REDUCTION AND EMULATION OF ADMS URBAN

Vivien Mallet[1,2], Anne Tilloy[1,2], David Poulet[3] and Fabien Brocheton[3]

[1]INRIA, Paris-Rocquencourt research center, France
[2]CEREA, joint laboratory Ecole des Ponts ParisTech – EDF R&D, Université Paris-Est, Marne la Vallée, France
[3]Numtech, Aubière, France

Abstract: ADMS Urban is a non-linear static model whose input data p varies one simulated hour after the other. The model computes a high-dimensional concentration vector $y = \mathcal{M}(p)$ which can contain 10^5 concentrations. A full-year simulation of NO_2 concentrations can take dozens of days of computations, which greatly limits the range of methods that can be applied to the model, especially for uncertainty quantification. This work proposes a method to replace ADMS Urban with a so-called emulator, i.e., a close approximation of ADMS Urban whose computational cost is negligible. First, the output concentration field y is projected on a few modes of a proper orthogonal decomposition$[\Psi_1 \dots \Psi_N]$, so that $y \simeq \sum_{j=1}^{N} \alpha_j \Psi_j$ where α_j is the projection coefficient on *j*-th mode and *N* smaller than 10. Then, the reduced model $f(p) = \mathbf{\Psi}^T \mathcal{M}(p)$ is replaced by a statistical emulator $\hat{f}$ so that $\hat{f}(p) \simeq f(p)$ and the computational cost of $\hat{f}(p)$ is negligible.

Key words: *Dimension reduction, statistical emulation, ADMS Urban.*

INTRODUCTION

ADMS Urban is a non-linear static model whose input data *p* varies from one simulated hour to the next. The input vector $p \in \mathbb{R}^K$ may essentially contain scalar values, such as meteorological variables, background concentrations and a few emission factors (in case the spatial distribution of the emissions is fixed). The model computes a high-dimensional concentration vector $y = \mathcal{M}(p)$ which can contain 10^5 concentrations. A full-year simulation of NO_2 concentrations over a city can take dozens of days of computations, which greatly limits the range of methods that can be applied to the model, especially for uncertainty quantification. This work proposes a method to replace ADMS Urban with a so-called emulator, i.e., a reasonably close approximation of ADMS Urban whose computational cost is negligible. The ADMS Urban output field is first projected onto a reduced subspace. The relations between the projection coefficients and the input vector *p* are smooth enough to be emulated by a surrogate model whose computational cost is negligible.

METHOD

Dimension reduction

First, we want to project any output concentration field *y* onto a subspace spanned by the reduced basis $\mathbf{\Psi} = [\Psi_1 \dots \Psi_N]$. The reduced basis is chosen so as to represent the variability of the concentration fields. It can be determined by a principal component analysis on some training period. Then we write $y \simeq \sum_{j=1}^{N} \alpha_j \Psi_j$ where $\alpha_j = y^T \Psi_j$ is the projection coefficient on *j*-th principal component. In practice, we found that $N = 8$ principal components are enough to provide a good approximation, on average, to a sequence of concentration fields for a full year.

In some cases, the input vector *p* might be of high dimension. Then a dimension reduction should be carried out on it as well. In this study, its dimension ($K = 10$) is low enough to skip this step.

Emulation

Second, the reduced model $f(p) = \mathbf{\Psi}^T \mathcal{M}(p)$ is replaced by a statistical emulator $\hat{f}$ so that $\hat{f}(p) \simeq f(p)$ and the computational cost of $\hat{f}(p)$ is negligible. In practice, one independent emulator is built for every component $f_j(p)$. In order to build the emulator $\hat{f}_j$, we first need *M* training samples $f_j(p^{(i)})$. The samples $p^{(i)}$ are generated with latin hypercube sampling. The emulator is then made of two parts: a

regression part, and a part that is essentially an interpolation between different samples $f_j(p^{(i)})$ (or, more precisely, the regression residuals of the samples). With a linear regression, the emulator reads as in equation (1).

$$\hat{f}_j(p) = \sum_{k=1}^{K} \beta_{j,k} p_k + \sum_{i=1}^{M} w_j(p, p^{(1)}, \dots, p^{(M)}) \left[f_j(p^{(i)}) - \sum_{k=1}^{K} \beta_{j,k} {p_k}^{(i)} \right] \quad (1)$$

The regression is usually a multiple linear regression, but a nonlinear regression may be applied as well. It represents the basic dependencies between the input variables and the projection coefficient α_j. We found that the dependence of α_j with respect to the input variables is often smooth, yet nonlinear. In practice, the regression part is not the key part of the emulator.

The most important part is the interpolation of the residuals $f_j(p^{(i)}) - \sum_{k=1}^{K} \beta_{j,k} {p_k}^{(i)}$ of the regression. The weights $w_j(p, p^{(1)}, \dots, p^{(M)})$ essentially depend on the distance between p and the different samples $p^{(i)}$. There exist many methods to compute these weights. A classical method is kriging, which is often used in Gaussian process emulators (Sacks et al., 1989). However this approach is computationally intensive, even with K = 10 input variables. We therefore investigated the inverse distance weighting as proposed in Joseph and Kang (2011) and the interpolation with radial basis functions. In this study, we simply use the mean of the n closest neighbors, i.e., w_j is zero for all samples except for the closest samples $p^{(i)}$ to p. The n closest samples are averaged, hence the weights are $\frac{1}{n}$. The metric d to determine the closest samples is in the form $d(p, p^{(i)}) = \sqrt{\sum_{k=1}^{K} \theta_k (p_K - p_K^{(i)})^2}$ where the coefficients $\theta_k > 0$ are determined either by manual tuning or by cross-validation. The most important input variables should be given a high θ_k so as to strongly constrain the metric.

CASE STUDY

We evaluate the accuracy of the reduction and the emulation, and summarize the computational costs for the generation of the emulator. The application is the simulation of NO_2 concentrations every 3 hours across the city Clermont-Ferrand (France) for the full year 2008 (2928 dates).

Input variables

The meteorological data are the wind speed and the wind direction at a meteorological station. The temperature is used as well, for the chemical mechanism. The cloud coverage, homogeneous over the domain, is used to compute the vertical mixing. The Julian day is necessary to compute the solar radiation. ADMS Urban also takes into account the rain intensity. In total, we therefore have 6 meteorological input scalars.

The spatial distribution of the emissions is fixed and is part of the model $\mathcal{M}$. ADMS Urban relies on emission factors, which depends on the month and on the day type (weekday, Saturday and Sunday). All the emission factors are also part of the model. So, the emissions vary according to the Julian day and the hour of the day - which makes 1 additional variable in p. We need the background concentrations (one scalar for the whole domain) for NO_2, NO_x and O_3. We decided to set the COV background concentrations to 15 µg m^{-3}. Hence we have 3 additional input variables. In total, p contains K = 10 components.

Dimension reduction

We applied the principal component analysis to the full year 2008, but we also noticed that similar results are obtained with 6-month training period. We decided to project the output concentration fields on 8 principal components. The unexplained variance over the full year is less than 1% of the total variance, as depicted in Figure 1. The first principal component, as shown in Figure 2(a) corresponds to the emission areas, especially linked to traffic. The second principal component enables to modulate the intensity of

emissions. The third and the forth principal components are presumably related to dispersion due to principal wind directions, as illustrated in Figure 2(b). The fifth principal component is an isotropic dispersion. For the other principal components, it is more difficult to identify a physical or chemical driving process.

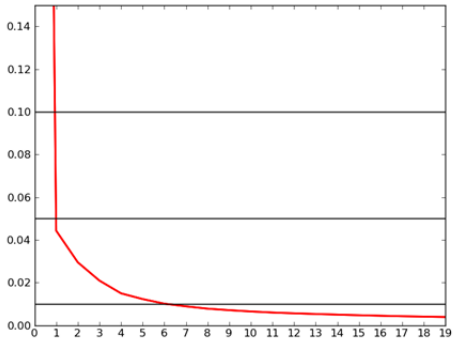

Figure 1. Unexplained variance against the number of selected principal components.

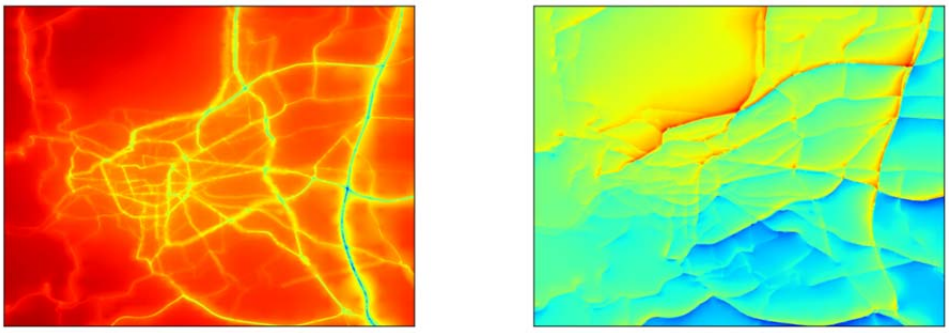

Figure 2. First principal component on left (a) and third principal component on right (b).

We compare the simulation to its projection on the 8 principal components over the full year 2008. The fields are correlated at 99% and unbiased. The root mean square error (RMSE) is of 2.37 µg m^{-3}, which represents about 10% of the mean simulated concentration of 22.6 µg m^{-3}.

Now we compare the projection of the full-year simulation to the observations. The performance is similar to the non-projected simulation. The correlation with the observations is 0.65, and the RMSE is 18.0 µg m^{-3}.

Note that the projection basis can be determined with a smaller temporal sequence of ADMS Urban outputs. For instance, the basis can be determined with just the first 6 months of the year, and applied to the next 6 months, without significant changes in performance.

Emulation

We rely on M = 1000 samples to build the 8 emulators (one for each projection coefficient). We use a linear regression and the mean residuals from the 7 closest neighbors. The metric coefficient θ_k is determined with leave-one-out cross-validation by the optimization algorithm COBYLA (Powell, 1994), which stands for Constrained Optimization BY Linear Approximations.

We compare the simulation to its approximation after reconstruction on the 8 principal components over the full year 2008. The fields are correlated at 86%. The bias is of 1.4 µg m^{-3}. The RMSE is of 7.9 µg m^{-3} with a mean emulated concentration of 20.6 µg m^{-3}, which represents more than 30%. This high value is

moderated by the analysis of errors distributions. The distribution of the scores (computed per date) of the emulation are shown in Figure 3. Most of the approximate fields over the year have intermediate but still high scores.

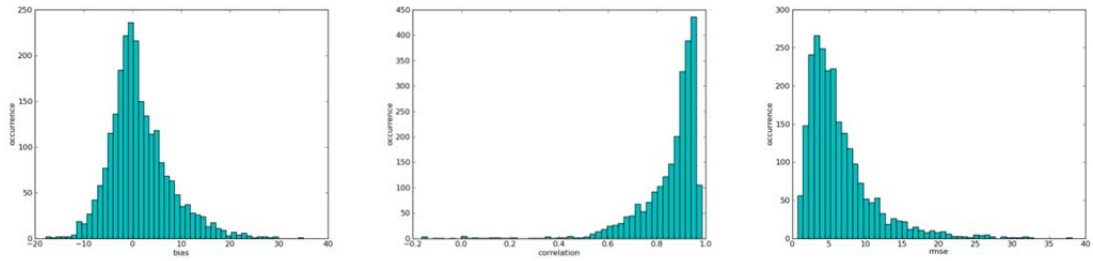

Figure 3. Distributions of scores (per date) of the emulation compared to the full simulation: (a) bias on left, (b) correlation on center and (c) RMSE on right.

Compared to observations, the performance of the approximated fields is slightly deteriorated compared to the performance of the simulation. The relative RMSE increases by about 4% and the correlation decreases by about 1%. The RMSE of the emulated fields is 19.1 µg m^{-3}. The distribution of scores, illustrated in Figure 4, show the same trend as described above: the simulation obtains the highest scores, but the emulation is more competitive for intermediate but still acceptable scores.

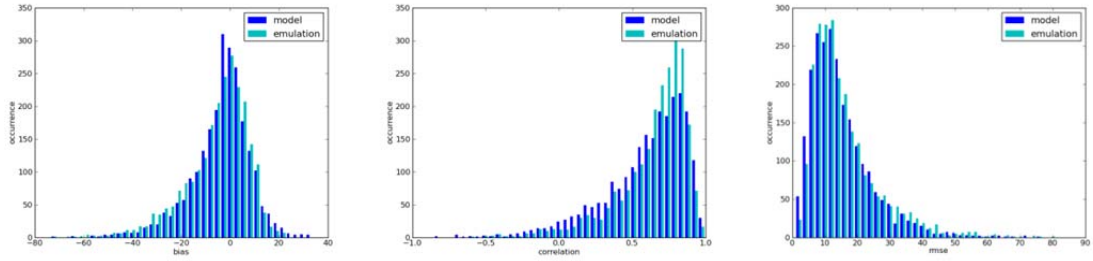

Figure 4. Distributions of scores to observations: (a) bias on left, (b) correlation on center and (c) RMSE on left for the simulation in blue and for the emulation in cyan.

CONCLUSION

In this paper, ADMS Urban has been replaced by a so-called emulator, after a reduction step based on principal component analysis. For a full year simulation of 3-hour NO_2 concentrations, the estimated computational cost to build the emulator is less than the full year simulation: the principal component analysis may require about 6 months of simulation, and M = 1000 samples (i.e., the equivalent of a 4-month simulation) were used to build the coefficient emulators. The emulator can become a useful, if not essential, tool for uncertainty quantification over long time periods. Indeed, one can then generate a huge ensemble of simulations over a full year, with negligible cost, while it would be intractable with the full model. It may even be possible to approximate the full probability density function of the concentrations, after large Monte Carlo simulations.

The emulator may be used in a context of impact studies (i.e., focusing on yearly or monthly averages) and in an operational context. It could easily be combined with data assimilation (Tilloy et al., 2013). Another application could be inverse modeling of emissions where the computational cost of the model is a key constraint. More generally, the availability of an emulator will allow to investigate directions and new methods that were previously out of reach because of the computational cost of the model.

REFERENCES

Joseph, V. R. and L. Kang: Regression-based inverse distance weighting with applications to computer experiments. *Technometrics*, **53**, 254–265, 2011.

Powell, M. J. D.: A direct search optimization method that models the objective and constraint functions by linear interpolation. In S. Gomez and J. P. Hennart, editors, *Advances in Optimization and Numerical Analysis*, pages 51–67. Kluwer Academic, 1994.

Sacks, J., W. J. Welch, T. J. Mitchell and H. P. Wynn: Design and analysis of computer experiments. *Statistical Science*, **4**, 409–423, 1989.

Tilloy, A., V. Mallet, D. Poulet, C. Pesin, and F. Brocheton: Blue-based NO2 data assimilation at urban scale. *Journal of Geophysical Research*, **118**, 1–10, 2013.