Reduction and Emulation of ADMS Urban

Anne Tilloy$^{1,2}$, Vivien Mallet$^{1,2}$, David Poulet$^3$ and Fabien Brocheton$^3$

$^1$ INRIA

$^2$ CEREA, joint laboratory École des Ponts ParisTech - EDF R&D, Université Paris-Est

$^3$ Numtech

May 5, 2013
Introduction

- ADMS Urban is an air quality static model at urban scale.
- Input vector $p \in \mathbb{R}^K$: meteorological variables, background concentrations, hour of the day, Julian day.
- $p$ varies from one simulated hour to the next.
- High-dimensional output concentration vector $y = \mathcal{M}(p)$.

But a full-year simulation of NO$_2$ concentrations over a city can take dozens of days of computation!

Objective

Replace ADMS Urban with an emulator.

Principle

- $y$ is first projected onto a reduced subspace.
- The relations between the projection coefficients and the input vector $p$ are then emulated.
Dimension reduction

- A reduced basis $[\Psi_1 \ldots \Psi_N]$ must represent the variability of the concentration field and is determined by Principal Component Analysis over a training period.
- In practice just few principal components are enough to provide a good approximation.

$$y \approx \sum_{j=1}^{N} \alpha_j \Psi_j$$

where $\alpha_j = y^T \Psi_j$ projection coefficient on $j$-th principal component.
Case study

The application is the simulation of NO$_2$ concentrations every 3 hours across the city Clermont-Ferrand (France) for the full year 2008.

**Input variables** $p$ contains $K = 10$ components.

- 5 meteorological input scalars from a meteorological station: wind speed, the wind direction, the temperature, the cloud coverage and the rain intensity.
- 3 background concentrations for NO$_2$, NO$_x$ and O$_3$.
- 2 emission variables: emissions vary according to the Julian day and the hour of the day (spatial distribution of the emissions and emissions factors are part of the model).
Dimension reduction

- We applied the principal component analysis to the full year 2008.
- Unexplained variance over the full year against the size of the projection basis $N$:

$$y \simeq \sum_{j=1}^{8} \alpha_j \psi_j$$

99% of the total variance with only $N = 8$ principal components.
$\Psi_1$ and $\Psi_3$: effects of mean emissions and wind from south.

**Scores**

The simulation over the full year 2008 is compared to its projection on 8 principal components.

<table>
<thead>
<tr>
<th>Mean $\mu g m^{-3}$</th>
<th>Bias $\mu g m^{-3}$</th>
<th>Corr. %</th>
<th>RMSE $\mu g m^{-3}$</th>
<th>Rel. RMSE %</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>0</td>
<td>99</td>
<td>2.37</td>
<td>10</td>
</tr>
</tbody>
</table>
Emulation

Every component \( f_j(p) = \mathcal{M}(p)^T \Psi_j \) is replaced by a statistical emulator \( \hat{f}_j \) whose computational cost is negligible.

- \( M \) training samples \( f_j(p^{(i)}) \) by latin hypercube sampling.
- The emulator in \( p \) is made of two parts:

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

\( \beta_{j,k} \) are the regression coefficients.

- Interpolation of the residuals:
  - Interpolation with radial basis functions.
  - Mean of the closest neighbors.
  - Kriging (computationally intensive).
Case study

The components $\alpha_j = y^T \Psi_j$ of the projection are emulated, and the emulator is applied over a full year.

Emulation

- $M = 2000$ samples to build the 8 emulators.
- Linear regression.
- Interpolation of residuals: radial basis function (Python module Scipy).

Scores

<table>
<thead>
<tr>
<th></th>
<th>Bias $\mu g \text{ m}^{-3}$</th>
<th>Corr. %</th>
<th>RMSE $\mu g \text{ m}^{-3}$</th>
<th>Rel. RMSE %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radial basis functions</td>
<td>$-0.2$</td>
<td>90</td>
<td>6.7</td>
<td>30</td>
</tr>
</tbody>
</table>
Comparison of ADMS Urban and the RBF emulator

200 frames over the year 2008

Demonstration of the RBF emulator
Conclusion

- The dimension reduction needs a 6-month simulation.
- Building the complete emulator requires the equivalent of a 8-month simulation.
- Then the prediction for any $p$ is essentially instantaneous!

Perspectives

- Improve the interpolation part of the emulator, with error control: inverse distance weighting or even kriging.
- Generate an ensemble of simulations for uncertainty quantification (Monte Carlo simulation).
- Operational air quality forecast: Urban Air System by Numtech.
- Impact studies: focusing on yearly or monthly averages.
- All the new methods that were previously out of reach because of the computational cost of the model!
\[ \hat{f}_j(p) = \sum_{k=1}^{K} \beta_{j,k} p_k + \sum_{i=1}^{M} w_{j,i}(p, \ldots, p^{(M)}) \left( f_j(p^{(i)}) - \sum_{k=1}^{K} \beta_{j,k} p_k^{(i)} \right) \]

Let the residual be \( r_j(p) = f_j(p) - \sum_{k=1}^{K} \beta_{j,k} p_k \), and the emulated residual be \( \hat{r}_j(p) = \hat{f}_j(p) - \sum_{k=1}^{K} \beta_{j,k} p_k \)

\[ \hat{r}_j(p) = \sum_{i=1}^{M} \alpha_{j,i} d(p, p^{(i)}) \] where \( d \) is a distance, here \( d \) is a metric in \( K \) dimensions.

We want to obtain the exact value at training points:
\[ \hat{r}_j(p^{(k)}) = r_j(p^{(k)}) = \sum_{i=1}^{M} \alpha_{j,i} d(p^{(k)}, p^{(i)}) \]

We denote \( \Delta_j \) the matrix whose element \((k, i)\) is \( d(p^{(k)}, p^{(i)}) \), and \( D_j(p) \) the vector whose \( i \)th component is \( d(p, p^{(i)}) \)

We get: \( \alpha_j = \Delta_j^{-1} R_j \) if \( R_j = (r_j(p^{(1)}), \ldots, r_j(p^{(M)}))^T \)

Note that \( \hat{r}_j(p) = \alpha_j^T D_j(p) = R_j^T \Delta_j^{-1} D_j(p) = D_j^T(p) \Delta_j^{-1} R_j \)

We therefore obtain: \( w_j(p) = D_j^T(p) \Delta_j^{-1} \).