Inverse Theory Practical Exercise: Atmospheric profile retrievals.

Hugh C. Pumphrey

October 15, 2008

1 Introduction

This exercise is intended to provide some experience in applying the MAP formula to atmospheric profile retrievals. The example used is a similar one to the one used in the lectures, i.e. a nadir sounding microwave instrument used to measure the temperature of the atmosphere. The exercise consists of the following steps:

- Read in and make plots of several temperature profiles.
- Calculate the radiances which the instrument would measure if the atmosphere had each of these profiles.
- Attempt to retrieve the original profiles from the radiances you calculated using:
  - The MAP formula
  - The Twomey-Tikhonov formula

You should produce a report describing your results. Do not duplicate large amounts of material out of the notes or these instructions. Do give answers to all of the questions asked in these instructions. I expect all figures to be clear and to have readable axis labels. To this end, please make sure that you use sufficiently large text for your labels. Don't include your figures as screenshots or other low-resolution bitmaps — include your postscript figures directly in the word processor document[1].

2 Profiles

There are four temperature profiles in the file `/home/hcp/wrk/invth_pract/tprofs.dat`. This file contains five columns of numbers: the first column is the vertical co-ordinate which is $\zeta = -\log_{10}(\text{Pressure/mb})$ – one of these units is about 16 km. The remaining four columns are the four temperatures in Kelvin. The first line of the file contains two integers: the number of temperature profiles and the number of elements per profile, n.

Use R (or some other plotting package) to read in the file and plot the four profiles.

3 Applying the forward model

The exercise assumes an entirely linear forward model *i.e.* to obtain a set of radiances from the test profile just implies multiplication by a matrix:

$$\mathbf{y} = \mathbf{K}\mathbf{x}. \qquad (1)$$

In this case $\mathbf{x}$ is the temperature profile and $\mathbf{y}$ is the set of measurements that the instrument would make. (Note that the units of $\mathbf{y}$ are K, the same as for $\mathbf{x}$.) The $m \times n$ matrix $\mathbf{K}$ is in the file `/home/hcp/wrk/invth_pract/nadout.dat`. (Instructions for reading this file are given in Section 7.) Plot the rows of $\mathbf{K}$ *i.e.* the influence functions. For which regions of the atmosphere the instrument will provide useful measurements? Why? Look again at the profiles you plotted in section 2. Which of these profiles are we likely to have the most problems measuring with our imaginary instrument, and why?

Write a program that will apply equation 1 to each of the four profiles. Present and examine the results, in particular, look at the temperature at $\zeta = -0.5$ and the radiance you calculate in the first channel (whose influence function peaks at height $\zeta = -0.5$).

[1] If you don't know why this is important, I have put a document that explains it on the course web site at `http://www.geos.ed.ac.uk/abs/Postgraduate/inverse_theory`

4 MAP formula

In this section you will attempt to retrieve one of the profiles from the radiances you just calculated by applying the MAP formula. Your program will need to perform several steps:

- Read in the $m \times n$ matrix $\mathbf{K}$ from file `/home/hcp/wrk/invth_pract/nadout.dat`
- Calculate the vector $\mathbf{y}$ as you did in section 3.
- Add a bit of random noise to $\mathbf{y}$.
- Set up a $m \times m$ diagonal matrix $\mathbf{S}_y$ with the diagonal elements set to the variance of the noise you added to $\mathbf{y}$.
- Set up an *a priori* profile $\mathbf{x}_a$. Make a sensible choice and explain why you chose it.
- Set up a $n \times n$ *a priori* covariance matrix $\mathbf{S}_a$. You can make this a diagonal matrix – choose sensible values for the diagonal elements and remember to state what you chose and why. If you choose a non-diagonal form for $\mathbf{S}_a$, explain all the decisions you made while choosing it.
- Calculate the $m \times n$ contribution function matrix $\mathbf{D} = \mathbf{S}_a\mathbf{K}^T(\mathbf{K}\mathbf{S}_a\mathbf{K}^T + \mathbf{S}_y)^{-1}$
- Calculate the retrieved profile $\hat{\mathbf{x}} = \mathbf{x}_a + \mathbf{D}(\mathbf{y} - \mathbf{K}\mathbf{x}_a)$

Try this with the standard deviation of the noise set to 0.000001 K, 0.1 K and 1 K. Also, try it (just for the 0.1 K noise) with all four profiles, using the same $\mathbf{x}_a$ in all four cases. Comment on why it works better for some than for others. When plotting your results, put the true profile and the *a priori* profile onto the plot along with your retrieved profile.

5 Twomey-Tikhonov formula

In this section you will attempt to retrieve one of the profiles from the radiances you just calculated by applying the Twomey-Tikhonov formula. Your program will need to perform several steps:

- Read in the $m \times n$ matrix $\mathbf{K}$ from file `/home/hcp/wrk/invth_pract/nadout.dat`
- Calculate the vector $\mathbf{y}$ as you did in section 3.
- Add a bit of random noise to $\mathbf{y}$. You only need to try one noise level this time: use 0.1 K.
- Set up a $n \times n$ constraint matrix $\mathbf{H}$. Do not use a diagonal matrix as we did in the lectures. Use the matrix

$$\begin{bmatrix} 1 & -2 & 1 & & & & \\ -2 & 5 & -4 & 1 & & & \\ 1 & -4 & 6 & -4 & 1 & & \\ & 1 & -4 & 6 & -4 & 1 & \\ & & 1 & -4 & 6 & -4 & \cdots \\ & & & 1 & -4 & 6 & \cdots \\ & & & & \vdots & \vdots & \ddots \end{bmatrix}$$

  which makes the retrieved profile as smooth as possible.
- Set up an *a priori* profile $\mathbf{x}_a$. With the above choice of $\mathbf{H}$, the choice of $\mathbf{x}_a$ is not that important except at the top and bottom of the profile. You can re-use your $\mathbf{x}_a$ from section 4.
- Choose a value for the parameter γ.
- Calculate the $m \times n$ contribution function matrix $\mathbf{D} = (\gamma^{-1}\mathbf{H} + \mathbf{K}^T\mathbf{K})^{-1}\mathbf{K}^T$
- Calculate the retrieved profile $\hat{\mathbf{x}} = \mathbf{x}_a + \mathbf{D}(\mathbf{y} - \mathbf{K}\mathbf{x}_a)$
- Do you think your profile is over-smoothed or under-smoothed? Re-run your program several times with different values of γ and decide what is the best value. You will need to try at least three or four widely spaced values: All of the range between 10^7 and 10^{-5} is worth exploring. You only need try the Twomey-Tikhonov formula on one of the four profiles.

6 For real, this time

You have now tested your retrieval skills on a case where you knew the right answer. Now, try applying your code to some cases where you do *not* know the answer. The file `newrads.dat` contains four sets of radiances from four new profiles to which you do not have access, just as you wouldn't with real measurements. The measurement error on these radiances is 0.1 K.

Choose either the MAP formula (with your preferred values of $\mathbf{S}_a$ and $\mathbf{x}_a$) or the Twomey-Tikhonov formula (with your preferred value of γ) and apply it to these four profiles. Explain why you chose the formula that you chose.

7 Computing Appendix

You may use any computing language you like to do the calculations. You are only likely to get help from your instructor if you program in R or IDL. Both of these provide enough graphics facilities for you to plot your output. (If you choose to use a language without graphics facilities then gnuplot would be quite sufficient for plotting your results.)

Try not to waste too much time struggling to get the data files read in to your programming language of choice. It is more important that you get experience getting your programs to apply the formulae you have learned in the lectures, and in interpreting the results. To this end, I provide some code snippets here to help you get the data read in. This code may be found in the same directory as the data files (`/home/hcp/wrk/invth_pract`) so you don't have to type it in. Or you can find an electronic version of this document on the course web page and cut-and-paste the code from that.

7.1 The test profiles:

Here is a short R program (`readprofs.R`) which reads in the temperature profiles.

```
## Read in temp profiles and plot.
hdr <- scan(file="/home/hcp/wrk/invth_pract/tprofs.dat",nlines=1)
nprofs<-hdr[1]
nlevs<-hdr[2]
zspacing<-hdr[3]

tmpdat <- scan(file="/home/hcp/wrk/invth_pract/tprofs.dat",nlines=nlevs,skip=1)
dim(tmpdat) <- c(nprofs+1,nlevs)
zeta <- tmpdat[1,]
tprofs <- tmpdat[2:(nprofs+1),]
```

7.2 K matrix file.

Here is a bit of R code (`readks.R`) to read in the K matrix file.

```
## Program to read in Ks etc. First, we need to read the header and the
## temperature profile used to calculate the Ks
## You don't need this, but you could use it as an a-priori if you liked.

ksfile <- "/home/hcp/wrk/invth_pract/nadout.dat"
hdr <- scan(file=ksfile,nlines=1)
nfreq<-hdr[1]
nlevs<-hdr[2]
zspacing<-hdr[3]
tmpdat <- scan(file=ksfile,nlines=nlevs,skip=1)
dim(tmpdat) <- c(3,nlevs)
zeta <- tmpdat[1,]
z <- tmpdat[2,]
tem.l <- tmpdat[3,]


## read in the actual Ks int kfns. The file also contains the transmissivities,
## these are read into tau but you don't need them.
```

```
y.l <- 0
kfns <- matrix(0,nrow=nfreq,ncol=nlevs)
tau <- kfns
for(j in 1:nfreq){
  y.l[j] <- scan(file=ksfile,nlines=1,skip=j*(nlevs+1))
  tmpdat <- scan(file=ksfile,nlines=nlevs,skip=j*(nlevs+1)+1)
  dim(tmpdat) <- c(2,nlevs)
  kfns[j,] <- tmpdat[2,]
  tau[j,] <- tmpdat[1,]
}
```

7.3 Measurements with unknown profiles

The following code snippet (`readrads.R`) will read in the set of radiances corresponding to some profiles that you do not know.

```
## Read in the "measured" radiances
hdr <- scan(file="/home/hcp/wrk/invth_pract/newrads.dat",nlines=1)
nprofs.rad<-hdr[1]
nfreq.rad<-hdr[2]
## nprofs.rad is the number of sets of radiances and hence the
## number of profiles you are to try to estimate.
## nfreq.rad should be the number of channels and will be the same as
## in the other files.

y.meas <- scan(file="/home/hcp/wrk/invth_pract/newrads.dat",
               nlines=nfreq.rad,skip=1)
dim(y.meas) <- c(nprofs.rad,nfreq.rad)
## y.meas is now an array containing the radiances for which you need to
## estimate the profiles.
```

7.4 Multiplying and Inverting Matrices, random noise

Remember that R has the `%*%` operator to multiply matrices and the function `t()` to take the transpose of a matrix. You may also need to invert a matrix and/or to solve some simultaneous equations. The R function that you need for this is called `solve()`. You will need to generate some random numbers to simulate experimental noise. The R function which you need for this is `rnorm()`.

7.5 Using IDL

If you are already an IDL user, you can do the project in IDL. There are IDL versions of the above code snippets in the directory `/home/hcp/wrk/invth_pract`