Web Appendix A: Matlab code to find the Geodesic distance

As noted in Section 2.4 it is better to run the Floyd algorithm and obtain the matrix of Geodesic distances for the set of all locations for which we want to make predictions. Let $\tilde{N}$ be the number of prediction locations. There are two required arguments in the function $\text{floyd}(\text{dataset},K)$:

1. $\text{dataset}$: the $\tilde{N} \times 2$ data matrix with $x_i$ as its $i$-th row, $i = 1, \ldots, \tilde{N}$, the geographical coordinates.
2. $K$: the number of nearest neighbours, size of the restricted graph $G_k$. See Sections 2.3 and 2.4 for discussion on the choice of this value.

The output of the function is given by the matrices of geodesic and Euclidean distances, $\text{DistGeo}$ and $\text{DistEuclid}$, respectively.

```matlab
function [DistGeo,DistEuclid]=floyd(dataset,K)

% Step 0: Set up initial parameters
[N,d] = size(dataset);
if K+2 > N;
    disp('The number of nearest neighbours is too large');
end;

% Step 1: Calculate the pairwise Euclidean distances
DistEuclid=reshape(sqrt(sum((kron(dataset',ones(1,N)) - repmat(dataset',1,N)).^2)),N,N);

% Step 2: Find the K-th nearest neighbour
INF= 1000*max(max(DistEuclid))*.N;
DistGeo=DistEuclid;
[OrderDist, ind] = sort(DistGeo);
for i=1:N
    DistGeo(i,ind((K+2):end,i)) = INF;
end
```
DistGeo = min(DistGeo, DistGeo');
%
% Step 3: Find the shortest path
%
for k = 1:N
    DistGeo = min(DistGeo, repmat(DistGeo(:,k),1,N) + repmat(DistGeo(k,:),N,1));
end
return
Web Table 1

Minimum, maximum and mean value of observed mercury, predictions and related standard deviations together with the three quartiles.

<table>
<thead>
<tr>
<th></th>
<th>Min.</th>
<th>1st.Qu.</th>
<th>Median</th>
<th>Mean</th>
<th>3rd.Qu.</th>
<th>Max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>observed mercury</td>
<td>0.010</td>
<td>0.110</td>
<td>0.210</td>
<td>0.232</td>
<td>0.300</td>
<td>0.766</td>
</tr>
<tr>
<td>predictions at obs locations</td>
<td>0.163</td>
<td>0.172</td>
<td>0.186</td>
<td>0.232</td>
<td>0.267</td>
<td>0.399</td>
</tr>
<tr>
<td>st.dev at obs locations</td>
<td>0.006</td>
<td>0.011</td>
<td>0.020</td>
<td>0.018</td>
<td>0.021</td>
<td>0.031</td>
</tr>
<tr>
<td>predictions at nonobs locations</td>
<td>0.250</td>
<td>0.260</td>
<td>0.262</td>
<td>0.271</td>
<td>0.268</td>
<td>0.436</td>
</tr>
<tr>
<td>st.dev at nonobs locations</td>
<td>0.148</td>
<td>0.148</td>
<td>0.148</td>
<td>0.148</td>
<td>0.148</td>
<td>0.152</td>
</tr>
</tbody>
</table>
Web Figure 1. Knot selection for the estuary data based on space filling algorithms; ○ = data and × = knots.
Web Figure 2. Estimation Bias for LTPS, FELS and GLTPS with $k = 3$ for the test function on a U-shaped domain.
Web Figure 3. Estimation Bias for FELS and GLTPS with $k = 3$ for the test function on a U-shaped domain.
Web Figure 4. Estimated maps of mercury concentration: top panel, LTPS; bottom panel, GLTPS ($k = 5$).