Understanding the uncertainty in the biospheric carbon flux for England and Wales

John Paul Gosling[†] and Anthony O'Hagan

30th November, 2007

Abstract

Uncertainty analysis is the evaluation of the distribution of a computer code output Y given uncertainty about the code's true inputs X. When we sum the outputs of a number of different codes, it is important to account for the different sources of uncertainty that can arise.

An estimate of the biospheric carbon flux for England and Wales requires the outputs of many different computer codes to be aggregated. In this report, the techniques needed for output aggregation will be discussed and the process of producing estimates of the carbon flux will be reviewed. This includes an independent normal approximation to a Dirichlet distribution for proportion parameters.

1 Introduction

Kennedy *et al.* (2007) gives an estimate of the mean and variance of net biome productivity (NBP) for England and Wales in the year 2000 based on output from a dynamic vegetation model (DVM). We would like to use the DVM to calculate NBP at 707 sites across Engalnd and Wales; however, due to the complexity of the model and the fact that the inputs are unknown, it is not feasible to do this. The estimate is found using Bayesian uncertainty analysis techniques (see Oakley and O'Hagan, 2002) that incorporate an extension of typical uncertainty analyses: uncertainty analysis for a linear combination of computer code

[†]Address for correspondence: John Paul Gosling, Department of Probability and Statistics, University of Sheffield, Hounsfield Road, Sheffield, S3 7RH, UK.
Email: J.P.Gosling@sheffield.ac.uk

outputs. This report serves to explain the theory behind the uncertainty analysis of aggregated computer code output and to clarify the sources of uncertainty in the final estimate of the NBP.

In Kennedy *et al.* (2007), three sources of uncertainty are considered. There is uncertainty due to the fact that the inputs into the DVM are unknown; for example, we do not know the exact percentage of soil that is clay at any site as each site has an area of about 210km^2. This is the type of uncertainty that is dealt with in conventional uncertainty analyses. The DVM cannot be evaluated for every possible input configuration; therefore, it is unknown at all input configurations apart from the ones we have run it with. We use emulation as detailed in Oakley and O'Hagan (2002) to account for this uncertainty. The third source of uncertainty stems from the fact that the emulation process would be too costly to carry out for all 707 sites, and we must interpolate from the sites where we use emulation to cover all of England and Wales.

In Section 2, a generic analysis is presented that introduces the key issues covering parameter uncertainty, code uncertainty and interpolation uncertainty. In that section, the theory required to carry out uncertainly analysis on aggregated computer code output is built up from the simple case where the output of the computer code is known for all inputs to the situation where the computer code is known for only a few sites (as it is in the estimation of NBP). The calculation of the NBP estimate is considered in Section 3 using the theory developed in Section 2.

Some of the inputs of the DVM are correlated proportions and, in the uncertainty analysis of Kennedy *et al.* (2007), independent normal distributions were used to represent our expert's beliefs instead of a more appropriate Dirichlet distribution. In Section 4, the validity of using independent normal distributions to model dependent proportion parameters is investigated.

2 A linear combination of code outputs

Consider the linear combination

$$Y = \sum_{i=1}^{n} \alpha_i f_i(\mathbf{x}_i), \tag{1}$$

where α_i is a known scalar and $f_i(\mathbf{x}_i)$ is the scalar output of a computer code run for inputs specified by $\mathbf{x}_i$ for each i. In Kennedy *et al.* (2007), two indices are summed over rather than the one given in equation (1); however, it is straightforward to convert this into the form given in equation (1) as the indexing sets are both finite. Different aspects of Y may be unknown to us: the true inputs of the code may be unknown and the outputs of $\{f_i\}$ may not be known for all possible inputs. To understand the sources of uncertainty, we first consider the effect of knowing the outputs of $\{f_i\}$ for all possible input values whilst being uncertain about the true values of the inputs.

2.1 Unknown computer code inputs

In this case, we know the value of each $f_i(\mathbf{x}_i)$ for all possible inputs. We would like to know what value we expect Y to take and what is the uncertainty around this estimate induced by the uncertainty in the inputs. Our expectation and uncertainty for Y is given by the following:

$$\begin{aligned} E_{\mathbf{X}}(Y|\{f_i\}) &= M = E_{\mathbf{X}}\left(\sum_{i=1}^{n} \alpha_i f_i(\mathbf{X}_i) \middle| \{f_i\}\right) \\ &= \sum_{i=1}^{n} \alpha_i E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right), \end{aligned} \tag{2}$$

$$\begin{aligned}
\mathrm{Var}_{\mathbf{X}}(Y|\{f_i\}) &= V = \mathrm{Var}_{\mathbf{X}}\left(\sum_{i=1}^{n}\alpha_i f_i(\mathbf{X}_i)\middle|\{f_i\}\right) \\
&= \sum_{i=1}^{n}{\alpha_i}^2\mathrm{Var}_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right) \\
&+ 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\alpha_i\alpha_j\mathrm{Cov}_{\mathbf{X}_i,\mathbf{X}_j}\left(f_i(\mathbf{X}_i),f_j(\mathbf{X}_j)|f_i,f_j\right). \qquad (3)
\end{aligned}$$

In the light of the uncertainty about the inputs $\mathbf{x}_i$, M can be seen as our best estimate of Y and V describes the uncertainty in this estimate. In principle, both M and V can be calculated either analytically or numerically depending on $\{f_i(.)\}$ as we know $\{f_i(.)\}$ for all possible inputs. For instance, they might be found by Monte Carlo sampling, in which random samples are drawn from the distributions of each of the $\mathbf{x}_i$s and $f_i(\mathbf{x}_i)$ derived for each sample value. Then the means, variances and covariances required in (2) and (3) can be computed as sample values. Of course, such a computation is not exact, but can be made as exact as required by taking sufficiently large samples. The variances of errors in the computations can also be measured in the usual way for Monte Carlo computation.

In the following sections, we address the situation where this kind of computation is not feasible. That is, we cannot evaluate means, variances and covariances analytically, and conventional numerical methods such as Monte Carlo or quadrature are not practical because the functions f_i are so costly or time-consuming to evaluate that we cannot perform enough function evaluations to produce accurate computations. This is the situation with the DVM studied in Kennedy *et al.* (2007).

It is therefore important to distinguish between uncertainty in Y due to uncertainty about model inputs and uncertainty due to imperfect computation. The implications of input uncertainty are encapsulated in the quantities M and V, that we wish to compute. The uncertainty due to imperfect computation can be

viewed as a consequence of the functions f_i being unknown.

2.2 Unknown computer code outputs

Suppose that both the functions and their inputs are unknown. Although in principle each function is known, so that $f_i(\mathbf{x})$ can be determined for any $\mathbf{x}$ the complexity of the simulator means that before running the computer code its output is unknown in practice. From a Bayesian perspective, we regard $f(.)$ as an unknown function, and, in line with O'Hagan (1992), we represent it with a Gaussian process model. We do have some training data $\mathbf{D}_i = \{(\mathbf{x}_j, f_i(\mathbf{x}_j))\}$ for each f_i so that we can build a statistical emulator for each f_i using the Gaussian process model. Kennedy *et al.* (2007) use a program called GEM-SA (see `http://www.tonyohagan.co.uk/academic/GEM/index.html`) that uses $\mathbf{D}_i$ to build the emulator for each f_i. GEM-SA can calculate a number of quantities that we need in our uncertainty analysis. An explanation of GEM-SA and its outputs can be found in Kennedy (2004).

The emulator describes uncertainty about each f_i given the training data $\mathbf{D}_i$, and so enables us to quantify uncertainty in M and V due to imperfect computation. We will refer to this as emulation uncertainty.

Given that we are uncertain about the function outputs for any inputs, our expectation of M from equation (2) is

$$\begin{aligned} E_{\{f_i\}}(M|\{\mathbf{D}_i\}) &= E_{\{f_i\}}(E_{\mathbf{X}}(Y|\{f_i\})|\{\mathbf{D}_i\}) \\ &= \sum_{i=1}^{n} \alpha_i E_{f_i}\left(E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\mathbf{D}_i\right). \end{aligned} \tag{4}$$

We need to calculate $E_{f_i}\left(E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\mathbf{D}_i\right)$; in GEM-SA, it is called the *mean of expected code output*.

Our uncertainty about $\{f_i\}$ leads to uncertainty about the calculation of M.

The variance about M due to this emulation uncertainty is given by

$$\mathrm{Var}_{\{f_i\}}(M|\{\mathbf{D}_i\}) = \mathrm{Var}_{\{f_i\}}(E_{\mathbf{X}}(Y|\{f_i\})|\{\mathbf{D}_i\}),$$

and from (2) this becomes

$$\begin{aligned}\mathrm{Var}_{\{f_i\}}(M|\{\mathbf{D}_i\}) = &\sum_{i=1}^{n} {\alpha_i}^2 \mathrm{Var}_{f_i}\left(E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\mathbf{D}_i\right) \\ &+ 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \alpha_i\alpha_j \mathrm{Cov}_{f_i,f_j}\left(E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right), E_{\mathbf{X}_j}\left(f_j(\mathbf{X}_j)|f_j\right)|\mathbf{D}_i,\mathbf{D}_j\right). \quad (5)\end{aligned}$$

Kennedy *et al.* (2007) treat the emulators as independent (this assumption will be discussed in Section 3). With this assumption, equation (5) becomes simply

$$\mathrm{Var}_{\{f_i\}}(M|\{\mathbf{D}_i\}) = \sum_{i=1}^{n} {\alpha_i}^2 \mathrm{Var}_{f_i}\left(E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\mathbf{D}_i\right). \quad (6)$$

We must calculate $\mathrm{Var}_{f_i}\left(E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\mathbf{D}_i\right)$; in GEM-SA, it is called the *variance of expected code output*.

Given our uncertainty about $\{f_i\}$, we must consider what value we expect V to take. From (3) we have

$$\begin{aligned}E_{\{f_i\}}(V|\{\mathbf{D}_i\}) = &\sum_{i=1}^{n} {\alpha_i}^2 E_{f_i}\left(\mathrm{Var}_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\,\mathbf{D}_i\right) \\ &+ 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \alpha_i\alpha_j E_{f_i,f_j}\left(\mathrm{Cov}_{\mathbf{X}_i,\mathbf{X}_j}\left(f_i(\mathbf{X}_i), f_j(\mathbf{X}_j)|f_i,f_j\right)\big|\,\mathbf{D}_i,\mathbf{D}_j\right). \quad (7)\end{aligned}$$

Again, we can use GEM-SA to calculate $E_{f_i}\left(\mathrm{Var}_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\,\mathbf{D}_i\right)$; it is called the *mean of total variance in code output*. The expected covariances in the second

part of equation (7) can be calculated using

$$E_{f_i,f_j}\left(\mathrm{Cov}_{\mathbf{X}_i,\mathbf{X}_j}\left(f_i(\mathbf{X}_i),f_j(\mathbf{X}_j)|f_i,f_j\right)\middle|\,\mathbf{D}_i,\mathbf{D}_j\right) = \left\{E_{f_i,f_j}[\mathrm{Var}_{\mathbf{X}_i,\mathbf{X}_j}(f_i(\mathbf{X}_i)+f_j(\mathbf{X}_j))|\mathbf{D}_i,\mathbf{D}_j] - E_{f_i}[\mathrm{Var}_{\mathbf{X}_i}(f_i(\mathbf{X}_i))|\mathbf{D}_i] - E_{f_j}[\mathrm{Var}_{\mathbf{X}_j}(f_j(\mathbf{X}_j))|\mathbf{D}_j]\right\}/2. \quad (8)$$

However, the calculation of $E_{f_i,f_j}[\mathrm{Var}_{\mathbf{X}_i,\mathbf{X}_j}(f_i(\mathbf{X}_i)+f_j(\mathbf{X}_j))|\mathbf{D}_i,\mathbf{D}_j]$ can be difficult due to the stronger assumptions and extra numerical complexity of bivariate emulation (see Conti and O'Hagan, 2007). We use $E_{f_i+f_j}[\mathrm{Var}_{\mathbf{X}_i,\mathbf{X}_j}(f_i(\mathbf{X}_i)+f_j(\mathbf{X}_j))|\mathbf{D}_i,\mathbf{D}_j]$ as an approximation instead as this only requires a univariate emulator. When emulating f_i+f_j, we hope to capture the joint structure between the two functions. This approximation will be close to the true expectation provided that the emulator of f_i+f_j is sufficiently accurate. It can be shown that

$$\mathrm{Var}_{\mathbf{X}_i,\mathbf{X}_j}(f_i(\mathbf{X}_i)+f_j(\mathbf{X}_j)) = \int f_i(\mathbf{x}_i)^2 dG_i(\mathbf{x}_i) + \int f_j(\mathbf{x}_j)^2 dG_i(\mathbf{x}_i) + 2\int f_i(\mathbf{x}_i)f_j(\mathbf{x}_j)dG_{ij}(\mathbf{x}_i,\mathbf{x}_j) - \left(\int f_i(\mathbf{x}_i)dG_i(\mathbf{x}_i) + \int f_j(\mathbf{x}_j)dG_j(\mathbf{x}_j)\right)^2, \quad (9)$$

where G_i is the distribution that characterises the beliefs about the input vector $\mathbf{x}_i$. Now, $E_{f_i+f_j}[\mathrm{Var}_{\mathbf{X}_i,\mathbf{X}_j}(f_i(\mathbf{X}_i)+f_j(\mathbf{X}_j))|\mathbf{D}_i,\mathbf{D}_j]$ can be calculated using the relationship in equation (9) and the emulator of the f_i+f_j. Notice now that, in addition to n emulators for the n individual functions f_i, we need to build $n(n-1)/2$ emulators for the sums f_i+f_j. However, this is a simple exercise and requires no extra evaluations of the functions themselves.

The results (4), (6) and (7) describe the consequences of uncertainty about the functions f_i which result in imperfect calculation of M and V. Thus, (4) is our estimate of M, and hence of Y; (6) describes uncertainty about M due

to emulation; and (7) is our estimate of the uncertainty about Y due to input uncertainty. It would be possible to develop equations for the variance of V due to emulation uncertainty, but these three results are enough to characterise the principal components of uncertainty. In particular, the emulation variance (6) and the input variance (7) can be combined using the identity

$$\begin{aligned} \mathrm{Var}_{\mathbf{X}}(Y|\{\mathbf{D}_i\}) &= E_{\{f_i\}}(\mathrm{Var}_{\mathbf{X}}(Y|\{f_i\})|\{\mathbf{D}_i\}) + \mathrm{Var}_{\{f_i\}}(E_{\mathbf{X}}(Y|\{f_i\})|\{\mathbf{D}_i\}) \\ &= E_{\{f_i\}}(V|\{\mathbf{D}_i\}) + \mathrm{Var}_{\{f_i\}}(M|\{\mathbf{D}_i\}); \end{aligned} \tag{10}$$

this gives us the total uncertainty in (4) as an estimate of Y, due to both input and emulation uncertainties.

2.3 No training data for some f_i

In this case, the functions and inputs are again unknown. We also do not have training data for every one of the f_is. Let I denote the set of indices i for which we have data $\mathbf{D}_i$. We proceed by interpolating the emulation results for the functions that have training data available using a set of functions $\{\mu\}$; this implies that we must have a spatial structure in which we can position the functions. We only know the exact value of $\{\mu\}$ for the functions with training data available; hence, there is an extra source of uncertainty here. Note that by kriging the emulator results we are now acknowledging correlation between emulators, whereas, in the previous section, we stated that the emulators would be treated as independent. If the emulators are sufficiently accurate, then this interpolation is reasonable provided that the functions we have training data for cover the range of different behaviour in the complete set of functions.

We define the sets of emulator results for the functions where we have training

data as follows:

$$\begin{aligned}
\mathcal{M}_E &= \{E_{f_i}\left(E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\mathbf{D}_i\right) : i \in I\}, \\
\mathcal{M}_V &= \{\mathrm{Var}_{f_i}\left(E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\mathbf{D}_i\right) : i \in I\}, \\
\mathcal{V}_E &= \{E_{f_i}\left(\mathrm{Var}_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\,\mathbf{D}_i\right) : i \in I\}, \qquad (11)
\end{aligned}$$

where the set elements are as defined in the previous section. We interpolate these sets of results using ordinary kriging techniques (see Matheron, 1971). In order to krige each set of results, we must specify a correlation structure between the unknown functions. We find the correlation structure for each interpolation by fitting covariance models to the emulator results using standard variogram fitting techniques of Cressie (1993). The fitted correlation structures for $\mathcal{M}_E$, $\mathcal{M}_V$ and $\mathcal{V}_E$ are denoted by $C(\mathcal{M}_E)$, $C(\mathcal{M}_V)$ and $C(\mathcal{V}_E)$ respectively. Note that there should be uncertainty about these parameters as we estimate the correlation structure from a limited number of emulator results. However, in the NBP calculations of Kennedy *et al.* (2007), the extra uncertainty caused by the estimation of the correlation parameters was found to be unimportant.

For $i \notin I$, we interpolate $\mathcal{M}_E$ using the kriging function μ_1 to find values for $E_{f_i}\left(E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\mathbf{D}_i\right)$. When kriging a set of results, we allow for uncertainty in the interpolation; hence, we can calculate a mean result for each unknown function and the uncertainty around that estimate. We interpolate $\mathcal{M}_V$ using μ_2 and $\mathcal{V}_E$ using μ_3.

Our expectation of Y is now given by

$$\begin{aligned}
E_{\mu_1}\left(E_{\{f_i\}}(M|\{\mathbf{D}_i\})\middle|\,\mathcal{M}_E, C(\mathcal{M}_E)\right) = \\
\sum_{i=1}^{n} \alpha_i E_{\mu_1}\left(E_{f_i}\left(E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\mathbf{D}_i\right)|\,\mathcal{M}_E, C(\mathcal{M}_E)\right), \qquad (12)
\end{aligned}$$

where E_{μ_1} is the expected function over the interpolating kriging functions of

$\mathcal{M}_E$. For $i \in I$,

$$E_{\mu_1}\left(E_{f_i}\left(E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\mathbf{D}_i|\right)|\mathcal{M}_E, C(\mathcal{M}_E)\right) = E_{\{f_i\}}(E_{\mathbf{X}_1}(f_i(\mathbf{X}_i)|f_i)|\mathbf{D}_i), \quad (13)$$

which is found using GEM-SA. This is due to the fact that μ_1 interpolates the emulator results. For $i \notin I$, the expectation on the right of equation (12) is found using the posterior mean from the kriging. We must allow for more uncertainty due to the use of the kriging estimate to calculate our expectation of M; from (4),

$$\mathrm{Var}_{\mu_1}(E_{\{f_i\}}(M|\{\mathbf{D}_i\})|\mathcal{M}_V, C(\mathcal{M}_V)) = \sum_{i=1}^{n} {\alpha_i}^2 \mathrm{Var}_{\mu_1}\left(E_{f_i}(E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\mathbf{D}_j)|\mathcal{M}_V, C(\mathcal{M}_V)\right)$$
$$+2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \alpha_i\alpha_j \mathrm{Cov}_{\mu_1}\left(E_{f_i}(E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\mathbf{D}_i), E_{f_j}(E_{\mathbf{X}_j}\left(f_j(\mathbf{X}_j)|f_j\right)|\mathbf{D}_j)|\mathcal{M}_V, C(\mathcal{M}_V)\right). \quad (14)$$

We calculate $\mathrm{Var}_{\mu_1}(E_{\{f_i\}}(M|\{\mathbf{D}_i\})|\mathcal{M}_V, C(\mathcal{M}_V))$ using the kriging covariances as shown in equation (14). For $i \in I$,

$$\mathrm{Var}_{\mu_1}\left(E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\mathcal{M}_V, C(\mathcal{M}_V)\right) = 0. \quad (15)$$

Our uncertainty about $\{f_i\}$ again leads to uncertainty about the calculation of M. The variance about M due to this uncertainty in the emulation is, from (6),

$$E_{\mu_2}\left(\mathrm{Var}_{\{f_i\}}(M|\{\mathbf{D}_i\})\right)|\mathcal{M}_V, C(\mathcal{M}_V)\big) = \sum_{i=1}^{n} {\alpha_i}^2 E_{\mu_2}\left(\mathrm{Var}_{f_i}\left(E_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i\right)|\mathbf{D}_i\right)\right)|\mathcal{M}_V, C(\mathcal{M}_V)\big). \quad (16)$$

For $i \in I$, the expectation on the right of equation (16) is found using the posterior mean from the kriging. Equations (14) and (16) give our uncertainty about M

due respectively to interpolation and emulation. We will label their sum $\mathrm{Var}_{\mathrm{Int}}$, which measures the total uncertainty in M due to imperfect computation.

We must consider what value we expect V to take. Since V is variance due to our uncertainty about the inputs, we will label this expectation $\mathrm{Var}_{\mathrm{Inp}}$. From (7), it is

$$E_{\mu_3}\left(E_{\{f_i\}}(V|\{\mathbf{D}_i\})\middle|\,\mathcal{V}_E, C(\mathcal{V}_E)\right) = \sum_{i=1}^{n} {\alpha_i}^2 E_{\mu_3}\left(E_{f_i}\left(\mathrm{Var}_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i)\right|\mathbf{D}_i\right)\middle|\,\mathcal{V}_E, C(\mathcal{V}_E)\right) + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\alpha_i\alpha_j C_{ij}, \quad (17)$$

where for $i \neq j$

$$C_{ij} = E_{\mu_3}\left(E_{f_i,f_j}\left(\mathrm{Cov}_{\mathbf{X}_i,\mathbf{X}_j}\left(f_i(\mathbf{X}_i), f_j(\mathbf{X}_j)|f_i, f_j\right)\middle|\,\mathbf{D}_i, \mathbf{D}_j\right)\middle|\,\mathcal{V}_E, C(\mathcal{V}_E)\right). \quad (18)$$

For $i = j$, we will also define

$$C_{ii} = E_{\mu_3}\left(E_{f_i}\left(\mathrm{Var}_{\mathbf{X}_i}\left(f_i(\mathbf{X}_i)|f_i)\right|\mathbf{D}_i\right)\middle|\,\mathcal{V}_E, C(\mathcal{V}_E)\right).$$

A strategy for computing the terms C_{ij} for $i \neq j$ both in I is given in Section 2.2 around equation (8). For $i = j$, C_{ii} is obtained from the kriging function μ_3. Let $\mathbf{C}$ be the matrix with elements C_{ij} and denote by $\mathbf{C}_I$ the submatrix corresponding to $i, j \in I$. We need values for the off-diagonal elements C_{ij} of $\mathbf{C}$ outside $\mathbf{C}_I$. We specify these using the simple assumption $C_{ij} = \rho\sqrt{C_{ii}C_{jj}}$. We choose the value of ρ such that, if the same assumption were used for the off-diagonal elements of $\mathbf{C}_I$ then the sum of all these elements would be unchanged. This construction will be reasonable when we believe that correlation is broadly stable between pairs of functions.

This completes the analysis of uncertainty due to both emulation and interpo-

lation. In the presence of input uncertainty alone, our estimate of Y would be M, and the uncertainty around that estimate would be given by V. Allowing also for emulation and interpolation uncertainty, we estimate M by equation (12); this is also our estimate of Y. Uncertainty about M due to emulation uncertainty alone would be given by (6), which we estimate (since we also have interpolation uncertainty) by (16). Interpolation adds further uncertainty about M in the form of (14). The sum of these last two components is the total uncertainty about M due to imperfect computation. Finally, we estimate V by equation (17).

If the emulation and interpolation methods have been effective, the sum of (16) and (14) will be small relative to V, and its square root will be small relative to M.

3 Aggregation for NBP

The preceding theory underlies the analysis in Kennedy *et al.* (2007), but there are a number of differences that we now address. In Kennedy *et al.* (2007), the linear combination of computer code outputs is

$$Y = \sum_{k=1}^{707} \sum_{t=1}^{4} a_k \gamma_t(\boldsymbol{y}_k) f_{tk}(\mathbf{x}_{tk}), \tag{19}$$

where k is the site number, each site's coordinates are given by the vector $\boldsymbol{y}_k$, t is the plant functional type (PFT) index, a_k is the area of site k, $\gamma_t(\boldsymbol{y}_k)$ is the proportion of PFT t at site k and $f_{tk}(\mathbf{x}_{tk})$ is the computer code NBP output of PFT t at site k for input $\mathbf{x}_{tk}$. The true inputs to the computer code for each site and PFT are unknown. As shown in equation (19), we have 707 sites and 4 PFTs. There is a single computer code, the dynamic vegetation model, which is run 2828 times with different combinations of a large number of inputs in order to produce all these values. However, for our purposes we regard each f_{tk} as a different unknown function, as if produced by a different computer model. This

is because it would be impractical to try to build an emulator for the DVM over the very large input space, and in effect by building an emulator for f_{tk} we are emulating the DVM over the much smaller part of the space needed to represent uncertainty in the inputs $\mathbf{x}_{tk}$ for PFT t at site t.

Even then, it would require an unfeasible number of DVM runs to build 2828 emulators. We performed just a limited number of runs for various input combinations at just 33 out of the 707 sites. At the 33 sites, we have data for all four PFTs. The $f_{tk}(\mathbf{x}_{tk})$ are unknown except at these 33 sites and input values.

This places us broadly in the situation that was analysed in Section 2.3. We have a double summation in (19), but we could represent this as a single summation as in (1), with the index i running over all 2828 pairs (t, k). However, although the sites have a spatial structure that could be exploited as in Section 2.3, the (t, k) pairs do not. Furthermore, there is interest not just in the grand total Y but also in the site NBF aggregated over PFTs,

$$Y_{\cdot k} = \sum_{t=1}^{4} \gamma_t(\boldsymbol{y}_k) f_{tk}(\mathbf{x}_{tk}), \tag{20}$$

and in each PFT's contribution to the total NBP, aggregated over England and Wales,

$$Y_{t\cdot} = \sum_{k=1}^{707} a_k \gamma_t(\boldsymbol{y}_k) f_{tk}(\mathbf{x}_{tk}). \tag{21}$$

Although the methods developed in the preceding section do not apply to the double summation, they apply to a single summation over t or over k, and this gives us two ways to quantify uncertainty in Y. We can first analyse the uncertainty in each $Y_{\cdot k}$ and then in $Y = \sum_{k=1}^{707} a_k Y_{\cdot k}$. Alternatively, we could first analyse the uncertainty in each $Y_{t\cdot}$ and then in $Y = \sum_{t=1}^{4} Y_{t\cdot}$. In this section, we consider these issues.

3.1 Making the maps

Kennedy *et al.* (2007) present maps (their Figure 7) of the mean and standard deviation of $Y_{\cdot k}$ over England and Wales. For each of the 33 sites at which we have emulators, this is straightforward using the theory of Section 2.2 since we have emulator data for each of the four PFTs. Indeed, Appendix C of Kennedy *et al.* (2007) presents closed forms for the integrals needed to do this. (Note that the methods of Appendix C require emulation data to be available for each function; the claim there that the methods apply to aggregation over sites is an error that slipped through the checking and proof reading of that paper, the methods apply instead to aggregation over PFTs at the 33 sites for which we have such data.)

We might then complete the maps by kriging the means and variances, as discussed in Section 2.3. However, this is not a good way to proceed because these site aggregates do not vary smoothly over the country. This is because the land cover, represented by the $\gamma_t(\mathbf{y}_k)$, is not smooth. Instead, we chose to do the kriging on each PFT separately. Figure 3 of Kennedy *et al.* (2007) shows maps of the standard deviations for each PFT. Their Figure 2, however, shows another way in which their analysis differs slightly from the general approach presented here. Instead of kriging the estimated NBP values at each site, we applied kriging to the difference between these estimates and the plug-in estimates. The distinction is explained in Kennedy *et al.* (2007), but in our notation consider the value $Y_{tk} = f_{tk}(\mathbf{X}_{tk})$ of the model output NBP at site k for PFT t and using the true input values $\mathbf{X}_{tk}$. We have uncertainty about this because of uncertainty about $\mathbf{X}_{tk}$, and the expectation with respect to this uncertainty is $M_{tk} = E_{\mathbf{X}_{tk}}(Y_{tk})$. Because we also have emulation uncertainty, the estimate of Y_{tk} is the expectation of M_{tk} with respect to the emulator distribution of f_{tk}. The plug-in estimate is just $f_{tk}(\mathbf{x}^\star_{tk})$, where $\mathbf{x}^\star_{tk} = E_{\mathbf{X}_{tk}}(\mathbf{X}_{tk})$ is the expected value of the uncertain input. The plug-in value is the naive user's 'best estimate' of the model output obtained by using ('plugging in') the best estimate of the model inputs, and is available

at every site for every PFT. The difference between this plug-in estimate and the expected output, taking account of uncertainty, is called the *correction* in Kennedy *et al.* (2007), and it is the corrections that are interpolated by kriging in their Figure 2. We then estimate NBP at each site for each PFT by adding the interpolated correction to the plug-in estimate.

To aggregate these maps across the four PFTs, we need to evaluate covariances between the values for different PFTs at a given site, which was again done using kriging. Details are given below.

3.2 Aggregating over sites

The aggregation over sites for a given PFT follows exactly the procedures described in Section 2.3. In particular, the estimate of $Y_{t\cdot}$ is obtained using equation (12), the components of variance in this estimate due to emulation and interpolation uncertainties are given respectively by (16) and (14), and the estimated variance due to input uncertainty by (17).

We then need to aggregate over PFTs to obtain the corresponding analysis for $Y = \sum_{t=1}^{4} Y_{t\cdot}$. The estimate is of course just the sum of the estimates of $Y_{t\cdot}$. The variance due to input uncertainty is

$$\begin{aligned}
\mathrm{Var}_{\mathbf{X}}(Y|\{f_{tk}\}) &= \mathrm{Var}_{\mathbf{X}}(Y_{1\cdot} + \cdots + Y_{4\cdot}|\{f_{tk}\}) \qquad (22)\\
&= \sum_{t=1}^{4} \mathrm{Var}_{\mathbf{X}_t}(Y_{t\cdot}|\{f_{tk}\}) + 2\sum_{t=1}^{3}\sum_{t'=t+1}^{4} \mathrm{Cov}_{\mathbf{X}_t,\mathbf{X}_{t'}}(Y_{t\cdot}, Y_{t'\cdot}|\{f_{tk}\},\{f_{t'k}\})\\
&= \sum_{t=1}^{4} V_t + 2\sum_{t=1}^{3}\sum_{t'=t+1}^{4} \mathrm{Cov}_{\mathbf{X}_t,\mathbf{X}_{t'}}(Y_{t\cdot}, Y_{t'\cdot}|\{f_{tk}\},\{f_{t'k}\}).
\end{aligned}$$

Section 2.3 gives the formula for $E_{\mu_3}\left(E_{\{f_{tk}\}}(V_t|\{\mathbf{D}_{tk}\})\right)|\mathcal{V}_E, C(\mathcal{V}_E)\big)$ provided that we assume the between site correlations to be fixed for all sites for each

PFT. We must now consider the covariance between Y_t and $Y_{t'}$:

$$\begin{aligned}\mathrm{Cov}_{\mathbf{X}_t,\mathbf{X}_{t'}}(Y_{t\cdot}, Y_{t'\cdot}|\{f_{tk}\},\{f_{t'k}\}) &= \sum_{k=1}^{707}\sum_{j=1}^{707} a_k\gamma_t(\boldsymbol{y}_k)a_j\gamma_{t'}(\boldsymbol{y}_j)E_{C_{t,t'}} \\ &= \sum_{k=1}^{707}\sum_{j=1}^{707} a_k\gamma_t(\boldsymbol{y}_k)a_j\gamma_{t'}(\boldsymbol{y}_j)\mathrm{Cov}_{\mathbf{X}_{tk},\mathbf{X}_{t'j}}(f_{tk}(\mathbf{X}_{tk}), f_{t'j}(\mathbf{X}_{t'j})|f_{tk}, f_{t'j}). \quad (23)\end{aligned}$$

Emulators at different sites are assumed to be independent so equation (23) becomes

$$\begin{aligned}\mathrm{Cov}_{\mathbf{X}_t,\mathbf{X}_{t'}}(Y_{t\cdot}, Y_{t'\cdot}|\{f_{tk}\},\{f_{t'k}\}) &= \sum_{k=1}^{707} {a_k}^2\gamma_t(\boldsymbol{y}_k)\gamma_{t'}(\boldsymbol{y}_k)E_{C_{t,t'}} \\ &= \sum_{k=1}^{707} {a_k}^2\gamma_t(\boldsymbol{y}_k)\gamma_{t'}(\boldsymbol{y}_k)\mathrm{Cov}_{\mathbf{X}_{tk},\mathbf{X}_{t'k}}(f_{tk}(\mathbf{X}_{tk}), f_{t'k}(\mathbf{X}_{t'k})|f_{tk}, f_{t'k}), \quad (24)\end{aligned}$$

which can be calculated at sample sites using equation (8). These results can then be extended to non-sample sites using kriging. Hence, we krige six inter-PFT covariances, given by $E_{C_{t,t'}}$, that we have calculated at the 33 sample sites using the same kind of technique that was described in Section 2.3.

The assumption of independence between emulators was validated by building a multivariate emulator of multiple outputs using the theory of Conti and O'Hagan (2007). It was found that the correlation between emulators was small provided the emulation was sufficiently accurate. In this case, the emulators were deemed to be accurate enough to assume independence.

3.3 Results

We can use the method described in this paper to create an estimate for Y and to quantify the uncertainty about Y from the two sources: uncertainty about the code at different sites and uncertainty about the true code inputs. Table 3 of Kennedy *et al.* (2007) presents the plug-in estimates, the mean estimates

accounting for the various sources of uncertainty, and the overall variance, for each PFT and for the aggregate over PFTs.

Table 1 below extends those results to show the variances separated into two components. $\mathrm{Var}_{\mathrm{Int}}$ is the variance due to emulation and interpolation, while $\mathrm{Var}_{\mathrm{Inp}}$ is the estimated variance due to input uncertainty. In principle, the first of these can be reduced by doing more accurate computation, which in the present case would mean doing more runs to increase the accuracy of each of the 33×4 emulators that we built (reducing emulation uncertainty) or building emulators at more sites (to reduce interpolation uncertainty). The aim is for $\mathrm{Var}_{\mathrm{Int}}$ to be small relative to $\mathrm{Var}_{\mathrm{Inp}}$ and for its sqare root to be small relative to the mean. We see that this has been achieved for the overall total and for the Grassland and ENL PFTs, but that $\mathrm{Var}_{\mathrm{Int}}$ is perhaps not as small as we would like for the other two PFTs.

The computations with the DVM needed to carry out the analysis described here are already very substantial, and it would not be practical to increase them. It is important to be aware that the emulation methodology is already very much more efficient than more traditional Monte Carlo computation, and that it would not have been possible obtain even the accuracy achieved here without emulation. Despite the remaining imprecision, Kennedy *et al.* (2007) succeed in estimating the total NBP in England and Wales in 2000 with sufficient accuracy to show the substantial bias in the plug-in estimates, and have identified the relative magnitudes of the contributions of each of the PFTs to that total.

4 Uncertainty in soil inputs

Table 2 of Kennedy *et al.* (2007) lists the normal distributions that were used to represent beliefs about three input parameters of SDGVM: the percentage of sand in the soil S_S, the percentage of clay in the soil S_C and the soil's bulk density. There is a clear link between the first two of these parameters as knowing one of

Table 1: Contribution to the mean and variance of total NBP from different plant functional types and covariances between these types.

PFT	Plug-in (MtC)	Mean (MtC)	Var_{Int} $(MtC)^2$	Var_{Inp} $(MtC)^2$	Var_{Tot} $(MtC)^2$
Grassland	5.2793	4.6389	0.0090	0.2598	0.2689
Crop	0.8525	0.4454	0.0090	0.0248	0.0338
DBL	2.1319	1.6826	0.0048	0.0080	0.0128
ENL	0.7976	0.7807	8.801×10^{-6}	0.0005	0.0005
Covariances				0.0010	0.0010
Total	9.0613	7.5475	0.0229	0.2941	0.3170

the percentages will put bounds on the size of the other. By using independent normal distributions for these two parameters, we ignore this structure. In Table 2 of Kennedy *et al.* (2007), site 17 has the following distributions:

$$\begin{aligned} S_S &\sim N(69.91, 163.47), \\ S_C &\sim N(13.23, 52.70). \end{aligned} \tag{25}$$

If we treat these as independent, we get $E(S_S + S_C) = 83.14$ and $Var(S_S + S_C) = 216.17$. The upper bound for $S_S + S_C$ of 100% is below its expectation plus two standard deviations; hence, there is appreciable probability of a percentage of over 100. In order to examine the impact of this, an analysis based on better-suited distributions for S_S and S_C has been performed.

A Dirichlet distribution is a natural choice of distribution for these types of parameters. The constraint that the percentages add to 100 can be automatically included in the Dirichlet distribution. To use the Dirichlet distribution, we introduce a third parameter: the percentage of soil that is not sand or clay S_R ($S_R = 100 - S_S - S_C$). Hence, we have

$$\{S_S, S_C, S_R\} \sim \text{Dir}(\alpha_S, \alpha_C, \alpha_R), \tag{26}$$

where

$$\begin{aligned} E(S_I|\alpha_S,\alpha_C,\alpha_R) &= 100\frac{\alpha_I}{\alpha_0}, \\ Var(S_I|\alpha_S,\alpha_C,\alpha_R) &= 10000\frac{\alpha_I(\alpha_0-\alpha_I)}{\alpha_0^2(\alpha_0+1)}, \\ \alpha_0 &= \alpha_S+\alpha_C+\alpha_R. \end{aligned} \tag{27}$$

With this distribution, we have three hyperparameters to set: α_S, α_C and α_R. Values for these are selected using the means of S_S and S_C and the variance of S_S from Table 2 of Kennedy *et al.* (2007) (the variance of S_C calculated using the Dirichlet fit is similar to the variance reported in Table 2). After specifying values for our hyperparameters, we have arrived at a joint distribution that upholds the sum to 100 constraint.

The program we used to carry out the uncertainty and sensitivity analysis of SDGVM, *GEM-SA*, only allows a joint distribution for the inputs that is product normal or product uniform. In order to find normal distributions that reflect the additional structure we have added by using the Dirichlet distribution, we use a transformation of one of the parameters. If we set

$$P_C = \frac{100S_C}{(100-S_S)} \tag{28}$$

and use properties of the Dirichlet distribution, we find that S_S and P_C are independent with the following distributions:

$$\begin{aligned} S_S/100 &\sim \text{Be}(\alpha_S,\alpha_0-\alpha_S), \\ P_C/100 &\sim \text{Be}(\alpha_C,\alpha_R). \end{aligned} \tag{29}$$

The distributions given in (29) can then be approximated by normal distributions in the following manner:

1. The mode of the beta distribution is used as the mode of our approximate

normal distribution.

2. If the mode is less than or equal to 0.5, set the variance of our approximate normal distribution so that the 1st percentiles match for both distributiions. If the mode is greater than 0.5, set the variance of our approximate normal distribution so that the 99th percentiles match for both distributiions.

The second step here helps to reduce the probability of the parameter going outside its bounds.

For site 17, we find the following normal approximations:

$$\begin{aligned} S_S &\sim N(73.95, 70.56), \\ P_C &\sim N(36.29, 205.35). \end{aligned} \tag{30}$$

The normal approximation for S_S given in (30) gives a probability of less than 0.001 of S_S falling outside the range of 0–100%. The variable P_C has been constructed to prevent S_C being a value that will make $S_S + S_C$ at most 100% provided S_S is at most 100%. However, P_C must still be in the range 0–100%; the normal approximation in (30) gives a probability of 0.006 of P_C falling outside the range, which is makes it very unlikely.

Repeating the uncertainty analysis for site 17 using the new normal distributions, the biggest differences are found for grassland. There is a 0.5% increase in mean NBP and a 8% increase in the standard deviation of NBP. This is the greatest change at any site using the normal approximation of the Dirichlet distribution. Also, the mean of NBP and the standard deviation of NBP decreases at some sites. The change to the approximate Dirichlet would make very little difference to the overall figures presented in Table 3 of Kennedy *et al.* (2007). There is only likely to be a relatively small decrease in the variance due to the extra constraint that the Dirichlet distribution puts on the parameters.

References

Conti, S. and O'Hagan, A. (2007). Bayesian emulation of complex multi-output and dynamic computer models. *Research report 569/07*, Department of Probability and Statistics, University of Sheffield. Submitted to *Journal of Statistical Planning and Inference*.

Cressie, N. (1993). *Statistics for Spatial Data* (rev. ed.). New York: Wiley.

Kennedy, M.C. (2004). Description of the Gaussian process model used in GEM-SA. *GEM-SA help documentation*.

Kennedy, M.C., Anderson, C.W., O'Hagan, A., Lomas, M.R., Woodward, F.I., Gosling, J.P. and Heinemeyer, A. (2007). Quantifying uncertainty in the biospheric carbon flux for England and Wales. To appear in *J. R. Statist. Soc. Ser. A*.

Matheron, G., (1971). The theory of regionalized variables and its applications. *Cahier du Centre de Morphologie Mathematique*, **5**. Fontainebleau, France.

Oakley, J.E. and O'Hagan, A. (2002). Bayesian inference for the uncertainty distribution of computer model outputs. *Biometrika*, **89**, 769–784.

O'Hagan, A. (1992). Some Bayesian numerical analysis. In *Bayesian Statistics 4* (eds. Bernado, J.M. *et al.*), 345–363. Oxford: Oxford University Press.