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

301/1332-6

"Application for Retrieval Theory to the EOS Microwave Limb Sounder (MLS)"

N. LIVESEY
Jet Propulsion Laboratory
CALTECH
Pasadena CA
USA

lecture presented by:

H. PUMPHREY
Institute for Meteorology
The University of Edinburgh
Scotland

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

Application of retrieval theory to the EOS Microwave Limb Sounder (MLS)

Inverse methods in atmospheric science.

Nathaniel Livesey

Jet Propulsion Laboratory

California Institute of Technology

[livesey@mls.jpl.nasa.gov]

1–12 October 2001 (v1.5)

Outline of talk.

- ❑ Description of the EOS MLS instrument.
- ❑ Overview of retrieval problem, construction of vectors etc.
- ❑ Issues related to retrieval 'phasing' and constrained quantity error propagation.
- ❑ A 'two dimensional' approach to the retrieval problem.
- ❑ A discussion of forward model issues.
- ❑ Issues related to 'noisy' products (if time).
- ❑ Implementation of the algorithms in software.
- ❑ Some initial results from the algorithms.
- ❑ Plans for future development.

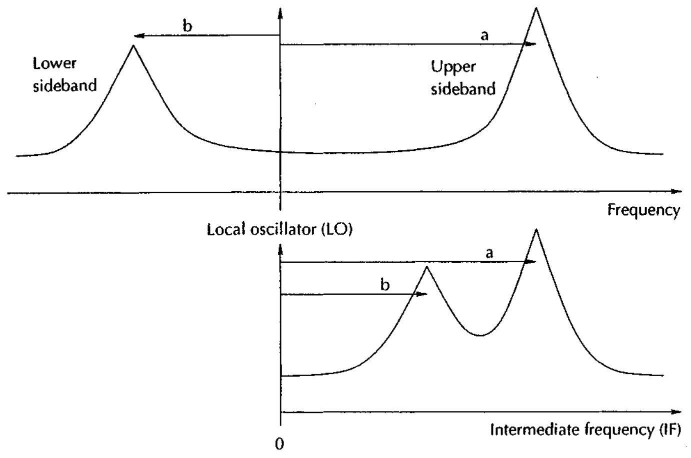
The EOS MLS instrument.

- ❑ The EOS MLS instrument is a follow on to the successful MLS instrument flown on UARS, launched in 1991.
- ❑ It is designed to measure aspects of the chemistry and dynamics of the stratosphere and upper troposphere.
- ❑ It will fly on the EOS Aura platform, along with the HIRDLS, OMI and TES instruments.
- ❑ The Aura launch is currently scheduled for July 2003.
- ❑ EOS MLS is designed, built and calibrated by the Jet Propulsion Laboratory.
- ❑ The instrument uses the *Microwave Heterodyne* technique (described later) to measure thermal emission from the earth's limb.

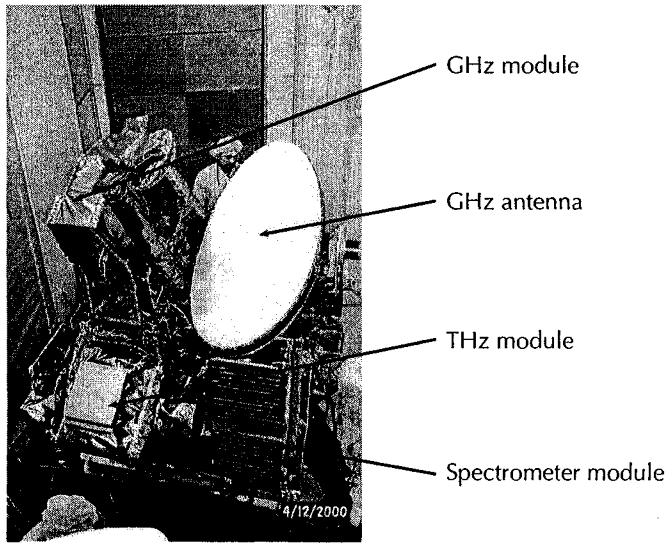
Some aspects of microwave limb sounding.

- ❑ One major difference between microwave sounding and some other techniques (e.g. infrared radiometry) is that spectral lines are easily resolved.
- ❑ In the stratosphere and troposphere spectral line shapes are dominated by pressure broadening effects, as opposed to Doppler broadening.
- ❑ Lines broaden by ~3 MHz per hPa.
- ❑ Most information comes from observations of *spectral contrast* (the shape of the line), as opposed to absolute radiance (*baseline* effects).
- ❑ So, for example if we can resolve lines with widths up to 1 GHz, we can sound down to 300 hPa.
- ❑ Once the lines get broader than we can resolve issues of absolute radiance come into play, and errors get larger.
- ❑ Radiances described in terms of a *brightness temperature*, in Kelvins.
- ❑ Observed radiance can thus be a rough indication of the temperature in regions where the radiances saturate (black-out).

The microwave heterodyne technique.



The EOS MLS instrument.

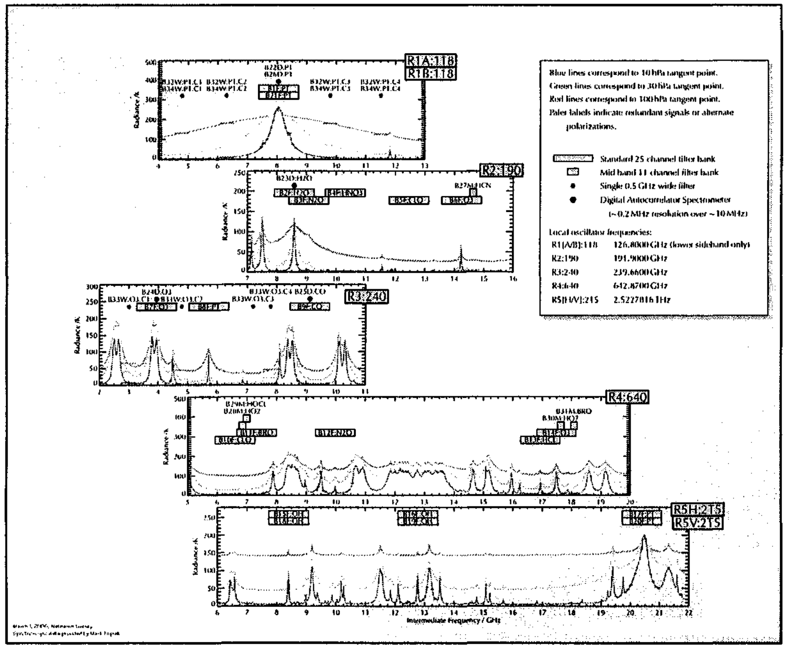


EOS MLS Receivers.

- EOS MLS contains 7 microwave receivers, measuring radiation in the regions around
  - **118 GHz** (two redundant receivers) measuring O_2 emission for temperature/pressure.
  - **190 GHz** measuring some stratospheric species and upper tropospheric water.
  - **240 GHz** mainly intended for measurements of CO and upper tropospheric O_3.
  - **640 GHz** the main stratospheric chemistry 'workhorse'.
  - **2.5 THz** (two receivers) for measuring stratospheric OH.
- The 118 GHz receiver is a single sideband receiver, all the others are double sideband.

EOS MLS Spectrometers.

- The signals from each receiver are sent to several spectrometers.
- These are centered on various spectral lines of interest.
- EOS MLS has four different types of spectrometers.
  - **FB25** A filter bank containing 25 discrete channels, with narrow (6 MHz) channels near the line center, broadening to 96 MHz in the at 575 MHz away from line center.
  - **MB11** A filter bank containing only 11 channels, corresponding to the center 11 in the FB25 spectrometers.
  - **DACS** Digital auto-correlating spectrometers, giving ~0.2 MHz resolution over 10 MHz
  - **WF4** These consist of four 500 MHz wide channels judiciously placed within the IF spectrum.



The EOS MLS Orbit and scan.

- ❑ Aura will fly in a 98° inclined sun-synchronous orbit, performing ~14.5 orbits per day.
- ❑ EOS MLS observes the limb directly in front of the spacecraft.
  - ⇨ This has interesting and very useful implications to be discussed later in this presentation.
- ❑ The GHz and THz telescopes make a complete vertical scan of the atmosphere every ~24 seconds.
- ❑ The scan pattern is designed that the observed latitudes are essentially unchanged from orbit to orbit.
- ❑ All parts of the globe are measured twice per day.
  - ⇨ Once on an ascending orbital node, once descending.
- ❑ There are plans to have Aqua, Aura, Cloudsat and other platforms fly in formation.
- ❑ This will allow for near simultaneous observations.

Overview of the retrieval process.

- ❑ Like most instruments the EOS MLS data are divided into 'Levels'.

  Level 0 Raw data from the instrument.

  Level 1 Calibrated radiances.

  Level 2 Geophysical data along the orbit / tangent point track.

  Level 3 Geophysical data mapped onto some regular lat/lon grid.
- ❑ The rest of this talk concerns the Level 2 processing.
- ❑ The method applied is the standard *optimal estimation* approach.
- ❑ While we have some new approaches to the problem, the fundamental mathematical approach is standard.

The retrieval equation.

- ❑ We choose to represent the state of the atmosphere by the vector $\mathbf{x}$.
- ❑ Radiances from various bands within the instrument are gathered into measurement vectors $\mathbf{y}_i$, with (typically diagonal) covariances $\mathbf{S}_i$.
- ❑ The standard *Gauss Newton* iteration is given by

$$\mathbf{x}^{(r+1)} = \mathbf{x}^{(r)} + \left[\sum_i \mathbf{K}_i^T \mathbf{S}_i^{-1} \mathbf{K}_i\right]^{-1} \sum \mathbf{K}_i^T \mathbf{S}_i^{-1} \left[\mathbf{y}_i - \mathbf{f}_i(\mathbf{x}^{(r)})\right]$$

  where $\mathbf{f}_i$ is the *forward model*, and

$$\mathbf{K}_i = \frac{\partial \mathbf{f}_i(\mathbf{x})}{\partial \mathbf{x}}$$

  is the matrix of *weighting functions* or *Jacobians*.
- ❑ The solution covariance is given by

$$\mathbf{S}_x = \left[\sum_i \mathbf{K}_i^T \mathbf{S}_i^{-1} \mathbf{K}_i\right]^{-1}$$

Virtual measurements and *a priori*.

- As it stands, the matrix inversion above is typically not possible.
  - ⇨ There are aspects of the state vector for which the measurements have yielded no information.
- We introduce *virtual measurements*, in the form of *a priori* estimates of the state vector values.
- It will later prove useful to make these separate measurements, rather than one of the $\mathbf{y}_i$ vectors.
  - ⇨ The vector $\mathbf{a}$ is a *virtual measurement* of $\mathbf{x}$ with covariance $\mathbf{S_a}$.
- The iteration thus becomes

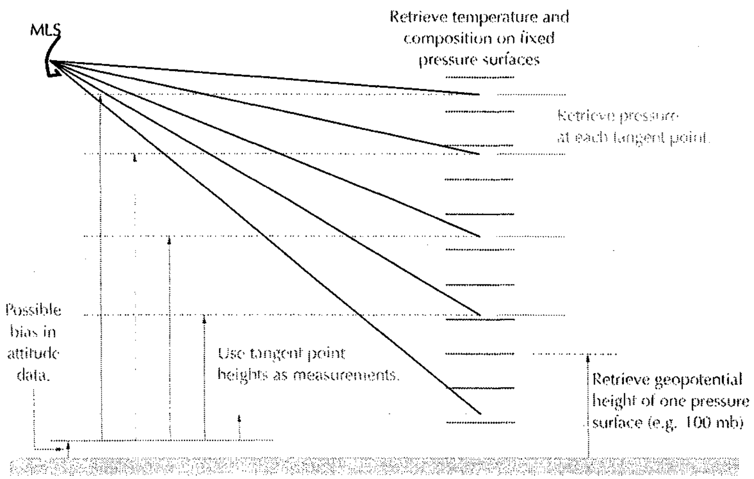
$$\mathbf{x}^{(r+1)} = \mathbf{x}^{(r)} + \left[\mathbf{S}_a^{-1} + \sum_i \mathbf{K}_i^T \mathbf{S}_i^{-1} \mathbf{K}_i\right]^{-1} \left\{\mathbf{S}_a^{-1}\left[\mathbf{a} - \mathbf{x}^{(r)}\right] + \sum_i \mathbf{K}_i^T \mathbf{S}_i^{-1}\left[\mathbf{y}_i - \mathbf{f}_i(\mathbf{x}^{(r)})\right]\right\}$$

- For elements of $\mathbf{x}$ that don't need *a priori* information, the corresponding rows and columns of $\mathbf{S_a}$ are set to zero.

Construction of the MLS state vector.

- The most important components of the MLS state vector are the temperature and composition of the atmosphere, as a function of pressure.
  - ⇨ These are the *standard products* from MLS.
- Note that we use a pressure grid as opposed to a height grid.
- In addition, we include the geopotential height of a single pressure surface (e.g. 100 hPa).
  - ⇨ To include a complete profile of geopotential height is unnecessary, as the temperature profile already conveys this information.
- However, the forward model requires more information in order to model radiances.
- The most important information is the atmospheric pressure at each tangent point.
- Also, the angular offset between the various MLS radiometers fields of view is required.

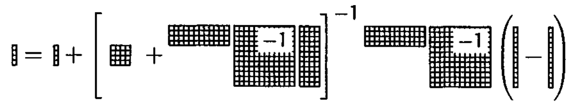
Construction of the MLS state vector (cont.)



Sizing the MLS retrieval task.

- Consider the retrieval of a single MLS ozone profile from one scan's worth of MLS 205 GHz ozone observations.
- Retrieve ozone at 12 surfaces per decade from 1000 hPa to 0.1 hPa.
  - ⇨ Length of state vector $\mathbf{x}$, $n = 48$
- We use 120 minor frames worth of radiances from 25 channels.
  - ⇨ Length of measurement vector $\mathbf{y}$, $m = 3000$
- The linear form of the optimal estimation equation gives:

$$\mathbf{x} = \mathbf{a} + \left[\mathbf{S}_a^{-1} + \mathbf{K}^T \quad \mathbf{S}_y^{-1}\mathbf{K}\right]^{-1} \mathbf{K}^T \quad \mathbf{S}_y^{-1}(\mathbf{y} - \mathbf{f})$$



Sizing the MLS retrieval task (cont.)

- a is the *a priori* state vector with $n \times n$ covariance matrix S_a.
- f is the forward model measurement vector, (the predicted radiances corresponding to the *a priori* state.
- S_y is the $m \times m$ measurement covariance matrix.
- K is the $m \times n$ matrix of *weighting functions*:

$$K = \frac{\partial y}{\partial x}.$$

- The most time consuming aspects of the calculation are the inversion of S_y. (m^3) and the computation of $K^T S_y^{-1} K$ ($n^2 m + m^2 n$).
- However, if S_y is diagonal, we are left with only $n^2 m + nm + m$.

The measurement covariance matrix S_y.

- Clearly, having S_y as a diagonal matrix would be a real advantage.
- What does is mean if the S_y matrix is diagonal?
  - ⇨ The 'errors' in the radiances are all uncorrelated.
  - ⇨ If the radiance in channel 0 is 'too high' that doesn't mean that channel 1 is any more or less likely to also be too high.
- What causes non diagonal covariance matrices?
  - ⇨ Certain instrumental effects such as gain variation. These can be taken into account by retrieving quantities such as 'baseline'.
  - ⇨ The use of constrained quantity error propagation in multi-phase retrieval processes.

Constrained quantity error propagation.

- Many previous retrieval algorithms (e.g. UARS MLS, ISAMS) implemented a multi-phase approach
  - ⇨ e.g. a retrieval of temperature/pressure first, from O_2 radiances,
  - ⇨ followed by retrievals of various species, using the temperature and pressure data from the earlier phase in the forward model.
- The previously retrieved quantities c (e.g. temperature and pressure) are *constrained* in the later phases.
- However, our knowledge of these quantities is not perfect, they have a covariance S_c, estimated by the early phase.
- This uncertainty needs to be propagated through the forward model into an additional radiance uncertainty.
- We should modify our S_y matrices in the later phases according to $S_y \rightarrow S_y + K_c S_c K_c^T$, where K_c describes the sensitivity of the radiances to these constrained quantities $K_c = \partial y / \partial c$.

Constrained quantity error propagation (cont.)

- This will make S_y non-diagonal.
  - ⇨ If the temperature we are using is too high, then all the forward model radiances will be too large 'in concert'.
- One could choose to ignore the non-diagonal elements of the new S_y matrix.
- However, previous experience has shown that this can be a poor approximation.
  - ⇨ The retrieval algorithm puts less 'trust' in the measurements than they deserve.
- Is there an alternative approach which avoids the costly calculations, and yet retains accuracy?

A new approach to multi-phase retrievals.

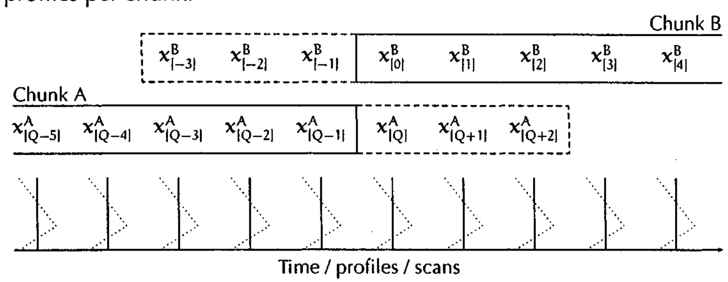
- Avoiding constrained quantity error propagation makes $\mathbf{S_y}$ diagonal.
- Instead we retrieve everything at once from every channel.
- However, as the real MLS system is non linear, we may have to perform several iterations on this 'big' system.
- To improve the efficiency, we re-introduce phasing with a new twist.
- In the early phases we retrieve the most non-linear quantities (those needing several iterations to converge) from appropriate bands.
  - ⇨ For example, retrieve temperature and pressure from O_2 radiances.
- Once a good estimate is obtained for these quantities, add more linear items to the state and measurement vectors, *while still retrieving the earlier quantities.*
- This larger system will need fewer iterations to converge, as the non-linear quantities are already close to the solution.
- Think of the earlier phases getting 'initial guesses' for the final phase.

A 'two dimensional' approach.

- Unlike UARS MLS, the EOS MLS instrument looks forward from the spacecraft.
- This means that all the observations are within the orbital plane.
  - ⇨ Although the rotation of the earth has an impact on this.
- Each limb ray is affected by the state of the atmosphere over a ~1000 km path length.
- This corresponds to several adjacent retrieved profiles.
- Note that the scan can be arranged to stack the tangent points in a vertical profile.
  - ⇨ As you scan up, the tangent point gets closer to you.
  - ⇨ If this happens at the same rate as the spacecraft moves forward, the tangent point locus is vertical.
- In the EOS MLS case, we scan slowly through the troposphere and lower stratosphere, then speed up in the upper stratosphere and mesosphere.
- How can we devise an algorithm that takes most advantage of this geometry?

Dividing the data processing into chunks.

- We process the data in chunks of ~1/8–1/4 orbit in length.
- The measurement vectors $\mathbf{y}_i$ contain information from M scans.
- The state vector $\mathbf{x}$ describes N vertical profiles.
- Typically we choose $N = M$ but this is not a requirement.
- The chunks to overlap slightly, to account for 'edge' effects.
- We have q profiles of overlap (e.g. 3) giving $Q = N - 2q$ non overlapped profiles per chunk.



Weighting functions for this problem.

- The efficiency gain in this case comes from noting that the weighting function matrices $\mathbf{K}_i$ are very sparse.
- For example, the values of temperature for profile 1 have no effect on the radiances for scan 10.
- This gives a block structure for $\mathbf{K}_i$ similar to.

$$\mathbf{K}_i = \frac{\partial \mathbf{y}_i}{\partial \mathbf{x}} = \text{Scans}\downarrow \overset{\text{Profiles}\longrightarrow}{\begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ 0 & \times & \times & \times & 0 & 0 \\ 0 & 0 & \times & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}}$$

- Where 'profile' is taken to mean the complete state (temperature and composition profiles) for one location.

$K_i^T S_i^{-1} K_i$, the 'Normal equations'.

- Given the form for K_i shown above, and assuming S_i is diagonal (more on this later), the matrix $K_i^T S_i^{-1} K_i$, needed in the retrieval, is of the form:

$$K_i^T S_i^{-1} K_i = \begin{array}{c} \text{Profiles} \longrightarrow \\ \text{Profiles} \downarrow \left[\begin{array}{cccccc} \times & \times & \times & 0 & 0 & 0 \\ \times & \times & \times & \times & 0 & 0 \\ \times & \times & \times & \times & \times & 0 \\ 0 & \times & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \end{array}\right] \end{array}.$$

- We know that we can ignore any block products involving absent (completely 0) blocks in K_i
- This matrix is sometimes known as the matrix of *normal equations.*
- Given a matrix K_i with block bandwidth p, $K_i^T S_i^{-1} K_i$ will have a block bandwidth $2p$.
- Forming this matrix product is the most CPU intensive part of the inverse model calculation, as $m \gg n$

Computational effort in the retrieval

- The formation of the normal equations scales according to Npn^2m.
- The key point is that this operation scales as N, not N^2. Therefore.
- *It takes the same time to retrieve one 200 profile chunk as to retrieve two 100 profile chunks!*
- The limitation on the size of N becomes the memory capacity of the computer.
- Solving this matirx with a 'sparsity aware' Cholesky decomposition scales as N^2pn^3.
- Thus, the matrix solver will typically be faster than the $K^T S_y^{-1} K$ by a factor of $\sim m/Nn$.
- Of course, in real situations we have more complex state and measurement vectors, introducing more sparsity.
  - ⇨ For example, very few MLS bands have sensitivity to minor species such as ClO.

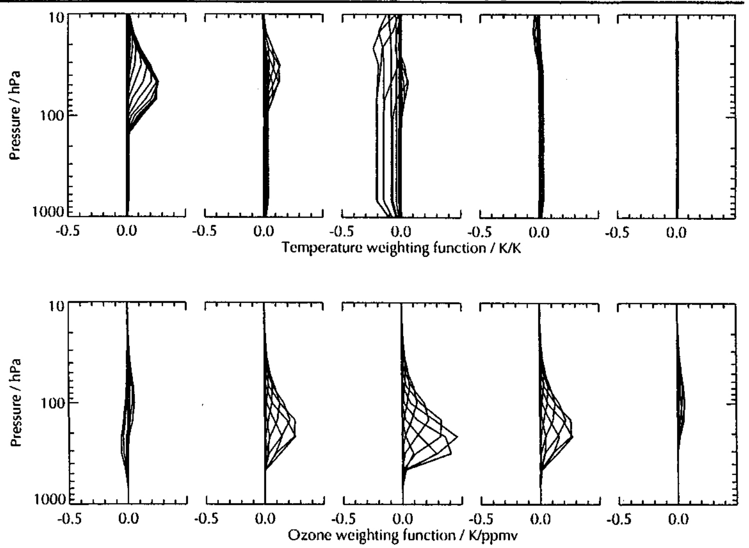
A prototype retrieval

- A 'proof of concept' prototype has been designed.
- Forward model and retrieval both written in IDL.
- Forward model contains all 2D radiative transfer methods required.
- Retrieval linearises this forward model to the form

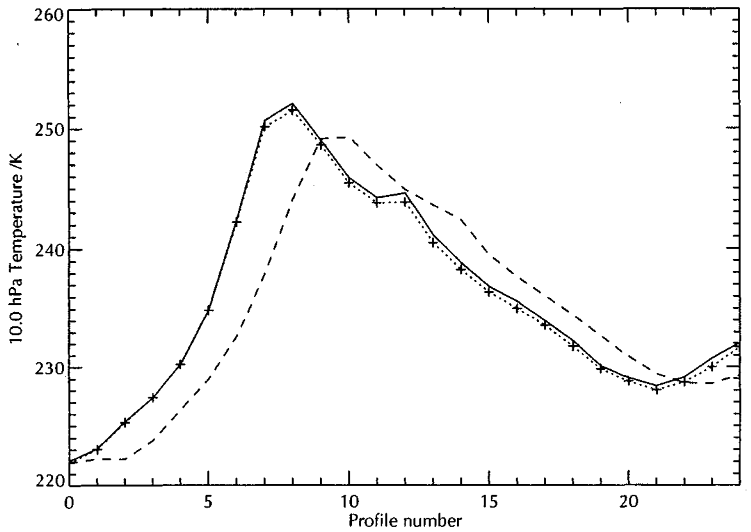
$$y^* = y^* + K^* (x - x^*)$$

- 25 profiles of UARS MLS data have been taken as 'truth'.
  - ⇨ Note that the horizontal resolution of UARS MLS is ~500 km.
  - ⇨ For EOS MLS it is ~150 km.
  - ⇨ Thus the gradients in this model atmosphere are probably a little severe.
- A retrieval of Temperature, tangent pressure and ozone was performed.
- Radiances from `R1:118.B1F:PT` and `R2:190.B6F:O3` were used.

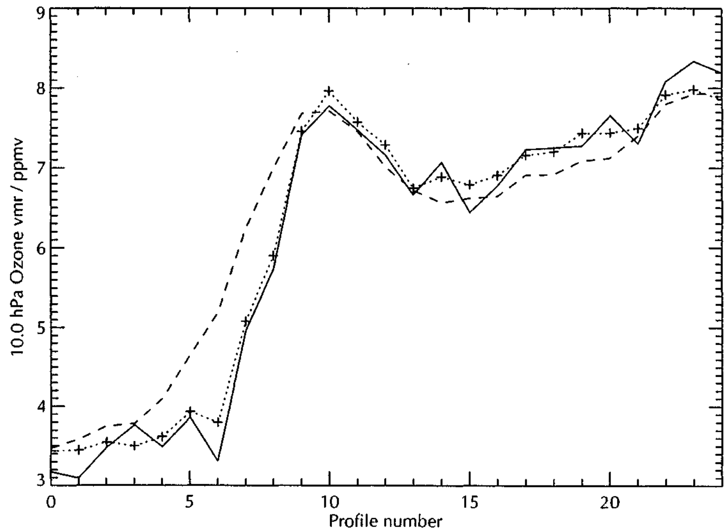
Weighting functions for the prototype



Results from a prototype — Temperature



Results from a prototype — Ozone



The 'full' MLS forward model.

- ❑ The forward model is similar to that developed for UARS MLS, extended to two dimensions.
- ❑ It is a microwave line by line model, using pressure as the independent vertical coordinate.
- ❑ Radiances are computed for a set of fixed tangent pressures.
  - ⇨ A different fixed frequency mesh is used for each tangent pressure.
  - ⇨ Typically covering one or more 25 channel filter bank.
  - ⇨ The radiances at these frequencies are then convolved with the individual MLS channel responses.
- ❑ These profiles are then convolved with the MLS field of view (FOV) response, and interpolated to the required tangent presssure.
  - ⇨ This interpolation yields the derivative of radiance wrt. tangent pressure virtually for free.

The 'full' MLS forward model (cont.)

- ❑ The state vector profiles are taken to represent tie points in a 'linear spline' interpolation.
- ❑ Note that the state vector contains vmr not log vmr (except for H_2O in the troposphere).
- ❑ The forward model accounts for the linear variations in temperature and composition accross it's integration layers.
- ❑ A Gauss-Legendre quadrature (3–6 point) integration scheme is applied.
- ❑ Radiance derivatives with respect to composition, temperature, and some spectroscopic parameters can be computed analytically.
- ❑ The mixing ratio derivatives are cheap to compute.
- ❑ Temperature derivatives are somewhat more expensive.
  - ⇨ It transpires that the most significant terms are those due to the effects of the FOV.
  - ⇨ Changes in temperature affect the shape of the FOV when viewed in pressure space.

The linear MLS forward model

- For many of the MLS spectral bands, the system is highly linear.
- Given this, it is possible to construct a simple linear forward model as

$$y = y^\star + K^\star [x - x^\star] \quad (1)$$

- $y^\star$ and $K^\star$ are pre-tabulated radiances and derivatives for state $x^\star$.
- In UARS MLS the $x^\star$ linearisation states were divided up according to latitude band and month.
- For EOS we intend a more dynamic scheme, tabulating standard cases (e.g. inside polar vortex, tropical spring,...), and choosing the most appropriate given the value of x.
- The radiances and derivatives are tabulated for fixed tangent pressures.
- These are then interpolated to the tangent pressures in the state vector. The interpolation yielding the derivative with respect to tangent pressure as an added bonus.

Forward model implementation

- We plan to use the linear forward model for most channels.
- As each channel becomes optically thick, the linear forward model becomes a poorer approximation.
- Optical depth increases with decreasing tangent height and increasing proximity to the line center.
- We use each channel down to the tangent heights where it is too optically thick, and then ignore it.
- Information will still be obtained from the channels further from the line center at these heights.
- For the 'wing' channels (furthest from line center), we'll have to use the full non-linear forward model if we wish to get useful information.
- We may still take the derivatives from the linear model for speed.

Approach for 'noisy' products.

- Many of the MLS data products will be 'noisy'
- Products such as BrO with very low concentrations and/or signal strengths.
- Some form of averaging (e.g. weekly zonal mean, monthly map) will be needed to yield Level 2 data with useful signal to noise.
-

Obvious approaches for 'noisy' products

- There are two obvious approaches for dealing with 'noisy' products.
- **Average in measurement space:**
  - ⇨ Average together many radiance observations and perform retrievals on the averaged radiances.
  - ⇨ This works, provided the signal of interest is not affected by a varying contaminant who's effect on the radiances is non-linear.
- **Average in state space:**
  - ⇨ Perform single profile retrievals as with the 'standard' products and average the results appropriately.
  - ⇨ This solves the problem with non linear contaminants.
  - ⇨ However, the *a priori* information is introduced in *each* profile retrieval and will thus strongly bias an average.
  - ⇨ But one can't make the individual profile *a priori* too loose without risking instability in the individual retrievals.

A new approach: Average in 'information' space.

- Construct a state vector which is the desired result.
  - ⇨ For example a monthly zonal mean BrO field.
- Then we consider each relevant set of radiance observations as an individual measure of this quantity.
- We use the previously retrieved quantities such as temperature, pressure, ozone etc. as constrained quantities in the forward model for these retrievals.
  - ⇨ This deals with the issues of having non linear contaminants.
  - ⇨ Error propagation for these quantities is tbd.
  - ⇨ We include the *a priori* in the retrieval *once only*.
- Mathematically this comes down to:

$$x = \left[S_a^{-1} + \sum_i K_i^T S_i^{-1} K_i \right]^{-1} \sum_i K_i^T S_i^{-1} \left[y_i - f_i\left(a\right) \right],$$

where the summation is over all the relevant sets of radiance profiles.

Making it even easier

- It is possible to improve on the efficiency of this algorithm further,
- if the routine (i.e. daily) processing outputs forward model radiances or the radiance residuals.
- These are equivalent to the $y - f$ terms in the retrieval equation.
- As the signal of interest is small, weighting functions can be pre-tabulated.
- The daily processing can even retrieve its own estimates of the 'noisy' species.
  - ⇨ These can be useful in spotting 'freak' events.
- Again, as the signals are small, a linear correction can be made for the amount inferred by the daily processing.
- The UARS MLS instrument has a filter bank centered on a spectral line from H_2O_2.
- This will be an ideal test case for the algorithm.

Implementation of the algorithms in software.

- We want to process one days worth of data in less than one day.
  - ⇨ We're aiming for 6 hours, to allow for parallel reprocessing streams.
- The software is written at JPL, and will be run at a Raytheon facility in Pasadena under contract to JPL.
- The data will then be sent to the Goddard DAAC (EOS data repository) for archive and distribution.
- We're anticipating running the code on Beowulf style cluster.
- Having divide the data into the chunks described earlier, we have the nodes work on them independently.
- If the nodes themselves have multiple (e.g. 2) processors, either:
  - ⇨ give them each two chunks (may take too much memory).
  - ⇨ write the chunk processing code in a parallel manner (a little harder to implement).
- We currently have a 64 node, 128 processor IBM linux cluster.

Implemetation in software (cont.)

- Software written in Fortran 95.
  - ⇨ F95 has many powerful features that Fortran 77 really lacks.
  - ⇨ Higher level languages such as IDL and Matlab lack the speed, also cost too much in cluster environment.
  - ⇨ C does not handle arrays as well, and is typically harder to optimize.
- We implemented a somewhat object orientated approach.
  - ⇨ Defining 'vector' and 'matrix' types and overloading some appropriate operators.
- The code is driven by the 'Level 2 Configuration File' (l2cf).
  - ⇨ This is essentially a language devised to describe retrievals, forward model calculations and related activities.

Vectors, quantities and matrices in the software

- At the heart of the software is the concept of a 'quantity'.
  - ⇨ This is a collection of data for a chunk.
  - ⇨ For example, a set of temperature profiles, or ozone profiles, tangent pressures or radiances.
  - ⇨ Simpler items such as instrument calibration parameters, isotope ratios etc. are also stored as 'quantities'.
- Quantities are collected together to make 'vectors'.
  - ⇨ For example the state vector, and measurement vectors.
  - ⇨ For efficiency we divorce the vector 'template' (quantity geolocation information etc.) from the 'value'.
- We also define the concept of a matrix.
  - ⇨ These have attached vectors describing their rows and columns.
  - ⇨ Typically describe derivative of one vector with respect to another (e.g. weighting functions), or the covariance of a single vector.

Storage and manipulation of matrices

- The quantities in a vector a themeselves divided into 'instances'.
  - ⇨ These are horizontal realizations of the quantity.
  - ⇨ Individual temperature profiles, separate radiance scans etc.
- The matrices are divided up into blocks by quantity and index.
  - ⇨ The derivative of band 1, scan 10 radiance with respect to temperature profile 11 etc.
- The blocks in the matrices can be of four types:

  Absent All zeros, nothing stored.

  Full A 'full' block.

  Banded A block with a few clustered non-zero elements per column.

  Sparse A block with a few non-zero elements in random locations.
- The banded and sparse representations are only typically worthwhile for blocks with <~20% non-zero elements.
- The matrix algebra in the code effitiently deals with all of these.

A flexible Level 2 program.

- The Level 2 software is very flexible:
  - ⇨ Can read and write to/from both Level 1 and Level 2.
  - ⇨ Manipulates gridded data from climatological sources.
  - ⇨ Can perform stand alone forward model calculations, in addition to retrievals.
- This means that the one program can do:
  - ⇨ Standard retrievals.
  - ⇨ Simulations of radiance fields.
  - ⇨ Pre-computation of the tables for the linear forward model.
  - ⇨ Or even all three together!
- This is much easier than writing three separate programs, each using slightly different I/O and initialisation code.
- Clearly, the configuration needs to be described in a clear manner.

The Level 2 Configuration File (l2cf)

- The l2cf is in many ways a computer language for describing retrievals.
- Can define quantities, vectors, matrices etc. in a very similar manner to the definitions of types and variables in other languages.
- The syntax is somewhat reminiscent of IDL.

```
; Define a vertical coordinate system in -log10(pressure/hPa),
; with 25 surfaces at 12 per decade starting at 1000mb, followed
; by 24 surfaces at 6 per decade.
standardSurfaces: vGrid, coordinate=Zeta, type=Logarithmic, $
  start=1000mb, formula=[25:12, 24:6]

; Place profiles where GHz tangent point height first crosses
; 15km each scan.
standardProfiles: hGrid, type=height, height=15km, module=GHz

; Define a template for temperature, GHz tangent pressure,
; ozone and band 6 radiances.
temperature: Quantity, vGrid=standardSurfaces, hGrid=standardProfiles, $
  type=temperature
ptahGHz: Quantity, type=ptan, module=GHz
ozone: Quantity, type=vmr, molecule=O3, vGrid=standardSurfaces, $
  hGrid=standardProfiles
band6: Quantity, type=radiance, signal='R2:190.B6:O3'
```

The Level 2 Configuration File (cont.)

```
; Define templates for state and measurement vectors
stateTemplate: vectorTemplate, quantities=[temperature, ozone, ptanGHz]
measTemplate: vectorTemplate, quantities=[band6]

; Define various vectors
x: vector, template=stateTemplate ; State vector
a: vector, template=stateTemplate ; A priori state vector
y: vector, template=measTemplate  ; Measurement vector
yNoise: vector, template=measTemplate  ; Measurement noise

; Set up appropriate default states
Fill, quantity=x.temperature, method=gridded, source=aprioriTemp
   ; aprioriTemp is a gridded field read by earlier lines
   ; in the l2cf
.
.
Fill, quantity=y.band6, method=l1b ; Fill radiances from L1 file
.
.
```

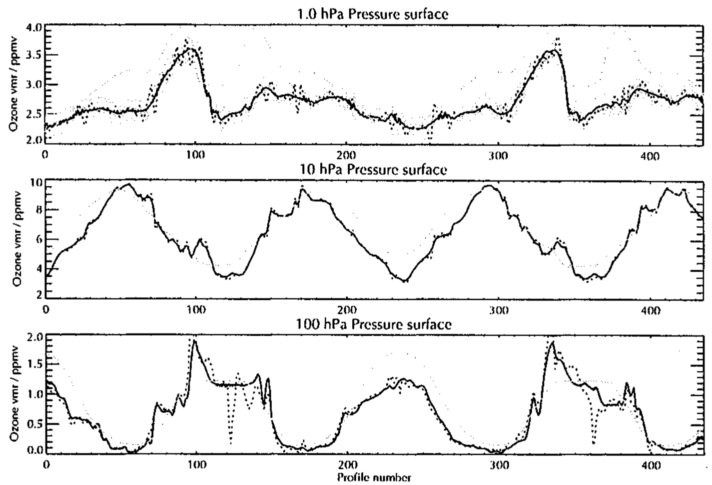

The Level 2 Configuration File (cont.)

```
; Perform a very simple retrieval (definition of some terms omitted)
Retrieve, state=x, measurements=y, measurementSD=yNoise, $
      forwardModel=retFwm, $
      covariance=myCovariance, apriori=a, columnScale=norm, $
      maxF=2, maxJ=1, lambda=0.0,outputSD=sdOut
; One defines forward model configurations (e.g. retFwm) earlier
; in the l2cf.  A retrieval can use more than one forward model.
.
.
; Later parts of the l2cf deal with joining together data from the
; chunks and outputting them in the appropriate files.
```


Production code.

- The production code is still under development and testing.
- Essential functionality in place.
- What remains are many 'small but vital' features that will be needed.
- Also need to ensure code is 'bomb proof'.
  - ⇨ Must be able to cope with missing data, broken radiometers etc.
- Hoping to add an 'on-line diagnostic' capability.
  - ⇨ A separate IDL task that can communicate with the fortran code during testing display results etc.
- Note that the software is flexible enough that it can easily be modified to process data from UARS MLS.
  - ⇨ No code changes required, just some changes to the l2cf and new calibration files.
- Some results from a simple retrieval, similar to the 'prototype' follow.

Results from production code.



Red – truth, Black – retrieved (dotted 1D), Cyan – *a priori*

Summary.

- MLS is a passive microwave instrument designed to measure the chemistry and dynamics of earth's atmosphere from 5–80 km.
- The retrieval algorithms use the standard *optimal estimation* approach.
- One new aspect is a two dimensional 'tomographic' approach to the problem.
- Avoid error propagation problems by doing simultaneous retrievals.
- Implemented in a very flexible software setup.
- Work proceeding well.