united nations
educational, scientific
and cultural
organization

international atomic
energy agency

the
abdus salam
international centre for theoretical physics

Course on "Inverse Methods in Atmospheric Science"
1 - 12 October 2001

301/1332-3

"Atmospheric Remote Sensing: The Inverse Problem"

Clive D. RODGERS
University of Oxford
UK

Please note: These are preliminary notes intended for internal distribution only.

Atmospheric Remote Sensing: The Inverse Problem

Inverse Methods in Atmospheric Science
Abdus Salam International Centre for Theoretical Physics

1st – 12th October 2001

Clive D Rodgers

University of Oxford

Atmospheric Remote Sensing: The Inverse Problem

Inverse Methods in Atmospheric Science
Abdus Salam International Centre for Theoretical Physics

1st – 12th October 2001

Clive D Rodgers

University of Oxford

Topics

1. Introduction
2. Bayesian approach
3. Information content
4. Error analysis and characterisation
5. Optimal methods
6. Nonlinear problems
7. Kalman Filter
8. Ad-hoc methods
9. Forward modelling and Jacobians

Appendix: Matrix algebra

Advertisment:

C. D. Rodgers, *Inverse Methods for Atmospheric Sounding: Theory and Practice*, World Scientific Publishing Co., 2000.

1.00

WHAT IS AN INVERSE OR RETRIEVAL PROBLEM?

- *Almost any measurement you make...*

  When you measure some function of the quantity you really want, you have a retrieval problem.

  Sometimes it's trivial, sometimes it isn't.

- *Various aspects:*

  – Formulate the problem properly:

  Describe the measurement in terms of some Forward Model

  Don't forget experimental error!

  – Finding a solution, inverting the forward model

  Algebraic

  Numerical

  No unique solution

  No solution at all

  – Finding the 'best' solution

  Uniqueness - a *unique* solution may not be the best...

  Accuracy

  Efficiency

  – Understanding the answer

1.01

THINGS TO THINK ABOUT

- Why isn't the problem trivial?

  – Forward models which are not explicitly invertible

  – Ill-conditioned or ill-posed problems

  – Errors in the measurement (and in the forward model) can map into errors in the solution in a non-trivial way.

- What to measure?

  – Does it actually contain the information you want?

- Updating existing knowledge

  – You always have *some* prior knowledge of the 'unknown'

  – the measurement improves that knowledge

  – the measurement may not be enough by itself to completely determine the unknown

- Ill-posed problems

  – You cannot solve an ill-posed problem. You have to convert it into a well-posed problem.

  – Which of an infinite manifold of solutions do you want?

1.02

MATHEMATICAL CONCEPTS I

Measurement Vector: $\mathbf{y} = (y_1, y_2, ...y_m)$

- Any measurement is of a finite number of quantities.
- Arrange them as a vector for computational purposes

State Vector: $\mathbf{x} = (x_1, x_2, ...x_n)$

- The desired quantity is often continuous - e.g. a temperature profile
- We can only make a finite number of measurements and calculations
- Express the unknown in terms of a finite number of parameters
- They do not all have to be of the same type
- Arrange them as a vector for computational purposes
- Examples:
  - Temperature on a set of pressure levels, with a specified interpolation rule.
  - Fourier coefficients for a set of waves

MATHEMATICAL CONCEPTS II

Using vectors, it is convenient to think in terms of linear algebra and vector spaces - even if the forward model is not linear

Measurement Space

- Measurement space is the space of measurement vectors, dimension m.

State Space

- State space is the space of state vectors, dimension n.

Generally the two vector spaces will have different dimensions,

Forward Function and Model

- The Forward *Function* $\mathbf{f}(\mathbf{x})$ maps from state space onto measurement space, depending on the physics of the measurement.
- The Forward *Model* $\mathbf{F}(\mathbf{x})$is the best we can do in the circumstances to model the forward function

Inverse or Retrieval Method

- The inverse problem is one of finding an inverse mapping $\mathbf{R}(\mathbf{y})$:

Given a point in measurement space, which point or set of points in state space could have mapped into it?

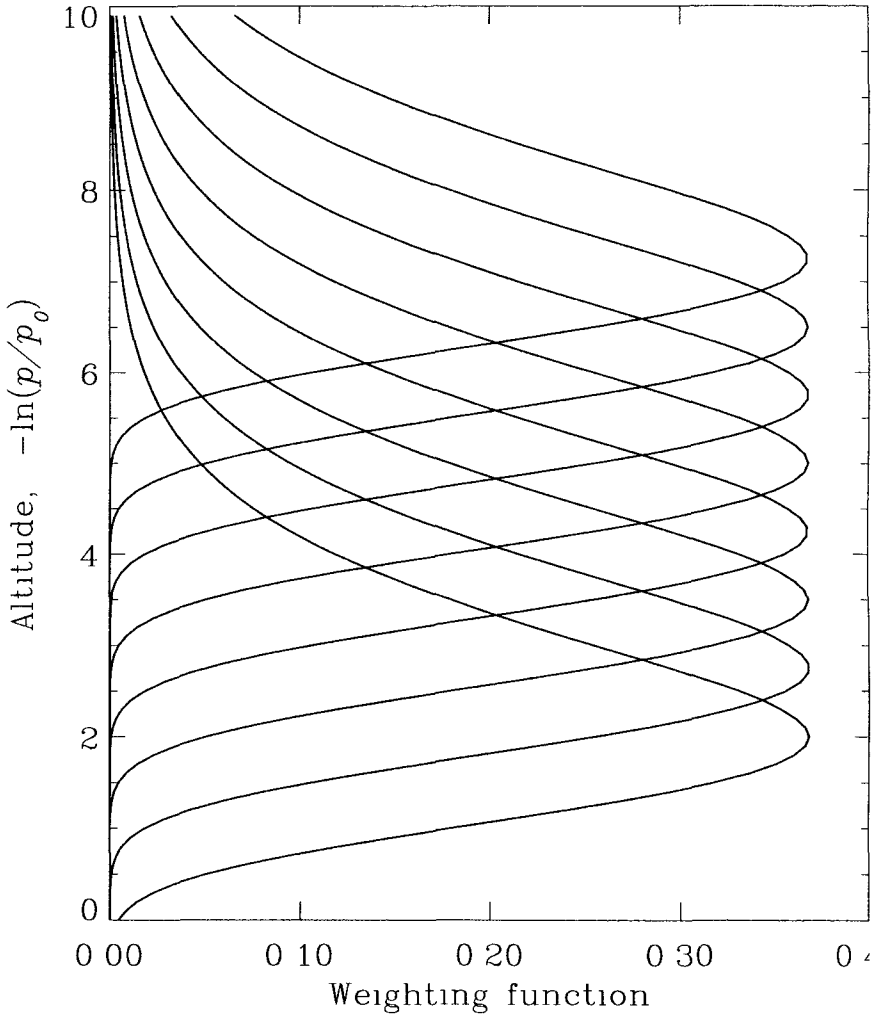
STANDARD ILLUSTRATION

Idealised thermal-emission nadir sounder represented as a linear forward model

$$\mathbf{y} = \mathbf{K}\mathbf{x} + \epsilon$$

$\mathbf{K}$ is the 'weighting function' matrix, ϵ is measurement error or noise

- Vertical coordinate is notionally $\ln(p)$, discretised at 100 levels from 0 in steps of 0 1 to 9 9 – around 0 to 70 km
- Eight channels (elements of $\mathbf{y}$)
- State vector is notionally temperature at 100 levels
- Measurement error (when considered) is 0 5 K



NOISE FREE MEASUREMENTS

Row Space and Null Space

Consider an error-free linear measurement, equivalent to solving linear equations:

$$\mathbf{y} = \mathbf{K}\mathbf{x}$$

The *rows* of $\mathbf{K}$ are the weighting functions $\mathbf{k}_i$:

$$y_i = \mathbf{k}_i\mathbf{x}$$

The $\mathbf{k}_i$ are a set of vectors in state space; the measurements are projections of the state $\mathbf{x}$ onto them.

They span a subspace called the *row space*, of dimension equal to the rank of $\mathbf{K}$, $p \leq \min(n, m)$. If $p < m$ then the weighting functions are not linearly independent.

Only those components of $\mathbf{x}$ in the row space can be measured.

The *null space* is the part of state space which is not in the row space.

ILL-POSED AND WELL-POSED PROBLEMS

Ill or well posed – Under- or over-determined – Under- or over-constrained

The last two seem to mean the same thing.
Ill posed includes both under- and over-determined

Which is which?

1. $p = m = n$. Well posed.
The number of unknowns is equal to the number of measurements, and they are all independent.

2. $p < m = n$. Both underconstrained and overconstrained.
The number of unknowns is equal to the number of measurements, but the measurements are not independent, so they could be inconsistent, and the number of independent pieces of information is less than the number of unknowns.

3. $p = m < n$. Underconstrained.
More unknowns than measurements, but the measurements are all independent.

4. $p < m < n$. Both underconstrained and overconstrained.
More unknowns than measurements, but the measurements are not independent, so they could be inconsistent.

5. $p = n < m$. Overconstrained.
More measurements than unknowns, so they could be inconsistent, but the unknowns are all in the row space, so there is information about all of them.

6. $p < n < m$. Both underconstrained and overconstrained.
More measurements than unknowns, so they could be inconsistent, but the unknowns are not all in the row space, so there is information about all of them.

Summary

If $p < n$ then the system is underconstrained; there is a null space.

If $p < m$ then the system is overconstrained in some part of the row space.

Identifying the row space and the null space

Let $\mathbf{R}$ and $\mathbf{N}$ be bases for the row space and null space respectively. $\mathbf{R}$ is $n \times p$, and $\mathbf{N}$ is $n \times (n-p)$. Together they form a basis for state space. The $n \times n$ matrix $(\mathbf{R}, \mathbf{N})$ is orthonormal:

$$(\mathbf{R}, \mathbf{N})^T(\mathbf{R}, \mathbf{N}) = (\mathbf{R}, \mathbf{N})(\mathbf{R}, \mathbf{N})^T = \mathbf{I}_n$$

so that $\mathbf{R}^T\mathbf{R} = \mathbf{I}_p$, $\mathbf{N}^T\mathbf{N} = \mathbf{I}_{n-p}$ and $\mathbf{R}\mathbf{R}^T + \mathbf{N}\mathbf{N}^T = \mathbf{I}_n$. For each row $\mathbf{k}_i$ of $\mathbf{K}$ we must have

$$\mathbf{N}^T\mathbf{k}_i = 0 \quad \text{and} \quad \mathbf{R}^T\mathbf{k}_i \neq 0$$

It can also be shown that $\mathbf{R}\mathbf{R}^T\mathbf{k}_i = \mathbf{k}_i$, so $\mathbf{R}^T\mathbf{k}_i$ form the coefficients of a representation of $\mathbf{k}_i$ in the $\mathbf{R}$ space.

There are many ways of choosing a matrix $\mathbf{R}$ satisfying these conditions. The simplest is *Gram-Schmidt orthogonalisation*:

Select a unit vector in the direction of $\mathbf{k}_1$ to be $\mathbf{r}_1$: $\mathbf{r}_1 = \mathbf{k}_1/(\mathbf{k}_1 \cdot \mathbf{k}_1)^{1/2}$

Find the components of all of the other $\mathbf{k}_i$'s in this direction, and subtract: $\mathbf{k}'_i = \mathbf{k}_i - \mathbf{k}_i\mathbf{r}_1$ so that all of the $\mathbf{k}'_i$ are orthogonal to $\mathbf{r}_1$.

Choose a unit vector in the direction of $\mathbf{k}'_2$ to be $\mathbf{r}_2$.

Repeat until the remaining $\mathbf{k}''_i \cdots$ are all zero.

Is the neatest way of doing the job. Express $\mathbf{K}$ as

$$_m\mathbf{K}_n = {}_m\mathbf{U}_p\mathbf{\Lambda}_p\mathbf{V}_n^T$$

where the subscripts indicate the sizes of the matrices.
Then the forward model becomes:

$$_m\mathbf{y}_1 = {}_m\mathbf{K}_n\mathbf{x}_1 = {}_m\mathbf{U}_p\mathbf{\Lambda}_p\mathbf{V}_n^T\mathbf{x}_1$$

so that

$${}_p\mathbf{U}_m^T\mathbf{y}_1 = {}_p\mathbf{\Lambda}_p\mathbf{V}_n^T\mathbf{x}_1$$

or

$$\mathbf{y}' = \mathbf{\Lambda}\mathbf{x}'$$

where $\mathbf{y}' = \mathbf{U}^T\mathbf{y}$ and $\mathbf{x}' = \mathbf{V}^T\mathbf{x}$ are both of order p.

The rows of $\mathbf{V}^T$, or the columns of $\mathbf{V}$ (in state space) are a basis for the row space of $\mathbf{K}$.

Similarly the columns of $\mathbf{U}$ (in measurement space) are a basis of its *column space.*

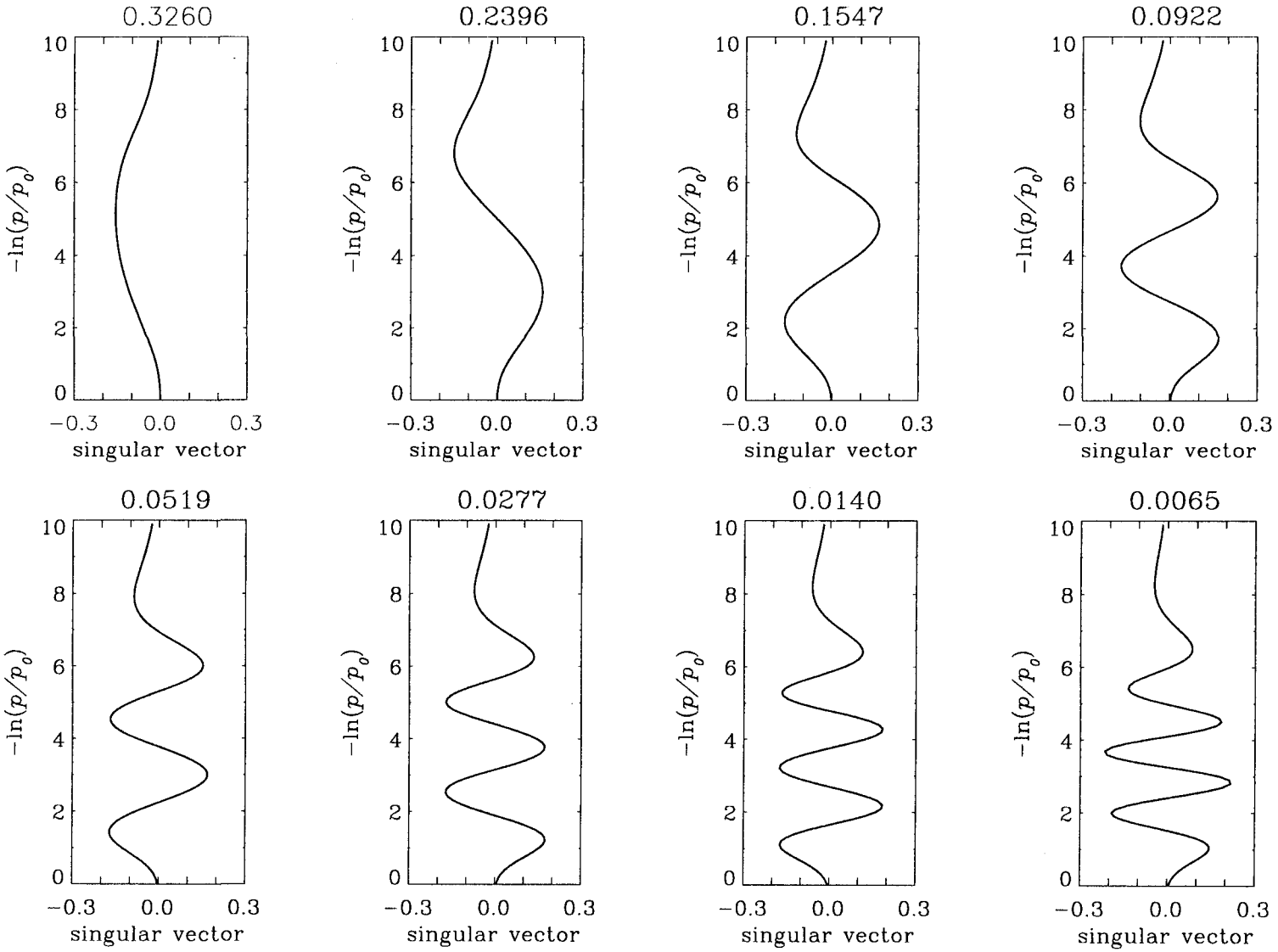
We can also see that *an* exact solution is

$$\mathbf{x}' = \mathbf{\Lambda}^{-1}\mathbf{y}'$$

or

$$\hat{\mathbf{x}} = \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{U}^T\mathbf{y}$$

This is only a unique solution if $p = n$.


0.3260
0.2396
0.1547
0.0922
0.0519
0.0277
0.0140
0.0065
$-\ln(p/p_o)$
singular vector
0
2
4
6
8
10
−0.3
0.0
0.3

Bayesian Approach

– What is the pdf of the state, given the measurement and the *a priori*?

Optimisation Approaches:

– Maximum Likelihood

– Maximum A Posteriori

– Minimum Variance

– Backus-Gilbert - resolution/noise trade-off

Ad hoc Approaches

– Relaxation

– Exact algebraic solutions

This is the most general approach to the problem (that I know of).

Knowledge is represented in terms of *probability density functions*:

- $P(\mathbf{x})$ is the *a priori* p.d.f. of the state, describing what we know about the state before we make the measurement.
- $P(\mathbf{y})$ is the *a priori* p.d.f. of the measurement.
- $P(\mathbf{x},\mathbf{y})$ is the joint *a priori* p.d.f. of $\mathbf{x}$ and $\mathbf{y}$.
- $P(\mathbf{y}|\mathbf{x})$ is the p.d.f. of the measurement given the state - this depends on experimental error and the forward function.
- $P(\mathbf{x}|\mathbf{y})$ is the p.d.f. of the state given the measurement - this is what we want to find.

Bayes theorem states:

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

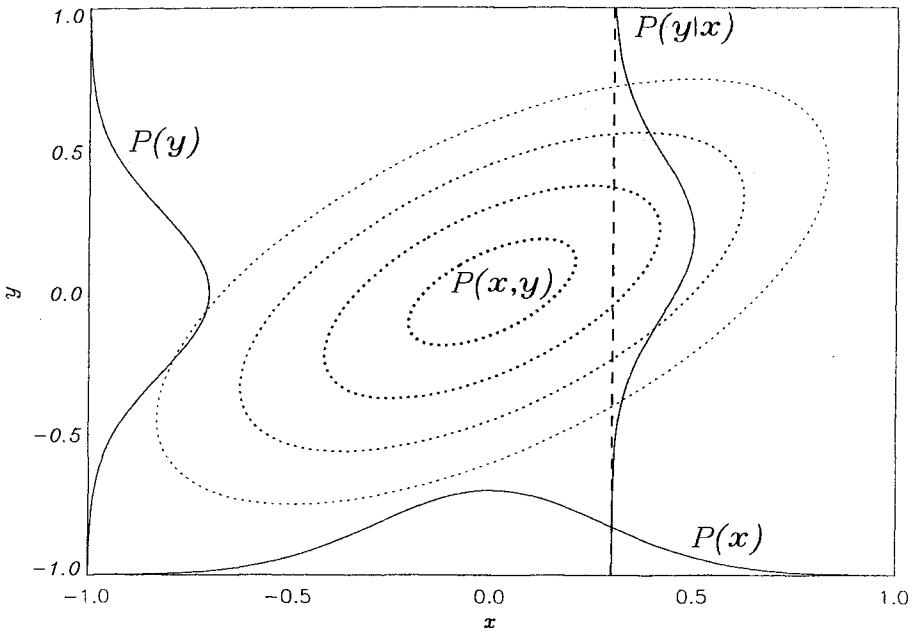
and of course

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

so that

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

If we have a prior p.d.f. for $\mathbf{x}$, $\mathbf{P}(\mathbf{x})$, and we know statistically how $\mathbf{y}$ is related to $\mathbf{x}$ *via* $P(\mathbf{y}|\mathbf{x})$, then we can find an un-normalised version of $P(\mathbf{x}|\mathbf{y})$, namely $P(\mathbf{y}|\mathbf{x})P(\mathbf{x})$, which can be normalised if required.



Explicit forms for the p.d.f's:

- Assume that experimental error is Gaussian:

$$-\ln P(\mathbf{y}|\mathbf{x}) = \frac{1}{2}(\mathbf{y}-\mathbf{F}(\mathbf{x}))^T \mathbf{S}_\epsilon^{-1}(\mathbf{y}-\mathbf{F}(\mathbf{x})) + \text{const}$$

where $\mathbf{F}(\mathbf{x})$ is the Forward model:

$$\mathbf{y} = \mathbf{F}(\mathbf{x}) + \epsilon$$

and $\mathbf{S}_\epsilon$ is the covariance matrix of the experimental error, ϵ:

$$\mathbf{S}_\epsilon = E\{\epsilon\epsilon^T\} = E\{(\mathbf{y}-\mathbf{F}(\mathbf{x}))(\mathbf{y}-\mathbf{F}(\mathbf{x}))^T\}$$

- On the less justifiable assumption that the *a priori* p.d.f. is Gaussian we can write:

$$-\ln P(\mathbf{x}) = \frac{1}{2}(\mathbf{x}-\mathbf{x}_a)^T \mathbf{S}_a^{-1}(\mathbf{x}-\mathbf{x}_a) + \text{const}$$

if $\mathbf{x}$ is distributed normally with mean $\mathbf{x}_a$ and covariance $\mathbf{S}_a$.

- Thus:

$$-2\ln P(\mathbf{x}|\mathbf{y}) = [\mathbf{y}-\mathbf{F}(\mathbf{x})]^T \mathbf{S}_\epsilon^{-1}[\mathbf{y}-\mathbf{F}(\mathbf{x})] + [\mathbf{x}-\mathbf{x}_a]^T \mathbf{S}_a^{-1}[\mathbf{x}-\mathbf{x}_a] + \text{const}$$

- If we want a state estimate $\hat{\mathbf{x}}$ rather than a p.d.f., then we must calculate some function of $P(\mathbf{x}|\mathbf{y})$, such as its mean or its maximum

$$\hat{\mathbf{x}} = \int P(\mathbf{x}|\mathbf{y})\mathbf{x}\, d\mathbf{x} \quad \text{or} \quad \frac{dP(\hat{\mathbf{x}}|\mathbf{y})}{d\mathbf{x}} = 0$$

- The accuracy of the estimate is given by the covariance of $P(\mathbf{x}|\mathbf{y})$ about its mean or maximum.

2.03

The linear problem has a forward model:

$$\mathbf{F}(\mathbf{x}) = \mathbf{K}\mathbf{x}$$

where $\mathbf{K}$ is called the *weighting function matrix* for historic reasons. The p.d.f. $P(\mathbf{x}|\mathbf{y})$ becomes:

$$-2\ln P(\mathbf{x}|\mathbf{y}) = [\mathbf{y}-\mathbf{K}\mathbf{x}]^T\mathbf{S}_\epsilon^{-1}[\mathbf{y}-\mathbf{K}\mathbf{x}] + [\mathbf{x}-\mathbf{x}_a]^T\mathbf{S}_a^{-1}[\mathbf{x}-\mathbf{x}_a] + c_1$$

which is quadratic, so has to be of the form:

$$-2\ln P(\mathbf{x}|\mathbf{y}) = [\mathbf{x}-\hat{\mathbf{x}}]^T\hat{\mathbf{S}}^{-1}[\mathbf{x}-\hat{\mathbf{x}}] + c_2$$

Equate the terms that are quadratic in $\mathbf{x}$:

$$\mathbf{x}^T\mathbf{K}^T\mathbf{S}_\epsilon^{-1}\mathbf{K}\mathbf{x} + \mathbf{x}^T\mathbf{S}_a^{-1}\mathbf{x} = \mathbf{x}^T\hat{\mathbf{S}}^{-1}\mathbf{x}$$

giving

$$\hat{\mathbf{S}}^{-1} = \mathbf{K}^T\mathbf{S}_\epsilon^{-1}\mathbf{K} + \mathbf{S}_a^{-1}$$

Equating the terms linear in $\mathbf{x}$, (or, more easily, in $\mathbf{x}^T$), gives:

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

This must be valid for any $\mathbf{x}$. Cancel the $\mathbf{x}^T$'s, and substitute for $\hat{\mathbf{S}}^{-1}$:

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

giving:

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

This gives the full pdf, including both the mean and its covariance.

2.04

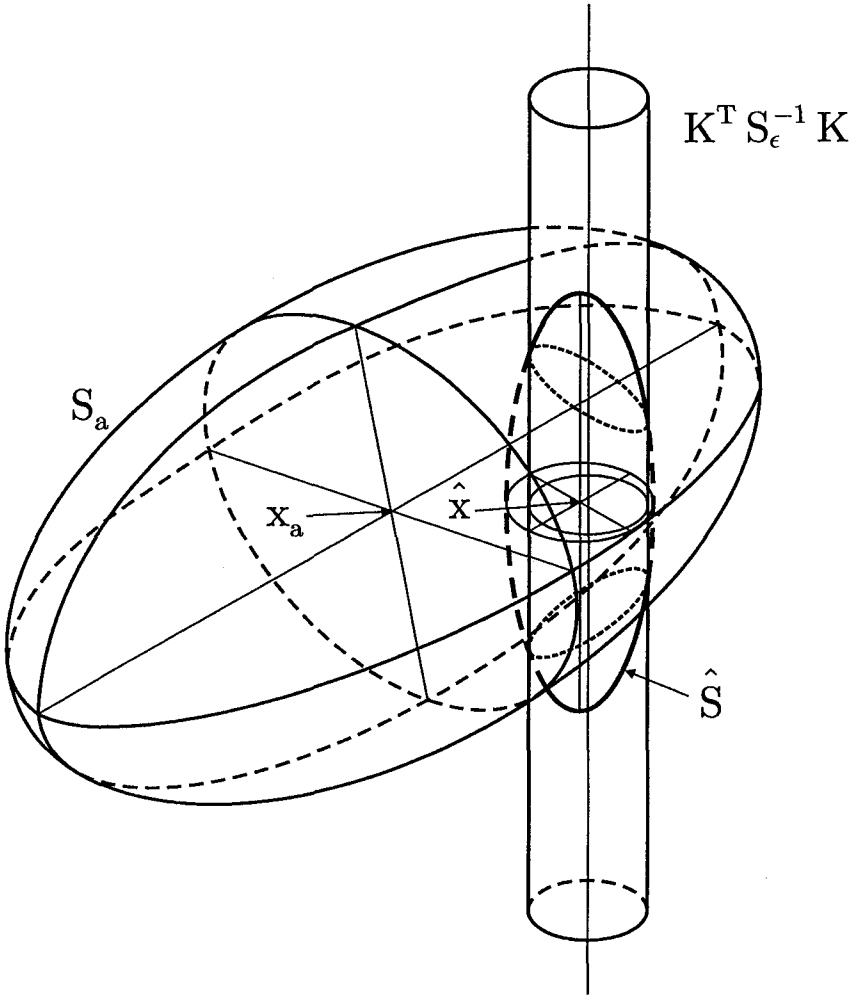


Figure 1:

1

The expected value is:

$$\hat{\mathbf{x}} = (\mathbf{K}^T\mathbf{S}_\epsilon^{-1}\mathbf{K} + \mathbf{S}_a^{-1})^{-1}(\mathbf{K}^T\mathbf{S}_\epsilon^{-1}\mathbf{y} + \mathbf{S}_a^{-1}\mathbf{x}_a) \qquad (1)$$

Underconstrained case

There exists at least one 'exact' solution $\mathbf{x}_e = \mathbf{G}\mathbf{y}$ in the sense that $\mathbf{K}\mathbf{x}_e = \mathbf{y}$, i.e. $\mathbf{K}\mathbf{G} = \mathbf{I}$. For example $\mathbf{G} = \mathbf{K}^T(\mathbf{K}\mathbf{K}^T)^{-1}$.
Replace $\mathbf{y}$ by $\mathbf{K}\mathbf{x}_e$ in (1):

$$\hat{\mathbf{x}} = (\mathbf{K}^T\mathbf{S}_\epsilon^{-1}\mathbf{K} + \mathbf{S}_a^{-1})^{-1}(\mathbf{K}^T\mathbf{S}_\epsilon^{-1}\mathbf{K}\mathbf{x}_e + \mathbf{S}_a^{-1}\mathbf{x}_a)$$

Overconstrained case

The least squares solution $\mathbf{x}_l$ satisfies $\mathbf{K}^T\mathbf{S}_\epsilon^{-1}\mathbf{K}\mathbf{x}_l = \mathbf{K}^T\mathbf{S}_\epsilon^{-1}\mathbf{y}$.
Inserting this in (1) gives:

$$\hat{\mathbf{x}} = (\mathbf{K}^T\mathbf{S}_\epsilon^{-1}\mathbf{K} + \mathbf{S}_a^{-1})^{-1}(\mathbf{K}^T\mathbf{S}_\epsilon^{-1}\mathbf{K}\mathbf{x}_l + \mathbf{S}_a^{-1}\mathbf{x}_a)$$

Both represent a weighted mean of an exact solution $\mathbf{x}_e$ or a least squares solution $\mathbf{x}_l$, with $\mathbf{x}_a$ using relative weights $\mathbf{K}^T\mathbf{S}_\epsilon^{-1}\mathbf{K}$ and $\mathbf{S}_a^{-1}$ respectively – their Fisher information matrices.

This is exactly like the familiar combination of scalar measurements x_1 and x_2 of an unknown x, with variances σ_1^2 and σ_2^2 respectively:

$$\hat{x} = (1/\sigma_1^2 + 1/\sigma_2^2)^{-1}(x_1/\sigma_1^2 + x_2/\sigma_2^2)$$

Information in a general qualitative sense:

Conceptually, what does $\mathbf{y}$ *tell you about* $\mathbf{x}$*?*

We need to answer this to determine if a conceptual instrument design actually works, and to optimise designs.

Use the linear problem for simplicity to illustrate the ideas.

$$\mathbf{y} = \mathbf{K}\mathbf{x} + \epsilon$$

Shannon Information

The information content of a measurement of $\mathbf{x}$ is the change the *entropy* of the probability density function describing our knowledge of $\mathbf{x}$, defined by:

$$S\{P\} = -\int P(\mathbf{x}) \log(P(\mathbf{x})/M(\mathbf{x}))d\mathbf{x}$$

$M(\mathbf{x})$ is a measure function. We will take it to be constant.

Compare this with the statistical mechanics definition of entropy:

$$S = -k\sum_i p_i \ln p_i$$

The Shannon information content of a measurement is the change in entropy between the p.d.f. before, $P(\mathbf{x})$, and the p.d.f. after, $P(\mathbf{x}|\mathbf{y})$, the measurement:

$$H = S\{P(\mathbf{x})\} - S\{P(\mathbf{x}|\mathbf{y})\}$$

ENTROPY OF A BOXCAR PDF

Consider a uniform pdf in one dimension, constant in (0,a):

$$P(x) = 1/a \quad 0 < x < a$$

and zero outside. The entropy is given by

$$S = -\int_0^a \frac{1}{a} \ln\left(\frac{1}{a}\right) \mathrm{d}x = \ln a$$

An similarly, the entropy of any constant pdf in a finite volume V of arbitrary shape is:

$$S = -\int_V \frac{1}{V} \ln\left(\frac{1}{V}\right) \mathrm{d}v = \ln V$$

i.e the entropy is the log of the volume of state space occupied by the pdf.

ENTROPY OF A GAUSSIAN PDF

What does this mean? Consider the Gaussian distribution:

$$P(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}}|\mathbf{S}|^{\frac{1}{2}}} \exp[-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^T\mathbf{S}^{-1}(\mathbf{x}-\bar{\mathbf{x}})]$$

If you evaluate the entropy of a Gaussian distribution you will find it is proportional to $\log|\mathbf{S}|^{\frac{1}{2}}$.

The contours of $P(\mathbf{x})$ in n-space are ellipsoidal, described by

$$(\mathbf{x}-\bar{\mathbf{x}})^T\mathbf{S}^{-1}(\mathbf{x}-\bar{\mathbf{x}}) = \text{constant}$$

The principal axes of the ellipsoid are the eigenvectors of $\mathbf{S}$, and their lengths are proportional to the square roots of the corresponding eigenvalues.

The volume of the ellipsoid is proportional to the root of the product of the eigenvalues, which is proportional to $|\mathbf{S}|^{\frac{1}{2}}$.

Entropy is the log of the volume enclosed by some particular contour of $P(\mathbf{x})$. A 'volume of uncertainty'.

The information content of a measurement is the log of the ratio of the volumes of uncertainty before and after making a measurement.
A generalisation of 'signal to noise'.

In our case:

$$H = \log|\mathbf{S}_a| - \log|\hat{\mathbf{S}}| = -\log|(\mathbf{K}^T\mathbf{S}_\epsilon^{-1}\mathbf{K} + \mathbf{S}_a^{-1})^{-1}\mathbf{S}_a^{-1}|$$

minus the log of the determinant of the weight of $\mathbf{x}_a$ in the Bayesian expectation.

The state estimate that maximises $P(\mathbf{x}|\mathbf{y})$ in the linear Gaussian case is the one which minimises

$$\chi^2 = [\mathbf{y} - \mathbf{K}\mathbf{x}]^T \mathbf{S}_\epsilon^{-1} [\mathbf{y} - \mathbf{K}\mathbf{x}] + [\mathbf{x} - \mathbf{x}_a]^T \mathbf{S}_a^{-1} [\mathbf{x} - \mathbf{x}_a]$$

The r.h.s. has initially $m + n$ degrees of freedom, of which n are fixed by choosing $\hat{\mathbf{x}}$, so the expected value of χ^2 is m.

These m degrees of freedom can be assigned to degrees of freedom for noise d_n and degrees of freedom for signal d_s according to:

$$\begin{aligned} d_n &= E\{[\mathbf{y} - \mathbf{K}\mathbf{x}]^T \mathbf{S}_\epsilon^{-1} [\mathbf{y} - \mathbf{K}\mathbf{x}]\} \\ d_s &= E\{[\mathbf{x} - \mathbf{x}_a]^T \mathbf{S}_a^{-1} [\mathbf{x} - \mathbf{x}_a]\} \end{aligned}$$

Using tr(**CD**)=tr(**DC**), we can see that

$$\begin{aligned} d_s &= E\{[\hat{\mathbf{x}} - \mathbf{x}_a]^T \mathbf{S}_a^{-1} [\hat{\mathbf{x}} - \mathbf{x}_a]\} \\ &= E\{\mathrm{tr}([\hat{\mathbf{x}} - \mathbf{x}_a][\hat{\mathbf{x}} - \mathbf{x}_a]^T \mathbf{S}_a^{-1})\} \\ &= \mathrm{tr}(E\{[\hat{\mathbf{x}} - \mathbf{x}_a][\hat{\mathbf{x}} - \mathbf{x}_a]^T\} \mathbf{S}_a^{-1}) \end{aligned}$$

With some manipulation we can find

$$\begin{aligned} d_s &= \mathrm{tr}((\mathbf{K}^T \mathbf{S}_\epsilon^{-1} \mathbf{K} + \mathbf{S}_a^{-1})^{-1} \mathbf{K}^T \mathbf{S}_\epsilon^{-1} \mathbf{K}) \\ &= \mathrm{tr}(\mathbf{K}\mathbf{S}_a \mathbf{K}^T (\mathbf{K}\mathbf{S}_a \mathbf{K}^T + \mathbf{S}_\epsilon)^{-1}) \\ d_n &= \mathrm{tr}((\mathbf{K}^T \mathbf{S}_\epsilon^{-1} \mathbf{K} + \mathbf{S}_a^{-1})^{-1} \mathbf{S}_a^{-1}) + m - n \\ &= \mathrm{tr}(\mathbf{S}_\epsilon (\mathbf{K}\mathbf{S}_a \mathbf{K}^T + \mathbf{S}_\epsilon)^{-1}) \end{aligned}$$

The elements of the measurement vector will not be statistically independent if the covariance is not diagonal. Likewise for the *a priori*.

The measurements will not be independent if $\mathbf{K}$ is not diagonal.

Therefore it is helpful to transform to a different basis.

First, statistical independence. Define:

$$\tilde{\mathbf{y}} = \mathbf{S}_\epsilon^{-\frac{1}{2}} \mathbf{y} \qquad \tilde{\mathbf{x}} = \mathbf{S}_a^{-\frac{1}{2}} \mathbf{x}$$

The transformed covariances $\tilde{\mathbf{S}}_a$ and $\tilde{\mathbf{S}}_\epsilon$ both become unit matrices.

The forward model becomes:

$$\tilde{\mathbf{y}} = \tilde{\mathbf{K}}\tilde{\mathbf{x}} + \tilde{\boldsymbol{\epsilon}}$$

where $\tilde{\mathbf{K}} = \mathbf{S}_\epsilon^{-\frac{1}{2}} \mathbf{K} \mathbf{S}_a^{\frac{1}{2}}$.

The solution covariance becomes:

$$\hat{\tilde{\mathbf{S}}} = (\mathbf{I}_n + \tilde{\mathbf{K}}^T \tilde{\mathbf{K}})^{-1}$$

Now make $\tilde{\mathbf{K}}$ diagonal. Rotate both $\mathbf{x}$ and $\mathbf{y}$ to yet another basis, defined by the singular vectors of $\tilde{\mathbf{K}}$:

$$\tilde{\mathbf{K}} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$$

Define:

$$\mathbf{x}' = \mathbf{V}^T\tilde{\mathbf{x}} \quad \mathbf{y}' = \mathbf{U}^T\tilde{\mathbf{y}} \quad \boldsymbol{\epsilon}' = \mathbf{U}^T\tilde{\boldsymbol{\epsilon}}$$

The forward model becomes:

$$\mathbf{y}' = \mathbf{\Lambda}\mathbf{x}' + \boldsymbol{\epsilon}' \tag{1}$$

This separates the original problem into a set of independent scalar problems.

The Jacobian is diagonal, $\mathbf{\Lambda}$, and the *a priori* and noise covariances are still unit matrices, hence the solution covariance is:

$$\hat{\mathbf{S}}' = (\mathbf{I}_n + \mathbf{\Lambda}^2)^{-1}$$

which is diagonal, and the solution itself is

$$\hat{\mathbf{x}}' = (\mathbf{I}_n + \mathbf{\Lambda}^2)^{-1}(\mathbf{\Lambda}\mathbf{y}' + \mathbf{x}'_a)$$

not $\hat{\mathbf{x}}' = \mathbf{\Lambda}^{-1}\mathbf{y}'$ as you might expect from (1).

Elements for which $\lambda_i \gg 1$ or $(1+\lambda_i^2)^{-1} \ll 1$ are well measured

Elements for which $\lambda_i \ll 1$ or $(1+\lambda_i^2)^{-1} \gg 1$ are poorly measured.

Shannon Information in the Transformed Basis

Because it is a ratio of volumes, the linear transformation does not change the information content. So consider information in the $\mathbf{x}'$, $\mathbf{y}'$ system:

$$\begin{aligned} H &= S\{\mathbf{S}'_a\} - S\{\hat{\mathbf{S}}'\} \\ &= -\frac{1}{2}\log(|\mathbf{I}_n|) + \frac{1}{2}\log(|(\mathbf{\Lambda}^2 + \mathbf{I})^{-1}|) \\ &= \sum_i \frac{1}{2}\log(1+\lambda_i^2) \end{aligned}$$

Degres of Freedom in the Transformed Basis

The number of independent quantities measured can be thought of as the number of eigenvalues for which $\lambda_i \gg 1$

The degrees of freedom for signal is

$$d_s = \sum_i \lambda_i^2(1+\lambda_i^2)^{-1}$$

It is also the sum of the eigenvalues of $\mathbf{I}_n - \hat{\hat{\mathbf{S}}}$.

For each independent component x'_i

- The information content is $\frac{1}{2}\log(1+\lambda_i^2)$
- The degrees of freedom for signal is $\lambda_i^2(1+\lambda_i^2)^{-1}$

The part of measurement space that can be seen is that spanned by the weighting functions. Anything outside that is in the null space of $\mathbf{K}$.

Any orthogonal linear combination of the weighting functions will form a basis (coordinate system) for the observable space. An example is those singular vectors of $\mathbf{K}$ which have non-zero singular values.

The vectors which have zero singular values form a basis for the null space.

Any component of the state in the null space maps onto the origin in measurement space.

This implies that there are distinct states, in fact whole subspaces, which map onto the same point in measurement space, and cannot be distinguished by the measurement.

However

– the solution can have components in the null space - from the *a priori.*

– components observable in principle can have near zero contributions from the measurement, the 'near null space'

In the $\mathbf{x}'$, $\mathbf{y}'$ *system:*

– vectors with $\lambda = 0$ are in the null space

– vectors with $\lambda \ll 1$ are in the near null space

– vectors with $\lambda \gtrsim 1$ are in the non-null space

Table 2.1 Singular values of $\tilde{\mathbf{K}}$, together with contributions of each vector to the degrees of freedom d_s and information content H for both covariance matrices. Measurement noise variance is 0.25 K^2.

	Diagonal covariance			Full covariance		
i	λ_i	d_{si}	H_i (bits)	λ_i	d_{si}	H_i (bits)
1	6.51929	0.97701	2.72149	27.81364	0.99871	4.79865
2	4.79231	0.95827	2.29147	18.07567	0.99695	4.17818
3	3.09445	0.90544	1.70134	9.94379	0.98999	3.32105
4	1.84370	0.77269	1.06862	5.00738	0.96165	2.35227
5	1.03787	0.51858	0.52731	2.39204	0.85123	1.37443
6	0.55497	0.23547	0.19368	1.09086	0.54337	0.56546
7	0.27941	0.07242	0.05423	0.46770	0.17948	0.14270
8	0.13011	0.01665	0.01211	0.17989	0.03135	0.02297
totals		4.45653	8.57024		5.55272	16.75571

FTIR Measurements of CO.

30 levels; 894 measurements:

Apparently heavily overconstrained.

Singular Values of K

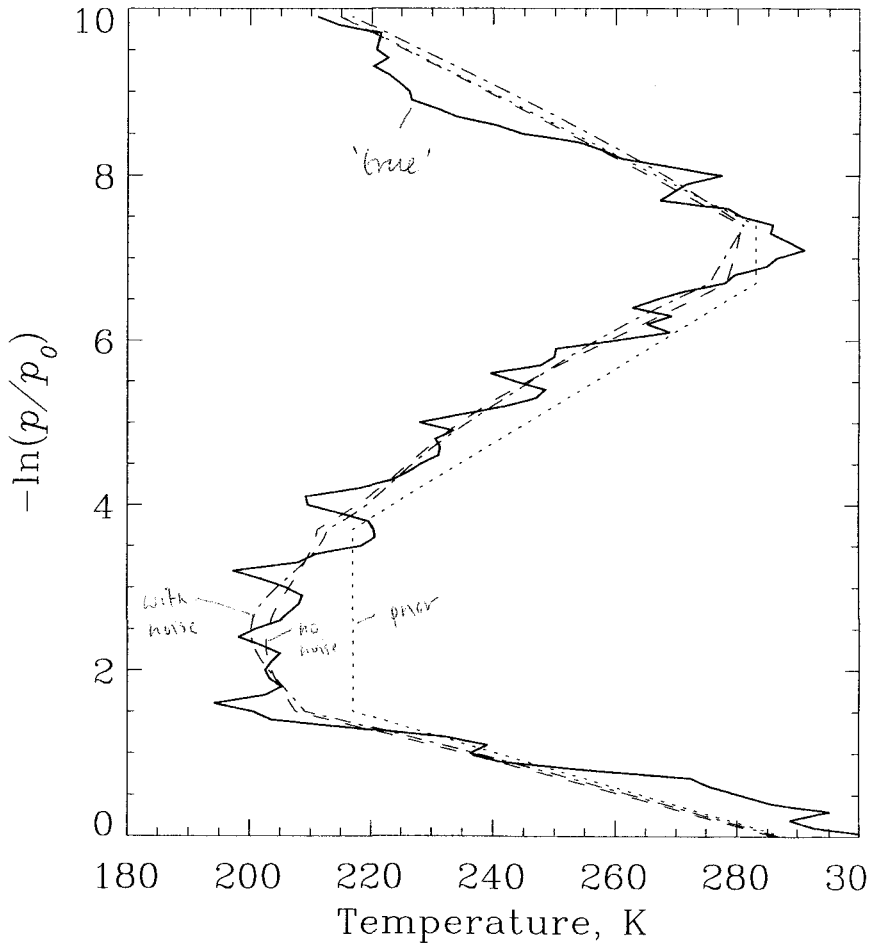
5.345	3.498	0.033
0.0046	7.15e-04	2.56e-04
7.76e-05	2.13e-05	1.71e-05
1.38e-05	9.01e-06	6.73e-06
5.82e-06	4.79e-06	2.87e-06
3.52e-06	3.748e-06	1.91e-06
9.83e-07	2.37e-07	7.71e-07
1.18e-07	1.48e-06	1.95e-07
1.37e-07	6.67e-08	3.50e-08
3.37e-08	5.83e-09	6.29e-09

Noise is 0.03 in these units;
Prior variance is ~1.

There are about 2.5 degrees of freedom.

Lot of near null space.

How do we understand the nature of a retrieval?



If we want our retrieval to optimise something, we had better find out what properties a retrieval might have, so we can choose what we want to optimise.

We will therefore set up a formal characterisation that can apply to any retrieval method

- The measurement $\mathbf{y}$ is conceptually a function of some unknown state $\mathbf{x}$:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}, \mathbf{b}) + \epsilon$$

where:
- $\mathbf{y}$ is the measurement vector, length m
- $\mathbf{x}$ is the state vector, length n
- $\mathbf{f}$ is a function describing the physics of the measurement,
- $\mathbf{b}$ is a set of 'known' parameters of this function,
- ϵ is measurement error, with covariance $\mathbf{S}_\epsilon$.

- The retrieval $\hat{\mathbf{x}}$ is conceptually a function of the form:

$$\hat{\mathbf{x}} = \mathbf{R}(\mathbf{y}, \hat{\mathbf{b}}, \mathbf{c})$$

where:
- $\mathbf{R}$ represents the retrieval method
- $\hat{\mathbf{b}}$ is our estimate of the forward function parameters $\mathbf{b}$
- $\mathbf{c}$ represents any parameters used in the inverse method that do not affect the measurement, e.g. *a priori.*

Thus the retrieval is related to the 'truth' $\mathbf{x}$ formally by:

$$\hat{\mathbf{x}} = \mathbf{R}(\mathbf{f}(\mathbf{x}, \mathbf{b}) + \epsilon, \hat{\mathbf{b}}, \mathbf{c})$$

which may be regarded as the transfer function of the measurement and retrieval system as a whole.

Characterisation means evaluating:

- $\partial\hat{\mathbf{x}}/\partial\mathbf{x} = \mathbf{A}$, sensitivity to actual state: *Averaging Kernel*

Error analysis involves evaluating:

- $\partial\hat{\mathbf{x}}/\partial\epsilon = \mathbf{G}_y$, sensitivity to noise (or to measurement!)

- $\partial\hat{\mathbf{x}}/\partial\mathbf{b} = \mathbf{G}_b$, sensitivity to non-retrieved parameters

- $\partial\hat{\mathbf{x}}/\partial\mathbf{c} = \mathbf{G}_c$, sensitivity to retrieval method parameters

and understanding the effect of replacing $\mathbf{f}$ by a numerical *Forward Model* $\mathbf{F}$.

THE FORWARD MODEL

We often need to approximate the forward function by a *Forward Model*:

$$\mathbf{F}(\mathbf{x}, \mathbf{b}) \simeq \mathbf{f}(\mathbf{x}, \mathbf{b})$$

Where $\mathbf{F}$ *models* the physics of the measurement, including the instrument, as well as we can.

- It usually has parameters $\mathbf{b}$ which have experimental error
- The vector $\mathbf{b}$ is *not* a target for retrieval
- There may be parameters $\mathbf{b}'$ of the forward function that are not included in the forward model:

$$\mathbf{F}(\mathbf{x}, \mathbf{b}) \simeq \mathbf{f}(\mathbf{x}, \mathbf{b}, \mathbf{b}')$$

LINEARISE THE TRANSFER FUNCTION

The retrieved quantity is expressed as:

$$\hat{\mathbf{x}} = \mathbf{R}(\mathbf{f}(\mathbf{x}, \mathbf{b}, \mathbf{b}') + \boldsymbol{\epsilon}, \hat{\mathbf{b}}, \mathbf{x}_a, \mathbf{c})$$

where we have also separated $\mathbf{x}_a$, the *a priori*, from other components of $\mathbf{c}$.

Replace $\mathbf{f}$ by $\mathbf{F} + \Delta\mathbf{f}$:

$$\hat{\mathbf{x}} = \mathbf{R}(\mathbf{F}(\mathbf{x}, \mathbf{b}) + \Delta\mathbf{f}(\mathbf{x}, \mathbf{b}, \mathbf{b}') + \boldsymbol{\epsilon}, \hat{\mathbf{b}}, \mathbf{x}_a, \mathbf{c})$$

where $\Delta\mathbf{f}$ is the error in the forward model relative to the real physics:

$$\Delta\mathbf{f} = \mathbf{f}(\mathbf{x}, \mathbf{b}, \mathbf{b}') - \mathbf{F}(\mathbf{x}, \mathbf{b})$$

Linearise $\mathbf{F}$ about $\mathbf{x} = \mathbf{x}_a$, $\mathbf{b} = \hat{\mathbf{b}}$:

$$\hat{\mathbf{x}} = \mathbf{R}(\mathbf{F}(\mathbf{x}_a, \hat{\mathbf{b}}) + \mathbf{K}_x(\mathbf{x} - \mathbf{x}_a) + \mathbf{K}_b(\mathbf{b} - \hat{\mathbf{b}}) + \Delta\mathbf{f}(\mathbf{x}, \mathbf{b}, \mathbf{b}') + \boldsymbol{\epsilon}, \hat{\mathbf{b}}, \mathbf{x}_a, \mathbf{c})$$

where $\mathbf{K}_x = \partial\mathbf{F}/\partial\mathbf{x}$ (the *weighting function*) and $\mathbf{K}_b = \partial\mathbf{F}/\partial\mathbf{b}$.

Linearise $\mathbf{R}$ with respect to its first argument, $\mathbf{y}$:

$$\hat{\mathbf{x}} = \mathbf{R}[\mathbf{F}(\mathbf{x}_a, \hat{\mathbf{b}}), \hat{\mathbf{b}}, \mathbf{x}_a, \mathbf{c}] + \mathbf{G}_y[\mathbf{K}_x(\mathbf{x} - \mathbf{x}_a) + \mathbf{K}_b(\mathbf{b} - \hat{\mathbf{b}}) + \Delta\mathbf{f}(\mathbf{x}, \mathbf{b}, \mathbf{b}') + \boldsymbol{\epsilon}]$$

where $\mathbf{G}_y = \partial\mathbf{R}/\partial\mathbf{y}$ (the *contribution function*)

Some rearrangement gives:

$$\begin{aligned}\hat{\mathbf{x}} - \mathbf{x}_a =& \mathbf{R}(\mathbf{F}(\mathbf{x}_a, \hat{\mathbf{b}}), \hat{\mathbf{b}}, \mathbf{x}_a, \mathbf{c}) - \mathbf{x}_a && \textit{bias}\\ &+ \mathbf{A}(\mathbf{x} - \mathbf{x}_a) && \textit{smoothing}\\ &+ \mathbf{G}_y\epsilon_y && \textit{error}\end{aligned}$$

where

$$\mathbf{A} = \mathbf{G}_y\mathbf{K}_x = \partial\hat{\mathbf{x}}/\partial\mathbf{x}$$

the *averaging kernel*, and

$$\epsilon_y = \mathbf{K}_b(\mathbf{b} - \hat{\mathbf{b}}) + \Delta\mathbf{f}(\mathbf{x}, \mathbf{b}, \mathbf{b}') + \epsilon$$

is the total error in the measurement relative to the forward model.

Bias

This is the error you would get on doing a simulated error-free retrieval of the *a priori*.

A priori is what you know about the state before you make the measurement. Any reasonable retrieval method should return the *a priori* given a measurement vector that corresponds to it, so the bias should be zero.

But check your system to be sure it is. . .

The retrieved state is a smoothed version of the true state with smoothing functions given by the rows of $\mathbf{A}$, plus error terms:

$$\hat{\mathbf{x}} = \mathbf{x}_a + \mathbf{A}(\mathbf{x} - \mathbf{x}_a) + \mathbf{G}_y\boldsymbol{\epsilon}_y = (\mathbf{I} - \mathbf{A})\mathbf{x}_a + \mathbf{A}\mathbf{x} + \mathbf{G}_y\boldsymbol{\epsilon}_y$$

You can either:

- accept that the retrieval is an estimate of a smoothed state, not the true state

  or

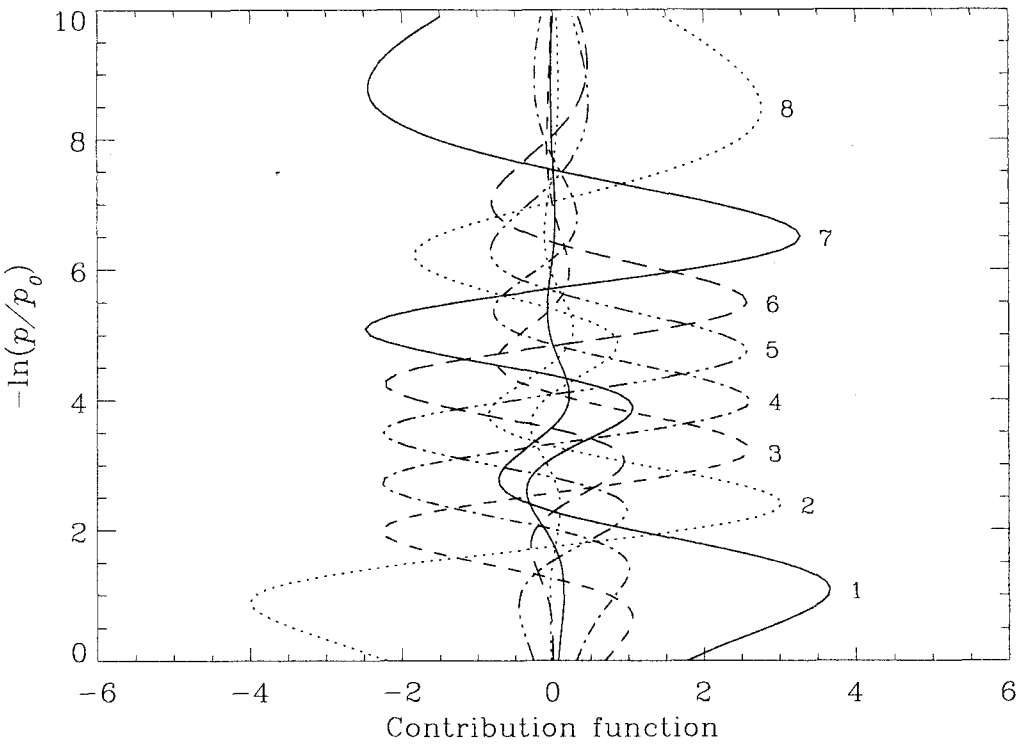
- consider the retrieval as an estimate of the true state, with an error contribution due to smoothing.

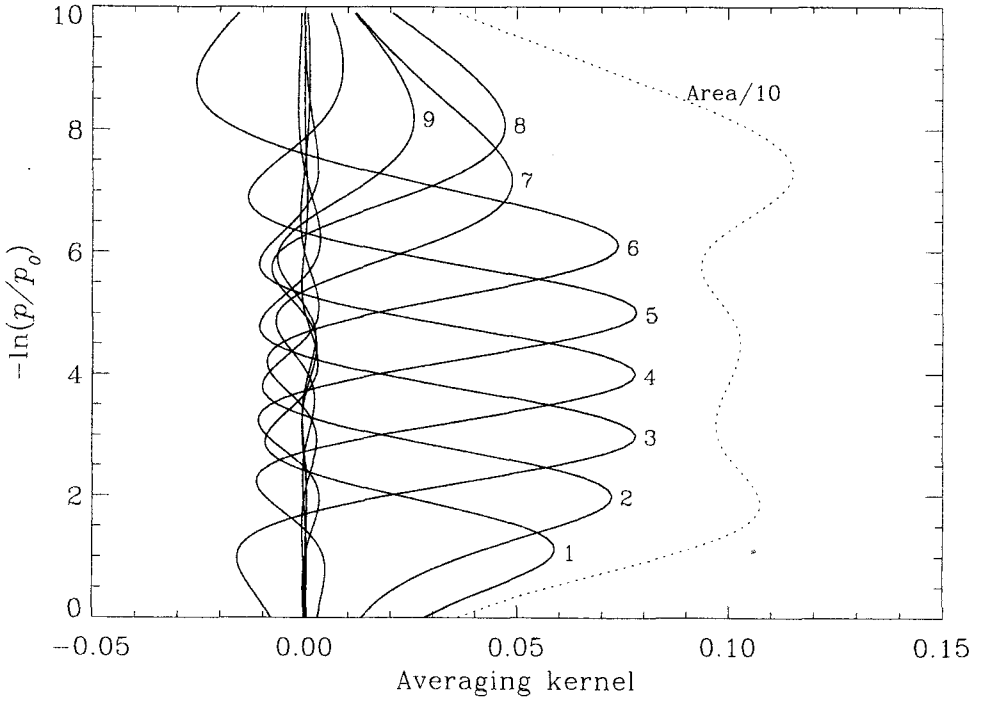
  The error analysis is different in the two cases, because in the second case there is an extra term.

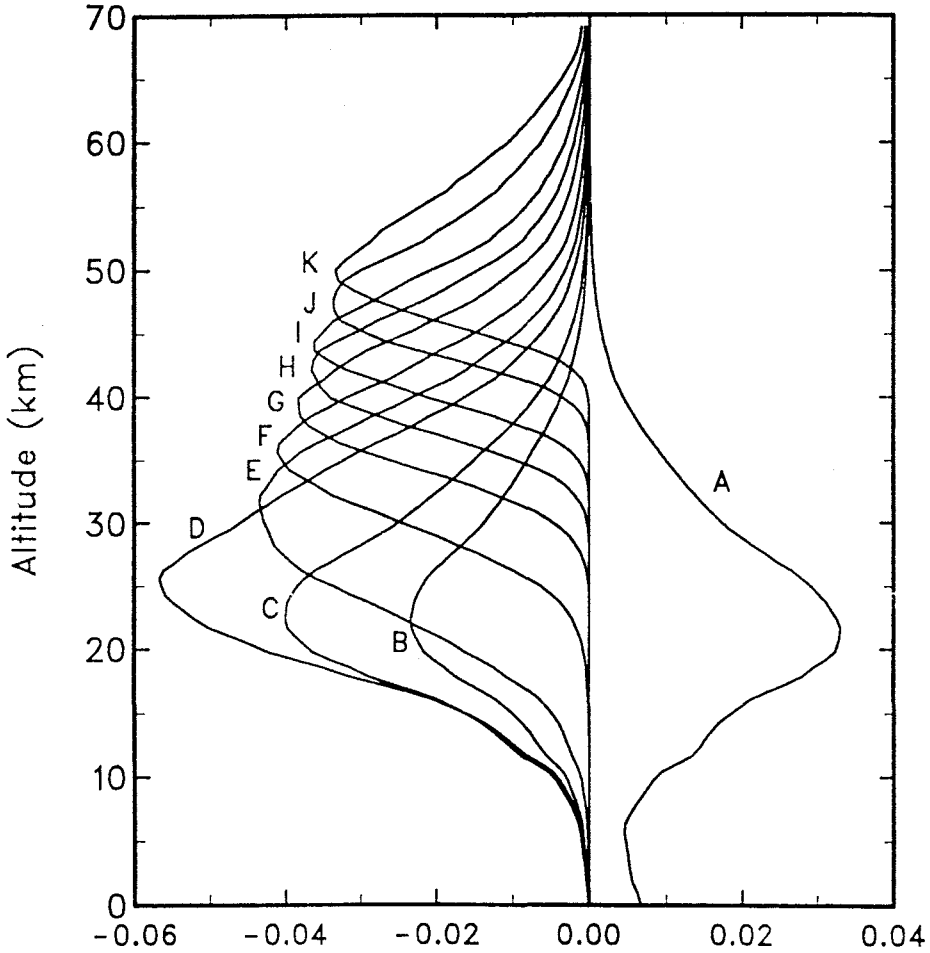
- If the state represents a profile, then the averaging kernel is a smoothing function with a *width* and an *area*.

  – The *width* is a measure of the resolution of the observing system

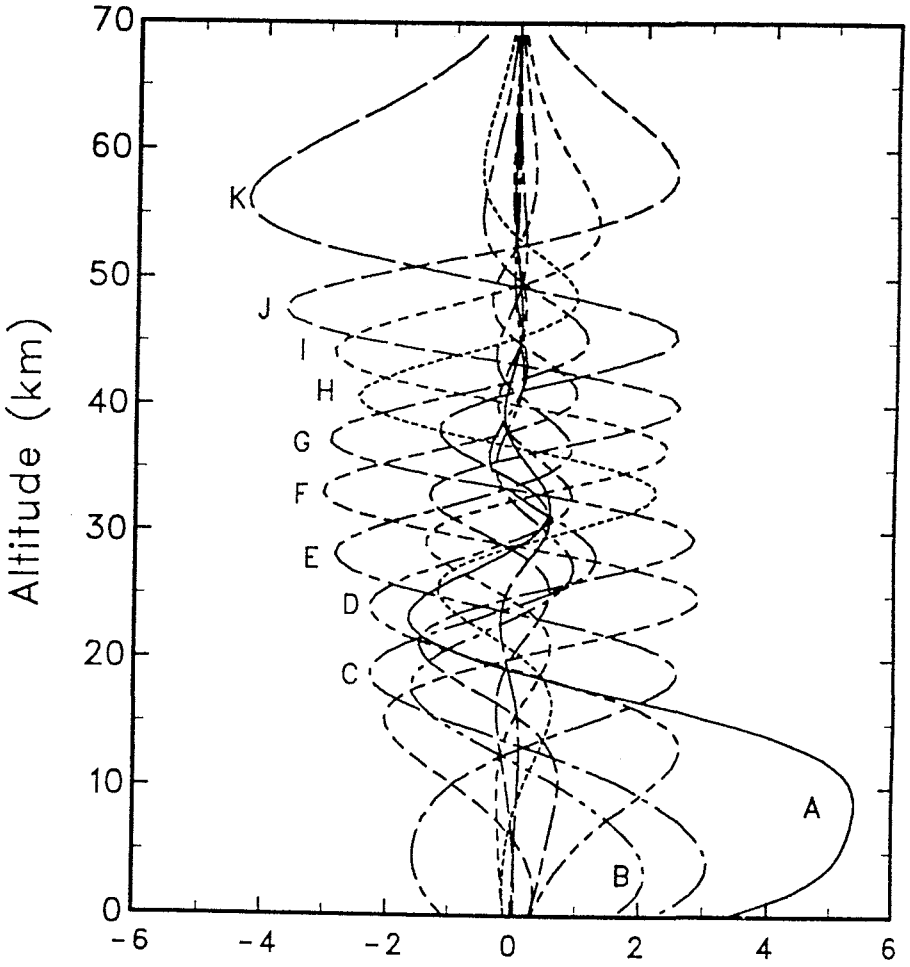
  – The *area* (generally between zero and unity) is a simple measure of the amount of real information that appears in the retrieval.


10
8
6
4
2
0
−ln(p/p_o)
8
7
6
5
4
3
2
1
−6
−4
−2
0
2
4
6
Contribution function

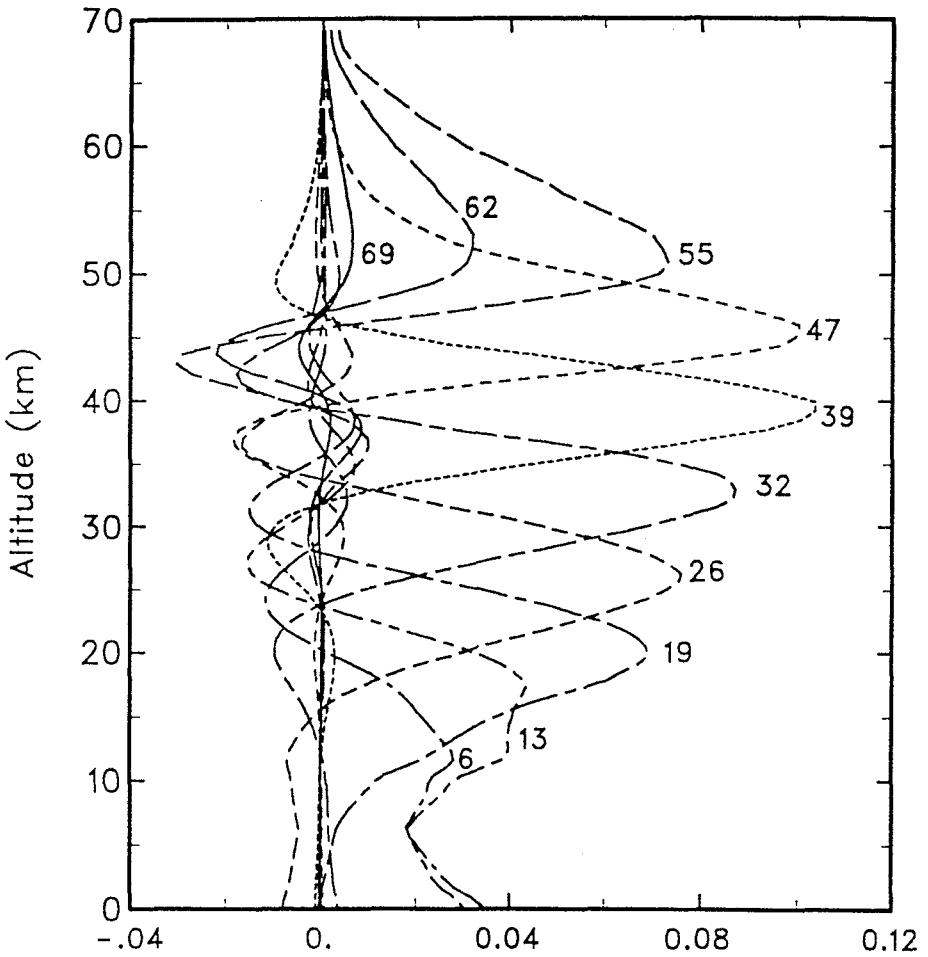

10
8
6
4
2
0
−ln(p/p_o)
Area/10
9
8
7
6
5
4
3
2
1
−0.05
0.00
0.05
0.10
0.15
Averaging kernel



SBUV Weighting function



SBUV Gain Matrix



SBUV Averaging Kernels

"The retrieved quantity is expressed as:

$$\hat{\mathbf{x}} = \mathbf{R}(\mathbf{f}(\mathbf{x}, \mathbf{b}, \mathbf{b}') + \epsilon, \hat{\mathbf{b}}, \mathbf{x}_a, \mathbf{c})$$

where we have also separated $\mathbf{x}_a$, the *a priori*, from other components of $\mathbf{c}$."

We should also look at the sensitivity of the retrieval to the inverse model parameters, $\mathbf{x}_a$ and $\mathbf{c}$.

The linear expansion in $\mathbf{x}$, $\mathbf{b}$ and ϵ gave:

$$\hat{\mathbf{x}} = [\mathbf{R}(\mathbf{F}(\mathbf{x}_a, \hat{\mathbf{b}}), \hat{\mathbf{b}}, \mathbf{x}_a, \mathbf{c})] + \mathbf{A}(\mathbf{x} - \mathbf{x}_a) + \mathbf{G}_y \epsilon_y$$

We argued that the term in square brackets should be equal to $\mathbf{x}_a$, for any reasonable retrieval.

This has the consequence that, at least to first order, $\partial \mathbf{R}/\partial \mathbf{c} = 0$ for any reasonable retrieval.

It also follows that:

$$\frac{\partial \mathbf{R}}{\partial \mathbf{x}_a} = \frac{\partial \hat{\mathbf{x}}}{\partial \mathbf{x}_a} = \mathbf{I}_n - \mathbf{A}$$

For any reasonable retrieval.

Thus the reasonableness criterion has the consequence that there can be no inverse model parameters $\mathbf{c}$ that matter, other than $\mathbf{x}_a$.

Incidentally: The term in square brackets should not depend on $\hat{\mathbf{b}}$ either. This implies that $\mathbf{G}_b + \mathbf{G}_y \mathbf{K}_b = 0$.

4.11

Some further rearrangement gives for the error in $\hat{\mathbf{x}}$:

$$\begin{aligned} \hat{\mathbf{x}} - \mathbf{x} =& (\mathbf{A} - \mathbf{I})(\mathbf{x} - \mathbf{x}_a) && \textit{smoothing} \\ &+ \mathbf{G}_y \epsilon_y && \textit{measurement error} \end{aligned}$$

where the bias term has been dropped, and:

$$\epsilon_y = \mathbf{K}_b(\mathbf{b} - \hat{\mathbf{b}}) + \Delta\mathbf{f}(\mathbf{x}, \mathbf{b}, \mathbf{b}') + \epsilon$$

Thus the error sources can be split up as:

$$\begin{aligned} \hat{\mathbf{x}} - \mathbf{x} =& (\mathbf{A} - \mathbf{I})(\mathbf{x} - \mathbf{x}_a) && \textit{smoothing} \\ &+ \mathbf{G}_y \mathbf{K}_b(\mathbf{b} - \hat{\mathbf{b}}) && \textit{model parameters} \\ &+ \mathbf{G}_y \Delta\mathbf{f}(\mathbf{x}, \mathbf{b}, \mathbf{b}') && \textit{modelling error} \\ &+ \mathbf{G}_y \epsilon && \textit{measurement noise} \end{aligned}$$

Some of these are easy to estimate, some are not.

$$\textit{measurement noise} = \mathbf{G}_y \epsilon$$

This is the easiest component to evaluate.

ϵ is usually random noise, and is often unbiassed and uncorrelated between channels, and has a known covariance matrix.

The covariance of the measurement noise is:

$$\mathbf{S}_n = \mathbf{G}_y \mathbf{S}_\epsilon \mathbf{G}_y^T$$

Note that $\mathbf{S}_n$ is not necessarily diagonal, so there will be errors which are correlated between different elements of the state vector.

This is true of all of the error sources.

To estimate the *actual* smoothing error, you need to know the true state:

$$\textit{smoothing error} = (\mathbf{A} - \mathbf{I})(\mathbf{x} - \mathbf{x}_a)$$

To characterise the statistics of this error, you need its mean and covariance over some ensemble.

The mean should be zero.

The covariance is:

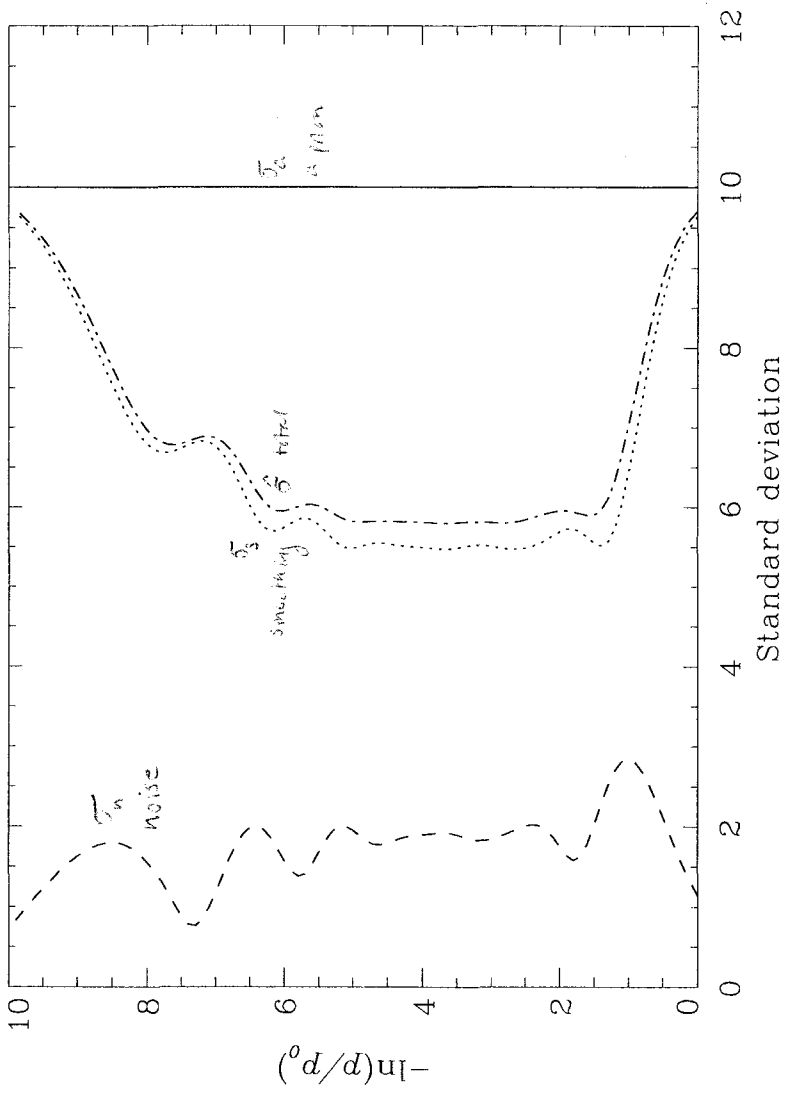
$$\begin{aligned}\mathbf{S}_s =& \mathcal{E}\{(\mathbf{A} - \mathbf{I})(\mathbf{x} - \mathbf{x}_a)(\mathbf{x} - \mathbf{x}_a)^T(\mathbf{A} - \mathbf{I})^T\} \\ =& (\mathbf{A} - \mathbf{I})\mathcal{E}\{(\mathbf{x} - \mathbf{x}_a)(\mathbf{x} - \mathbf{x}_a)^T\}(\mathbf{A} - \mathbf{I})^T \\ =& (\mathbf{A} - \mathbf{I})\mathbf{S}_a(\mathbf{A} - \mathbf{I})^T\end{aligned}$$

where $\mathbf{S}_a$ is the covariance of an ensemble of states about the *a priori* state. This is best regarded as the covariance of a climatology.

To estimate the smoothing error, you need to know the climatological covariance matrix.

To do the job properly, you need the real climatology, not just some *ad hoc* matrix that has been used as a constraint in the retrieval.

The real climatology is often not available. Much of the smoothing error can be in fine spatial scales that may never have been measured.



To estimate the *actual* smoothing error, you need to know the true state:

$$\textit{smoothing error} = (\mathbf{A}-\mathbf{I})(\mathbf{x}-\mathbf{x}_a)$$

To characterise the statistics of this error, you need its mean and covariance over some ensemble.

The mean should be zero.

The covariance is:

$$\begin{aligned}\mathbf{S}_s =& \mathcal{E}\{(\mathbf{A}-\mathbf{I})(\mathbf{x}-\mathbf{x}_a)(\mathbf{x}-\mathbf{x}_a)^T(\mathbf{A}-\mathbf{I})^T\}\\ =&(\mathbf{A}-\mathbf{I})\mathcal{E}\{(\mathbf{x}-\mathbf{x}_a)(\mathbf{x}-\mathbf{x}_a)^T\}(\mathbf{A}-\mathbf{I})^T\\ =&(\mathbf{A}-\mathbf{I})\mathbf{S}_a(\mathbf{A}-\mathbf{I})^T\end{aligned}$$

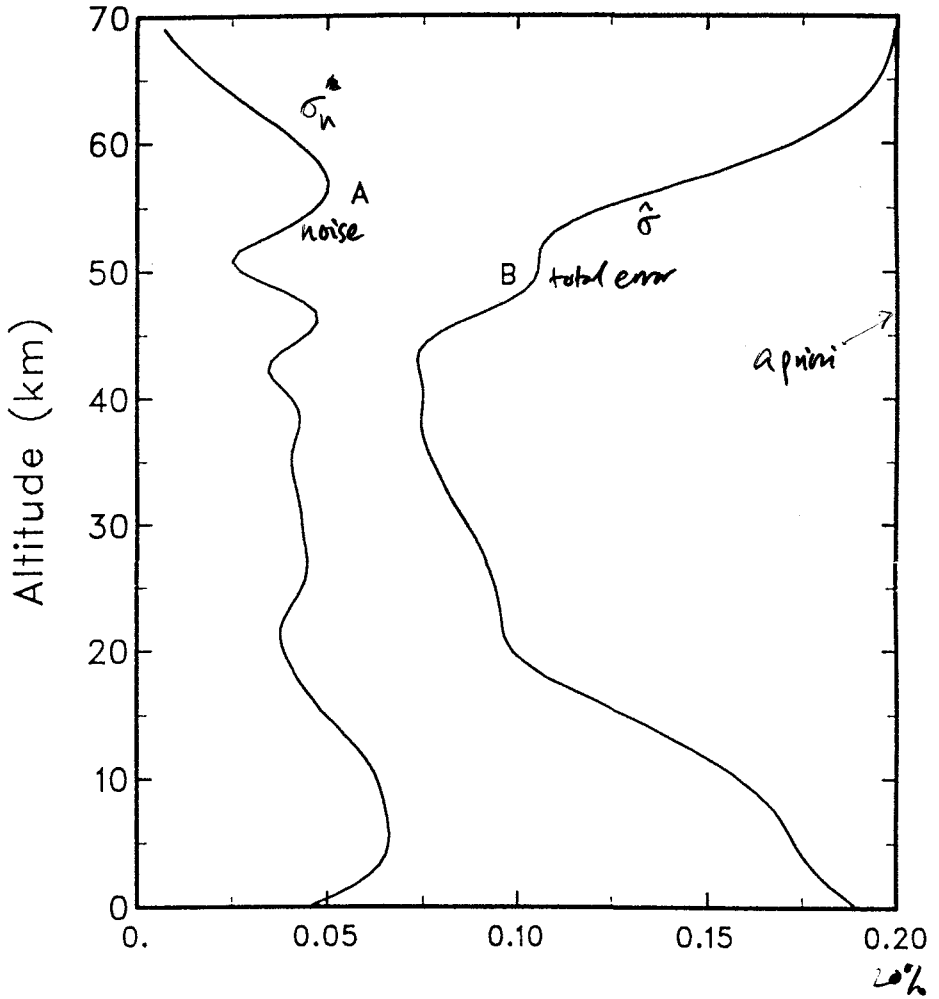
where $\mathbf{S}_a$ is the covariance of an ensemble of states about the *a priori* state. This is best regarded as the covariance of a climatology.

To estimate the smoothing error, you need to know the climatological covariance matrix.

To do the job properly, you need the real climatology, not just some *ad hoc* matrix that has been used as a constraint in the retrieval.

The real climatology is often not available. Much of the smoothing error can be in fine spatial scales that may never have been measured.

4.14



SBUV noise components

$\sigma = S_{ii}^{1/2}$

Forward model parameters

$$\textit{forward model parameter error} = \mathbf{G}_y\mathbf{K}_b(\mathbf{b} - \hat{\mathbf{b}})$$

This one is easy. (In principle)

If you have estimated the forward model parameters properly, their individual errors will be unbiassed, so the mean error will be zero.

The covariance is:

$$\mathbf{S}_f = \mathbf{G}_y\mathbf{K}_b\mathbf{S}_b\mathbf{K}_b^T\mathbf{G}_y^T$$

where $\mathbf{S}_b$ is the error covariance matrix of $\mathbf{b}$, namely $\mathcal{E}\{(\mathbf{b} - \hat{\mathbf{b}})(\mathbf{b} - \hat{\mathbf{b}})^T\}$

However remember that this is most probably a systematic, not a random error.

Modelling error

$$\textit{modelling error} = \mathbf{G}_y\Delta\mathbf{f} = \mathbf{G}_y(\mathbf{f}(\mathbf{x}, \mathbf{b}, \mathbf{b}') - \mathbf{F}(\mathbf{x}, \mathbf{b}))$$

Note that this is evaluated at the true state, and with the true value of $\mathbf{b}$, but hopefully its sensitivity to these quantities is not large.

This can be hard to evaluate, because it requires a model $\mathbf{f}$ which includes the correct physics. If $\mathbf{F}$ is simply a numerical approximation for efficiency's sake, it may not be too difficult, but if $\mathbf{f}$ is not known in detail, or so horrendously complex that no proper model is feasible, then modelling error can be tricky to estimate.

This is also usually a systematic error.

An Error Covariance Matrix $\mathbf{S}$ is defined as

$$S_{ij} = \mathcal{E}\{\epsilon_i\epsilon_j\}$$

Diagonal elements are the familiar error variances.

The corresponding probability density function (PDF), if Gaussian, is:

$$P(\mathbf{y}) \propto \exp\left(-\frac{1}{2}(\mathbf{y} - \bar{\mathbf{y}})^T\mathbf{S}^{-1}(\mathbf{y} - \bar{\mathbf{y}})\right)$$

Contours of the PDF are

$$(\mathbf{y} - \bar{\mathbf{y}})^T\mathbf{S}^{-1}(\mathbf{y} - \bar{\mathbf{y}}) = const$$

i.e. ellipsoids, with arbitrary axes.

- How do we conceptualise an error covariance matrix?
- What corresponds to *error bars* for a profile?

We would really like a PDF to be of the form

$$P(\mathbf{z}) \propto \prod_i \exp(-z_i^2/2\sigma_i^2)$$

i.e. each z_i to have independent errors.

This can be done by diagonalising $\mathbf{S}$, and substituting $\mathbf{S} = \mathbf{L}\Lambda\mathbf{L}^T$, where $\mathbf{L}$ is matrix of eigenvectors $\mathbf{l}_i$. Then

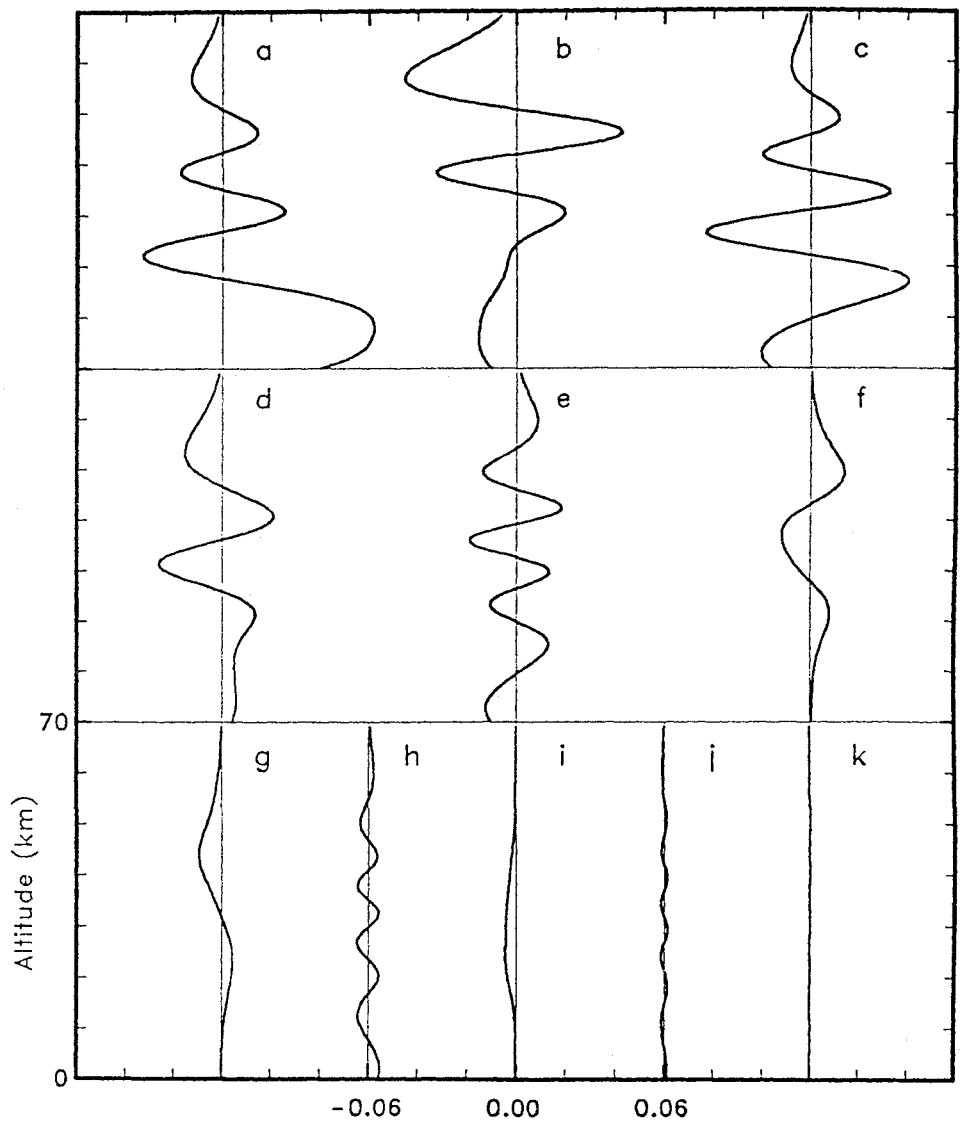
$$P(\mathbf{y}) \propto \exp\left(-\frac{1}{2}(\mathbf{y}-\bar{\mathbf{y}})^T\mathbf{L}\Lambda^{-1}\mathbf{L}^T(\mathbf{y}-\bar{\mathbf{y}})\right)$$

So if we put $\mathbf{z} = \mathbf{L}^T(\mathbf{y}-\bar{\mathbf{y}})$ then the z_i are independent with variance λ_i.

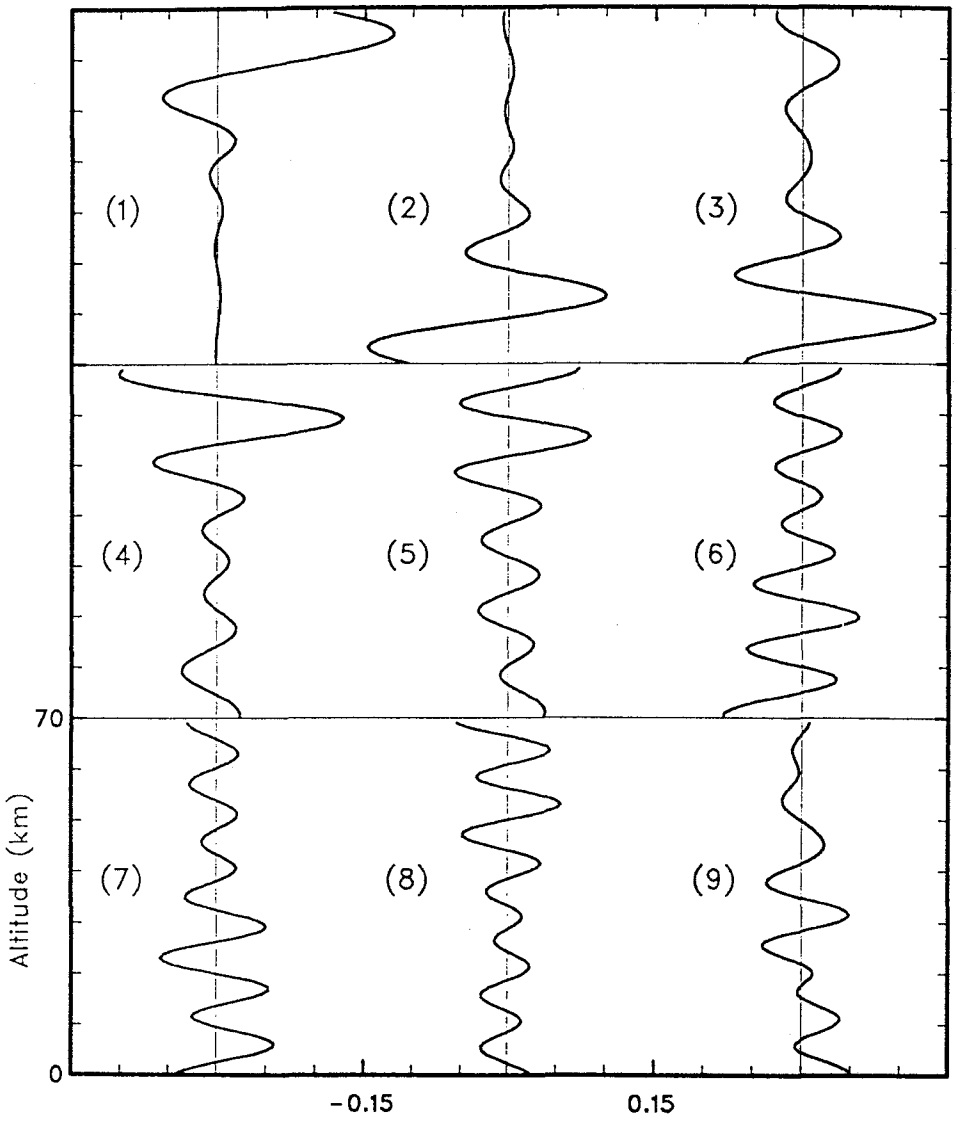
Thus we can express the error in e.g. a state vector estimate $\hat{\mathbf{x}}$, due to an error covariance $\hat{\mathbf{S}}$, in terms of *Error Patterns* $\mathbf{e}_i = \lambda_i^{\frac{1}{2}}\mathbf{l}_i$ such that the total error is of the form

$$\hat{\mathbf{x}} - \mathbf{x} = \sum_i \beta_i \mathbf{e}_i$$

where the error patterns are orthogonal, and the coefficients β are independent random variables with unit variance.



SBUV Retrieval noise error patterns



SBUV Smoothing error patterns - the 9 largest

We have derived the Bayesian description of the *ensemble* of possible solutions. To select *one* profile from the ensemble as *the solution* we can choose to optimise something.

- We have looked at the following characteristics of a retrieval:

  Averaging kernel; smoothing error; measurement error; modelling error

- Possible qualities that can be used to choose a profile include:

Maximum likelihood
– The profile at which $\partial P(\mathbf{x}|\mathbf{y})/\partial \mathbf{x} = 0$

Expectation value

$$\hat{\mathbf{x}} = \int P(\mathbf{x}|\mathbf{y})\mathbf{x}\, d\mathbf{x}$$

Minimum variance
– The method for which $\mathcal{E}\{(\mathbf{x}-\hat{\mathbf{x}})^T(\mathbf{x}-\hat{\mathbf{x}})\}$ is minimum.

Minimum measurement error
– The method for which $\mathbf{G}_y\mathbf{S}_y\mathbf{G}_y^T$ is minimum.

Minimum smoothing error
– The method for which $(\mathbf{A}-\mathbf{I})\mathbf{S}_a(\mathbf{A}-\mathbf{I})^T$ is minimum.

Minimum averaging kernel width
– We must first formally define the width of an averaging kernel.

Minimum modelling error
– The method for which modelling error and/or model parameter error is minimum.

Minimum total error
– The method for which $\mathbf{G}_y\mathbf{S}_y\mathbf{G}_y^T + (\mathbf{A}-\mathbf{I})\mathbf{S}_a(\mathbf{A}-\mathbf{I})^T$ is minimum.

ML maximises $P(\mathbf{y}|\mathbf{x})$ (the 'likelihood'), while MAP maximises $P(\mathbf{x}|\mathbf{y})$. The term 'ML' has been widely misused in retrieval to mean MAP. The difference is just the presence of *a priori* in the cost function.

For Gaussian noise and *a priori* we can find the MAP $\hat{\mathbf{x}}$ by maximising

$$-2\ln(P(\mathbf{x}|\mathbf{y}) = [\mathbf{y} - F(\mathbf{x})]^T \mathbf{S}_\epsilon^{-1}[\mathbf{y} - F(\mathbf{x})] + [\mathbf{x} - \mathbf{x}_a]^T \mathbf{S}_a^{-1}[\mathbf{x} - \mathbf{x}_a] + \text{const}$$

giving

$$\frac{d}{d\mathbf{x}}[-2\ln(P(\mathbf{x}|\mathbf{y})] = 0 = -\left[\frac{dF(\mathbf{x})}{d\mathbf{x}}\right]^T \mathbf{S}_\epsilon^{-1}[\mathbf{y} - F(\mathbf{x})] + \mathbf{S}_a^{-1}[\mathbf{x} - \mathbf{x}_a]$$

This cannot be solved explicitly for $\mathbf{x}$ because of the possibly non-linear term $F(\mathbf{x})$. So linearise $F(\mathbf{x})$ about some point $\mathbf{x}_0$ (which could be the same as $\mathbf{x}_a$):

$$F(\mathbf{x}) = F(\mathbf{x}_0) + \frac{dF(\mathbf{x})}{d\mathbf{x}}[\mathbf{x} - \mathbf{x}_0] = \mathbf{y}_0 + \mathbf{K}[\mathbf{x} - \mathbf{x}_0]$$

where $\mathbf{y}_0 = F(\mathbf{x}_0)$ and $\mathbf{K} = \frac{dF(\mathbf{x})}{d\mathbf{x}}$.
(Strictly, I should have linearised $\frac{dF(\mathbf{x})}{d\mathbf{x}}$ too, but it is often small)

Hence:

$$-\mathbf{K}^T \mathbf{S}_\epsilon^{-1}(\mathbf{y} - \mathbf{y}_0 - \mathbf{K}[\mathbf{x} - \mathbf{x}_0]) + \mathbf{S}_a^{-1}[\mathbf{x} - \mathbf{x}_a] = 0$$

which can be rearranged to give:

$$\mathbf{K}^T \mathbf{S}_\epsilon^{-1}(\mathbf{y} - \mathbf{y}_0 - \mathbf{K}[\mathbf{x}_a - \mathbf{x}_0]) = \mathbf{K}^T \mathbf{S}_\epsilon^{-1}\mathbf{K}[\mathbf{x} - \mathbf{x}_a] + \mathbf{S}_a^{-1}[\mathbf{x} - \mathbf{x}_a]$$

or:

$$\hat{\mathbf{x}} - \mathbf{x}_a = (\mathbf{K}^T \mathbf{S}_\epsilon^{-1}\mathbf{K} + \mathbf{S}_a^{-1})^{-1}\mathbf{K}^T \mathbf{S}_\epsilon^{-1}(\mathbf{y} - \mathbf{y}_0 - \mathbf{K}[\mathbf{x}_a - \mathbf{x}_0])$$

For the non-linear case, this can be used as an iteration, by using $\hat{\mathbf{x}}$ for $\mathbf{x}_0$ next time around.

The minimum variance solution is the state $\hat{\mathbf{x}}$ such that the variance about $\hat{\mathbf{x}}$ is minimised. Using the Bayesian solution p.d.f.:

$$\frac{\partial}{\partial \hat{\mathbf{x}}} \int (\mathbf{x} - \hat{\mathbf{x}})^2 P(\mathbf{x}|\mathbf{y})\, d\mathbf{x} = 0$$

which gives:

$$\hat{\mathbf{x}} = \int \mathbf{x} P(\mathbf{x}|\mathbf{y})\, dx$$

The minimum variance solution is the expected value for any p.d.f.

Minimum variance and maximum likelihood will be the same if the p.d.f. is symmetric about the maximum likelihood.

For the linear problem, the minimum variance solution is also the one which minimises the diagonal elements of:

$$\hat{\mathbf{S}} = E\{(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T\}$$

when $\hat{\mathbf{x}} = \mathbf{D}\mathbf{y}$. It is straightforward to show that the solution is

$$\mathbf{D} = E\{\mathbf{x}\mathbf{y}^T\}[E\{\mathbf{x}\mathbf{x}^T\}]^{-1} = \mathbf{S}_a\mathbf{K}^T(\mathbf{K}\mathbf{S}_a\mathbf{K}^T + \mathbf{S}_\epsilon)^{-1}$$

(Exercise!) This does not require Gaussian p.d.f's.

It is equivalent to multiple regression of $\mathbf{x}$ on $\mathbf{y}$, and can also be carried out by direct regression between measurements and independently measured states, without knowing $\mathbf{K}$.

Another exercise – show:

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

MINIMISE MEASUREMENT ERROR SENSITIVITY AND/OR SMOOTHING ERROR

- The error in the retrieval due to measurement error ϵ_y is $\mathbf{G}\epsilon_y$
  The retrieval method which minimises this error is found by choosing $\mathbf{G}$ to minimise the covariance $\mathbf{G}\mathbf{S}_y\mathbf{G}^T$.
  We need another constraint because the answer is $\mathbf{G} = \mathbf{O}$.
  The smoothing error would be rather poor!

- Smoothing error is $(\mathbf{I} - \mathbf{A})(\mathbf{x} - \mathbf{x}_a)$ with covariance $(\mathbf{I} - \mathbf{A})\mathbf{S}_a(\mathbf{I} - \mathbf{A})^T$.
  Put $\mathbf{A} = \mathbf{G}\mathbf{K}$ and minimise this with respect to $\mathbf{G}$ to find the retrieval method that minimises smoothing error:

$$0 = \frac{\partial}{\partial \mathbf{G}}(\mathbf{I} - \mathbf{G}\mathbf{K})\mathbf{S}_a(\mathbf{I} - \mathbf{G}\mathbf{K})^T = (\mathbf{I} - \mathbf{G}\mathbf{K})\mathbf{S}_a\mathbf{K}^T$$

  leading to

$$\mathbf{G} = \mathbf{S}_a\mathbf{K}^T(\mathbf{K}\mathbf{S}_a\mathbf{K}^T)^{-1}$$

  which is reminiscent of the smoothest exact solution, and is the same as the minimum variance solution with $\mathbf{S}_y$ omitted.
  Note that this gives an exact solution for the underconstrained problem, because $\mathbf{K}\mathbf{G} = \mathbf{I}$.

- We can minimise a weighted sum of the measurement error and smoothing err terms, to trade off noise sensitivity and smoothing error:

$$0 = \frac{\partial}{\partial \mathbf{G}}[(\mathbf{I} - \mathbf{G}\mathbf{K})\mathbf{S}_a(\mathbf{I} - \mathbf{G}\mathbf{K})^T + \gamma\mathbf{G}\mathbf{S}_y\mathbf{G}^T]$$

  This is identical to the minimum variance formulation if $\gamma = 1$, and the solution is:

$$\mathbf{G} = (\mathbf{K}\mathbf{S}_a\mathbf{K}^T + \gamma\mathbf{S}_y)^{-1}\mathbf{S}_a\mathbf{K}^T$$

  This allows a trade-off, but at the expense of departure from theoretical optimality.

5.04

RESOLUTION

- What do we mean by *resolution*:
  - separation of 'distinguishable' δ-functions?
  - 'width' of the δ-function response?
  - 'width' of the averaging kernel?
  - sine wave response?

- How do we define *width*?

  It should be possible to use in optimisation algebra, and should produce sensible results if $\mathbf{A}$ (or $A(z)$) has negative lobes.

  - Full width at half height - not helpful for algebra.

  - Second moment about mean

$$w(A(z)) = \left(\frac{\int A(z)(z - \bar{z})^2 dz}{\int A(z) dz}\right)^{\frac{1}{2}}$$

  where $\bar{z} = \int zA(z)dz / \int A(z)dz$. This is reasonable for positive $A(z)$.

  - Backus and Gilbert defined 'spread'

$$s(z_0) = 12 \int A^2(z)(z - z_0)^2 \, dz$$

  as the spread of the function $A(z)$ about the position z_0, for functions with unit area, $\int A(z)\, dz = 1$
  The factor 12 is chosen so that a top hat function has a spread equal to its width.
  Functions without unit area can be normalised before the spread is calculated.
  The quantity z_0 can be an arbitrary level, or the mean of $A(z)$.

5.05

Find the retrieval method $\mathbf{G}$ such that the resolution of the retrieved profile $\hat{\mathbf{x}} = \mathbf{Gy}$ is best, i.e. find $\mathbf{G}$ such that the spread of the averaging kernel at each level is minimum, and $\mathbf{A}$ has unit area.

Use continuous functions, because I can't do this in matrix algebra!

$$\hat{x}(z) = \sum_i G_i(z) y_i = \sum_i G_i(z) \int K_i(z') x(z') dz'$$

i.e.

$$\hat{x}(z) = \int A(z, z') x(z') dz'$$

where

$$A(z, z') = \sum_i G_i(z) K_i(z')$$

is the averaging kernel for level z.
Substitute into the expression for spread gives:

$$s(z) = 12 \int (z - z')^2 \sum_{ij} G_i(z) K_i(z') G_j(z) K_j(z') dz'$$

Define $Q_{ij}(z)$ as

$$Q_{ij}(z) = 12 \int (z - z')^2 K_i(z') K_j(z') dz'$$

which is a computable matrix function of z, then

$$s(z) = \sum_{ij} G_i(z) Q_{ij}(z) G_j(z)$$

We now minimise $s(z)$ subject to the unit area constraint.

$$\frac{\partial}{\partial G_k(z)} \left[\sum_{ij} G_i(z) Q_{ij}(z) G_j(z) + \lambda \sum_i G_i(z) k_i \right] = 0$$

where $k_i = \int K_i(z') dz'$.

5.06

This looks neater in matrix notation

$$\frac{\partial}{\partial \mathbf{g}(z)} \left[\mathbf{g}^T(z) \mathbf{Q}(z) \mathbf{g}(z) + \lambda \mathbf{g}^T(z) \mathbf{k} \right] = 0$$

but we must remember that the vector $\mathbf{g}$ and the matrix $\mathbf{Q}$ are functions of z. The solution is

$$\mathbf{g} = \frac{\mathbf{Q}^{-1}\mathbf{k}}{\mathbf{k}^T\mathbf{Q}^{-1}\mathbf{k}}$$

The spread is $s(z) = \mathbf{g}^T(z)\mathbf{Q}(z)\mathbf{g}(z)$. Substitute the solution:

$$s(z) = \frac{\mathbf{k}^T\mathbf{Q}^{-1}}{\mathbf{k}^T\mathbf{Q}^{-1}\mathbf{k}} \mathbf{Q} \frac{\mathbf{Q}^{-1}\mathbf{k}}{\mathbf{k}^T\mathbf{Q}^{-1}\mathbf{k}} = \frac{1}{\mathbf{k}^T\mathbf{Q}^{-1}\mathbf{k}}$$

The measurement noise of this solution is

$$\sigma^2(z) = \mathbf{g}^T \mathbf{S}_\epsilon \mathbf{g} = \frac{\mathbf{k}^T\mathbf{Q}^{-1}\mathbf{S}_\epsilon\mathbf{Q}^{-1}\mathbf{k}}{(\mathbf{k}^T\mathbf{Q}^{-1}\mathbf{k})^2}$$

and tends to be dreadful, as you might expect.

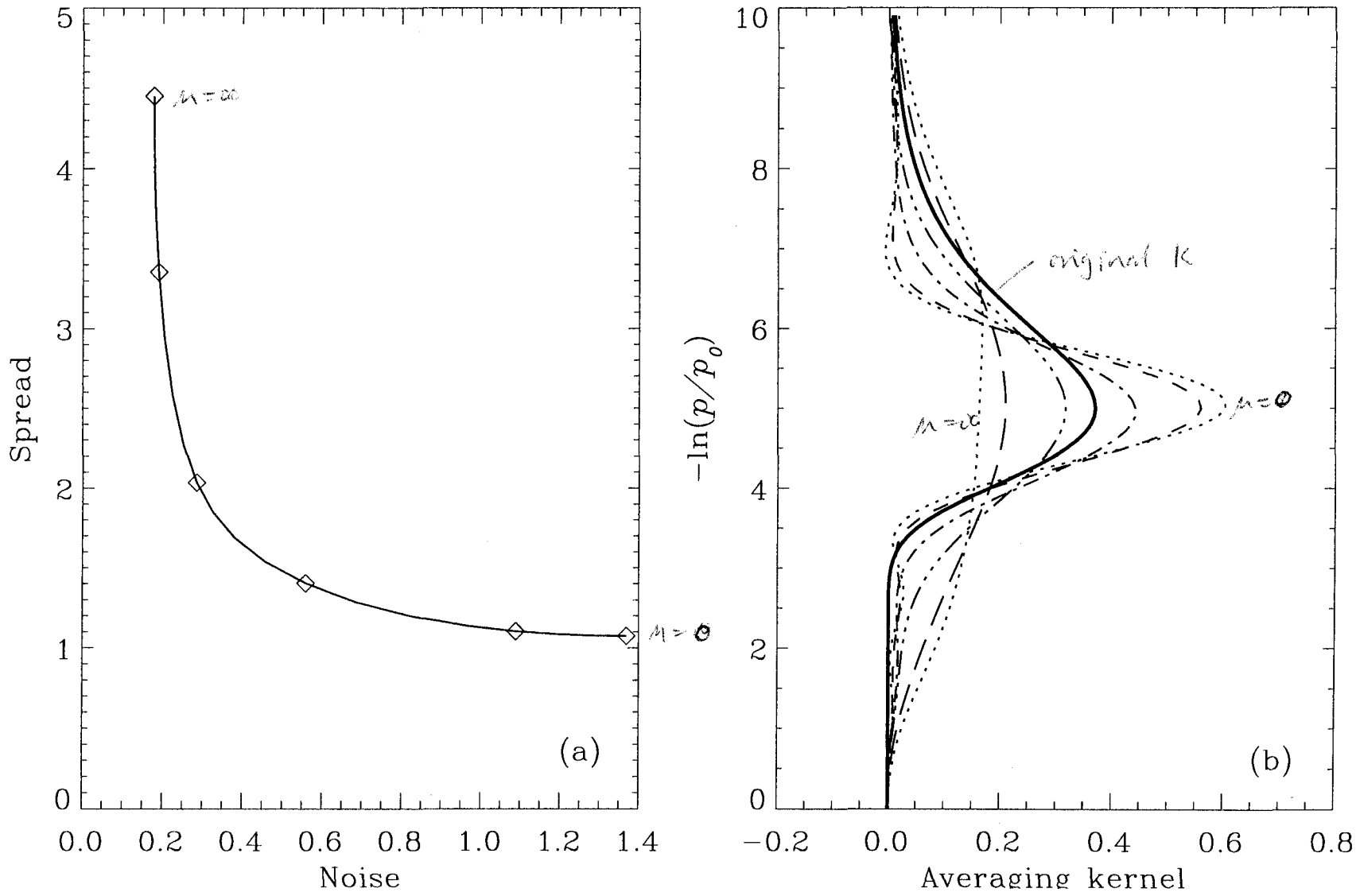
Therefore we minimise a weighted sum of spread and noise variance:

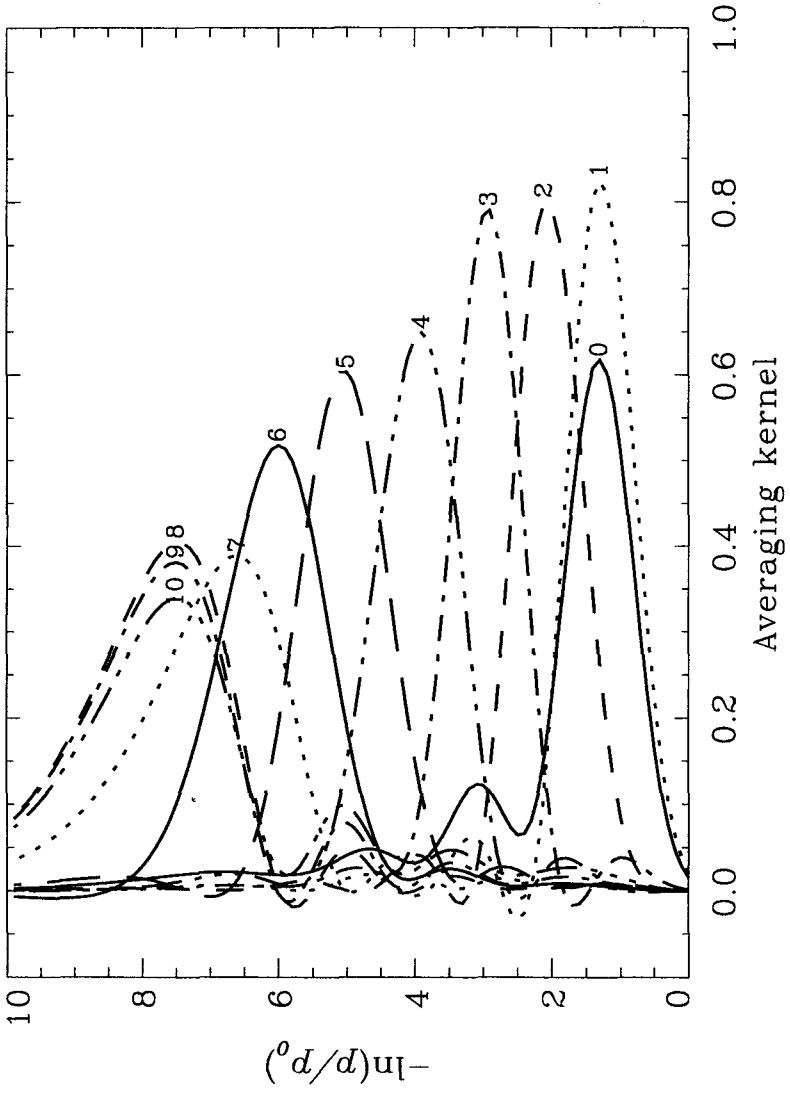
$$\frac{\partial}{\partial \mathbf{g}} \left[\mathbf{g}^T\mathbf{Q}\mathbf{g} + \lambda \mathbf{g}^T\mathbf{k} + \mu \mathbf{g}^T \mathbf{S}_\epsilon \mathbf{g} \right] = 0$$

where μ is a 'trade-off' parameter, trading resolution for noise performance. The solution is obviously

$$\mathbf{g} = \frac{(\mathbf{Q} + \mu\mathbf{S}_\epsilon)^{-1}\mathbf{k}}{\mathbf{k}^T(\mathbf{Q} + \mu\mathbf{S}_\epsilon)^{-1}\mathbf{k}}$$

5.07


Spread
Noise
(a)
λ=∞
λ=0
−ln(p/p_o)
Averaging kernel
original K
λ=∞
λ=0
(b)



- There are two sources of non-linearity of the solution:
  - A non-linear problem, i.e. a non-linear forward model
  - Non-Gaussian statistics.

  The Bayesian solution for Gaussian statistics is:

  $$-2\ln(P(\mathbf{x}|\mathbf{y})) = [\mathbf{y}-\mathbf{F}(\mathbf{x})]^T\mathbf{S}_\epsilon^{-1}[\mathbf{y}-\mathbf{F}(\mathbf{x})] + [\mathbf{x}-\mathbf{x}_a]^T\mathbf{S}_a^{-1}[\mathbf{x}-\mathbf{x}_a] + \text{const}$$

  The maximum likelikood solution requires the solution of:

  $$\nabla_{\mathbf{x}}[-2\ln(P(\mathbf{x}|\mathbf{y})] = 0 = -\mathbf{K}^T\mathbf{S}_\epsilon^{-1}[\mathbf{y}-\mathbf{F}(\mathbf{x})] + \mathbf{S}_a^{-1}[\mathbf{x}-\mathbf{x}_a]$$

  If the statistics were non-Gaussian, this equation would be non-linear in $\mathbf{x}$, even if F were linear.

  Once we have found a solution, we can linearise and do the error analysis. We can then find a solution pdf.

- But in the nonlinear case, how do we find a solution?
  - Numerical integration of $\int \mathbf{x}P(\mathbf{x}|\mathbf{y})\,d\mathbf{x}$ is expensive
  - Numerical minimisation of $-\ln(P(\mathbf{x}|\mathbf{y}))$
  - Linearise and iterate
  - *ad hoc* iteration which exploits the algebraic form
  - transform the problem to be as linear as possible

6.01

Newton's method for finding the zero of a scalar function $f(x)$ of one variable is

$$x_{n+1} = x_n - \left(\frac{df(x_n)}{dx}\right)^{-1} f(x_n)$$

The version for a vector-valued function of a vector, $\mathbf{g}(\mathbf{x})$, is:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - (\nabla_{\mathbf{x}}\mathbf{g}(\mathbf{x}_n))^{-1}\,\mathbf{g}(\mathbf{x}_n)$$

where the inverse is a matrix inverse.

If you apply this to the maximum a posteriori problem:

$$\mathbf{g}(\mathbf{x}) = -\mathbf{K}^T\mathbf{S}_\epsilon^{-1}[\mathbf{y} - \mathbf{F}(\mathbf{x})] + \mathbf{S}_a^{-1}[\mathbf{x} - \mathbf{x}_a] \qquad (1)$$

you get something quite complicated, because there is a term involving $\nabla_{\mathbf{x}}\mathbf{K}^T$ - an object with three subscripts:

$$\nabla_{\mathbf{x}}\mathbf{g} = -\nabla_{\mathbf{x}}\mathbf{K}^T\mathbf{S}_\epsilon^{-1}[\mathbf{y} - \mathbf{F}(\mathbf{x})] + \mathbf{K}^T\mathbf{S}_\epsilon^{-1}\mathbf{K} + \mathbf{S}_a^{-1}$$

If we ignore that term - it is usually small - we get:

$$\mathbf{x}_{n+1} = \mathbf{x}_n - (\mathbf{S}_a^{-1} + \mathbf{K}_n^T\mathbf{S}_\epsilon^{-1}\mathbf{K}_n)^{-1}(-\mathbf{K}_n^T\mathbf{S}_\epsilon^{-1}[\mathbf{y} - \mathbf{F}(\mathbf{x}_n)] + \mathbf{S}_a^{-1}[\mathbf{x}_n - \mathbf{x}_a])$$

This can also be obtained by linearising $\mathbf{F}(\mathbf{x})$ about $\mathbf{x}_n$ in (1) and solving for $\mathbf{x}$.

The iteration can be put in the form:

$$\mathbf{x}_{n+1} = \mathbf{x}_a + \mathbf{G}_n[\mathbf{y} - \mathbf{F}(\mathbf{x}_n)] + \mathbf{G}_n\mathbf{K}_n[\mathbf{x}_n - \mathbf{x}_a]$$

where

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

Converging to:

$$\mathbf{x}_\infty = \mathbf{x}_a + \mathbf{G}_\infty[\mathbf{y} - \mathbf{F}(\mathbf{x}_\infty)] + \mathbf{G}_\infty\mathbf{K}_\infty[\mathbf{x}_\infty - \mathbf{x}_a]$$

Put $\hat{\mathbf{x}} = \mathbf{x}_\infty$ and consider the case when $\mathbf{G}$ and $\mathbf{K}$ don't vary with n:

$$\mathbf{x}_{n+1} - \hat{\mathbf{x}} = -\mathbf{G}[\mathbf{F}(\mathbf{x}_n) - \mathbf{F}(\hat{\mathbf{x}})] + \mathbf{G}\mathbf{K}[\mathbf{x}_n - \hat{\mathbf{x}}]$$

Expand $\mathbf{F}(\mathbf{x})$:

$$\mathbf{x}_{n+1} - \hat{\mathbf{x}} = -\mathbf{G}[\mathbf{K}(\mathbf{x}_n - \hat{\mathbf{x}}) + O(\mathbf{x}_n - \hat{\mathbf{x}})^2] + \mathbf{G}\mathbf{K}[\mathbf{x}_n - \hat{\mathbf{x}}] = O(\mathbf{x}_n - \hat{\mathbf{x}})^2$$

Thus this has quadratic convergence whatever $\mathbf{G}$ might be!
(But it only converges to the right answer if $\mathbf{G}$ has the right value.)

If we allow $\mathbf{G}$ and $\mathbf{K}$ to vary with n, but in a well behaved way so that they can be expanded in $\mathbf{x} - \hat{\mathbf{x}}$ about $\hat{\mathbf{G}}$ and $\hat{\mathbf{K}}$, then the algebra is messy, but we can still show that the convergence is close to quadratic.

WARNING

Don't confuse $\mathbf{x}_a$ and $\mathbf{x}_n$ in a Gauss-Newton iteration!

Firstly, consider the meaning of what you are doing...
[Exercise for the student]

Then, see what happens:
The result is an iteration of the form

$$\begin{aligned}\mathbf{x}_{n+1} =& \mathbf{x}_n + \mathbf{G}_n[\mathbf{y} - \mathbf{F}(\mathbf{x}_n)] + \mathbf{G}_n\mathbf{K}_n[\mathbf{x}_n - \mathbf{x}_n] \\ =& \mathbf{x}_n + \mathbf{G}_n[\mathbf{y} - \mathbf{F}(\mathbf{x}_n)]\end{aligned}$$

i.e. a linear relaxation which tries to converge to an exact solution. In the limit:

$$\hat{\mathbf{x}} = \hat{\mathbf{x}} + \hat{\mathbf{G}}[\mathbf{y} - \mathbf{F}(\hat{\mathbf{x}})]$$

independent of $\mathbf{x}_a$. Take the difference:

$$\mathbf{x}_{n+1} - \hat{\mathbf{x}} = \mathbf{x}_n - \hat{\mathbf{x}} - \mathbf{G}(\mathbf{F}(\mathbf{x}_n) - \mathbf{F}(\hat{\mathbf{x}})) = (\mathbf{I}_n - \mathbf{GK})(\mathbf{x}_n - \hat{\mathbf{x}})$$

The convergence is first order, not second order, because $\mathbf{GK} \neq \mathbf{I}_n$:

$$\mathbf{I}_n - \mathbf{GK} = (\mathbf{S}_a^{-1} + \mathbf{K}^T\mathbf{S}_\epsilon^{-1}\mathbf{K})^{-1}\mathbf{S}_a^{-1}$$

HOW DO WE KNOW IT HAS CONVERGED?

- $\mathbf{y} - \mathbf{F}(\mathbf{x}_n)$ compared with $\mathbf{S}_\epsilon$?

  The goodness of the fit compared with the experimental error, for example:

  $$\chi^2 = (\mathbf{y} - \mathbf{F}(\mathbf{x}_n))^T\mathbf{S}_\epsilon^{-1}(\mathbf{y} - \mathbf{F}(\mathbf{x}_n))$$

  In any particular case $\mathbf{y} - \mathbf{F}(\mathbf{x}_n)$ may be larger or smaller than anticipated. It should only be $\sim \mathbf{S}_\epsilon$ on the average, and should have a χ^2-distribution. But $\mathbf{S}_\epsilon$ is the wrong covariance anyway...

- $\mathbf{F}(\mathbf{x}_{n+1})$ compared with $\mathbf{F}(\mathbf{x}_n)$?

  This is better. For quadratic convergence, the difference $\mathbf{F}(\mathbf{x}_n) - \mathbf{F}(\mathbf{x}_{n+1})$ is a good estimate of the convergence error in $\mathbf{y}_n$, so it is a useful overestimate of the error in $\mathbf{y}_{n+1}$. The difference should be small relative to $\mathbf{K}\hat{\mathbf{S}}\mathbf{K}^T + \mathbf{S}_\epsilon$:

  $$(\mathbf{F}(\mathbf{y}_{n+1}) - \mathbf{F}(\mathbf{x}_n))^T(\mathbf{K}\hat{\mathbf{S}}\mathbf{K}^T + \mathbf{S}_\epsilon)^{-1}(\mathbf{F}(\mathbf{y}_{n+1}) - \mathbf{F}(\mathbf{x}_n)) \ll m$$

  where m is the number of degrees of freedom, i.e. the expected value of $(\mathbf{y} - \mathbf{F}(\hat{\mathbf{x}}))^T(\mathbf{K}\hat{\mathbf{S}}\mathbf{K}^T + \mathbf{S}_\epsilon)^{-1}(\mathbf{y} - \mathbf{F}(\hat{\mathbf{x}}))$. $\mathbf{S}_\epsilon$ by itself is probably good enough for a convergence test.

- $\mathbf{x}_{n+1}$ compared with $\mathbf{x}_n$?

  Alternatively, the difference $\mathbf{x}_n - \mathbf{x}_{n+1}$ is a good estimate of the convergence error in $\mathbf{x}_n$, so it is a useful overestimate of the error in $\mathbf{x}_{n+1}$. The test is:

  $$\chi^2 = (\mathbf{x}_n - \mathbf{x}_{n+1})^T\hat{\mathbf{S}}^{-1}(\mathbf{x}_n - \mathbf{x}_{n+1}) \ll n$$

  because n is the expected value of $(\hat{\mathbf{x}} - \mathbf{x})\hat{\mathbf{S}}^{-1}(\hat{\mathbf{x}} - \mathbf{x})$.

- Does the retrieval fit the radiances?

  How does $\hat{\mathbf{y}} - \mathbf{y}$ fit with the appropriate covariance? Remembering that $\hat{\mathbf{y}} - \mathbf{y}_a = \mathbf{KG}(\mathbf{y} - \mathbf{y}_a)$, the correct covariance is given by:

$$\begin{aligned}\mathbf{S}_{\delta\hat{\mathbf{y}}} =& E\{(\hat{\mathbf{y}} - \mathbf{y})(\hat{\mathbf{y}} - \mathbf{y})^T\} \\ =& E\{(\mathbf{KG} - \mathbf{I})(\mathbf{y} - \mathbf{y}_a)(\mathbf{y} - \mathbf{y}_a)^T(\mathbf{KG} - \mathbf{I})^T\} \\ =& (\mathbf{KG} - \mathbf{I})(\mathbf{KS}_a\mathbf{K}^T + \mathbf{S}_\epsilon)(\mathbf{KG} - \mathbf{I})^T\end{aligned}$$

  For the optimal estimator, $\mathbf{G} = \mathbf{S}_a\mathbf{K}^T(\mathbf{KS}_a\mathbf{K}^T + \mathbf{S}_\epsilon)^{-1}$ so that $\mathbf{I} - \mathbf{KG} = \mathbf{S}_\epsilon(\mathbf{KS}_a\mathbf{K}^T + \mathbf{S}_\epsilon)^{-1}$ hence:

$$\chi^2 = (\hat{\mathbf{y}} - \mathbf{y})^T\mathbf{S}_\epsilon^{-1}(\mathbf{KS}_a\mathbf{K}^T + \mathbf{S}_\epsilon)\mathbf{S}_\epsilon^{-1}(\hat{\mathbf{y}} - \mathbf{y})$$

- Does the retrieval fit the climatology?

  We first need to know what the covariance of an ensemble of retrievals is:

$$\begin{aligned}\mathbf{S}_{\hat{\mathbf{x}}} =& E\{(\hat{\mathbf{x}} - \mathbf{x}_a)(\hat{\mathbf{x}} - \mathbf{x}_a)^T\} \\ =& E\{\mathbf{GK}(\mathbf{x} - \mathbf{x}_a)(\mathbf{x} - \mathbf{x}_a)^T\mathbf{K}^T\mathbf{G}^T + \mathbf{GS}_\epsilon\mathbf{G}^T\} \\ =& \mathbf{GKS}_a\mathbf{K}^T\mathbf{G}^T + \mathbf{GS}_\epsilon\mathbf{G}^T \\ =& \mathbf{GS}_{\mathbf{y}_a}\mathbf{G}^T\end{aligned}$$

  where $\mathbf{S}_{\mathbf{y}_a}$ is the radiance ensemble covariance. For the optimal estimator:

$$\mathbf{S}_{\hat{\mathbf{x}}} = \mathbf{S}_a\mathbf{K}^T(\mathbf{KS}_a\mathbf{K}^T + \mathbf{S}_\epsilon)^{-1}\mathbf{KS}_a = \mathbf{S}_a - \hat{\mathbf{S}}$$

  This has rank m, and will be singular if $m < n$, so that it is hard to compute $\chi^2 = (\hat{\mathbf{x}} - \mathbf{x}_a)^T\mathbf{S}_{\hat{\mathbf{x}}}^{-1}(\hat{\mathbf{x}} - \mathbf{x}_a)$. Transform to the non-null-space.

You can use this test to check whether there is something wrong with an individual measurement, or the climatology. The value of χ^2 is:

$$\chi^2 = (\mathbf{y} - \mathbf{y}_a)^T\mathbf{S}_{\mathbf{y}_a}^{-1}(\mathbf{y} - \mathbf{y}_a) = (\mathbf{y} - \mathbf{y}_a)^T(\mathbf{KS}_a\mathbf{K}^T + \mathbf{S}_\epsilon)^{-1}(\mathbf{y} - \mathbf{y}_a)$$

with m degrees of freedom. If χ^2 is outside the appropriate range, the cause may be: a problem with the climatological mean, the covariance, the forward model, or the measurement.
It may help to look at an ensemble of measurements, and compare their mean and covariance with the climatological mean and covariance.

- **Does the retrieval fit the radiances**, again

  Using

$$\hat{\mathbf{y}} - \mathbf{y} = (\mathbf{KG} - \mathbf{I})(\mathbf{y} - \mathbf{y}_a) = -\mathbf{S}_\epsilon(\mathbf{KS}_\mathbf{x}\mathbf{K}^T + \mathbf{S}_\epsilon)^{-1}(\mathbf{y} - \mathbf{y}_a)$$

  in the χ^2 for $\hat{\mathbf{y}} - \mathbf{y}$ we get:

$$\begin{aligned}\chi^2 &= (\hat{\mathbf{y}} - \mathbf{y})^T\mathbf{S}_\epsilon^{-1}(\mathbf{KS}_\mathbf{x}\mathbf{K}^T + \mathbf{S}_\epsilon)\mathbf{S}_\epsilon^{-1}(\hat{\mathbf{y}} - \mathbf{y}) \\ &= (\mathbf{y} - \mathbf{y}_a)^T(\mathbf{KS}_\mathbf{x}\mathbf{K}^T + \mathbf{S}_\epsilon)^{-1}\mathbf{S}_\epsilon \cdot \mathbf{S}_\epsilon^{-1}(\mathbf{KS}_\mathbf{x}\mathbf{K}^T + \mathbf{S}_\epsilon)\mathbf{S}_\epsilon^{-1} \\ &\qquad \cdot \mathbf{S}_\epsilon(\mathbf{KS}_\mathbf{x}\mathbf{K}^T + \mathbf{S}_\epsilon)^{-1}(\mathbf{y} - \mathbf{y}_a) \\ &= (\mathbf{y} - \mathbf{y}_a)^T(\mathbf{KS}_\mathbf{x}\mathbf{K}^T + \mathbf{S}_\epsilon)^{-1}(\mathbf{y} - \mathbf{y}_a)\end{aligned}$$

Thus if a measurement fits the measurement climatology, then an optimally retrieved measurement will fit the actual measurement with the same value of χ^2.
Computing it separately from $\hat{\mathbf{y}} - \mathbf{y}$ becomes a test of the numerics, the non-linearity, and the convergence.

VALIDATION

- Measurments vs. *a priori* climatology using a χ^2 test.
  - look at distribution of χ^2
  - correctness of the forward model
  - knowledge of the measurement error
  - knowledge of the climatological mean and covariance
  - climatology pdf
- Measurements vs. retrieved measurements using a χ^2 test
  - numerics
  - correct convergence test
  - measurements consistent with forward model
  - knowledge of the measurement error
- Comparison of retrievals with the climatology, looking for bias and unexpected variance about mean.
  - Determine whether the discrepancy is due to the retrieval or the climatology.
- Examine retrievals that fail to converge properly, and determine reasons.
- Examine retrievals that converge outside limits set by a measurement error χ^2 test, and determine reasons.
- Internal consistency, e.g:
  - comparison of retrievals for different instrument settings where appropriate;
  - comparison of co-located measurements on ascending/descending sections of the orbit.
- Comparison of retrievals with other instruments.
- Comparison with models, but taking due note of possible modelling errors.

EFFICIENCY

There are various ways of arranging the Newtonian iteration:

$$\begin{aligned}\mathbf{x}_{n+1} &= \mathbf{x}_n + (\mathbf{S}_a^{-1} + \mathbf{K}_n^T\mathbf{S}_\epsilon^{-1}\mathbf{K}_n)^{-1}(\mathbf{K}_n^T\mathbf{S}_\epsilon^{-1}[\mathbf{y} - \mathbf{F}(\mathbf{x}_n)] - \mathbf{S}_a^{-1}[\mathbf{x}_n - \mathbf{x}_a]) \\ &= \mathbf{x}_a + (\mathbf{S}_a^{-1} + \mathbf{K}_n^T\mathbf{S}_\epsilon^{-1}\mathbf{K}_n)^{-1}\mathbf{K}_n^T\mathbf{S}_\epsilon^{-1}([\mathbf{y} - \mathbf{F}(\mathbf{x}_n)] + \mathbf{K}_n[\mathbf{x}_n - \mathbf{x}_a])\end{aligned}$$

and, in the other form:

$$\mathbf{x}_{n+1} = \mathbf{x}_a + \mathbf{S}_a\mathbf{K}_n^T(\mathbf{S}_\epsilon + \mathbf{K}_n\mathbf{S}_a\mathbf{K}_n^T)^{-1}([\mathbf{y} - \mathbf{F}(\mathbf{x}_n)] + \mathbf{K}_n[\mathbf{x}_n - \mathbf{x}_a])$$

All of these require $\mathbf{F}(\mathbf{x}_n)$ and $\mathbf{K}_n$ to be evaluated at every iteration.

- You can't avoid evaluating $\mathbf{F}(\mathbf{x}_n)$ – that is the quantity that must converge to approximately $\mathbf{y}$
- You can avoid recalculating $\mathbf{K}_n$, with luck, if the problem is not too non-linear:
  - Start at $\mathbf{x}_0 = \mathbf{x}_a$, and use $\mathbf{K}_0$ for $\mathbf{K}_n$.
  - Compute $\mathbf{S}_a\mathbf{K}_0^T(\mathbf{S}_\epsilon + \mathbf{K}_0\mathbf{S}_a\mathbf{K}_0^T)^{-1}$ once, and use it each time.
  - $\mathbf{F}(\mathbf{x}_n)$ will converge to approximately $\mathbf{y}$, but not to the optimal solution.

The two solutions, $\hat{\mathbf{x}}$ and $\mathbf{x}_\infty$, satisfy:

$$\begin{aligned}\hat{\mathbf{x}} &= \mathbf{x}_a + \mathbf{S}_a\hat{\mathbf{K}}^T\mathbf{S}_\epsilon^{-1}[\mathbf{y} - \mathbf{F}(\hat{\mathbf{x}})] \\ \mathbf{x}_\infty &= \mathbf{x}_a + \mathbf{S}_a\mathbf{K}^T\mathbf{S}_\epsilon^{-1}[\mathbf{y} - \mathbf{F}(\mathbf{x}_\infty)]\end{aligned}$$

Putting $\mathbf{F}(\mathbf{x}_\infty) = \mathbf{F}(\hat{\mathbf{x}}) + \hat{\mathbf{K}}(\mathbf{x}_\infty - \hat{\mathbf{x}})$ gives the difference to first order:

$$\hat{\mathbf{x}} - \mathbf{x}_\infty = \mathbf{S}_a\hat{\mathbf{K}}^T\mathbf{S}_\epsilon^{-1}\hat{\mathbf{K}}(\mathbf{x}_\infty - \hat{\mathbf{x}}) + \mathbf{S}_a(\hat{\mathbf{K}}^T - \mathbf{K}^T)\mathbf{S}_\epsilon^{-1}[\mathbf{y} - \mathbf{F}(\hat{\mathbf{x}})]$$

hence

$$\hat{\mathbf{x}} - \mathbf{x}_\infty = (\mathbf{S}_a^{-1} + \hat{\mathbf{K}}^T\mathbf{S}_\epsilon^{-1}\hat{\mathbf{K}})^{-1}(\hat{\mathbf{K}}^T - \mathbf{K}^T)\mathbf{S}_\epsilon^{-1}[\mathbf{y} - \mathbf{F}(\hat{\mathbf{x}})]$$

You could re-evaluate $\mathbf{K}$ once, as a final stage.

A single iteration in the $n-$form is

$$\mathbf{x}_{i+1} = \mathbf{x}_a + (\mathbf{K}_i^T \mathbf{S}_\epsilon^{-1} \mathbf{K}_i + \mathbf{S}_a^{-1})^{-1} \mathbf{K}_i^T \mathbf{S}_y^{-1} [\mathbf{y} - \mathbf{F}(\mathbf{x}_i) + \mathbf{K}_i(\mathbf{x}_i - \mathbf{x}_a)]$$

and after the final iteration

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

The beginners method is usually something like

$\mathbf{W}_1 = \mathbf{S}_\epsilon^{-1} \mathbf{K}$	mn (*or* $1.5m^2n$)
$\mathbf{W}_2 = \mathbf{S}_a^{-1}$	0 (*or* $1.5n^3$)
$\mathbf{W}_3 = \mathbf{K}^T \mathbf{W}_1 + \mathbf{W}_2$	n^2m
$\mathbf{w}_1 = \mathbf{K}(\mathbf{x}_i - \mathbf{x}_a)$	nm
$\mathbf{w}_2 = \mathbf{W}_1^T(\mathbf{y} - \mathbf{F}(\mathbf{x}_i) + \mathbf{w}_1)$	nm
$\hat{\mathbf{S}} = \mathbf{W}_3^{-1}$	n^3
$\hat{\mathbf{x}} = \mathbf{x}_a + \hat{\mathbf{S}}\mathbf{w}$	n^2

In normal circumstances ($\mathbf{S}_y$ diagonal, $\hat{\mathbf{S}}$ required, $\mathbf{S}_a^{-1}$ precomputed), this takes $n(3m + n + nm + n^2)$.

However:

- Don't invert matrices! Gaussian elimination takes $n^3/3$ operations and back substitution takes $n^2/2$.
- Cholesky decomposition applies to symmetric matrices and is even faster, $n^3/6$.
- QR decomposition is better conditioned – we can avoid computing $\mathbf{K}^T\mathbf{K}$.

A single iteration in the $m-$form is

$$\mathbf{x}_{i+1} = \mathbf{x}_a + \mathbf{S}_a \mathbf{K}_i^T (\mathbf{S}_\epsilon + \mathbf{K}_i \mathbf{S}_a \mathbf{K}_i^T)^{-1} [\mathbf{y} - \mathbf{F}(\mathbf{x}_i) + \mathbf{K}_i(\mathbf{x}_i - \mathbf{x}_a)]$$

and after the final iteration

$$\hat{\mathbf{S}} = \mathbf{S}_a - \mathbf{S}_a \mathbf{K}_i^T (\mathbf{S}_\epsilon + \mathbf{K}_i \mathbf{S}_a \mathbf{K}_i^T)^{-1} \mathbf{K}_i \mathbf{S}_a$$

A Cholesky solution of

$$(\mathbf{S}_\epsilon + \mathbf{K}_i \mathbf{S}_a \mathbf{K}_i^T)\mathbf{z} = [\mathbf{y} - \mathbf{F}(\mathbf{x}_i) + \mathbf{K}_i(\mathbf{x}_i - \mathbf{x}_a)]$$

will take around $m^3/6$ operations, plus $n^2m + nm^2/2$ for evaluating the term $\mathbf{K}_i \mathbf{S}_a \mathbf{K}_i^T$. Then

$$\mathbf{x}_{i+1} = \mathbf{x}_a + \mathbf{S}_a \mathbf{K}_i^T \mathbf{z}$$

will only take a further nm.

Sequential updating

If S_ϵ is diagonal we can update the solution separately with each measurement y_j, and avoid the solution of simultaneous equations.

$$\mathbf{x}_i^j := \mathbf{x}_a + \frac{\mathbf{S}_{j-1}\mathbf{k}_j}{\mathbf{k}_j^T \mathbf{S}_{j-1} \mathbf{k}_j + \sigma_j^2} (y_j - F_j(\mathbf{x}_i^j) - \mathbf{k}_j^T(\mathbf{x}_i^j - \mathbf{x}_a))$$

$$\mathbf{S}_j := \mathbf{S}_{j-1} - \mathbf{S}_{j-1}\mathbf{k}_j\mathbf{k}_j^T\mathbf{S}_{j-1}/(\mathbf{k}_j^T\mathbf{S}_{j-1}\mathbf{k}_j + \sigma_j^2)$$

where $\mathbf{k}_j$ is the jth row of $\mathbf{K}$. Repeat this for each $j = 1 \ldots m$ to update from x_i to x_{i+1}.

TIME EVOLUTION OF THE STATE

- Consider measuring a system whose state is evolving according to a known set of equations:

$$\mathbf{x}_{t+1} = E_t(\mathbf{x}_t) + \boldsymbol{\xi}_t$$

where E_t represents the evolution operator, and $\boldsymbol{\xi}$ is a random vector representing *random forcing or model deficiencies.*

- We measure the state at every time t using a forward model:

$$\mathbf{y}_t = \mathbf{F}_t(\mathbf{x}_t) + \boldsymbol{\epsilon}_t$$

- The problem is to find the best estimate of the state at each time t using the measurements made so far, and possibly prior information.

Our knowledge of the state $\mathbf{x}_t$ at time t is $P(\mathbf{x}_t|\mathbf{y}_t, \mathbf{y}_{t-1}, \ldots)$

Prior knowledge at time $t+1$ is this pdf evolved according to the time evolution model:

$$P(\mathbf{x}_{t+1}|\mathbf{y}_t, \mathbf{y}_{t-1}, \ldots) = \int P(\mathbf{x}_{t+1}|\mathbf{x}_t) P(\mathbf{x}_t|\mathbf{y}_t, \mathbf{y}_{t-1}, \ldots) d\mathbf{x}_t$$

We make a measurement $\mathbf{y}_{t+1}$ at time $t+1$, and use Bayes theorem to obtain:

$$P(\mathbf{x}_{t+1}|\mathbf{y}_{t+1}, \mathbf{y}_t, \mathbf{y}_{t-1}, \ldots) = P(\mathbf{y}_{t+1}|\mathbf{x}_{t+1}) P(\mathbf{x}_{t+1}|\mathbf{y}_t, \mathbf{y}_{t-1}, \ldots) / P(\mathbf{y}_{t+1})$$

This describes *in principle* how we sequentially update a state estimate with new measurements.

THE KALMAN FILTER

The basic Kalman filter operates on the linear version of the problem:

$$\begin{aligned}\mathbf{x}_t &= \mathbf{E}_t \mathbf{x}_{t-1} + \boldsymbol{\xi}_t \\ \mathbf{y}_t &= \mathbf{K}_t \mathbf{x}_t + \boldsymbol{\epsilon}_t\end{aligned}$$

where $\mathbf{E}$ and $\mathbf{K}$ are known matrices, which may be time varying.

The filter operates sequentially in t, which we may think of as time, though it could be space.

At time $t-1$ an estimate of $\mathbf{x}_{t-1}$ has been made, namely $\hat{\mathbf{x}}_{t-1}$, with error covariance $\hat{\mathbf{S}}_{t-1}$.

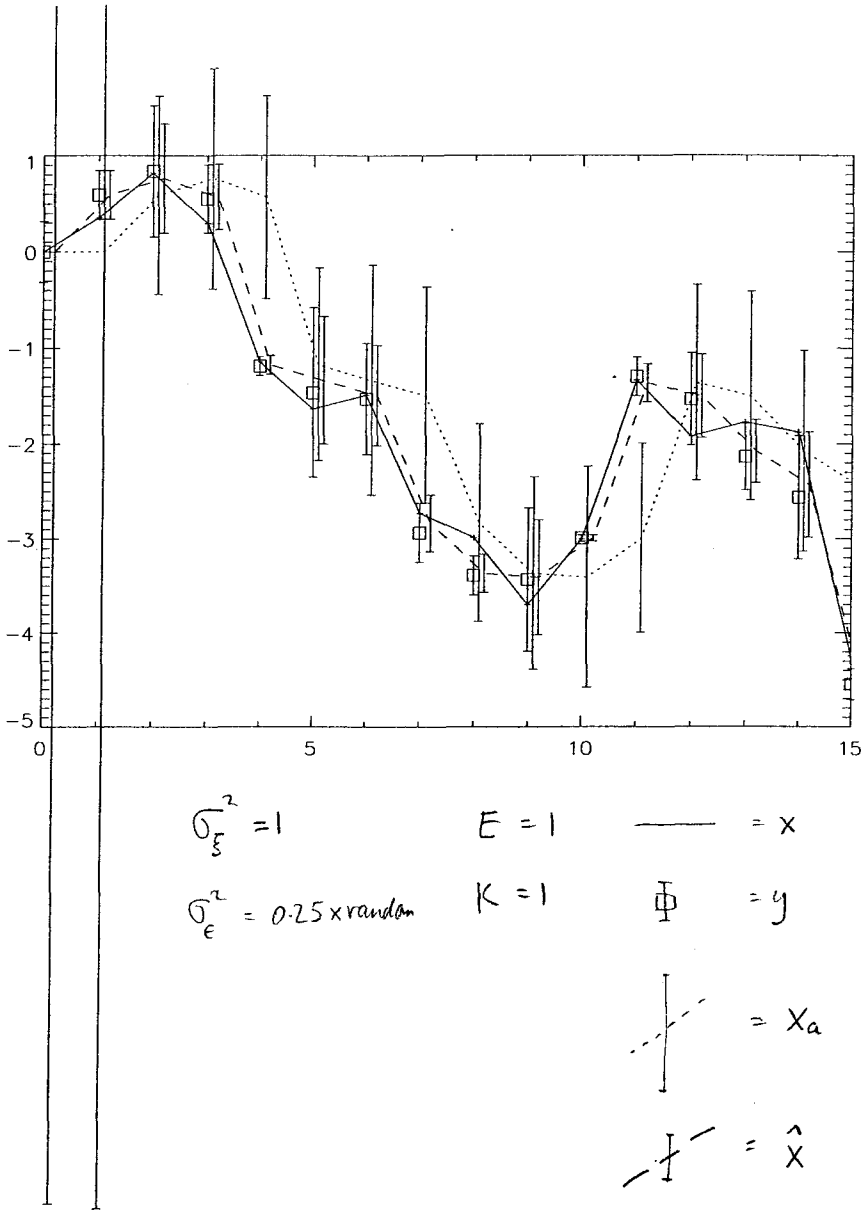
The stochastic prediction equation is used to make a prior estimate for time t:

$$\begin{aligned}\mathbf{x}_{at} &= \mathbf{E}_t \hat{\mathbf{x}}_{t-1} \\ \mathbf{S}_{at} &= \mathbf{E}_t \hat{\mathbf{S}}_{t-1} \mathbf{E}_t^T + \mathbf{S}_{\xi t}\end{aligned}$$

which is then combined optimally with the measurement at time t:

$$\begin{aligned}\mathbf{G}_t &= \mathbf{S}_{at} \mathbf{K}_t (\mathbf{K}_t \mathbf{S}_{at} \mathbf{K}_t^T + \mathbf{S}_\epsilon)^{-1} \\ \hat{\mathbf{x}}_t &= \mathbf{x}_{at} + \mathbf{G}_t (\mathbf{y}_t - \mathbf{K}_t \mathbf{x}_{at}) \\ \hat{\mathbf{S}}_t &= \mathbf{S}_{at} - \mathbf{G}_t \mathbf{K}_t \mathbf{S}_{at}\end{aligned}$$

where $\mathbf{G}_t$ is known as the Kalman gain matrix.

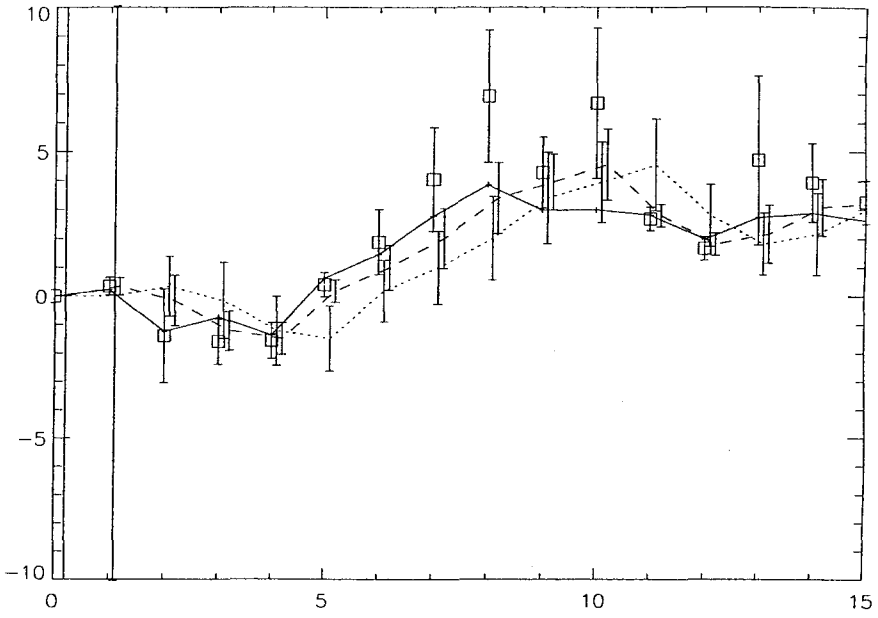


$\sigma_\xi^2 = 1$ $E = 1$ —— = x

$\sigma_\epsilon^2 = 0.25 \times \text{random}$ $K = 1$ ⊕ = y

= X_a

= $\hat{X}$



$\sigma_\xi^2 = 1$ $E = 1$ —— x

$\sigma_\epsilon^2 = 4 \times \text{random}$ $K = 1$ ⊕ y

X_a

$\hat{X}$

The non-linear case is dealt with by appropriate linearisation:

$$\mathbf{x}_{at} = E_t(\hat{\mathbf{x}}_{t-1})$$

is evaluated non-linearly, but it is linearised about $\hat{\mathbf{x}}_{t-1}$ to obtain the prior state estimate covariance $\hat{\mathbf{S}}_{at}$:

$$\mathbf{E}_t = \frac{\partial E_t(\hat{\mathbf{x}}_{t-1})}{\partial \mathbf{x}}$$
$$\mathbf{S}_{at} = \mathbf{E}_t \hat{\mathbf{S}}_{t-1} \mathbf{E}_t^T + \mathbf{S}_{\xi t}$$

The forward model is linearised at the measurement time about the prior estimate to obtain $\mathbf{K}_t$ for calculating the gain matrix:

$$\mathbf{K}_t = \frac{\partial F(\mathbf{x}_{at})}{\partial \mathbf{x}}$$
$$\mathbf{G}_t = \mathbf{S}_{at}\mathbf{K}_t(\mathbf{K}_t\mathbf{S}_{at}\mathbf{K}_t^T + \mathbf{S}_\epsilon)^{-1}$$

but the nonlinear forward model is used to update $\mathbf{x}_{at}$:

$$\hat{\mathbf{x}}_t = \mathbf{x}_{at} + \mathbf{G}_t(\mathbf{y}_t - F(\mathbf{x}_{at}))$$
$$\hat{\mathbf{S}}_t = \mathbf{S}_{at} - \mathbf{G}_t\mathbf{K}_t\mathbf{S}_{at}$$

If necessary this stage may be iterated as for the non-linear retrieval.

The filter uses only data at and before the estimate time. It is appropriate for real-time estimation.

If estimation is being done after the event, information from both sides of the estimate time can be used:

- Filter forward in time.
  At each estimate time save the estimate and its covariance
- Filter backwards in time. At each estimate time:
  - combine the backward *prior* estimate and its covariance with the forward estimate and its covariance, producing the smoothed estimate and its covariance
  - combine the backward prior estimate and its covariance with the measurement in the usual way to produce the backward filtered estimate.
  - *do not* include any forward data in the backward filter.

Consider filtering or smoothing a random walk sampled at fixed intervals

– No missing or bad data

– Measurement model: $y_t = x_t + \epsilon_t$, σ_ϵ^2 =constant

– Prediction model: $x_t = x_{t-1} + \xi_t$, σ_ξ^2 =constant.

Then:

A priori state and variance are:

$$x_t^a = \hat{x}_{t-1} \qquad \sigma_a^2 = \hat{\sigma}_{t-1}^2 + \sigma_\xi^2$$

Kalman gain is:

$$D_t = \frac{\hat{\sigma}_{t-1}^2 + \sigma_\xi^2}{\hat{\sigma}_{t-1}^2 + \sigma_\xi^2 + \sigma_\epsilon^2}$$

Updated state is:

$$\hat{x}_t = x_t^a + D_t(y_t - x_t^a) = (1 - D_t)x_t^a + D_t y_t$$

Updated variance is:

$$\hat{\sigma}_t^2 = \sigma_a^2 - D_t\sigma_a^2 = (1 - D_t)(\hat{\sigma}_{t-1}^2 + \sigma_\xi^2) = \frac{\sigma_\epsilon^2(\hat{\sigma}_{t-1}^2 + \sigma_\xi^2)}{\hat{\sigma}_{t-1}^2 + \sigma_\xi^2 + \sigma_\epsilon^2}$$

In the steady state, far from end effects, D_t and $\hat{\sigma}_t^2$ will tend to constant values.

$\hat{\sigma}^2$ tends to a value given by:

$$\hat{\sigma}^2(\hat{\sigma}^2 + \sigma_\xi^2 + \sigma_\epsilon^2) = \sigma_\epsilon^2(\hat{\sigma}^2 + \sigma_\xi^2)$$

or

$$\hat{\sigma}^2 = \frac{1}{2}(\sqrt{\sigma_\xi^2(\sigma_\xi^2 + \sigma_\epsilon^2)} - \sigma_\xi^2)$$

and D tends to

$$D = \frac{\hat{\sigma}^2 + \sigma_\xi^2}{\hat{\sigma}^2 + \sigma_\xi^2 + \sigma_\epsilon^2} = \frac{\hat{\sigma}^2}{\sigma_\epsilon^2}$$

The estimate tends to

$$\begin{aligned}\hat{x}_t &= Dy_t + (1 - D)\hat{x}_{t-1} \\ &= Dy_t + (1 - D)[Dy_{t-1} + (1 - D)\hat{x}_{t-2}] \\ &= D[y_t + (1 - D)y_{t-1} + (1 - D)^2 y_{t-2} + \ . \ . \ .]\end{aligned}$$

i.e. is an exponentially weighted mean of the historic measurements.

Kalman Smoothing will obviously give:

$$\hat{x}_t = \frac{D}{2 - D}[y_t + (1 - D)(y_{t-1} + y_{t+1}) + (1 - D)^2(y_{t-2} + y_{t+2}) + \ . \ .]$$

- If we really understand what's going on, there is no problem.
- If we don't we have to guess – assume an algebraic form, and fit parameters from samples of real data.

The simple case $x_t = x_{t-1}+\xi_t$ is unrealistic, because the statistics diverge:

$$\mathcal{E}\{x^2\} = \mathcal{E}\{x_t^2\} = \mathcal{E}\{(x_{t-1}+\xi_t)^2\} = \mathcal{E}\{x^2\} + \sigma_\xi^2$$

The modified equation $x_t = \alpha x_{t-1} + \xi_t$ works:

$$\mathcal{E}\{x^2\} = \mathcal{E}\{x_t^2\} = \mathcal{E}\{(\alpha x_{t-1}+\xi_t)^2\} = \alpha^2\mathcal{E}\{x^2\} + \sigma_\xi^2$$

So

$$\sigma_x^2 = \mathcal{E}\{x^2\} = \frac{\sigma_\xi^2}{1-\alpha^2}$$

However, we don't know what x_t is, so we have to use y_t.

We can estimate σ_ξ^2 and σ_ϵ^2 from the variance of $y_t - y_{t-1}$:

$$\mathcal{E}(y_t - y_{t-1})^2 = \mathcal{E}(\xi_t + \epsilon_t - \epsilon_{t-1})^2 = \sigma_\xi^2 + 2\sigma_\epsilon^2$$

Or lagging even further:

$$\mathcal{E}(y_t - y_{t-s})^2 = \mathcal{E}(\sum_{r=0}^{s-1}\xi_{t-r} + \epsilon_t - \epsilon_{t-s})^2 = s\sigma_\xi^2 + 2\sigma_\epsilon^2$$

A linear fit in s will give both σ_ξ^2 and σ_ϵ^2.

Estimate α from the lag correlation, $\mathcal{E}\{y_t y_{t-n}\}$, of the measured time series.

Note that

$$\begin{aligned} x_t =& \alpha x_{t-1} + \xi_t \\ =& \alpha^2 x_{t-2} + \alpha\xi_{t-1} + \xi_t \\ =& \alpha^n x_{t-n} + \sum_{j=0}^{n-1}\alpha^j\xi_{t-j} \end{aligned}$$

So that

$$\mathcal{E}\{x_t x_{t-n}\} = \mathcal{E}\{\alpha^n x_{t-n}^2 + \sum_{j=0}^{n-1}\alpha^j\xi_{t-j}x_{t-n}\} = \alpha^n\sigma_x^2 + 0$$

Hence the lag correlation of y is the same:

$$\begin{aligned} \mathcal{E}\{y_t y_{t-n}\} =& \mathcal{E}\{(x_t+\epsilon_t)(x_{t-n}+\epsilon_{t-n})\} \\ =& \mathcal{E}\{x_t x_{t-n}\} \qquad n \neq 0 \\ =& \alpha^n\sigma_x^2 \end{aligned}$$

For $n = 0$ there is an extra term:

$$\mathcal{E}\{y_t^2\} = \sigma_x^2 + \sigma_\epsilon^2$$

So we compute the lag correlation, subtract σ_ϵ^2 from the zero lag term, and try to fit $\alpha^n\sigma_x^2$ to the series.

We will usually find it doesn't fit, because the prediction equation is inadequate.

HIGHER ORDER PREDICTION MODELS

- Lag correlation may not be a simple exponential decrease.

  Adjust the prediction equation to fit the lag correlation.
  It may require a second (or more) order prediction equation, e.g.

$$x_t = \alpha_t x_{t-1} + \beta_t x_{t-2} + \xi_t$$

  which could be transformed to a first order equation with two unknowns, e.g:

$$x_t = \alpha_t x_{t-1} - \beta_t z_{t-1} + \xi_t$$
$$z_t = x_{t-1}$$

  where z_t is defined as x_{t-1}. This is called *state vector augmentation.*

- The general autoregressive prediction model with constant coefficients is

$$x_t = \sum_{i=1}^{n} \alpha_i x_{t-i} + \xi_t$$

  Consider the lag covariance s_j with lag $j \leq n$:

$$\begin{aligned} s_j &= \mathcal{E}\{x_t x_{t-j}\} \\ &= \sum_{i=1}^{n} \alpha_i \mathcal{E}\{x_{t-j} x_{t-i}\} \\ &= \sum_{i=1}^{n} \alpha_i s_{j-i} \end{aligned}$$

  Thus if we can estimate the lag covariances from a large sample of the time series, we can solve these linear equations for the model parameters.

  These are known as the Yule-Walker equations.

7.09

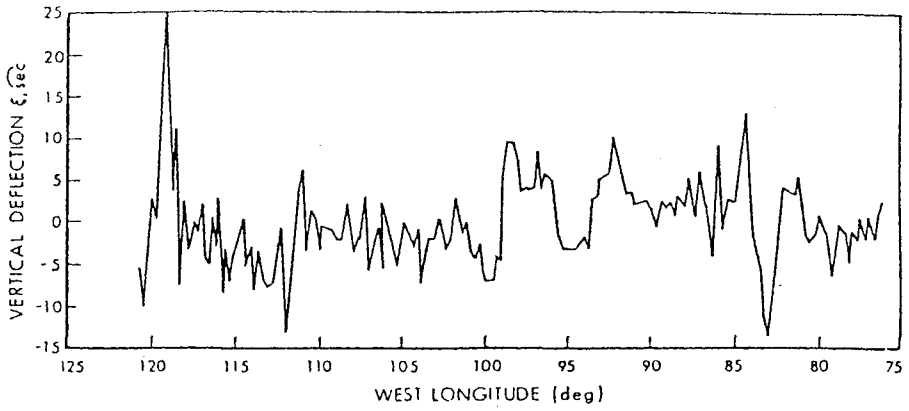


Figure 3.9-2 Meridian Component, Vertical Deflections of Gravity – 35th Parallel, United States (Ref. 9)

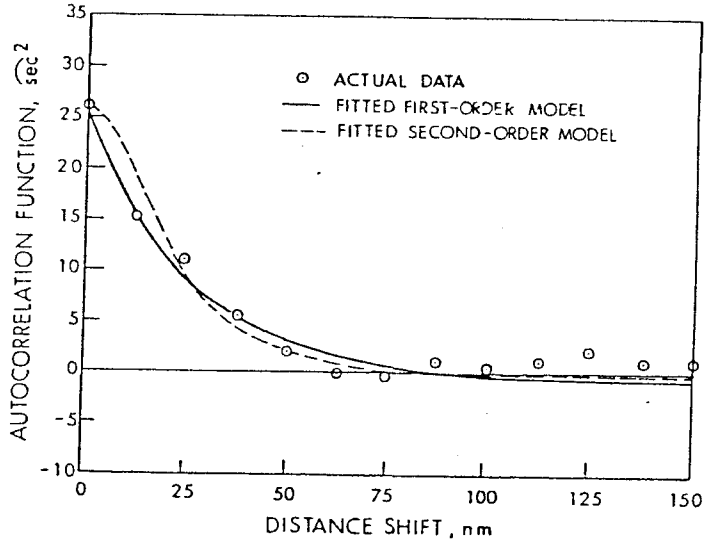


Figure 3.9-3 Vertical Deflection Sample Autocorrelation Function – 35th Parallel U.S.

- The classic application: Update of satellite orbital parameters.

  – The state vector is the orbit parameters.

  – The measurements are satellite position and velocity, known functions of the orbital parameters, at arbitrary time intervals.

- ISAMS temperature retrieval.

  – The state vector is the temperature at a set of tangent points and the pressure at one of these points.

  – The measurement is a set of radiances from a different subset of directions every 2.048s, corresponding to the tangent points plus the satellite roll angle

  – The stochastic equation is ad hoc

- Kalman filter mapping

  – Separately filtered at each latitude and height

  – The state vector is a set of fourier coefficients around a latitude circle

  – the measurement is a quantity measured at one longitude at each crossing of the measurement track

  – the stochastic equation is ad hoc, eg first order in amplitude and phase

- *Mainly Useless Methods: for illustration only*

  – Constrained exact solutions: linear or non-linear

  – Truncated eigenvalue expansions

  – Least squares

- *Sometimes Useful Methods*

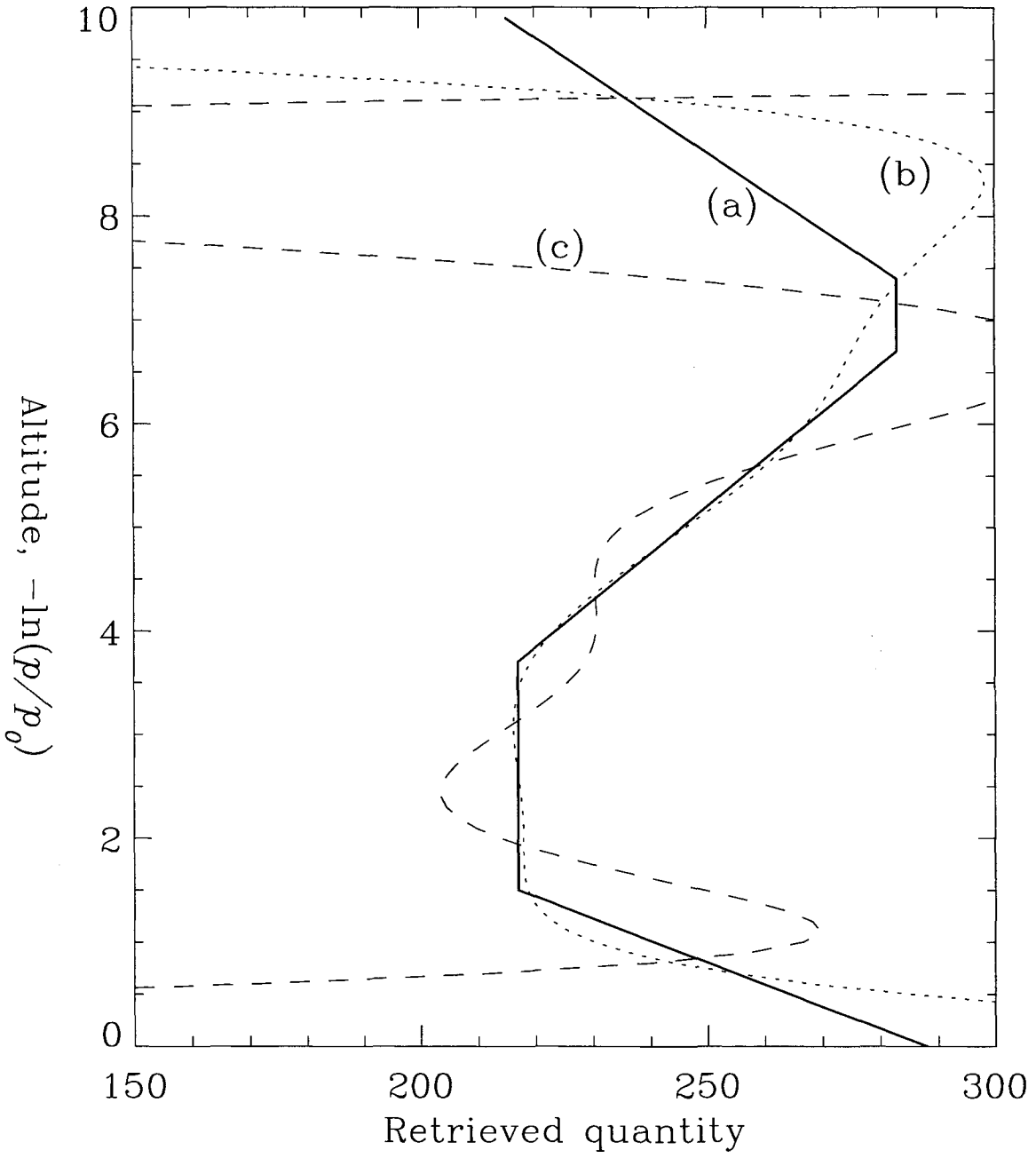
  – Constrained linear/non-linear: Twomey-Tikhonov

  – Linear Relaxation

  – Non-linear Relaxation: Chahine etc.

  – Approximations to optimal methods

  – Onion-peeling



(a) 'true' profile

(b) retrieved as a 7 degree polynomial - no noise

(c) ditto, with noise

- Virtually all practical retrieval problems are *formally* under-determined

  There are a finite number of measurements, m, and the state usually includes a continuous function of position, $X(z)$.

- The continuous part of the state can be constrained by a representation with n parameters, e.g:

$$X(z) = \sum_{i=1}^{n} W_i(z) x_i = \mathbf{w}^T(z)\mathbf{x}$$

  Can usually be done accurately enough for practical purposes, if n is large enough.

- To attempt an 'exact' solution, approximate state so that $m = n$

  - This may or may not make the problem well-determined. (Strictly n should equal the rank of $\mathbf{K}$.)
  - This may or may not be an accurate enough representation.

  The problem becomes:

$$\mathbf{y} = \mathbf{F}(\mathbf{x}) + \boldsymbol{\epsilon} = \mathbf{F}(\mathbf{x}_0) + \mathbf{K}(\mathbf{x} - \mathbf{x}_0) + \boldsymbol{\epsilon}$$

  where $\mathbf{K}$ is square, and the solution is:

$$\hat{\mathbf{x}} = \mathbf{x}_0 - \mathbf{K}^{-1}(\mathbf{y} - \mathbf{F}(\mathbf{x}_0))$$

  which may or may not have to be iterated.

The problem is in the error analysis, the solution converges to:

$$\hat{\mathbf{x}} = \mathbf{x} + \mathbf{K}^{-1}\boldsymbol{\epsilon}$$

The term $\mathbf{K}^{-1}\boldsymbol{\epsilon}$ can be large.

Express $\mathbf{K}$ in terms of its eigenvectors to examine the error term:

$$\mathbf{K} = \mathbf{R}\boldsymbol{\Lambda}\mathbf{L}^T \text{ and } \mathbf{K}^{-1} = \mathbf{R}\boldsymbol{\Lambda}^{-1}\mathbf{L}^T$$

so that

$$\mathbf{K}^{-1}\boldsymbol{\epsilon} = \mathbf{R}\boldsymbol{\Lambda}^{-1}\mathbf{L}^T\boldsymbol{\epsilon}$$

Eigenvectors are normalised, so the important bit is $\boldsymbol{\Lambda}^{-1}$.

If $\mathbf{K}$ has small eigenvalues, then $\hat{\mathbf{x}}$ will have large errors.

The ratio between the smallest and largest eigenvalues is the *condition number* of the matrix.

Transform the problem with $\mathbf{L}^T$. The solution is of the form:

$$\mathbf{L}^T\hat{\mathbf{x}} = \mathbf{L}^T\mathbf{x} - \boldsymbol{\Lambda}^{-1}\mathbf{L}^T\boldsymbol{\epsilon}$$

or

$$\hat{\mathbf{x}}' = \mathbf{x}' - \boldsymbol{\Lambda}^{-1}\boldsymbol{\epsilon}'$$

Thus elements of $\mathbf{x}'$ which correspond to small eigenvalues are not well determined.

The aim here is to deal with the near-null-space.

$$\hat{\mathbf{x}}' = \mathbf{x}' - \Lambda^{-1}\epsilon'$$

Thus elements of $\mathbf{x}'$ which correspond to small eigenvalues and are not well determined are set to zero, rather than to a random noise value.

Choose the cut off so that the signal to noise in the last retained term is about unity.

This can be done using SVD rather than eigenvectors, so that $\mathbf{K}$ does not have to be square.

However – the use of a minimum variance solution effectively combines these ill-determined components optimally with *a priori*, and doesn't involve computing eigenvectors. In a slightly different transformation:

$$\hat{\mathbf{x}}' = (\mathbf{I}_n + \Lambda^2)^{-1}(\Lambda\mathbf{y}' + \mathbf{x}'_a)$$

has a noise term:

$$(\mathbf{I}_n + \Lambda^2)^{-1}\Lambda\epsilon'$$

In truncated SVD the last term retained contains too much noise, the first term ignored contains information.

- *The over-constrained case*

  In the case where $m > n$ an exact solution is not normally possible. The least squares solution minimises:

  $$(\mathbf{y} - \mathbf{F}(\mathbf{x}))^T(\mathbf{y} - \mathbf{F}(\mathbf{x})) \quad \text{or} \quad (\mathbf{y} - \mathbf{K}\mathbf{x})^T(\mathbf{y} - \mathbf{K}\mathbf{x})$$

  The solution in the linear case is the standard *normal equations*:

  $$\hat{\mathbf{x}} = (\mathbf{K}^T\mathbf{K})^{-1}\mathbf{K}^T\mathbf{y}$$

  In the non-linear case you can, e.g. iterate a linearised version:

  $$\mathbf{x}_{n+1} = \mathbf{x}_n + (\mathbf{K}_n^T\mathbf{K}_n)^{-1}\mathbf{K}_n^T(\mathbf{y} - \mathbf{F}(\mathbf{x}_n))$$

  Note the similarity to the optimal solutions. It can be obtained by putting $\mathbf{S}_a^{-1} = \mathbf{O}$ and $\mathbf{S}_\epsilon = \mathbf{I}$.

- *The under-constrained case*

  In the case where $m < n$ there are an infinite number of exact solutions. The one whose state vector has the shortest length is found by

  minimising $\quad \mathbf{x}^T\mathbf{x} \quad \text{or} \quad (\mathbf{x} - \mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0)$

  subject to $\quad \mathbf{y} = \mathbf{K}\mathbf{x} \quad \text{or} \quad \mathbf{y} = \mathbf{F}(\mathbf{x})$

  The solution is, in the linear case:

  $$\mathbf{x} = \mathbf{K}^T(\mathbf{K}\mathbf{K}^T)^{-1}\mathbf{y}$$

  or in the non-linear case, iterate:

  $$\mathbf{x}_{n+1} = \mathbf{x}_n + \mathbf{K}_n^T(\mathbf{K}_n\mathbf{K}_n^T)^{-1}(\mathbf{y} - \mathbf{F}(\mathbf{x}_n))$$

  Note the similarity to the optimal solutions. It can be obtained by putting $\mathbf{S}_a = \mathbf{I}$ and $\mathbf{S}_\epsilon = \mathbf{O}$.

- *The optimal solution minimised a properly weighted sum of these two.*

This approach minimises a cost function:

$$(\mathbf{x}-\mathbf{x}_0)^T\mathbf{H}(\mathbf{x}-\mathbf{x}_0)+\gamma(\mathbf{y}-\mathbf{K}\mathbf{x})^T(\mathbf{y}-\mathbf{K}\mathbf{x})$$

where the first term represents some weighted departure from an *a priori* $\mathbf{x}_0$, and the second term constrains the solution to approximately fit the measurements. The factor γ is chosen to give appropriate relative weighting to the two constraints.

The matrix $\mathbf{H}$ can be chosen to minimise e.g.:

- the squared difference between $\mathbf{x}$ and $\mathbf{x}_0$ ($\mathbf{H}=\mathbf{I}$)
- the squared second difference for a smooth solution
- the log of a pdf. This would make it a statistically optimal method.

The solution is:

$$\hat{\mathbf{x}}=\mathbf{x}_0+(\gamma^{-1}\mathbf{H}+\mathbf{K}^T\mathbf{K})^{-1}\mathbf{K}^T(\mathbf{y}-\mathbf{K}\mathbf{x}_0)$$

Consider the case $\mathbf{H}=\mathbf{I}$, and use the singular vector transformation $\mathbf{K}=\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$, noting that $\mathbf{K}$ need not be square.

$$\hat{\mathbf{x}}=\mathbf{x}_0+(\gamma^{-1}\mathbf{I}+\mathbf{V}\mathbf{\Lambda}^2\mathbf{V}^T)^{-1}\mathbf{V}\mathbf{\Lambda}\mathbf{U}^T(\mathbf{y}-\mathbf{U}\mathbf{\Lambda}\mathbf{V}^T\mathbf{x}_0)$$
$$\hat{\mathbf{x}}'=\mathbf{x}_0'+(\gamma^{-1}\mathbf{I}+\mathbf{\Lambda}^2)^{-1}\mathbf{\Lambda}(\mathbf{y}'-\mathbf{\Lambda}\mathbf{x}_0')$$

where $\mathbf{x}'=\mathbf{V}^T\mathbf{x}$, $\mathbf{y}'=\mathbf{U}^T\mathbf{y}$ and $\mathbf{y}'=\mathbf{\Lambda}\mathbf{x}'+\epsilon'$.

This falls apart into separate equations, one for each element:

$$\hat{x}_i'=x_{0i}'+\frac{\lambda_i}{\gamma^{-1}+\lambda_i^2}(y_i'-\lambda_i x_{0i}')$$

elements for which $\lambda_i^2 \ll \gamma^{-1}$ contribute at reduced weight, elements for which $\lambda_i^2 \gg \gamma^{-1}$ contribute at full weight.

Linear relaxation is an iteration of the form:

$$\mathbf{x}_{n+1}=\mathbf{x}_n+\mathbf{D}(\mathbf{y}-\mathbf{F}(\mathbf{x}_n))$$

I.e. if $\mathbf{y}\neq\mathbf{F}(\mathbf{x})$ then something gets added to $\mathbf{x}$ to adjust it

$\mathbf{D}$ is chosen ad-hoc so that $(\mathbf{y}-\mathbf{F}(\mathbf{x}_n))$ decreases at each stage. Useful if $\mathbf{K}$ is difficult or impossible to obtain.

At convergence $\hat{\mathbf{x}}=\mathbf{x}_\infty$, so that we must have:

$$\hat{\mathbf{x}}=\hat{\mathbf{x}}+\mathbf{D}(\mathbf{y}-\mathbf{F}(\hat{\mathbf{x}}))$$

This implies that $\mathbf{y}=\mathbf{F}(\hat{\mathbf{x}})$ if $\mathbf{D}$ has rank m, i.e. an exact solution. Subtract:

$$(\mathbf{x}_{n+1}-\hat{\mathbf{x}})=(\mathbf{x}_n-\hat{\mathbf{x}})-\mathbf{D}(\mathbf{F}(\mathbf{x}_n)-\mathbf{F}(\hat{\mathbf{x}}))$$

Expand $\mathbf{F}(\mathbf{x})$ about $\hat{\mathbf{x}}$: $\mathbf{F}(\mathbf{x})=\mathbf{F}(\hat{\mathbf{x}})+\hat{\mathbf{K}}(\mathbf{x}-\hat{\mathbf{x}})+O(\mathbf{x}-\hat{\mathbf{x}})^2$ then

$$\begin{aligned}\mathbf{x}_{n+1}-\hat{\mathbf{x}}&=\mathbf{x}_n-\hat{\mathbf{x}}-\mathbf{D}\hat{\mathbf{K}}(\mathbf{x}_n-\hat{\mathbf{x}})+O(\mathbf{x}_n-\hat{\mathbf{x}})^2\\&=(\mathbf{I}-\mathbf{D}\hat{\mathbf{K}})(\mathbf{x}_n-\hat{\mathbf{x}})+O(\mathbf{x}_n-\hat{\mathbf{x}})^2\end{aligned}$$

If $\mathbf{D}=\hat{\mathbf{K}}^{-1}$, then you get quadratic convergence. This is the exact solution in the linear case.

For example the **H**-matrix for the squared second difference is, for order 10, of the form

$$\begin{array}{cccccccccc} 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2 & 5 & -4 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 6 & -4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 6 & -4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -4 & 6 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -4 & 6 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -4 & 5 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \end{array} \tag{6.33}$$

This matrix is singular (its columns sum to zero), so it cannot be thought of as the inverse of some covariance matrix, but it can be taken to be an information matrix. If the state vector is expressed in a polynomial representation, this constraint will be found not to constrain the constant and linear coefficient. The solution is

$$\hat{\mathbf{x}} = \mathbf{x}_a + (\gamma^{-1}\mathbf{H} + \mathbf{K}^T\mathbf{K})^{-1}\mathbf{K}^T(\mathbf{y} - \mathbf{K}\mathbf{x}_a) \tag{6.34}$$

which bears a stronger resemblance to the maximum likelihood method than either of the least squares methods, the only difference being the interpretation of the constraint matrices.

In the case of first order convergence, decompose $\mathbf{I} - \mathbf{D}\hat{\mathbf{K}}$ into eigenvectors:

$$\mathbf{I} - \mathbf{D}\hat{\mathbf{K}} = \mathbf{R}\Lambda\mathbf{L}^T$$

then convergence follows

$$\mathbf{x}_{n+1} - \hat{\mathbf{x}} = \mathbf{R}\Lambda\mathbf{L}^T(\mathbf{x}_n - \hat{\mathbf{x}}) + O(\mathbf{x}_n - \hat{\mathbf{x}})^2$$

or

$$\mathbf{L}^T(\mathbf{x}_{n+1} - \hat{\mathbf{x}}) = \Lambda\mathbf{L}^T(\mathbf{x}_n - \hat{\mathbf{x}}) + O(\mathbf{x}_n - \hat{\mathbf{x}})^2$$

i.e. you can express $(\mathbf{x}_n - \hat{\mathbf{x}})$ in terms of components such that each iteration multiplies the coefficient of the ith component by a factor λ_i.
The iteration converges if $|\lambda_i| < 1$ for all i.

For a reasonable iteration you will find that the small eigenvalues correspond to large scale structure, and vice versa, and you can stop iterating before the fine structure blows up.

The solution is of the form:

$$\hat{\mathbf{x}} = \mathbf{x}_0 + \mathbf{D}\mathbf{a}$$

where $\mathbf{a} = \sum_n \mathbf{y} - \mathbf{F}(\mathbf{x}_n)$, i.e. a linear combination of the columns of $\mathbf{D}$.

If you know $\mathbf{K}$, a good choice is something like:

$\mathbf{D} = \mathbf{K}_0^T(\mathbf{K}_0\mathbf{K}_0^T + \gamma\mathbf{I})^{-1}$ in the underconstrained case

$\mathbf{D} = (\mathbf{K}_0^T\mathbf{K}_0 + \gamma\mathbf{I})^{-1}\mathbf{K}_0^T$ in the overconstrained case.

Non-linear relaxation was popularised by Chahine:

$$x_i^{n+1} = x_i^n \frac{y_i}{F_i(\mathbf{x}^n)}$$

where the state represented by values of the profile at the peaks of the weighting functions, so each x_i corresponds to a y_i.

For well peaked weighting functions, intuition tells us this will converge to a profile which produces the right measurements.

Convergence can be analysed by taking logs:

$$\ln x_i^{n+1} = \ln x_i^n + (\ln y_i - \ln F_i(\mathbf{x}^n))$$

We can see that this is the same as the linear relaxation, with:

– a transformed state vector whose elements are logs of the original,

– a similarly transformed measurement vector,

– $\mathbf{D} = \mathbf{I}$.

The transformed weighting function has elements

$$K'_{ij} = \frac{x_j}{F_i(\mathbf{x})}\frac{\partial F_i(\mathbf{x})}{\partial \mathbf{x}_j}$$

and the convergence depends on the eigenvalues of $\mathbf{I} - \mathbf{K}'$.

This is useful if $\mathbf{K}$ is difficult or expensive to evaluate, and the state representation is appropriate.

Table 6.1 Eigenvalues corresponding to the eigenvectors in Fig. 6.3(a), and number of iterations required to reduce error by a factor of 10. The negative eigenvalue corresponds to an error component that alternates in sign as the iteration proceeds.

Number	Eigenvalue	Iterations
1	0.0371	0.7
2	0.3426	2.2
3	-0.6033	4.6
4	0.6413	5.2
5	0.8444	13.6
6	0.9454	41
7	0.9847	149
8	0.9965	666

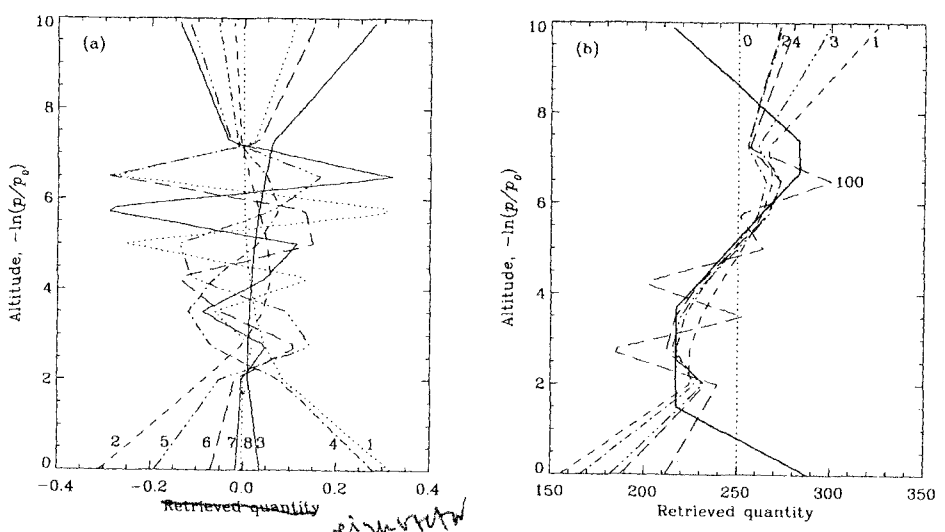


Fig. 6.3 (a) Eigenvector analysis for the convergence of the linear relaxation retrieval. The annotation corresponds to the eigenvalue number in table 6.1. (b) Convergence of a linear relaxation retrieval for the standard case. The true profile is the solid line, and the iterations are labelled with iteration number.

This version does not need a representation determined by the positions of the peaks of the weighting functions. Any reasonable vertical grid will do.

The iteration given by Twomey et al (1977) is:

$$x_j^{n+1} = x_j^n \frac{y_i}{F_i(\mathbf{x}^n)} \tilde{K}_{ij} + x_j^n [1 - \tilde{K}_{ij}]$$

j is the height index, i is the channel index, n is the iteration number, $\tilde{K}_{ij}$ is normalised so that the peak value is unity.
(Although the description of the iteration is not clear).
It can be rearranged as:

$$x_j^{n+1} = x_j^n \prod_i [1 + \frac{y_i - F_i(\mathbf{x}^n)}{F_i(\mathbf{x}^n)} \tilde{K}_{ij}]$$

Taking logs:

$$\ln x_j^{n+1} = \ln x_j^n + \sum_i \ln [1 + \frac{y_i - F_i(\mathbf{x}^n)}{F_i(\mathbf{x}^n)} \tilde{K}_{ij}]$$

Close to the solution we can expand the logs:

$$\ln x_j^{n+1} = \ln x_j^n + \sum_i \frac{y_i - F_i(\mathbf{x}^n)}{F_i(\mathbf{x}^n)} \tilde{K}_{ij}$$

which is a linear relaxation with

$$D_{ij} = \frac{\tilde{K}_{ij}}{F_i(\mathbf{x}^n)}$$

This relaxation requires $\mathbf{K}$ to be evaluated. If that is to be done, and optimal estimator might be more appropriate.

With emphasis on infrared measurements...

Considerations:

- Choice of state vector for convenience and/or linearity
  - coordinates: height or pressure?
  - temperature, pressure or density? mixing ratio or concentration? log or linear?
- Is the calibration part of the forward model?
  - could include calibration parameters in the state vector
- Approximations in forward model for speed and accuracy
  - Avoiding line-by-line
- Efficient methods of computation of derivatives:
  - Perturbing the forward model code. (easiest)
  - Coding the derivative of the forward model algebra
  - Coding the derivative of the forward model code (most satisfactory)

Nadir sounding of the temperature profile, e.g. O_2 *or* CO_2.

The equation of transfer is of the form:

$$L(\nu) = \int_0^{\infty} B(\nu, T(z)) \frac{d\tau(\nu, z, \infty)}{dz} dz + \ldots$$
$$= \int_0^{p_0} B(\nu, T(p)) \frac{d\tau(\nu, 0, p)}{dp} dp + \ldots$$

Something like B is the obvious state variable, for near linearity.

But is z or p the most appropriate coordinate?

Transmittance is given by

$$\tau(\nu, z, \infty) = \exp\left(- \int_z^{\infty} k[\nu, p(z'), T(z')] \rho_a(z') dz' \right)$$

and the density of absorber (assuming a known constant mass mixing ratio c) is such that $\rho_a(z)dz = (c/g)dp$. It is much simpler in p:

$$\tau(\nu, 0, p) = \exp\left(- \int_0^{p} k(\nu, p', T(p')) \frac{c}{g} dp' \right)$$

It is easy to see that the profile is indeterminate in terms of z, because you could move the whole atmosphere vertically by n km, and make no difference to the measured radiance.

CHOICE OF STATE VECTOR: EXAMPLE II

Temperature profile from solar occultation

The equation of transfer is of the form:

$$L(\nu, z_t) = L_o(\nu) \exp\left(-\int_{-\infty}^{\infty} k[\nu, p(s), T(s)] c\rho(s) ds \right)$$

where s is distance from the tangent point along the LOS.
The relation between z_t, z and s, ignoring refraction, is roughly:

$$(R_e + z_t)^2 + s^2 = (R_e + z)^2$$

Measurements are made at known intervals in z_t, so a height grid is probably most convenient.

If $k[\nu, p(s), T(s)]$ is fairly constant (e.g. UV), then $\rho(z)$ is going to be the most linear and straightforward state variable.

$$\rho(s) ds = \rho(z(s)) \frac{ds}{dz} dz$$

and obtain $p(z)$ and $T(z)$ by integrating the hydrostatic equation downwards from the top:

$$p(z) = \int_z^{\infty} \rho(z') g(z') dz' \quad \text{and} \quad T(z) = \frac{Mg}{R} \frac{p(z)}{\rho(z)}$$

You may need to define a top level z_T, and include $p(z_T)$ in the state vector.

Refraction is best done in terms of $\rho(z)$.

Remember to convert the error covariance:

$$\mathbf{S}_T = \mathbf{C}\mathbf{S}_\rho\mathbf{C}^T \quad \text{where} \quad \mathbf{C} = \frac{\partial \mathbf{T}}{\partial \boldsymbol{\rho}}$$

CHOICE OF STATE VECTOR: EXAMPLE III

Temperature profile from limb emission

This one is messy:

$$L(\nu, z_t) = \int_{-\infty}^{\infty} B(\nu, T(z(s))) \frac{d\tau(\nu, s, \infty)}{ds} ds$$

Measurements are made at known intervals in z_t, so a height grid is probably most convenient. Transmittance is given by

$$\tau(s) = \exp\left(-\int_s^{\infty} k[\nu, p(s'), T(s')] c\rho(s') ds' \right)$$

The mass path is best expressed in terms of density, but the Planck function is best expressed in terms of temperature.
B or T is probably the best choice for linearity.

The hydrostatic equation has to be used to relate ρ and T. If $T(z)$ is the used:

$$\ln p(z) = \ln p(z_0) + \int_{z_0}^{z} \frac{Mg}{RT(z')} dz'$$

so we need pressure at a reference height as part of the state vector.

$\Rightarrow T(z)$ alone is inadequate.

But if we do not have accurate absolute pointing, then z has to be a relative to some tangent view from the instrument.

$\Rightarrow T(p)$ alone is inadequate too.
You can get $\rho(p)$ directly from the equation of state, and then a relative $z(p)$ from the hydrostatic equation, but you need e.g. the tangent pressure of a reference scan direction as part of the state vector.

And relating a geometric scan to a pressure grid is messy.

$\rho(z)$ where the z's are the measurement tangent point grid is the only state variable that doesn't need a reference.

Atmospheric model:

- State vector is mixing ratio x_j in layers
- n layers, $1 \ldots n$
- $n+1$ levels, $0 \ldots n$, surface to space.
- Planck function of layer i is B_i, of the surface is B_g.
- Transmittance from level i to level n is τ_i.

Radiance y emitted by atmosphere is:

$$y = \sum_{\nu} f_\nu y_\nu$$

- f_ν is the normalised filter function, $\sum_\nu f_\nu = 1$,
- y_ν is the radiance at wavenumber index ν

The equation of transfer gives for y_ν:

$$y_\nu = \int_0^1 B\, d\tau = \sum_{i=1}^{n} B_i(\tau_{\nu i} - \tau_{\nu,i-1}) + B_g \tau_{\nu 0} = B_n + \sum_{i=0}^{n-1} b_i \tau_{\nu i}$$

thus defining b_i

Now look at the transmittance:

$$\tau_{\nu i} = \exp(-\chi_{\nu i}) \quad \text{where} \quad \chi_{\nu i} = \sum_{j=i+1}^{n} \kappa_{\nu j} m_j x_j$$

where in layer j:
- $\kappa_{\nu j}$ is the absorption coefficient,
- m_j is the mass of air and
- x_j is the mass mixing ratio.

Then:

$$y_\nu = B_n + \sum_{i=0}^{n-1} b_i \exp\Big(-\sum_{j=i+1}^{n} \kappa_{\nu j} m_j x_j\Big)$$

Both sums here can be accumulated from the top in one loop.

The derivative is

$$K_{\nu k} = \frac{\partial y_\nu}{\partial x_k} = -\sum_{i=0}^{n-1} b_i \exp\Big(-\sum_{j=i+1}^{n} \kappa_{\nu j} m_j x_j\Big) \cdot \begin{cases} \kappa_{\nu k} m_k & \text{for } i < k \\ 0 & \text{for } i \geq k \end{cases}$$

hence

$$K_{\nu k} = \kappa_{\nu k} m_k \sum_{i=0}^{k-1} b_i \exp\Big(-\sum_{j=i+1}^{n} \kappa_{\nu j} m_j x_j\Big)$$

which can be computed at the same time as y_ν:

- Accumulate χ_i downwards from the top on one loop.

- In a subsequent loop accumulate y_ν upward from the bottom, computing $K_{\nu k}$ from the partial sums as you go.

The extra cost of computing the weighting function at the same time as the radiance is one multiplication in the inner loop. Almost negligible.

And finally

$$K_k = \sum_{\nu} f_\nu \kappa_{\nu k} \quad \text{and} \quad y = \sum_{\nu} f_\nu y_\nu$$

The eigenvalue problem associated with an **A**rbitrary square matrix $\mathbf{A}$, of order n, is to find *eigenvectors* $\mathbf{l}$ and scalar *eigenvalues* λ which satisfy

$$\mathbf{A}\mathbf{l} = \lambda\mathbf{l}$$

If $\mathbf{A}$ is a coordinate transformation, then $\mathbf{l}$ has the same representation in the untransformed and transformed coordinates, apart from a factor λ.

This is the same as $(\mathbf{A} - \lambda\mathbf{I})\mathbf{l} = 0$, a homogeneous equation, which can only have a solution if $|\mathbf{A} - \lambda\mathbf{I}| = 0$, giving a polynomial equation of degree n, with n solutions for λ. They will be complex in general.

An eigenvector can be scaled by an arbitrary factor. It is conventional to normalise them so that $\mathbf{l}^T\mathbf{l} = 1$ or $\mathbf{l}^\dagger\mathbf{l} = 1$ (Hermitian adjoint)

Matrix of eigenvectors, $\mathbf{L}$:

$$\mathbf{A}\mathbf{L} = \mathbf{L}\Lambda$$

where the columns of $\mathbf{L}$ are the eigenvectors, and Λ is a diagonal matrix, with the eigenvectors on the diagonal.

Transpose	$\mathbf{L}^T\mathbf{A}^T = \Lambda\mathbf{L}^T$
Multiply by $\mathbf{R} = (\mathbf{L}^T)^{-1}$	$\mathbf{A}^T = \mathbf{R}\Lambda\mathbf{L}^T$
Postmultiply by $\mathbf{R}$	$\mathbf{A}^T\mathbf{R} = \mathbf{R}\Lambda$

Thus $\mathbf{R}$ is the matrix of eigenvectors of $\mathbf{A}^T$.
$\mathbf{A}^T$ has the same eigenvalues as $\mathbf{A}$.

In the case of a **S**ymmetric matrix, $\mathbf{S} = \mathbf{S}^T$ we must have $\mathbf{L} = \mathbf{R}$, so that $\mathbf{L}^T\mathbf{L} = \mathbf{L}\mathbf{L}^T = \mathbf{I}$ or $\mathbf{L}^T = \mathbf{L}^{-1}$, and the eigenvectors are *orthogonal*.
The eigenvalues are real and positive.

EIGENVECTORS - GEOMETRIC INTERPRETATION

Consider the scalar equation:

$$\mathbf{x}^T\mathbf{S}\mathbf{x} = 1$$

where $\mathbf{S}$ is symmetric. This is the equation of a quadratic surface centered on the origin, in n-space. The normal to the surface is the vector $\partial(\mathbf{x}^T\mathbf{S}\mathbf{x})/\partial\mathbf{x} = \mathbf{S}\mathbf{x}$, and $\mathbf{x}$ is the radius vector, so

$$\mathbf{S}\mathbf{x} = \lambda\mathbf{x}$$

is the problem of finding points where the normal and the radius vector are parallel. These are where the principal axes intersect the surface. At these points, $\mathbf{x}^T\mathbf{S}\mathbf{x} = 1$ too, so $\mathbf{x}^T\lambda\mathbf{x} = 1$ or:

$$\lambda = \frac{1}{\mathbf{x}^T\mathbf{x}}$$

So the eigenvalues are the reciprocals of the squares of the lengths of the principal axes.

The lengths are independent of the coordinate system, so will also be invariant under an arbitrary orthogonal transformation, i.e. one in which $(\textit{distance})^2 = \mathbf{x}^T\mathbf{x}$ is unchanged.

Consider using the eigenvectors of $\mathbf{S}$ to transform the equation for the quadratic surface:

$$\mathbf{x}^T\mathbf{L}\Lambda\mathbf{L}^T\mathbf{x} = 1 \quad \text{or} \quad \mathbf{y}^T\Lambda\mathbf{y} = 1 \quad \text{or} \quad \sum\lambda_i y_i^2 = 1$$

where $\mathbf{y} = \mathbf{L}^T\mathbf{x}$ or $\mathbf{x} = \mathbf{L}\mathbf{y}$. This transforms the surface into its principal axis representation.

SINGULAR VALUE DECOMPOSITION

The standard eigenvalue problem is meaningless for non-square matrices.

A 'shifted' eigenvalue problem associated with an arbitrary non-square matrix $\mathbf{K}$, m rows and n columns can be constructed:

$$\begin{aligned}\mathbf{K}\mathbf{v} &= \lambda\mathbf{u} \\ \mathbf{K}^T\mathbf{u} &= \lambda\mathbf{v}\end{aligned} \tag{1}$$

where $\mathbf{v}$, of length n, and $\mathbf{u}$, of length m, are called the *singular vectors* of $\mathbf{K}$.

This is equivalent to the symmetric problem:

$$\begin{pmatrix}\mathbf{O} & \mathbf{K} \\ \mathbf{K}^T & \mathbf{O}\end{pmatrix}\begin{pmatrix}\mathbf{u} \\ \mathbf{v}\end{pmatrix} = \lambda\begin{pmatrix}\mathbf{u} \\ \mathbf{v}\end{pmatrix}$$

From (1) we can get

$$\mathbf{K}^T\mathbf{K}\mathbf{v} = \lambda\mathbf{K}^T\mathbf{u} = \lambda^2\mathbf{v}$$
$$\mathbf{K}\mathbf{K}^T\mathbf{u} = \lambda\mathbf{K}\mathbf{v} = \lambda^2\mathbf{u}$$

so $\mathbf{u}$ and $\mathbf{v}$ are the eigenvectors of $\mathbf{K}\mathbf{K}^T$ $(m \times m)$ and $\mathbf{K}^T\mathbf{K}$ $(n \times n)$ respectively.

Care is needed in constructing a matrix of singular vectors, because individual $\mathbf{u}$ and $\mathbf{v}$ vectors correspond to each other, yet there are potentially different numbers of $\mathbf{v}$ and $\mathbf{u}$ vectors. If the rank of $\mathbf{K}$ is p, then there will be p non-zero singular values, and both $\mathbf{K}\mathbf{K}^T$ and $\mathbf{K}^T\mathbf{K}$ will have p non-zero eigenvalues. The surplus eigenvectors will have zero eigenvalues, and can be discarded and we can write:

$$\begin{pmatrix}\mathbf{O} & \mathbf{K} \\ \mathbf{K}^T & \mathbf{O}\end{pmatrix}\begin{pmatrix}\mathbf{U} \\ \mathbf{V}\end{pmatrix} = \begin{pmatrix}\mathbf{U} \\ \mathbf{V}\end{pmatrix}\mathbf{\Lambda}$$

where $\mathbf{\Lambda}$ is $p \times p$, $\mathbf{U}$ is $m \times p$, and $\mathbf{V}$ is $n \times p$. There will be $n + m - p$ more eigenvectors of the composite matrix, all with zero eigenvalue.

EIGENVECTORS - USEFUL RELATIONSHIPS

Asymmetric Matrices	Symmetric Matrices
$\mathbf{AR} = \mathbf{R}\Lambda$	$\mathbf{SL} = \mathbf{L}\Lambda$
$\mathbf{L}^T\mathbf{A} = \Lambda\mathbf{L}$	
$\mathbf{L}^T = \mathbf{R}^{-1}$, $\mathbf{R}^T = \mathbf{L}^{-1}$	$\mathbf{L}^T = \mathbf{L}^{-1}$
$\mathbf{LR}^T = \mathbf{L}^T\mathbf{R} = \mathbf{I}$	$\mathbf{LL}^T = \mathbf{L}^T\mathbf{L} = \mathbf{I}$
$\mathbf{A} = \mathbf{R}\Lambda\mathbf{L}^T = \sum \lambda_i \mathbf{r}_i \mathbf{l}_i^T$	$\mathbf{S} = \mathbf{L}\Lambda\mathbf{L}^T = \sum \lambda_i \mathbf{l}_i \mathbf{l}_i^T$
$\mathbf{A}^T = \mathbf{L}\Lambda\mathbf{R}^T = \sum \lambda_i \mathbf{l}_i \mathbf{r}_i^T$	
$\mathbf{A}^{-1} = \mathbf{R}\Lambda^{-1}\mathbf{L}^T$	$\mathbf{S}^{-1} = \mathbf{L}\Lambda^{-1}\mathbf{L}^T$
$\mathbf{A}^n = \mathbf{R}\Lambda^n\mathbf{L}^T$	$\mathbf{S}^n = \mathbf{L}\Lambda^n\mathbf{L}^T$
$\mathbf{L}^T\mathbf{AR} = \Lambda$	$\mathbf{L}^T\mathbf{SL} = \Lambda$
$\mathbf{L}^T\mathbf{A}^n\mathbf{R} = \Lambda^n$	$\mathbf{L}^T\mathbf{S}^n\mathbf{L} = \Lambda^n$
$\mathbf{L}^T\mathbf{A}^{-1}\mathbf{R} = \Lambda^{-1}$	$\mathbf{L}^T\mathbf{S}^{-1}\mathbf{L} = \Lambda^{-1}$
$\lvert\mathbf{A}\rvert = \prod_i \lambda_i$	$\lvert\mathbf{A}\rvert = \prod_i \lambda_i$

SINGULAR VECTORS - USEFUL RELATIONSHIPS

$$
\begin{aligned}
\mathbf{KV} &= \mathbf{U}\Lambda \\
\mathbf{K}^T\mathbf{U} &= \mathbf{V}\Lambda \\
\mathbf{U}^T\mathbf{KV} &= \mathbf{V}^T\mathbf{K}^T\mathbf{U} = \boldsymbol{\lambda} \\
\mathbf{K} &= \mathbf{U}\Lambda\mathbf{V}^T \\
\mathbf{K}^T &= \mathbf{V}\Lambda\mathbf{U}^T \\
\mathbf{V}^T\mathbf{V} &= \mathbf{U}^T\mathbf{U} = \mathbf{I}_p \\
\mathbf{KK}^T\mathbf{U} &= \mathbf{U}\Lambda^2 \\
\mathbf{K}^T\mathbf{KV} &= \mathbf{V}\Lambda^2
\end{aligned}
$$

SQUARE ROOTS OF MATRICES

The square root of an arbitrary matrix is defined as $\mathbf{A}^{\frac{1}{2}}$ where

$$\mathbf{A}^{\frac{1}{2}}\mathbf{A}^{\frac{1}{2}} = \mathbf{A}$$

Using $\mathbf{A}^n = \mathbf{R}\Lambda^n\mathbf{L}^T$ for $n = 1/2$:

$$\mathbf{A}^{\frac{1}{2}} = \mathbf{R}\Lambda^{\frac{1}{2}}\mathbf{L}^T$$

This square root of a matrix is not unique, because the diagonal elements of $\Lambda^{\frac{1}{2}}$ in $\mathbf{R}\Lambda^{\frac{1}{2}}\mathbf{L}^T$ can have either sign, leading to 2^n possibilities.

We only use square roots of symmetric covariance matrices. In this case $\mathbf{S}^{\frac{1}{2}} = \mathbf{L}\Lambda^{\frac{1}{2}}\mathbf{L}^T$ is symmetric.

Symmetric matrices can also have non-symmetric roots satisfying $\mathbf{S} = (\mathbf{S}^{\frac{1}{2}})^T\mathbf{S}^{\frac{1}{2}}$, of which the Cholesky decomposition:

$$\mathbf{S} = \mathbf{T}^T\mathbf{T}$$

where $\mathbf{T}$ is upper triangular, is the most useful.

There are an infinite number of non-symmetric square roots: if $\mathbf{S}^{\frac{1}{2}}$ is a square root, then clearly so is $\mathbf{XS}^{\frac{1}{2}}$ where $\mathbf{X}$ is any orthonormal matrix.

The inverse symmetric square root is $\mathbf{S}^{-\frac{1}{2}} = \mathbf{L}\Lambda^{-\frac{1}{2}}\mathbf{L}^T$, and the inverse Cholesky decomposition is $\mathbf{S}^{-1} = \mathbf{T}^{-1}\mathbf{T}^{-T}$. The inverse square root $\mathbf{T}^{-1}$ is triangular, and its numerical effect is implemented efficiently by back substitution.