

Simulation of Ocean, Atmosphere and Climate

Project: Inverse Modelling

Matthias Aengenheyster and Mark Dekker

`m.aengenheyster@students.uu.nl`

October 3, 2016

Contents

1	Chemical transport model	1
2	Forward Model	2
3	Inverse Model	4

In this project we use a chemical transport model and its inverse to infer the strength and location of emission sources based on the observed concentration timeseries at a discrete number of stations.

1 Chemical transport model

We use the following one-dimensional chemical transport model for a tracer C that is emitted by sources $E(x)$, advected by the wind u_0 and decays with rate constant k :

$$\frac{\partial C}{\partial t} = E - u \frac{\partial C}{\partial x} - kC \quad (1)$$

We use periodic boundary conditions in a domain of length L . We obtain a characteristic advection timescale $T = L/u_0$ that is required for a signal to propagate once through the domain back to its initial position. For constant and space-independent sources $E(x) = \text{const.}$ this model can be solved analytically for the constant steady state solution $C = \frac{E}{k}$.

We for now let all parameters to be time-independent and choose the following values:

L (m)	Δx (m)	T_{tot} (s)	Δt (s)	u_0 (m s ⁻¹)	k (s ⁻¹)
100	1	1000	0.1	5	0.1

Table 1: Model parameters

These choices result in the Courant number

$$\gamma = u_0 \frac{\Delta t}{\Delta x} = 0.5 \quad (2)$$

The source distribution $E(x)$ is given by n point sources at a number of discrete locations

$$E(x) = \sum_{i=0}^n E_i \delta(x - x_i) \quad (3)$$

where the E_i are constant in time.

2 Forward Model

In order to numerically compute the solution of the model we discretize it using an Upwind scheme with Forward Euler in time.

$$c_j^{n+1} = c_j^n - \gamma(c_j^n - c_{j-1}^n) + E_j \Delta t - k \Delta t c_j^n \quad (4)$$

$$= b_{-1} c_{j-1}^n + b_0 c_j^n + b_{+1} c_{j+1}^n - k_j^n \Delta t c_j^n + E_j \Delta t \quad (5)$$

$$b_{-1} = \gamma \quad (6)$$

$$b_0 = 1 - \gamma \quad (7)$$

$$b_{+1} = 0 \quad (8)$$

As we have chosen $k = \text{const.}$ we absorb it into $b_0 = 1 - \gamma - k \Delta t$, combine the advection and decay terms into the matrix \mathbf{M} and write

$$\vec{c}^{n+1} = M \vec{c}^n + \vec{E} \Delta t \quad (9)$$

$$M = \begin{pmatrix} b_0 & 0 & & & b_{-1} \\ b_{-1} & b_0 & \ddots & & \\ 0 & b_{-1} & b_0 & \ddots & \\ & \ddots & \ddots & \ddots & 0 \\ 0 & & 0 & b_{-1} & b_0 \end{pmatrix} \quad (10)$$

where the entry in the upper right corner implements the periodic boundary conditions and $b_{+1} = 0$ due to the discretization. \vec{c}^n and \vec{E} are vectors of length nx (the number of spatial cells $nx = L/\Delta x$) and \mathbf{M} is a $nx \times nx$ matrix. The so established model can be easily and efficiently integrated by repeated application of equation (9). However we observe that due to the sources it is not possible to obtain the solution for timestep n \vec{c}^n directly from the initial step \vec{c}^0 . It will prove useful to rewrite the model in a way that this becomes possible.

We introduce the new state vector \vec{x} and timestep matrix \mathbf{F} :

$$\vec{x}^n = \begin{pmatrix} \vec{c}^n \\ \vec{E} \end{pmatrix} \quad (11)$$

$$\mathbf{F} = \begin{pmatrix} \mathbf{M} & \mathbf{T} \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \quad (12)$$

$$\mathbf{T} = \begin{pmatrix} \Delta t & & \\ & \ddots & \\ & & \Delta t \end{pmatrix} \quad (13)$$

$$\mathbf{I} = \begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{pmatrix} \quad (14)$$

We see that we can now write the model (9) as

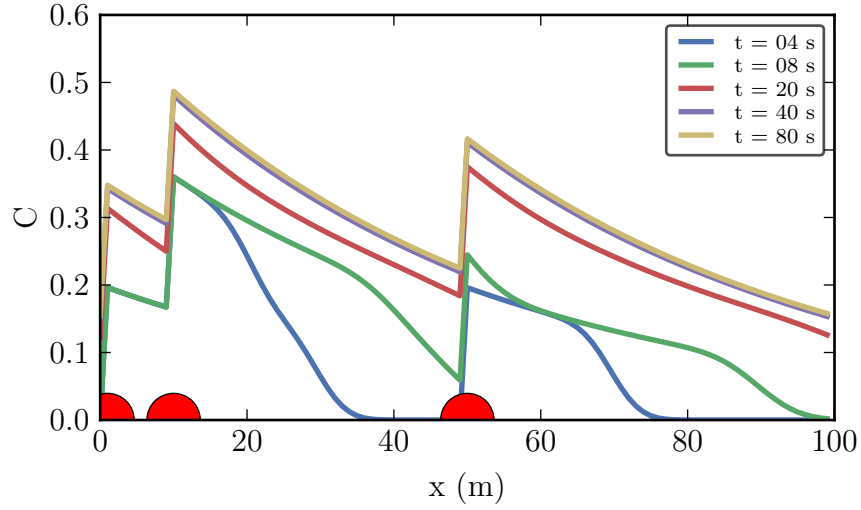
$$\vec{x}^{n+1} = \mathbf{F} \vec{x}^n \quad (15)$$

and even more importantly we now have a easy relation between timesteps $n_1, n_2, n_2 > n_1$:

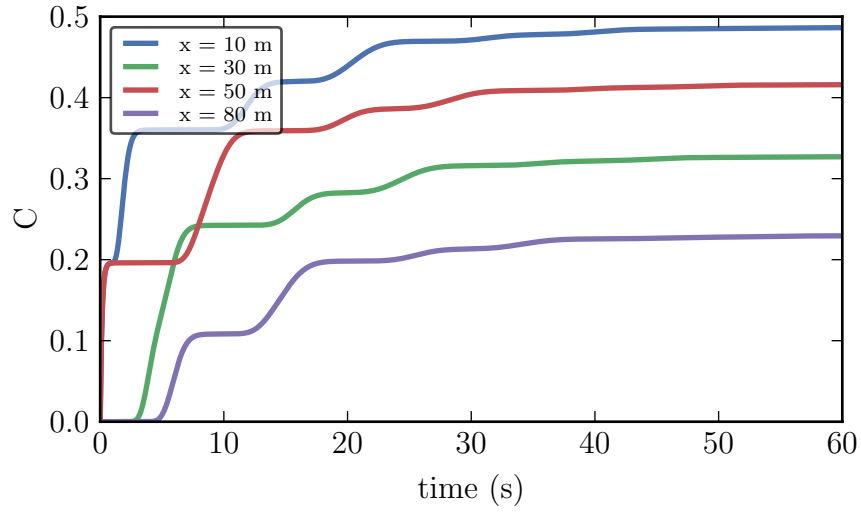
$$\vec{x}^{n_2} = \mathbf{F}^{n_2 - n_1} \vec{x}^{n_1} \quad (16)$$

with the power of a matrix defined as usual.

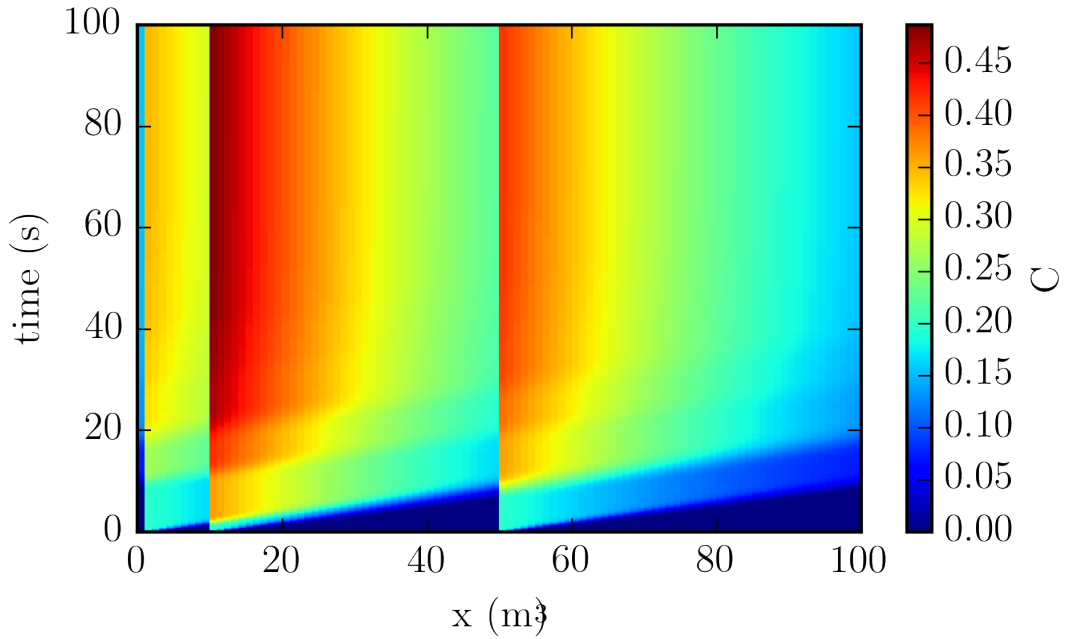
We integrate the model described above using a self-defined python Class. We show an exemplary simulation in figure (1)



(a) Concentration profile at various times. Source locations in red with $E_i = 1$



(b) Concentration as function of time at various locations.



(c) Concentration as function of time and space

Figure 1: Sample simulation. Top: $C(x)$ for various t . Bottom: $C(t)$ for various x

3 Inverse Model

The purpose of the inverse model is now to reconstruct the sources from the timeseries of the observations, i.e. get the red dots (and ideally also their strength) in figure (1a) from the data in figure (1b). This is essentially an exercise in linear algebra. We define a cost function J that takes into account the deviation of the prediction from the actual result and the deviation of the found parameters from their a priori guessed values. This is written as

$$J(\vec{x}) = (\vec{x} - \vec{x}_a)^T \mathbf{S}_a^{-1} (\vec{x} - \vec{x}_a) + (\vec{y} - \mathbf{K}\vec{x})^T \mathbf{S}_\epsilon^{-1} (\vec{y} - \mathbf{K}\vec{x}) \quad (17)$$

The scalar function J gives the cost of the parameter set \vec{x} depending on the prior guess \vec{x}_a , the observed data \vec{y} , the model matrix \mathbf{K} relating \vec{x}, \vec{y} as

$$\vec{y} = \mathbf{K}\vec{x} + \vec{\epsilon} \quad (18)$$

with model uncertainty term $\vec{\epsilon}$, and the error covariance matrices for the observations \mathbf{S}_ϵ and the prior guess \mathbf{S}_a . \vec{x} is found by minimizing J with respect to \vec{x} , so

$$\frac{\partial J}{\partial \vec{x}} = 2\mathbf{S}_a^{-1}(\vec{x} - \vec{x}_a) + 2\mathbf{K}^T \mathbf{S}_\epsilon^{-1}(\mathbf{K}\vec{x} - \vec{y}) = 0 \quad (19)$$

With the matrix \mathbf{G} as

$$\mathbf{G} = (\mathbf{K}^T \mathbf{S}_\epsilon^{-1} \mathbf{K} + \mathbf{S}_a^{-1})^{-1} \mathbf{K}^T \mathbf{S}_\epsilon^{-1} \quad (20)$$

the solution is found as

$$\vec{\hat{x}} = \vec{x}_a + \mathbf{G}(\vec{y} - \mathbf{K}\vec{x}_a) \quad (21)$$

So it is obvious that the challenge now lies primarily in finding the particular matrix \mathbf{K} that relates the source strengths $x = \vec{E}$ to an observed time series $y = x_i(t)$.

$$x_{i,t} = \mathbf{K}\vec{x}^0 \quad (22)$$

It is clear that \mathbf{K} will be different for different spatial points i in order to get the correct timeseries from the (invariant) vector \vec{x}^0 . To define \mathbf{K} we use the matrix \mathbf{F} that we defined previously. We can clearly write

$$\vec{x}^n = \mathbf{F}^n \vec{x}^0 \quad (23)$$

so the n -th power of the matrix \mathbf{F} relates timestep n to the initial vector. Obviously the i -th element in the vector \vec{x}^n is related to the i -th row of the matrix \mathbf{F}^n . We now define the matrix $K_{tj}(i)$ as

$$K_{tj}(i) = (\mathbf{F}^t)_{ij} \quad (24)$$

$$x_{i,t} = \sum_{j=1}^J K_{tj}(i) x_j^0 \quad (25)$$

So, in order to obtain the timeseries for location x_i at grid point i for every timestep t we compute the t -th row of matrix \mathbf{K} by taking the i -th row of the t -th power of matrix \mathbf{F} . So \mathbf{K} is a matrix with N (number of timesteps) rows and $2J$ columns for \vec{x} as defined in (11). This is a very large matrix. In the light of the operations to be performed to obtain $\vec{\hat{x}}$ we aim to reduce its size. We note that the second half of the vector \vec{x} does not change as the sources are time-independent and the first half of the vector \vec{x}^0 is zero anyway as we start from zero concentrations¹. Therefore we define the matrix $\hat{\mathbf{K}}$ as the *right* half of matrix \mathbf{K} with size $N \times J$. We therefore have

$$c_{i,t} = \sum_{j=1}^J \hat{K}_{tj}(i) E_j \quad (26)$$

¹We may also in principle start with some arbitrary initial profile, but the solution will quickly converge to a steady state and the memory will be lost. The initial profile enters into the solution of \vec{c}^n as $\mathbf{M}^n \vec{c}^0$. As $\max(\mathbf{M}) < 1$ for n large enough this term will be insignificant relative to the \vec{E} -terms.

or, if we define the vector \vec{c}_i as having N elements for each timestep

$$\vec{c}_i = \hat{\mathbf{K}}(i)\vec{E} \quad (27)$$

which gives us the matrix required from equation (18).

In order to compute \mathbf{G} and \vec{x} we still need to define the error matrices \mathbf{S}_ϵ of size $N \times N$ and \mathbf{S}_a of size $J \times J$. We assume uncorrelated errors so both matrices are diagonal and we have no reason to assume variations in the observation error in time or the prior error in space. Hence we let

$$\mathbf{S}_\epsilon = \sigma_\epsilon \text{diag}(N) \quad (28)$$

$$\mathbf{S}_a = \sigma_a \text{diag}(J) \quad (29)$$

where $\text{diag}(n)$ is a diagonal identity matrix of size $n \times n$. With these ingredients computing \vec{x} is simply a computational exercise that can be easily solved using python's scipy matrix algebra.

However we have one more problem to solve: In the procedure we have laid out up to here we compute \vec{x} , that is, the sources \vec{E} , only using one timeseries of concentrations \vec{c}_i . This is clearly suboptimal as stations at different locations will be more or less sensitive to particular emission sources (most sensitive to the ones just upstream and least sensitive to the ones just downstream). Therefore we want to extend our method to incorporate multiple observational timeseries and compute the optimal \vec{E} using all available information. To tackle this problem we construct a new state vector \vec{b} using L timeseries at different points and the corresponding forward model $\tilde{\mathbf{K}}$:

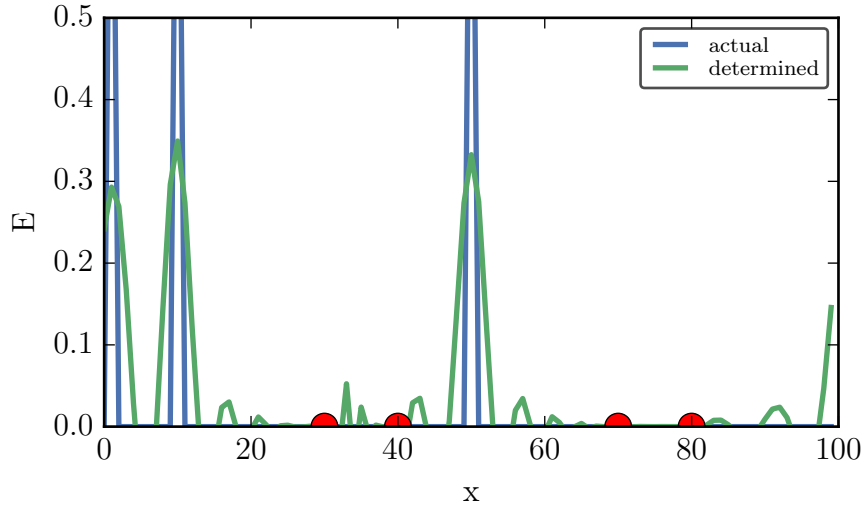
$$\vec{b} = \begin{pmatrix} c_{1,0} \\ \vdots \\ c_{1,N} \\ c_{2,0} \\ \vdots \\ c_{2,N} \\ \vdots \\ c_{L,N} \end{pmatrix}, \quad \tilde{\mathbf{K}} = \begin{pmatrix} \hat{\mathbf{K}}(1) \\ \hat{\mathbf{K}}(2) \\ \vdots \\ \hat{\mathbf{K}}(L) \end{pmatrix} \quad (30)$$

where \vec{b} now has length $L \cdot N$ and $\tilde{\mathbf{K}}$ has dimension $L \cdot N \times J$. The indices $1, 2, \dots, L$ label arbitrarily sorted distinct timeseries and their matrices $\hat{\mathbf{K}}$. We can therefore write the extended forward problem as

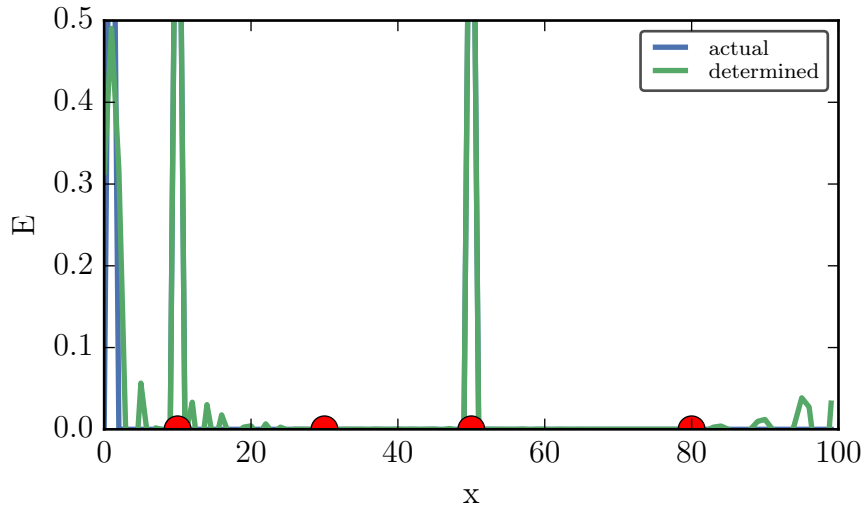
$$\vec{b} = \tilde{\mathbf{K}}\vec{E} \quad (31)$$

With this we can now use the same machinery introduced at the beginning of this section to compute \vec{x} , i.e. the best estimate for \vec{E} , based on the observations.

In figure (2) we show the recovered sources for the simulation shown in figure (1) where we have chosen $\sigma_\epsilon = 2 \times 10^{-8}$, $\sigma_a = 0.01$ starting with an initial guess $x_a = \vec{0}$ and using the timeseries from the indices 30,40,70,80 (2a) or 10,30,50,80 (2b) (the timeseries for the latter case where shown in figure 1b). We can see that the source locations are recovered in both cases. The quality of the reconstruction depends strongly on the selected sites. In the second case the agreement is very good, likely because two of the stations corresponded to emission sites.



(a)



(b)

Figure 2: Best estimate for sources (green) and actual sources (blue) for two different sets of four measurement stations (red)