EVLA Memo 143: Improving the frequency resolution of the default atmospheric opacity model

J. Marvil
NRAO
April 6, 2010

ABSTRACT

In consideration of the wider bandwidths and improved frequency accessibility of the EVLA, we seek to improve the accuracy of the default opacity model. Herein, we present a method to estimate the zenith atmospheric opacity at any EVLA frequency, by extrapolating from a given 22 GHz K-band opacity. The extrapolation is based on the Atmospheric Transmission of Microwaves (ATM) model[1], using parameters tuned to the EVLA site. In this memo, we document the development of this method and provide a detailed description of its implementation.

1. Purpose

Studies of atmospheric transmission show that the opacity can change significantly across a single band, especially within K and Q. At present, the default atmospheric corrections preformed with AIPS tasks FILLM and INDXR only provide a single opacity per band, which is most accurate at the band's default observing frequency. Because we expect many EVLA observations to occur away from these default frequencies, we desire to improve the accuracy of the default model by enhancing its frequency resolution.

2. Method

Between 1 and 50 GHz, the atmospheric opacity is primarily due to oxygen (the stable, 'dry' term) and water vapor (the variable, 'wet' term). The quantity of atmospheric water vapor can be conveniently expressed as precipitable water vapor (PWV). We use the 22 GHz zenith opacity, τ_{22}, to estimate PWV, and from this PWV we estimate the opacity at each 250 MHz interval from 1 to 50 GHz using a table of polynomial coefficients. In Section 2.1 we discuss the relationship between τ_{22} and PWV, and in Section 2.2 we describe the conversion of PWV to opacity at an arbitrary frequency.

---

[1] http://damir.iem.csic.es/PARDO/atm.html

2.1. PWV vs. τ_{22}

We begin by simulating atmospheres using the CASA interface to the ATM model, with parameters given in Table 1 below. We vary an additional parameter, the relative humidity, to cover a large range of PWV in our model atmospheres. For each of these atmospheres, we record the τ_{22} and PWV quantities reported by ATM. We fit a 1st order polynomial in τ_{22} to describe PWV, given in Equation 1.

Table 1: EVLA Site Parameters

Altitude	2124 m
H_2O scale height	2 km
Ground temperature	270 K
Ground pressure	790 mbar
Temperature lapse rate	-5.6 K/km

$$\mathrm{PWV}\,(\mathrm{mm}) = -1.71 + 136.47 * \tau_{22} \tag{1}$$

Equation 1 is used to convert a given (input) 22 GHz zenith opacity to an estimate of the atmospheric water vapor quantity. Note that this conversion is approximately consistent with the expression given in Butler (2010), although it is important to use Equation 1 in the method presented here to ensure self-consistancy (*i.e.* that the final 22 GHz zenith opacity matches the input). This input opacity is intended to be supplied from the K-band seasonal model, estimated from the dew point measurement of the EVLA weather station, or the average of the two (as is the default for the current AIPS calibration routines– see Butler (2010) for details of the current default model). However, the method presented here is compatible with any available estimate of the 22 GHz zenith opacity, such as that derived from an EVLA tipping procedure, weather satellite, GPS, etc.

2.2. τ vs. PWV

We now present a method to use the PWV value found above to estimate the zenith opacity at any frequency between 1 and 50 GHz. Using the same model atmospheres discussed above, we query ATM for the zenith opacity at every 250 MHz interval from 1 to 50 GHz. At each interval, we fit a 1st order polynomial in PWV to describe τ_ν, the zenith opacity at frequency ν. We then generate a table of polynomial coefficients, given in Table 3 and Appendix 2, which are structured as in Table 2 below:

Table 2: Table Format

ν [GHz]	A_ν	B_ν
1.00	$A_{1.00}$	$B_{1.00}$
1.25	$A_{1.25}$	$B_{1.25}$
1.50	$A_{1.50}$	$B_{1.50}$
...	...	...

Using these coefficients and the PWV value found with Equation 1, the opacity at frequency ν, τ_ν, can be estimated using the form of Equation 2.

$$\tau_\nu = 10^{-3} * (A_\nu + B_\nu * \mathrm{PWV}) \tag{2}$$

Note that opacities at frequencies between these 250 MHz intervals can be estimated *via* linear interpolation. By using this combined method, the zenith opacity at 22 GHz can be extrapolated to any frequency between 1 and 50 Ghz. This is demonstrated in Figure 1, using input 22 GHz opacities of 0.1 and 0.2.

3. Appendices

A single CASA script, provided in Appendix 1, will print to the console the coefficients of Equation 1, and will save to file the table of polynomial coefficients used in Equation 2. The same site parameters (set at the header of the script) are used to derive the coefficients for both equations, to ensure that the frequency-dependent opacity at 22 GHz matches the input opacity. In Appendix 2, we give the same table of coefficients as in Table 3, but in a machine-readable format. In Appendix 3, we provide a python script to demonstrate the full method. This script (which produced Figure 1) extrapolates from a given zenith opacity to estimate opacities across the full 1-50 GHz range.

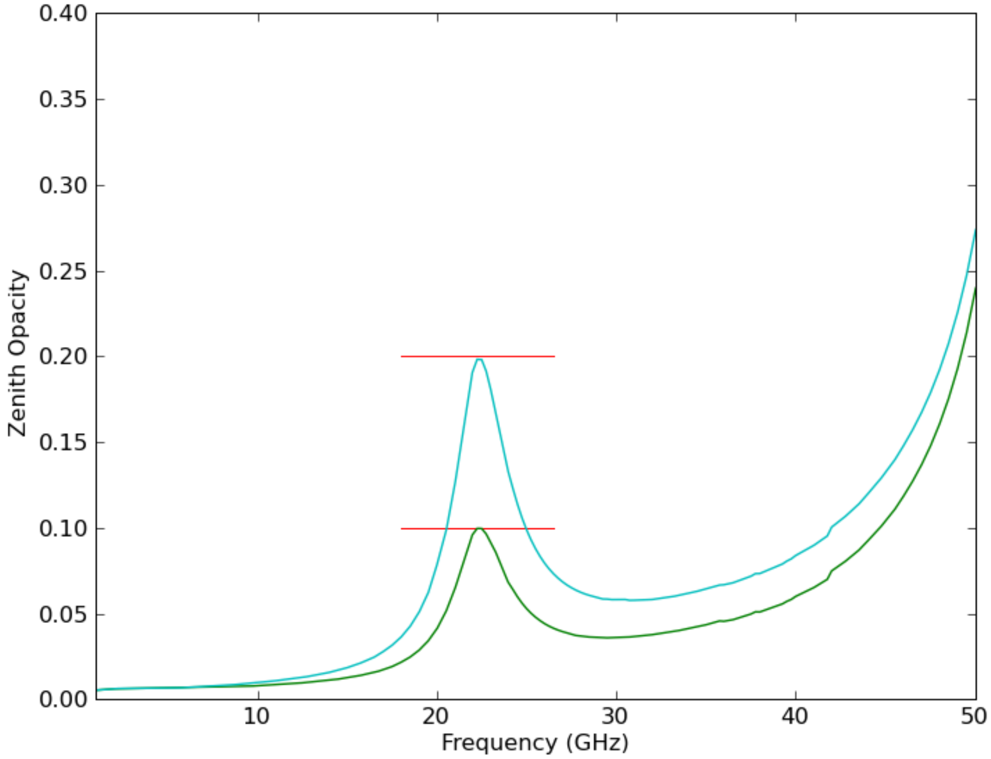

Fig. 1.— K-band zenith opacities of 0.1 and 0.2 as would presently be applied by AIPS (red), and the extrapolations based on the method presented in this memo (cyan, green). This figure was generated using the demonstration python code in Appendix 3.

REFERENCES

Butler, 2010, VLA Test Memo 232: Atmospheric Opacity at the VLA.

This preprint was prepared with the AAS LaTeX macros v5.2.

Table 3: Polynomial Coefficients

ν [GHz]	A_ν	B_ν	ν [GHz]	A_ν	B_ν	ν [GHz]	A_ν	B_ν	ν [GHz]	A_ν	B_ν	ν [GHz]	A_ν	B_ν
1.0	5.27	0.0	11.0	6.84	0.17	21.0	10.78	4.79	31.0	18.42	1.62	41.0	43.67	1.91
1.25	5.46	0.0	11.25	6.86	0.19	21.25	11.2	5.44	31.25	18.77	1.61	41.25	44.98	1.93
1.5	5.58	0.0	11.5	6.91	0.2	21.5	11.67	6.12	31.5	19.13	1.6	41.5	46.35	1.94
1.75	5.65	0.0	11.75	6.95	0.21	21.75	12.13	6.78	31.75	19.5	1.6	41.75	47.78	1.96
2.0	5.71	0.0	12.0	7.0	0.22	22.0	12.52	7.33	32.0	19.87	1.59	42.0	52.35	1.98
2.25	5.75	0.01	12.25	7.06	0.23	22.25	12.78	7.64	32.25	20.26	1.59	42.25	53.99	1.99
2.5	5.78	0.01	12.5	7.11	0.25	22.5	12.87	7.63	32.5	20.66	1.59	42.5	55.73	2.01
2.75	5.81	0.01	12.75	7.16	0.26	22.75	12.82	7.35	32.75	21.08	1.58	42.75	57.63	2.03
3.0	5.83	0.01	13.0	7.22	0.28	23.0	12.67	6.92	33.0	21.5	1.58	43.0	59.46	2.05
3.25	5.86	0.01	13.25	7.28	0.29	23.25	12.49	6.43	33.25	21.94	1.59	43.25	61.42	2.06
3.5	5.88	0.01	13.5	7.34	0.31	23.5	12.34	5.91	33.5	22.39	1.59	43.5	63.58	2.08
3.75	5.9	0.02	13.75	7.4	0.33	23.75	12.26	5.42	33.75	22.85	1.59	43.75	65.85	2.1
4.0	5.92	0.02	14.0	7.46	0.35	24.0	12.15	4.96	34.0	23.26	1.59	44.0	68.61	2.12
4.25	5.94	0.02	14.25	7.52	0.37	24.25	12.13	4.55	34.25	23.75	1.6	44.25	71.06	2.14
4.5	5.96	0.02	14.5	7.59	0.4	24.5	12.17	4.18	34.5	24.25	1.6	44.5	73.65	2.16
4.75	5.98	0.03	14.75	7.66	0.42	24.75	12.25	3.85	34.75	24.78	1.61	44.75	76.42	2.18
5.0	6.0	0.03	15.0	7.73	0.45	25.0	12.36	3.56	35.0	25.32	1.62	45.0	79.3	2.2
5.25	6.03	0.03	15.25	7.79	0.48	25.25	12.5	3.31	35.25	25.88	1.62	45.25	82.33	2.22
5.5	6.05	0.04	15.5	7.85	0.52	25.5	12.69	3.09	35.5	26.45	1.63	45.5	85.57	2.24
5.75	6.07	0.04	15.75	7.92	0.55	25.75	12.86	2.9	35.75	27.06	1.64	45.75	89.01	2.26
6.0	6.1	0.04	16.0	8.0	0.59	26.0	13.01	2.74	36.0	26.71	1.65	46.0	92.74	2.28
6.25	6.12	0.05	16.25	8.08	0.64	26.25	13.21	2.59	36.25	27.07	1.66	46.25	96.64	2.3
6.5	6.15	0.05	16.5	8.14	0.69	26.5	13.42	2.46	36.5	27.67	1.67	46.5	100.8	2.32
6.75	6.18	0.06	16.75	8.22	0.74	26.75	13.64	2.35	36.75	28.32	1.68	46.75	105.23	2.34
7.0	6.21	0.06	17.0	8.3	0.81	27.0	13.86	2.25	37.0	28.99	1.69	47.0	109.98	2.36
7.25	6.24	0.07	17.25	8.38	0.88	27.25	14.1	2.16	37.25	29.69	1.7	47.25	115.07	2.38
7.5	6.27	0.07	17.5	8.46	0.96	27.5	14.34	2.09	37.5	30.43	1.71	47.5	120.54	2.4
7.75	6.3	0.08	17.75	8.54	1.05	27.75	14.59	2.02	37.75	31.36	1.72	47.75	126.42	2.43
8.0	6.33	0.08	18.0	8.63	1.15	28.0	14.84	1.96	38.0	31.15	1.74	48.0	132.77	2.45
8.25	6.36	0.09	18.25	8.72	1.26	28.25	15.1	1.91	38.25	31.94	1.75	48.25	139.64	2.47
8.5	6.4	0.09	18.5	8.82	1.4	28.5	15.37	1.86	38.5	32.76	1.76	48.5	147.1	2.49
8.75	6.44	0.1	18.75	8.92	1.56	28.75	15.65	1.82	38.75	33.61	1.78	48.75	155.21	2.52
9.0	6.47	0.11	19.0	9.03	1.74	29.0	16.03	1.78	39.0	34.49	1.79	49.0	164.07	2.54
9.25	6.52	0.11	19.25	9.15	1.95	29.25	16.22	1.75	39.25	35.41	1.8	49.25	173.79	2.56
9.5	6.55	0.12	19.5	9.28	2.19	29.5	16.51	1.72	39.5	36.38	1.82	49.5	184.51	2.58
9.75	6.59	0.13	19.75	9.44	2.48	29.75	16.82	1.7	39.75	37.36	1.83	49.75	196.42	2.61
10.0	6.64	0.14	20.0	9.62	2.82	30.0	17.35	1.68	40.0	38.97	1.85	50.0	209.87	2.63
10.25	6.73	0.15	20.25	9.83	3.21	30.25	17.57	1.66	40.25	40.07	1.86	-	-	-
10.5	6.72	0.16	20.5	10.09	3.67	30.5	17.88	1.64	40.5	41.22	1.88	-	-	-
10.75	6.77	0.16	20.75	10.4	4.19	30.75	18.1	1.63	40.75	42.42	1.89	-	-	-

APPENDIX 1: CASA script to run the ATM model

```
#initialize ATM model
import casac
attool = casac.homefinder.find_home_by_name('atmosphereHome')
at = attool.create()

#define site parameters
tmp = quantity(270.0,'K')
pre = quantity(790.0,'mbar')
alt = quantity(2124,'m')
h0  = quantity(2.0,'km')
wvl = quantity(-5.6,'K/km')
mxA = quantity(48,'km')
dpr = quantity(10.0,'mbar')
dpm = 1.2
att = 1
nb=1

#define spectral window
fC=quantity(25.,'GHz')
fW=quantity(50.,'GHz')
fR=quantity(0.25,'GHz')
hum = 20.0
myatm = at.initAtmProfile(alt, tmp, pre, mxA, hum, wvl, dpr, dpm, h0, att)
at.initSpectralWindow(nb,fC,fW,fR)
sg = at.getSpectralWindow()
mysg = sg.value

#compute the opacity at each frequency for each of nstep PWV values
nstep = 20
pwv = []
opac = pl.zeros((len(mysg),nstep))

for i in range(nstep):
    hum = 20.0*(i+1)
    myatm = at.initAtmProfile(alt, tmp, pre, mxA, hum, wvl, dpr, dpm, h0, att)
    w = at.getGroundWH2O()
    pwv.append(w.value)
    at.initSpectralWindow(nb,fC,fW,fR)
    sg = at.getSpectralWindow()
    mysg = sg.value
```

```
    sdry=at.getDryOpacitySpec()
    swet=at.getWetOpacitySpec()
    sd=sdry['dryOpacity']
    sw=swet['wetOpacity'].value
    stot = pl.array(sd)+pl.array(sw)
    opac[:,i]=stot

#fit a 1st order polynomial, opacity vs. PWV, at each frequency
pwv_coef = pl.zeros((len(mysg),2))
for i in range(len(mysg)):
    a=pl.polyfit(pwv,opac[i,:], 1)
    pwv_coef[i,:]=a
    if mysg[i] == 22.*1e9:
        k_coef2 = (1.0/pwv_coef[i,0])
        k_coef1 = -(pwv_coef[i,1]/pwv_coef[i,0])

#print inverse fit
print 'PWV(mm) = '+str(round(k_coef1,2))+ ' + ' + str(round(k_coef2,2))+ '*Tau_22'

#write table of polynomial coefficients
fname = 'Kband_PWV_coef.txt'
output=open(fname, 'w')
for i in range(len(mysg)):
    if mysg[i] < 1e9: continue   #do not write frequencies below 1 GHz
    list0=str(round(mysg[i]/1e9,2)) +' '+ str(round(pwv_coef[i,1]/1e-3,3)) \
    +' '+ str(round(pwv_coef[i,0]/1e-3,3)) + ' \n'
    output.writelines(list0)
output.close()
print 'Table of polynomial coefficients saved to: ' + fname
```

APPENDIX 2: Machine-Readable Table of Polynomial Coefficients

```
1.0 5.272 0.0
1.25 5.464 0.0
1.5 5.579 0.001
1.75 5.654 0.002
2.0 5.707 0.003
2.25 5.748 0.005
2.5 5.78 0.006
2.75 5.808 0.008
3.0 5.832 0.009
3.25 5.855 0.011
3.5 5.876 0.013
3.75 5.897 0.016
4.0 5.918 0.018
4.25 5.939 0.021
4.5 5.96 0.023
4.75 5.981 0.026
5.0 6.003 0.03
5.25 6.026 0.033
5.5 6.049 0.037
5.75 6.073 0.04
6.0 6.098 0.044
6.25 6.123 0.048
6.5 6.15 0.052
6.75 6.178 0.057
7.0 6.206 0.061
7.25 6.235 0.066
7.5 6.266 0.071
7.75 6.297 0.077
8.0 6.33 0.082
8.25 6.364 0.088
8.5 6.398 0.094
8.75 6.435 0.1
9.0 6.473 0.107
9.25 6.522 0.114
9.5 6.55 0.122
9.75 6.59 0.129
10.0 6.635 0.137
10.25 6.726 0.146
10.5 6.722 0.155
10.75 6.766 0.164
11.0 6.835 0.174
11.25 6.864 0.185
11.5 6.905 0.196
11.75 6.954 0.207
12.0 7.003 0.22
12.25 7.055 0.233
12.5 7.109 0.247
12.75 7.164 0.262
```

```
13.0 7.22 0.277
13.25 7.277 0.294
13.5 7.336 0.312
13.75 7.396 0.331
14.0 7.458 0.352
14.25 7.521 0.374
14.5 7.586 0.398
14.75 7.661 0.423
15.0 7.726 0.451
15.25 7.785 0.482
15.5 7.854 0.516
15.75 7.924 0.553
16.0 8.0 0.593
16.25 8.081 0.638
16.5 8.144 0.688
16.75 8.219 0.744
17.0 8.296 0.806
17.25 8.375 0.876
17.5 8.456 0.955
17.75 8.54 1.045
18.0 8.634 1.148
18.25 8.724 1.265
18.5 8.819 1.4
18.75 8.92 1.557
19.0 9.028 1.738
19.25 9.147 1.949
19.5 9.281 2.195
19.75 9.435 2.483
20.0 9.615 2.82
20.25 9.83 3.214
20.5 10.09 3.67
20.75 10.404 4.194
21.0 10.776 4.786
21.25 11.204 5.436
21.5 11.669 6.117
21.75 12.129 6.778
22.0 12.523 7.328
22.25 12.782 7.638
22.5 12.872 7.625
22.75 12.815 7.351
23.0 12.667 6.924
23.25 12.491 6.426
23.5 12.335 5.914
23.75 12.256 5.42
24.0 12.146 4.961
24.25 12.134 4.546
24.5 12.171 4.176
24.75 12.249 3.849
25.0 12.362 3.562
25.25 12.503 3.312
```

```
25.5 12.689 3.093
25.75 12.859 2.902
26.0 13.01 2.735
26.25 13.21 2.589
26.5 13.419 2.461
26.75 13.638 2.349
27.0 13.864 2.25
27.25 14.097 2.163
27.5 14.342 2.087
27.75 14.586 2.019
28.0 14.838 1.96
28.25 15.098 1.907
28.5 15.367 1.861
28.75 15.653 1.82
29.0 16.03 1.784
29.25 16.215 1.752
29.5 16.506 1.724
29.75 16.82 1.7
30.0 17.345 1.679
30.25 17.567 1.66
30.5 17.879 1.644
30.75 18.096 1.63
31.0 18.423 1.619
31.25 18.769 1.609
31.5 19.127 1.601
31.75 19.495 1.595
32.0 19.871 1.59
32.25 20.262 1.587
32.5 20.663 1.585
32.75 21.078 1.584
33.0 21.501 1.584
33.25 21.939 1.585
33.5 22.39 1.587
33.75 22.853 1.59
34.0 23.256 1.594
34.25 23.747 1.598
34.5 24.254 1.604
34.75 24.777 1.61
35.0 25.317 1.616
35.25 25.875 1.624
35.5 26.451 1.631
35.75 27.058 1.64
36.0 26.713 1.649
36.25 27.065 1.658
36.5 27.674 1.668
36.75 28.319 1.678
37.0 28.991 1.689
37.25 29.692 1.7
37.5 30.431 1.712
37.75 31.355 1.724
```

```
38.0 31.152 1.736
38.25 31.939 1.749
38.5 32.756 1.762
38.75 33.606 1.776
39.0 34.491 1.789
39.25 35.414 1.803
39.5 36.384 1.818
39.75 37.363 1.832
40.0 38.973 1.847
40.25 40.074 1.862
40.5 41.222 1.878
40.75 42.42 1.894
41.0 43.671 1.91
41.25 44.978 1.926
41.5 46.345 1.942
41.75 47.775 1.959
42.0 52.349 1.976
42.25 53.994 1.993
42.5 55.726 2.011
42.75 57.631 2.028
43.0 59.46 2.046
43.25 61.421 2.064
43.5 63.577 2.083
43.75 65.854 2.101
44.0 68.61 2.12
44.25 71.062 2.139
44.5 73.653 2.158
44.75 76.416 2.178
45.0 79.303 2.197
45.25 82.334 2.217
45.5 85.573 2.237
45.75 89.014 2.257
46.0 92.743 2.277
46.25 96.64 2.298
46.5 100.796 2.319
46.75 105.234 2.34
47.0 109.982 2.361
47.25 115.072 2.382
47.5 120.538 2.404
47.75 126.423 2.426
48.0 132.773 2.448
48.25 139.643 2.47
48.5 147.095 2.492
48.75 155.207 2.515
49.0 164.067 2.538
49.25 173.788 2.561
49.5 184.509 2.584
49.75 196.415 2.608
50.0 209.874 2.632
```

APPENDIX 3: Demonstration Python Script

```
from pylab import *

#input K-band opacity
tau_k = 0.1
plot([18, 26.5],[tau_k, tau_k],'r-')
tau_k2 = 0.2
plot([18, 26.5],[tau_k2, tau_k2],'r-')

#convert K-band opacity to PWV (Equation 1)
pwv = -1.71 + 136.47*tau_k
pwv2= -1.71 + 136.47*tau_k2

#read table of polynomial coefficients
myfile='Kband_PWV_coef.txt'
freqs = []
coef0 = []
coef1 = []
for line in open(myfile,'r'):
    l1=line.split()
    if l1 == empty: continue
    freqs.append(float(l1[0]))
    coef0.append(float(l1[1]))
    coef1.append(float(l1[2]))

#convert PWV to opacity at all frequencies (Equation 2)
tau_allf = (array(coef0)+array(coef1)*pwv) * 1e-3
tau_allf2 = (array(coef0)+array(coef1)*pwv2) * 1e-3

#make the figure
plot(freqs, tau_allf,'g-')
plot(freqs, tau_allf2,'c-')
xlabel('Frequency (GHz)')
ylabel(' Zenith Opacity')
axis([1,50,0,.4])
show()
```