of infection starts too slowly, and that it must be assumed that the epidemic started earlier than suggested in Report 9, which is in line with the findings of [7]. Hence, if one aims to validate CovidSim in a probabilistic sense (i.e. obtaining a distribution which captures validation data with high probability), it is crucial to either tune the scenario parameters, or to quantify the scenario uncertainty.

CovidSim also has a number of random seeds, whose influence on the death count is examined in the Supplementary Section 5. For confidence intervals on quantities of interest other than the cumulative death count, see the Supplementary Section 7.

Finally, we emphasize that the authors of Report 9 did not claim that their parametrization at the time would be able to match the death count data of the coming months. The main message was that it would "be necessary to layer multiple interventions, regardless of whether suppression or mitigation is the overarching policy goal" [2], and it also showed that doing nothing at all would have disastrous consequences.

2.3 Sensitivity analysis

With sensitivity analysis, the aim is to apportion the uncertainty of the model output to specific (combinations of) input parameter uncertainties. To this end, Sobol indices measure the fraction of the output variance that each combination of input parameters is responsible for, when given a distribution on the inputs [23]. They can be computed in a post-processing step, once the input uncertainties are propagated through the computational model [24]; see Section 4.1.3.

The first-order Sobol indices S_i are defined as $S_i := \mathbb{V}[q_i] / \mathbb{V}[q] \in [0, 1]$, for $i = 1, \cdots, d$. Here $\mathbb{V}[q]$ is the total output variance, and $\mathbb{V}[q_i]$ is the partial variance attributed to one particular input parameter [23]. Figure 2 displays the three S_i with the highest values for scenarios $\mathcal{S}_1$ and $\mathcal{S}_2$; see the Supplementary Section 8 for more results. The Sobol indices are plotted against time, which shows that in the beginning, the latent period is the most influential, although only for a short amount of time. The latent period is the period in which a patient is infected but not yet infectious. A longer latent period therefore means that the rate of disease spread is slower in this early exponential growth stage, when there are still relatively few cases present.

The second important parameter is the 'Relative spatial contact rate given social distancing' parameter, which indicates the assumed effectiveness of social distancing. Finally, the third parameter (in both scenarios), to dominate the variance is the 'Delay to start case isolation'. The latent period originally belonged to the Disease parameter group of Section 2.1.2, whereas the other two inputs are intervention parameters. Overall, it can be said that the intervention parameters, which influence control measures and human behaviour, are most influential. The inputs from the spatial/geographic group have a comparatively small effect.

In Figure 2 we also plot the sum of all 19 first-order Sobol indices. This shows that first-order effects, i.e. the fraction of the variance obtained by varying individual parameters, account for a little under 80% in the case of $\mathcal{S}_1$, and roughly 90% of the variance for $\mathcal{S}_2$. Conversely, interaction effects between parameters therefore account for no more than 10-20% in our chosen scenarios. We also show the sum of the first-order indices for just the three most important parameters, i.e. those actually plotted in Figure 2, which already accounts for roughly 50% and 67% of the observed variance in cumulative deaths for $\mathcal{S}_1$ and $\mathcal{S}_2$ respectively.

2.4 Uncertainty amplification

Although we based our input distributions (see Supplementary Table 1) on a combination of available data and expert knowledge, (in general) a certain level of ambiguity remains with respect to the choice of input distribution. As explained later, we therefore devise a measure that examines the *amplification* of uncertainty in the outputs, with respect to a given set of input distributions. This relative measure of output-to-input variability is based on the coefficients of variation ratio (CVR), and is given by

$$CVR := CV\left(\bar{q}\right)/CV\left(\bar{\boldsymbol{\xi}}\right) = \left(\frac{1}{N}\sum_{n=1}^{N}\frac{\sigma_{q_n}}{\mu_{q_n}}\right) \Bigg/ \left(\frac{1}{d}\sum_{i=1}^{d}\frac{\sigma_{\xi_i}}{\mu_{\xi_i}}\right). \tag{2}$$

A coefficient of variation is a dimensionless quantity that measures the variability of a random variable with respect to its mean, and is defined as the standard deviation over the mean (σ/μ). In (2), $CV(\bar{q})$ and $CV(\bar{\boldsymbol{\xi}})$ are the mean coefficients of variation of the output $q \in \mathbb{R}^N$ and input $\boldsymbol{\xi} \in \mathbb{R}^d$, respectively. The results for CovidSim using (2) are displayed in Table 1, which shows that the uncertainty in the input is amplified by a factor of 3 for scenario 1. In contrast, CovidSim is more robust under $\mathcal{S}_2$, in which case the same input uncertainty is still amplified to the output, although now by a factor of 2.

Note that $\mathcal{S}_1$ has a higher CVR, while it imposes a stronger control compared to $\mathcal{S}_2$. Even though the stronger control results in a much lower absolute number of predicted deaths, the output is more uncertain in a relative sense due to the long tail (see Figure 1a), which results in a higher output CV and therefore a higher CVR.

3 Discussion

Conditional on a given scenario $\mathcal{S}$, we found that the Report 9 predictions are captured by the parametric uncertainty at the lower bound of the 68% confidence interval. The PDF of the total death count is skewed, and can support low probability events with a predicted death count that is about 5 to 6 times higher.

We find that CovidSim amplifies the input uncertainty, to the extent of 300% (i.e. roughly by a factor of 3, see Table 1), depending on the chosen NPI scenario. Despite this amplification of uncertainty, the distribution of the output does not envelope available validation data well for the two scenarios we considered. We do note, however, that the predictions will be very sensitive to the chosen scenario $\mathcal{S}$, which therefore must be tuned if one wishes to validate CovidSim against available data; see e.g. [7]. Tuning the ICU triggers alone is not sufficient. In the Supplementary Section 4, we show the results of an additional UQ campaign where we sought to extract the 'best-guess' ICU trigger values from data. These results are similar to those presented in the main manuscript.

Predicting the uncertainty in computational models is already considered as vitally important in weather and climate models. For instance, the author of [25] claims that "no weather or climate prediction can be considered complete without a forecast of the associated flow-dependent predictability". We also argue that in the case of COVID-19 predictions, a single deterministic prediction paints an incomplete picture, as we showed that such a prediction is better viewed as only one member of a much wider distribution. Hence, some measure of uncertainty is required for a correct interpretation of the results, so that those tasked with policy making are presented with a more complete picture of the outcomes that the model is capable of predicting.

For instance, if the policy maker is presented with just the deterministic model outcome of Figure 1b, he or she may draw the conclusion that by adopting scenario $\mathcal{S}_2$, the UK will suffer 50.000 deaths after approximately 600 days. However, by taking some reasonable input uncertainty into account, we see that the same model, with the same NPI settings, can also predict that number in less than 200 days. Another example concerns predictions with hard thresholds (such as the maximum number of available ICU beds). A single prediction might lie on the safe side of the threshold, yet the model may hold a significant non-zero probability that this threshold can be exceeded, if it were admitted that the models are uncertain. We expect that such kinds of information pertaining to uncertainty would influence the decision-making process in a significant way.

Let us briefly discuss applying the proposed method to models other than CovidSim, which may well be beneficial for the same reasons mentioned above. The dimension-adaptive sampling scheme has a black-box assumption, and can therefore be applied without modification to other models. However, note that EasyVVUQ requires that a template for the input file must be created [17]. To execute the ensembles on a supercomputer (in our case the PSNC Eagle machine), we used the FabSim3 automation toolkit [26]; see the Code Availability section for the relevant links to our software. In summary, (dimension-adaptive) parametric uncertainty propagation is general enough to be applied to other models, and it is important to do so moving forward. That said, although the dimension-adaptive approach is efficient, it is ultimately still limited by the dimension of the input space. We could not have applied our method to all inputs of CovidSim for example.

To conclude, to retrofit the model's outputs with the observed data requires additional *post-hoc* tuning of certain parameters that control the scenario in which the model is applied. These issues need to be addressed in seeking to provide a more quantitative albeit strongly probabilistic version of the code which might be suitable for its future application in healthcare and governmental decision-making. Our findings exemplify how sensitivity analysis and uncertainty quantification can help improve model development efforts, and in this case support the creation of epidemiological forecasting with quantified uncertainty.

As an alternative to retrofitting the scenario parameters, one could attempt to quantify the uncertainty related to the scenario the model is applied in. One such potential route for future research could involve creating cheap surrogate models for CovidSim, e.g. in the stochastic space of the most influential parameters identified, which opens up the possibility of Bayesian inference [27]. This will allow us to update our assumptions on the input distributions, and obtain posterior input distributions conditioned on observed data instead. In addition, such a statistical calibration can eliminate a bias between the mean prediction and real-world observations. Repeating the procedure for a discrete set of scenario parameters then allows for the combined estimation of the parametric and the scenario uncertainty using Bayesian ensemble methods; see e.g. [15, 16].

Acknowledgements

This project was undertaken on behalf of the Royal Society's RAMP initiative. The work was funded as part of the European Union Horizon 2020 research and innovation programme undergrant agreements #800925 (VECMA project, www.vecma.eu) and #823712 (CompBioMed2 Centre of Excellence, www.compbiomed.eu), as well as the UK EPSRC for the UK High-End Computing Consortium (EP/R029598/1). We are grateful to Mike Cates, Graeme Ackland and Ken Rice for helpful discussions while this work was in progress. Finally, we would like to thank Prof. Neil Ferguson for his support of our investigation and his response to our questions. The calculations were performed in part at the Poznan Supercomputing and Networking Center.

Author contributions

WE: Performed much of the analysis, carried out many of the computational campaigns and co-wrote the paper. HA, RS, DS: Performed sensitivity analysis on sub-groups of the parameters within the model. BB: Provided support for the computational aspects of the work performed at PSNC. DG, IM, DC: Provided scientific and technical advice. PVC: Initiated and directed the research and co-wrote the paper.

Competing interests

The authors declare no competing interests.

Corresponding author

Author to whom correspondence and requests for materials should be addressed: Professor P V Coveney.

References

[1] N.M. Ferguson, D.A.T. Cummings, C. Fraser, J.C. Cajka, P.C. Cooley, and D.S. Burke. Strategies for mitigating an influenza pandemic. *Nature*, 442(7101):448–452, 2006.

[2] N. Ferguson, D. Laydon, G. Nedjati-Gilani, N. Imai, K. Ainslie, M. Baguelin, S. Bhatia, A. Boonyasiri, Z. Cucunubá Perez, G. Cuomo-Dannenburg, et al. Report 9: Impact of non-pharmaceutical interventions (NPIs) to reduce COVID19 mortality and healthcare demand, `https://www.imperial.ac.uk/mrc-global-infectious-disease-analysis/covid-19/report-9-impact-of-npis-on-covid-19/`, 2020.

[3] A.J. Kucharski, T.W. Russell, C. Diamond, Y. Liu, J. Edmunds, S. Funk, R.M. Eggo, et al. Early dynamics of transmission and control of COVID-19: a mathematical modelling study. *The Lancet. Infectious Diseases*, 20(5):553, 2020.

[4] A. Clark, M. Jit, C. Warren-Gash, B. Guthrie, H.H.X. Wang, S.W. Mercer, C. Sanderson, M. McKee, C. Troeger, K.L. Ong, et al. Global, regional, and national estimates of the population at increased risk of severe COVID-19 due to underlying health conditions in 2020: a modelling study. *The Lancet Global Health*, 2020.

[5] N.G. Davies, A.J. Kucharski, R.M. Eggo, A. Gimma, W.J. Edmunds, T. Jombart, K. O'Reilly, A. Endo, J. Hellewell, E.S. Nightingale, et al. Effects of non-pharmaceutical interventions on COVID-19 cases, deaths, and demand for hospital services in the UK: a modelling study. *The Lancet Public Health*, 2020.

[6] C.C. Kerr, Robyn M. Stuart, D. Mistry, R.G. Abeysuriya, G. Hart, K. Rosenfeld, P. Selvaraj, R.C. Nunez, B. Hagedorn, L. George, et al. Covasim: an agent-based model of COVID-19 dynamics and interventions. *medRxiv*, 2020.

[7] K. Rice, B. Wynne, V. Martin, and G.J. Ackland. Effect of school closures on mortality from coronavirus disease 2019: old and new predictions. *British Medical Journal*, 371, 2020.

[8] S. Eglen. Codecheck certificate 2020-010. `https://zenodo.org/record/3865491#.XuPW_y-ZPGI`.

[9] T. Gerstner and M. Griebel. Dimension–adaptive tensor–product quadrature. *Computing*, 71(1):65–87, 2003.

[10] D. Loukrezis, U. Römer, and H. De Gersem. Assessing the performance of Leja and Clenshaw-Curtis collocation for computational electromagnetics with random input data. *International Journal for Uncertainty Quantification*, 9(1), 2019.

[11] K.L. Judd, L. Maliar, S. Maliar, and R. Valero. Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain. *Journal of Economic Dynamics and Control*, 44:92–123, 2014.

[12] M. Griebel and M. Holtz. Dimension-wise integration of high-dimensional functions with applications to finance. *Journal of Complexity*, 26(5):455–489, 2010.

[13] B. Ganapathysubramanian and N. Zabaras. Sparse grid collocation schemes for stochastic natural convection problems. *Journal of Computational Physics*, 225(1):652–685, 2007.

[14] D. Draper. Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):45–70, 1995.

[15] P.D. Meyer, M. Ye, M.L. Rockhold, S.P. Neuman, and K.J. Cantrell. Combined estimation of hydrogeologic conceptual model, parameter, and scenario uncertainty with application to uranium transport at the Hanford Site 300 Area. Technical report, Pacific Northwest National Lab.(PNNL), Richland, WA (United States), 2007.

[16] W.N. Edeling, P. Cinnella, and R.P. Dwight. Predictive RANS simulations via Bayesian Model-Scenario Averaging. *Journal of Computational Physics*, 275:65–91, 2014.

[17] R.A. Richardson, D.W. Wright, W. Edeling, V. Jancauskas, J. Lakhlili, and P.V. Coveney. EasyVVUQ: A Library for Verification, Validation and Uncertainty Quantification in High Performance Computing. *Journal of Open Research Software*, 8(1), 2020.

[18] D.W. Wright, R.A. Richardson, W.N. Edeling, J. Lakhlili, R.C. Sinclair, V. Jancauskas, D. Suleimenova, B. Bosak, M. Kulczewski, T. Piontek, P. Kopta, I. Chirca, H. Arabnejad, O.O. Luk, O. Hoenen, J. Węglarz, D. Crommelin, D. Groen, and P.V. Coveney. Building Confidence in Simulation: Applications of EasyVVUQ. *Advanced Theory and Simulations*, 3(8), 2020.

[19] D. Groen, R.A. Richardson, D.W. Wright, V. Jancauskas, R. Sinclair, P. Karlshoefer, M. Vassaux, H. Arabnejad, T. Piontek, P. Kopta, et al. Introducing VECMAtk-verification, validation and uncertainty quantification for multiscale and HPC simulations. In *International Conference on Computational Science*, pages 479–492. Springer.

[20] MRC Centre for Global Infectious Disease Analysis. COVID-19 CovidSim Model - Report 9 folder. `https://github.com/mrc-ide/covid-sim/tree/master/report9`.

Figure 1: **Distribution of cumulative death predictions.** **a**: The mean cumulative death prediction for scenario $\mathcal{S}_1$ ($R_0 = 2.4$, ICU on/off triggers 60/15) and **b**: $\mathcal{S}_2$ ($R_0 = 2.6$, ICU on/off triggers 400/300), plus confidence intervals (CI), and at the right of each figure, the PDF of the total death count after 800 days. Day 0 corresponds to January 1st, 2020. In addition, we plot the observed cumulative death count data for the UK (squares) in both figures, obtained from [22]. The striped line is a single sample from CovidSim (current release), run with the baseline parameter values of Report 9.

Figure 2: **Fraction of variance from the model parameters.** The first-order Sobol indices for two scenarios (**a**: $\mathcal{S}_1$, **b**: $\mathcal{S}_2$), of the same 3 most dominant parameters, plotted against time at one month intervals. It shows the fraction of the variance that each parameter is responsible for, over time. In addition, we show the sum of all 19 first-order indices (stars). The sum of the 3 most dominant parameters is also shown (diamonds).

[21] Office for National Statistics. Deaths involving COVID-19 in the care sector, England and Wales Deaths involving COVID-19 in the care sector, England and Wales: deaths occurring up to 12 June 2020 and registered up to 20 June 2020 (provisional). `https://www.ons.gov.uk/peoplepopulationandcommunity/birthsdeathsandmarriages/deaths/articles/deathsinvolvingcovid19inthecaresectorenglandandwales/latest`.

[22] GOV.UK Coronavirus (COVID19) in the UK. Deaths in United Kingdom. `https://coronavirus-staging.data.gov.uk/deaths`.

[23] I.M. Sobol. On sensitivity estimation for nonlinear mathematical models. *Matematicheskoe Modelirovanie*, 2(1):112–118, 1990.

[24] J.D. Jakeman, M.S. Eldred, G. Geraci, and A. Gorodetsky. Adaptive multi-index collocation for uncertainty quantification and sensitivity analysis. *International Journal for Numerical Methods in Engineering*, 121(6):1314–1343, 2020.

[25] T.N. Palmer. Predicting uncertainty in forecasts of weather and climate. *Reports on progress in Physics*, 63(2):71, 2000.

[26] D. Groen, A.P. Bhati, J. Suter, J. Hetherington, S.J. Zasada, and P.V. Coveney. FabSim: facilitating computational research through automation on large-scale and distributed e-infrastructures. *Computer Physics Communications*, 207:375–385, 2016.

[27] Y. Marzouk and D. Xiu. A stochastic collocation approach to Bayesian inference in inverse problems. *Communications in Computational Physics*, 6(4):826–847, 2009.

scenario	$CV\left(\bar{\boldsymbol{\xi}}\right)$	$CV\left(\bar{q}\right)$	CVR
$\mathcal{S}_1$	0.1950	0.6097	3.13
$\mathcal{S}_2$	0.1950	0.3872	1.99

Table 1: The mean coefficients of variation for the input ($CV\left(\bar{\boldsymbol{\xi}}\right)$) and output ($CV\left(\bar{q}\right)$), and the ratio of the two (CVR), which serves as our robustness score.

4 Methods

In this section, we first describe our method for computing the statistical results. Following that, the uncertainty amplification factor is described.

4.1 Statistics

Here we describe how we compute the probability distribution of the code output, the corresponding ensemble execution, and how the Sobol indices are calculated.

4.1.1 Dimension-adaptive uncertainty propagation

The traditional forward uncertainty quantification methods present in EasyVVUQ (e.g. stochastic collocation and polynomial chaos), are subject to the curse of dimensionality. To illustrate the problem, consider first the standard stochastic collocation (SC) method, which creates a polynomial approximation of the code output q, as a function of the uncertain inputs $\boldsymbol{\xi} = (\xi_d, \cdots, \xi_d) \in \mathbb{R}^d$:

$$q(\boldsymbol{\xi}) \approx \tilde{q}(\boldsymbol{\xi}) = \sum_{j_1=1}^{m_1} \cdots \sum_{j_d=1}^{m_d} q(\xi_{j_1}, \cdots, \xi_{j_d})\; a_{j_1}(\xi_1) \otimes \cdots \otimes a_{j_d}(\xi_d) \tag{3}$$

Here, $\tilde{q}$ denotes the polynomial approximation of q, and $q(\xi_{j_1}, \cdots, \xi_{j_d})$ is the actual code output, evaluated at some location inside the stochastic domain of $\boldsymbol{\xi} \in \mathbb{R}^d$. Each input $\xi_i \in \boldsymbol{\xi}$ is assigned an independent probability density function $p(\xi_i)$, and the goal is to propagate these through CovidSim in order to examine the corresponding distribution of the output q. The basic building blocks for the SC method are one-dimensional quadrature and interpolation rules, which are extended to higher dimension through a tensor-product construction. In (3), $a_{j_1}(\xi_1) \otimes \cdots \otimes a_{j_d}(\xi_d)$ is the tensor product of one-dimensional Lagrange interpolation polynomials, used to interpolate the code outputs $q(\xi_{j_1}, \cdots, \xi_{j_d})$ to a (potentially) unsampled location $\boldsymbol{\xi}$. For instance, unlike the Monte Carlo method, the sample locations $(\xi_{j_1}, \cdots, \xi_{j_d})$ are not random. Instead, each ξ_{j_i} is a point drawn from a one-dimensional quadrature rule, used to approximate integrals weighted by the chosen input distribution $p(\xi_i)$. The order of the quadrature rule for the i-th input determines the number of points m_i, and due to the tensor product construction the total number of code evaluations for d inputs equals

$M = m_1 \cdot m_2 \cdots m_d$, or $M = m^d$ if all inputs receive the same quadrature order (see Supplementary Figure 1 for an example). The exponential increase with d, known as the curse of dimensionality, renders the SC method intractable beyond $d \sim 10$. Hence, even though our parameter analysis in the main article indicates that only roughly 6% of the inputs will be varied at some point, due to the large number of inputs this is far too much for such 'brute force' UQ methods.

Therefore, a dimension-adaptive version of the Stochastic Collocation (SC) method, based on the work of [28, 9], has been implemented in EasyVVUQ. It is reasonable to expect that the output q will not be equally sensitive to each input ξ_i. Hence, even though our input space is d-dimensional, a dimension-adaptive approach banks on the existence of a lower *effective* dimension. The basic idea is to start with a 0-th order quadrature rule for all inputs, and to adaptively rank order the inputs, keeping all ineffective inputs at a low (possible 0-th) order, while increasing the order of those that are effective, see again Supplementary Figure 1 for an example in two dimensions.

The dimension-adaptive approach is explained in detail in [9], here we only provide a general outline. Let Λ be the set containing all selected quadrature-order multi indices (the gray squares of Supplementary Figure 1), which is initialized as $\Lambda := \{(0, , \cdots, 0)\}$. Let the *forward neighbours* of any multi index l be defined by the set $\{l + e_i \mid 1 \leq i \leq d\}$, where e_i is the elementary basis vector in the i-th direction, e.g. $e_2 = (0, 1, 0, \cdots, 0)$. The forward neighbours of the set Λ are then the forward neighbours for all $l \in \Lambda$, which are not already in Λ. Similarly, the *backward neighbours* of l are given by $\{l - e_i \mid l_i > 0, 1 \leq i \leq d\}$. An index set Λ is said to be *admissible* if all backward neighbours of Λ are in Λ.

To adaptively refine the sampling plan , a 'look-ahead step' [29] is executed, where the computational model is evaluated at the new unique sample locations generated by those forward neighbours l where $\Lambda \cup \{l\}$ remains an admissible set, corresponding to the × symbols of Supplementary Figure 1. For each admissible forward neighbour l, a local error measure is computed. As proposed in [10], we will base our error measure on the so-called hierarchical surplus, defined as the difference between the code output q and the surrogate prediction $\tilde{q}$, evaluated at new sample locations of an admissible forward neighbour l,

$$s\left(\boldsymbol{\xi}_j^{(l)}\right) := q\left(\boldsymbol{\xi}_j^{(l)}\right) - \tilde{q}_\Lambda\left(\boldsymbol{\xi}_j^{(l)}\right), \quad \boldsymbol{\xi}_j^{(l)} \in X_l \backslash X_\Lambda. \tag{4}$$

Here, X_Λ is the sampling plan generated by the 1D quadrature rules in Λ, and X_l is the sampling plan generated by $\Lambda \cup \{l\}$. Futhermore, $\tilde{q}_\Lambda$ is the polynomial surrogate constructed from points in X_Λ alone. A local error measure can now be defined as

$$\eta^{(l)} := \frac{1}{\#(X_l \backslash X_\Lambda)} \sum_{\boldsymbol{\xi}_j^{(l)} \in X_l \backslash X_\Lambda} \| s\left(\boldsymbol{\xi}_j^{(l)}\right) \|. \tag{5}$$

Note that other error measures, based on quadrature errors [9, 30], or Sobol sensitivity indices [29] can also be defined. The admissible forward neighbour with the highest error measure $\eta^{(l)}$ is added to Λ, which can cause new forward neighbours to become admissible, and the algorithm repeats.

Note that every index $\mathbf{l} = (l_1, \cdots, l_d) \in \Lambda$ constitutes a separate tensor product of 1D quadrature rules with orders given by $\mathbf{l}$. Therefore, unlike the standard approach (3), the SC expansion in the adaptive case is constructed as a linear combination of tensor products, i.e.

$$q(\boldsymbol{\xi}) \approx \tilde{q}(\boldsymbol{\xi}) = \sum_{\mathbf{l}\in\Lambda} c_{\mathbf{l}} \sum_{j_1=1}^{m_{l_1}} \cdots \sum_{j_d=1}^{m_{l_d}} q(\boldsymbol{\xi}_{\mathbf{j}}^{(\mathbf{l})})\; a_{j_1}^{(l_1)}(\xi_1) \otimes \cdots \otimes a_{j_d}^{(l_d)}(\xi_d), \tag{6}$$

where $q(\boldsymbol{\xi}_{\mathbf{j}}^{(\mathbf{l})}) = q(\xi_{j_1}^{(l_1)}, \cdots, \xi_{j_d}^{(l_d)})$, and m_{l_i} is the number of points generated by a one-dimensional rule of order l_i. The coefficients c_l are computed as

$$c_l = \sum_{k_1=0}^{1} \cdots \sum_{k_d=0}^{1} (-1)^{|\mathbf{k}|_1} \cdot \chi(\mathbf{l}+\mathbf{k}), \quad \text{where} \quad \chi(\mathbf{l}) = \begin{cases} 1 & \mathbf{l} \in \Lambda_{\mathbf{l}} \\ 0 & \text{otherwise} \end{cases}; \tag{7}$$

see [28] for details.

Since Eqn.(6) consists of a linear combination of tensor products, the choice of the quadrature rule chosen to generate the one-dimensional points significantly affects the total number of code evaluations. It is common practice to select a *nested* rule, which has the property that a rule of a given order contains all points generated by that same rule at lower orders. When taking linear combinations of tensor products built from nested 1D rules of different order, may points will overlap. This leads to a more efficient sparse sampling plan, especially in higher dimensions. For our calculations, we employ the well-known Clenshaw-Curtis quadrature rule; see e.g. [10].

4.1.2 Ensemble execution

Hence, through the use of adaptive methods we make the uncertainty analysis of CovidSim tractable, but our analysis nevertheless required us to perform thousands of runs, each with its own unique set of input parameters. Specifically, we used the Eagle supercomputer at Posnan Supercomputing and Network Centre [31], which has a track record of reliably supporting large ensemble calculations. The workflows associated with these UQ/SA procedures are large, multi-faceted and iterative, and to handle and curate them efficiently, we rely on the FabSim3 automation toolkit [26]. FabSim3 allows us to capture commonly used workflow patterns in single-line bash commands, and it automatically captures all the relevant input parameters, output data and variables of both the job submission environment and the local machine environment in which each simulation has been executed.

4.1.3 Sobol index calculation

Sobol indices are variance-based sensitivity measures of a function $q(\boldsymbol{\xi})$ with respect to its inputs $\boldsymbol{\xi} \in \mathbb{R}^d$ [23, 32]. Let $\mathbb{V}[q_{\mathbf{u}}]$ be a so-called partial variance, where the multi-index $\mathbf{u}$ can be any subset of $\mathcal{U} := \{1, 2, \cdots, d\}$. Each partial variance measures the fraction of the total variance in the output q that can be attributed to the input parameter combination indexed by u. The Sobol indices are defined as the normalised partial variances, i.e.

$$S_{\mathbf{u}} := \frac{\mathbb{V}\left[q_{\mathbf{u}}\right]}{\mathbb{V}[q]} \tag{8}$$

where $\mathbb{V}[q] = \sum_{u\subseteq\mathcal{U}} \mathbb{V}[q_{\mathbf{u}}]$ is the is the total variance of q [32]. Since all partial variances are positive, the sum of all possible $S_{\mathbf{u}}$ equals 1.

To perform the Sobol sensitivity analysis, we employ the method described in [24], which is an adaptation of a method originally proposed in [33]. The general idea is to transform the adaptive SC expansion into a polynomials chaos expansion (PCE), to facilitate the computation of the Sobol indices. The PCE equivalent of (3) reads

$$q(\boldsymbol{\xi}) \approx \tilde{q}(\boldsymbol{\xi}) = \sum_{\mathbf{k}\in\mathcal{K}} \eta_{\mathbf{k}}\, \phi^{(k_1)}(\xi_1) \otimes \cdots \otimes \phi^{(k_d)}(\xi_d) \tag{9}$$

Here, the basis functions $\phi_{\mathbf{k}}$ are usually constructed to be orthonormal to the input density, and the response coefficients $\eta_{\mathbf{k}}$ are normally computed via a spectral projection technique, or via a regression method. Unlike (3), summation does not take place over the collocation points $\boldsymbol{\xi}_{\mathbf{j}}$. Instead, it takes place over multi indices $\mathbf{k} = (k_1, \cdots, k_d) \in \mathcal{K}$, determined by a selected truncation scheme (see below). The PCE method is a well-know technique, and we refer to e.g. [34, 35] for more details.

The PCE method is particularly suited for sensitivity analysis, since the Sobol indices can be calculated from the response coefficients $\eta_{\mathbf{k}}$ in a post-processing procedure [36]. The PCE mean and variance (when the ϕ_k are orthonormal), are given by

$$\mathbb{E}\left[\tilde{q}\right] = \eta_{\mathbf{0}} \quad \text{and} \quad \mathbb{V}\left[\tilde{q}\right] = \sum_{\substack{\mathbf{k}\in\mathcal{K}\\ \mathbf{k}\neq\mathbf{0}}} \eta_{\mathbf{k}}^2 \tag{10}$$

[34]. Similarly, the partial variances can be computed with

$$\mathbb{V}\left[\tilde{q}_{\mathbf{u}}\right] = \sum_{\mathbf{k}\in\mathcal{K}_{\mathbf{u}}} \eta_{\mathbf{k}}^2 \quad \text{where} \quad \mathcal{K}_{\mathbf{u}} = \{\mathbf{k} \mid k_i > 0 \text{ when } k_i \in \mathbf{u},\;\; j = 0 \text{ when } j \notin \mathbf{u}\}. \tag{11}$$

The multi index set $\mathcal{K}_{\mathbf{u}}$ can be interpreted as the set of all multi indices corresponding to varying only the inputs indexed by $\mathbf{u}$. That is, if for instance $\mathbf{u} = (1, 3)$, $\mathcal{K}_{\mathbf{u}}$ is the subset of $\mathcal{K}$, with all indices $\mathbf{k}$ where $k_1 > 0$ and $k_3 > 0$, with all other $k_j = 0$. Note that with (10) and (11), the Sobol indices (8) are readily available, provided we have the PCE coefficients $\eta_{\mathbf{k}}$.

To compute the PCE coefficients from our anisotropic sparse grid, we can transform the Lagrange basis to a PCE basis on the level of the one-dimensional basis functions [24]. Applying this transformation $\mathcal{T}$ to (6) yields

$$\mathcal{T}\left[\tilde{q}\right] = \sum_{\mathbf{l}\in\Lambda} c_{\mathbf{l}}\, \mathcal{T}\left[\sum_{j_1=1}^{m_{l_1}} \cdots \sum_{j_d=1}^{m_{l_d}} q(\boldsymbol{\xi}_{\mathbf{j}}^{(\mathbf{l})})\, a_{j_1}^{(l_1)}(\xi_1) \otimes \cdots \otimes a_{j_d}^{(l_d)}(\xi_d)\right], \tag{12}$$

and so we have to apply the transformation separately to each tensor product. Equating a tensor product of (12) to a corresponding PCE expansion (9) yields

$$\sum_{j_1=1}^{m_{l_1}} \cdots \sum_{j_d=1}^{m_{l_d}} q(\boldsymbol{\xi}_{\mathbf{j}}^{(\mathbf{l})})\; a_{j_1}^{(l_1)}(\xi_1) \otimes \cdots \otimes a_{j_d}^{(l_d)}(\xi_d) = \sum_{\mathbf{k}\in\Lambda_{\mathbf{l}}} \eta_{\mathbf{k}}^{(\mathbf{l})} \phi^{(k_1)}(\xi_1) \otimes \cdots \otimes \phi^{(k_d)}(\xi_d), \tag{13}$$

where the PCE truncation is $\Lambda_{\mathbf{l}} := \{\mathbf{k} \mid \mathbf{k} \leq \mathbf{l}\}$ [24]. By using the orthogonality property of the PCE basis functions (and the independence of the input distributions), we can find an expression for each coefficient $\eta_{\mathbf{k}}^{(\mathbf{l})}$ as

$$\eta_{\mathbf{k}}^{(\mathbf{l})} = \sum_{j_1=1}^{m_{l_1}} \cdots \sum_{j_d=1}^{m_{l_d}} q(\boldsymbol{\xi}_{\mathbf{j}}^{(\mathbf{l})})\; \upsilon_{k_1}^{(l_1,j_1)} \otimes \cdots \otimes \upsilon_{k_d}^{(l_d,j_d)}, \tag{14}$$

where each univariate transformation coefficient $\nu_{k_i}^{(l_i,j_i)}$ is given by

$$\nu_{k_i}^{(l_i,j_i)} = \int a_{j_i}^{(l_i)} \phi_{k_i}\, p(\xi_i)\mathrm{d}\xi, \quad i = 1 \cdots, d. \tag{15}$$

This is integrated over the support of $p(\xi_i)$ using Gaussian quadrature. To generate the orthonormal ϕ_{k_i} we use the Chaospy package [37].

Once in possession of the $\eta_{\mathbf{k}}^{\mathbf{l}}$, we can compute the statistics and the Sobol indices corresponding to an adaptive sparse grid. The first two moments are given by

$$\mathbb{E}\left[\tilde{q}\right] = \sum_{\mathbf{l}\in\Lambda} c_{\mathbf{l}} \cdot \eta_{\mathbf{0}}^{(\mathbf{l})} \quad \text{and} \quad \mathbb{V}\left[\tilde{q}\right] = \sum_{\substack{\mathbf{l}\in\Lambda \\ \mathbf{l}\neq\mathbf{0}}} \left[\sum_{\mathbf{k}\in\mathcal{K}_{\mathbf{l}}} c_{\mathbf{k}} \eta_{\mathbf{l}}^{(\mathbf{k})}\right]^2 \tag{16}$$

where $\mathcal{K}_{\mathbf{l}} := \{\mathbf{k} \mid \mathbf{l} \in \Lambda_{\mathbf{k}}, \forall \mathbf{k} \in \Lambda\}$. The expression for the variance is obtained by i) inserting (6) in $\mathbb{E}\left[\tilde{q}^2\right] - \mathbb{E}\left[\tilde{q}\right]^2$; ii) grouping all terms with like $\mathbf{k}$ in $\mathbb{E}\left[\tilde{q}^2\right]$, which is what $\mathcal{K}_{\mathbf{l}}$ indicates; and iii) using the orthogonality of the $\phi_{\mathbf{k}}$ to remove all cross terms $\phi_{\mathbf{k}}\phi_{\mathbf{j}}$, $\mathbf{j} \neq \mathbf{k}$. The statistics (16) represent a more general version than those given in (10), and will revert to these equations when given the combination coefficients $c_{\mathbf{l}}$ corresponding to a standard, non-adaptive SC grid. The partial variances $\mathbb{V}\left[\tilde{q}_{\mathbf{u}}\right]$, and by extension the Sobol indices, are computed in the same way as before, namely by summing individual variance contributions indexed by the set $\mathcal{K}_{\mathbf{u}}$ shown in (11).

4.2 Uncertainty amplification factor

The aim here is to find a 'robustness score' of a computational model, under uncertainty in the input parameters. A simple (dimensionless) measure for variability in some random variable X is the coefficient of variation (CV), defined as the standard deviation over the mean, i.e.

$$CV(X) = \frac{\sigma_X}{\mu_X}, \quad \text{if } \mu_X \neq 0. \tag{17}$$

Any forward uncertainty propagation method approximates the first two moments of the output $q \in \mathbb{R}^N$, and so $CV(q) \in \mathbb{R}^N$ is readily available. Assuming we can (analytically) compute the first two moments of each input $\xi_i \in \boldsymbol{\xi}$, $i = 1, \cdots, d$, $CV(\xi_i) \in \mathbb{R}$ is also easily computed. Although $\boldsymbol{\xi}$ may contain inputs defined on vastly different scales, since the CV is a dimensionless quantity, this will not pose a problem. We propose to use the ratio of $CV(Q)$ and $CV(\xi)$ as a relative measure of variability between the input and the output. To do so we first have to account for the fact that in general, $d \neq N$. Here we choose to average over all points:

$$CVR := CV\left(\bar{q}\right)/CV\left(\bar{\boldsymbol{\xi}}\right) = \left(\frac{1}{N}\sum_{n=1}^{N}\frac{\sigma_{q_n}}{\mu_{q_n}}\right) \Bigg/ \left(\frac{1}{d}\sum_{i=1}^{d}\frac{\sigma_{\xi_i}}{\mu_{\xi_i}}\right). \tag{18}$$

The basic idea of (18) is to say something about the robustness of the code to input uncertainty, given the fact that in all likelihood the choice of input distributions can be at least partly ambiguous. We have for instance prescribed an input distribution for ‘Relative household contact rate after closure’ with end points located at 20% of the default value (see Supplementary Table 1). Although this was within the range suggested by expert opinion, the number of 20% is still just a user-specified choice, and it might as well have been for instance 15%. It therefore makes sense to look at the relative input-to-output uncertainty. Thus, given a user-specified average input perturbation of say 20% ($CV(\bar{\boldsymbol{\xi}}) = 0.2$), Eqn. (18) tell us to what extent the code (which is a nonlinear mapping from the input to the output), amplifies this assumed uncertainty. Relative damping of uncertainty is also possible, corresponding to CVR < 1.

Data availability

Figures 1a and 1b display publicly available cumulative death count data for the UK, obtained from [22]. The Source Data for Figures 1 and 2 are available with this manuscript. Furthermore, the ‘parameter list’ with all input parameters, a description, their default values and reasons for inclusion or exclusion from the Imperial College CovidSim team is available as a Supplementary Data file.

Code availability

The version of EasyVVUQ that was used to generate our results has been pushed to a separate, publicly available GitHub branch, see [38] or [39] for a Zenodo link. Likewise, the FabSim3 interface between EasyVVUQ and CovidSim that was used to execute the ensembles on the PNSC Eagle supercomputer can be found here [40] or here [41].

References

[28] T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numerical Algorithms*, 3(18):209–232, 1998.

[29] R.P. Dwight, S.G.L. Desmedt, and P.S. Omrani. Sobol indices for dimension adaptivity in sparse grids. In *Simulation-Driven Modeling and Optimization*, pages 371–395. Springer, 2016.

[30] A. Narayan and J.D. Jakeman. Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation. *SIAM Journal on Scientific Computing*, 36(6):A2952–A2983, 2014.

[31] Poznan Supercomputing and Networking Center HPC WIKI . The Eagle supercomputer. https://wiki.man.poznan.pl/hpc/index.php?title=Eagle.

[32] I.M. Sobol. Global sensitivity indices for nonlinear mathematical models and their monte carlo estimates. *Mathematics and Computers in Simulation*, 55(1-3):271–280, 2001.

[33] G.T. Buzzard. Global sensitivity analysis using sparse grid interpolation and polynomial chaos. *Reliability Engineering & System Safety*, 107:82–89, 2012.

[34] B. Sudret. Polynomial chaos expansions and stochastic finite element methods. *Risk and Reliability in Geotechnical Engineering*, pages 265–300, 2015.

[35] D. Xiu and G.E. Karniadakis. The Wiener–Askey polynomial chaos for stochastic differential equations. *SIAM Journal on Scientific Computing*, 24(2):619–644, 2002.

[36] B. Sudret. Global sensitivity analysis using polynomial chaos expansions. *Reliability Engineering & System Safety*, 93(7):964–979, 2008.

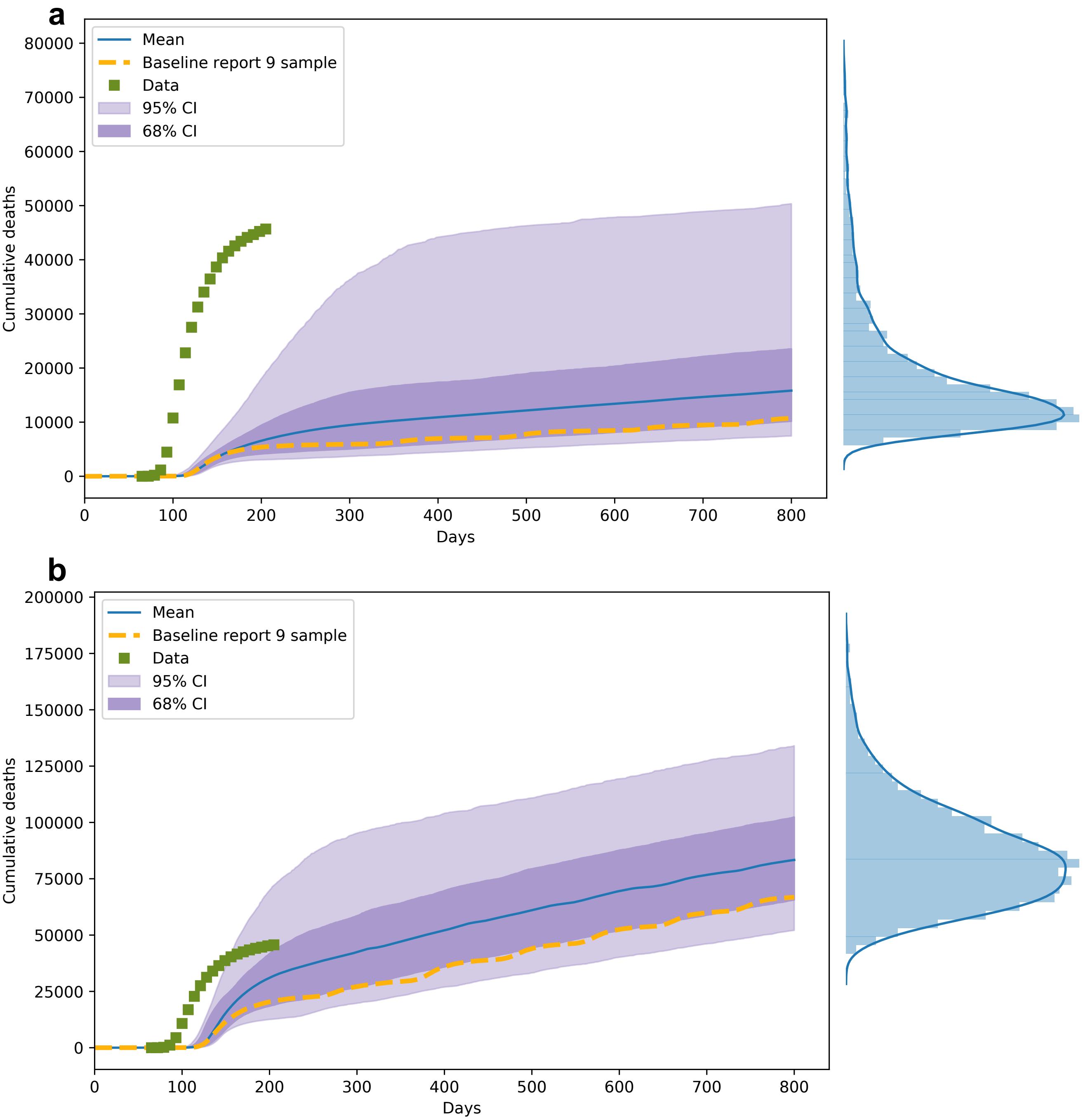
[37] J. Feinberg and H.P. Langtangen. Chaospy: An open source tool for designing methods of uncertainty quantification. *Journal of Computational Science*, 11:46–57, 2015.

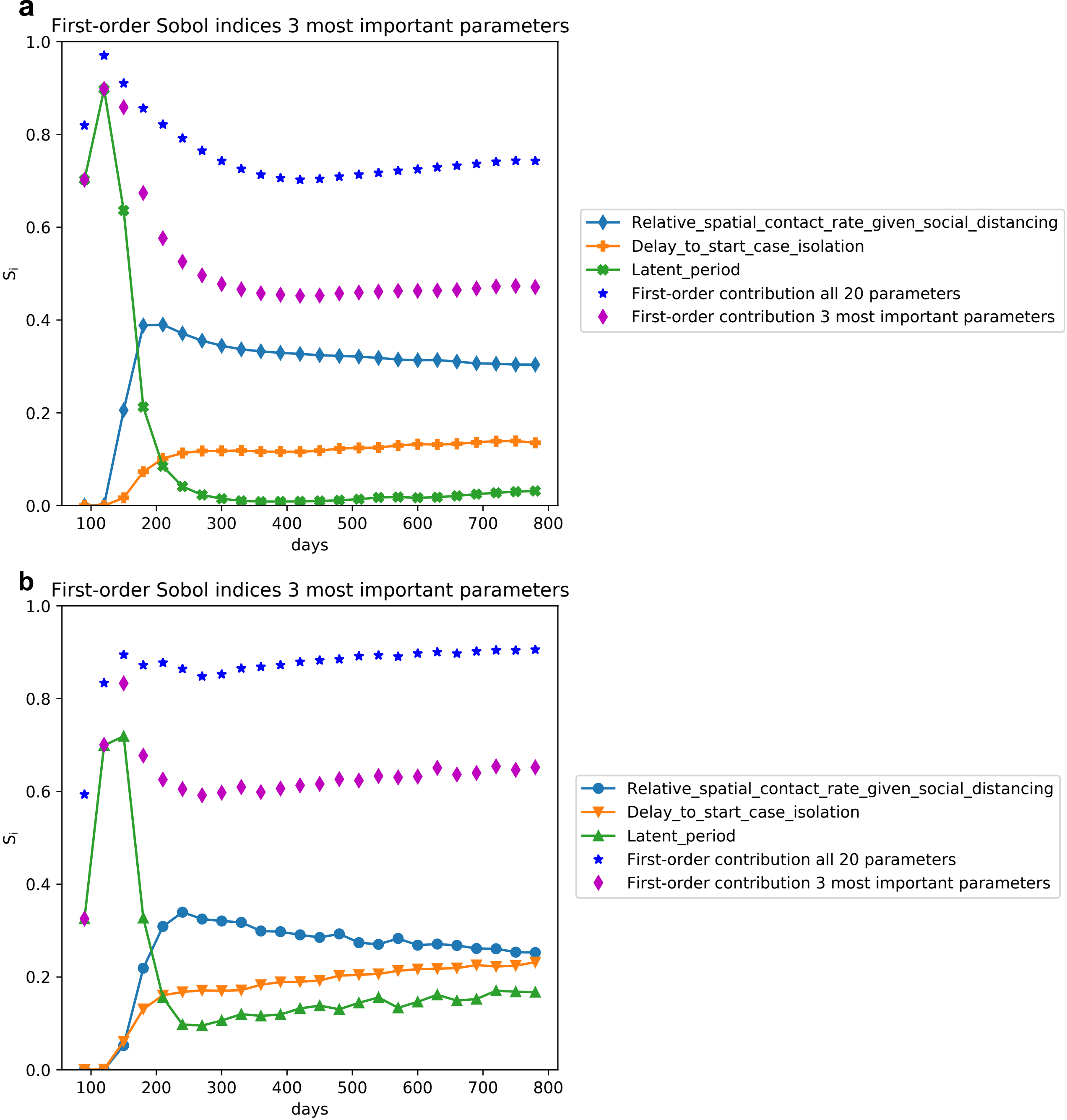
[38] VECMA Consortium. EasyVVUQ software - CovidSim branch. https://github.com/UCL-CCS/EasyVVUQ/tree/CovidSim.

[39] V. Jancauskas, R. Richardson, D.W. Wright, J. Lakhlili, W. Edeling, D.P. Coster, D. Groen, D. Suleimenova, A. Lebedev, and B. Bosak. wedeling/EasyVVUQ: Covidsim version. `https://doi.org/10.5281/zenodo.4445140`, 2021.

[40] VECMA Consortium. FabSim3 - EasyVVUQ interface for CovidSim. https://github.com/arabnejad/FabCovidsim/tree/dev.

[41] W. Edeling, H. Arabnejad, R. Sinclair, D. Suleimenova, K. Gopalakrishnan, B. Bosak, Derek Groen, I. Hashmi, D. Crommelin, and P.V. Coveney. Fabcovidsim. `https://doi.org/10.5281/zenodo.4445290`, 2021.


a
Mean
Baseline report 9 sample
Data
95% CI
68% CI
Cumulative deaths
80000
70000
60000
50000
40000
30000
20000
10000
0
0
100
200
300
400
500
600
700
800
Days
b
Mean
Baseline report 9 sample
Data
95% CI
68% CI
Cumulative deaths
200000
175000
150000
125000
100000
75000
50000
25000
0
0
100
200
300
400
500
600
700
800
Days


a
First-order Sobol indices 3 most important parameters
1.0
0.8
0.6
0.4
0.2
0.0
S_i
100
200
300
400
500
600
700
800
days
Relative_spatial_contact_rate_given_social_distancing
Delay_to_start_case_isolation
Latent_period
First-order contribution all 20 parameters
First-order contribution 3 most important parameters
b
First-order Sobol indices 3 most important parameters
1.0
0.8
0.6
0.4
0.2
0.0
S_i
100
200
300
400
500
600
700
800
days
Relative_spatial_contact_rate_given_social_distancing
Delay_to_start_case_isolation
Latent_period
First-order contribution all 20 parameters
First-order contribution 3 most important parameters