FINAL REPORT

FOR

NAG2-175

ATMOSPHERE AND WIND MODELING FOR ATC

PRINCIPAL INVESTIGATOR:

Gary L. Slater
Dept. AsE/EM
University of Cincinnati
Cincinnati, OH 45221

November 1990

(NASA-CR-196786) ATMOSPHERIC AND WIND MODELING FOR ATC Final Report (Cincinnati Univ.) 48 p

N95-13725

Unclas

G3/47 0022282

Atmospheric and Wind Modeling for ATC

G.L. Slater

University of Cincinnati

Cincinnati, Ohio

November 1990

1. Atmospheric Modeling

Pressure, temperature and density of the atmosphere vary considerably with altitude, and in addition, exhibit additional spatial and temporal variations. Since aircraft motions and sensors are highly dependent on these atmospheric properties, it is important to understand and to model these variations in the development of an ATC advisory scheme. In this section we review briefly the concept of the model atmosphere, and how this model should be corrected to allow for real variations.

1.1 The Model atmosphere

To predict the gross pressure and density properties of the atmosphere at typical aircraft altitudes, it is sufficient to model the atmosphere as a perfect gas, and as a fluid in equilibrium on a non-rotating, flat earth. Such a fluid satisfies:

(1) Perfect gas law: $p = \rho RT$

(2) The hydrostatic equation: $dp = -\rho g dh$

where p, ρ, T are the pressure, density and temperature respectively, R is the gas constant (Universal gas constant R^* divided by the molecular weight of air) and h is the geometric altitude measured positive upwards.

To determine the atmosphere completely, one additional relation between these variables is required. In a theoretical sense this should be a thermodynamic equation expressing conservation of energy; that is the heat balance between internal energy, and heat flux through radiation, conduction and convection. This last relationship is difficult to express; instead, for the purposes of defining an "average" atmosphere, an average temperature variation with altitude is assumed, and this together with equations (1)-(2) define the model atmosphere.

Various atmospheric models have been proposed and are in use. The most common and universally used is the ICAO (International Civil Aviation Authority) which was introduced in 1952 to cover altitudes up to 20km (approximately 65,500ft). The ICAO atmosphere is similar to, but supercedes, the NACA (National Advisory Committee for Aeronautics) atmosphere developed in the United States and the ICAN (International Commission for Aerial Navigation) atmosphere developed in Europe. Since 1952 this atmosphere has been refined to cover altitudes up to satellite levels but the levels up to 20km are essentially unchanged. (See, for example, the US Standard Atmosphere, 1962[1]) This atmosphere is a statistical average over the year for mid-latitudes in the northern hemisphere.

In the altitude range of interest the temperature is defined by two linear functions which define two atmospheric regions:

a) Troposhere:

$0 < h < 11km \quad T = 288.15 - 0.0065h \quad (\deg K)$

b) Stratosphere:

$11km < h \qquad T = 216.65 \ (\deg K)$

The mean temperature at the earth surface is 288.15 K = 15 C = 59 F = 518.67 R . The slope of the temperature versus altitude curve is referred to as the lapse rate and in the troposphere has the defined value -0.0065 deg C/m (or -0.00367 deg F/ft). The stratosphere is modeled as an isothermal region at a temperature of -56.5 deg C (216.65 K) or -69.7 deg F (389.97 deg R). The dividing line between the troposphere and the stratosphere is referred to as the tropopause and in the model atmosphere is defined at 11km = 36089ft (commonly rounded off to 36,000ft). The pressure at sea level (h=0) is defined in the model atmosphere to be exactly $p_0 = 1.013250$ 10^5newtons/m^2 = 1013.25mbar. This corresponds to 760mm Hg or approximately 29.92 in Hg. In English units the pressure is 2116.22psf.

Using this assumed temperature profile the pressure and density can be easily generated by integration. The analytical expressions are given by:

$$p/p_0 = [\ 1 + \alpha h/T_0]^{-n} \qquad h < 11km$$

$$p/p_{11} = \exp[\ -(h - h_{11})/h_s] \qquad h > 11\ km$$

where p_0 is the assumed pressure at sea level (1013.25 mb)

$n = g/\alpha R = 5.25588$

$h_s = RT/g = 6.3416\text{km} = 20{,}610\text{ft}$ (scale height)

$p_{11} = 226.321$ mb (Calculated pressure at 11 km)

$h_{11} = 11\text{km} = 36089\text{ft}$ (Tropopause altitude)

One additional note is that in the model atmosphere the gravitational constant 'g' is taken to be constant in all equations and is defined as $g = 9.80665\ m/s^2$. In fact, g varies with altitude (as well as with latitude), hence the resultant altitude in the model atmosphere is referred to as 'geopotential altitude' and is not the geometric (or physical) altitude. Since 'g' decreases with altitude, the geopotential altitude will be slightly less than the geometric altitude. The relation between geometric altitude and geopotential altitude is easily derived (See Ref. 1) but for the altitudes covered by commercial air traffic the difference between these two is considered insignificant. (At the tropopause the difference is about 60ft.) In this report no distinction is made between geopotential and geometric altitudes.

1.2 Atmospheric Variations

Obviously there are few 'average' days where the temperature profile exactly matches the profile assumed in the model atmosphere. Also surface pressure is highly variable from the standard value although percent deviations are smaller than those in temperature. Surface temperatures in particular exhibit considerable variation from 15 deg C, but even at higher altitudes temperatures may deviate significantly from the model values. Figure 1, taken from Reference 1, depicts the historically observed statistical variations of atmospheric density and temperature. These fluctuations are large enough to have a significant effect on aerodynamic and thrust forces on an aircraft, hence consideration of an accurate ambient atmospheric is necessary for accurate trajectory prediction.

The effect on aircraft performance is felt several ways. Since an aircraft altimeter is calibrated to translate pressure into an altitude reading based on the model atmosphere (see Section 1.4), the aircraft altimeter will not read the true geometric altitude. Nor will the calibrated airspeed measured in the cockpit, have the standard relationship to true airspeed due to an implicit temperature dependency. Engine performance will be different than expected since the air density and temperature are different from the standard values. Finally, wind errors will be introduced because of the difference between true altitude and the altimeter reading. What this translates to is that uncertainty in the temperature profile can be a major source of error in the prediction algorithms which determine expected landing times based on standard descent procedures.

1.3 Incorporation of Actual Atmospheric Data into CTAS

One of the primary elements of the CTAS (Center/Tracon Advisory System) software being developed by NASA is the trajectory prediction component which takes aircraft entering into the center advisory region and calculates precise trajectories to touchdown. Key to the success of the CTAS system is its ability to project the arrival times at various waypoints so as to ensure smooth uninterrupted flow through the system. To ensure the accuracy of such a system two components are needed: 1) An accurate database of aircraft types, force coefficients and aerodynamic data, and 2) accurate atmospheric models with requisite spacial and temporal fidelity. One component of this atmospheric model is the accurate knowledge of pressure, temperature and density in the atmosphere. Another is the problem of determining accurate wind information which is discussed in the next section.

From the previous section it is obvious that the atmospheric parameters are determined by the surface pressure and the variation of temperature with altitude. This latter requirement causes difficulty since often only surface temperatures are available. Standard correction procedures advocated for use by pilots use the standard lapse rate of the model atmosphere, coupled with surface temperature data, to compute density and true altitude variations from the pressure altitude, or altimeter reading. (See e.g. Ref.2) This may also include either an adjustment of the tropopause height or an adjustment of the stratosphere temperature. Such corrections however may be significantly in error, since surface conditions are quite variable, and do not always reflect upper atmosphere changes. This is especially true for high altitude airports, such as Denver's Stapleton Airport which is above 5000ft. The model atmosphere temperature at this height is 5 deg C (41 deg F) while surface temperatures of 90 deg F are quite common in Summer. Projecting temperature variations at upper altitudes based on this number alone introduces the possibility of significant errors. Even if surface pressure is 'standard', temperature deviations from standard values in the vicinity of the surface can introduce significant differences between the 'pressure altitude' (i.e. the altitude corresponding to a pressure level in the standard atmosphere) and the true altitude at much higher altitudes. Figure 2a shows a temperature profile which deviates from the standard in the altitude range from 0-5000ft, then follows the model atmosphere temperature profile above 5000ft. Assuming the surface pressure is standard the deviation between the pressure altitude and the geometric altitude is shown on Figure 2(b). For this profile peak altitude deviations are about 1500 ft.

If no high altitude temperature data is available, then some apriori modeling procedure is clearly necessary. If however additional temperature data is available, as, for example, from descending or ascending aircraft, then these temperature variations can easily be included into an accurate atmospheric model.

1.4 Implementation of an Atmosphere model for CTAS

In this section we assume that a temperature profile and the surface pressure in the atmosphere are known. What we detail here is how to implement this into the CTAS algorithms. We assume that temperature is known at a series of altitudes (h_i, T_i), i=1,n. Then assuming a linear temperature variation between points, the pressure and temperature profiles are:

$$(T/T_i) = 1 + \alpha_i(h - h_i)/T_i$$

$$p/p_i = (T/T_i)^{-g/\alpha_i R}$$

for $h_i < h < h_{i+1}$ and where

$$\alpha_i = \frac{T_{i+1} - T_i}{h_{i+1} - h_i} \qquad i = 1,\ldots,n\text{-}1$$

At the last point h_n, T_n, or if $\alpha_i = 0$, (that is, if the atmosphere is locally isothermal) the pressure relation above is replaced by the exponential form

$$p/p_i = \exp(\ -g(h - h_i)/RT_i\ )$$

Our procedure then is to sequential compute the temperature and pressure at the interface between successive layers, and to store the atmospheric exponent or the scale height for each layer. This array of numbers is computed by CURR_ATM (See Appendix 1) and is used by routines which specifically compute temperature and pressure as functions of altitude. Density calculation is not explicitly added but should be computed from the perfect gas law (ρ = p/RT).

If temperature data comes from aircraft aloft, then it should be recognized that the aircraft altitude will be a pressure altitude and not the geometric altitude required in the equations above. This can be corrected by using the model atmosphere relations and the lower altitude temperature data to explicitly calculate the difference between geometric and pressure altitudes.

The coding for the current atmosphere routines is done as a script file in Matlab. The Matlab code is 'C-like' however and a conversion to 'C' or some other higher order language is relatively straight-forward. See Appendix 1 for a source listing of the routines available.

1.4.1 Note on Altimeter Settings

An aircraft altimeter is calibrated to read out an altitude above mean sea level based on the model atmosphere. There is an additional setting parameter however which allows the altimeter to deviate from this value. For safety of flight, when operating below 18,000ft, pilots are required to adjust their altimeters to a fictitious pressure broadcast by the local station. This fictitious 'corrected' pressure is such that at the physical ground level the altimeter reads the actual height above sea level for the local point in question. For commercial air traffic this generally corresponds to the airport, from which the aircraft is departing from, or arriving at. This corrected barometric pressure is entered manually by an adjustment screw on the altimeter and displayed in the 'Kollsman window' of the altimeter. In the United States, the Kollsman window is calibrated to inches of mercury. Nominal sea level pressure is 29.92 in Hg and this setting is always used when flying above 18,000ft. In general the atmosphere is not 'standard' and hence for an aircraft at the runway altitude, the altimeter will not read the posted runway altitude without some adjustment . The broadcast Kollsman window setting is simply a fictitious pressure that rotates the altimeter dial so that a correctly calibrated altimeter reads correct runway height when the aircraft is on the runway. The physical interpretation of the Kollsman setting is shown in Figure 3, taken from Ref. 2. It is important to note that for a non-standard atmospheric pressure variation, the altimeter will read the correct geometric altitude only at the runway. At other altitudes the altitude must be corrected to obtain geometric altitude just as in the standard atmosphere case.

2. Wind modeling

Lack of accurate data on current atmospheric winds would be a serious impediment to the accurate predicting of aircraft arrival times at the runway and at various intermediate waypoints. While this would not pose a safety problem for the CTAS system, the lack of accurate winds would result in increased controller advisories, and a loss of aircraft efficiency due to the extra maneuvering required to maintain spacing and arrival rates. Fortunately there are several sources of wind data available which make wind profile generation feasible. Wind data is available from: (1) Meteorological information available in the vicinity of most airports, (2) Enroute information from inertially equipped aircraft, and (3) Wind data from the NOAA Wind Profiler network which is available (or soon will be) for 31 selected sites across the country.

Access to, and usage of, the NOAA Profiler system data is discussed extensively in the next few sections of this report. Data processing requirements for the other data types (items (1) and (2) above) is similar;

only the methods of data collection are different. In the current study, data collection for these data types has not been considered and is a topic for further research.

2.1 Wind data: NOAA Profiler system

The wind profiler system currently being implemented by the National Oceanic and Atmospheric Administration (NOAA) was designed with the primary goal of weather prediction, particularly to help model and predict thunderstorm activity in the central section of the United States. There are currently 31 sites planned. The locations of these sites are shown in Figure 4 and Table 2.1. For background on the Profiler system see Reference 3.

The physical setup for a typical profiler site is depicted on Figure 5. At the Profiler site three doppler radar beams are used. One beam is vertical, the other two are each 15 degrees from the vertical and oriented orthogonal to each other, preferably in the North and East directions, but generally skewed due to potential interference with aircraft and satellite communications. (See Table 2.1). Back scattering of the radar beam from particulates and small scale turbulence in the atmosphere allow determination of the radial wind velocity along the three beams as a function of altitude. The altitude dependency is separated out by sending out radar pulses and processing reflected radar returns as a function of the time delay. These three non-orthogonal velocity components can be transformed to give magnitude and heading of the horizontal component of the wind, which are available as outputs of the Profiler station, as well as a vertical component (which is not available to external users.) For details on the operation of the Profiler, see Ref. 3.

The wave length of the radar determines the altitude resolution of the beam and its maximum altitude capability. Higher frequencies (hence shorter wave lengths) give better resolution but have a lower usable altitude range due to the lack of smaller scale turbulence at higher altitudes. The standard profiler site uses a frequency signal of 404 MHz ($\lambda \approx 70$cm), which can give data, up to about 60,000ft. This system is operated in a high resolution mode and a low resolution mode by varying the pulse width and the averaging interval. In the low resolution mode, winds can only be obtained up to about 35,000ft. There is an additional Profiler site at Denver Stapleton airport that is not part of the Profiler Network. While functioning quite similarly to the Profiler sites, this site is an experimental station and operates at a frequency of 915 MHz. Hence it provides more accurate data over a smaller altitude range, and as discussed later, is more sensitive to environmental conditions.).

At the time of this writing, only the station at Platteville, Colorado is available for data, although several other stations are currently operating in a test mode. The complete complement of 31 sites should be available by the end of 1991.

For use in wind prediction in the ATC problem, some knowledge of the data characteristics of this system are needed to aid in utilization of the data. Several factors can influence the quality of the received data. The primary adverse factor is precipitation. Since falling precipitation can produce a large return signal, any precipitation can seriously affect the quality of the data, particularly due to spatial inhomogeneities in droplet speeds and distribution. This effect is especially true at higher radar frequencies, hence the Denver experimental facility frequently is most often affected by precipitation.

This effect is magnified if local inhomogeneities affect the different beams in different ways. In fact, local inhomogeneities of any type can cause discrepancies since the velocity transformations to compute horizontal velocity assume spatial homogeneity. Due to the beam deflection angles this means that at high altitudes, beam locations are several miles apart. The velocity calculation effectively requires differencing to remove the vertical wind component. Any spatial variation in vertical winds between the different beams will be translated into erroneous horizontal velocity components.

A third factor affecting the Profiler can be local changes in reflectivity of any type. For example the operation of the doppler radar is dependent on backscattering of the emitted beam from dust and density fluctuations in the air. The reflection is particularly sensitive to fluctuations whose length scale is half a wavelength. Particularly at high altitudes, inherent atmospheric stability and lack of turbulence can make the reflected signal extremely weak. In these cases backscattering from other sources in the side lobes of the radar can cause spurious reading.

At the radar site a number of separate averages is performed to determine wind speed components at each altitude. As part of the Profiler output the number of averages and the returned signal power is given as output data also. If insufficient averages are available, or the data fails a consistency check, the value '-999' is given for wind magnitude to indicate a bad data point. While the number of averages and the power value

should give some indication of data quality, our experience has been that this is a rather unreliable indicator. A better choice appears to be to view the data graphically,and to visually scan for bad data points. Data currently available at NASA Ames is for the Platteville and Denver sites. Typical data received from these stations is shown in Table 2.2.
Some Profiler stations will be equipped to report on additional surface conditions. In particular, the temperature and pressure are of interest for the ATC problem.

2.2 Wind Profile estimation

Typical wind versus altitude data is shown in Figure 6 for the Platteville and Denver reporting stations. This data needs to be further smoothed, and outlier points removed for a consistent trajectory prediction algorithm. Several types of assumptions on wind variation with altitude could be used. For our purposes we have assumed that the data is to be fit to a series of contiguous straight line segments of velocity versus altitude. The number of segments is variable and is an input from the analyst at the time the data is to be processed. Fitting is done through a weighted least squares algorithm, which is identical to the prediction phase of a Kalman filter. Data is fit independently for the North and East components of the wind. Note that since the Platteville site is aligned in the North-East direction, this is equivalent to smoothing the raw data directly. For other sites with a non-North-East orientation, estimation applied to the natural orientation of the site is probably preferable to

North-East. Note also that while the wind components each vary linearly with altitude, the magnitude and wind heading will not.

For a set of fixed altitude break points for the various segments the data is fit to the segments by the weighted least square algorithm. (see e.g. Bryson[3]) Generally for Profiler data, all data weights for all wind values are assumed to be the same. Outliers are generally explicitly removed from the data although a much reduced data weighting would have virtually the same effect. A few typical fits to the data are shown in Figure 7. If desired, the breakpoint altitudes can also be included as parameters in the data fit. This can lead to improvement in the data residuals, at the expense of more lengthy and possibly ill-conditioned computations. A typical fit with variable breakpoints is shown in Figure 8. Note while this case is well-behaved, the fitting process here is non-linear, and hence an iterative improvement process is required to produce a final fit. If initial points are chosen reasonably, in our experience this fitting process usually proceeds with little difficulty. With a poor initial guess however, this fit can, and has, diverged in some of our simulations. Since divergence in an operational scenario could be quite serious, some safeguards to prevent this would be necessary in an operational setup.

The details of the fitting procedure follow the standard least squares procedure. The Matlab code which has been developed to perform the fit for the Profiler data is given in Appendix 2. Note that in this code, we have added separate outlier logic which searches the fitted data for points which are significantly distant from the fit and automatically marks such data for deletion. In practice,this section worked quite well, and removed only those points which visually we felt to be outliers. Occasionally, if data was particularly noisy and data residuals were high, then some points which visually appeared to be outliers were not detected by this algorithm. In such cases however, the entire Profiler wind data was often suspect, and consideration of other data types may be necessary.

2.3 Other Data types

Other wind data types should be available in an operational system. This will be necessary if the appropriate spatial resolution of the winds is to be determined, and will be necessary at sites not close to a wind profiler site. Such data is most likely data from inertially equipped aircraft, which can determine winds from the combination of air and inertial data. This data is currently available to the airlines- what is needed is a mechanism for obtaining this data for purposes of Air Traffic Control.

Data processing of additional data types is considered to be identical to that of the Profiler data. The only decision is to estimate the relative accuracy of the data so that the appropriate data weighting relative to the other data types can be established.

2.4 Spatial & temporal variation

Wind profiles can exhibit considerable spatial and temporal variation. This variation must be accounted for and captured in a wind estimation strategy for CTAS. For example, a plot of some typical wind profiles measured at close time intervals at the Denver and Platteville profiler sites is shown in Figure 9. Similarly, overplots of the Denver and Platteville winds reported at the same times demonstrate the spatial variability from the two sites (which are less than 50 mile apart). See Figure 10. Because of this, the wind models proposed must be dynamic. That is, it is anticipated that once sufficient data becomes available, a fully time varying, three dimensional wind profile should be the goal of this system. Looking at aircraft arriving on major jetways, the initial goal is to formulate separate wind profiles on major jetways. Temporal variations are included by appropriate filtering of past and current wind data. In particular, a Kalman filter structure with added process noise to fade out old data is a straight-forward and reliable way to ensure optimum usage of wind data. Specific filter design characteristics such as required process noise will need to be determined once real data is available.

3. Conclusion

This report summarizes the author's initial efforts toward formulation of a wind and atmospheric model for implementation into CTAS. A basic approach and set of algorithms has been developed for the wind and atmospheric estimation problems. Additional data types can easily be included in the estimation algorithm. Attention to the logistics of data collection is a significant problem to be addressed in the future to achieve the level of accuracy sought in the CTAS system.

References

[1] U.S. Standard Atmosphere, 1962, US Government Printing Office.

[2] Simonson, Leroy, Commercial Pilot Studyguide, Aviation Book Co.,Glendale, CA.

[3] ______________, Wind Profiler, Training Manual, vols 1-5, National Weather Service, Office of Meteorology, August 1989.

[4] Bryson, A., Ho, Y.C., Applied Optimal Control and Estimation, Halsted Press, J. Wiley & Sons, New York, 1975.

NOAA Wind Profiler Demonstration Network
Site Information - A
January 26, 1990

Nearby Town	State	Site No.	Site ID	Lat. (N) (°, ', ")	Long. (W) (°, ', ")	Elev. (ft.)	Magnetic N Decln (°, ')	Site Orientation from True N (°)
Platteville	CO	1	PLTC2	40 10 48	104 43 10	5000	11 09	0
Lathrop	MO	2	LTHM7	39 34 48	94 10 12	975	04 34	-13
Fairbury	NE	3	FBYN1	40 06 00	97 20 24	1420	06 44	-10
Hillsboro	KS	4	HBRK1	38 18 33	97 17 44	1465	06 38	-8
White Sands MR	NM	5	WSMN5	32 24 22	106 20 57	4016	10 33	-5
Haviland	KS	6	HVLK1	37 39 08	99 05 28	2125	07 44	-10
Neodesha	KS	7	NDSK1	37 22 48	95 38 05	835	05 32	-14
Lamont	OK	8	LMNO2	36 41 28	97 28 57	1005	06 44	-12
Vici	OK	9	VCIO2	36 04 19	99 13 03	2125	07 43	-10
Haskell	OK	10	HKLO2	35 48 28	95 46 54	695	05 40	-14
Purcell	OK	11	PRCO2	34 58 47	97 31 07	1085	06 43	-13
Conway	MO	12	CNWM7	37 31 24	92 42 09	1280	03 30	-16
Slater	IA	13	SLAI4	41 54 03	93 41 57	1035	03 53	-11
Neligh	NE	14	NLGN1	42 12 26	97 47 37	1720	07 00	-8
Winchester	IL	15	WNCI2	39 39 22	90 28 48	557	01 29	-16
Blue River	WI	16	BLRW3	43 13 18	90 31 54	740	00 56	-17
Wolcott	IN	17	WLCI3	40 48 36	87 03 00	695	-01 40	-19
Bloomfield	MO	18	BLMM7	36 53 02	89 58 19	425	01 31	-19
Maynard	MA	19	SBYM3	42 24 51	71 29 16	305	-15 30	-70
Wood Lake	MN	20	WDLM5	44 40 18	95 26 54	1045	05 08	-9
DeQueen	AR	21	DQUA4	34 06 40	94 17 26	640	04 45	-17
Okolona	MS	22	OKOM6	34 05 23	88 51 52	410	01 06	-22
Winnfield	LA	23	WNFL1	31 53 50	92 46 57	305	03 58	-20
Palestine	TX	24	PATT2	31 46 45	95 42 48	391	05 41	-17
Jayton	TX	25	JTNT2	33 01 00	100 58 48	2320	08 26	-11
Tucumcari	NM	26	TCUN5	35 05 03	103 36 33	4070	09 51	-6
Granada	CO	27	GDAC2	37 46 18	102 10 42	3790	09 38	-6
McCook	NE	28	RWDN1	40 05 09	100 39 13	2625	08 52	-6
Merriman	NE	29	MRRN1	42 54 20	101 41 41	3250	09 49	-4
Medicine Bow	WY	30	MBWW4	41 54 09	106 11 09	6551	13 00	+1
Aztec	NM	31	AZCN5	36 50 28	107 54 24	6240	11 56	0

Profiler Program Office
Contact: Margot Ackley
Comm: 303/497-6791
FTS: 320-6791

Table 2.1 Profiler sites

Table 2.2 Raw data received from Denver/Platteville profiler

SITE: DENVER -- horizontal low mode WIND profile
DATE: 90/07/23
TIME: 20:00:00 (UTC) at start of acquisition period

PROFILES:	12	
TIME DOMAIN AVGS:	124	
SPECTRA:	10	
PULSE WIDTH:	1.33	MICRO-SECS
PULSE REP PERIOD:	50.00	MICRO-SECS
MAX HOR VEL:	51.05	M/S
FIRST HEIGHT:	289.78	M/AGL
DELTA HEIGHT:	193	METERS
HEIGHTS (GATES):	32	

GATE	SPEED M/S	DIRECT DGRS	HEIGHT KM/MSL	#E	#N	POWER (EAST) DB	
1	.9	306	1.90	11	9	65.2	C
2	2.8	83	2.09	9	7	77.5	C
3	4.1	39	2.29	11	6	71.0	C
4	.9	43	2.48	11	11	73.1	C
5	1.2	326	2.67	10	10	74.8	C
6	3.3	307	2.87	12	10	75.0	C
7	3.0	299	3.06	12	10	75.1	C
8	6.0	302	3.25	12	11	67.3	C
9	9.5	281	3.45	12	11	59.8	C
10	9.5	290	3.64	12	12	57.0	C
11	9.9	290	3.83	12	12	58.4	C
12	11.3	290	4.03	12	12	54.5	C
13	11.9	298	4.22	12	12	50.3	C
14	12.4	296	4.41	12	11	48.4	C
15	11.6	300	4.61	12	12	55.3	C
16	12.3	304	4.80	12	12	56.9	C
17	12.3	309	4.99	12	12	52.6	C
18	12.8	309	5.18	12	12	52.0	C
19	13.4	308	5.38	12	12	49.2	C
20	13.7	304	5.57	12	12	50.2	C
21	14.1	301	5.76	12	12	49.9	C
22	14.6	296	5.96	12	11	50.5	C
23	14.1	297	6.15	12	10	46.9	C
24	14.7	299	6.34	12	10	48.0	C
25	15.7	309	6.54	11	11	51.5	C
26	17.1	306	6.73	12	12	45.7	C
27	18.1	302	6.92	12	12	43.0	C
28	18.3	304	7.12	12	12	37.1	C
29	18.5	304	7.31	11	12	37.8	C
30	18.5	309	7.50	11	12	41.9	C
31	18.4	310	7.70	12	12	43.4	C
32	18.1	315	7.89	11	12	41.1	C

SITE: PLATTEVILLE -- horizontal low mode WIND profile
DATE: 90/07/23
TIME: 20:00:00 (UTC) at start of acquisition period

PROFILES:	12	
TIME DOMAIN AVGS:	350	
SPECTRA:	8	
PULSE WIDTH:	3.67	MICRO-SECS
PULSE REP PERIOD:	238.00	MICRO-SECS
MAX HOR VEL:	69.81	M/S

Table 2.2 Raw data received from Denver/Platteville profiler(cont)

FIRST HEIGHT: 1786.96 M/AGL
DELTA HEIGHT: 289 METERS
HEIGHTS (GATES): 24

GATE	SPEED M/S	DIRECT DGRS	HEIGHT KM/MSL	#E	#N	POWER (EAST) DB
1	2.8	91	3.31	10	12	61.1
2	1.4	323	3.60	11	11	61.5
3	4.7	290	3.89	12	11	58.2
4	8.8	297	4.18	12	11	62.6
5	11.1	298	4.47	12	11	65.7
6	12.2	296	4.76	12	12	66.8
7	12.8	295	5.05	12	12	66.1
8	13.1	296	5.34	12	12	65.2
9	13.3	298	5.63	12	11	66.4
10	13.4	303	5.92	12	11	68.5
11	13.4	304	6.21	12	11	67.5
12	13.2	306	6.50	12	12	62.4
13	14.3	305	6.79	12	12	55.9
14	16.3	302	7.08	12	11	50.2
15	15.4	307	7.37	12	12	47.6
16	15.4	313	7.66	12	12	50.1
17	16.2	314	7.95	12	10	49.1
18	18.9	309	8.24	11	10	47.5
19	19.3	308	8.53	11	8	44.9
20	20.9	306	8.82	11	8	40.3
21	19.8	300	9.11	12	11	38.9
22	20.1	293	9.40	12	12	40.5
23	20.6	290	9.69	12	11	39.5
24	21.1	292	9.98	12	10	33.7

SITE: PLATTEVILLE -- horizontal high mode WIND profile
DATE: 90/07/23
TIME: 20:00:00 (UTC) at start of acquisition period

PROFILES: 12
TIME DOMAIN AVGS: 124
SPECTRA: 16
PULSE WIDTH: 9.67 MICRO-SECS
PULSE REP PERIOD: 672.00 MICRO-SECS
MAX HOR VEL: 69.78 M/S
FIRST HEIGHT: 4201.77 M/AGL
DELTA HEIGHT: 869 METERS
HEIGHTS (GATES): 16

GATE	SPEED M/S	DIRECT DGRS	HEIGHT KM/MSL	#E	#N	POWER (EAST) DB
1	13.3	302	5.73	12	11	71.8
2	13.7	304	6.59	12	11	67.8
3	15.0	308	7.46	11	11	55.6
4	16.7	311	8.33	11	12	51.0
5	18.9	298	9.20	12	11	44.6
6	20.7	294	10.07	11	10	40.2
7	22.8	292	10.94	10	8	35.6
8	23.2	293	11.81	11	10	36.5
9	23.7	285	12.68	11	9	39.6
10	21.8	285	13.55	11	9	39.8
11	19.5	282	14.42	12	10	33.9

Table 2.2 Raw data received from Denver/Platteville profiler(cont)

12	13.2	303	15.29	11	7	31.0
13	9.3	321	16.16	7	11	29.9
14	4.3	261	17.03	7	10	26.6
15	6.6	279	17.90	4	5	25.5
16	-999.0	-999	18.77	5	3	27.5

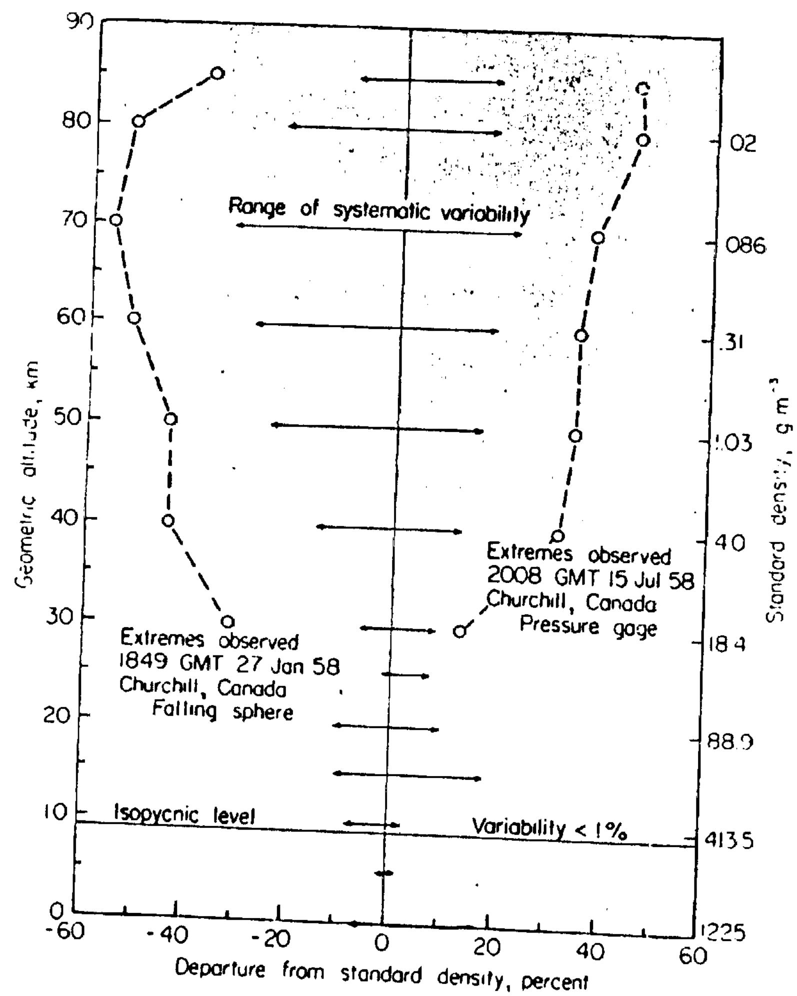


Figure 1 Statistical Variation of Atmospheric Parameters

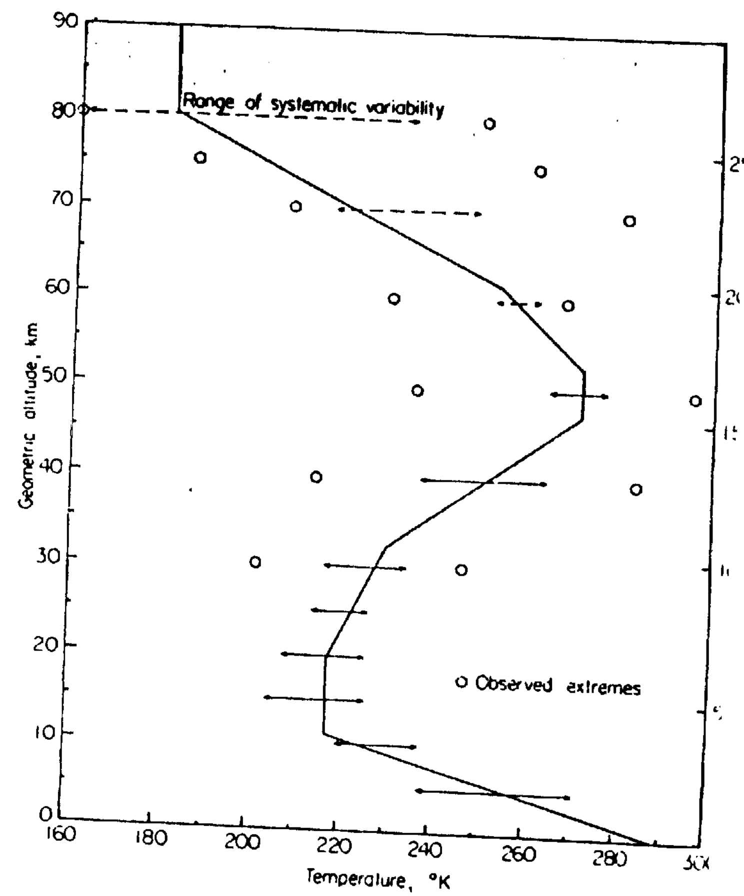


Figure 2 Temperature deviation from standard and the induced altitude errors.

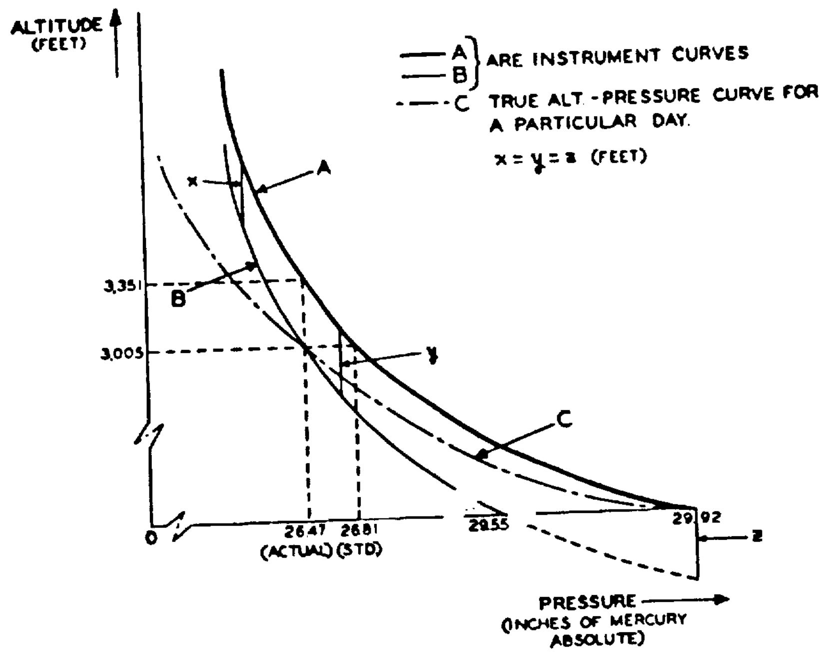


Figure 3 Explanation of the Kollsman window setting.

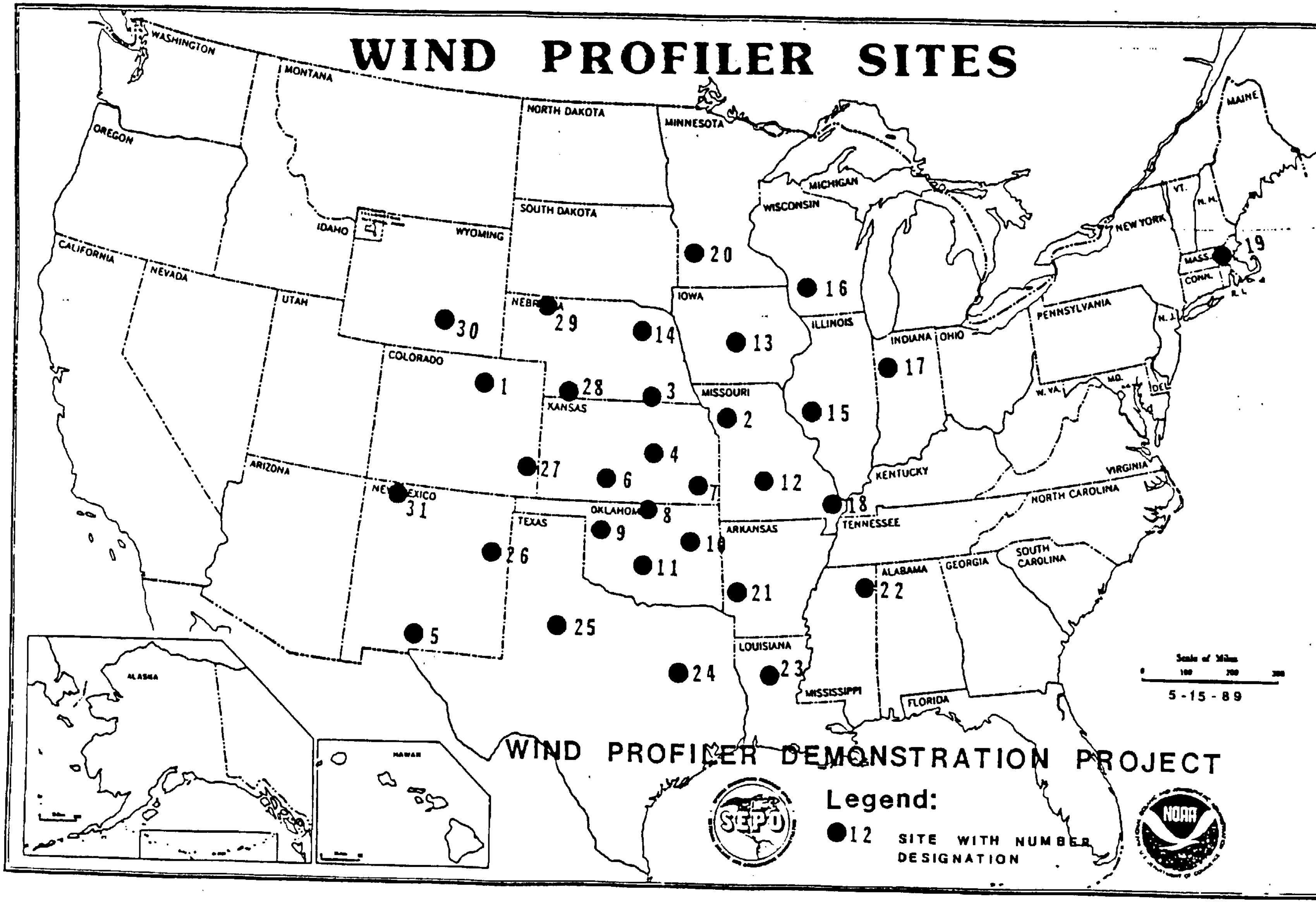


Figure 4 Profiler sites across the United States.

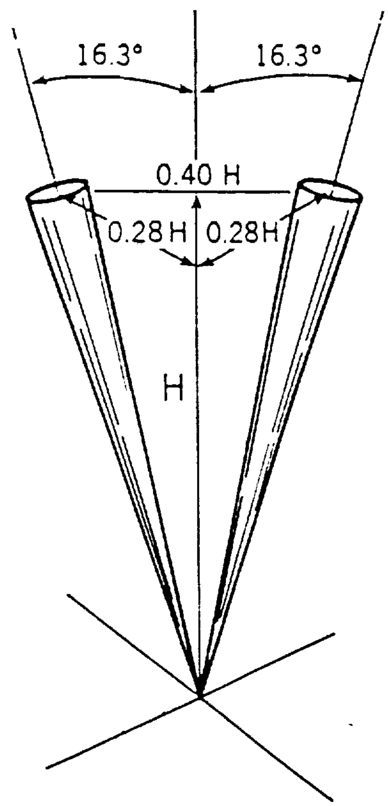


Figure 5 Geometry of a Profiler Radar beam.

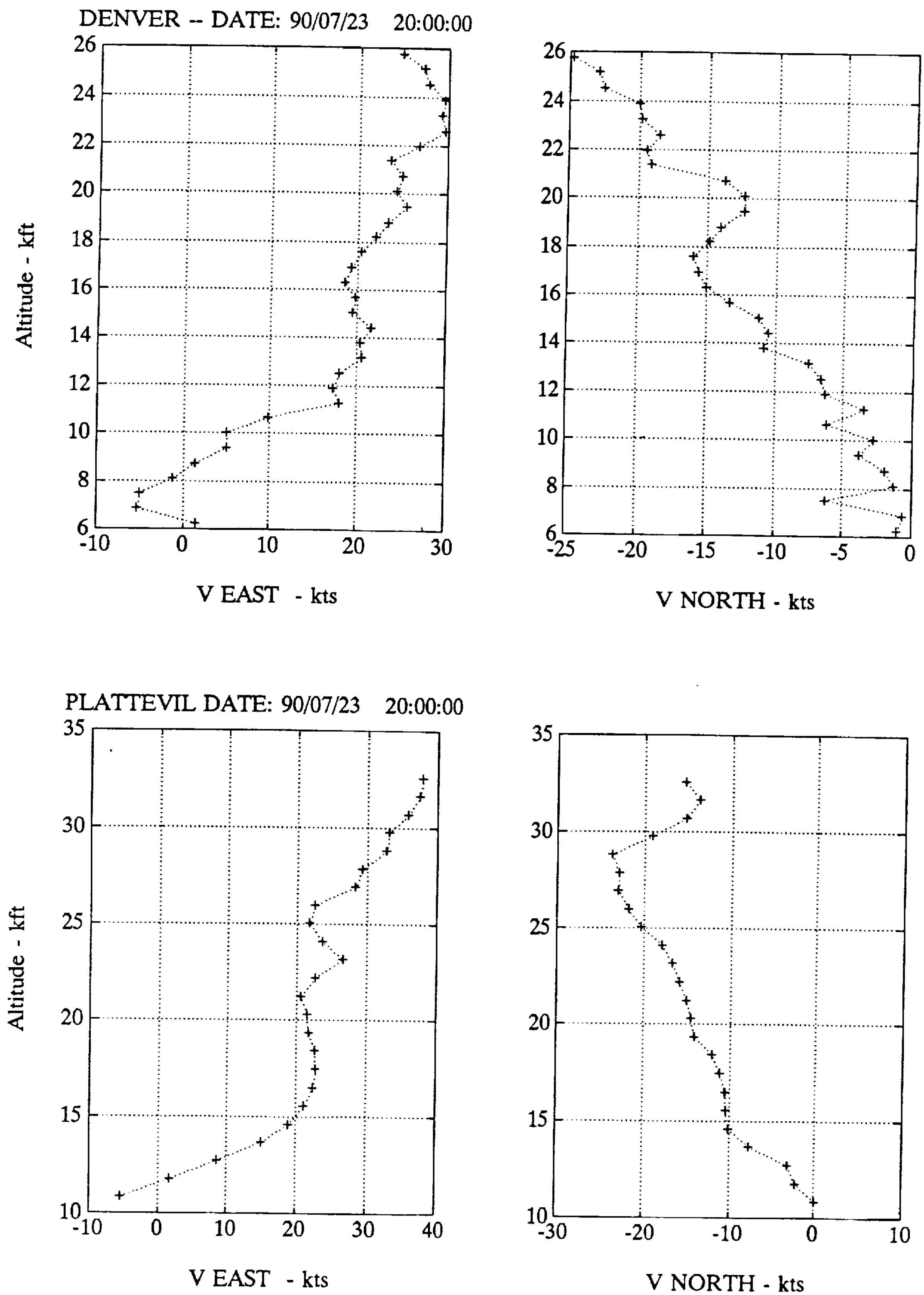
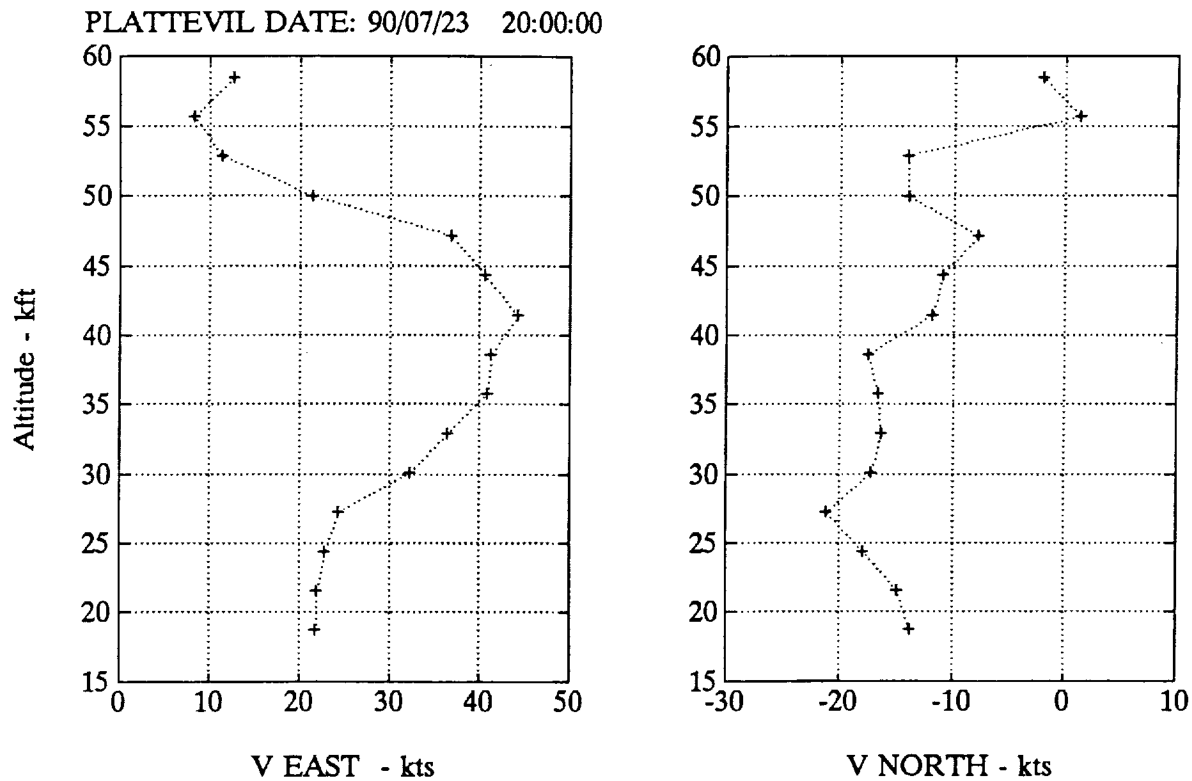
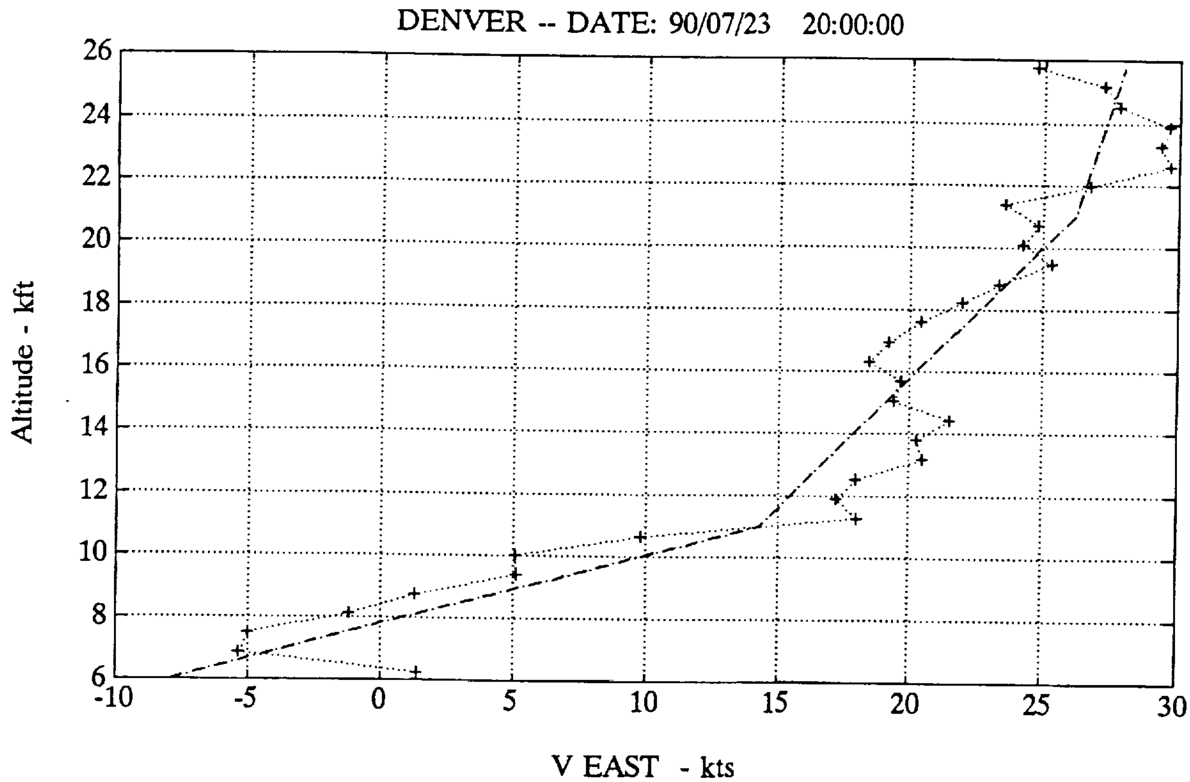


Figure 6 On-line data from Denver-Platteville Profiler site.


PLATTEVIL DATE: 90/07/23 20:00:00
Altitude - kft
V EAST - kts
V NORTH - kts
60
55
50
45
40
35
30
25
20
15
0
10
20
30
40
50
-30
-20
-10
0
10



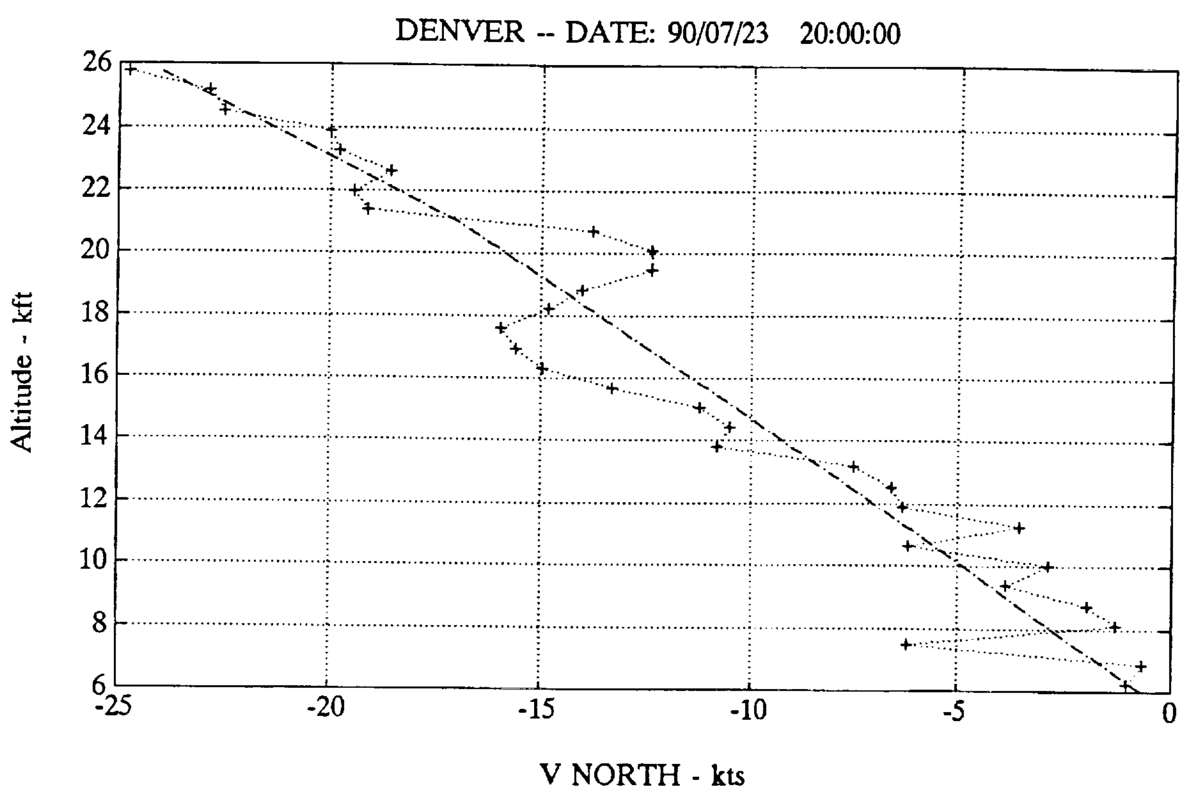


Figure 7 Typical straight line fits to Profiler wind data.

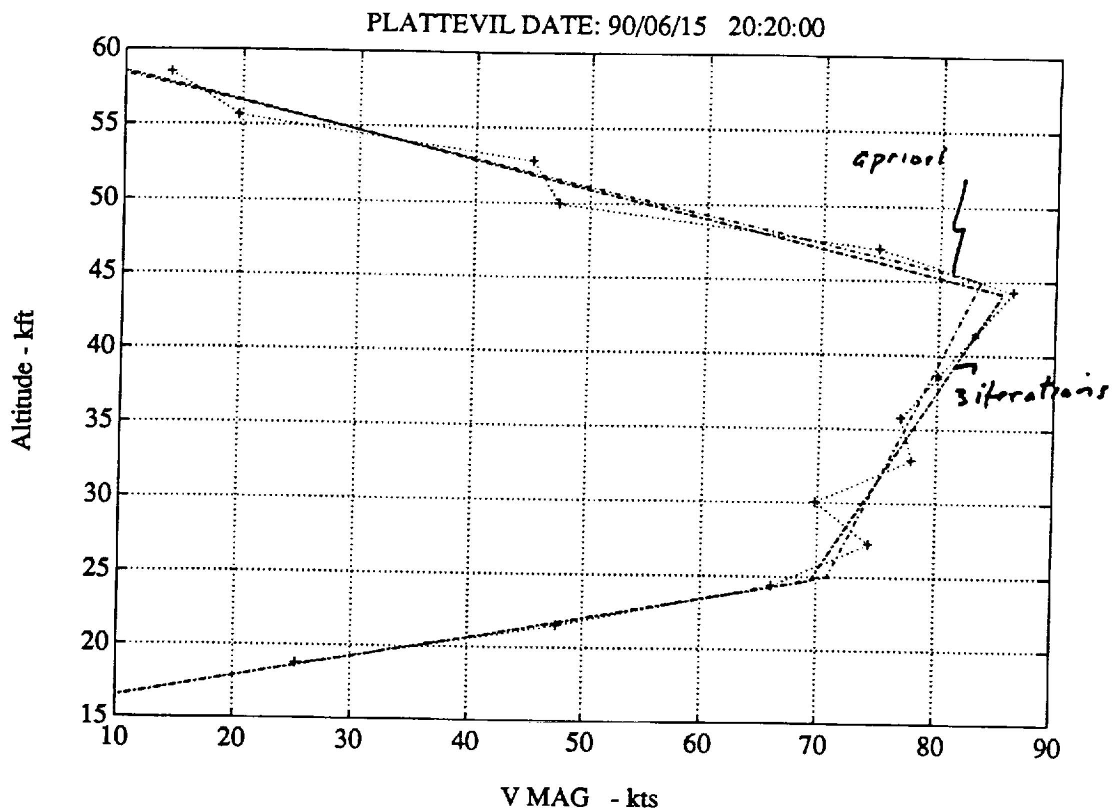


Figure 8 Fit to Profiler data showing iteration of variable breakpoints.

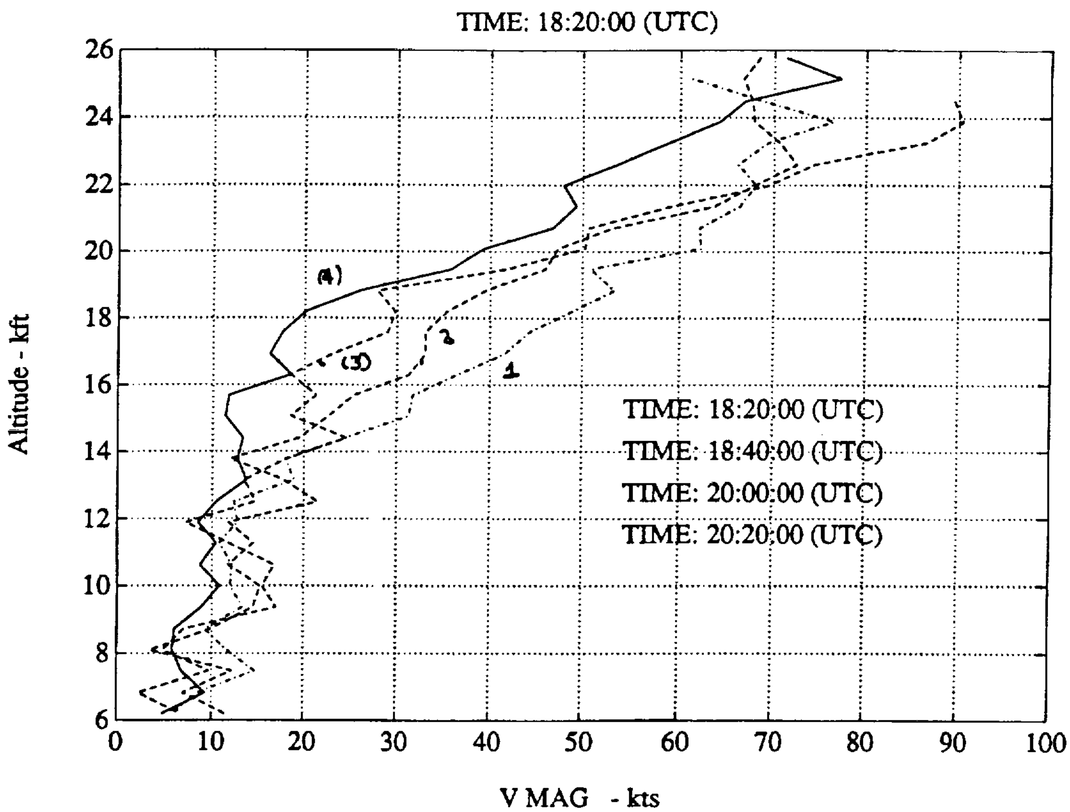
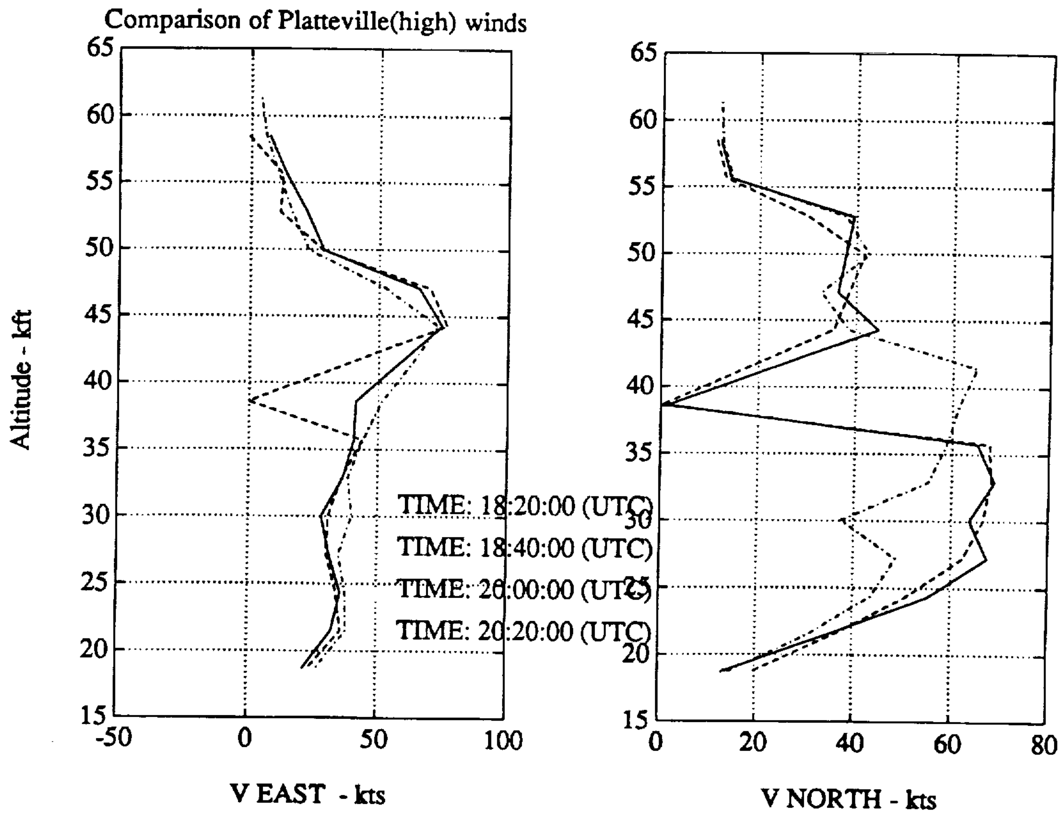
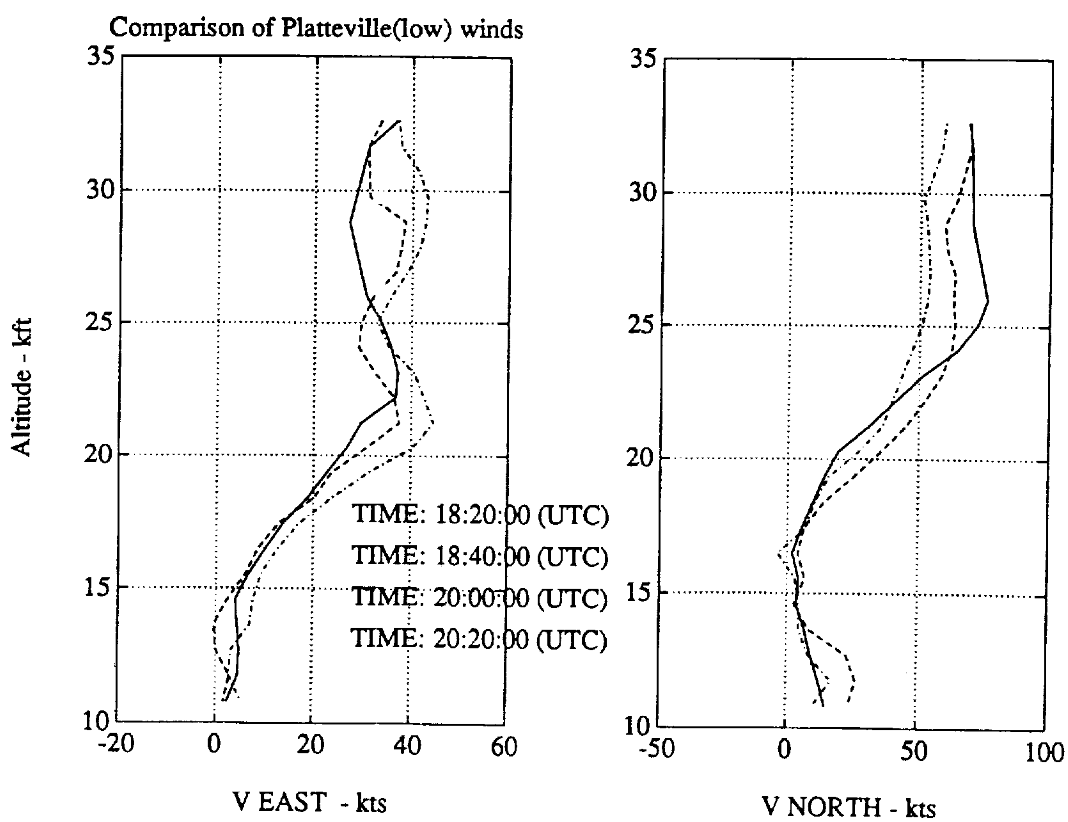


Figure 9 Temporal variation of winds at Denver


Comparison of Platteville(high) winds
Altitude - kft
V EAST - kts
V NORTH - kts
TIME: 18:20:00 (UTC)
TIME: 18:40:00 (UTC)
TIME: 20:00:00 (UTC)
TIME: 20:20:00 (UTC)


Comparison of Platteville(low) winds
Altitude - kft
V EAST - kts
V NORTH - kts
TIME: 18:20:00 (UTC)
TIME: 18:40:00 (UTC)
TIME: 20:00:00 (UTC)
TIME: 20:20:00 (UTC)

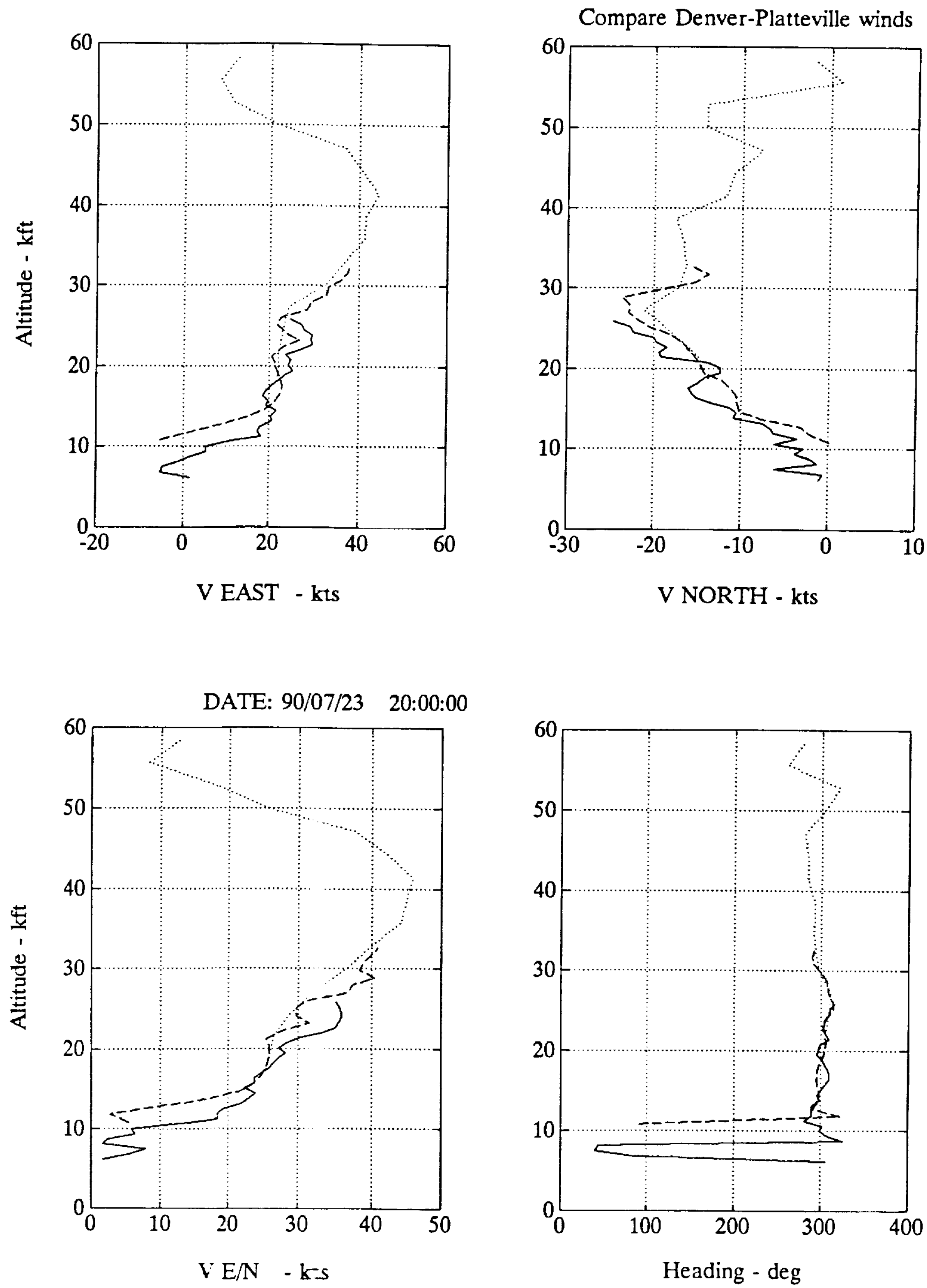


Figure 10 Comparison of winds at Denver and Platteville sites showing spatial variation.

APPENDIX A

```
%NASA Ames atmospheric routines (Matlab code)
%     contains:
                 curr_atm.m  : Generates atmosphere from temperature data
                 inter.m     : Interpolatioon utility
                 pres_alt.m  : Return pressure altitude given h
                 pressure.m  : Return pressure (uses curr_atm) given h
                 rd_alt.m    : Returns altimeter reading given pressure
                 ttempf.m    : Returns temperature given h
                 value.m     : Interpolation utility

% curr_atm.m     gls 8/29/90
%Program to generate a matrix of current atmosphere parameters
%   to determine pressure, density, temperature as functions of
%   geopotential altitude.  Output is matrix ATM defined by
%   ATM=[tempf;    %Input row vector of temperature (deg R)
%        alt;      %Input row vector of altitude (ft)
%        pres0;    %Output row vector of pressure (pres0(1) input)
%        alph;     %Output lapse rate between alt(i)-alt(i+1)/Hs at last pt
%        nexp];    %Pressure exponent or scale height if alph==0
%
%   Used by function Pressure
%
%   The current temperature, altitude data points are fixed here.
%   Change to read from a file if desired
%N.B.  HERE input tempf is deg F--output is deg R
tempf=[100 80 -69.7];
alt=[0 5000 36089];
p0=1013.25;

%THIS IS THE STANDARD ATMOSPHERE
%tempf=[59 41.169 -69.7]; %temp deg F for altitudes in h
%alt= 1000*[0 5 36.089]; %altitude in thousands of ft
%p0= 1013.25;  % pressure at h0

%  INTERNAL variables are all metric for easy reference
cm2f = 1/.3048; %meter to ft EXACT
g0 = 9.80665; R = 1000*8.31432/28.9644;
conv= 1.8/cm2f; % deg K/m  to deg R/ft
gqR = conv*g0/R; % units now deg R/ft
cN2p =1/(0.45359237*g0); %Newtons to pounds
tref= 1.8*273.15 -32; % 0 deg F
eps = 1.e-7;   %tolerance for "0" lapse rate
% end setup

LL= length(tempf);
% Form an array of lapse rates, exponents and pressures
% assume isothermal above last temperature given
pres0(1) = p0;
for ii=1:LL-1
   alph(ii) = (tempf(ii+1) - tempf(ii) )/( alt(ii+1) - alt(ii) );
   if abs( alph(ii)) > eps;
```

```
      nexp(ii) = -gqR/alph(ii);
      pres0(ii+1) = pres0(ii)*( (tref+tempf(ii+1))/(tref+tempf(ii)))^nexp(ii);
    else
        nexp(ii) = (tempf(ii) +tref)/gqR; %this is scale height
        pres0(ii+1) = pres0(ii)*exp(-( alt(ii+1) - alt(ii))/nexp(ii) );
      end;
end; %for
alph(LL)= 0;
nexp(LL) = (tempf(LL) +tref)/gqR; %this is scale height
for ii=1:LL, tempf(ii)= tempf(ii)+tref;end;
ATM=[tempf;alt;pres0;alph;nexp];
```

****

```
%function to return interpolation indices for one dimensional array
%   See function value to return interpolated value
%function zzz = inter(h,harray)
%   returns zzz=[iref,a] such that
%   h = (1-a)*harray(iref) + a*harray(iref+1);
%   iref = 0  if h < harray(1)
%   iref = -1 if h > harray(max)
%         and  a= h - harray(max)
%   gls 8/23/90

function zzz = inter(h,harray)
[nr,nc]=size(harray);
if min([nr,nc]) ==1
   L1=max([nr,nc]);
else disp('Not a 1-d array');return
end;
if h>=harray(L1)
    iref=L1; a= h-harray(L1);
elseif h<harray(1),
    iref=0; a= h- harray(1);
else
    stop=0; i=2;
    while stop==0;
       if h<harray(i)
           iref=i-1;
           a = (h - harray(iref))/(harray(iref+1)-harray(iref) );
           stop=1;
       else i=i+1;
       end;
    end;
end;
zzz=[iref, a];
```

****

```
% function to return pressure altitude for a given pressure
%
%  altpres = pres_alt(press);OR pres_alt(press,str)  str='mb' or 'hg'
```

```
%          uses English units (ft, psf) unless mb - milibars
%                                                  or hg - in hg specified
%  INTERNAL variables are all metric for easy reference
%  Use the NASA (ICAO) standard atmosphere
%function altpres = pres_alt(press,str);

function altpres = pres_alt(press,str);

cm2f = 1/.3048; %meter to ft EXACT
g0 = 9.80665; R = 1000*8.31432/28.9644;
gqR = g0/R;
alphstd= -.0065;
htropo = 11000;
tempstd= 288.15; %15deg C
t_tropo= tempstd + alphstd*htropo;
hscale = R*t_tropo/g0;
pstd= 1013.25; %millibars
cN2p =1/(0.45359237*g0); %Newtons to pounds
pstdE = 100*pstd*cN2p/(cm2f^2);
                  %pstdE= 2116.2; %English units (psf)
M2E= pstdE/pstd;
p_tropo = pstd*(t_tropo/tempstd)^(-gqR/alphstd);

% end setup
if nargin ==2
    if str=='mb', disp('mb')
    elseif str=='hg', press = press*pstd/29.9213;disp('hg')
    end;
else press = press/M2E;
end;

if press > p_tropo
      altpres  = (tempstd/alphstd)*( (press/pstd)^(-alphstd/gqR) - 1);
else  altpres = htropo + hscale*log(p_tropo/press);
end;
altpres=cm2f*altpres;  %remove this to return in meters
```

****

```
% Given a temperature profile, generate pressure vs. altitude
% Assume linear temperature profile between points
%
%   function  press = pressure(alt,ATM)
%    alt  in "ft"; pressure in "mb"
%    Matrix ATM generated by curr_atm.m contains current atmosphere model
%
% Uses functions inter.m and value.m to get table values
function press= pressure(alt,ATM)

temp0= ATM(1,:);
harray=ATM(2,:);
press0=ATM(3,:);
alph=ATM(4,:);
```

```
nexp=ATM(5,:);

zzz= inter(alt,harray);
iref=zzz(1); a=zzz(2);
temph= value(temp0,zzz);
if alph(iref) ~=0
   press= press0(iref)*(temph/temp0(iref))^nexp(iref);
else  press = press0(iref)*exp(-(alt - harray(iref))/nexp(iref));
end;
```

****

```
% function to return altimeter reading for a given pressure
%
%  altpres = rd_alt(press,kollsman);OR rd_alt(press,kollsman,str)  str='mb' or
'hg'
%              press is the current pressure
%                  English units (psf) unless mb - milibars
%                                              or hg - in hg specified
%              kollsman is the altimeter setting (kollsman window)
%                assumed in in Hg unless str='mb' --> then in mb also
%              altpres is in (ft)
%
%  Use function pres_alt.m which uses
%  the NASA (ICAO) standard atmosphere
%function altpres = rd_alt(press,kollsman,str);

function altpres = rd_alt(press,kollsman,str);

% function
if nargin ==3
    if str=='mb',
       h1=pres_alt(press,'mb');
       h2=pres_alt(kollsman,'mb');
       altpres= h1 -h2;
    elseif str=='hg',
       h1=pres_alt(press,'hg');
       h2=pres_alt(kollsman,'hg');
       altpres= h1 -h2;
    end;
elseif nargin==1, kollsman=29.92;
end;
if nargin<3,
   h1= pres_alt(press);
   h2= pres_alt(kollsman,'hg');
   altpres=h1-h2;
end;
```

****

```
% Given a temperature profile return temperature for a given h
```

```
% Assume linear temperature profile between points
%
%   function  temp=ttempf(h,ATM)
%    h is alt  in "ft"
%   Matrix ATM generated by curr_atm.m contains current atmosphere model
%
% Uses functions inter.m and value.m to get table values
function ttemp= ttempf(h,ATM)
temp0= ATM(1,:);
harray=ATM(2,:);
zzz= inter(h,harray);
iref=zzz(1); a=zzz(2);
ttemp= value(temp0,zzz);
```

****

```
%   function to return function value using linear interpolation
%   on one dimensional array.
%   Input are interpolation indices
%   See function value to return interpolated value
%
%   function h= value(harray,[iref, a])
%
%   [iref,a] returned from inter.m such that
%   h = (1-a)*harray(iref) + a*harray(iref+1);
%   iref = 0  if h < harray(1)
%            returns h=harray(1)
%   iref = max if h > harray(max)
%            returns h=harray(max)
%   gls 8/23/90

function h = value(harray,zz)
iref= zz(1); a= zz(2);
[nr,nc]=size(harray);
if min([nr,nc]) ==1
   L1=max([nr,nc]);
else disp('Not a 1-d array');return
end;
if iref<=0
   h=harray(1);
elseif iref==L1
   h= harray(iref);
else
   h = (1-a)*harray(iref) + a*harray(iref+1);
end;
```

****

APPENDIX B

```
%NASA Ames routines for wind profile fit (Matlab code)
%       contains:
                 wnd7.m      : Read Profiler data and plot
                 fit7.m      : Least squares fit to data
                               both these current versions are hardwired
                               to eliminate options available in early code

****************************************************************************

% wnd7.m  8/22/90  Master in /matlab/gls ALL Plat mod
%
%test wind profile
%load wind profile from data file
disp('Data should be loaded in standard format')
PLAT= 'PLAT'; DENV='DENV';
E1=exist('txt1a'); wher1=' ';
E2=exist('txt2a'); wher2=' ';
E3=exist('txt3a'); wher3=' ';
nam=[ 'V EAST '
      'V NORTH'
      'V MAG  '
      'V E/N  '];
disp(' Plot data for')
if E1, wher1 = txt1a(7:10); disp([' 1: ' txt1a]);end;
if E2, wher2 = txt2a(7:10); disp([' 2: ' txt2a]);end;
if E3, wher3 = txt3a(7:10); disp([' 3: ' txt3a]);end;
disp([' 4: ' 'Platt.- ALL']);end;
disp(' ')
numin= input('     -->');
if numin==4,  %this finds the Plattville data--IF IT EXISTS--
  plat1=0;plat2=0;
  allplat=1;
  if wher1==PLAT, plat1=1;numin=1;
  end;
  if wher2==PLAT,
         if plat1==1, plat2=2;
         else plat1=2;numin=2;
         end;
  end;

  if wher3 == PLAT,
          if plat1>0, plat2=3;
          else plat1=3;numin=3;
          end;
  end;

else allplat=0;
end; %numin==4

data= eval( ['data' num2str(numin)]);
txta= eval(['txt' num2str(numin) 'a']);
txta=[txta(7:15) ' ' txtb(1:15) ' ' txtc(6:14)];
```

```
if allplat,
        [lowlen,temp] = size(data);%Save this for future use in fit
        data= [data;eval( ['data' num2str(plat2)])];
        txta=[txta(1:15) '-ALL' txta(16:35)];
        numin = 4;
        [hilen,temp]=size(data);
end;

disp(' ')
%  EAST & NORTH winds only
iplt=0;
   vmps= data(:,2);
   degs= data(:,3);
   altkm=data(:,4);

%
cd2r= pi/180.;
cm2f= 39.2/12;
cfps2kt= 1/1.69;
% make invalid data to nan
LL = length(vmps)
for ii=1:LL
   if vmps(ii) < 0,
      degs(ii)=nan; vmps(ii)= nan;
   end;
end; %ii loop
vxfps= -cm2f*vmps.*sin(cd2r*degs);
vyfps= -cm2f*vmps.*cos(cd2r*degs);
altkft= cm2f*altkm;
vplot= cfps2kt*vxfps;
vplot2= cfps2kt*vyfps;

%make sure hold is off and screen is clear
hold off
clg
subplot(121);  % Do multiple plots
   if allplat,
      plot(vplot,altkft,'+');hold
      plot(vplot(1:lowlen),altkft(1:lowlen),':');
      plot(vplot(lowlen+1:hilen),altkft(lowlen+1:hilen),':');
    else  plot(vplot,altkft,':',vplot,altkft,'+');
    end;
title(txta);grid;
xlabel([nam(1,:) ' - kts']); ylabel('Altitude - kft');
   lftaxis=axis;axis; %capture axes then release
   hold off
  if allplat,
      subplot(122)
      plot(vplot2,altkft,'+');hold;
      plot(vplot2(1:lowlen),altkft(1:lowlen),':');
      plot(vplot2(lowlen+1:hilen),altkft(lowlen+1:hilen),':');
  else  plot(vplot2,altkft,':',vplot2,altkft,'+');
  end;
  xlabel([nam(2,:) ' - kts']);
```

```
  grid;
rgtaxis=axis;axis; %capture axes then release
%[xc,yc]=ginput(1,'sc');
%text(xc,yc,txtb,'sc')
%text(xc,yc+.05,txtc,'sc')
hold on
disp('do auto fit')
fit7
```

```
***************************************************************************
```

```
%fit7.m
% fit wind data to best straight line segments using least square fit
% Break points for altitudes are variable in this version
% and number of segments are fixed.s
% See fit1.m for fixed break points.
%
%  8/19/90  mod for fit7.m  in iter==1 only remove outliers automatically
%          if fitted data point is very far out of bounds
% gls 6/18/90  mod to fit4 7/18/90 for fixed or variable
% 6/19/90 add for fit on subplots -->fit5.m
% 6/20/90  add weighted ls + add other vel. data
% keyboard command disabled due to problems on SUN's

sig1s=1;   %Covariance of basic wind data
sig2s=.01;  %Covariance of "added" data
epsp= 1.e-3; parmchg=1;
hseg1=[ 25.0]; %altitudes in kft
hMAX= 1000;
disp('This program will fit straight lines to standard data')
disp('Assume equal noise covariance of sig1s=')
disp(sig1s)
disp('It will also pick up additional data from arrays')
disp('VEAST VNORTH  HNEW if  EXISTNEW is a defined variable')
disp('This data has covariance sig2s=')
disp(sig2s)
disp('Assume for this case data is plotted in E/N form')
itermax=10  %fixed for now
disp('itermax=1 for fixed breakpoints');
disp(' ')
itermax=input('itermax= ')
disp('Determine least squares fit')
disp(' to straight line wind profile segments')
hseg1=input('Input initial breakpoint vector: ');
nsegs= length(hseg1)+1;
nparm= nsegs+1;
NEWDATA = exist('EXISTNEW'); %NEWDATA=1 if data
if NEWDATA, lnewdata= length(VEAST);end;
%number of parameters (per curve)in fit (exclude breakpoints!)
if iplt==0 | iplt==4, nparm2=nparm;
  else nparm2=0;end;
nparmt= nparm+nparm2;
%Get rid of nan's in data (+ outliers marked as nan)
```

```
LL= vplot~=nan;
vfit=vplot(LL);
h= altkft(LL);
% Make sure its a column vector
[nr,nc]=size(vfit); if nr==1 vfit= vfit';end;
[nr,nc]=size(h); if nr==1 h = h';end;
if NEWDATA, vfit=[vfit;VEAST]; h=[h;HNEW]; end;
     %if subplots, alt data string is same on both
if nparm2 > 0, vfit2= vplot2(LL);h2 = h;
    if NEWDATA, vfit2=[vfit2;VNORTH];end;
    vfitt=[vfit;vfit2];ht=[h;h2];
 else vfitt=vfit; ht=h; end;
%
onevec=ones(vfit);
onevect= ones(vfitt);
% now work with only reduced data
lvel= length(vfit);
lold= lvel- lnewdata;
%
iter = 1;
%
%  Enter main loopsize(
%
while parmchg > epsp  %loop over estimator until convergence is met
  hseg= [ 0 hsegl hMAX]; %segment boundaries
  A= zeros(lvel,nparm);
  A(:,1)=ones(vfit);
  L0=  h > hseg(1)*onevec;
  for jcol=2:nsegs+1
    L1=  h > hseg(jcol)*onevec;
    L01 = L0 - L1;
    A(:,jcol) =  hseg(jcol)*L1 - hseg(jcol-1)*L0 + h.*L01 ;
    L0 = L1;
  end; %for jcol
  if nparm2 > 0,
      zA= 0*A;
      A= [A zA; zA A];ht=[h;h];
  else ht=h; end;
  % Now add columns for breakponts
  %   but only in the second and higher iteration
  if iter ==1,  oldparm=zeros(nparmt,1);
  else  %for iterations above 1, estimate break points
    A12= zeros(lvel,nsegs-1); A22= A12;
    for jcol=2:nsegs
       L1=  h > hseg(jcol)*onevec;
       A12(:,jcol-1) = ( parm(jcol)- parm(jcol+1) )*L1;
     end; %for jcol
     if nparm2 > 0
       for jcol=2:nsegs
          L1=  h > hseg(jcol)*onevec;
          A22(:,jcol-1) = (parm(nparm +jcol)- parm(nparm+jcol+1) )*L1;
       end; %for jcol
     A12= [A12;A22];
     end;%nparm2
```

```
      A= [A A12];
      if iter==2, oldparm=[parm; hseg(2:nsegs)'];
         else oldparm = parm;
      end;
   end; %if iter
   Vfit= A(:,1:nparmt)*oldparm(1:nparmt); %Estimated measurements
   RIA=A;
   if NEWDATA,
      for irow=1:lold,
             RIA(irow,:)= RIA(irow,:)/sig1s;
             RIA(lvel+irow,:)= RIA(lvel+irow,:)/sig1s;
             end;
      for irow=lold+1:lvel;
             RIA(irow,:)= RIA(irow,:)/sig2s;
             RIA(lvel+irow,:)= RIA(lvel+irow,:)/sig2s;
             end;
   end;%if NEWDATA
   Api= inv(A'*RIA)*RIA';
   dparm=Api*( vfitt - Vfit);
   parmchg= sqrt(dparm'*dparm);
   parm= dparm+oldparm;
%Get new residuals for test
   Vfit= A(:,1:nparmt)*parm(1:nparmt); %Estimated measurements
   delv = ( vfitt - Vfit);
   rmsdv = sqrt( delv'*delv/length(delv));
   maxdelv= max( abs(delv) );
  disp(['RMS fit deviation = ' num2str(rmsdv) ' kts'])
  disp(['MAX fit deviation = ' num2str(maxdelv) ' kts'])
  % Check velocity deviations here on first iteration
  % auto delete outliers
    if iter == 1
       rmst= max([rmsdv 2.5]); %Use 4*rms or 10 as outlier test
      Lol= abs(delv) > 4*rmst*onevect;
      lLol= Lol'*Lol; %this dot product gives the number of outliers
      if lLol == 0, disp('NO OUTLIERS FOUND')
       else
             disp([ num2str(lLol) ' OUTLIERS LOCATED AT alt='])
             disp(ht(Lol))
             %autodelete outlier--tricky logic as we will have to delete
             %data from both sets if E/N data being used
             if nparm2 > 0
                  Lol1= Lol(1:lvel); Lol2= Lol(lvel+1:2*lvel);
                  Lol1 = Lol1 + (Lol2 > Lol1);
                  Lol2 = Lol2 + (Lol1 > Lol2);
                  Lol1 = onevec - Lol1;
                  Lol2 = onevec - Lol2; %Here Lol1 & Lol2 should be the same!
                  Lol = [Lol1;Lol2];
                  %Now get rid of outlier
                  vfitt= vfitt(Lol); h=h(Lol1);
                  vfit=vfit(Lol1); vfit2= vfit2(Lol2);
             else
                 Lol = onevec - Lol;
                 vfitt= vfitt(Lol);
                 vfit= vfit(Lol); h= h(Lol);
```

```
            end;%nparm2
            iter = 0; %final step- set iter back so iter=1 done again
            onevect = ones(vfitt);
            onevec=ones(vfit);
            lvel= length(vfit);%****NOTE ASSUME FOR NOW DELETED DATA WAS IN
OLDDATA SET
    end;%if lLol
  end;%if iter==1
%   if iter > 0 disp('Estimated parameters:')
%                disp(parm)
%       end;
   if iter >1, hsegl= parm(nparmt+1:nparmt+nsegs-1)';
        end;
   iter= iter+1;
   oldparm= parm;
status=[   iter, parmchg,hsegl]; %display output
disp(status)
  if iter >itermax
     disp('--quit');
     parmchg=-parmchg; end;%if itermax
end; %while loop
%
% converged or quit. Now plot fit
%
htt= max( h );
hplot=[0 hsegl htt]; %plot breakpoints and last point
vfit=[parm(1)];
for ii=2:nparm
   vfit(ii)= vfit(ii-1) + parm(ii)*(hplot(ii) - hplot(ii-1) );
end;
hold on
if nparm2 > 0,
  subplot(121);axis(lftaxis);plot(vfit,hplot,'b')
  vfit2=[parm(nparm+1)];
  for ii=2:nparm
    vfit2(ii)= vfit2(ii-1) + parm(nparm+ii)*(hplot(ii) - hplot(ii-1) );
  end;
  subplot(122);axis(rgtaxis);
  plot(vfit2,hplot,'b')
else plot(vfit,hplot,'b')
end;
disp('End of fit7.m')
end % fit program
```

****