# LECTURES ON INVERSE MODELING

## by Daniel J. Jacob, Harvard University

## January 2007

# FOREWORD

These lectures on inverse modeling are part of a half-course, "Models of Atmospheric Transport and Chemistry", that I have taught to graduate students at Harvard since 2002. Most of that course focuses on the construction of chemical transport models, but I also cover inverse modeling as it relates to atmospheric chemistry applications: retrieving concentrations from satellite radiance measurements, retrieving emissions from observed concentrations, chemical data assimilation. Originally I dedicated just one lecture to inverse modeling, but this has grown to about four lectures as my interest and experience in this area have grown.

I first educated myself in inverse modeling in 2000 through a reading course with my former student Colette L. Heald, in which we worked through the book by Clive D. Rodgers, "*Inverse Methods for Atmospheric Sounding*" (World Scientific, 2000). I found it to be wonderfully rigorous and clear, but difficult. . My lectures are heavily influenced by this book but try to be easily accessible to a beginning graduate student with a basic background in linear algebra. I have made a point of using Rodgers' notation because I consider it a model of clarity and in order to encourage the interested reader to go to his book for further information and insights.

I would greatly appreciate feedback from readers on any topics that need to be clarified, any errors, and any general issues of content. Send me an email at djacob@fas.harvard.edu.

I would like to thank my students and postdocs who have worked on inverse modeling as part of their research and educated me in the process: Colette L. Heald, Emily Jin, Dylan B. Jones, Monika A. Kopacz. Paul I. Palmer. I would like to extend a special thanks to Monika Kopacz for her comments on the adjoint modeling chapter.

Daniel J. Jacob

January 2007

## 1. INTRODUCTION

*Inverse modeling* is a formal approach for estimating the variables driving the evolution of a system by taking measurements of the observable manifestations of that system, and using our physical understanding to relate these observations to the driving variables. We call the variables that we wish to estimate the *state variables*, and assemble them into a *state vector* **x**. We similarly assemble the observations into an *observation vector* **y.** Our understanding of the relationship between **x** and **y** is described by a physical model **F,** called the *forward model***:**

$$\mathbf{y} = \mathbf{F}(\mathbf{x}, \mathbf{b}) + \boldsymbol{\varepsilon} \tag{1.1}$$

where **b** is a *parameter vector* including all model variables that we do not seek to optimize (we call them *model parameters*), and $\boldsymbol{\varepsilon}$ is an error vector including contributions from errors in the observations, in the forward model, and in the model parameters. From inversion of equation (1.1), we can obtain **x** given **y**. In the presence of error ($\boldsymbol{\varepsilon} \neq \mathbf{0}$), the best that we can achieve is a statistical estimate, and we need to weigh the resulting information against our prior (*a priori*) knowledge $\mathbf{x_a}$ of the state vector before the observations were made. The optimal solution of **x** reflecting this ensemble of constraints is called the *a posteriori,* the *optimal estimate,* or the *retrieval*. We will use these words interchangeably. The choice of state vector (i.e., which variables to include in **x** vs. in **b**) depends on what variables we wish to optimize, what information is contained in the observations, and what computational costs are associated with the inversion.

Inverse models have three major applications in atmospheric chemistry:

1. **Retrieval of atmospheric concentrations from observed radiances.** Consider the problem of using nadir spectra measured from space to retrieve the vertical profile of a trace gas. The measured radiances at different wavelengths represent the observation vector **y**, and the trace gas concentrations at different vertical levels represent the state vector **x.** The forward model solves the radiative transfer equation to calculate **y** as a function of **x** and of additional parameters **b** including surface emissivity, temperatures, clouds, spectroscopic data, etc. Inversion of this forward model using the observed spectra then provides a retrieval of **x**.
2. **Optimal estimation of surface fluxes.** Consider the problem of quantifying surface fluxes of a gas on a (latitude x longitude x time) grid. The fluxes on that grid represent the state vector **x**. We make observations of atmospheric concentrations of the gas from a network of sites, representing an observation vector **y**. The forward model is a chemical transport model (CTM) relating **x** to **y**. The parameter vector **b** includes meteorological variables and any characteristics of the surface flux (such as diurnal variability) that are simulated in the CTM but not resolved in the state vector. The information on **x** from the observations is called a *top-down* constraint on the surface fluxes. *A priori* information on **x** based on our knowledge of the processes

determining the fluxes (such as fuel combustion statistics, land type data bases, etc) is called a *bottom-up* constraint. Combination of the top-down and bottom-up constraints provides the optimal estimate of **x**. Instead of the surface fluxes themselves, we may wish to optimize the driving variables of a surface flux model; the approach is exactly the same but the dimension of **x** may be greatly reduced.

3. **Chemical data assimilation.** Consider the problem of constructing a continuous 3-D field of concentrations of a trace gas over the globe on the basis of limited measurements of concentrations at isolated points and scattered times. Such a construction may be useful to produce chemical forecasts, to assess the consistency of measurements made from different platforms, or to improve estimates of the concentrations of non-measured species from measurements of chemically linked species. We define the state vector $\mathbf{x}(t)$ as the 3-D ensemble of gridded concentrations at time $t$, and **y** as the vector of observations available over the time interval $[t-\Delta t, t]$. The forward model is a CTM initialized with $\mathbf{x}(t-\Delta t)$ and providing a *forecast* for time $t$. This forecast represents our *a priori* information for $\mathbf{x}(t)$, to be corrected on the basis of the observations over the time interval $[t-\Delta t, t]$. Our state vector here is in general very large, while the observation vector at any given time is relatively sparse. We refer to this type of inverse model application as *data assimilation.*

Proper consideration of errors is crucial in inverse modeling. To appreciate this, let us examine what happens if we ignore errors. We linearize the forward model **y = F(x, b)** around the *a priori* estimate $\mathbf{x_a}$ taken as first guess**:**

$$\mathbf{y} = \mathbf{F}(\mathbf{x_a}, \mathbf{b}) + \mathbf{K}(\mathbf{x} - \mathbf{x_a}) + \mathbf{O}((\mathbf{x} - \mathbf{x_a})^2) \qquad (1.2)$$

where $\mathbf{K} = \partial \mathbf{y} / \partial \mathbf{x}$ is the *Jacobian matrix* of the forward model with elements $k_{ij} = \partial y_i / \partial x_j$ evaluated at $\mathbf{x} = \mathbf{x_a}$**.** Let $n$ and $m$ be the dimensions of **x** and **y**, respectively. In the absence of error, $m = n$ independent measurements constrain **x** uniquely. The Jacobian matrix is then a $n$x$n$ matrix of full rank and hence invertible. We obtain for **x**:

$$\mathbf{x} = \mathbf{x_a} + \mathbf{K}^{-1}(\mathbf{y} - \mathbf{F}(\mathbf{x_a}, \mathbf{b})) \qquad (1.3)$$

If **F** is nonlinear, the solution (1.3) must be iterated with recalculation of the Jacobian around successive guesses for **x** until satisfactory convergence is achieved.

Now what happens if we make additional observations, such that $m > n$? In the absence of error these observations must necessarily be redundant. However, we know from experience that useful constraints on an atmospheric system typically require a very large number of measurements, $m >> n$. This reflects errors in the observations and in the forward model, described by the error vector $\boldsymbol{\varepsilon}$ in equation (1.1). Thus equation (1.3) is not applicable in practice; successful inversion requires adequate characterization of the error and consideration of *a priori* information on **x**. The *a priori* estimate has its own error:

$$\mathbf{x_a} = \mathbf{x} + \boldsymbol{\varepsilon_a} \qquad (1.4)$$

and the inverse problem then involves weighting the error statistics of $\boldsymbol{\varepsilon}$ and $\boldsymbol{\varepsilon}_a$ to solve the optimal estimation problem, "what is the best estimate of **x** given **y**?". This is done using Bayes' theorem, presented below.

## 2. BAYES' THEOREM

Bayes' theorem provides the general foundation for inverse models. Consider a pair of vectors **x** and **y.** Let $P(\mathbf{x})$, $P(\mathbf{y})$, $P(\mathbf{x},\mathbf{y})$ represent the corresponding probability distribution functions (*pdfs*), so that the probability of **x** being in the range [**x, x+dx**] is $P(\mathbf{x})\mathbf{dx}$, the probability of **y** being in the range [**y, y+dy**] is $P(\mathbf{y})\mathbf{dy}$, and the probability of (**x, y**) being in the range [**x, x+dx, y, y+dy**] is $P(\mathbf{x},\mathbf{y})\mathbf{dxdy}$. Let $P(\mathbf{y}|\mathbf{x})$ represent the pdf of **y** when **x** is assigned to a certain value. We can write $P(\mathbf{x},\mathbf{y})d\mathbf{x}d\mathbf{y}$ equivalently as

$$P(\mathbf{x},\mathbf{y})d\mathbf{x}d\mathbf{y} = P(\mathbf{x})d\mathbf{x}P(\mathbf{y}|\mathbf{x})d\mathbf{y} \tag{2.1}$$

or as

$$P(\mathbf{x},\mathbf{y})d\mathbf{x}d\mathbf{y} = P(\mathbf{y})d\mathbf{y}P(\mathbf{x}|\mathbf{y})d\mathbf{x} \tag{2.2}$$

Eliminating $P(\mathbf{x},\mathbf{y})$, we obtain Bayes' theorem:

$$P(\mathbf{x}\,|\,\mathbf{y}) = \frac{P(\mathbf{y}\,|\,\mathbf{x})P(\mathbf{x})}{P(\mathbf{y})} \tag{2.3}$$

This theorem formalizes the inverse problem posed in chapter 1. Here $P(\mathbf{x})$ is the pdf of the state vector **x** before the measurements are made (that is, the *a priori* pdf). $P(\mathbf{y}|\mathbf{x})$ is the pdf of the observation vector **y** given the true value for **x,** which the instrument knows about. $P(\mathbf{x}|\mathbf{y})$ is the *a posteriori* pdf for the state vector reflecting the information from the measurements – that is, it is the pdf of **x** given the measurements **y**. The optimal or *maximum a posteriori (*MAP*)* solution for **x** is given by the maximum of $P(\mathbf{x}|\mathbf{y})$, that is, the solution to $\nabla_{\mathbf{x}} P(\mathbf{x}\,|\,\mathbf{y}) = \mathbf{0}$ where $\nabla_{\mathbf{x}}$ is the the gradient operator in the state vector space. The probability function $P(\mathbf{y})$ in the denominator of (2.3) is independent of **x**, and we can view it merely as a normalizing factor to ensure that $\int_0^\infty P(\mathbf{x}\,|\,\mathbf{y})\mathbf{dx} = 1$. It plays no role in determining the MAP solution (since it is independent of **x**) and we ignore it in what follows.

## 3. INVERSE PROBLEM FOR SCALARS

Application of Bayes' theorem to obtain the MAP solution is easiest to first understand using scalars. Consider a source releasing a species *X* to the atmosphere with

an emission flux *x*. We have an *a priori* estimate $x_a \pm \sigma_a$ for the value of *x*, where $\sigma_a^2$ is the error variance. For example, if *X* is emitted from a power plant, the *a priori* information would be based on knowledge of the type and amount of fuel being burned in that plant, any emission control equipment, etc. We set up a sampling site to measure the concentration of *X* downwind of the source. We measure a concentration $y \pm \sigma_i$ where $\sigma_i^2$ is the instrument error variance. We then use a CTM to obtain a relationship between *x* and *y* as

$$y = F(x) \pm \sigma_m \tag{3.1}$$

where $\sigma_m^2$ is the CTM error variance. Let us assume that the CTM relationship between *x* and *y* is linear. Let us further assume that the instrument and CTM errors are uncorrelated so that the corresponding variances are additive. The measured concentration *y* is then related to the *true* source *x* by

$$y = kx \pm \sigma_\varepsilon \tag{3.2}$$

where the coefficient *k* is obtained from the CTM, and $\sigma_\varepsilon^2$ is the *observational error variance* defined as the sum of the instrumental and CTM errors:

$$\sigma_\varepsilon^2 = \sigma_i^2 + \sigma_m^2 \tag{3.3}$$

The observational error includes the forward model error, so it is not purely from "observations". Think of it as the error in the *observing system* designed to place constraints on the state vector.

After making the measurement, we seek an improved estimate $\hat{x}$ of *x* that optimally accommodates the top-down constraint from the measurement and the bottom-up constraint from the *a priori*. We uses Bayes' theorem. Assuming Gaussian error distributions, we have

$$P(x) = \frac{1}{\sigma_a \sqrt{2\pi}} \exp[-\frac{(x - x_a)^2}{2\sigma_a^2}] \tag{3.4}$$

$$P(y \mid x) = \frac{1}{\sigma_\varepsilon \sqrt{2\pi}} \exp[-\frac{(y - kx)^2}{2\sigma_\varepsilon^2}] \tag{3.5}$$

Applying Bayes' theorem (2.3) and ignoring the normalizing terms that are independent of *x*, we obtain:

$$P(x \mid y) \sim \exp[-\frac{(x - x_a)^2}{2\sigma_a^2} - \frac{(y - kx)^2}{2\sigma_\varepsilon^2}] \tag{3.6}$$

Finding the maximum value for $P(x \mid y)$ is equivalent to finding the minimum in the *cost function J(x):*

$$J(x) = \frac{(x - x_a)^2}{\sigma_a^2} + \frac{(y - kx)^2}{\sigma_\varepsilon^2} \tag{3.7}$$

which is a least-squares cost function weighted by the variance of the error in the individual terms. It is called a $\chi^2$ cost function, and *J(x)* as formulated in equation (3.7) is called the $\chi^2$ statistic.

The optimal estimate $\hat{x}$ is the solution to $\partial J / \partial x = 0$, which is straightforward to obtain analytically*:*

$$\hat{x} = x_a + g(y - kx_a) \tag{3.8}$$

where *g* is a *gain factor* given by

$$g = \frac{k\sigma_a^2}{k^2\sigma_a^2 + \sigma_\varepsilon^2} \tag{3.9}$$

In (3.8), the second term on the right-hand side represents the correction to the *a priori* on the basis of the measurement *y*. The gain factor is the sensitivity of the retrieval to the observation: $g = \partial\hat{x} / \partial y$. We see from (3.9) that the gain factor depends on the relative magnitudes of $\sigma_a$ and $\sigma_\varepsilon/k$. If $\sigma_a << \sigma_\varepsilon/k$, then $g \to 0$ and $\hat{x} \to x_a$; the measurement is useless because the observational error is too large. If by contrast $\sigma_a >> \sigma_\varepsilon/k$, then $g \to 1/k$ and $\hat{x} \to y/k$; the measurement is so precise that it constrains the solution without recourse to the *a priori* information.

We can also express the retrieval $\hat{x}$ in terms of its proximity to the true solution *x*. We have

$$y = kx + \varepsilon \tag{3.10}$$

where $\varepsilon$ (with variance $\sigma_\varepsilon^2$) is the observational error. Replacing in equation (3.8) we obtain

$$\hat{x} = ax + (1 - a)x_a + g\varepsilon \tag{3.11}$$

where $a = gk$ is an *averaging kernel* describing the relative weights of the *a priori* $x_a$ and the true value *x* in contributing to the retrieval. The averaging kernel represents the sensitivity of the retrieval to the true state: $a = \partial\hat{x} / \partial x$. The gain factor is now applied to the observational error in the third term on the right hand side. We see from equation (3.9) that the averaging kernel simply weights the error variances $\sigma_a^2$ and $(\sigma_\varepsilon/k)^2$. In the limit $\sigma_a >> \sigma_\varepsilon/k$, then $a \to 1$ and the *a priori* does not contribute to the solution. However, our ability to approach the true solution is limited by the third term $g\,\varepsilon$ in equation (3.11) with variance $(g\,\sigma_\varepsilon)^2$. Since in the above limit $g \to 1/k$, we obtain in fact $\hat{x} \to y/k$ as derived previously. The error on the retrieval is then defined by the observational error.

We call (1-*a*)$x_a$ the *smoothing error* since it limits the ability of the retrieval to obtain solutions departing from the *a priori*, and we call $g\varepsilon$. the *retrieval error.*

We can derive a general expression for the retrieval error variance by starting from equation (3.6) and expressing it in terms of a Gaussian distribution for the error in ($x$-$\hat{x}$). We thus obtain a form in $\exp[-(x-\hat{x})^2/2\hat{\sigma}^2]$ where $\hat{\sigma}^2$ is the variance of the error in the *a posteriori* $\hat{x}$. The calculation is laborious but straightforward, and yields

$$\frac{1}{\hat{\sigma}^2} = \frac{1}{\sigma_a^2} + \frac{1}{\left(\sigma_\varepsilon / k\right)^2} \tag{3.12}$$

Notice that the *a posteriori* error is always less than the *a priori* and observational errors, and tends towards one of the two in the limiting cases that we described.

Let us now consider a situation where our single measurement $y$ is not satisfactory in constraining the solution. We could in principle remediate this problem by making $m$ measurements $y_i$, each adding a term to the cost function (6.10). Assuming the same observational error variance for each measurement:

$$J(x) = \frac{(x-x_a)^2}{\sigma_a^2} + \sum_{i=1}^{m} \frac{(y_i - kx)^2}{\sigma_\varepsilon^2} \tag{3.13}$$

We can re-express $J(x)$ as

$$J(x) = \frac{(x-x_a)^2}{\sigma_a^2} + \frac{\left\langle (y-kx)^2 \right\rangle}{\sigma_\varepsilon^2 / m} \tag{3.14}$$

Where < > denotes the mean value and $\sigma_\varepsilon^2 / m$ is the variance of the error on the mean of $(y-kx)^2$ (this is the *central limit theorem).* By increasing $m$, we could thus approach the true solution: $m \to \infty \Rightarrow \hat{x} \to < y > / k$ and $\hat{\sigma} \to 0$. However, this works only if the observational error is (1) truly random, (2) uncorrelated between different measurements. With regard to (1), it is critical to establish if there is any systematic error (also called *bias*) in the measurement. In the presence of bias, no number of measurements will allow convergence to the true solution; the bias will be propagated through the gain factor and correspondingly affect the retrieval. Accurate calibration of the measuring instrument and of the forward model is thus essential. With regard to (2), instrumental errors (as from photon-counting) are often uncorrelated; however, forward model errors rarely are. For example, two successive measurements at a site may sample the same air mass and thus be subject to the same transport error in the CTM used as forward model. It is thus important to determine the *error correlation* between the different measurements. This error correlation can best be described by assembling the measurements into a vector and constructing the *observational error covariance matrix*. Dealing with error correlations, and also dealing with multiple sources, requires that we switch to a vector-matrix notation in our formulation of the inverse problem. . We do so in the next section.

Onr last point before we move on. We assumed in the above a linear forward model $y = F(x) = kx$. What if the forward model is not linear? We can still calculate a MAP solution $\hat{x}$ as the minimum in the cost function (3.7), where we replace $kx$ by the nonlinear form $F(x)$. The error in this MAP solution is not Gaussian though, so equation (3.12) would not apply. An alternative is to linearize the forward model around $x_a$ as $k = \left.\frac{\partial y}{\partial x}\right|_{x = x_a}$ to obtain an initial guess $x_1$ of $\hat{x}$, and then iterate on the solution by recalculating $k = \left.\frac{\partial y}{\partial x}\right|_{x = x_1}$, and so on. As we will see, the latter is the only practical solution as we move from scalar to vector space.

## 4. VECTOR-MATRIX TOOLS FOR INVERSE MODELING

Let us now consider the problem of a state vector **x** of dimension *n* with *a priori* value $\mathbf{x_a}$ for which we seek an improved estimate on the basis of an ensemble of observations assembled into an observation vector **y** of dimension *m*. The forward model is

$$\mathbf{y} = \mathbf{F}(\mathbf{x}) + \boldsymbol{\varepsilon} \tag{4.1}$$

as in (1.1) but without the model parameters to simplify notation. Inverse analysis requires definition of error statistics and pdfs for vectors, and of the Jacobian matrix for the forward model. The error statistics are provided by *error covariance matrices*, and the pdfs are constructed in a manner that accounts for covariance between vector elements. Numerical construction of the Jacobian matrix may be done using either the forward model or its *adjoint*. We begin by describing these different objects before proceeding to the solution of the inverse problem in the following sections.

### 4.1 Error covariance matrices

The error covariance matrix for a vector is the analog of the variance for a scalar. Consider a vector $\mathbf{x} = (x_1,\ldots x_n)^T$ of dimension *n*. Its error covariance matrix **S** has as diagonal elements the error variances of the individual elements of **x,** and as off-diagonal elements the error covariances between elements of **x.** Let us construct the *a priori* error covariance matrix $\mathbf{S_a}$ of the state vector for the inversion, i.e., the analog of $\sigma_a$ in the scalar problem (section 6.2). The *a priori* value for the state vector is $\mathbf{x_a}$ and the true value is **x.** The error variance var $(x_i - x_{a,i})$ for element $x_i$ is defined as the expected value of $(x_i - x_{a,i})^2$ when $x_i$ is sampled over its *a priori* pdf $P(x_i)$. The error covariance $\text{cov}((x_i - x_{a,i}), (x_j - x_{a,j}))$ for the pair $(x_{,i}, x_j)$ is defined as the expected value of the product $(x_i - x_{a,i})(x_j - x_{a,j})$ when $x_i$ and $x_j$ are sampled over their respective *a priori* pdfs. The matrix is thus constructed as:

$$\mathbf{S_a} = \begin{pmatrix} \mathrm{var}(x_1 - x_{a,1}) & \cdots & \mathrm{cov}(x_1 - x_{a,1}, x_n - x_{a,n}) \\ \vdots & \ddots & \vdots \\ \mathrm{cov}(x_1 - x_{a,1}, x_n - x_{a,n}) & \cdots & \mathrm{var}(x_n - x_{a,n}) \end{pmatrix} \tag{4.2}$$

We express it in compact mathematical form as $\mathbf{S}_a = E[(\mathbf{x} - \mathbf{x_a})(\mathbf{x} - \mathbf{x_a})^T]$ where $E$ is the *expected value operator* returning the expected value of the quantity, i.e., its average value over a large number of determinations.

Constructing an accurate error covariance matrix requires detailed statistical information and knowledge that is often difficult to obtain. Depending on the problem, simple estimates of errors may be sufficient.

*Example.* If we assume a uniform 50% error on the individual elements of $\mathbf{x_a}$ with no correlation between the errors on the different elements, then the diagonal elements of $\mathbf{S_a}$ are $0.25x_{a,i}^2$ and the off-diagonal elements are all zero.

Let us now similarly construct the observational error covariance matrix $\mathbf{S_\varepsilon}$ of the error vector $\boldsymbol{\varepsilon}$ in the forward model (4.1) used to relate $\mathbf{x}$ to $\mathbf{y}$:

$$\mathbf{y} = \mathbf{F}(\mathbf{x}) + \boldsymbol{\varepsilon} \tag{4.3}$$

$\mathbf{S_\varepsilon}$ is constructed as

$$\mathbf{S}_\varepsilon = \begin{pmatrix} \mathrm{var}(\varepsilon_1) & \cdots & \mathrm{cov}(\varepsilon_1, \varepsilon_n) \\ \vdots & \ddots & \vdots \\ \mathrm{cov}(\varepsilon_1, \varepsilon_n) & \cdots & \mathrm{var}(\varepsilon_n) \end{pmatrix} \tag{4.4}$$

and we can express it in compact form as $\mathbf{S_\varepsilon} = E[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^T]$. Included in $\boldsymbol{\varepsilon}$ are all the sources of error that would prevent the forward model from reproducing the observations. They can be separated into instrument errors ($\boldsymbol{\varepsilon_i}$) and forward model errors ($\boldsymbol{\varepsilon_m}$):

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon_i} + \boldsymbol{\varepsilon_m} \tag{4.5}$$

These errors are in general uncorrelated so the corresponding error covariance matrices are additive:

$$\mathbf{S}_\varepsilon = \mathbf{S}_{\varepsilon_i} + \mathbf{S}_{\varepsilon_m} \tag{4.6}$$

The instrument errors can be determined from calibration standards. The forward model errors are more difficult to estimate. They include errors in the model representations of all processes not correctable through adjustment of the state vector.

*Example.* Consider the problem of inverting $CO_2$ surface fluxes for individual continents on the basis of an ensemble of worldwide $CO_2$ atmospheric observations at surface sites, and using an Eulerian CTM as the forward model. Contributions to the forward model error will include:

1. The model *transport error*;
2. Spatial and temporal smoothing intrinsic to the model (grid resolution, time step), preventing it from resolving fine-scale variability in the observations – this is called the *representation error*;
3. Error in the distribution of *a priori* $CO_2$ surface fluxes on subcontinental scales that is not corrected by adjustment of the state vector on continental scales – this is called the *aggregation error*.

The error covariance matrix is in general a complicated object to interpret. Eigendecomposition can be a useful tool to identify the dominant error patterns. In the The error covariance matrix **S** for a vector **x** has full rank, since otherwise would imply that an element is perfectly known. It is also symmetric since the covariance operator is symmetric. It therefore has orthonormal eigenvectors $\mathbf{e_i}$ with eigenvalues $\lambda_i$. Eigenanalysis of **S** is useful to diagnose the dominant error patterns. **S** can be *spectrally decomposed* along its eigenvectors as

$$\mathbf{S} = \sum_{\mathbf{i}} \lambda_{\mathbf{i}} \mathbf{e_i} \mathbf{e_i}^T \tag{4.7}$$

Consider now the base for **x** defined by the eigenvectors. In that base, eigenvector $\mathbf{e_i}$ has a value of 1 for its $i^{th}$ element and a value of zero for all its other elements, so that the error covariance matrix is a diagonal matrix of eigenvalues:

$$\mathbf{S} = \begin{pmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{pmatrix} \tag{4.8}$$

The eigenvalue $\lambda_i$ thus represents the error variance associated with the orthonormal error pattern $\mathbf{e_i}$. By eigenvalue decomposition of **S** and ranking of eigenvalues, one can identify the dominant error patterns and the corresponding variances.

## 4.2 Gaussian probability distribution functions for vectors

Application of Bayes' theorem (chapter 2) requires formulation of probability distribution functions for vectors. We derive here the general Gaussian pdf for a vector **x** of dimension *n* with expected value <**x**> and error covariance matrix **S**. If the errors on the individual elements of **x** were uncorrelated (i.e., if **S** were diagonal), then the pdf of the vector would simply be the product of the pdfs for the individual elements. This simple solution can be achieved by transforming **x** to the basis of eigenvectors $\mathbf{e_i}$ of **S**. Let

us assemble the eigenvectors as the columns of a matrix **E;** then $\mathbf{z} = \mathbf{E}^T$ **(x - <x>)** is the transformed value of **x - <x>** in the eigenvector basis**.** The pdf of **z** is then

$$P(\mathbf{z}) = \prod_i \frac{1}{(2\pi\lambda_i)^{1/2}} \exp[-\frac{z_i^2}{2\lambda_i}] \tag{4.9}$$

The determinant of a matrix is the product of its eigenvalues:

$$|\mathbf{S}| = \prod_{\mathbf{i}} \lambda_{\mathbf{i}} \tag{4.10}$$

Further defining **Λ**as the diagonal matrix of eigenvalues, we can rewrite (4.9) as:

$$P(\mathbf{z}) = \frac{1}{(2\pi)^{n/2} |\mathbf{S}|^{1/2}} \exp[-\frac{1}{2}\mathbf{z}^T \mathbf{\Lambda}^{-1}\mathbf{z}] \tag{4.11}$$

and replace **z**:

$$P(\mathbf{x}) = \frac{1}{(2\pi)^{n/2} |\mathbf{S}|^{1/2}} \exp[-\frac{1}{2}\mathbf{(x- < x >)}^T \mathbf{E\Lambda^{-1}E}^T \mathbf{(x- < x >)}] \tag{4.12}$$

The spectral decomposition (4.7) of **S** can be expressed in terms of **E** as

$$\mathbf{S = E\Lambda E}^T \tag{4.13}$$

Since **S** is a symmetric matrix, $\mathbf{E}^T = \mathbf{E}^{-1}$ , and therefore

$$\mathbf{S^{-1} = E\Lambda^{-1}E}^T \tag{4.14}$$

resulting in the general pdf expression for vector **x**:

$$P(\mathbf{x}) = \frac{1}{(2\pi)^{n/2} |\mathbf{S}|^{1/2}} \exp[-\frac{1}{2}\mathbf{(x- < x >)}^T \mathbf{S^{-1}(x- < x >)}] \tag{4.15}$$

### 4.3 Jacobian matrix

The Jacobian matrix is a linearization of the forward model that enables application of matrix algebra to the inverse problem. It represents the sensitivity of the observation variables **y** to the state variables **x**, assembled in matrix form:

$$\mathbf{K} = \nabla_{\mathbf{x}}\mathbf{F} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}} \tag{4.16}$$

with individual elements $k_{ij} = \partial y_i / \partial x_j$. If the forward model is linear, then **K** does not depend on **x** and fully describes the forward model for the purpose of the inversion. If the forward model is not linear, then **K** is a function of **x** and represents a linearization of the forward model around **x**. It needs to be calculated initially for the *a priori* value $\mathbf{x_a}$, representing the initial guess for **x**, and then re-calculated as needed for updated values of **x** during iterative convergence to the solution. Depending on the degree of non-linearity, **K** may not need to be re-calculated at each iteration.

Construction of the Jacobian matrix may be done analytically if the forward model is simple, as in a 0-D chemical model where the evolution of concentrations is determined by local reaction rates. If the forward model is complicated, such as a 3-D CTM, then the Jacobian must be constructed numerically. The standard approach, and the best to use if the dimension of the state vector is less than that of the observation vector ($n < m$), is to build the Jacobian matrix column by column by successively perturbing the individual elements $x_i$ of the state vector by small increments $\Delta x_i$, and applying the forward model to obtain the resulting perturbation $\mathbf{\Delta y}$. If the observations are sparse or the state vector is large so that $n > m$, then a more effective way to construct the Jacobian is through the model adjoint, as described below.

### 4.4 Model adjoint

The adjoint of a model is the transpose of its Jacobian matrix. It turns out to be very useful in inverse modeling applications where observed concentrations are used to constrain a state vector of sources or concentrations at previous times. We will discuss this in chapter 7. It can also be an efficient tool for numerical construction of the corresponding Jacobian matrix. Consider a CTM discretized over time steps $[t_0,\ldots t_i,\ldots t_n]$, and let $\mathbf{y_n}$ represent the vector of concentrations at time $t_{n,}$. We wish to determine its sensitivity to the state vector **x** at time $t_0$ (generalization to a time-invariant state vector or to state vector elements for different times will be shown later to be immediate). The corresponding Jacobian matrix is $\mathbf{K} = \partial \mathbf{y_n} / \partial \mathbf{x}$. By the chain rule,

$$\mathbf{K} = \frac{\partial \mathbf{y_n}}{\partial \mathbf{x}} = \frac{\partial \mathbf{y_n}}{\partial \mathbf{y_{n-1}}} \frac{\partial \mathbf{y_{n-1}}}{\partial \mathbf{y_{n-2}}} \ldots \frac{\partial \mathbf{y_1}}{\partial \mathbf{y_0}} \frac{\partial \mathbf{y_0}}{\partial \mathbf{x}} \tag{4.17}$$

where the RHS is a product of matrices. As discussed in section 4.3, the standard way to construct the Jacobian numerically is by successively perturbing the individual elements of **x** and applying the CTM to obtain the resulting perturbation $\mathbf{\Delta y}$. Another way is to take the transpose of the Jacobian matrix, i.e., the adjoint of the CTM:

$$\mathbf{K}^T = \left( \frac{\partial \mathbf{y_n}}{\partial \mathbf{y_{n-1}}} \frac{\partial \mathbf{y_{n-1}}}{\partial \mathbf{y_{n-2}}} \ldots \frac{\partial \mathbf{y_1}}{\partial \mathbf{y_0}} \frac{\partial \mathbf{y_0}}{\partial \mathbf{x}} \right)^T = \left( \frac{\partial \mathbf{y_0}}{\partial \mathbf{x}} \right)^T \left( \frac{\partial \mathbf{y_1}}{\partial \mathbf{y_0}} \right)^T \ldots \left( \frac{\partial \mathbf{y_{n-1}}}{\partial \mathbf{y_{n-2}}} \right)^T \left( \frac{\partial \mathbf{y_n}}{\partial \mathbf{y_{n-1}}} \right)^T \tag{4.18}$$

where we have made use of the property that the transpose of a product of matrices is equal to the product of the transposed matrices in reverse order. The adjoint model described by (4.18) offers an alternate way of constructing the Jacobian matrix by

applying successive perturbations $\Delta y_i$ to the individual elements of $\mathbf{y_n}$ and successively applying the operators $\left(\partial \mathbf{y_n} / \partial \mathbf{y_{n-1}}\right)^T$, $\left(\partial \mathbf{y_{n-1}} / \partial \mathbf{y_{n-2}}\right)^T$ …all the way to $\left(\partial \mathbf{y}_0 / \partial \mathbf{x}\right)^T$ to obtain the sensitivity $\Delta y_i / \mathbf{\Delta x}$ which is a row of the Jacobian matrix. Thus we construct the Jacobian matrix row by row, instead of column by column as previously. This approach is more economical if dim(**y**) < dim(**x**), because as we will see later the cost of running the adjoint model is comparable to that of running the forward model. A single run of the adjoint model from $t_n$ to $t_0$ can determine the sensitivity of $\mathbf{y_n}$ to state vector elements at different times, or to a time-invariant state vector – in the latter case, one sums the sensitivities to the state vector for the successive steps back in time.

Construction of the adjoint model $\mathbf{K}^T$ requires linearization of the forward model (CTM) to express it as a product of matrices that we can then transpose. Consider the matrices $\mathbf{Z_i} = \partial \mathbf{y_i} / \partial \mathbf{y_{i-1}}$ describing the evolution of the CTM over individual time steps [$t_{i-1}$, $t_i$], and the matrix $\mathbf{Z}_0 = \partial \mathbf{y}_0 / \partial \mathbf{x}$, such that

$$\mathbf{K}^T = \mathbf{Z_o}^T \mathbf{Z_1}^T \ldots \mathbf{Z_i}^T \ldots \mathbf{Z_n}^T \tag{4.19}$$

To determine the individual matrices $\mathbf{Z_i}$ we need to decompose the CTM in terms of its individual linearized operators. Consider a CTM with successive application of operators for emissions ($E$), chemistry ($C$), convection ($C_o$), and advection ($A$) over a model time step [$t_{i-1}$, $t_i$]. Each operator updates the concentration over the time step, and this update can be described by a matrix representing the linearized operator. For example, the update from advection can be written $\mathbf{y_i} = \nabla A_i \mathbf{y_{i-1}}$ where $\nabla A_i$ is the linearized advection operator. Thus we can write $\mathbf{Z_i}$ as a product of matrices representing the linearized operators:

$$\mathbf{Z}_i = \nabla A_i \nabla C_{o,i} \nabla C_i \nabla E_i \tag{4.20}$$

and its transpose is then

$$\mathbf{Z}_i^T = \nabla E_i^T \nabla C_i^T \nabla C_{o,i}^T \nabla A_i^T \tag{4.21}$$

Construction of $\mathbf{Z}_0 = \partial \mathbf{y}_0 / \partial \mathbf{x}$ is done in exactly the same way except that the linearization of the relevant CTM operator is done with respect to **x** rather than to **y**. In the common application where we are interested in the sensitivity of concentrations to emissions, the linearization with respect to **x** is done in the emission operator $\nabla E_i$. This is straightforward to do, as discussed below.

Construction of the adjoint of a forward model thus involves two steps. The first is the construction of a Tangent Linear Model (TLM) that linearizes the individual operators of the forward model. The second is the transposition of these linear operators. Construction of the TLM from non-linear operators is an arduous task and commercial software packages are available for this purpose.

For simple linear operators, however, construction of the adjoint model may be straightforward. Consider for example a linear advection operator in which the transport of mass from gridbox $j$ to gridbox $k$ over time step $[t_{i-1}, t_i]$ is described by the matrix coefficient $a_{jk}$ such that $\partial y_{k,i} / \partial y_{j,i-1} = a_{jk}$. In the transposed operator, this coefficient applies to the reverse flow from gridbox $k$ to gridbox $j$, that is, $\partial y_{j,i} / \partial y_{k,i-1} = a_{jk}$. We thus see that reversing the winds is all we need to do to obtain the transpose of a linear advection operator. This makes sense in terms of the general adjoint model philosophy of propagating sensitivities back in time. Even if the advection scheme is weakly non-linear, using reverse winds may be an acceptable approximation for the sake of ease in building the adjoint. The validity of the approximation can be tested as will be discussed in chapter 8.

As another example of simple adjoint construction, consider a linear chemical operator consisting of local first-order loss. The matrix for that operator is diagonal since there are no interactions between species or gridboxes. Transposition does not change the operator, which is then called *self-adjoint*; we can apply the original operator in the adjoint model. Again, this makes sense; if a species decays with a certain time constant, then the sensitivity of concentrations to conditions backward in time will decay with that same time constant. The emission operator is similarly self-adjoint when expressed in terms of the sensitivity of concentrations to emissions (i.e., for construction of the matrix $\mathbf{Z}_0 = \partial \mathbf{y}_0 / \partial \mathbf{x}$). It is a null matrix when expressed in terms of the sensitivity to concentrations for the previous time step (i.e., for construction of the matrix $\mathbf{Z_i} = \partial \mathbf{y_i} / \partial \mathbf{y_{i-1}}$).

## 5. INVERSE PROBLEM FOR VECTORS

The vector-matrix tools presented in chapter 4 allow us to apply Bayes's theorem to obtain an optimal estimate of a state vector $\mathbf{x}$ (dim $n$) on the basis of the observation vector $\mathbf{y}$ (dim $m$), the *a priori* information $\mathbf{x}_a$, the forward model $\mathbf{F}$, and the error covariance matrices $\mathbf{S_a}$ and $\mathbf{S_\varepsilon}$ (the reader is encouraged to return to chapter 3 as needed for simple application of Bayes' theorem to the scalar inversion problem, which helps develop intuition for the material presented in the present chapter). We need to linearize the forward model, if it is not already, in order to use matrix algebra.. This is done by Taylor expansion about the *a priori* value as described by (1.2), where $\mathbf{K} = \partial \mathbf{y} / \partial \mathbf{x} = \nabla_{\mathbf{x}} \mathbf{F(x)}$ is the Jacobian matrix. If the forward model is linear, $\mathbf{K}$ is invariant with $\mathbf{x}$. If it is not, then $\mathbf{K}$ must be calculated initially for $\mathbf{x} = \mathbf{x}_a$ and re-calculated iteratively as the inversion progresses. In this chapter we assume that the forward model is linear or has been linearized so that

$$\mathbf{y} = \mathbf{K}\mathbf{x} + \boldsymbol{\varepsilon} \tag{5.1}$$

where $\boldsymbol{\varepsilon}$ is the observational error vector previously introduced in chapter 1.

## 5.1 Analytical maximum a posteriori (MAP) solution

Folllowing the general pdf formulation for vectors (section 4.2), the pdfs from Bayes' theorem in chapter 2 are given by

$$-2\ln P(\mathbf{x}) = (\mathbf{x}-\mathbf{x_a})^T \mathbf{S_a}^{-1}(\mathbf{x}-\mathbf{x_a}) + c_1 \tag{5.2}$$

$$-2\ln P(\mathbf{y}\,|\,\mathbf{x}) = (\mathbf{y}-\mathbf{K}\mathbf{x})^T \mathbf{S}_\varepsilon^{-1}(\mathbf{y}-\mathbf{K}\mathbf{x}) + c_2 \tag{5.3}$$

$$-2\ln P(\mathbf{x}\,|\,\mathbf{y}) = (\mathbf{x}-\mathbf{x_a})^T \mathbf{S_a}^{-1}(\mathbf{x}-\mathbf{x_a}) + (\mathbf{y}-\mathbf{K}\mathbf{x})^T \mathbf{S}_\varepsilon^{-1}(\mathbf{y}-\mathbf{K}\mathbf{x}) + c_3 \tag{5.4}$$

where $c_1$, $c_2$, $c_3$ are constants. The maximum *a posteriori* (MAP) solution is the value of $\mathbf{x}$ that yields the maximum of $P(\mathbf{x}|\mathbf{y})$, or equivalently the minimum of the scalar-valued cost function $J(\mathbf{x})$:

$$J(\mathbf{x}) = (\mathbf{x}-\mathbf{x_a})^T \mathbf{S_a}^{-1}(\mathbf{x}-\mathbf{x_a}) + (\mathbf{y}-\mathbf{K}\mathbf{x})^T \mathbf{S}_\varepsilon^{-1}(\mathbf{y}-\mathbf{K}\mathbf{x}) \tag{5.5}$$

To find this minimum, we solve for $\nabla_{\mathbf{x}} J(\mathbf{x}) = \mathbf{0}$:

$$\nabla_{\mathbf{x}} J(\mathbf{x}) = 2\mathbf{S_a}^{-1}(\mathbf{x}-\mathbf{x_a}) + 2\mathbf{K}^T \mathbf{S}_\varepsilon^{-1}(\mathbf{K}\mathbf{x}-\mathbf{y}) = \mathbf{0} \tag{5.6}$$

The solution is straightforward and can be expressed in compact form as

$$\hat{\mathbf{x}} = \mathbf{x_a} + \mathbf{G}(\mathbf{y}-\mathbf{K}\mathbf{x_a}) \tag{5.7}$$

with $\mathbf{G}$ given by

$$\mathbf{G} = \mathbf{S_a}\mathbf{K}^T(\mathbf{K}\mathbf{S_a}\mathbf{K}^T + \mathbf{S}_\varepsilon)^{-1} \tag{5.8}$$

or equivalently by:

$$\mathbf{G} = (\mathbf{K}^T\mathbf{S}_\varepsilon^{-1}\mathbf{K} + \mathbf{S_a}^{-1})^{-1}\mathbf{K}^T\mathbf{S}_\varepsilon^{-1} \tag{5.9}$$

(the second form is more expeditious to compute if $m > n$). $\mathbf{G}$ is the *gain matrix* and describes the sensitivity of the retrieval to the observations, i.e., $\mathbf{G} = \partial\hat{\mathbf{x}}/\partial\mathbf{y}$.

The error covariance matrix $\hat{\mathbf{S}}$ of $\hat{\mathbf{x}}$ can be calculated as in chapter 3 for the scalar problem by rearraging the right-hand side of (5.4) to be of the form $(\mathbf{x}-\hat{\mathbf{x}})^T\hat{\mathbf{S}}^{-1}(\mathbf{x}-\hat{\mathbf{x}})$. The algebra is straightforward and yields

$$\hat{\mathbf{S}} = (\mathbf{K}^T\mathbf{S}_\varepsilon^{-1}\mathbf{K} + \mathbf{S_a}^{-1})^{-1} \tag{5.10}$$

Note the similarity of our equations for $J(\mathbf{x})$, $\hat{\mathbf{x}}$, $\mathbf{G}$, $\hat{\mathbf{S}}$ to those derived for scalars in chapter 3.

We pointed out in the scalar problem the danger of over-interpreting the apparent reduction in error variance on the state vector from $\sigma_a$ to $\hat{\sigma}$ as a result of accumulating a large number of observations. The same concern applies here. The reduction in error from $\mathbf{S_a}$ to $\hat{\mathbf{S}}$ assumes that the observational error is truly random and that error covariances in the observations are fully accounted for. These requirements are often not satisfied, in which case $\hat{\mathbf{S}}$ will underestimate the actual *a posteriori* error. An often more realistic way of assessing the error in $\hat{\mathbf{x}}$ is through an *ensemble* of inverse calculations with various perturbations to model parameters, observational values, and covariance error estimates within their expected uncertainties.

## 5.2 Averaging kernel matrix

A useful way to express the ability of an observational system to constrain the true value of the state vector is with the *averaging kernel matrix* $\mathbf{A} = \partial\hat{\mathbf{x}}/\partial\mathbf{x}$, representing the sensitivity of the MAP solution $\hat{\mathbf{x}}$ to the true state $\mathbf{x}$. $\mathbf{A}$ is the product of the gain matrix $\mathbf{G} = \partial\hat{\mathbf{x}} / \partial\mathbf{y}$ and the Jacobian matrix $\mathbf{K} = \partial\mathbf{y} / \partial\mathbf{x}$:

$$\mathbf{A} = \mathbf{GK} \tag{5.11}$$

Replacing (5.11) and (5.1) into (5.7) we obtain an alternate form of the MAP solution:

$$\hat{\mathbf{x}} = \mathbf{Ax} + (\mathbf{I_n} - \mathbf{A})\mathbf{x_a} + \mathbf{G\varepsilon} \tag{5.12}$$

where $\mathbf{I_n}$ is the identity matrix of dimension *n*. Note the similarity to (3.11) in the scalar problem. $\mathbf{A}$ is a weighting factor for the relative contributions to the retrieval from the true state vs. the *a priori* estimate. $\mathbf{Ax}$ represents the contribution of the true state to the solution, $(\mathbf{I_n} - \mathbf{A})\mathbf{x_a}$ represents the contribution from the *a priori*, and $\mathbf{G\varepsilon}$ represents the contribution from the random observational error mapped onto state space by the gain matrix $\mathbf{G}$. A perfect observational system would have $\mathbf{A} = \mathbf{I_n}$. We call $(\mathbf{I_n} - \mathbf{A})\mathbf{x_a}$ the *smoothing error* (because it smoothes the solution towards the *a priori*) and $\mathbf{G\varepsilon}$ the *retrieval error*. Equation (5.12) also provides an alternate expression for the MAP error covariance matrix $\hat{\mathbf{S}}$:

$$\begin{aligned}\hat{\mathbf{S}} &= E[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T] = E((\mathbf{I_n} - \mathbf{A})(\mathbf{x} - \mathbf{x_a})(\mathbf{x} - \mathbf{x_a})^T(\mathbf{I_n} - \mathbf{A})^T) + E[\mathbf{G\varepsilon\varepsilon}^T\mathbf{G}] \\ &= (\mathbf{I_n} - \mathbf{A})\mathbf{S_a}(\mathbf{I_n} - \mathbf{A})^T + \mathbf{GS_\varepsilon G}^T\end{aligned} \tag{5.13}$$

from which we see that $\hat{\mathbf{S}}$ can be decomposed into the sum of a *smoothing error covariance matrix* $(\mathbf{I_n} - \mathbf{A})\mathbf{S_a}(\mathbf{I_n} - \mathbf{A})^T$ and a *retrieval error covariance matrix* $\mathbf{GS_\varepsilon G}^T$.

Algebraic manipulation yields an alternate form of the averaging kernel matrix as

$$\mathbf{A} = \mathbf{I_n} - \hat{\mathbf{S}}\mathbf{S_a^{-1}} \tag{5.14}$$

which shows how the relative reduction in error enabled by the observational system relative to the *a priori* provides improved knowledge of the state vector.

The averaging kernel matrix is a very useful thing to know about an observation system, and is essential for testing or comparing two different observation systems used to determine **x**. In the case of an analytical MAP solution involving explicit calculation of the Jacobian and gain matrices, as described above, **A** comes out of the solution analytically either by (5.11) or (5.14) (whichever form is most convenient to compute). Other approaches to the inverse problem, involving for example neural networks that fit **x** to **y** empirically on the basis of prior correlations, or the adjoint approach described in chapter 7 that solves numerically for $\nabla_{\mathbf{x}} J(\mathbf{x}) = \mathbf{0}$, do not provide averaging kernel matrices as part of their solutions. An averaging kernel matrix can still be constructed numerically column by column by (1) taking small perturbations $\Delta x_i$ to individual elements of the state vector, (2) applying the forward model to obtain the resulting perturbation $\mathbf{\Delta y}$, (3) applying observational error $\boldsymbol{\varepsilon}$ to $\mathbf{\Delta y}$, and (4) applying the retrieval to $\mathbf{\Delta y} + \boldsymbol{\varepsilon}$ to obtain $\Delta\hat{\mathbf{x}}$.

## 5.3 Pieces of information in an observing system

A concept related to the average kernel matrix is the *number of pieces of information* in an observing system towards constraining an *n*-dimensional state vector. The number of pieces of information is often called the number of degrees of freedom for signal (DOFS) with notation $d_s$. It can be determined as the reduction in the normalized error on **x** due to the measurement. We express the normalized error on **x** prior to the measurement with the $\chi^2$ cost function $J_a(\mathbf{x_a})$:

$$J_a(\mathbf{x}_a) = (\mathbf{x} - \mathbf{x_a})^T \mathbf{S_a}^{-1} (\mathbf{x} - \mathbf{x_a}) \tag{5.15}$$

which has an expected value of *n*, representing the number of pieces of information to be obtained for a perfect knowledge of the system. After the measurement has been made, the value of this cost function becomes

$$J_a(\hat{\mathbf{x}}) = (\mathbf{x} - \hat{\mathbf{x}})^T \mathbf{S_a}^{-1} (\mathbf{x} - \hat{\mathbf{x}}) \tag{5.16}$$

The DOFS is given by the difference in the expected values of $J_a(\mathbf{x_a})$ and $J_a(\hat{\mathbf{x}})$:

$$d_s = E[(\mathbf{x} - \mathbf{x_a})^T \mathbf{S_a}^{-1} (\mathbf{x} - \mathbf{x_a})] - E[(\mathbf{x} - \hat{\mathbf{x}})^T \mathbf{S_a}^{-1} (\mathbf{x} - \hat{\mathbf{x}})] = n - E[(\mathbf{x} - \hat{\mathbf{x}})^T \mathbf{S_a}^{-1} (\mathbf{x} - \hat{\mathbf{x}})] \tag{5.17}$$

The quantity $(\mathbf{x} - \hat{\mathbf{x}})^T \mathbf{S_a}^{-1} (\mathbf{x} - \hat{\mathbf{x}})$ is a scalar and is thus equivalent to its trace in matrix notation:

$$(\mathbf{x} - \hat{\mathbf{x}})^T \mathbf{S_a}^{-1} (\mathbf{x} - \hat{\mathbf{x}}) = \mathrm{tr}((\mathbf{x} - \hat{\mathbf{x}})^T \mathbf{S_a}^{-1} (\mathbf{x} - \hat{\mathbf{x}})) = \mathrm{tr}((\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T \mathbf{S_a}^{-1}) \tag{5.18}$$

where the last equality exploits the commutativity of the trace operator: tr(**AB**) = tr(**BA**). We thus obtain for the posterior value of $J_a$:

$$E[(\mathbf{x}-\hat{\mathbf{x}})^T \mathbf{S_a^{-1}}(\mathbf{x}-\hat{\mathbf{x}})] = E[\mathrm{tr}((\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^T \mathbf{S_a^{-1}})] = \mathrm{tr}(\hat{\mathbf{S}}\mathbf{S_a^{-1}}) \quad (5.19)$$

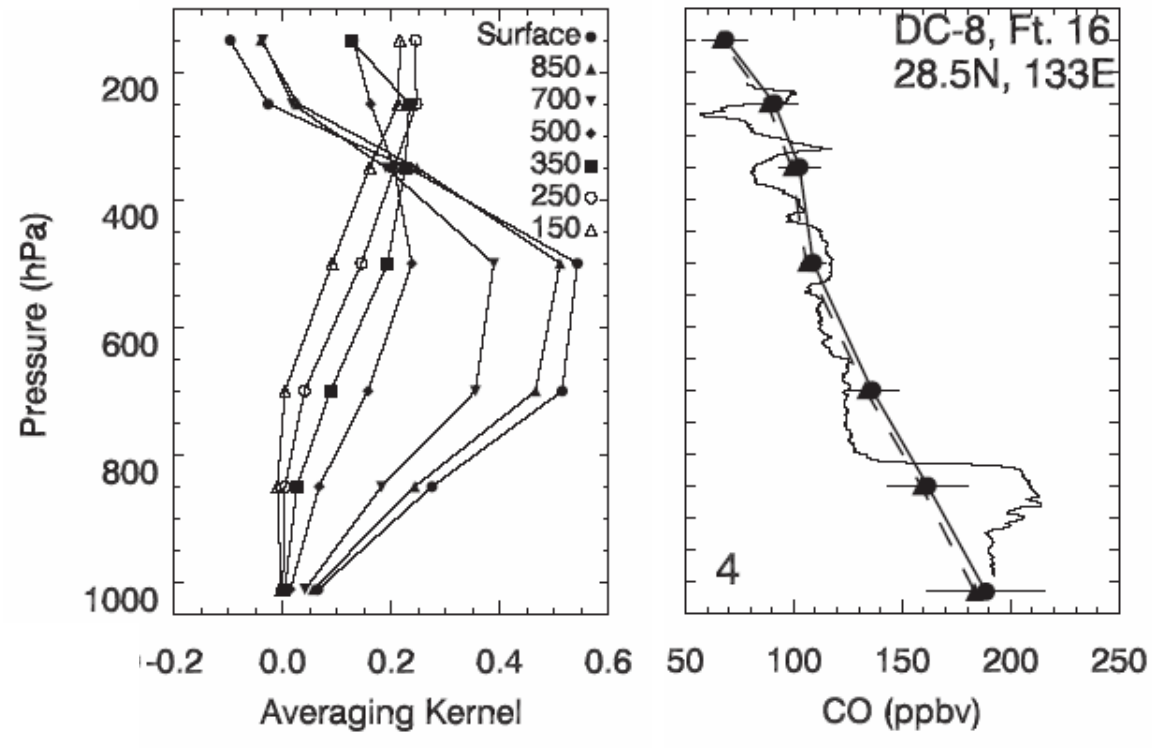
so that

$$d_s = n - \mathrm{tr}(\hat{\mathbf{S}}\mathbf{S_a^{-1}}) = \mathrm{tr}(\mathbf{I_n} - \hat{\mathbf{S}}\mathbf{S_a^{-1}}) = \mathrm{tr}(\mathbf{A}) \quad (5.20)$$

The number of pieces of information in an observing system (or degrees of freedom for signal) is given by the trace of the averaging kernel matrix.

## 5.4 Example application

The figure below (from Jacob et al., J. Geophys. Res. 2003) shows in its left panel a typical averaging kernel matrix for retrieval of vertical profiles of carbon monoxide (CO) mixing ratios from the MOPITT satellite instrument. This instrument makes nadir measurements of IR terrestrial emission around the 4.6 μm CO absorption band. The radiances measured at different wavelengths represent the observation vector for the inverse problem, and the CO mixing ratios at $n$ = 7 different vertical levels from the surface to 150 hPa represent the state vector. The averaging kernel matrix is represented in the figure row by row, i.e., each line with symbol represents one row of the averaging kernel matrix for the level indicated by the symbol, and describes the sensitivity of the retrieval at that level to the true CO mixing ratios at different altitudes.



Consider in this figure the retrieval of the CO mixing ratio at 700 hPa (inverted triangles). We see that the retrieved value is actually sensitive to CO at all altitudes, i.e., it is not possible from the retrieval to narrowly identify the CO mixing ratio at 700 hPa (or at any other specific altitude). The temperature contrast between vertical levels is not sufficient. We retrieve instead a broad CO column weighted toward the middle troposphere (700-500 hPa). In fact, the retrieval at 700 hPa is more sensitive to the CO mixing ratio at 500 hPa than at 700 hPa. Physically, this means that a certain mixing ratio of CO at 500 hPa will give a spectral response similar to a larger mixing ratio at 700 hPa, because 500 hPa has greater temperature contrast with the surface.

We also notice in the averaging kernel matrix some negative values, for example the retrieval at 700 hPa is negatively dependent on CO in the stratosphere at 150 hPa. This is not a physical result but is allowed by the MAP statistical fit; it is reflected in the *a posteriori* error covariance matrix $\hat{\mathbf{S}}$ by a negative correlation between the retrieved values at 700 and 150 hPa.

The trace of this particular averaging kernel matrix is 1.2, so that MOPITT provides 1.2 pieces of information on the vertical profile. One may thus expect good information on some vertically weighted column but not on gradients. Visual inspection of the left panel shows that the retrievals at the surface, 850, 700, and 500 hPa all give the same CO column weighted towards the middle troposphere, whereas the retrievals at 350, 250, and 150 hPa all give a similar weak signal in the upper troposphere and lower stratosphere. The largest piece of information in MOPITT is thus the CO column weighted toward the middle troposphere, with a smaller piece of information weighted toward higher altitudes.

The right panel of the Figure shows the application of the averaging kernel matrix to the validation of the MOPITT instrument with an underpass aircraft vertical profile extending from the surface to 200 hPa. The CO measurements from aircraft (thin solid line) have high accuracy and can be viewed as defining the true profile. Applying the averaging kernel matrix to this true profile (binned by the MOPITT retrieval levels) yields the dashed line; this is what MOPITT would see if its capability were as advertised by the error analysis that led to the averaging kernel matrix. We see that the vertical structure is largely lost, as would be expected (the vertical gradient is mainly from the *a priori*). This aircraft profile processed with the averaging kernel matrix can be then compared to the MOPITT retrieval, shown by the thick solid line. The two have similar vertical gradients but this merely reflects the *a priori* information. More instructive is that the columns are similar, with a 6% positive bias for MOPITT. This bias represents a systematic error in the retrieval that is not accounted for in the error analysis..

### 5.5 Sequential updating

Analytical derivation of the MAP solution using (5.7) or equivalently (5.12) requires construction of the Jacobian matrix $\mathbf{K} = \partial\mathbf{y}/\partial\mathbf{x}$ and of the gain matrix $\mathbf{G}$. Numerical construction of $\mathbf{K}$ requires a number $n$ of forward model calculations, to be possibly iterated if the forward model is nonlinear. The associated computational costs

can be tremendous and limit the manageable size of the state vector. As we will see in chapter 7, this difficulty can be addressed with an *adjoint approach* to the inverse problem.

The size *m* of the observational vector is also of concern because of the associated matrix multiplications involved in the construction of **G**. Limitations on *m* can however be circumvented within the framework of the analytical solution by using *sequential updating*. In this approach, the observation vector is partitioned into smaller "packets" of observations that are successively ingested into the inverse analysis. The MAP solution $(\hat{\mathbf{x}},\hat{\mathbf{S}})$ obtained after processing of one packet is then used as *a priori* for the next packet, and so on. The final solution is exactly the same as if the entire observation vector were ingested at once. The only limitation is that observations in different packets must be taken to be uncorrelated, i.e., $\mathbf{S}_\varepsilon$ for the ensemble of observations must be viewed as a block diagonal matrix where the blocks are the individual packets.

## 6. KALMAN FILTER ("3-D Var")

So far we have used inverse analysis of observations to constrain a fixed value of the state vector. In fact, we may want to use observations distributed in time to constrain a state vector evolving with time, subject to some *a priori* knowledge of this state vector and its evolution with time, and not solving for all times at once (which would quickly make the problem computationally intractable). A straightforward way to do this, commonly called the *Kalman filter* or "3-D Var", is to iterate in time the analytical MAP solution presented in section 5.1. "3-D" here refers to the use of observations at a given time step to constrain the state vector at that time step, subject to *a priori* information from prior or posterior times. "4-D Var", described in the next section, refers to the use observations at a given time to constrain the state vector over a range of times.

The Kalman filter can be run either forward or backward. We describe the forward filter first.. Consider an ensemble of observations collected at discrete time steps over an interval $[t_0, t_n]$. let $\mathbf{y_i}$ be the ensemble of observations collected at time $t_i$, and $\mathbf{x_i}$ the corresponding value of the state vector. Starting from *a priori* knowledge $(\mathbf{x_a}, \mathbf{S_a})$ at time $t_0$, we use the observations $\mathbf{y_0}$ and the analytical MAP solution described in section 5.1 to derive best estimates $(\hat{\mathbf{x}}_\mathbf{0},\hat{\mathbf{S}}_\mathbf{0})$ for that time. We then progress forward in time, using *a priori* knowledge of the time evolution of **x** expressed by a linear (or linearized) *evolution operator* $\mathbf{M_i}$ with error $\boldsymbol{\varepsilon}_\mathbf{M}$:

$$\mathbf{x_i} = \mathbf{M_i}\mathbf{x_{i-1}} + \boldsymbol{\varepsilon}_\mathbf{M} \qquad (6.1)$$

In the simplest case, we could assume persistence for **x**, in which case $\mathbf{M_i}$ would be the identity matrix. Consider a time $t_{i-1}$ for which we have the MAP solution $(\hat{\mathbf{x}}_{\mathbf{i-1}},\hat{\mathbf{S}}_{\mathbf{i-1}})$. We use (6.1) to derive an *a priori* value for $\mathbf{x_i}$ as

$$\mathbf{x_{a,i}} = \mathbf{M_i}\hat{\mathbf{x}}_{\mathbf{i-1}} \tag{6.2}$$

and the associated *a priori* error covariance matrix as

$$\mathbf{S_{a,i}} = E[\mathbf{M_i}\hat{\boldsymbol{\varepsilon}}\hat{\boldsymbol{\varepsilon}}^T\mathbf{M_i}^T] + E[\boldsymbol{\varepsilon}_\mathbf{M}\boldsymbol{\varepsilon}_\mathbf{M}^T] = \mathbf{M_i}\hat{\mathbf{S}}_{\mathbf{i-1}}\mathbf{M}_i^T + \mathbf{S_M} \tag{6.3}$$

where $\hat{\boldsymbol{\varepsilon}}$ is the error on $\hat{\mathbf{x}}_{\mathbf{i-1}}$ and $\mathbf{S_M}$ is the error covariance matrix for the evolution operator. We thus obtain time-dependent MAP solutions for $\mathbf{x}$ over the interval [$t_1$, $t_n$].

A problem with the forward filter for some applications is that the observations at time $t_i$ do not constrain the state vector at prior times. An alternative is to use a backward filter in which we start from *a priori* knowledge ($\mathbf{x_a}$, $\mathbf{S_a}$) at time $t_n$. The observations $\mathbf{y_n}$ at time $t_n$ are used to obtain a MAP solution for $\mathbf{x_n}$, and we then progress backward in time following the same procedure as with the forward filter. The only difference is that we must use an evolution operator $\mathbf{M_i}'$ running backward:

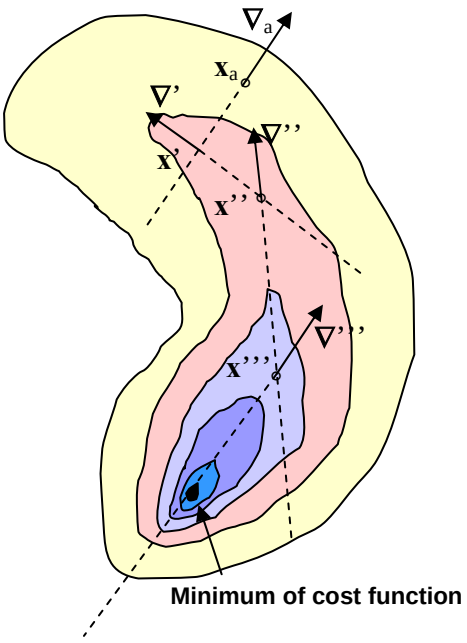
$$\mathbf{x_{i-1}} = \mathbf{M}_\mathbf{i}'\mathbf{x_i} + \boldsymbol{\varepsilon}_\mathbf{M}' \tag{6.4}$$

In the backward filter, observations at time $t_i$ are not allowed to constrain the state vector at posterior times. It is possible to run the Kalman filter for the same observational data set first forward, then backward, to allow observations to constrain the state vector in both temporal directions.

## 7. ADJOINT APPROACH ("4-D Var")

When a very large number of observations is available, as from satellites, we would like to use this information to constrain a very large state vector featuring high spatial and temporal resolution, commensurate with the detail in the observations and limited solely by the resolution of the forward model. This is impractical in the analytical solutions to the inverse problem described above, as these require explicit construction of the Jacobian matrix as well as multiplications of matrices having the dimension of the state vector. The *adjoint approach* addresses this difficulty through the application of the adjoint model.

The adjoint approach, like the analytical approach, seeks to minimize the cost function $J(\mathbf{x})$ given by (5.5), but it does so numerically rather than analytically. Starting from the initial guess $\mathbf{x_a}$, it computes the cost function gradient $\nabla_\mathbf{x} J(\mathbf{x})$ iteratively in combination with a steepest-descent numerical algorithm to find min($J(\mathbf{x})$). Standard steepest-descent algorithms are described in textbooks on numerical methods; a popular one is the BFGS algorithm. The figure below illustrates how a steepest-descent algorithm applied to successive guesses $\mathbf{x_a}$, $\mathbf{x'}$, $\mathbf{x''}$, $\mathbf{x'''}$…approaches min($J(\mathbf{x})$).




The main task in the adjoint approach is the efficient computation of $\nabla_{\mathbf{x}} J(\mathbf{x})$ at each iteration of the steepest-descent algorithm. $\nabla_{\mathbf{x}} J(\mathbf{x})$ is given by (equation (5.6)):

$$\nabla_{\mathbf{x}} J(\mathbf{x}) = 2\mathbf{S_a}^{-1}(\mathbf{x} - \mathbf{x_a}) + 2\nabla_{\mathbf{x}}\mathbf{F}^T \mathbf{S}_\varepsilon^{-1}(\mathbf{F(x)} - \mathbf{y}) \qquad (7.1)$$

where $\mathbf{F(x)}$ is the forward model (not necessarily linear), and $\nabla_{\mathbf{x}}\mathbf{F}^T = \mathbf{K}^T$ is the model adjoint (section 4.4) applied here to the vector $\mathbf{S}_\varepsilon^{-1}(\mathbf{F(x)} - \mathbf{y})$ which represents the weighted error in the ability of our guess for $\mathbf{x}$ to match the observations. The components of $\mathbf{S}_\varepsilon^{-1}(\mathbf{F(x)} - \mathbf{y})$ are called the *adjoint forcings.*

Implementation of the adjoint approach is as follows. We start from the *a priori* $\mathbf{x_a}$ as initial guess and make one pass of the forward model through the period [$t_0$, $t_n$] of the observational record of interest. Observations may be scattered over that period. We collect the corresponding values $\mathbf{S}_\varepsilon^{-1}(\mathbf{y} - \mathbf{F}(\mathbf{x_a}))$ of the adjoint forcings and calculate $\nabla_{\mathbf{x}} J(\mathbf{x_a})$ following (7.1):

$$\nabla_{\mathbf{x}} J(\mathbf{x_a}) = 2\mathbf{K}^T \mathbf{S}_\varepsilon^{-1}(\mathbf{F(x_a)} - \mathbf{y}) \qquad (7.2)$$

This calculation is done by applying the adjoint model to the adjoint forcings as described in section 4.4. We start from the adjoint forcings at time $t_n$ and then work backward in time, picking up additional adjoint forcings along the way, until we reach $t_0$. The variables through which the adjoint forcings are propagated backward in time with the adjoint model are called the *adjoint variables.* An important aspect of this calculation is that the Jacobian matrix is never explicitly constructed. The value of $\nabla_{\mathbf{x}} J(\mathbf{x_a})$ computed

from (7.2) is passed to the steepest-descent algorithm, which make an updated guess **x'**. We then recalculate $\nabla_{\mathbf{x}} J(\mathbf{x}')$ for that updated guess,

$$\nabla_{\mathbf{x}} J(\mathbf{x}') = 2\mathbf{S}_{\mathbf{a}}^{-1}(\mathbf{x}' - \mathbf{x}_{\mathbf{a}}) + 2\mathbf{K}^T \mathbf{S}_{\varepsilon}^{-1}(\mathbf{F}(\mathbf{x}') - \mathbf{y}) \tag{7.3}$$

pass the result to the steepest-descent algorithm which makes an updated guess **x''**, and so on. Each iteration thus involves one pass through the forward model over $[t_o, t_n]$ followed by one pass of the adjoint model over $[t_n, t_0]$. Forward and adjoint models typically have comparable computational requirements, and the requirements for the adjoint model are insensitive to the dimension of **x**. Increasing the dimension of **x** still entails some penalty, however, as it generally increases the number of iterations required for convergence.

The correctness of the cost function gradients $\nabla_{\mathbf{x}} J(\mathbf{x})$ produced by the inverse model can be checked with a simple finite-difference test and this is standard procedure. To do this test, apply the forward model to $\mathbf{x}_{\mathbf{a}}$ **,** calculate the cost function $J(\mathbf{x}_{\mathbf{a}})$, and repeat for a small perturbation $\mathbf{x}_{\mathbf{a}} + \Delta\mathbf{x}_{\mathbf{a}}$. The resulting finite-difference approximation

$$\nabla_{\mathbf{x}} J(\mathbf{x}_{\mathbf{a}}) \approx \frac{J(\mathbf{x}_{\mathbf{a}} + \Delta\mathbf{x}_{\mathbf{a}}) - J(\mathbf{x}_{\mathbf{a}})}{\Delta\mathbf{x}_{\mathbf{a}}} \tag{7.4}$$

can then be compared to the value obtained with the adjoint model.

## 8. OBSERVING SYSTEM SIMULATION EXPERIMENTS

After having designed an observation system and related inverse model to optimize the estimate of a state vector **x**, it is a good idea to test this machinery with an observing system simulation experiment (OSSE), both to test the value of the observing system and also to test that the inverse model is working properly. The OSSE consists of examining the ability of synthetically generated observations with the forward model (*pseudo-observations)* to retrieve values of **x** consistent with expectations.

In an OSSE, we start by selecting some reasonable but arbitrary value of **x** as the true value for the purpose of the test. We then generate pseudo-observations **y** following equation (4.3) by applying the forward model to **x** and adding a random noise **ε** consistent with our knowledge of $\mathbf{S}_{\boldsymbol{\varepsilon}}$**.** We then ask the question, how well can our observing system retrieve **x** given **y**? To this end we start from an *a priori* estimate $\mathbf{x}_{\mathbf{a}}$ constructed by apply random noise to **x** in a manner consistent with the *a priori* error covariance matrix $\mathbf{S}_{\mathbf{a}}$**.** From this *a priori* and using the pseudo-observations **y,** we then calculate a MAP solution $\hat{\mathbf{x}}$, compare it to the true value **x** to assess the merit of the observation system, and determine whether the two are consistent within the *a posteriori* error covariance matrix $\hat{\mathbf{S}}$ as a test of the inverse model**.**

OSSEs are routinely conducted in the design phase of a satellite mission to test whether the mission can satisfactorily address its scientific objectives. In that case,

the focus is on whether the MAP solution provides a significantly improved estimate of $\mathbf{x}$ relative to $\mathbf{x_a}$. The averaging kernel matrix $\mathbf{A}$ is a useful diagnostic. It will typically provide an overoptimistic assessment of the capability of the observing system, as discussed in chapter 5, because the errors used to generate the pseudo-data are random. But if the observing system fails that test then it should clearly be redesigned.