RADIANCES AND CLOUD ANALYSES FROM GOES DWELL SOUNDINGS

Space Science and Engineering Center, University of Wisconsin

User Guide

Version 1.03

Compiled by

Francis Bretherton and Don Wylie

June 26, 1998

Table of Contents

---

1. Science Objectives

This data set makes available, in summary form, a 7 day sample during May 1988 of the previously inaccessible archived holdings of VAS dwell-soundings from the U.S. Geostationary Orbiting Environmental Satellites (GOES). The sample is Phase 1 of the Project "Radiances and Cloud Analyses from GOES Dwell Soundings" at the University of Wisconsin, Space Science and Engineering Center.

The VAS observations provide unique constraints on the distribution of water vapor and related cloudiness in the middle and upper troposphere throughout the diurnal cycle over the continental U.S. for the period 1981-1996, as well as measures of surface brightness temperature, and of the horizontal distribution of temperature in thick vertical layers. They provide an important quantitative record of variations over this period in atmospheric properties from hour to hour up to time-scales of years, which, because it has been inaccessible, has not so far been incorporated any systematic manner into our corpus of understanding of weather and climate. Because the dwell-sounding data are confined to the infrared wavelengths, available information about the lower troposphere is less specific and is not emphasized in this summary. However, to enable complementary estimates of daily fluctuations in the absorption of solar energy at the Earth surface, the visible reflectance measured approximately concurrently by the GOES imager is also included in the summary, at a horizontal resolution of about 120 km. A novel feature of this data set is the provision of situation-dependent estimates of uncertainty for the derived variables. These are intended to facilitate assimilation into and comparison with dynamical models and to enable statistical studies of potential bias in climatologies derived from the data set.

The target user for this prototype data set is a research scientist who is engaged in the next generation of re-analysis of weather data including both climate and forecast applications, and who is is equipped to handle the direct assimilation of

radiances in addition to radiosondes and other conventional data into an atmospheric General Circulation Model (GCM). Another major target group of users consists of Global Climate Model developers seeking validation and statistical evaluation for alternative model algorithms which control the simulated relationship between cloudiness and water vapor for various seasons and times of day. Other possible applications include a variant of the GOES Precipitation Index (GPI; Arkin, 1997). Using the CO2 channels to report more reliably the area of thick as opposed to thin cirrus, a more refined correlation should be obtainable over the U.S. with rain-gage data, enhancing possible uses of the GPI as a surrogate for direct measurements convective rainfall. Another potential application of the dataset is in broad scale estimates of potential evapotranspiration from the land surface.

To facilitate assimilation and other applications of what, in raw form, is a large and unwieldy dataset, the data products are statistical summaries relevant to a grid of squares bounded by circles of latitude and longitude 1 degree apart over the continental U.S. and neighboring ocean, at times of day which correspond to the available observations, which average every 90 minutes for 1987-88. Because of exigencies in the observing schedules and imperfections in the archive, there are inhomogeneities and some gaps in the coverage at different temporal epochs, particularly in the early years, and the full complement of channels is often not available. Each grid area-time cell is associated with a fixed suite of derived variables, specifically the average radiance for each available VAS channel, and estimates of the channel radiances for clear sky, the effective cloud fraction, cloud pressure level and radiating temperatures, for high, middle and low cloud, together with measures of uncertainty for each variable. The visible radiance from a contemporary VISSR image averaged over a cell is included without further interpretation. This suite of variables has been chosen to convey for the target users as much of the useful information contained within the original data set as possible, consistent with a substantial reduction in the original data volume to a design goal of a few GByte/year of record. For many purposes such as documenting the diurnal cycle of the relationship between synoptic scale water vapor and cloudiness at various levels, it will be appropriate to substantially further reduce the data volume by inferring secondary climatologies. However, the specific implementation is left to the user communities concerned.

It should be understood that, around each nominal time, the actual measurements were spread over approximately 1/2 hour, and that, depending on location, the values reported for each grid square are derived between 36 and 240 raw pixels with a nominal resolution of 8 km. Depending on the channel, either the whole or approximately half the area of the grid square is actually sampled each nominal time. It is anticipated that, within the statistical uncertainties inherent in the summarization, users will interpolate to their grid of choice, provided it is of comparable or coarser resolution.

Apart from the cloud information, full and accurate interpretation of these cell radiances requires comparison, channel by channel, with the results of a directional forward calculation starting from user supplied vertical profiles of temperature and moisture, including the temperature and emissivity of the underlying surface, and concentrations of other absorbing gases such as CO2 and ozone. Where discrepancies with the observed radiances are found, sensitivity studies may be needed to establish probable cause. A reference Fortran code is supplied to enable such forward calculations under clear sky conditions with what is believed to be sufficient accuracy. Comparisons with some state of the art line-by line calculations are also provided. If users prefer their own code, results may be compared with this reference. However, some rough guidance about the normal information content of the various channels is given in the section Introduction to VAS below, which may provide a crude substitute for extensive sensitivity calculations. In particular, atmospheric temperatures may usefully be discussed using only channels 2-5 and 8.

The design philosophy for this product has been to provide for the target user groups as much useful scientific information as possible from the original data, consistent with a manageable product volume, while minimizing extraneous modeling assumptions. Because cloud models play a central role in the interpretation of the raw radiances, the algorithms are inevitably non-linear and complex, with several tunable parameters. Supplemental variables have been included in the output to enable evaluation of the sensitivity of the results to the values of these parameters, and to provide situation dependent estimates of uncertainty for all inferred atmospheric variables. A central issue is how to summarize in an objective manner what can reasonably be inferred from a full-resolution satellite image to a spatial scale commensurate with a general circulation model. Spatial averaging interacts with non-linearity in ways which depend on the sub-grid scale statistical structure of the cloud fields. Modeling simulations of the workings of the algorithm for different assumptions about such structure, coupled to statistical analysis of a large sample of cases covering a wide range of meteorological conditions, are probably necessary to elucidate the true bias and probable error in these results. However, such evaluations are feasible only once the data are readily accessible. This sample product is a first step in such a scientific evaluation.

2. **Glossary**

VISSR: Visible Infrared Spin Scanning Radiometer.

GOES/VAS: Geostationary Orbiting Satellite System/VISSR Atmospheric Sounder.

Grid: A regular array of Earth located data. Here the *standard grid* refers to 1 degree squares of latitude and longitude, bounded by 50.5 N, 130.5 W and 24.5 N, 39.5 W. It covers the continental U.S. and adjacent ocean.

Cell: The unit of reporting, which is a grid square at a given nominal date and time. Products are reported for uniformly space grid of cells sized 1 degree latitude by 1 degree longitude.

Pixel: An element of an electronic image of a scene. Also the approximate spatial area sampled by an individual data value from a satellite data stream. For GOES/VAS, pixels are nominally 8 km x 8 km at zenith, but increase in size with increasing values of the angle {ASaZ} between the line of sight to the satellite and the local zenith at the point of emission or reflection. In practice the areas sampled by neighboring pixels may overlap or have gaps between them, and a single data value should be considered merely representative of the neighborhood. Information from individual pixels within a cell is averaged before reporting.

Channel: The data stream corresponding to a particular narrow band of wavenumbers. GOES/VAS has 12 channels, detailed in Table 1, which are designed to maximize the useful information about the state of the atmosphere. Sequential scans of the same line of pixels with different channel filters are completed within a few seconds to provide 12 effectively simultaneous views of the same earth scene. The visible channel at the same nominal date-time is scanned separately within 30 minutes.

Missing data: A gap in the ability to derive a reliable product, usually because the requisite observations were not made because of other operational priorities, or because they cannot be recovered from the archive, or because of excessively noisy or unreliably calibrated data. It is designated in the product fields by the value -1.

Radiance (directional): The radiant energy per unit area, per unit solid angle and per unit wavenumber, incident on the telescope of a satellite after emission or reflection from the earth or its atmosphere. This radiance depends on {ASaZ}. For reflected solar radiation it depends also on the solar zenith angle {ASoZ}, and the solar scattering angle {ASoS}. For emitted radiation it is supposed independent of azimuth.

Clear sky radiance: The directional radiance that would be emitted from a cloud free area. It is assumed to be adequately calculable using radiative transfer theory from an atmospheric profile expressing temperature, humidity, and ozone for a cell as functions of pressure, with a black underlying surface. The clear sky radiances in all channels are estimated for each cell as the observed average from those pixels within the square which are deemed to be clear, or by interpolation from surrounding squares. Clear sky is determined by the cloud mask.

Channel cloud forcing: The effect of clouds on the directional radiance in a given channel. Operationally, it is the difference between the observed radiance in a cell and the estimated clear sky radiance. The term *cloud forcing* (unqualified) describes the same difference for broadband fluxes of radiant energy .

Brightness temperature: The temperature at which a black body would emit radiance {RAn} corresponding to a particular satellite channel (see Table 1). For the GOES-VAS channels

   $\{RAn\} = \{c1\}*\{kn\}^3 / (exp(\{c2\}*\{kn\}/\{Ta\}) -1.)$ mW m^-2 steradian^-1 cm^-1,

where {Ta} is the brightness temperature in degrees Kelvin, {kn}, n = 1, ...,12 is the central wavenumber in cm^-1 for channel n (see Table 1), {c1} = 1.19107 10^-5 mW / (m^2 *steradian*cm^-4), and {c2} = 1.43884 K cm.

Cloud level temperature: The brightness temperature that would be associated with the upward radiation from a layer of cloud if the cloud were opaque and the cloud fraction were 100%. It corresponds roughly to the temperature of the air at a level within the cloud above which the emission optical depth is unity. This temperature is assumed to be the same as that

of the surrounding clear sky elsewhere in the cell at the same pressure. Depending on the concentration and absorptive properties of cloud particles, it is typically somewhat warmer than the temperature at the physical cloud top.

Cloud pressure level: The pressure in a cell profile which corresponds to the cloud level temperature.

High Cloud: Cloud with top above 440 mb. *Dense* or *solid* high cloud has effective cloud fraction for a pixel of at least 96% .

Middle Cloud: Cloud with top between 440 mb and 680 mb.

Low Cloud: Cloud with top below 680 mb.

Effective cloud fraction: The fractional area of opaque cloud which, radiating as a black body at the cloud level temperature, would account for the channel 8 cloud forcing for those pixels within the cell which are not reporting the presence of cloud in a higher category. The transmittance for such pixels is unity minus the effective cloud fraction. This relationship between transmittance and effective cloud fraction still holds for thin clouds with no discernible gaps. However, the interpretation in terms of physical cloud fraction is more complex.

Cloud mask: A determination from the satellite data set of which individual pixels within a scene are probably clear sky, as opposed to partially or completely contaminated by cloud. Since clear sky is itself a relative term, this determination is somewhat arbitrary and is subject to errors of both exclusion and inclusion. Systematic errors in the cloud mask will be reflected to some extent in the reported clear sky radiances for a cell, and hence in the inferred channel cloud forcing and in the cloud analysis, and users should be sensitive to this possibility.

Base temperature: An intermediate product in determining the cloud mask, representative of the distribution over different, apparently clear, areas within a cell of the temperature of the underlying land or water surface.

Probable error: There is no absolute standard for measuring the accuracy of the reported variables in this product. However, internal consistency checks do permit assigning levels of confidence to individual inferences and groups of inferences. Such levels of confidence change with varying meteorological conditions and associated cloud fields, and the various fields {...UNC} are a systematic, though somewhat qualitative, measure of such uncertainty.

Granule: A unit of the data product defined for a particular date and nominal time. Physically, for this prototype a granule consists of 3 files with the same file name but extensions .hdf, .odl, and .html respectively.

Product Class: A classification of a granule based upon the VAS channels that were available as input data. Possible values are "A", "S", or "C". "A" implies All (channels 1 - 12); "S" implies Standard (at least 2, 3, 4, 5, 7, 8, 9,10 but not "A") ; and "C" implies Cloud Analysis (at least 3, 4, 5, 8 but not "S" or "A"). In each case, available soundings at the nominal time must cover at least 25 degrees of latitude within the standard grid, otherwise the granule is not reported. The scientific value of granules in each class is discussed in the Introduction to VAS below

Collection: A collection of similar granules together with accompanying descriptive documentation and searchable metadata. The collection metadata and documentation, including this User Guide, are an integral part of the product set.

Product Set: The collection, comprising the contents of the goes_vas_product_set directory, which is submitted to NASA for the management by EOSDIS.

---

3. Introduction to VAS

Key properties of the VAS radiometric channels involved in the dwellsondes are summarized in Table 1. These are closely similar, but not identical, to a subset of the channels on the High Resolution Infrared Sounder (HIRS) instrument on NOAA polar orbiting spacecraft. A full description and evaluation of the instrument is in Montgomery and Uccellini (1985), to which users are referred for authoritative details. The term dwell-sounding arises because the sensor scans lines approximately parallel to circles of latitude as the GOES spacecraft spins in space 100 times every minute. In imaging mode, the scan line is stepped to a new latitude each revolution, whereas in dwell sounding mode lines are repeated to enhance the signal to noise, and to permit sensing through different channel filters. The fraction of total observing time devoted to dwell soundings as opposed to imaging, and hence the coverage of the former, has varied from day to day according to operational requirements. The 11 um window channel, channel 8, and the 6.7 um water vapor channel,

channel 10, are used in both modes.

Table 1: Summary of GOES-VAS channels

n	1	2	3	4	5	6	7	8	9	10	11	12
{kn}	678.7	690.6	701.6	713.6	750.6	2210.0	790.0	895.0	1377.0	1487.0	2250.0	2535.0
Purpose	temp	temp	temp	temp	temp	temp, cloud	moist	surf	moist	moist	temp, cloud	surf
Absorb Gas	CO_2	CO_2	CO_2	CO_2	CO_2	N_2O	H_2O	H_2O	H_2O	H_2O	N_2O, CO_2	H_2O
Other	O_3	O_3	O_3	O_3	O_3	sun	CO_2, dust	CO_2, dust	CO_2	-	sun	sun. dust

n: Channel number
{kn}: Central wavenumber in cm^-1

Channel 8 in the 11 um infrared window has the most favorable ratio of signal to noise, and images the surface radiating temperature and any clouds which may be present. In this product, channel 8 is the principal source of information for the cloud mask which attempts to discriminate clear sky from partially or entirely cloudy pixels, regardless of the altitude above the surface at which the cloud occurs. Such discrimination depends, however, on the negative temperature contrast between cloud top and the underlying surface, and is ineffective for fog or cloud with a very low top. Thus "clear sky" should be interpreted as potentially including these two last conditions. In addition, the "window" is somewhat contaminated by absorption lines for water vapor, so that under genuinely clear sky but humid conditions the channel brightness temperature may be several K lower than the true radiating temperature of the underlying surface. Furthermore, over land and to a lesser extent over water, the surface emissivity may be significantly less than unity. No corrections have been applied to allow for these effects.
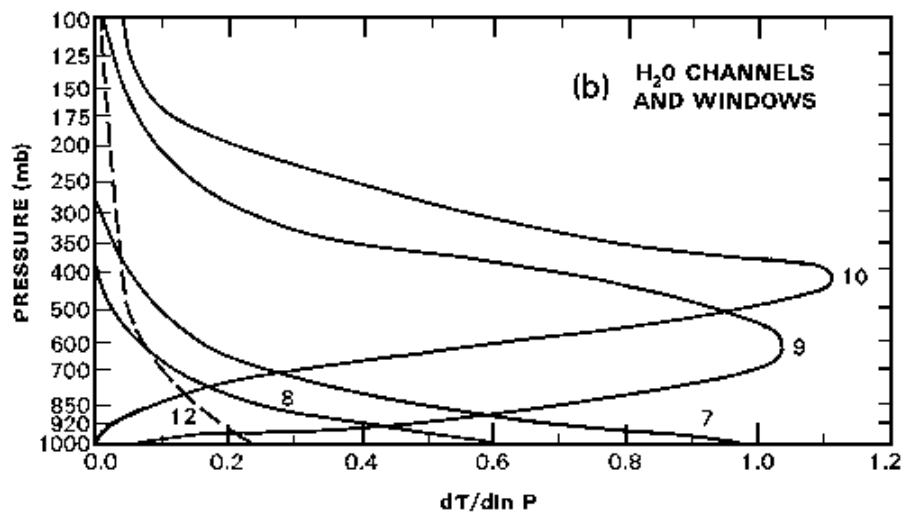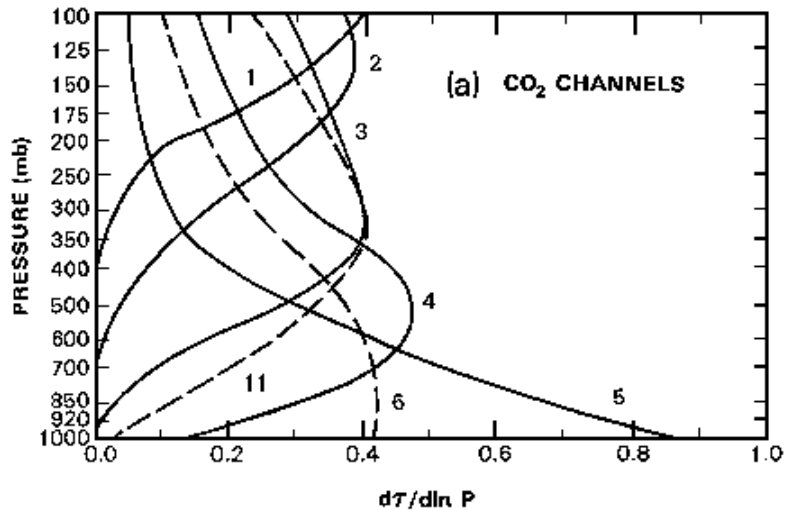
Weighting functions for VAS channels

Channels 1, 2, 3, 4, 5 respond to wavelengths in the center, and successively closer to the fringe, of the 15 um CO2 band. As shown in the Figure, under clear sky conditions they are sensitive to radiation from successively deeper layers of the atmosphere, which are cut off below at a given pressure when the corresponding optical depth (opacity) of the CO2 above exceeds about 3. Each channel records the weighted average of the temperature of their layer. Channel 1 is weighted primarily above 200 mb, while channel 5 averages most of the atmosphere, including substantial radiation from the earth surface. However, any tropospheric cloud within the layer intercepts up-welling radiation from warmer air or from any surface below, replacing it by an intensity corresponding to the black body temperature of a level which is generally close to, though below, the cloud top. High cloud thus affects all channels except (usually) channel 1, whereas low cloud only affects channels 5 and, to a lesser extent channel 4. The ratios among the channels of the difference between a pixel radiance and surrounding clear sky (i.e. the channel cloud forcing) thus can provide a signature of cloud pressure level (Smith, 1968). The magnitude of the channel cloud forcing then indicates the average transmissivity of the pixel, here called the effective cloud faction. Cloud reflectance for these channels may be ignored. Note that "cloud level" here is the inferred level at which the surrounding clear sky temperature would give black body radiances in the required ratios; the effective cloud fraction is then the ratio of the channel cloud forcing to what it would have been had the whole area been covered by opaque cloud. If the cloud properties themselves are independent of channel number, which is a good approximation for channels 1-5 but less satisfactory for the remainder, the ratios should be independent of which pair of channels are used to determine them. Significant failure of this requirement indicates either a gross error in the estimated clear sky radiances or, more probably, multiple layers of cloud visible to the satellite within a pixel. In the latter case, the pixel estimate of cloud pressure level will be intermediate, and the reported standard deviation among pixels about the cell average will be increased accordingly.

Channels 9, 10 are located in the main absorption band for water vapor, in order of increasing opacity. Channel 7 includes some water vapor lines within a broader window. Under clear sky conditions, and after allowance in channel 9 for possible transmission from the underlying surface, each channel indicates primarily the relative humidity averaged over a broad range of pressure, with a secondary sensitivity to temperature. This range depends in known ways on the humidity itself, the channel number, and the satellite zenith angle. Precise interpretation requires a sensitivity calculation from a candidate model profile. However, for channel 10 the range almost always lies in the upper troposphere with a cut-off above the earth surface, increasing in height for humid conditions or large zenith angle (Soden & Bretherton, 1993). For channel 9 the range is lower in the atmosphere. The difference in brightness temperature between channels 7 and 8 is normally an indication of total moisture in the lowest levels. With 100% cloud cover the cloud level temperature and pressure replace those of the underlying surface. Thus, if cloud level temperature and effective cloud fraction are well known from other channels, the estimated cloud forcing in these channels may be interpreted in terms of either (1) more detail about the cell wide vertical profile of humidity above the level of the cloud top, or of (2) an association of cloud elements with local increases within the cell of vertically integrated humidity, or of (3) some combination of both possibilities. The choice is left to the user.

Channels 6, 11, 12 are in the 4 um window and are contaminated by reflection of solar radiation during the day. At night for clear sky they provide independent estimates of the radiating temperature of the middle troposphere, lower troposphere, and underlying surface respectively. The window provided by channel 12 is even clearer than channel 8. However, because of the lower emissivity of channel 12 from continuous snow or ice particles versus water or soil, a comparison enables detection of snow cover over land, and or cirrus cloud aloft. The primary effects of cloud at different altitudes are similar to those for the CO_2 and water vapor channels. The differences of surface emissivities are smaller than the estimated errors in cloud emissivities from other sources.

The VAS instrument scans approximately along circles of latitude 100 times each minute, with successive selections of channel from a movable filter wheel. In dwellsonde mode, each scan is normally repeated according to an operational schedule to improve the signal to noise, but may be omitted to conserve observing time. The line of pixels is stepped in latitude after completing a pass through the required channels. Channels 8 11 are normally scanned continuously in latitude, but for the remaining channels data swaths nominally 32 km wide alternate with unsampled areas. Coverage of the continental United States normally took two observing periods of some 15 minutes each, separated by time devoted to global images, and the VAS schedule was often modified during periods of severe weather or special requirements.

These operational characteristics show in the data products in various ways. Data coverage extends to less high latitudes around the subsatellite longitude. The range of latitudes actually covered may vary from hour to hour and day to day, though the products are presented only when there is reasonably complete coverage in the central regions of the reporting grid. Product classes reflect the choices of channel priority at different times, and hence imply some grids that are largely or entirely missing data. The number of pixels within a reporting grid area decreases with increasing satellite zenith angle. The channel 8 cloud forcing based on VAS pixels may not be precisely equivalent to that based on channel 8 itself.

It would be presumptuous to ascribe in advance relative values to all the scientific analyses that will be done when data of this type is widely accessible to the research community. Nevertheless, for planning purposes it is necessary to be aware of the opportunities and limitations that are inherent in the data which is actually available. Thus three possible product classes have been defined for the time slice represented by an individual granule. For Product Class A, all the channels discussed above are recorded for the most of the standard grid, enabling studies which relate simulated temperature, ozone, and moisture soundings to observed radiances, including a full cloud analysis, as well as supplementary information by night from the three 3-4 um channels. For the Standard Product Class S, most of the studies for Class A are still appropriate, except the stratospheric temperature is less well determined (channel 1) and the specialist information from the 3-4 um channels may not be present. For Product Class C, it should still be possible to provide a cloud analysis suitable for inclusion in a cloud climatology, but correlative information on water vapor and stratospheric temperature may be absent.

4. Calibration

Calibration procedures for VAS data were applied in software before distribution to users or entry into the GOES archive, using pre-launch calibration tables specific to the satellite concerned. These calibrations are thus incorporated through the input data into this data product. The procedures are discussed in Montgomery and Uccelini (1985) and will not be repeated here. If the prelaunch calibrations of stable devices, such as channel filters or the thermistors measuring the temperature of the on board warm body, are inaccurate or have drifted slowly with time, the primary impact is likely to be

a small change in the absolute magnitude of the observed radiances and possibly in the ratios among channels, while relative values within a grid or over a few days or weeks are unaffected. Should at some future date such innacuracies be reliably determined from intercomparisons among satellites or by other means, the present products can be corrected after the fact by simple proportional multipliers.

The only known departure from the archived data is the use of improved navigational parameters to specify the position and pointing direction of the satellite. These parameters are derived by McIDAS software in routine poeration at SSEC, using known landmarks in the imagery data stream, coupled to predictive modeling of the satellite orbit and spin dynamics. They routinely yield locations for individual pixels to within 2.5 km. Such accuracy is fully adequate for the purposes of this data product. The temperature clamp on the telescope optics failed on GOES-6. However the telescope temperatures were monitered, and the line by line calibration tables in the archive do enable reliable results to be derived during this period (Menzel, 1995). This epoch does not overlap with the present sample.

The 8 visible sensors on VISSR operated in parallel but were liable to drift independently on all instruments. The resulting striping of the visible image was corrected by operator adjustments of the gain prior to data distribution. The records of such adjustments have been lost. Though the data reported here has been averaged over all 8 channels, the effects of adjustment may not all cancel out, and there remains some uncertainty in the absolute calibration of the visible data. Though it probably indicates correctly the day to day and most month to month variations, longer term changes, particularly between satellites, are not trustworthy. Comparison of sample scenes with AVHRR data by Frouin and Simpson, 1995 gives a general indication of the data quality and approximate absolute calibrations. For the algorithmic relationship between VISSR Counts (CNi) used by Frouin and Simpson and the GVAR radiances reported here, see the section on Visible Radiances below.

---

5. VAS Archive

Motivation

This sample product set is a prototype for a similar product covering the entire useful duration of the GOES/VAS observations. In order to evaluate the scientific benefit that might accrue from nearly 20 years of record, a study is reported here to ascertain what raw data is actually available. Because VAS dwell soundings were defined as research rather than operations, the observational schedule changed unpredictably according to the needs of particular campaigns, or to better define the environment of outbreaks of severe storms, or to allow more frequent images of huricanes, etc. In addition, a few observations were lost because of failure in ground communications or in the archive itself.

The Archive Inventory

An inventory of the GOES archive has been maintained throughout its history. The inventory software for the pre-GOES-8 archive was developed and implemented in 1986. It covers the full period of the archive from February 1978 to October 1997. It was designed as an inventory by exception, that is, the primary database is the published data schedule for each day and exceptions to the schedule or data quality which were noted . These notes were made operators who, among other duties, were manually scanning images of the data as they were acquired. Exceptions were frequently ascribed to all data for a day, based on manual observation without verification. The result is an inventory that sometimes overstates the problems when observed, and missed problems if they weren't observed. Fully reliable information requires accessing and scanning every individual tape in the archive, which for reasons of cost has so far never been done.

The Process of Inventory Analysis

The original inventory was first converted to an inventory by inclusion, that is, new files were created that contained a record for every image or dwell sounding that was scheduled. These files were reduced by eliminating all records for images and for soundings that were not recorded on archive tape. In the period 6 March 1982 through 11 January 1996, 137,629 soundings were scheduled and presumably recorded.

The second step in the analysis determined the number of soundings that are estimated to be of good quality. The records with comments that indicated a reduced quality or that the satellite was drifting to a new station were flagged as unacceptable. The latter rule was invoked because data taken during a period when the satellite was drifting from one permanent station to another usually had significantly reduced navigation (geo-location) accuracy. While it is likely that renavigation will recover these data to an acceptable level, it is indeterminate at this time. 116,404 soundings were

acceptable based on these criteria.

Step three was the identification of the soundings that fell into one of the three product classes. These product classes are determined by the channels that were available in a particular sounding. Specifically, the "Cloud Analysis" class required that channels 3, 4, 5, 7 and 8 be present. "Standard" further required that channels 2,3,4,5,7,8,9,and 10 be present. And "All" required that all 12 channels be present. These classifications flagged 98,281 soundings as acceptable. Very few (30) "All" soundings were taken prior to the start of mode AAA operations in May 1987. During mode AAA operations till October 1997, 31,322 sounding of class "All" were recorded. Class "Standard" flagged 45,402 distributed uniformly without the archive period with the exception of a lull in 1983 and 1984 and a shift to "All" class soundings between 1987 and 1990. "Cloud Analysis" class soundings were used extensively starting in the latter half of 1992 to the end of the archive period. In all, 21,527 "Cloud Analysis" were recorded.

Normal operation of the satellites required two images of 20 minutes duration in each hour. Therefore it was usually necessary to use two soundings of 10 minutes each to achieve a complete sounding of the continental United States. These soundings were sandwiched between the half-hourly images. The final analysis step determined which soundings met the minimum coverage criteria (maximum latitude > 45 deg. N to minimum latitude < 30 deg. N with at least 20 degrees of latitude range) adequately within the hour window.

The result is the estimated number of granules (46,755) that will be included in the product set. Of these, 8,023 are in class "All", 17,409 of class "Standard" and 21,323 fell into class "Cloud Analysis". No distinction was made for the satellite location (East, West or Central) as the same products will be generated, regardless. However, western grid points will be missing from the products created from GOES-East data, and likewise, eastern grid points will be missing from GOES-West derived products. It is intended to process a number of additional granules which contain potentially useful data but for which the dwell sonde coverage does not reach 20 degrees of latitude.

Summary of Results

Three distinct trends were obvious from the analysis. The frequency of soundings was significantly improved once the VAS instrument became operational in May 1987. The peak for frequent "All" or "Standard" soundings was in the years, 1987 through 1991. The peak for "Cloud Analysis" sounding was from 1991 to the end of operations.

The sporadic nature of sounding operations during 1983 and 1984 hinder a thorough climatological analysis of the diurnal cycle for the full period of the archive, but from 1985 on, the inventory analysis identified an average of 12.2 acceptable soundings per day. This should provide an adequate basis for climatological analysis.

6. Algorithms

6.1 VAS Observed Radiances

The only product variables which were directly observed are the channel radiances {RAn}, n=1, ..., 12, and the visible radiance {VISIBLE}. For each sounding, individual pixels were geolocated using standard McIDAS navigational software and assigned to the appropriate cell. The reported value is the unweighted average over the pixels within each cell.

6.2 Visible Radiances

The visible radiances are included here as the grid {VISIBLE} to enable day to day analysis of the solar radiation component of the land surface energy budget, complementing the more detailed information in the infrared. The absolute calibration of this data is suspect, (Frouin and Simpson, 1995) but short term relative variations should be well mapped.

The data are stored in the archive as raw VISSR 8-bit brightness (the units referred to by Frouin and Simpson as counts), but are reported here as radiances analogous to the procedure used for GVAR, the successor to the VISSR/VAS instrument on GOES-8 and GOES-9. This is an invariant, reversible transformation, added for convenience in physical interpretation. No radiance units were defined for VISSR. First, as detailed by Chesters, the VISSR brightness is converted to a six-bit value (i.e., divided by 4) and then squared to become a GVAR count, and averaged over each cell. The GVAR count is converted to a radiance (R) per unit wavelength (W/(m^2-sr-um)) using the formula R = (M*counts + b), where the counts are computed above from the VISSR six-bit values. The values of (M, b), which are constants derived experimentally in the factory, are: M=0.5507281 and b=-15.3300. These particular values are valid for detector 8

on GOES 8 -- they are close to the values for each filter on GOES 8 and GOES 9.

After the conversion from VISSR greyscale to GVAR radiance, we noted that for dark images, a negative radiance could result. For example, a 8-bit value of 8 for a VISSR greyscale gives a GVAR count of about 1, yielding a Radiance of -14.76. To avoid this unphysical result, all negative radiances were set to 0. The negative values should not have been a surpise, because, as described by Frouin, zero drift on individual visible sensors is indicated by the non zero brightness of outer space space count at the end of each scan line, resulting in a positive bias in all inferred radiances. On average, the offset b should account for this bias, but for some individual cells it can lead to negative net values. Following Frouin, a more accurate approach to this issue would probably have been to subtract for each observation time the square of the space brightness, averaged over all 8 visible sensors, but that would have involved more extensive data processing and was not done. According to NESDIS tables, GVAR counts of less than 29 have albedos of 0. The VISSR- to-GVAR count formula will yield a value of 29 when the VISSR count is around 41. Thus, VISSR greyscale brightness values of 41 or less yielded equivalent GVAR radiances of 0. Note that this truncated linear conversion from VISSR brightness squared to GVAR radiance makes no allowance for long term sensor drift, or calibration differences among VISSR satellites. Users who care about long-term stability should make empirical adjustments to fit data from aircraft or other satellites, as described by Frouin and Simpson.

6.3 Cloud Mask

A critical step in the algorithms applied to the observed radiances is the cloud mask, which determines which pixels are deemed to be cloud free (i.e. clear sky). Brightness temperatures for channel 8 in the infra red window are used in a multi-step decision process, involving a inference of a base surface radiating temperature {TB} for each cell, and a subsequent comparison with that base of the brightness temperature for each pixel within the cell. If the contrast exceeds a certain cloud mask threshold, the pixel is deemed at least partially cloudy, otherwise it is deemed clear. This operational definition of clear sky conditions is based upon a conceptual model in which:

(1) cloud tops are presumed always to be colder than the land or water surface beneath. Note that in the exceptional case of a strong night-time temperature inversion this may not be true;

(2) there are small areas within an individual cell or within nearby cells which are free of cloud;

(3) the surface radiating temperature reflects primarily the synoptic scale temperature in the deep atmosphere above, upon which is superimposed a diurnal cycle which may be substantial;

(4) for land under clear sky, terrain altitude, surface heat capacity, and vegetation cover cause an additional variability in the surface radiating temperature from pixel to pixel within a cell, but its typical magnitude may be determined empirically; and

(5) over water, pixel to pixel variability within a cell which significantly exceeds instrument noise is most probably due to contamination by cloud.

This model is applied in a 4 step process, in which each threshold which controls a decision in the algorithm at the cell level is in principle decomposed into a numerical tuning coefficient times an observable variability measure which is averaged over many different but similar cells. A key parameter in assessing the appropriateness of these thresholds is the average variability {TC8ySD} of brightness temperature among clear sky pixels within a typical cell, determined as a statistic during production separately over water (y=WATER) and over land (y=LAND). Though this decomposition does not eliminate the need for tuning based upon the judgement of a scientist who is experienced at interpreting individual images, the conceptual model greatly restricts the range of plausible values for the tunable coefficient and provides a theoretical framework for comparing alternative judgements and systematic analysis of their consequences. The process is initiated using thresholds and acceptable ranges for {TB} which are based solely on the judgement and prior knowledge of a scientist examining sample cases, but in principle these thresholds may be reformulated after the first pass into the product of dimensionless coefficients and variability measures. For this small data sample, the variability measures are objectively determined ,but implementation of the reformulation from the intial statement in physical units into dimensionless terms has been left until a science evaluation of its implications can be completed.

Step 1: The determination of a provisional value for {TB} involves identifying with a reasonably high degree of confidence any gaps among the clouds, from which the channel 8 radiance is taken as representative of the whole cell. For the remaining mixed cells the value of {TB} for each type of surface is borrowed in Step 3 from adjacent areas

following the primary analysis.   This is not known in advance, but is determined estimated during production from observations of channel 8 radiance {RC8p}under unequivocally clear sky conditions, and averaged over many different cells covering a continent and season.

Specifically, members of a 16 x 16 block of pixels centered on the middle of each cell are grouped into overlapping 2×2 arrays. For each such 2 x 2 array, the mean {T8a} and standard deviation {T8aSD} of the channel 8 brightness temperature over the array are calculated. From this set, all those arrays are rejected for which the standard deviation {T8aSD} exceeds a threshold which was here kept constant but may be written in the form {Ca}*{TC8ySD}. {TC8ySD} is a continent- or ocean-wide statistic averaged over many cells and throughout a season, which measures the variability of individual pixels about the cell mean under clear sky conditions. It is a function of time of day and land/water classification, and is derived from the corresponding measure {RC8ySD} for radiance, for which the daily contributions to a seasonal average are reported in the granule metadata. Application of {TC8ySD} to neighboring pixels within an 2x2 array supposes that the total within cell variability is dominated, or at least proportional, to that on a spatial scale less than 16 km. {Ca} is a dimensionless, tunable coefficient that should theoretically lie in the approximate range [1.0, 2.0]. Based upon expert judgement, suitable values for the threshold (kept constant throughout this 7 day sample) are

Land: {Ca}*{TC8LANDSD} = 0.45 K
Water: {Ca}*{TC8WATERSD} = 0.3 K

This requirement for spatial coherence (Coakley and Bretherton) excludes most situations with significant cloud, particularly over water, though it leaves open the possibility of either clear sky or 100% cloud over the array at a uniform cloud level temperature. After rejection of those 2x2 arrays with excessive internal variability, the brightness temperatures of the remaining {Na} arrays associated with a cell form a distribution. These are sorted by brightness, and the coldest member of the warmest 20% is selected, with brightness {T8a20}. All values colder than {T8a20} - {DT8y}are then discarded, where {DT8y} is a fixed threshold. Because of the discard, {T8a20} may no longer be in the warmest 20%, so the selection is repeated until no further discards take place. The remaining {Nb} values of {T8a} are deemed to be probably cloud free, and hence have an internal variability comensurate with {TC8ySD}. {Cb} is another positive, tunable coefficient, of which the function is to eliminate all the arrays which, based upon prior expectation of the within-cell clear sky variability {TC8ySD}, are too cold to be reasonably likely to be cloud free.   Expert judgement for the threshold {DT8y} thoughout this 7-day sample is

{DT8LAND} = {CbLAND*{TC8LANDSD} = 2.5 K
{DT8WATER} = {CbWATER}*{TC8WATERSD} = 2.5 K

In theory, plausible values for this threshold should be related to the pixel variability {TC8ySD} about the cell mean under circumstances when the sky was clear, and the coefficients {Cby} should be the same for land and water. However evaluating the consequences of such a change throughout different seasons would require a more extensive sample than the 7 days here available, so the values of {Cby} have not been separately tuned.

If {Nb} is less than {Nbase}=20 , the cell is deemed probably contaminated, and estimates of clear sky conditions are "borrowed" from surrounding cells according to Step 3 . If {Nb} is greater than or equal to {NBase}, the uncertainty in {TBy} by this measure is very low, but there remains the possibility of widespread 100% cloud cover with uniform cloud level temperature, e.g. a dense cumulonimbus anvil or marine stratus. Step 2 is designed to circumscribe this possibility.

Step 2 is a buddy check with other cells for which there is confidence in the assigned value of {TBy}. The values {TBprev} and {TBsubs} for the same location at the same time of day on the previous and subsequent days are examined. The test passes if

 EITHER   {TBprev} is flagged "confident" and abs(TBy-TBprev} < {TBuddy}
 OR   {TBsubs} is flagged "confident" and abs(TBy-TBsubs} < {TBuddy}

In this case, {TBy} is marked "confident" and Step 3 is omitted. Otherwise, the test fails, the value of {TBy} is replaced temporarily by "missing" and Step 3 is followed.

{TBuddy} is another threshold, set throughout this sample at 2.5 K for both land and water. If the test is passed, the probability is low that the same cell was overcast with essentially the same cloud level temperature on two successive days. Then the quantity

$$\{TByUNC\} = \min(abs(\{TBy\}\text{-}\{Tbprev\}),abs(\{TBy\}\text{-}\{TBsubs\}))$$

is used to determine the day to day changes (up to 2.5K) to be expected at that location and season in the value of {TBy}, setting a range on the cloud mask thresholds that are used throughout the subsequent analysis. If the test fails, it merely means that confidence is lacking, and borrowing the value of {TBy} and its uncertainties from surrounding areas is preferred.

Step 3. If, for a particular cell, the selection yielded fewer than {Nbase} pixels or the buddy check "failed", an estimate for {TBy} must be interpolated from cells of the same Land/Water type y which have been flagged "confident". Centered on the cell (0) concerned, a square of side 2*{NS}+1 degrees of latitude and longitude is searched for such cells, starting with {NS} = 1 and increasing {NS} as required till {NI} >= 8 have been found. Weights {W(I)} are then assigned to each I = 1,..., {NI}, inversely proportional to the square of the distance:

$$\{W(I)\} = K/((Long(I) - Long(0))^2 + (Lat(I) - Lat(0))^2),$$

where the constant K is chosen so that

$$SUM\_I = 1 \text{ to } \{NI\} \quad \{W(I)\} \quad = 1.$$

Then

$$\{TBy(0)\} = SUM\_I = 1 \text{ to } \{NI\} \quad \{W(I)\}*\{TBy(I)\}.$$

Note that the interpolated {TBy} are NOT flagged as "confident", so the order in which cells of the grid are processed is irrelevant. The process ultimately fills all the cells of the grid with and estimate for both y=LAND and y=WATER. For those cells for which {LANDFRACTION} = 0 % or 1 % only the relevant values are used. For cells of mixed type, separate cloud masks are maintained, and the results for each type are prorated at the end. The uncertainty {TByUNC} of interpolated values is discussed in the section on Uncertainty below.

Step 4 is the cloud mask itself. According to the standard mask, an individual pixel in any cell is regarded as cloud free if the base temperature {TBy} exceeds the observed channel 8 brightness temperature {T8p} by less than a threshold {DT8y} = 2.5 K. Because the rationale is similar, this threshold is the identical to already invoked when arriving at direct measurements of {TB}. Otherwise the pixel is regarded as partially cloudy. This mask is applied to every data pixel individually.

The general approach to this cloud mask is based upon that of ISCCP and the MODIS Science Team, though the details are different. A novel feature here is the explicit linking of thresholds to monitored variability of the quantities used in the estimation of the cloud mask. The cutoffs are still to some extent tuned by adjusting the dimensionless multplier according to the judgement of a scientist looking at a range of special cases ("where does it fail?"), but the tuning process is made more explicit and is more accessible to retrospective statistical studies of the sensitivity of the outcome to reasonable adjustments.

6.4 Clear Sky Radiances

If the value of the base temperature was measured and passed the buddy check, the estimated clear sky radiances {RCn} are then the average for each channel {n}over cloud free pixels in the cell . Note that these averages are with respect to pixel radiances, not brightness temperatures. Because of the scanning strategy of the instrument, data pixels for some VAS channels cover only about half the area within each cell, whereas the cloud mask from channel 8 covers the entirety. Where there was not confidence in the determination of cloud free conditions, the value of {TB} has been interpolated from surrounding cells, along with an interpolated uncertainty. The reported {RCn} and {RCnUNC} are then similarly interpolated using the same weights {W(I)}.   In addition, standard deviations {RCnSD} among clear sky pixels within a cell are computed for all the "confident" cells.   These are reported for each observation time as a statistical distribution over the entire grid area in each LAND, WATER category (ignoring mixed cells). Collectively, these distributions indicate the statistical effects in each channel of instrument noise, of sub grid scale clear sky atmospheric variability, and of variations in the underlying surface, and the channel 8 mean value relates to appropriate thresholds for the cloud mask.

6.5 Atmospheric Profiles

The cloud analysis requires that for each cell a black body radiance be assignable to a hypothetical opaque cloud at arbitrary cloud pressure level. This relationship is used to match the ratios of the observed channel cloud forcing among the CO_2 channels with an effective cloud fraction and cloud pressure level. It implies a grid-scale relationship between temperature and pressure known as a sounding, and considerable energy has been devoted to retrieving such soundings from satellite data for the purpose of weather forecasts, including extensive requirements for supplemental radiosonde data and numerical assimlation models. However, the final precision of cloud analysis undertaken here does not warrant more than a simple algorithm which modifies a seasonally averaged sounding derived from radiosondes using local information from the estimated clear sky radiances. Such statistical retrievals inevitably miss details of the actual sounding, but their information content can easily be determined from sensitivity studies, and the errors involved in using them for the cloud analysis seem small compared to those intrinsic to the cloud model itself. The FORTRAN Code for the retrieval routine used here was supplied by H. A. Woolf , and is provided in related documentation.

Also provided is code to perform the forward calculation from a complete atmospheric profile to the directionsal
 radiances which should be seen by a satellite under clear sky conditions.   This may be useful in checking how well the temperature and moisture profiles in a GCM are consistent with the satellite radiances actually observed.

6.6 Cloud Analysis

Cloud pressure levels

The algorithm for determining cloud pressure level was originally proposed by Smith (1968), and has been applied to GOES-VAS data by Wylie and Menzel under the name of "CO2 Slicing". It reports only the uppermost cloud that is exposed to the observing satellite. Inferences or hypotheses about any hidden clouds beneath are left to the user. It assumes that each pixel is partially covered by at most one layer of cloud, for which the black body temperature is equal to that of the surrounding air at cloud pressure level. This cloud is assumed to absorb infra red radiation, but not to reflect it significantly in the channels under consideration. The layer may be regarded as either (1) opaque, but covering only a fraction {CFp} of the area of the pixel, or (2) covering the whole pixel, with partial transmissivity 1 - {CFp}. However, the two parameters, black body temperature {Tp} and effective cloud fraction {CFp}, are all the information that is available on a pixel scale. Cases (1) and (2) are thus extremes of a continuum of indistinguishable interpretations of the observations. Even with this caveat, the model is at best an approximation to real cloud layers, and under some circumstances such as penetrating cumulus, or broken high cloud above low, it may be a poor approximation. The algorithm then generally reports an intermediate value for {Tp}. The neglect of reflection may also involve errors of several percent in the relation between cloud fraction and transmissivity.

Aggregation of pixel level cloud analysis to cell wide statistics is essential to reduce the data volume and complexity. However, given the non-linearities and uncertainties in the cloud analysis, is not obvious how to retain the most useful real information about the clouds actually present in the cell. The aggregation procedure used here is a compromise, guided by some general consideration of the target user groups for this dataset, coupled to some judgements about the systematic impact of idealizations in the cloud model, and tempered by the conflict between full reporting and the need for simplicity and transparency.

There are two modifications here of the Wylie and Menzel approach . The first is an improved cloud mask which makes no assumptions about the diurnal cycle of surface temperature. The second is a different method for establishing a cell-wide table of temperature as a function of pressure which is needed to establish the radiative effects of hypothetical clouds at different levels. Here the profile is a statistical retrieval from the reported clear sky radiances in the CO_2 channels in the cell itself, rather than a more detailed first guess derived from a numerical forecast model. The statistical retrieval is almost certainly a simplified representation of the true atmospheric profile in that cell, but the errors introduced into the cloud analysis are small compared to others associated with the cloudiness model itself. More important, because of the internal consistency of this approach, it is much easier to track any systematic errors introduced by this step into the final product. In addition, the algorithm for calculating forward radiances is invoked explicitly.

The algorithm for cloud pressure level is applied to each pixel individually within a background determined by cell wide parameters. It invokes as pixel input the observed radiances {Rn}for the CO_2 channels 2-5 and the window channel 8 and the cloud mask. As cell input it uses the estimated clear sky radiances {RCn}, and, in tabular form, the retrieval {Ti} for the cell at tabular pressures {Pi}, i = 1, ..., 40 , derived from the clear sky radiances by the FORTRAN subroutine VASRTW

Step 1 For the cell background, a table of hypothetical radiances {RHn} is calculated using the Woolf VASFRWD Model code Version 1 for each of channels n = 3-5 at the appropriate satellite zenith angle, supposing 100% black cloud at tabular cloud pressure levels {Pi} i=1, ...,40 with no cloud above, using the retrieved temperatures {Ti} as the cloud temperature. Comparable values for clear sky {RCn} are obtained by setting the cloud pressure level and temperature equal to the cell mean surface pressure {PB}, and base temperature {TB} respectively. The difference {RHn} - {RCn} is then the hypothetical channel cloud forcing for 100% opaque cloud at the tabular level.

Steps 2-7 are for each pixel within the cell.

Step 2. If the cloud mask indicates "clear", it remains clear and no further processing is required.

Step 3 This is the default, when the CO2 slicing fails because a measurable channel cloud forcing appears within the pixel in less than 2 of channels 2-5. {Pp} is then assigned the pressure value corresponding to the interpolation of the channel 8 pixel brightness temperature {T} into the table {Ti}. This assignment automatically assumes that {CFp} = 1. In the event that pixel is only partially covered with cloud, the cloud pressure level will be systematically over estimated. This default assignment is over-ridden if step 6 succeeds.

Step 4 For each channel, if the channel cloud forcing {Rn} - {RCn} in negative or does not exceed 0.05*{RCn} no further processing is done for that channel. This threshold screens out small values of the denominator in step 5

Step 5 For each pair (n,m), n > m of the remaining CO2 channels 3-5, the ratios {fnm}are computed from

$$\{fnm\} = (\{Rn\}-\{RCn\})/(\{Rm\}-\{RCm\})$$

If no such pair exists, revert to the default. There are 3 possible ratios that can be calculated from 3 CO2 channels. Any pair may be used in the final estimate.

Step 6 For each such pair (n,m) the cloud pressure level {Pnm} that would give the same ratio {fnm} is determined from the table {RHi} of hypothetical radiances for the cell, by interpolating linearly between tabular values {Pi} to obtain an exact match. Let {RHn} and {RHm} be the radiances that correspond to that exact match. IF {fmn} is out of bounds or no match can be found with {Pnm} lying between the pressure at the Earth surface (i.e. the value of {P}corresponding to {TB}) and the pressure at the tropopause (i.e. the value of {P} corresponding the the minimum value of{T} in the profile), THEN revert to the default.

Step 7 According to the cloud model used for this analysis, the observed radiances in channels 2-5 should all be consistent within expected error with a single value of cloud pressure level {Pp} associated with a single value {CFp} of the cloud fraction. Each value {Pmn} provides an estimate for {Pp}, and

$$\{CFnm\} = (\{Rn\}-\{RCn\})/(\{RHn\} - \{RCn\})$$

is the corresponding estimate for the cloud fraction for the pixel. Each such estimate uses only the two CO2 channels m, n, and ignores any information in the remaining channels. Where there is a choice among such estimates, it is decided by estimating a total fit residual W for each available pair ({Pnm},{CFnm}), and picking that which has the smallest residual. Given any valid values for ({Pp},{CFp}), interpolation into the tables yields values of {RHk} for all the CO2 channels k = 3,..,5.   k= 8 provides an additional term in the total fit. The residual is then

$$W = SUM\_k \ (\{Rk\} - \{RHk\})^2 / \{RCkSD\}^2,$$

though for this sample the channels were all given equal weight in terms of brightness temperature ({RCkSD}/{Rk} independent of k) . Because of the way {Pmn} and {CFmn} were are calculated, the terms in this sum corresponding to k = m and k = n are automatically zero, but W is an appropriate measure of the overall consistency. The pair nm which gives the smallest residual, or channel 8 with {CF8}=1, is the one chosen.

Note that channel 8 is not used in the pairs of the CO2 slicing algorithm. This is because channels 2-5 are closely spaced in wavenumber, and the interactions of cloud particles with radiation are essentially the same in each channel, reducing the number of unknowns which have to be determined on a pixel basis. However, including the longer wavelengths associated with channel 8 would cause the cloud analysis to be sensitive to particle size. Given the present limitations on cloud climatologies and microphysical understanding, it seems preferable to reduce complexity by aggregating to the cell level or

higher in such a way that users of the data set can themselves interpret any systematic differences between channel 8 observations and what would be expected from the cloud analysis.

Step 8 Aggregation

Once {Pp} has been determined for a pixel in this manner, the pixel is categorized as

High: {Pp} =< 440 mb
Middle: 440 mb < {Pp} =< 680 mb
Low: 680 mb < {Pp}

For each class, the relevant pixels are aggregated separately. The average cloud pressure levels are the unweighted mean over pixels reporting cloud in that category.

IF {CFp} is everywhere zero for class z,

THEN {CFz} = 0 and both {Pz} and {PxSD} are reported missing,

ELSE
{Pz} = (SUM_pixels in z {Pp}) / (SUM_pixels_in_z)
{PzSD} = sqrt ((SUM_pixels in z ({Pp} - {Pz})^2) / (SUM_pixels_in_z))

Effective cloud fraction

There is no way of aggregating fractional cloud cover at different heights into a cell average which will accurately reproduce the channel cloud forcing in all the CO_2 channels simultaneously. The division into LOW, MIDDLE and HIGH categories, each separately reporting both cloud level temperature and cloud fraction is a compromise between the simple classifications typical of GCM models on a sub-grid scale and the complexities of representing the statistics of real cloud fields. For simplicity, the effective cloud fractions are all related to the cloud forcing in channel 8. Apart from determining the cloud pressure level, the other VAS channels are ignored in the cloud analysis itself. Given the best estimate of cloud level pressure for each individual pixel in a cell, the unweighted arithmetic average is reported as the cell mean {Pz}, z= LOW/MIDDLE/HIGH .

The rules for aggregating effective cloud fraction are somewhat more complicated because of obscuration by higher layers of cloud. Each class reports a non-overlapping area and the transmissivity from the underlying surface to space of that area

The area for each class is that of those pixels which potentially could report cloud of that class, after premption by higher levels, i.e.

{NOBSCLEAR} = Number of VAS pixels in cell reporting clear sky
{NOBSLOW} = {NOBSCLEAR} + Number reporting LOW cloud
{NOBSMIDDLE}= {NOBSLOW} + Number reporting MIDDLE cloud
{NOBSHIGH} = {NOBSMIDDLE } + Number reporting HIGH cloud

Note that all VAS pixels report in one of these classes, so

{NOBSHIGH} = {NOBSTOTAL}

and the remaing areas cloud equally well have been defined by subtraction. Then

{CFz}= (SUM_pixels_in_z {CFp}*{Pp}) / {NOBSz}

and the transmissivity of the area {NOBSz} from the underlying surface to space is 1 - {CFz}.

Uncertainty

Uncertainty pervades the products reported in this data set. For example, when skies are generally clear, clear sky radiances and cloud forcing may easily be estimated with determinable precision. However, widespread multiple cloud

layers make both the clear sky radiances and cloud analyses less reliable. In addition, it is almost impossible to detect widespread ground fog at night from a passive infra-red sensor in space, but failure to do so affects only the reported incidence of very low cloud. In general, the reported measures of probable error for each cell do increase with declining confidence, but should not be interpreted as defining independent random noise.

The combination of sampling and instrument noise associated with individual pixels in a cell is measured by the variables {RCnySD}, n=1,...,12. This may be compared to typical signals in the corresponding channel radiances. The corresponding measure for channel 8 is for convenience reported also as a brightness temperature {TC8ySD}. The statistical distributions of these variables are reported in the html file for each granule.

As explained above under Cloud Mask, for a cell marked "confident" the uncertainty {TByUNC} in the base temperature {TBy} is estimated from the change in {TBy} from the previous and subsequent day at the same time as part of the Buddy Check.

Where insufficient pixels within a cell are judged to be clear, the base temperature {TBy}for that cell must be estimated by interpolation, or "borrowing" from neighboring cells, which inevitably substantially increases its uncertainty by an amount which increases with the distance

$$\{D(I)\} = \text{SQRT}((\text{Long}(I) - \text{Long}(0))^2 + (\text{Lat}(I) - \text{Lat}(0))^2$$

over which borrowing is taking place. For an individual cell the RMS error may in principle be estimated by the structure function

$$S\{D\} = \text{SQRT}(\text{average } (TBy(I)-TBy(0))^2\},$$

where the average is taken over an ensemble of many cells and meteorological situations for pairs of which the members are flagged confident. True interpolation from {NI} surrounding cells should improve on this probable error, but in the common case of one sided extrapolation S(D) is a better estimate. For this pilot sample, S(D) is not known *a priori* so in every case of borrowing, the simple expression

$$\text{TByUNC}(0) = 2.0*\{NS\} \text{ K}$$

has been used instead, pending a more refined formulation based on *post hoc* analysis of the entire sample. Likewise, for other channels the preliminary formulation is

$$\{RCnUNC\} = 0.02* \{NS\}*\{RCn\}.$$

Though crude, these measures provide a first order indicator of diminished confidence in the companion fields due to the meteorological circumstances surrounding their estimation .

To clarify the sensitivities involved in the cloud mask, the cloud analysis have also been run with a second, colder, threshold, in which {DT8y} is replaced by {DT8y} - {TByUNC}. This second threshold always reports the same or smaller amounts of cloud, but reflects a realistic range of uncertainty in {TBy} itself. Thus for z = HIGH, MIDDLE or LOW, {CFzUNC} is the difference between the values of {CFz} computed with these thresholds. The differences are substantial for low cloud, but remain relatively small for high cloud, because of the large contrast between such cloud and the underlying surface.

It should be noted that some of the apparent structure of the reported fields for {TByUNC} and {RCnUNC} is an artefact of the method of calculation. In particular, an isolated cell marked confidant will tend to be surrounded by radials at equal increments of latitude and longitude, marking the corners of areas which all rely primarily on data from the same confident cell. Also, the value of {TBLAND} for an ocean cell is pure extrapolation, and should not be taken seriously more than a few cells from the coast. Similar considerations apply to {TBWATER} in the interior of continents, except near lakes of sufficient size and clear sky to enable a "confident" estimate. Extrapolation to fill the region enables simpler coding but in reality these fields are only used in circumstances where they are appropriate.

Related considerations arise in the blending of data within a cell from equally significant measurements by different sensors, or from comparable dwellsondes at somewhat different times. Such distinct data sources are analyzed separately, but used only in those regions for the data is relevant, and averaged in regions of overlap. For certain fields, such as solar scattering angle {ASoS} such blending may give rise to noticeable discontinuities.

If just one well defined middle or high cloud layer is present in a cell {Pz} will be normally be an unbiassed represention of its height. {PzSD} will then be relatively small and representative of a true pixel to pixel variability across the cell. If both high and middle layers are present but do not overlap geographically, the same will still be true. Though the cloud mask may systematically fail to detect very thin cirrus, and clear sky may erroneously be classified as very low cloud, such errors do not greatly affect cloud pressure level determinations in the other categories. However, when there are present multiple partially-overlapping layers, or convective clouds with a large sub-pixel and pixel scale variability in cloud height, this condition should be associated with large values of {PzSD} in the categories affected. Under these circumstances the pixel scale algorithm tends to be confused, and systematic errors should be expected which lead to under representation of the higher cloud tops, and hence to an under estimate of the true variability. {PzSD} is thus an important indicator of the quality of the cloud analysis in each individual cell, though it should not be intepreted too literally.

The minimum level of detection of cirrus is probably a blackbody cloud signal of about 2.5 K in the channel 8 window. The cloud mask will treat pixels with smaller channel cloud forcing from high cloud as clear. Correlation studies show that this implies a minimum optical depth of 0.1 in visible wavelengths.

Relating the reported effective cloud fraction in the High, Middle and Low categories to a physical cloud fraction requires modeling assumptions about the spatial overlap of different layers, such as their statistical independence and is not a trivial exercise. In addition, it should be noted that when a layer of middle or low cloud is present in the same pixel below cloud in a higher category, such underlying cloud is not reported directly. Instead, the channel 8 cloud forcing is all attributed to the higher cloud, resulting in that higher cloud being reported with a lower cloud top height and larger effective cloud fraction than is physically appropriate.   The extreme of this is the default determination of cloud pressure level.   Some of the channel cloud forcing attributable to the underlying layer is thus already acounted for, and randomly assigning additional cloud fraction to this layer would be incorrect. Before fitting model cloud climatology models to this data, it would be appropriate to evaluate such biasses using simulations of the cloud analysis algorithm based upon differing assumptions about sub-pixel scale overlap among cloud layers at different levels.

A related issue is the classification of sub-pixel scale penetrating cumulus clouds, for which the assumption of a layer with well defined cloud pressure level and temperature is quite inappropriate. The channel 8 cloud forcing is then a function of viewing angle, because for larger satellite zenith angles {ASaZ} the opaque sides of cumulus towers screen a greater fraction of the horizontal area beneath. A plausible diagnostic of either many pixels containing multi-layer clouds, or of widespread penetrating cumulus within a cell, is a large value for the reported standard deviation of the cell cloud pressure level {PHIGHSD},{PMIDDLESD} or {PLOWSD}.

7. **Quality Assurance**

Table 2 shows the standard deviation {TC8ySD} under clear sky conditions of the pixel radiances in a 1 degree grid square for various channels, averaged over the entire grid and sample. For channels 1-4, and 9-11, the values over land and water are very similar, indicating that the major cause of this variability is either in the atmosphere, or more likely, instrument noise. Channel 8 shows a 35% increase in variability of brightness temperature for land (0.81 K) versus water (0.61 K), consistent with the greater variability over land in surface properties, particularly reflectance and heat capacity. For the other channels which are somewhat sensitive to the underlying surface, the results are intermediate.

Table 2.   Mean within cell variability

SD	TC8	RC1	RC2	RC3	RC4	RC5	RC6	RC7	RC8	RC9	RC10	RC11	RC12
LAND	.81	4.99	1.50	1.40	1.16	1.33	.02	1.45	1.42	.49	.21	.02	.04
WATER	.61	5.03	1.51	1.41	1.16	1.26	.02	1.16	.97	.49	.22	.02	.03
UNITS	K	R	R	R	R	R	R	R	R	R	R	R	R

K = Kelvin, R= mW m^-2 str^-1 cm^-1

These measures are thus a valuable indicator of noise in the input data, which can be monitored during the production process to flag any deterioration in quality. A related indicator (the tail) is the fraction of cells at any one time for which the observed standard deviation is greater than twice the observed average over the cells reporting. This is particularly

valuable for channel 8 brightness temperature as an indicator of possible failures in the cloud mask. Too small a avalue for the base temperature {TBy} for a cell would lead to many partially cloudy pixels being declared cloud free, resulting in an anomalously high pixel to pixel variability and the cell joining the tail. Some such occurrences should be expected by chance as a matter of course, but an unusually large number deserves more detailed examination by a scientist. Histograms of {TC8ySD} are reported in the HTML file for each granule, because such data cannot usefully be fitted into the HDF GRID format.   If the tail exceeds a threshold currently set at 7%, a quality assurance flag {QAFLAG} for the granule is set to "YES", triggering an inspection of probable cause by an oversight scientist.   Scientific evaluation of such exceptions from a larger sample of granules may be expected to suggest some modification of the {QAFLAG} critieria.

---

8. Dataset Contents

From a user's perspective the Data Product consists of individual data granules, indexed and joined by hypertext links to documentation for the entire collection. For structural view see the Product Description.

The documentation centers on the User Guide (this document), but also includes html browse images for a sample granule, identified by 1988141_2100, i.e. 1988 Julian Day 141, nominal time 2100. It also includes FORTRAN source code enabling computation of clear sky channel radiances from an atmospheric profile of temperature, water vapor and ozone, and conversely. Less obvious are machine searchable text files (in Object Description Language) of keywords to enable the EOSDIS core system to index and advertise this product, and a simplified version of similar information in HTML for easier access by human users.

The data granules are each in three distinct files with the same identifier, but with extensions .html, .hdf, and .odl respectively. The first contains the primary data in the form of 59 HDF grids, and an HDF viewer is required to image them. A different format for an example of the same information is provided in the browse metadata. Some statistical summaries complementing these grids are to be found in the granule .html file. These could not be included in the same HDF file structure for want of suitable software at this time. The product variables and their location are all identified in the list of symbols, along with other names appearing in the equations which describe the algorithms used to derive the products.

9. Product Description

Structure

The overall product consists of a tree structured directory containing numerous files comprising a set of items of the types listed below. To preserve links between these items which are necessary for their full utilization, the sub-directory names and file names must be preserved. Strict hdf format has been used where practicable, but because the full ECS toolkit is unavailable, metadata are presented in text and/or html files and are not guaranteed to conform to EOSDIS format standards. Some metadata is text in Object Description Language, identified by the tag .odl .

In the following description of the directory structure and content files:
the directory names are written in UPPER CASE, even though in the digital version they are in lower case;
in the file names a bracket [....] means a Class Name which *may* have multiple instances in that location. With this exception, every named file should appear in that directory *just once*.

```
GOES_VAS_PRODUCT_SET
    index.html
    sseclogo.gif

    METADATA
        index.html
        COLLECTION
            index.html
            user.html
            usrguide.html
            figure.gif
            toc.html
            goes_vas.odl

            BROWSE
```

```
                        index.html
                        [GRID].gif

                QA
                        index.html
                        GOES_VAS_[C]_[DATE]_[TIME].html

                RADCODE
                        index.html
                        example.txt
                        sounding.txt
                        example.out
                        vasfrwrd.f
                        vasregtw.dat
                        vasxtbnd.dat
                        vasxtdry.dat
                        vasxtozo.dat
                        vasxtwco.dat
                        vasxtwtl.dat
                        vasxtwts.dat
                        radiation_code.tar

        DATA
                index.html
                GOES_VAS_[C]_[DATE]_[TIME].hdf
                GOES_VAS_[C]_[DATE]_[TIME].odl
                GOES_VAS_[C]_[DATE]_[TIME].html

        SCIENCE_EVAL
                index.html
                diurnal.html
                        std.gif
```

Here GOES_VAS is the base directory, of which the name may change on transfer to another server. The internal html links use URLs relative to this base, so subdirectory names and file names should NOT be altered.

Contents

```
Item: "Collection Metadata"
    Number of exemplars: 1
    Location:
        COLLECTION
    Consisting of:
        4 Files in .html format
        1 File in .gif format
        1 File in .odl format
        subdirectory BROWSE containing :
                1 index.html
                69 [GRID].gif
        subdirectory QA containing:
                1 index.html
                1 File in .html format
        subdirectory RADCODE containing:
                1 index.html
                2 files.txt
                1 file.f
                6 files.dat
    Identified by
        Filename="index.html"
        Filename="user.html"
        Filename="toc.html"
        Filename="usrguide.html"
        Filename="figure.gif"
        Filename="GOES_VAS.odl"
    Description:
         1 User Guide,
         ESDTDescriptor file in Object Definition Language
End item

Item: "Granule"
    Number of exemplars: 83
    Location:    DATA
    Consisting of:
        1 File in hdf Format
        1 File in html format
        1 file in .odl format
    Identified by:
        Filename="GOES_VAS_[C]_[ Date]_[Time].hdf"
        Filename="GOES_VAS_[C]_[ Date]_[Time].html"
        Filename="GOES_VAS_[C]_[ Date]_[Time].odl"
```

```
        Description:
            Data and metadata corresponding to a particular data and time.
        Contents:
            in .hdf  69 GRIDS, corresponding to variables
                    in the LIST of SYMBOLS
            in .odl  Archive metadata using EOSDIS version B0 terms
            in .html Granule wide array data and edite structural metadata
End item


Item: "Radiation Code"
    Number of exemplars: 1
    Location:   RADCODE
    Consisting of:
        1 File in html Format
    Identified by:
        Filename="index.html"

          Filename="example.txt"

        Filename="sounding.txt"

        Filename=" example.out"

        Filename="vasfrwrd.f"

        Filename="vasregtw.dat"

        Filename="vasxtbnd.dat"

        Filename="vasxtdry.dat "

        Filename="vasxtozo.dat"

        Filename="vasxtwco.dat"

        Filename="vasxtwtl.dat"

        Filename="vasxtwts.dat"

    Description:
        FORTRAN Source Code, instructions for use
End item


Item: "Quality Assurance Report"
    Number of exemplars: 1
    Location:   QA
    Consisting of:
        1 File in hdf Format
    Identified by:
        Filename="GOES_VAS_[C]_[ Date]_[Time].hdf"
    Description:
        Scientist's report on granule, triggered by {QAFLAG}=YES
End item


Item: "Science Evaluation"
    This item is NOT a deliverable for this project. It is listed for the benefit of users
    Location:   SCIENCE_EVAL
    Number of exemplars: 1
    Consisting of:  A collection of text and html files
    Identified by:  Filename="index.html" + miscellaneous filenames
    Description:   Science evaluations contributed by interested scientists
    Updated version: maintained on
URL="http://www.purl.oclc.org/NET/U_WISCONSIN_SSEC/VAS_Product/SCIENCE_EVAL/"


End item
```

10. List of Symbols

G = Reported as a Grid in each granule
M = Reported as Metadata in each granule
C = Reported as Metadata in collection
E = Used in equations describing algorithms - not reported

Variable Name		Description
{RAn}	G	Observed Average Radiance for VAS channel n (Cloudy and Clear) n=1,...,12
{RCn}	G	Estimated Clear Sky Radiance for VAS channel n
{RCnUNC}	G	Uncertainty in RCn
{RCnp}	E	Observed Radiance for a Clear Sky Pixel in channel n
{RCnLANDSD}	M	Histogram of Standard Deviation of {RCnp} within cells over Land
{RCnWATERSD}	M	Histogram of Standard Deviation of {RCnp} within cells over Water
{VISIBLE}	G	Observed Average Radiance (Cloudy and Clear)
{ASaZ}	G	Satellite Zenith Angle
{ASoZ}	G	Solar Zenith Angle
{ASoS}	G	Angle between lines Sun-> Scatter Point -> Satellite

{LANDFRACTION}	G	Fraction of Cell Area Covered by Land vs Water
{TC8}	G	Brightness Temperature for channel 8
{TC8ySD}	E	Change in brightness temperature coresponding to {RC8ySD} for y=LAND/WATER
{TB}	E	Base Temperature
{TBLAND}	G	Estimated Base Temperature for Land
{TBLANDCHCK}	G	Measured Base temperature for Land
{TBLANDUNC}	G	Uncertainty in TBLAND
{TBLANDSF}	E	Structure function from {TBLAND}
{TBWATER}	G	Estimated Base Temperature for Water
{TBWATERCHCK}	G	Measured Base Temperature for Water
{TBWATERUNC}	G	Uncertainty in TBWATER
{TBWATERSF}	E	Structure function from {TBLAND}
{NCLEAR}	G	Number (per Cell) of Totally Clear VAS Pixels
{NCLEARUNC}	G	Uncertainty in NCLEAR
{NOBSTOTAL}	G	Number (per Cell) of VAS Pixels
{Pp}	E	Cloud pressure level for a VAS Pixel
{CFnp}	E	Effective Cloud Fraction for a Pixel from channel n
{Pzp}	E	Pixel cloud pressure level for class z = LOW, MIDDLE, HIGH
{PHIGHp}	E	Pixel cloud pressure level for High Cloud
{PHIGH}	G	Cell Average of {PHIGHp}
{PHIGHSD}	G	Standard Deviation within cell of {PHIGHp}
{Tzp}	E	Pixel cloud level temperature for class z = LOW, MIDDLE, HIGH
{THIGH}	G	Blackbody Temperature Corresponding to {PHIGH}
{CFHIGHp}	E	Effective Cloud Fraction of a Pixel reporting High Cloud
{CFHIGH}	G	Cell Average of {CFHIGHp}
{CFHIGHUNC}	G	Uncertainty in CFHIGH
{CFHIGHSOLID}	G	Fraction of Dense High Cloud for All VAS Pixels
{NOBSMIDDLE}	G	Number (per Cell) of Pixels Potentially Reporting Middle Cloud
{PMIDDLE}	G	Cell Average cloud pressure level for Middle Cloud
{PMIDDLESD}	G	Pixel to Pixel Standard Deviation of PMIDDLE
{TMIDDLE}	G	Blackbody Temperature Corresponding to PMIDDLE
{CFMIDDLE}	G	Effective Middle Cloud Fraction for Pixels in NOBSMIDDLE
{CFMIDDLEUNC}	G	Uncertainty in CFMIDDLE
{NOBSLOW}	G	Number (per Cell) of Pixels Potentially Reporting Low Cloud
{PLOW}	G	Cell Average cloud pressure level for Low Cloud
{PLOWSD}	G	Pixel to Pixel Standard Deviation of PLOW
{TLOW}	G	Blackbody Temperature Corresponding to PLOW
{CFLOW}	G	Effective Low Cloud Fraction for Pixels in NOBSLOW
{CFLOWUNC}	G	Uncertainty in CFLOW
{PRODUCTCLASS}	M	Indicator of available VAS channels
{QAFLAG}	M	Quality Assurance Flag YES/NO

11. Collection Metadata

Data File Format

The data files included in this product were produced using Version 1 of the Hierarchical Data Format - Earth Observing

System (HDF-EOS) library tools. Documentation describing the file structure and use of the library APIs can be found in the HDF-EOS User's Guide for the ECS Project, June, 1996 (170-TP-005-001). The User guide accompanies the Version 1 library which is available to the user community on the EDHS1 server.

The name of the data file encodes the product name, processing class and temporal identification for all fields within the file. The format for the data file naming convention is as follows:

[NAME]_[CLASS]_[CCYYDDD]_[HHMM].hdf

Where:
NAME = product name (e.g. GOES_VAS)
CLASS = product class code, where:

        A = all VAS channels (1,2,3,4,5,6,7,8,9,10,11,12)

        S = standard VAS channel set (2,3,4,5,7,8,9,10)

        C = cloud VAS channel set (3,4,5,8)

CCYYDDD = julian date of the data (e.g. 1988001 = 1 Jan, 1988)
HHMM = nominal time of day for the data (e.g. 0230 = 2:30 GMT)

.hdf = file structure (signifies underlaying HDF file format)

Each data file contains 69 individual fields with their associated structual metadata. These fields (labeled NUMERIC DATA GROUP within each file) represent the EOS specific data structure of type GRID. This datatype is an extension to the underlying HDF framework.

Structural metadata (labeled Vgroup within each file) describing attributes of each grid field are appended to the file during creation. These fields hold tags describing dimensionality, geolocation and other ancillary information which act to define the grid fields. For more information on HDF, please refer the NCSA HDF User's Guide, Version 4.0, University of Illinois, February 1996.

NOTE: Please be aware that INVENTORY level metadata is NOT included within the data files. Due to processing and software limitations, SSEC sought and was granted a exclusion to the granule level metadata requirements under phase 1 of this project. SSEC has produced Object Description Language (ODL) files containing the manditory inventory level metadata for each of the data files within this product set.

Each grid field is composed of uniformily spaced grid points (1 degree in Latitude and Longitude). Grid cells are defined by the geographic region bounded by .5 degree of Latitude and Longitude to either side of the grid point locations. The matrix of grid cells defines the grid field. An individual cell is identified by its row and column rank within the matrix, where grid cell 1,1 is defined as the northwest most cell. Row values increase from north to south and column values increase from west to east.

Depending on the nature of the field, grid point data are stored as either signed 16-bit integers (DFNT_INT16) or signed 32-bit floating point reals (DFNT_FLOAT32). The size and format of the values are defined by the structural metadata fields corresponding to particular grid field. The physical properties of all the fields contained within a file do not support negative values. Therefore, the absolute limits of the individual grid point values are bounded by 0 (zero) and the maximum allowable signed value of the format for a particular grid field. The actual range of possible values for any given field are defined within the inventory level metadata associated with the field.

For those grid points at which a true physical value does not exist or can not be computed, the MISSING DATA CODE (-1) will be assigned.

Each grid field within a file is 26 rows by 91 columns with geographic bounds from 50.5N latitude, 130.5W longitude (grid cell 1,1) to 4.5N latitude, 39.5W longitude (grid cell 26,91). All fields have the same geographic projection defined by the U.S. Geographical Survey General Cartographic Transformation Package (GTCP) with a Projection Id of GEOGRAPHIC and a projection Code Id of 0 (zero). The Geographic projection defines a field of equally spaced grid points in both latitude and longitude. These latitude and longitude bounds represent the upper left corner of grid cell 1,1 and the lower right corner of grid cell 26,91. For futher details on the GCTP projection package, please refer to Section 6.3.4 and

Apendix G of the SDP Toolkit Users Guide for ECS Project, May, 1996, (333-CD-003-001).

The table below lists the names of the grid fields contained within each data file, the size (in bits) and format of the grid point values within the grids and a short description of the fields.

No	Name	Size	Format	Description
1	RA1	32	float	observed average radiance for VAS channel 1
2	RA2	32	float	observed average radiance for VAS channel 2
3	RA3	32	float	observed average radiance for VAS channel 3
4	RA4	32	float	observed average radiance for VAS channel 4
5	RA5	32	float	observed average radiance for VAS channel 5
6	RA6	32	float	observed average radiance for VAS channel 6
7	RA7	32	float	observed average radiance for VAS channel 7
8	RA8	32	float	observed average radiance for VAS channel 8
9	RA9	32	float	observed average radiance for VAS channel 9
10	RA10	32	float	observed average radiance for VAS channel 10
11	RA11	32	float	observed average radiance for VAS channel 11
12	RA12	32	float	observed average radiance for VAS channel 12
13	RC1	32	float	estimated clear sky radiance for VAS channel 1
14	RC2	32	float	estimated clear sky radiance for VAS channel 2
15	RC3	32	float	estimated clear sky radiance for VAS channel 3
16	RC4	32	float	estimated clear sky radiance for VAS channel 4
17	RC5	32	float	estimated clear sky radiance for VAS channel 5
18	RC6	32	float	estimated clear sky radiance for VAS channel 6
19	RC7	32	float	estimated clear sky radiance for VAS channel 7
20	RC8	32	float	estimated clear sky radiance for VAS channel 8
21	RC9	32	float	estimated clear sky radiance for VAS channel 9
22	RC10	32	float	estimated clear sky radiance for VAS channel 10
23	RC11	32	float	estimated clear sky radiance for VAS channel 11
24	RC12	32	float	estimated clear sky radiance for VAS channel 12
25	RC1UNC	32	float	uncertainty in RC1
26	RC2UNC	32	float	uncertainty in RC2
27	RC3UNC	32	float	uncertainty in RC3
28	RC4UNC	32	float	uncertainty in RC4
29	RC5UNC	32	float	uncertainty in RC5
30	RC6UNC	32	float	uncertainty in RC6
31	RC7UNC	32	float	uncertainty in RC7
32	RC8UNC	32	float	uncertainty in RC8
33	RC9UNC	32	float	uncertainty in RC9
34	RC10UNC	32	float	uncertainty in RC10
35	RC11UNC	32	float	uncertainty in RC11
36	RC12UNC	32	float	uncertainty in RC12
37	VISIBLE	32	float	observed average visible radiance
38	ASaZ	32	float	satellite zenith angle
39	ASoZ	32	float	solar zenith angle
40	ASoS	32	float	solar scattering angle
41	LANDFRACTION	16	integer	fraction of cell area covered by land
42	TC8	32	float	brightness temperature for channel 8
43	TBLAND	32	float	estimated base temperature for land
44	TBLANDCHCK	32	float	measured base temperature for land

45	TBLANDUNC	32	float	uncertainty in TBLAND
46	TBWATER	32	float	estimated base temperature for water
47	TBWATERCHCK	32	float	measured base temperature for water
48	TBWATERUNC	32	float	uncertainty in TBWATER
49	NCLEAR	16	integer	number of clear VAS pixels per cell
50	NCLEARUNC	16	integer	uncertainty in NCLEAR
51	NOBSTOTAL	16	integer	number of VAS pixels per cell
52	PHIGH	16	integer	cell average cloud pressure level for high cloud
53	PHIGHSD	16	integer	pixel standard deviation for PHIGH
54	THIGH	32	float	blackbody temperature for PHIGH
55	CFHIGH	16	integer	effective high cloud fraction
56	CFHIGHUNC	16	integer	uncertainty in CFHIGH
57	CFHIGHSOLID	16	integer	fraction of dense high cloud
58	NOBSMIDDLE	16	integer	number of pixels potentially reporting middle cloud
59	PMIDDLE	16	integer	cell average cloud pressure level for middle cloud
60	PMIDDLESD	16	integer	pixel standard deviation for PMIDDLE
61	TMIDDLE	32	float	blackbody temperature for PMIDDLE
62	CFMIDDLE	16	integer	effective middle cloud fraction for NOBSMIDDLE
63	CFMIDDLEUNC	16	integer	uncertainty in CFMIDDLE
64	NOBSLOW	16	integer	number of pixels potentially reporting low cloud
65	PLOW	16	integer	cell average cloud pressure level for low cloud
66	PLOWSD	16	integer	pixel standard deviation for PLOW
67	TLOW	32	float	blackbody temperature for PLOW
68	CFLOW	16	integer	effective low cloud fraction for NOBSLOW
69	CFLOWUNC	16	integer	uncertainty in CFLOW

Abstract of Collection Metadata (Human Readable from GOES_VAS.odl)

CollectionDescription:

ShortName = "GOES_VAS"

LongName = "Radiances and Cloud Analysis from GOES Dwell Soundings"

CollectionDescription = "Observed and estimated clear sky radiance, with uncertainty, at one degree resolution over Continental U.S. for IR sounding channels. Cloud pressure level and fraction."

VersionID = 1.03

LastRevisionDate = "1998-06-26"

SuggestedUsage = "Assimilation of radiances into atmospheric GCMs;
Evaluation of relation between cloudiness, water vapor, and radiation to space, on seasonal, daily, hourly basis;
Indicators of convective rainfall;
Evapotranspiration from land surface."

VersionDescription = "Demonstration product set covering seven days in May 1988, sampled approximately every 90 mins. Inventory of GOES VAS archives from 1982 to 1995."

CitationforExternalPublication:

    Title: "Radiances and Cloud Analysis from GOES Dwell Soundings"
    Authors: Bretherton, F. and Wylie, D.
    Publisher: Space Science and Engineering Center, University of Wisconsin - Madison

Date: May 1998
Permanent URL: http://www.purl.oclc.org./NET/U_WISCONSIN_SSEC/VAS_Product/EOSDIS/"

CollectionState = "In Work"

MaintenanceandUpdateFrequency = "Unknown"

AccessConstraints = "None"

ProcessingLevelDescription = "GRID"

ProcessingLevelID = "3 (resampled or derived environmental variables)"

TemporalClass = "Hourly_Cloud_Statistics_and_Radiances"

Platform:

    PlatformShortName = "GOES-7"
    PlatformLongName = "Geostationary Operational Environmental Satellite-7"
    PlatformType = "Spacecraft"

SpatialCoverageType = "HORIZ&VERT"
SpatialDomain:

    Horizontal - BoundingRectangle
    WestBoundingCoordinate = -130.5
    NorthBoundingCoordinate = 50.5
    EastBoundingCoordinate = -39.5
    SouthBoundingCoordinate = 24.5

Vertical - AltitudeSystemDefinition

    AltitudeDatumName = "zero pressure (free space)"
    AltitudeDistanceUnits = "millibars"
    AltitudeEncodingMethod = "Attribute Values"
    AltitudeResolution = 1.0

GeographicCoordinateSystem:

    LatitudeResolution = 1.00
    LongitudeResolution = 1.00
    GeographicCoordinateUnits = "Decimal Degrees"

Temporal Domain:

    TimeType = "UTC"
    DateType = "Julian"
    TemporalRangeType = "Periodic"
    PrecisionofSeconds = 1
    EndsatPresentFlag = "N"

RegularPeriodicGroupDescription:

    PeriodName = "Irregular, Hourly"
    Period1stDate = "1988-136"
    Period1stTime = "01:00:00"
    PeriodCycleDurationUnit = "minutes"
    PeriodCycleDurationValue = 90.0
    PeriodDurationUnit = "minutes"

PeriodDurationValue = 45.0

RangeDateTime

RangeBeginningDate = "1988-136"
RangeBeginningTime = "01:00:00"
RangeEndingDate = "1988-142"
RangeEndingTime = "22:00:00"

Contacts:

ContactPerson - Role = "Producer"

HoursofService = "9AM to 4PM Local Time"
JobPosition = "Director, Space Science and Engineering Center"
ContactName = "Francis Bretherton"
StreetAddress = "1225 West Dayton Street" "Madison, Wisconsin 53706"
Country = "USA"
VoiceTelephoneNumber = "(608) 262-7497"
ElectronicMailAddress = "fbretherton@ssec.wisc.edu"

ContactPerson - Role = "Producer"

HoursofService = "9AM to 4PM Local Time"
JobPosition = "Scientist"
ContactName = "Donald Wylie"
StreetAddress = "1225 West Dayton Street""Madison, Wisconsin 53706"
Country = "USA"
VoiceTelephoneNumber = "(608) 263-7458"
ElectronicMailAddress = "donw@ssec.wisc.edu"

ContactOrganization - Role "Producer"

HoursofService "9AM to 4PM Local Time"
ContactOrganizationName "Space Science and Engineering Center, University of Wisconsin - Madison"
StreetAddress = "1225 West Dayton Street" "Madison, Wisconsin 53706"
Country = "USA"
VoiceTelephoneNumber = "(608) 262-0544"
ElectronicMailAddress = "bobf@ssec.wisc.edu"

Relevant Keywords:

ECSDiscipline="Earth Science"

Field of Study:
Atmosphere, Land Surface, Oceans, Radiance

SubField:
Atmospheric Temperature, Atmospheric Water Vapor, Clouds, Emissivity, Energy Deposition, Evapotranspiration, Heat Flux, Infrared, Infrared Flux, Infrared Imagery, Land Surface Temperature, Land Temperature,   Liquid Water Equivalent, Longwave Radiation, Net Radiation, Ocean Heat Budget, Ocean Temperature, Optical Depth, Optical Thickness, Outgoing Longwave Radiation,   Potential Temperature, Precipitable Water, Precipitation, Precipitation Amount, Precipitation Rate, Radiation Budget, Radiative Flux, Radiative Forcing, Reflectance,Shortwave Radiation, Snow, Snow Energy Balance,   Snow/Ice Temperature, Soil Heat Budget, Soil Temperature, Surface Air Temperature, Temperature Anomalies, Thermal Infrared, Visible Flux, Visible Imagery, Water Temperature, Water Vapor

Science content:
Atmospheric Emitted Radiation, Bidirectional Reflectance, Brightness Temperature, Cloud Amount, Cloud Forcing, Cloud Height, Cloud Vertical Distribution, Outgoing Longwave Radiation, Solar Radiation, Transmittance, Visible Flux, Water

12. References

Arkin P. A. and P.P.Xie, 1994, "The Gobal Precipitation Climatology Project - 1st Algorithm Intercomparion Project", Bull. Amer. Meteor. Soc., **75**, 401-419.

Menzel, W. P., 1993, Private Communication

Montgomery, H.E. and Uccelini L.W.(eds), "VAS Demonstration:(VISSR Atmospheric Sounder) Description and Final Report", NASA Reference Publication 1151, October 1985

Coakley, J.R. and F. P. Bretherton. "Cloud cover from high resolution scanner data: Detecting and allowing for prtially filled fields of view", J. Geophys. Res. , 87, 4917-4932.

Rossow, W.B., A. Walker, and M. Roiter, 1996 revision: International Satellite Cloud Climatology Project (ISCCP) Description of Reduced Resolution Radiance Data. WMO/TD-No. 58. World Meteorological Organization, 163 pp.

Frouin, R.J. and J.J. Simpson, 1995, "Radiometric Calibration of GOES-7 VISSR Solar Channels during the GOES Pathfinder Benchmark Period." , *Remote Sens. Environ.* **52**, 95-115.

Soden, B.J., and F.P. Bretherton, 1993, "Upper Tropospheric Relative Humidity for GOES 6.7 um Channel: Method and Climatolgogy for July 1987", *J. Geophys. Res.*, **98**, 16669-16688.

Smith, W.L. 1968, "An improved method for calculating tropospheric temperature and moisture from satellite radiometer measurements." *Monthly Weather Review*, **96**, 387-396

Wylie, D. P., and W. P. Menzel, 1989: Two Years of Cloud Cover Statistics Using VAS, J. of Climate, Vol 2, 380-392

13. Acknowledgements

Compiled by: *gvp@ssec.wisc.edu*
Latest Revision: June 10,1998
Return to: Top   Project Page

Radiation Code

Purpose

This FORTRAN code has kindly been provided by Hal Woolf and should be referenced as the Woolf VASFRWRD Model Version 1.

After compilation, the code in file VASFRWRD.F will calculate the radiances that a VAS instrument in orbit would see given a temperature and moisture sounding. The purpose of this code is to show how to compare temperature and moisture analyses to the radiances at the top of the atmosphere measured by the VAS instruments on the GOES satellites. A dataset of observed clear sky radiances over the continental U.S. and surrounding ocean is available from the University of Wisconsin GOES_VAS_Product_ Set Project. These VAS clear radiance data can be compared to those derived from the temperature and moisture fields in atmospheric General Circulation Models. Furthermore, a user can run sensitivity experiments to determine the probable significance of any discrepancies between the observed radiances and those generated from atmospheric profiles simulated in a GCM. Output from this code has been compared to line by line calculations using state of the art transmittance models.

The subroutine VASRTW retrieves an approximate atmospheric profile from VAS radiances, using coefficients derived by correlation with a sample of radiosondes. This may be used to invert the transformation performed by VASFRWRD. Such a retrieval is inevitably under-determined, and is only one out of infinitely many which are consistent with the stated radiances and the probable errors in measurement. It is not recommended as a definitive determination of the true state of the atmosphere, but is included because it is used in the analysis of cloud top pressure for the partially cloudy pixels within view of the satellite. For this purpose it is sufficiently accurate.

Instructions

Below are the instructions for compiling and running the example program to calculate VAS radiances from a sounding such as that in SOUNDING.TXT.

NOTE: File names are in lower case, following the UNIX convention; they are shown here in upper case for clarity.

This code, VASFRWRD.F, and associated coefficient files, as delivered, will run ONLY on a UNIX platform. To download, compile, and run VASFRWRD perform the following steps:

    1. Download a tar file containing the source code for the VASFRWRD.F program, the SOUNDING.TXT and EXAMPLE.TXT text files, and the asscoiated binary coefficient files (.DAT extension). To untar the files enter the command below.

        tar -xvf radiation_code.tar

    2. Compile VASFRWRD.F on an UNIX platform using the appropriate command below.

        IBM f77 -qrndsngl -qnomaf -o vasfrwrd vasfrwrd.f
        SGI f77 -32 -O0 -mips2 -static -bytereclen -col72 -o vasfrwrd vasfrwrd.f
        SUN f77 -o vasfrwrd vasfrwrd.f

    3. Run the executable VASFRWRD which uses SOUNDING.TXT to produce the file EXAMPLE.OUT. The text of EXAMPLE.OUT should be the same as the provided

[EXAMPLE.TXT](#).

VASFRWRD expects temperature and moisture profiles from [SOUNDING.TXT](#). To experiment with other soundings similar files must be created.

Users are encouraged to contribute an e-mail description to gvp@ssec.wisc.edu of their experiences running this code on other platforms. These will be made available on the VAS Product Home Page, accessible through the permanent (redirecting) URL:

"http://www.purl.oclc.org/net/U_Wisconsin_SSEC/VAS_Product/index.html"

All level- or pressure-dependent calculations are performed at the levels shown in SOUNDING.TXT. An arbitrary sounding must be interpolated and/or extrapolated to the specified pressures; a standard atmosphere or climatology is sufficient for the highest levels. Temperature must be specified in Kelvin, and moisture (as mixing ratio) in grams/kilogram. Dewpoint temperature can be transformed to mixing ratio via the function "satmix" (contained in VASFRWRD.F) as follows:

$$w = satmix(p,d)$$

where w is mixing ratio, p is pressure (mb), and d is dewpoint (K).

There are 4 additional required variables, presently hard-coded in VASFRWRD.F starting at line 20:

ksat - GOES satellite number: valid values are 4,5,6,7.

zena - zenith angle of the GOES view through the atmosphere to the location of the sounding; used in transmittance calculations. This parameter is included with each VAS clear-radiance set as the Grid {ASaZ}.

kmon - numeric month of the calendar year, used together with latitude (see below) to obtain ozone mixing ratio from a set of climatological tables.

rlat - latitude in degrees, from +90 (North Pole) to -90 (South Pole).

zena and kmon are also used to select coefficients for statistical retrieval.

VASFRWRD produces vertical profiles of atmospheric transmittance for each of the 12 channels; they are printed in EXAMPLE.TXT. See [below](#) for information regarding construction of the transmittance model and related subject matter.

Calculated VAS radiances are displayed in two forms:

spectral radiance .... unit = mW/m^2-steradian-cm^-1
equivalent blackbody temperature ... unit = K

The calculated equivalent blackbody (or brightness) temperatures are then passed to subroutine "vasrtw" to produce a statistical-regression-based retrieval (described [below](#)) of temperature and moisture.

The resulting profiles, also printed in EXAMPLE.TXT, can be compared to the input profiles used to synthesize the VAS radiances.

Transmittance Modeling for VAS

(1) Line-by-line database

Calculations have previously been done, using LBLRTM release 3.26 (22 Jan 96) and HITRAN96, for a set of 32 atmospheres, consisting of the 1976 U.S. Standard and 31 diverse profiles representing a wide range of meteorological conditions, from arctic to tropical. The very high, variable spectral resolution output was reduced to uniform 0.1-wavenumber spacing by simple averaging. Four runs were made: ALL, using the seven "standard" molecular species defined in LBLRTM -- water vapor, carbon dioxide, ozone, nitrous oxide, carbon monoxide, methane, and oxygen; WCO, water vapor continuum only; WNC, water vapor with no continuum; and WVO, water vapor (including continuum) plus ozone. The calculations, which span the spectral range 550 to 2950 cm^-1, have been done for nadir view plus five zenith angles, corresponding to values of secant(theta) = 1.00, 1.25, 1.50, 1.75, 2.00, and 2.25. The values of theta are 0., 36.87, 48.19, 55.15, 60., and 63.61.

(2) Generation of instrument transmittances

The 0.1-wavenumber transmittances for the four absorber configurations described in (1) were convolved with the response functions for the 12 bands or channels of the VAS instruments on GOES-4 through -7. The resulting band transmittances were then subjected to the following operations: wet = wco * wnc ozo = wvo / wet dry = all / wvo where "dry" is a generic term applied to the remaining uniformly mixed gases, primarily carbon dioxide. Defining "dry" and "ozo(ne)" in this manner preserves the validity of the product rule for total transmittance, even though the data do not actually represent monochromatic conditions.

(3) Fast (parameterized regression) model

The transmittance profiles in each band were then used to generate separate sets of regression coefficients (for the constituents dry, ozo, wco, and wnc) employing the PLOD (PressureLayeredOpticalDepth) algorithm [AIRS LAYERS package science notes, S. Hannon and L. Strow, University of Maryland - Baltimore County (UMBC)]. In addition, a set of "band- correction" coefficients was generated to account for the polychromatic character of the bands or channels in calculating radiances and brightness temperatures. These procedures were executed for each of the four GOES-VAS instruments, and the results stored in the BigEndian floating-point binary direct-access files: vasxtbnd.dat vasxtdry.dat vasxtozo.dat vasxtwco.dat vasxtwtl.dat vasxtwts.dat The reader is referred to the routines "pfcvas" and "tauvas" for usage of these files.

Results:

The results for 13 profiles are shown here. Profiles 1-12 were part of the set of 32 used in generating the correlation coefficients, i.e. they are dependent. Profile 33 is from sounding.txt, and is quite independent. The entries in the table are residuals, the difference in brightness temperature using the Woolf model minus that using LBLRMTM. The first table shows the comparison for channels 1-6; the second table for channels 7-12

GOES-07 SET 2, ZENITH ANGLE 48.19 atm 1-12 dependent, 33 independent

CHANNEL	1	2	3	4	5	6
WAVE-NUM	681.448	691.717	704.717	715.731	750.220	2213.552
1	-.00346	-.00552	-.04311	-.05737	-.00839	.36905
2	-.00308	.00133	-.00168	.00307	-.01985	.00470
3	-.00139	.00409	.01221	.03107	.00342	-.00015
4	-.04024	-.05566	-.05173	-.05557	-.04056	-.11272
5	-.00217	.01089	.03233	.06825	.04643	.11107
6	-.00299	.00574	.01314	.03716	.02634	.16158
7	-.00853	-.00841	-.00330	.02074	.01389	-.08160
8	-.04210	-.03816	.07997	.12158	.05267	-.05663

9	-.04836	-.06532	-.04376	-.04970	-.05298	-.04533
10	-.00636	.00092	.05194	.10712	.06754	-.05966
11	-.00014	.00404	.00929	.02336	.01074	.25259
12	-.00282	.00203	.00757	.01714	.00554	.01813
33	-.00270	-.00218	-.02895	-.02690	-.03003	.15991

CHANNEL	7	8	9	10	11	12
WAVE-NUM	789.179	898.584	1376.914	1489.870	2232.314	2532.704
1	-.05664	-.01703	-.28665	-.57704	.35405	.04462
2	-.00719	.00320	-.01831	-1.39276	.00880	-.00046
3	-.00914	.00095	-.03230	-1.22119	-.00044	-.00203
4	-.00662	-.00156	.19528	-.33603	-.11430	-.00490
5	.00134	.00201	-.08461	-.67143	.11630	.00436
6	-.00011	.00114	.02936	-.82130	.14612	.01035
7	.00121	-.00169	.02359	-.58690	-.07533	-.00601
8	.00909	-.00327	-.03529	-.30779	-.09709	-.00494
9	-.01474	-.01358	.02629	-.16933	-.10196	-.00272
10	-.01788	-.01306	-.06340	-.38043	-.05591	-.00290
11	-.00464	-.00241	-.07436	-.42195	.24252	.01666
12	.00172	-.00243	-.00067	-1.82355	.02444	.00064
33	.03284	-.01111	-.25577	-.59581	.16434	.01089

The comparison shows little obvious difference among the dependent and independent profiles, with the residuals typically at most a few tenths K , except for channel 10, where the differences are typically about 1 K. It should be noted that, in the radiosonde dataset which was used to generate the regressions, the water vapor vapor in the upper troposphere to which channel 10 is primarily sensitive was routinely set to a constant value because the sensor was not trusted to to give reliable results.

LBLRTM: S.A.Clough, "Radiative transfer model development in support of the Atmospheric Radiation Measurement program," in Proceedings of the Third Atmospheric Radiation Measurement (ARM) Science Team Meeting, CONF-9303112, U.S.Department of Energy, Washington, D.C., 1994

HITRAN: L.S.Rothman, R.R.Gamache, R.H.Tipping, C.P.Rinsland, M.A.H.Smith, D.C.Benner, V.Molathy Devi, J-M.Flaud, C.Camy-Peyret, A.Perrin, A.Goldman, S.Massie,L.R.Brown, and R.A.Toth, "The HITRAN Molecular Database: Editions of 1991 and 1992," Journal of Quantitative Spectroscopy and Radiative Transfer 48, 469-507, 1992

PLOD: S.Hannon, L.L.Strow, and W.W.McMillan, "Atmospheric infrared fast transmittance models: A comparison of two approaches," in Proceedings of SPIE Conference 2830, Optical Spectroscopic Techniques and Instrumentation for Atmospheric and Space Research II (12pp.), 1996

Statistical Retrieval using synthetic VAS brightness temperatures

(1) Sounding database

The basis for the statistical retrieval procedure is 14 months (Dec 96 -Jan 98) of radiosonde data from the region 16-60N, 60-130W. The data were subjected to rigorous quality control and grouped in three-month overlapping sets: Dec-Jan-Feb, Jan-Feb-Mar,..., Nov-Dec-Jan. The number of soundingsranges from 11921 in

Feb-Mar-Apr to 12931 in Jul-Aug-Sep, with the average around 12400.

(2) Synthesis of radiometric data

The "forward model" (fast transmittance algorithm plus related radiative-transfer code) contained within VASFRWRD.F was applied to each of the 12 sounding-data files, for each VAS instrument (GOES-4 through -7), for the same 6 zenith angles utilized in the generation of the transmittance model (see Transmittance Modeling for VAS).

(3) Regression procedure

The brightness temperatures from (2) were then regressed against the temperature and mixing ratio profiles described in (1), with secant(theta) as an additional "channel", producing the set of files vasregtw.m01,...,vasregtw.m12 where Dec-Jan-Feb is designated m(onth)01, Jan-Feb-Mar is m02,..., and Nov-Dec-Jan is m12. Each file contains coefficient sets for the four satellites. The contents of these 12 files were then reorganized for efficiency of access and combined into the single file vasregtw.dat which is entered from subroutine "vasrtw" according to zenith angle, month, and satellite number.

Harold M. Woolf
Space Science and Engineering Center
University of Wisconsin-Madison
1225 West Dayton Street,
Madison, WI 53706, U.S.A.
hal.woolf@ssec.wisc.edu

```
      program vasfrwrd

c * Forward calculation of VAS radiances from a sounding.

c         Printed output includes the input sounding, transmittances
c         for the 12 VAS channels, calculated radiances and blackbody
c         temperatures, and the regression-retrieved sounding.

      parameter (iui=51,iuo=61,nc=12,nl=40)
      character*16 cfmt
      character*12 ifile,ofile
      dimension taut(nl),temp(nl),wvmr(nl),ozmr(nl),p(nl),
     +   tauout(nc,nl),tvas(nc),rvas(nc)
      dimension rtemp(nl),rwvmr(nl)

      data p/0.1,0.2,0.5,1.0,1.5,2.,3.,4.,5.,7.,10.,15.,20.,25.,30.,
     + 50.,60.,70.,85.,100.,115.,135.,150.,200.,250.,300.,350.,400.,
     + 430.,475.,500.,570.,620.,670.,700.,780.,850.,920.,950.,1000./

c * Specified parameters
      ksat=7
      zena=48.0
      kmon=3
      rlat=45.0

c * Input file
      ifile='sounding.txt'
c * Output file
      ofile='example.out'

c * Get the climatological ozone mixing ratio.
      call clmozo(rlat,kmon, ozmr)

c * Open the sounding file and read the format from the first line.
      open(iui,file=ifile,form='formatted',status='old',err=900)
c * Open the output (text) file
      open(iuo,file=ofile,form='formatted',status='unknown')
      read(iui,'(a16)') cfmt
      write(iuo,'(''Format read from file '',a12,'' is '',a16)')
     +   ifile,cfmt

      write(iuo,'(''Original sounding plus ozone:''//
     +          ''lev    pres    temp    wvmr    ozmr''/)')
       numb=0
c * Read in the sounding, print temp, wvmr plus ozmr from climatology
      do lev=1,nl
         read(iui,cfmt,err=100,end=100) temp(lev),wvmr(lev)
         write(iuo,'(i3,f8.1,f8.2,f8.3,f8.4)') lev,p(lev),temp(lev),
     +     wvmr(lev),ozmr(lev)
         numb=numb+1
      enddo
100   close(iui)
      if(numb.lt.nl) go to 901

c * Input the Planck coefficients for the specified satellite.
      call pfcvas(ksat,*902)

      ls=nl
c .. ls is the surface level.
      tsk=temp(ls)
c .. tsk is the surface temperature.
      do kan=1,nc
c ..... calculate transmittances for each channel, store for printing
         call tauvas(temp,wvmr,ozmr,zena,ksat,kan,taut,*903)
```

```
            do lev=1,nl
                tauout(kan,lev)=taut(lev)
            enddo
c ..... obtain blackbody temperature
            tvas(kan)=tbcvas(taut,temp,tsk,ls,kan)
c ..... convert back to radiance
            rvas(kan)=plavas(tvas(kan),kan)
        enddo

        write(iuo,'(/''Transmittances''//''chan'',12(4x,i4)/)')
     +   (k,k=1,nc)
        do l=1,nl
            write(iuo,'(i4,12f8.4)') l,(tauout(k,l),k=1,nc)
        enddo

        write(iuo,'(/''RAD '',12f8.3)') (rvas(kan),kan=1,nc)
        write(iuo,'(''      mW/m^2-str-cm^-1'')')
        write(iuo,'(/''TBB '',12f8.2)') (tvas(kan),kan=1,nc)
        write(iuo,'(''      degrees Kelvin'')')

c * Use the blackbody temperatures to retrieve
c           a sounding via regression.
        call vasrtw(tvas,zena,kmon,ksat,rtemp,rwvmr,*904)

        write(iuo,'(/''Sounding retrieved from VAS TBB:''//
     +        ''lev    pres    temp    wvmr''/)')
        do lev=1,nl
            write(iuo,'(i3,f8.1,f8.2,f8.3)') lev,p(lev),
     +        rtemp(lev),rwvmr(lev)
        enddo

        write(iuo,'(/''VASFRWRD -- normal termination'')')
        write(0,'(''VASFRWRD -- normal termination'')')
        go to 999

900     write(iuo,'(/''OPEN error, sounding file '',a12)') ifile
        go to 990
901     write(iuo,'(/''Did not get the complete sounding.'')')
        go to 990
902     write(iuo,'(/''Trouble (probably I/O) in PFCVAS'')')
        go to 990
903     write(iuo,'(/''Trouble (probably I/O) in TAUVAS'')')
        go to 990
904     write(iuo,'(/''Trouble (probably I/O) in VASRTW'')')
990     write(iuo,'(/''VASFRWRD -- abnormal termination'')')
        write(0,'(''VASFRWRD -- abnormal termination'')')
999     continue
        close(iuo)
        end


        function tbcvas(tau,tem,tsk,ls,kan)
c * Integrate radiative-transfer equation for GOES/VAS
c .... version of 10.12.97
c          tau = transmittance profile
c          tem = temperature profile (degK)
c          tsk = surface-skin temperature (degK)
c           ls = surface model-level (.le.40)
c          kan = channel number

        dimension tau(*),tem(*)

        rad=0.
        b1=plavas(tem(1),kan)
        tau1=tau(1)
```

```
      do i=2,ls
         b2=plavas(tem(i),kan)
         tau2=tau(i)
         rad=rad+.5*(b1+b2)*(tau1-tau2)
         b1=b2
         tau1=tau2
      enddo
      bs=plavas(tsk,kan)
      rad=rad+bs*tau(ls)
      tbcvas=brivas(rad,kan)
      return
      end



      function plavas(t,k)
c * Planck function for VAS
c .... version of 17.02.98

c * Call 'PFCVAS' before using this or 'BRIVAS'
c  t = temperature, deg K
c  k = channel number: 1,...,12

      parameter (nb=12,nt=2)
      common/vaspfc/wnum(nb),fk1(nb),fk2(nb),tc(nt,nb)
      tt=tc(1,k)+tc(2,k)*t
      expn=exp(fk2(k)/tt) - 1.
      plavas=fk1(k)/expn
      return
      end



      function brivas(r,k)
c * Inverse Planck function for VAS
c .... version of 17.02.98

c * Call 'PFCVAS' before using this or 'PLAVAS'
c  r = radiance, standard units
c  k = channel number: 1,...,12

      parameter (nb=12,nt=2)
      common/vaspfc/wnum(nb),fk1(nb),fk2(nb),tc(nt,nb)
      expn=fk1(k)/r+1.
      tt=fk2(k)/alog(expn)
      brivas=(tt-tc(1,k))/tc(2,k)
      return
      end



      subroutine pfcvas(ngoes,*)
c * Input VAS Planck-function and band-correction coefficients.
c .... version of 18.02.98
c * Valid for GOES-4, -5, -6, -7

      parameter (iuc=33,lenc=100,nb=12,nt=2,lenp=nb*(nt+3))
      parameter (lencb=lenc*4)
c     common/vaspfc/wnum(nb),fk1(nb),fk2(nb),tc(nt,nb)
      common/vaspfc/pbuf(lenp)
      dimension cbuf(lenc)
      character*12 cfile/'vasxtbnd.dat'/

      open(iuc,file=cfile,recl=lencb,access='direct',status='old',
     *      err=100)
      irec=ngoes-3
      read(iuc,rec=irec) cbuf
      close(iuc)
```

```
      do i=1,lenp
         pbuf(i)=cbuf(i)
      enddo
      return

  100 return1
      end



      subroutine tauvas(temp,wvmr,ozmr,theta,ngoes,kan,taut,*)
c ... Calculate VAS transmittances
c ... version of 18.02.98
c ... Valid for GOES-4, -5, -6, -7

c * Inputs:
c        temp = temperature profile (degK)
c        wvmr = water-vapor mixing-ratio profile (g/kg)
c        ozmr = ozone mixing-ratio profile (ppmv)
c       theta = local zenith angle (deg)
c       ngoes = satellite number (4...7)
c         kan = channel number   (1...12)

c * Outputs:
c        taut = total transmittance profile
c           * = error return for I/O problems

c * LarrabeeStrow/HalWoolf/PaulVanDelst regression model based on
c *     'FSCD3P' line-by-line transmittances.
c * Input temperatures, and water-vapor and ozone mixing ratios, must
c *     be defined at the pressure levels in array 'pstd'
c *     (see block data 'screfatm').
c * Units: temperature, deg-K; water vapor, g/kg; ozone, ppmv.
c * Logical unit 33 is used for coefficient files.
c * Component tau's are returned through common, product in 'taut'.

         parameter (iuc=33,nk=5,nl=40,nm=nl-1,nr=12)
       parameter (nxc= 4,ncc=nxc+1,lencc=ncc*nm,lenccb=lencc*4)
       parameter (nxd= 8,ncd=nxd+1,lencd=ncd*nm,lencdb=lencd*4)
       parameter (nxo= 9,nco=nxo+1,lenco=nco*nm,lencob=lenco*4)
       parameter (nxl= 2,ncl=nxl+1,lencl=ncl*nm,lenclb=lencl*4)
       parameter (nxs=11,ncs=nxs+1,lencs=ncs*nm,lencsb=lencs*4)
       parameter (nxw=nxl+nxs)
       common/atmstd/pstd(nl),tstd(nl),wstd(nl),ostd(nl)
       common/taudwo/taud(nl),tauw(nl),tauo(nl)
       dimension temp(*),wvmr(*),ozmr(*),taut(*)
       dimension coefd(ncd,nm,nr),coefo(nco,nm,nr),coefl(ncl,nm,nr)
         dimension coefs(ncs,nm,nr),coefc(ncc,nm,nr)
         dimension pavg(nm),tref(nm),wref(nm),oref(nm)
         dimension tavg(nm),wamt(nm),oamt(nm),secz(nm)
         dimension tauc(nl),tlas(nl),wlas(nl),olas(nl)
         dimension xdry(nxd,nm),xozo(nxo,nm),xwet(nxw,nm),xcon(nxc,nm)
       character*12 cfile/'vasxtcom.dat'/
       character*3 comp(nk)/'dry','ozo','wts','wtl','wco'/
       integer*4 lengcf(nk)/lencdb,lencob,lencsb,lenclb,lenccb/
         logical newang,newatm
         data init/0/,tlas/nl*0./,wlas/nl*0./,olas/nl*0./,zlas/-999./
         secant(z)=1./cos(0.01745329*z)

       if(ngoes.ne.init) then
           koff=(ngoes-4)*nr
         do l=1,nk
             cfile(6:8)=comp(l)
           lencf=lengcf(l)
           open(iuc,file=cfile,recl=lencf,access='direct',
```

```fortran
*             status='old',err=200)
          if(l.eq.1) then
           do k=1,nr
                krec=k+koff
                read(iuc,rec=krec) ((coefd(i,j,k),i=1,ncd),j=1,nm)
           enddo
          elseif(l.eq.2) then
           do k=1,nr
                krec=k+koff
                read(iuc,rec=krec) ((coefo(i,j,k),i=1,nco),j=1,nm)
           enddo
          elseif(l.eq.3) then
           do k=1,nr
                krec=k+koff
                read(iuc,rec=krec) ((coefs(i,j,k),i=1,ncs),j=1,nm)
           enddo
          elseif(l.eq.4) then
           do k=1,nr
                krec=k+koff
                read(iuc,rec=krec) ((coefl(i,j,k),i=1,ncl),j=1,nm)
           enddo
          elseif(l.eq.5) then
           do k=1,nr
                krec=k+koff
                read(iuc,rec=krec) ((coefc(i,j,k),i=1,ncc),j=1,nm)
           enddo
          endif
         close(iuc)
       enddo
       call conpir(pstd,tstd,wstd,ostd,nl,1,pavg,tref,wref,oref)
       init=ngoes
      endif

      dt=0.
      dw=0.
      do=0.
     do j=1,nl
         dt=dt+abs(temp(j)-tlas(j))
         tlas(j)=temp(j)
         dw=dw+abs(wvmr(j)-wlas(j))
         wlas(j)=wvmr(j)
         do=do+abs(ozmr(j)-olas(j))
         olas(j)=ozmr(j)
       taud(j)=1.0
       tauw(j)=1.0
       tauc(j)=1.0
       tauo(j)=1.0
       taut(j)=1.0
     enddo
      datm=dt+dw+do
      newatm=datm.ne.0.
      if(newatm) then
       call conpir(pstd,temp,wvmr,ozmr,nl,1,pavg,tavg,wamt,oamt)
      endif

      newang=theta.ne.zlas
      if(newang) then
         zsec=secant(theta)
         do l=1,nm
            secz(l)=zsec
         enddo
         zlas=theta
      endif

      if(newang.or.newatm) then
```

```
            call calpir(tref,wref,oref,tavg,wamt,oamt,pavg,secz,
     *    nm,nxd,nxw,nxo,nxc,xdry,xwet,xozo,xcon)
          endif

          k=kan
c * dry
       call taugen(ncd,nxd,nm,coefd(1,1,k),xdry,taud)
c * ozo
       call taugen(nco,nxo,nm,coefo(1,1,k),xozo,tauo)
c * wet
        call tauwet(ncs,ncl,nxs,nxl,nxw,nm,coefs(1,1,k),
     *         coefl(1,1,k),xwet,tauw)
        call taugen(ncc,nxc,nm,coefc(1,1,k),xcon,tauc)
          do j=1,nl
            tauw(j)=tauw(j)*tauc(j)
          enddo
c * total
        do j=1,nl
            taut(j)=taud(j)*tauo(j)*tauw(j)
        enddo
        return
  200 return1
        end


        block data screfatm
c * Reference Atmosphere is 1976 U.S. Standard
        parameter (nl=40)
        common/atmstd/pstd(nl),tstd(nl),wstd(nl),ostd(nl)
        data pstd/ .1,.2,.5,1.,1.5,2.,3.,4.,5.,7.,10.,15.,20.,25.,30.,
     + 50.,60.,70.,85.,100.,115.,135.,150.,200.,250.,300.,350.,400.,
     + 430.,475.,500.,570.,620.,670.,700.,780.,850.,920.,950.,1000./
        data tstd/
     +  231.70, 245.22, 263.35, 270.63, 264.07, 257.93, 249.51, 243.65,
     +  239.24, 232.64, 228.07, 225.00, 223.13, 221.72, 220.54, 217.28,
     +  216.70, 216.70, 216.70, 216.70, 216.70, 216.70, 216.70, 216.72,
     +  220.85, 228.58, 235.38, 241.45, 244.81, 249.48, 251.95, 258.32,
     +  262.48, 266.40, 268.61, 274.21, 278.74, 282.97, 284.71, 287.50/
        data wstd/
     +    0.003,  0.003,  0.003,  0.003,  0.003,  0.003,  0.003,  0.003,
     +    0.003,  0.003,  0.003,  0.003,  0.003,  0.003,  0.003,  0.003,
     +    0.003,  0.003,  0.003,  0.003,  0.003,  0.004,  0.005,  0.014,
     +    0.036,  0.089,  0.212,  0.331,  0.427,  0.588,  0.699,  1.059,
     +    1.368,  1.752,  1.969,  2.741,  3.366,  3.976,  4.255,  4.701/
        data ostd/
     + 0.65318,1.04797,2.13548,3.82386,5.26768,6.11313,7.35964,7.75004,
     + 7.82119,7.56126,6.92006,6.10266,5.55513,5.15298,4.59906,2.86792,
     + 2.29259,1.80627,1.28988,0.93973,0.72277,0.54848,0.46009,0.29116,
     + 0.16277,0.09861,0.06369,0.05193,0.04718,0.04097,0.03966,0.03614,
     + 0.03384,0.03342,0.03319,0.03249,0.03070,0.02878,0.02805,0.02689/
        end


        subroutine calpir(t_avg_ref,  amt_wet_ref, amt_ozo_ref,
     +                          t_avg,      amt_wet,     amt_ozo,
     +                          p_avg,      sec_theta,   n_layers,
     +                          n_dry_pred, n_wet_pred,  n_ozo_pred,
     +                          n_con_pred,
     +                          pred_dry,   pred_wet,    pred_ozo,
     +                          pred_con)
c ... version of 19.09.96

c  PURPOSE:

c    Routine to calculate the predictors for the dry (temperature),
c      wet and ozone components of a fast transmittance model for a
```

```
c        scanning satellite based instrument.

c   REFERENCES:

c      AIRS FTC package science notes and software, S. Hannon and L. Strow,
c         Uni. of Maryland, Baltimore County (UMBC)

c   CREATED:

c      19-Sep-1996 HMW

c   ARGUMENTS:

c        Input
c        -----------
c      t_avg_ref  - REAL*4 reference layer average temperature array (K)

c     amt_wet_ref - REAL*4 reference water vapour amount array (k.mol)/cm^2

c     amt_ozo_ref - REAL*4 reference ozone amount array (k.mol)/cm^2

c        t_avg    - REAL*4 layer average temperature array (K)

c       amt_wet   - REAL*4 water vapour amount array (k.mol)/cm^2

c       amt_ozo   - REAL*4 ozone amount array (k.mol)/cm^2

c        p_avg    - REAL*4 layer average pressure array (mb)

c      sec_theta  - REAL*4 secant of the zenith angle array

c       n_layers  - INT*4 Number of atmospheric layers

c     n_dry_pred  - INT*4 number of dry (temperature) predictors

c     n_wet_pred  - INT*4 number of water vapour predictors

c     n_ozo_pred  - INT*4 number of ozone predictors

c     n_con_pred  - INT*4 number of water vapour continuum predictors

c        Output
c        -----------
c      pred_dry   - REAL*4 dry gas (temperature) predictor matrix

c      pred_wet   - REAL*4 water vapour predictor matrix

c      pred_ozo   - REAL*4 ozone predictor matrix

c      pred_con   - REAL*4 water vapour continuum predictor matrix

c   COMMENTS:

c      Levels or Layers?
c      -----------------
c         Profile data is input at a number of *LAYERS*.

c      Layer Numbering pt. A
c      ---------------------
c         Layer 1   => Atmosphere between LEVELs 1 & 2
c         Layer 2   => Atmosphere between LEVELs 2 & 3
c                               .
c                               .
c                               .
c         Layer L-1 => Atmosphere between LEVELs L-1 & L
```

```
c     Layer Numbering pt. B
c     --------------------
c        For the HIS instrument, Layer 1 is at the top of the atmosphere
c           and Layer L-1 is at the surface.

c     Layer Numbering pt. C
c     --------------------
c        In this routine the number of *LAYERS* is passed in the argument
c           list, _not_ the number of LEVELS.  This was done to improve
c           the readability of this code, i.e. loop from 1->L(ayers)
c           rather than from 1->L(evels)-1.


C======================================================================


c----------------------------------------------------------------------
c                       Turn off implicit type declaration
c----------------------------------------------------------------------

       implicit none


c----------------------------------------------------------------------
c                                 Arguments
c----------------------------------------------------------------------

c -- Input

       integer*4 n_layers,
      +           n_dry_pred, n_wet_pred, n_ozo_pred, n_con_pred

       real*4    t_avg_ref(*), amt_wet_ref(*), amt_ozo_ref(*),
      +          t_avg(*),     amt_wet(*),     amt_ozo(*),
      +          p_avg(*),     sec_theta(*)

c -- Output

       real*4    pred_dry(n_dry_pred, *),
      +          pred_wet(n_wet_pred, *),
      +          pred_ozo(n_ozo_pred, *),
      +          pred_con(n_con_pred, *)

c----------------------------------------------------------------------
c                           Local variables
c----------------------------------------------------------------------

c -- Parameters

       integer*4 max_layers
       parameter ( max_layers = 41 )

       integer*4 max_dry_pred, max_wet_pred, max_ozo_pred, max_con_pred
       parameter ( max_dry_pred = 8,
      +            max_wet_pred = 13,
      +            max_ozo_pred = 9,
      +            max_con_pred = 4 )

c -- Scalars

       integer*4 l

c -- Arrays

c     ....Pressure
       real*4    p_dp(max_layers),
      +          p_norm(max_layers)
```

```
c       ....Temperature
        real*4    delta_t(max_layers),
       +          t_ratio(max_layers),
       +          pw_t_ratio(max_layers)       ! Pressure weighted

c       ....Water vapour
        real*4    wet_ratio(max_layers),
       +          pw_wet(max_layers),          ! Pressure weighted
       +          pw_wet_ref(max_layers),      ! Pressure weighted
       +          pw_wet_ratio(max_layers)     ! Pressure weighted

c       ....Ozone
        real*4    ozo_ratio(max_layers),
       +          pw_ozo_ratio(max_layers),    ! Pressure weighted
       +          pow_t_ratio(max_layers)      ! Pressure/ozone weighted

c**********************************************************************
c                        ** Executable code **
c**********************************************************************

c----------------------------------------------------------------------
c                   -- Check that n_layers is o.k. --
c----------------------------------------------------------------------

        if( n_layers .gt. max_layers )then
          write(*,'(/10x,''*** calpir : n_layers > max_layers'')')
          stop
        end if

c----------------------------------------------------------------------
c          -- Check that numbers of predictors is consistent --
c----------------------------------------------------------------------

c       -------------------------------
c       # of dry (temperature) predictors
c       -------------------------------

        if( n_dry_pred .ne. max_dry_pred )then
          write(*,'(/10x,''*** calpir : invalid n_dry_pred'')')
          stop
        end if

c       --------------------------
c       # of water vapour predictors
c       --------------------------

        if( n_wet_pred .ne. max_wet_pred )then
          write(*,'(/10x,''*** calpir : invalid n_wet_pred'')')
          stop
        end if

c       --------------------
c       # of ozone predictors
c       --------------------

        if( n_ozo_pred .ne. max_ozo_pred )then
          write(*,'(/10x,''*** calpir : invalid n_ozo_pred'')')
          stop
        end if

c       -------------------------------------
c       # of water vapour continuum predictors
c       -------------------------------------
```

```fortran
      if( n_con_pred .ne. max_con_pred )then
        write(*,'(/10x,''*** calpir : invalid n_con_pred'')')
        stop
      end if

c-----------------------------------------------------------------------
c           -- Calculate ratios, offsets, etc, for top layer --
c-----------------------------------------------------------------------

c       -----------------
c       Pressure variables
c       -----------------

      p_dp(1)   = p_avg(1) * ( p_avg(2) - p_avg(1) )
      p_norm(1) = 0.0

c       --------------------
c       Temperature variables
c       --------------------

      delta_t(1)    = t_avg(1) - t_avg_ref(1)
      t_ratio(1)    = t_avg(1) / t_avg_ref(1)
      pw_t_ratio(1) = 0.0

c       ----------------
c       Amount variables
c       ----------------

c       ....Water vapour

      wet_ratio(1)    = amt_wet(1) / amt_wet_ref(1)
      pw_wet(1)       = p_dp(1) * amt_wet(1)
      pw_wet_ref(1)   = p_dp(1) * amt_wet_ref(1)
      pw_wet_ratio(1) = wet_ratio(1)

c       ....Ozone

      ozo_ratio(1)    = amt_ozo(1) / amt_ozo_ref(1)
      pw_ozo_ratio(1) = 0.0
      pow_t_ratio(1)  = 0.0

c-----------------------------------------------------------------------
c           -- Calculate ratios, offsets, etc, for all layers --
c-----------------------------------------------------------------------

      do l = 2, n_layers

c         -----------------
c         Pressure variables
c         -----------------

        p_dp(l) = p_avg(l) * ( p_avg(l) - p_avg(l-1) )
        p_norm(l) = p_norm(l-1) + p_dp(l)

c         --------------------
c         Temperature variables
c         --------------------

        delta_t(l)    = t_avg(l) - t_avg_ref(l)
        t_ratio(l)    = t_avg(l) / t_avg_ref(l)
        pw_t_ratio(l) = pw_t_ratio(l-1) + ( p_dp(l) * t_ratio(l-1) )

c         ----------------
c         Amount variables
c         ----------------
```

```
c          ..Water vapour

          wet_ratio(l)   = amt_wet(l) / amt_wet_ref(l)
          pw_wet(l)      = pw_wet(l-1) + ( p_dp(l)  * amt_wet(l) )
          pw_wet_ref(l)  = pw_wet_ref(l-1) + ( p_dp(l) * amt_wet_ref(l) )

c          ..Ozone

          ozo_ratio(l)     = amt_ozo(l) / amt_ozo_ref(l)
          pw_ozo_ratio(l) = pw_ozo_ratio(l-1) +
     +                        ( p_dp(l) * ozo_ratio(l-1) )
          pow_t_ratio(l)  = pow_t_ratio(l-1) +
     +                        ( p_dp(l) * ozo_ratio(l-1) * delta_t(l-1) )

        end do

c-----------------------------------------------------------------------
c              -- Scale the pressure dependent variables --
c-----------------------------------------------------------------------

        do l = 2, n_layers

          pw_t_ratio(l)    = pw_t_ratio(l) / p_norm(l)
          pw_wet_ratio(l) = pw_wet(l) / pw_wet_ref(l)
          pw_ozo_ratio(l) = pw_ozo_ratio(l) / p_norm(l)
          pow_t_ratio(l)  = pow_t_ratio(l) / p_norm(l)

        end do

c-----------------------------------------------------------------------
c                    -- Load up predictor arrays --
c-----------------------------------------------------------------------

        do l = 1, n_layers

c        ---------------------
c        Temperature predictors
c        ---------------------

          pred_dry(1,l) = sec_theta(l)
          pred_dry(2,l) = sec_theta(l) * sec_theta(l)
          pred_dry(3,l) = sec_theta(l) * t_ratio(l)
          pred_dry(4,l) = pred_dry(3,l) * t_ratio(l)
          pred_dry(5,l) = t_ratio(l)
          pred_dry(6,l) = t_ratio(l) * t_ratio(l)
          pred_dry(7,l) = sec_theta(l) * pw_t_ratio(l)
          pred_dry(8,l) = pred_dry(7,l) / t_ratio(l)

c        ----------------------
c        Water vapour predictors
c        ----------------------

          pred_wet(1,l)  = sec_theta(l) * wet_ratio(l)
          pred_wet(2,l)  = sqrt( pred_wet(1,l) )
          pred_wet(3,l)  = pred_wet(1,l) * delta_t(l)
          pred_wet(4,l)  = pred_wet(1,l) * pred_wet(1,l)
          pred_wet(5,l)  = abs( delta_t(l) ) * delta_t(l) * pred_wet(1,l)
          pred_wet(6,l)  = pred_wet(1,l) * pred_wet(4,l)
          pred_wet(7,l)  = sec_theta(l) * pw_wet_ratio(l)
          pred_wet(8,l)  = pred_wet(2,l) * delta_t(l)
          pred_wet(9,l)  = sqrt( pred_wet(2,l) )
          pred_wet(10,l) = pred_wet(7,l) * pred_wet(7,l)
          pred_wet(11,l) = sqrt( pred_wet(7,l) )
          pred_wet(12,l) = pred_wet(1,l)
```

```
          pred_wet(13,l) = pred_wet(2,l)

c         ----------------
c         Ozone predictors
c         ----------------

          pred_ozo(1,l) = sec_theta(l) * ozo_ratio(l)
          pred_ozo(2,l) = sqrt( pred_ozo(1,l) )
          pred_ozo(3,l) = pred_ozo(1,l) * delta_t(l)
          pred_ozo(4,l) = pred_ozo(1,l) * pred_ozo(1,l)
          pred_ozo(5,l) = pred_ozo(2,l) * delta_t(l)
          pred_ozo(6,l) = sec_theta(l) * pw_ozo_ratio(l)
          pred_ozo(7,l) = sqrt( pred_ozo(6,l) ) * pred_ozo(1,l)
          pred_ozo(8,l) = pred_ozo(1,l) * pred_wet(1,l)
          pred_ozo(9,l) = sec_theta(l) * pow_t_ratio(l) * pred_ozo(1,l)


c         -------------------------------
c         Water vapour continuum predictors
c         -------------------------------

          pred_con(1,l) = sec_theta(l) * wet_ratio(l) /
     *                ( t_ratio(l) * t_ratio(l) )
          pred_con(2,l) = pred_con(1,l) * pred_con(1,l) / sec_theta(l)
          pred_con(3,l) = sec_theta(l) * wet_ratio(l) / t_ratio(l)
          pred_con(4,l) = pred_con(3,l) * wet_ratio(l)

       end do

       return
       end

       subroutine conpir( p, t, w, o, n_levels, i_dir,
     +                      p_avg, t_avg, w_amt, o_amt)
c ... version of 19.09.96

c  PURPOSE:

c    Function to convert atmospheric water vapour (g/kg) and ozone (ppmv)
c       profiles specified at n_levels layer BOUNDARIES to n_levels-1
c       integrated layer amounts of units (k.moles)/cm^2.  The average
c       LAYER pressure and temperature are also returned.

c  REFERENCES:

c    AIRS LAYERS package science notes, S. Hannon and L. Strow, Uni. of
c       Maryland, Baltimore County (UMBC)

c  CREATED:

c    19-Sep-1996 HMW

c  ARGUMENTS:

c     Input
c     --------
c        p      - REAL*4 pressure array (mb)

c        t      - REAL*4 temperature profile array (K)

c        w      - REAL*4 water vapour profile array (g/kg)

c        o      - REAL*4 ozone profile array (ppmv)

c    n_levels - INT*4 number of elements used in passed arrays
```

```
c      i_dir   - INT*4 direction of increasing layer number

c                    i_dir = +1, Level(1) == p(top)        } satellite/AC
c                                Level(n_levels) == p(sfc)  }    case

c                    i_dir = -1, Level(1) == p(sfc)        } ground-based
c                                Level(n_levels) == p(top)  }    case

c      Output
c      --------
c      p_avg   - REAL*4 average LAYER pressure array (mb)

c      t_avg   - REAL*4 average LAYER temperature (K)

c      w_amt   - REAL*4 integrated LAYER water vapour amount array (k.moles)/cm^2

c      o_amt   - REAL*4 integrated LAYER ozone amount array (k.moles)/cm^2

c  ROUTINES:

c    Subroutines:
c    ------------
c      gphite      - calculates geopotential height given profile data.

c    Functions:
c    ----------
c      NONE

c  COMMENTS:

c    Levels or Layers?
c    -----------------
c      Profile data is input at a number of *LEVELS*.  Number densitites
c        are calculated for *LAYERS* that are bounded by these levels.
c        So, for L levels there are L-1 layers.

c    Layer Numbering
c    ---------------
c      Layer 1   => Atmosphere between LEVELs 1 & 2
c      Layer 2   => Atmosphere between LEVELs 2 & 3
c                          .
c                          .
c                          .
c      Layer L-1 => Atmosphere between LEVELs L-1 & L

c=========================================================================

c-------------------------------------------------------------------------
c              -- Prevent implicit typing of variables --
c-------------------------------------------------------------------------

      implicit none

c-------------------------------------------------------------------------
c                          -- Arguments --
c-------------------------------------------------------------------------

c -- Arrays

      real*4     p(*), t(*), w(*), o(*),
     +           p_avg(*), t_avg(*), w_amt(*), o_amt(*)

c -- Scalars

      integer*4 n_levels, i_dir
```

```fortran
c-----------------------------------------------------------------------
c                       -- Local variables --
c-----------------------------------------------------------------------

c -- Parameters

      integer*4 max_levels
      parameter ( max_levels = 50 )         ! Maximum number of layers

      real*4    r_equator, r_polar, r_avg
      parameter ( r_equator = 6.378388e+06, ! Earth radius at equator
     +            r_polar   = 6.356911e+06, ! Earth radius at pole
     +            r_avg     = 0.5*(r_equator+r_polar) )

      real*4    g_sfc
      parameter ( g_sfc = 9.80665 )         ! Gravity at surface

      real*4    rho_ref
      parameter ( rho_ref = 1.2027e-12 )    ! Reference air "density"

      real*4    mw_dryair, mw_h2o, mw_o3
      parameter ( mw_dryair = 28.97,        ! Molec. wgt. of dry air (g/mol)
     +            mw_h2o    = 18.0152,       ! Molec. wgt. of water
     +            mw_o3     = 47.9982 )      ! Molec. wgt. of ozone

      real*4    R_gas, R_air
      parameter ( R_gas = 8.3143,           ! Ideal gas constant (J/mole/K)
     +            R_air = 0.9975*R_gas )     ! Gas constant for air (worst case)

c -- Scalars

      integer*4 l, l_start, l_end, l_indx

      real*4    rho1, rho2, p1, p2, w1, w2, o1, o2, z1, z2,
     +          c_avg, g_avg, z_avg, w_avg, o_avg,
     +          dz, dp, r_hgt, wg, og, A, B

c -- Arrays

      real*4    z(max_levels),              ! Pressure heights (m)
     +          g(max_levels),              ! Acc. due to gravity (m/s/s)
     +          mw_air(max_levels),         ! Molec. wgt. of air (g/mol)
     +          rho_air(max_levels),        ! air mass density (kg.mol)/m^3
     +          c(max_levels),              ! (kg.mol.K)/(N.m)
     +          w_ppmv(max_levels)          ! h2o LEVEL amount (ppmv)

c*********************************************************************
c                    ** Executable code **
c*********************************************************************

c-----------------------------------------------------------------------
c          -- Calculate initial values of pressure heights --
c-----------------------------------------------------------------------

      call gphite( p, t, w, 0.0, n_levels, i_dir, z)

c-----------------------------------------------------------------------
c     -- Set loop bounds for direction sensitive calculations --
c     -- so loop iterates from surface to the top             --
c-----------------------------------------------------------------------

      if( i_dir .gt. 0 )then

c        --------------------
```

```fortran
c        Data stored top down
c        --------------------

         l_start = n_levels
         l_end   = 1

      else

c        --------------------
c        Data stored bottom up
c        --------------------

         l_start = 1
         l_end   = n_levels

      end if

c-----------------------------------------------------------------------
c           -- Air molecular mass and density, and gravity --
c           -- as a function of LEVEL                       --
c-----------------------------------------------------------------------

c     ----------------------
c     Loop from bottom to top
c     ----------------------

      do l = l_start, l_end, -1*i_dir

c        --------------------------------
c        Convert water vapour g/kg -> ppmv
c        --------------------------------

         w_ppmv(l) = 1.0e+03 * w(l) * mw_dryair / mw_h2o

c        ------------------------------------------
c        Calculate molecular weight of air (g/mol)
c        ------------------------------------------

         mw_air(l) = ( ( 1.0 - (w_ppmv(l)/1.0e+6) ) * mw_dryair ) +
     +               ( ( w_ppmv(l)/1.0e+06 ) * mw_h2o )

c        ----------------
c        Air mass density
c        ----------------

         c(l) = 0.001 * mw_air(l) / R_air    ! 0.001 factor for g -> kg
         rho_air(l) = c(l) * p(l) / t(l)

c        -------
c        Gravity
c        -------

         r_hgt = r_avg + z(l)                      !  m
         g(l) = g_sfc -                            !  m/s^2
     +          g_sfc*( 1.0 - ( (r_avg*r_avg)/(r_hgt*r_hgt) ) ) )

      end do

c-----------------------------------------------------------------------
c                     -- LAYER quantities --
c-----------------------------------------------------------------------

c     ----------------------
c     Loop from bottom to top
c     ----------------------
```

```fortran
      do l = l_start, l_end+i_dir, -1*i_dir

c         ----------------------------------------------------------
c         Determine output array index.  This is done so that the
c         output data is always ordered from 1 -> L-1 regardless
c         of the orientation of the input data.  This is true by
c         default only for the bottom-up case.  For the top down
c         case no correction would give output layers from 2 -> L
c         ----------------------------------------------------------

          if( i_dir .gt. 0 )then

            l_indx = l - 1

          else

            l_indx = l

          end if

c         ---------------------------------------
c         Assign current layer boundary densities
c         ---------------------------------------

          rho1 = rho_air(l)
          rho2 = rho_air(l-i_dir)

c         ---------
c         Average c
c         ---------

          c_avg = ( (rho1*c(l)) + (rho2*c(l-i_dir)) ) / ( rho1 + rho2 )

c         ---------
c         Average t
c         ---------

          t_avg(l_indx) =
     +          ( (rho1*t(l)) + (rho2*t(l-i_dir)) ) / ( rho1 + rho2 )

c         ---------
c         Average p
c         ---------

          p1 = p(l)
          p2 = p(l-i_dir)

          z1 = z(l)
          z2 = z(l-i_dir)

          dp = p2 - p1

          A = log(p2/p1) / (z2-z1)
          B = p1 / exp(A*z1)

          p_avg(l_indx) = dp / log(p2/p1)

c         -----------------------------------------------
c         LAYER thickness (rather long-winded as it is not
c         assumed the layers are thin) in m. Includes
c         correction for altitude/gravity.
c         -----------------------------------------------

c         ...Initial values
```

```
          g_avg = g(l)
          dz = -1.0 * dp * t_avg(l_indx) / ( g_avg*c_avg*p_avg(l_indx) )

c         ...Calculate z_avg
          z_avg = z(l) + ( 0.5*dz )

c         ...Calculate new g_avg
          r_hgt = r_avg + z_avg
          g_avg = g_sfc - g_sfc*( 1.0 - ( (r_avg*r_avg)/(r_hgt*r_hgt) ) )

c         ...Calculate new dz
          dz = -1.0 * dp * t_avg(l_indx) / ( g_avg*c_avg*p_avg(l_indx) )

c         ----------------------------------------
c         Calculate LAYER amounts for water vapour
c         ----------------------------------------

          w1 = w_ppmv(l)
          w2 = w_ppmv(l-i_dir)

          w_avg =  ( (rho1*w1) + (rho2*w2) ) / ( rho1+rho2 )

          w_amt(l_indx) =
     +          rho_ref * w_avg * dz * p_avg(l_indx) / t_avg(l_indx)

c         -------------------------------
c         Calculate LAYER amounts for ozone
c         -------------------------------

          o1 = o(l)
          o2 = o(l-i_dir)

          o_avg =  ( (rho1*o1) + (rho2*o2) ) / ( rho1+rho2 )

          o_amt(l_indx) =
     +          rho_ref * o_avg * dz * p_avg(l_indx) / t_avg(l_indx)

        end do

       return
       end


       subroutine gphite( p, t, w, z_sfc, n_levels, i_dir, z)
c ... version of 19.09.96

c PURPOSE:

c  Routine to compute geopotential height given the atmospheric state.
c     Includes virtual temperature adjustment.

c CREATED:

c  19-Sep-1996 Received from Hal Woolf

c  ARGUMENTS:

c     Input
c     --------
c        p      - REAL*4 pressure array (mb)

c        t      - REAL*4 temperature profile array (K)

c        w      - REAL*4 water vapour profile array (g/kg)
```

```
c     z_sfc   - REAL*4 surface height (m).  0.0 if not known.

c    n_levels - INT*4 number of elements used in passed arrays

c     i_dir   - INT*4 direction of increasing layer number

c                   i_dir = +1, Level(1) == p(top)      } satellite/AC
c                               Level(n_levels) == p(sfc) }    case

c                   i_dir = -1, Level(1) == p(sfc)      } ground-based
c                               Level(n_levels) == p(top) }    case

c     Output
c     --------
c      z     - REAL*4 pressure level height array (m)

c COMMENTS:

c   Dimension of height array may not not be the same as that of the
c     input profile data.

c=============================================================================

c-----------------------------------------------------------------------------
c              -- Prevent implicit typing of variables --
c-----------------------------------------------------------------------------

      implicit none

c-----------------------------------------------------------------------------
c                           -- Arguments --
c-----------------------------------------------------------------------------

c -- Arrays

      real*4    p(*), t(*), w(*),
     +          z(*)

c -- Scalars

      integer*4 n_levels, i_dir

      real*4    z_sfc

c-----------------------------------------------------------------------------
c                         -- Local variables --
c-----------------------------------------------------------------------------

c -- Parameters

      real*4    rog, fac
      parameter ( rog = 29.2898,
     +            fac = 0.5 * rog )

c -- Scalars

      integer*4 i_start, i_end, l

      real*4    v_lower, v_upper, algp_lower, algp_upper, hgt

c****************************************************************************
c                      ** Executable code **
c****************************************************************************

c-----------------------------------------------------------------------------
```

```fortran
c  -- Calculate virtual temperature adjustment and exponential     --
c  -- pressure height for level above surface.  Also set integration --
c  -- loop bounds                                                    --
c-----------------------------------------------------------------------

      if( i_dir .gt. 0 )then

c         -------------------
c         Data stored top down
c         -------------------

          v_lower = t(n_levels) * ( 1.0 + ( 0.00061 * w(n_levels) ) )

          algp_lower = alog( p(n_levels) )

          i_start = n_levels-1
          i_end   = 1

      else

c         --------------------
c         Data stored bottom up
c         --------------------

          v_lower = t(1) * ( 1.0 + ( 0.00061 * w(1) ) )

          algp_lower = alog( p(1) )

          i_start = 2
          i_end   = n_levels

      end if

c-----------------------------------------------------------------------
c                     -- Assign surface height --
c-----------------------------------------------------------------------

      hgt = z_sfc

c-----------------------------------------------------------------------
c               -- Loop over layers always from sfc -> top --
c-----------------------------------------------------------------------

      do l = i_start, i_end, -1*i_dir

c         -----------------------------------------------------
c         Apply virtual temperature adjustment for upper level
c         -----------------------------------------------------

          v_upper = t(l)
          if( p(l) .ge. 300.0 )
     +      v_upper = v_upper * ( 1.0 + ( 0.00061 * w(l) ) )

c         ----------------------------------------------------
c         Calculate exponential pressure height for upper layer
c         ----------------------------------------------------

          algp_upper = alog( p(l) )

c         ----------------
c         Calculate height
c         ----------------

          hgt = hgt + ( fac*(v_upper+v_lower)*(algp_lower-algp_upper) )
```

```
c        -------------------------------
c        Overwrite values for next layer
c        -------------------------------

         v_lower = v_upper
         algp_lower = algp_upper


c        ----------------------------------------------
c        Store heights in same direction as other data
c        ----------------------------------------------

         z(l) = hgt

       end do

       return
       end



       subroutine clmozo(rlat,imon, omix)
c * Obtain climatological ozone mixing-ratio profile by
c      interpolating in latitude and month amongst the
c      FASCODE model atmospheres.
c .... version of 17.02.98
c * pressure coordinate is satellite-standard

c   input:
c        rlat = latitude (0-90,+N,-S)
c        imon = month    (1,...,12)

c   output:
c         omix = ozone mixing ratio profile (ppmv)

       parameter (nl=40,ns=2,nz=3)
       dimension tlat(nz),ozmr(nl,nz,ns),omr(nl,ns),omix(nl)
       data tlat/15.,40.,65./
c * tropical
        data (ozmr(i,1,1),i=1,nl)/
     + 0.55927,0.98223,1.94681,3.13498,4.30596,5.48908,7.41904,8.55498,
     + 9.22089,9.76594,9.60463,8.45820,6.99682,5.57585,4.30000,1.69985,
     + 1.23555,0.81780,0.39171,0.20944,0.14056,0.12283,0.10984,0.08405,
     + 0.06529,0.05392,0.04764,0.04366,0.04232,0.04052,0.03961,0.03735,
     + 0.03595,0.03534,0.03514,0.03385,0.03252,0.03107,0.03027,0.02901/
c * midlatitude winter
        data (ozmr(i,2,1),i=1,nl)/
     + 0.58382,1.06610,2.23416,3.87594,5.18854,6.20949,7.04493,7.17104,
     + 7.10972,6.86107,6.29020,5.71788,5.35257,5.03882,4.57679,3.16918,
     + 2.47482,1.95801,1.43279,1.11336,0.93068,0.81309,0.74765,0.46038,
     + 0.25869,0.15587,0.10257,0.07960,0.06900,0.05625,0.05211,0.04120,
     + 0.03513,0.03297,0.03176,0.02883,0.02821,0.02796,0.02790,0.02781/
c * subarctic winter
        data (ozmr(i,3,1),i=1,nl)/
     + 0.75492,1.20217,2.39283,3.75644,4.96742,5.64285,6.17910,6.22151,
     + 6.15197,5.88338,5.42542,4.91094,4.76030,4.63605,4.52229,3.70528,
     + 3.01350,2.39073,1.76424,1.38778,1.12046,0.82870,0.66060,0.36047,
     + 0.28088,0.17023,0.09235,0.06541,0.05169,0.04171,0.03941,0.03409,
     + 0.03095,0.02819,0.02673,0.02330,0.02159,0.01999,0.01933,0.01828/
c * tropical
        data (ozmr(i,1,2),i=1,nl)/
     + 0.55927,0.98223,1.94681,3.13498,4.30596,5.48908,7.41904,8.55498,
     + 9.22089,9.76594,9.60463,8.45820,6.99682,5.57585,4.30000,1.69985,
     + 1.23555,0.81780,0.39171,0.20944,0.14056,0.12283,0.10984,0.08405,
     + 0.06529,0.05392,0.04764,0.04366,0.04232,0.04052,0.03961,0.03735,
     + 0.03595,0.03534,0.03514,0.03385,0.03252,0.03107,0.03027,0.02901/
c * midlatitude summer
```

```fortran
      data (ozmr(i,2,2),i=1,nl)/
     + 0.61951,1.07099,1.77685,2.91535,3.98547,5.06939,7.01746,8.18549,
     + 8.74393,8.73998,7.87205,6.65253,5.84694,5.12966,4.37609,2.46410,
     + 1.97307,1.47696,0.91259,0.66705,0.57759,0.48610,0.44729,0.24487,
     + 0.16974,0.12153,0.10001,0.08397,0.07669,0.06661,0.06225,0.05355,
     + 0.04892,0.04505,0.04291,0.03815,0.03517,0.03283,0.03194,0.03053/
c * subarctic summer
      data (ozmr(i,3,2),i=1,nl)/
     + 0.58825,0.98312,1.64271,2.52996,3.56841,4.56575,6.36529,7.39635,
     + 7.78289,7.56976,6.69058,5.57509,5.21478,4.73043,4.34708,2.75529,
     + 2.05541,1.64657,1.17452,0.92611,0.78889,0.65317,0.58224,0.40198,
     + 0.24128,0.15314,0.10010,0.08052,0.07394,0.06544,0.06065,0.04986,
     + 0.04448,0.04090,0.03887,0.03446,0.03129,0.02823,0.02682,0.02456/

      alat=abs(rlat)
      if(alat.gt.15.) go to 110
      jl=1
      kk=1
      go to 120
  110 if(alat.lt.65.) go to 130
      jl=3
      kk=2
  120 do k=1,kk
         do i=1,nl
            omr(i,k)=ozmr(i,jl,k)
         enddo
      enddo
      if(kk.ne.1) go to 140
      do i=1,nl
         omix(i)=omr(i,1)
      enddo
      go to 150
  130 jl1=1
      if(alat.gt.40.) jl1=2
      jl2=jl1+1
      wt1=(tlat(jl2)-alat)/25.
      wt2=1.-wt1
      do k=1,ns
         do i=1,nl
            or1=ozmr(i,jl1,k)
            or2=ozmr(i,jl2,k)
            omr(i,k)=wt1*or1+wt2*or2
         enddo
      enddo
  140 nmon=imon
      if(rlat.lt.0.) nmon=nmon+6
      if(nmon.gt.12) nmon=nmon-12
      kmon=iabs(nmon-7)
      wt1=float(kmon)/6.
      wt2=1.-wt1
      do i=1,nl
         omix(i)=wt1*omr(i,1)+wt2*omr(i,2)
      enddo
  150 return
      end

      function satmix(p,t)
c * Saturation mixing ratio
c .... version of 18.02.98

      if(t.gt.253.) then
         es=svpwat(t)
      else
         es=svpice(t)
      endif
```

```
      satmix=622.*es/p

      return
      end

      function svpice(temp)
c * Saturation vapor pressure over ice
c .... version of 18.02.98

c **** temp must be in degrees Kelvin

      real*8 t,a0,a1,a2,a3,a4,a5
      data a0/.7859063157d0/,a1/.3579242320d-1/,a2/-.1292820828d-3/,
     *    a3/.5937519208d-6/,a4/.4482949133d-9/,a5/.2176664827d-10/
      t=temp-273.16
      e=a0+t*(a1+t*(a2+t*(a3+t*(a4+t*a5))))
      svpice=10.**e
      return
      end

      function svpwat(temp)
c * Saturation vapor pressure over water
c .... version of 18.02.98

c **** temp must be in degrees Kelvin

      real*8 b,s,t,a0,a1,a2,a3,a4,a5,a6,a7,a8,a9
      data a0/.999996876d0/,a1/-.9082695004d-2/,a2/.7873616869d-4/,
     *  a3/-.6111795727d-6/,a4/.4388418740d-8/,a5/-.2988388486d-10/,
     *  a6/.2187442495d-12/,a7/-.1789232111d-14/,a8/.1111201803d-16/,
     *  a9/-.3099457145d-19/,b/.61078d+1/
      t=temp-273.16
      s=a0+t*(a1+t*(a2+t*(a3+t*(a4+t*(a5+t*(a6+t*(a7+t*(a8+t*a9))))))))
      s=b/s**8
      svpwat=s
      return
      end

      subroutine taugen(nc,nx,ny,cc,xx,tau)
c ... general transmittance calculation ... Strow-Woolf model
c ... version of 19.09.96

      dimension cc(nc,ny),xx(nx,ny),tau(*)
      data trap/-999.99/

        taul=1.
        tau(1)=taul
      do 100 j=1,ny
      if(taul.eq.0.) go to 100
      yy=cc(nc,j)
      if(yy.eq.trap) then
         taul=0.
         go to 100
      endif
      do i=1,nx
         yy=yy+cc(i,j)*xx(i,j)
      enddo
      tauy=taul*exp(-yy)
      taul=amin1(tauy,taul)
  100 tau(j+1)=taul
      return
      end

      subroutine tauwet(ncs,ncl,nxs,nxl,nxw,ny,ccs,ccl,xx,tau)
c ... version of 17.02.98
```

```fortran
      dimension ccs(ncs,ny),ccl(ncl,ny),xx(nxw,ny),tau(*)
      data trap/-999.99/

        odsum=0.
        taul=1.
        tau(1)=taul
      do 100 j=1,ny
      if(taul.eq.0.) go to 100
        if(odsum.lt.5.) then
         yy=ccs(ncs,j)
         if(yy.eq.trap) then
            taul=0.
            go to 100
         endif
         do i=1,nxs
            yy=yy+ccs(i,j)*xx(i,j)
         enddo
           odsum=odsum+abs(yy)
        else
           yy=ccl(ncl,j)
           if(yy.eq.trap) then
              taul=0.
            go to 100
         endif
         do i=1,nxl
            yy=yy+ccl(i,j)*xx(i+11,j)
         enddo
           odsum=odsum+abs(yy)
        endif
      tauy=taul*exp(-yy)
      taul=amin1(tauy,taul)
  100 tau(j+1)=taul
      return
      end



        subroutine vasrtw(tbbv,zena,kmon,ksat, t,w,*)
c * Regression temperature & water-vapor mixing ratio
c       from VAS (GOES-4, -5, -6, -7)
c .... version of 18.02.98

c * Inputs:
c       tbbv = VAS brightness temperatures (degK), channels 1-12
c       zena = local (satellite) zenith angle (deg)
c       kmon = numerical month (1...12)
c       ksat = GOES satellite number (4...7)
c * Outputs:
c       t = temperature profile (degK) at 40 standard levels
c       w = water-vapor mixing ratio profile (g/kg) at same levels
c       * = error return for I/O problems

      parameter (nk=12,nm=12,nt=40,nv=20)
      parameter (nx=nk+1,nc=nx+1,ny=nt+nv)
      parameter (isat=4,nsat=7,iuc=77,lenc=nc*ny*4)
      dimension tbbv(*),t(*),w(*)
      dimension coef(nc,ny,nm,isat:nsat),p(nt),x(nx),y(ny)
      character*12 cfile/'vasregtw.dat'/
      data init/1/,wmin/0.003/
     data p/0.1,0.2,0.5,1.0,1.5,2.,3.,4.,5.,7.,10.,15.,20.,25.,30.,
    + 50.,60.,70.,85.,100.,115.,135.,150.,200.,250.,300.,350.,400.,
    + 430.,475.,500.,570.,620.,670.,700.,780.,850.,920.,950.,1000./
     secant(z) = 1./cos(0.01745329*z)

        if(init.ne.0) then
```

```fortran
c * store coefficients for all months and satellites
        open(iuc,file=cfile,recl=lenc,access='direct',
     +       status='old',err=100)
        krec=0
        do l=isat,nsat
          do k=1,nm
            krec=krec+1
            read(iuc,rec=krec) ((coef(i,j,k,l),
     *                           i=1,nc),j=1,ny)
          enddo
        enddo
        close(iuc)
        init=0
      endif

c * define predictors
      do i=1,nk
        x(i)=tbbv(i)
      enddo
      x(nx)=10.*secant(zena)+200.

c * obtain predictands
      k=kmon
      l=ksat
      do j=1,ny
        sum=coef(nc,j,k,l)
        do i=1,nx
          sum=sum+x(i)*coef(i,j,k,l)
        enddo
        y(j)=sum
      enddo

c * extract temperature profile
      do j=1,nt
        t(j)=y(j)
      enddo
c * extract mixing ratio profile
      do j=1,nv
        w(j)=wmin
        w(j+nv)=y(j+nt)
      enddo
c * guard against supersaturation
      do j=nv+1,nt
        wmax=satmix(p(j),t(j))
        w(j)=amin1(w(j),wmax)
      enddo

      return
100   return1
      end
```