# A VALIDATION STUDY OF THE SSM/I TEMPERATURE ALGORITHM AND COMPARISON WITH THE CAL/VAL LAND SURFACE TEMPERATURES

THESIS

Ronald L. Comoglio, Captain, USAF

AFIT/ENP/GM97M

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DTIC QUALITY INSPECTED 1

AFIT/GM/ENP/97M-02

# A VALIDATION STUDY OF THE SSM/I TEMPERATURE ALGORITHM AND COMPARISON WITH THE CAL/VAL LAND SURFACE TEMPERATURES

THESIS

Ronald L. Comoglio, Captain, USAF

AFIT/ENP/GM97M

19970402 085

Approved for public release; distribution unlimited

AFIT/GM/ENP/97M-02

# A VALIDATION STUDY OF THE SSM/I TEMPERATURE ALGORITHM AND COMPARISON WITH THE CAL/VAL LAND SURFACE TEMPERATURES

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Atmospheric Physics

Ronald L. Comoglio, B.S.E.E.

Captain, USAF

March 1997

Approved for public release, distribution unlimited

AFIT/GM/ENP/97M-02

# A VALIDATION STUDY OF THE SSM/I TEMPERATURE ALGORITHM AND COMPARISON WITH THE CAL/VAL LAND SURFACE TEMPERATURES

Ronald L. Comoglio, B.S.E.E.
Captain, USAF

Approved:

| | |
|---|---|
| Maj Clifton E. Dungey, Chairman | 11 MAR 97<br>date |
| Dr Thomas J. Kopp | 11 Mar 97<br>date |
| Maj Derrill T. Goldizen | 12 Mar 97<br>date |

## Acknowledgments

My first acknowledgment is to my Lord, Christ Jesus, without whom nothing in the realm of my life makes anywhere near as much sense nor carries near as much meaning without His guiding touch. He blesses me with health and the ability to study and conduct the research gathered here.

My advisor, Maj Clifton E. Dungey, allowed me the latitude to define this project and conduct it with precisely timed "course corrections" and occasional key FORTRAN assists. It ran, it ran! Additional code writing indebtedness goes to Lt Col Michael K. Walters, Maj Jason P. Tuell, and Capt Jay DesJardins.

Assistance during the literature searching was provided by Maj Derrill T. Goldizen, and the AFIT Academic Library staff were exceptionally helpful.

Statistical aid was expertly administered by Professor Dan Reynolds; eliminating a difficult "stall point" in the research.

The initial coding, satellite channel data, and other pertinent information was provided by Dr Thomas J. Kopp and Capt Louis Cantrell, of the Cloud Modeling Branch, Air Force Global Weather Center, Offutt AFB, Nebraska. I am sure my many questions were at times vexing, but the response was always sure and quick. Maj Thomas Neu, HQ AFMC/DOW, also helped to clarify many of the AFGWC processes for me during the concept definition.

The hardware gurus of the Atmospheric Sciences Laboratory, Department of Physics, AFIT, MSgt Peter J. Rahe and Mr. Russ Milliron were also much challenged and also provided immediate and stellar support. Much to their amazement I finished the number crunching with no equipment fatalities.

To the guys of the premier Combat Weather Class, GM-97M, I salute you and thank you for the camaraderie and support throughout the tour.

My last debt of gratitude I reserve for my family, my wife Beth and children (Heather, Jonathan, and Chiara): Daddy is finally finished and looking forward to leaving his work at work again!

Ronald L. Comoglio

Table of Contents


| | Page |
|---|---|
| Acknowledgments | iii |
| List of Figures | vi |
| List of Tables | vii |
| Abstract | viii |
| I. Introduction | 1 |
| Background | 1 |
| Statement of the Problem | 2 |
| II. Literature Review | 4 |
| Introduction | 4 |
| Remote Sensing Overview | 4 |
| Atmospheric Effects | 8 |
| Data Retrieval Techniques | 14 |
| DMSP Overview | 16 |
| Cloud Modeling in the USAF | 18 |
| III. Methodology | 21 |
| Introduction | 21 |
| Download Phase | 21 |
| Grid Transformation | 22 |
| Model Discussion | 23 |
| The CAL/VAL Model | 24 |
| The TMPSMI Model | 25 |
| Model Calculations and Output | 27 |
| IV. Results and Data Analysis | 34 |
| Introduction | 34 |
| August Results | 35 |
| October Results | 39 |
| V. Conclusions and Recommendations | 44 |
| Conclusions | 44 |
| Recommendations | 45 |

Table of Contents


| | Page |
|---|---|
| Appendix A: TMPSMI Code | 48 |
| Appendix B: CAL/VAL Code | 65 |
| Bibliography | 72 |
| Vita | 74 |

## List of Figures


| Figure | Page |
|---|---|
| 1. Planck Curve with Rayleigh-Jean's Approximation. The frequency range for the SSM/I sensor is portrayed between the verticals $f_l$ and $f_u$. | 6 |
| 2. A simple geometry for aerial remote sensing. | 10 |
| 3. Problem geometry with satellite height and atmosphere included. | 13 |
| 4. The Northern Hemisphere RTNEPH Grid Scheme. | 20 |
| 5. Schematic of Methodology Process. The output format is as follows: ICAO of observing station, the observed air temperature, the TMPSMI output, and then the CAL/VAL output. | 28 |
| 6. Super Neph gridding of points surrounding Malmstrom AFB, MT | 29 |
| 7. Nephboxes and CONUS military weather observing sites. | 31 |
| 8. Comparison plot of August temperatures. The circles are the observed temperatures, the diamonds indicate TMPSMI output values, and the crosses indicate the CAL/VAL output values. | 36 |
| 9. Probability Density Functions of August Theoretical Distributions. The solid line is the distribution of the differences (centered on -0.618), the dotted line is the distribution of TMPSMI departures from observed, and dashed is the distribution of CAL/VAL departures from observed. | 38 |
| 10. Comparison plot of October temperatures. The circles are the observed temperatures, the diamonds indicate TMPSMI output values, and the crosses indicate the CAL/VAL output values. Any data gaps indicate a zero value for the model output. | 40 |
| 11. Probability Density Functions of October Theoretical Distributions. The solid line is the distribution of the differences (centered on -4.798), the dotted is the line distribution of TMPSMI departures from observed, and the dashed line is the distribution of CAL/VAL departures from observed. | 43 |

## List of Tables


| Table | Page |
|---|---|
| 1. Microwave Imaging Frequencies | 17 |
| 2. CAL/VAL Land Surface Classifications | 25 |
| 3. TMPSMI Land Surface Classifications | 26 |
| 4. Example of Data Output | 32 |
| 5. August Statistics | 35 |
| 6. October Statistics | 39 |
| 7. Comparative Statistics | 42 |

## Abstract

Cloud modeling has been an emphasis at the Air Force Global Weather Center (AFGWC) for several decades. Our capability continues to improve by incorporating various data types to simulate the complex interaction of variables by which clouds form and advect. One of the keys to properly forecast these processes is through the initialization of the moisture fields over a synoptic area, a concept made possible by the extensive earth sensing satellite network.

The USAF operational cloud model, the Real-Time Nephanalysis (RTNEPH), uses standard infrared and visual scene scans as well as input from specialized sensors such as the Special Sensor Microwave Imager (SSM/I) featured on the Defense Meteorological Satellite Program (DMSP) polar orbiters. Of importance to the cloud modeler is the global surface temperature field to aid in identifying which pixel of a scene is cloud and which is a ground point. The SSM/I provides brightness temperatures sampled at four discrete frequencies to produce surface temperatures in data sparse regions of the world.

Two different algorithms produce surface temperatures from these brightness temperatures. One is currently in use at AFGWC as part of the Surface Temperature Model (SFCTMP); the other is an earlier model created by the U. S. Navy during the calibration and validation of the SSM/I sensor (a joint program), and is aptly named the CAL/VAL temperature algorithm. The problem as presented by AFGWC is to determine

how well the SFCTMP program performs against observed temperatures and to compare the results with CAL/VAL's performance.

This study uses remotely sensed brightness temperatures for the CONUS from days in August and October 1996, runs both algorithms and compares the output from these algorithms with collected U.S. reporting sites.

The results indicate both models produce similar output, exhibiting a cold bias with respect to the observed surface temperature. Additionally, the CAL/VAL temperatures approached the observed temperature values more closely than did the SFCTMP output.

# A VALIDATION STUDY OF THE SSM/I TEMPERATURE ALGORITHM AND COMPARISON WITH THE CAL/VAL LAND SURFACE TEMPERATURES

## I. Introduction

### Background

With the advent of space exploration in the late 1950s, the upper platform for remote sensing was elevated to space orbit levels. Meteorological emphasis began with the launch of Vanguard 2, the first weather satellite, on 17 February 1959, and reached a substantial advance with the genesis of the Defense Meteorological Satellite Program (DMSP), which launched its first polar orbiter on 16 September 1966 (Kidder and Vonder Haar, 1995). By the early 1990s, coverage of the earth was assured, with literally hundreds of man-made sensors in low earth or geosynchronous orbits providing measurements of every geophysical parameter imaginable.

Atmospheric satellite remote sensing uses different frequencies of the electromagnetic energy spectrum (e.g., infrared and microwave), and quantifies various facets of the atmosphere's composition. Examples of the different forms of this type of remote sensing would be active, when the sensor "beams" energy at the target and reads information from the return signal, versus passive, where the sensor surveys the scene and interprets the emitted blackbody radiances. Depending on the objective, sensors may scan cloud tops, the intervening profile between the earth surface and sensor, or look out "on the limb" for upper atmospheric constituents. To retrieve surface temperatures, the sensor must be able to look directly through the atmosphere (including clouds), requiring such frequencies which allow exploitation of atmospheric transmission windows as are

found in the microwave part of the spectrum. The emphasis here will thus be on the passive microwave type of radiometer.

This study attempts to determine which of two computer algorithms produces a better estimate of surface air temperature using remotely sensed brightness temperatures obtained from the Special Sensor Microwave Imager (SSM/I), in use on the DMSP satellites designated as F8, F10, F11, F12 and F13. This study incorporates only data measured by the SSM/I on satellite F13.

Both algorithms start with the SSM/I obtained brightness temperatures. The SSM/I radiometer functions in the passive mode using four frequencies, 19.35, 22.235, 37.0 and 85.5 GHz, each of which features a vertical and horizontal polarization on separate channels (except 22.235 GHz, which features only a vertical polarization). Upwelling brightness is recorded as a radiometer output voltage, then transmitted to AFGWC and the Fleet Numerical Meteorology and Oceanography Center (FNMOC), where it is converted to a field of sensor counts. The different data records often referenced in the literature: antenna temperature (TDR), brightness temperature (or Sensor Data Record, SDR) and derived geophysical parameters (Environmental Data Record, EDR) (Deuel, 1996) are all created from the sensor counts. To further complicate the matter, the SDR is not necessarily the surface temperature, but may represent a cloud or intervening level air temperature. One of the challenges of remote sensing is to de-convolve the data to retrieve the derived parameters correctly.

## Statement of the Problem

Since its inception, the USAF Air Weather Service has focused its efforts to better understand and model atmospheric processes, leading to theoretically improving weather support to military operations. As part of that effort, the DMSP satellites have been continuously streaming huge amounts of data to the Air Force Global Weather Center (AFGWC) at Offutt AFB, Nebraska. This data then feeds into several operational

applications at AFGWC, one of which is the Surface Temperature model (SFCTMP), which, in turn, creates input for the cloud model known as the Real-Time Nephanalysis (RTNEPH ). The algorithm used by SFCTMP, called TMPSMI, is a follow-on algorithm to one originally contained in the Calibration/Validation (CAL/VAL) study conducted by the Naval Research Laboratory (NRL). This project will take a portion of SSM/I collected data, validate the TMPSMI algorithm, and provide some basis of comparison with the CAL/VAL algorithm.

Surface temperature was not an early emphasis in the field of passive microwave remote sensing or an originally planned application of the SSM/I (Hollinger, 1983). Instead, such variables as liquid water content, inferred precipitation rates, and sea/ice determinations drove early research. These variables are firm U.S. Navy operational requirements. By the time of the CAL/VAL final report, the NRL had added a surface temperature algorithm which classified the land surface according to vegetation and soil type, then applied a temperature calculation based on empirically derived relationships (McFarland, 1991). That validation study heavily depended on climatology in the form of the Environmental Data Record (EDR).

The SFCTMP model was developed at AFGWC in 1988, but was revised in 1990 to its present form. As required inputs for the RTNEPH, it provides a skin temperature and a "shelter" temperature at each grid point for model run times every three hours. The term "shelter" indicates the ambient temperature independent of direct solar radiation effects, as measured at the standard World Meteorological Organization level of two meters. RTNEPH uses the temperatures in an infrared-based threshold technique to determine if clouds are present; however, RTNEPH also provides feedback to the SFCTMP in the form of resultant cloud information and clear sky infrared temperatures (Kopp, 1995). Clearly, the quality of the SFCTMP output directly affects AFGWC's ability to forecast clouds, and any improvements to surface temperature determination will result in improvement to the final RTNEPH model.

# II. Literature Review

## Introduction

Allowing synoptic scale weather features to be contained within a single sensor's scope requires environmental remote sensing from space altitudes. This paper assumes the reader has a working familiarity with space satellite operations through the early space age period (1964 until 1980) so the operation and usage of these platforms in the visible and infrared spectrums is understood fundamentally.

The desire to see "through the weather" limited the exploitation of the visual and infrared bandwidths and led to much experimentation with radar and microwave frequencies, a nice synopsis of which may be found in Ulaby (1981, section 2.5). The path to this paper's subject leads through a brief review of the science of geophysical remote sensing, including data retrieval techniques, an analysis of the DMSP SSM/I operation, and then the background necessary to understanding the applications in cloud modeling of remotely sensed data at AFGWC.

## Remote Sensing Overview

The initial basic principle is that all matter radiates energy in the electromagnetic realm, by virtue of that matter having a thermometric temperature. The accompanying theory of radiative transfer relates how absorption and emission raise and lower that temperature, respectively, and then how that energy propagates through some medium

(Ulaby, 1981). Remote sensing attempts to quantify the energy exchange from some distance.

Electromagnetic radiation has been exhaustively studied since Planck (1901) published his radiation law. Planck's law describes blackbody, or perfectly absorbing or emitting, radiation at given temperature and frequency by:

$$B_\nu = \frac{2h\nu^3}{c^2}\frac{1}{e^{\frac{h\nu}{kT}}-1} \quad , \tag{1}$$

where $B_\nu$ is the frequency specific brightness (frequently referred to as radiance), h is Planck's constant, k is Boltzmann's constant, c is the speed of light, ν is the frequency, and T is the temperature. For microwave and radio frequencies, (typically 5-100 GHz) the above equation simplifies to:

$$B_\nu = \frac{2\nu^2 kT}{c^2} \quad . \tag{2}$$

This equation simplifies by substitution to the Rayleigh-Jeans law, which is an excellent approximation, as shown by Figure 1, for terrestrial temperatures of about 300 K and frequencies less than 117 GHz (Ulaby, 1981). The accuracy of this expression relative to Planck's law is "better than 1% for objects at 300 K if the frequency is below about 125 GHz" (Rees, 1990).

The concept of brightness temperature, $T_B$, may be introduced from the last equation by algebraically manipulating equation (2):

$$T_B = \frac{B_\nu \lambda^2}{2k} \tag{3}$$

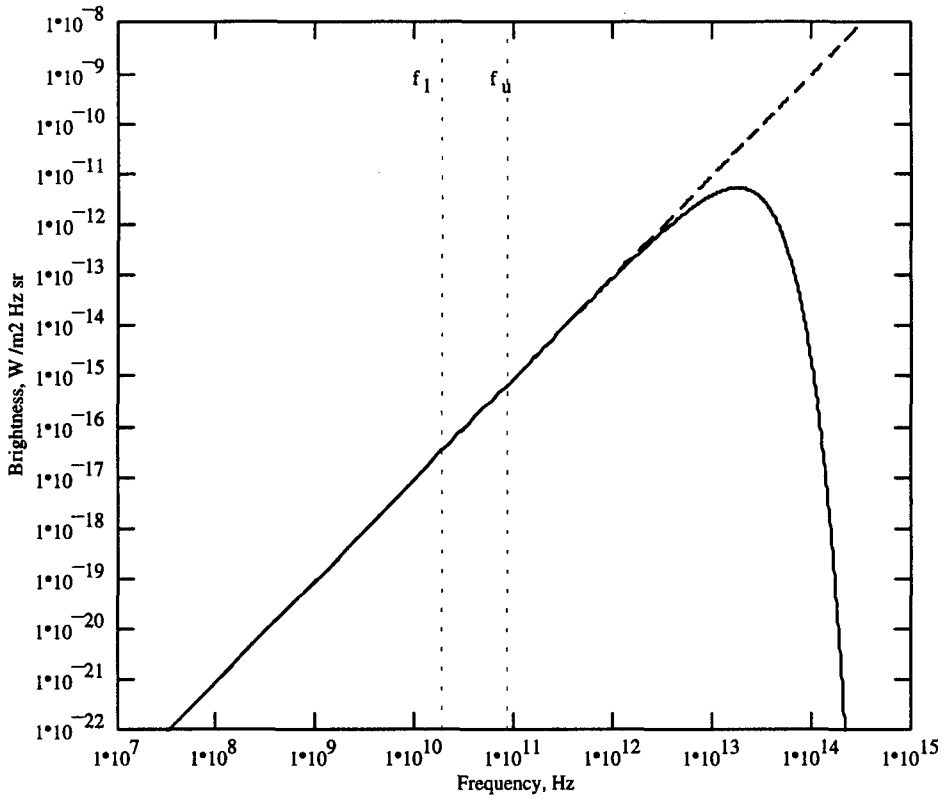


Figure 1. Planck Curve with Rayleigh-Jeans Approximation. The frequency range for the SSM/I sensor is portrayed between the verticals $f_l$ and $f_u$.

and defining brightness over a narrow frequency bandwidth ($\Delta f$) as follows:

$$B_v = \frac{2kT_B}{\lambda^2}\Delta f \quad . \tag{4}$$

By recognizing directional dependence,

$$B(\theta,\phi) = \frac{2k}{\lambda^2}T_B(\theta,\phi)\Delta f \quad , \tag{5}$$

where $T_B(\theta,\phi)$ is the brightness temperature, which is the entity the space-borne radiometer records. If the physical temperature of an object is known, its emissivity is given as the ratio of the brightness temperature to the physical temperature, assuming a homogeneous material with uniform temperature (Ulaby, 1981).

As with most scientific endeavors, complexities arise. The radiative transfer equation describing this scenario is overly simplified and practically useless because it assumes a transparent atmosphere and a perfect collection instrument. Antenna design and atmospheric attenuation are considerably more complicated. The atmospheric effects will be considered shortly.

The antenna theory is well covered in a variety of textbooks. Electrical circuit analogies commonly relate the thermal noise generated in receiving antennas to a resistance characteristic related to a certain temperature, called $T_A$. This thermal noise is "in essence caused by the Brownian motion of electrons in the resistance, and is often referred to as Johnson noise or Nyquist noise" (Rees, 1990). Thus, $T_A$ is commonly referred to as the radiometric antenna temperature and is quantified by the following (Ulaby, 1981):

$$T_A = \eta_m T_{ML} + (1 - \eta_m) T_{SL} \quad . \tag{6}$$

This equation brings in a directivity function where the antenna does not operate isotropically but in a preferred direction. This entails separating the main lobe and side lobe effects and $\eta_m$ is defined as the main beam efficiency, $T_{ML}$ is defined as the effective apparent temperature of the main lobe contribution and $T_{SL}$ is defined as the effective temperature contributed by side lobes, or off-axis power patterns.

The previous antenna temperature still assumed a lossless antenna however, and practice has shown the receiver "sees" effects from the antenna. Allowing $T_A'$ to be the lossy antenna temperature, Ulaby makes the following distinction:

> To the radar receiver, $T_A'$ represents a noise contribution. To a radiometer, on the other hand, $T_A'$ is the signal containing the information about the emission characteristics of the scene under observation. Specifically, the objective in radiometric remote sensing is to relate the radiometer receiver output voltage . . . to the apparent temperature of the resolution cell delineated by the main beam of the antenna. (Ulaby, 1981)

It is the radiometric output voltage which is transmitted to the receiving centers at AFGWC and FNMOC (Deuel, 1996). The conversion of the output voltage into sensor data counts and then a meaningful set of parameters constitutes the art of satellite data retrieval, which will be examined in the next section.

## Atmospheric Effects

The intervening atmosphere affects the transmittance of the radiant energy by either scattering or absorbing it. This is known collectively as extinction, and the absence of extinction implies the media is a vacuum with perfect transmittance. To quantify these

effects, extinction coefficients are introduced, such that

$$\kappa_e = \kappa_a + \kappa_s \quad , \tag{7}$$

where $\kappa_e$ is the total extinction coefficient, $\kappa_a$ is the absorption coefficient, and $\kappa_s$ is the scattering coefficient.

The differential brightness over a distance dr is given by

$$dB_{extinction} = \kappa_e \, B \, dr \quad . \tag{8}$$

Furthermore, if there is a single scattering constituent, its albedo is given by

$$a = \kappa_s / \kappa_e \quad . \tag{9}$$

Since not all scattering deflects energy away from the radiometer and since air molecules which absorb energy also tend to emit it (by Kirchoff's Law and assuming local thermal equilibrium conditions), there are the scattering and absorbing source functions, $J_s$ and $J_a$, respectively, which act to increase the received energy. These make up the effective total source function, J, the components of which are related by

$$J = (1 - a) \, J_a + a \, J_s \quad . \tag{10}$$

Given the simple geometry of Figure 2, the differential brightness, dB, is

$$dB = B(r) - B(0) \tag{11}$$

where B(r) is the brightness at some height along the radial, and B(0) is the initial brightness at the source located at the origin of the coordinate system. Therefore, the differential brightness is the difference between the source function (gain) and the scattering and absorption functions (loss):

$$dB = \kappa_e \, J \, dr - \kappa_e \, B \, dr \, , \tag{12}$$

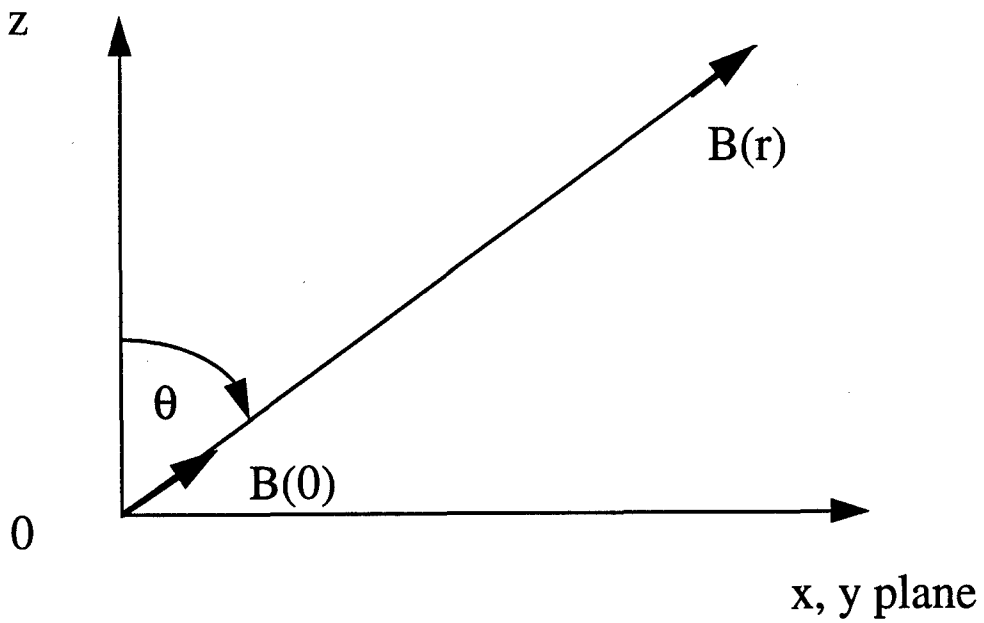


Figure 2: A Simple Geometry for Aerial Remote Sensing

which is usually written in the form of

$$dB = d\tau (J - B) . \tag{13}$$

Equation (13) introduces $d\tau$, which is the optical depth, and leads to the familiar form of the equation of transfer:

$$\frac{dB}{d\tau} = J - B . \tag{14}$$

Integration of the optical depth along the path yields optical thickness

$$\tau(0, r) = \int_0^r \kappa_e dr . \tag{15}$$

The optical thickness is particularly useful because the exponential of optical thickness is the transmittance, and is applied using Beer's Law. Thus, to calculate the received brightness at the end of the path, use

$$B(r) = B(0)e^{-\tau(0,r)} + e^{-\tau(0,r)} \int_0^r \kappa_e(r')J(r')e^{-\tau(0,r')}dr' . \tag{16}$$

Applying equation (3), because the measurement of interest is a temperature (not a brightness), to equation (16):

$$T_{AP}(r) = T_{AP}(0)e^{-\tau(0,r)} + e^{-\tau(0,r)} \int_0^r \kappa_e(r')[(1-a)T(r') + aT_{SC}(r')]e^{-\tau(0,r')}dr' . \tag{17}$$

$T_{AP}$ is the apparent radiometric temperature and $T_{SC}$ is described by the following. Note the source function is divided into $J_a$ and $J_s$ portions again, and

$$J_s(r) = \frac{2k}{\lambda^2} T_{SC}(r)\Delta f \quad , \tag{18}$$

where $T_{SC}$ is defined as the scattered radiometric temperature by

$$T_{SC} = \frac{1}{4\pi} \iint_{4\pi} \psi(r, r_i) T_{AP}(r_i) d\Omega_i \,, \tag{19}$$

and where $\Psi(r,r_i)$ is the phase function which accounts for energy scattered from radial $r_i$ into path r, and $d\Omega_i$ is the solid angle.

For microwave applications and under clear sky conditions, the intervening atmosphere is considered "scatter-free" (Ulaby, 1981):

$$T_{AP}(r) = T_{AP}(0) e^{-\tau(0,r)} + e^{-\tau(0,r)} \int_0^r \kappa_a(r') T(r') e^{-\tau(0,r')} dr' . \tag{20}$$

The scatter-free assumption is a good one even when some cloud is present, as cloud scattering is negligibly small, provided no raindrop-size particles are present. Raindrops have strong interactions with microwaves, which explains why one of the main applications of the SSM/I radiometer is in the field of remotely sensed precipitation rate retrieval. Clouds do weakly absorb microwave energy, so equation (20) is posed for clear sky case (Kidder and Vonder Haar, 1985).

If the geometry is modified for the upwelling temperature case (Figure 3, where H is the height of the radiometer), the transfer equation takes the form

$$T_{AP}(\theta, H) = T_{AP}(\theta, 0) e^{-\tau(0,H) \sec\theta} + \sec\theta \int_0^H \kappa_a(z') T(z') e^{-\tau(z',H') \sec\theta} dz' \tag{21}$$

and the second term defines the upwelling temperature.

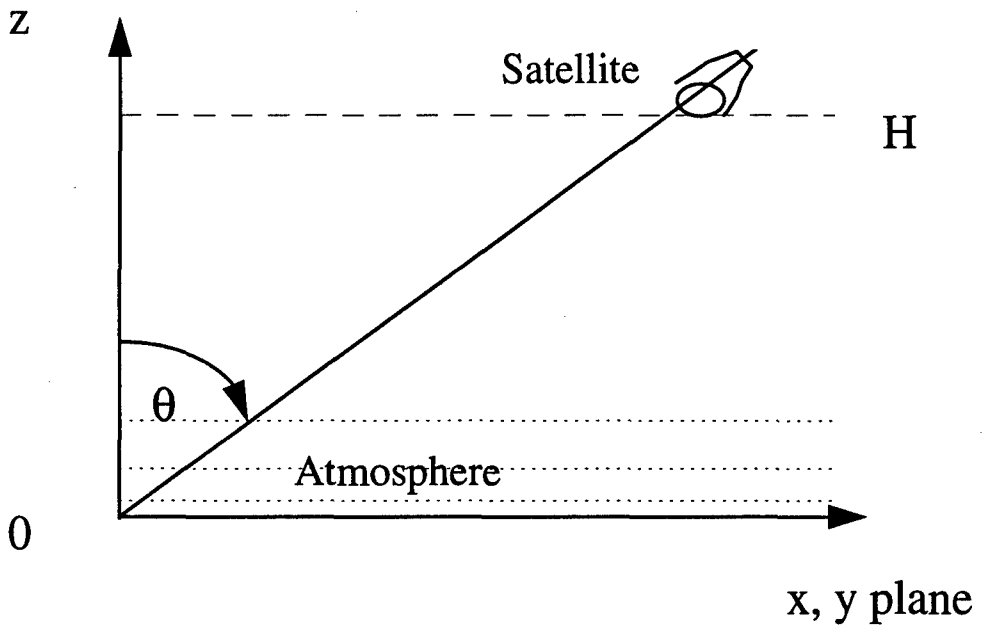


Figure 3: Problem geometry with satellite height and atmosphere included.

Data Retrieval Techniques

The idea behind data retrieval is illustrated by the general linear model of the form:

$$\underline{\mathbf{d}} = \mathbf{G}\,\underline{\mathbf{m}} \tag{22}$$

where $\underline{\mathbf{d}}$ is a vector of predicted values, $\underline{\mathbf{m}}$ is a vector of estimated values and **G** is a matrix which relates the two, usually called a data kernel (Menke, 1989). This analogy presupposes a linear relation and a Gaussian distribution.

The linearity assumption is easily supported for this case from the Rayleigh Jeans portion of the spectrum (Figure 1). If the operating frequencies in this study were in the infrared, the relation would be highly nonlinear (Ulaby, 1981).

Most atmospheric variables tend to be normally distributed, or at least exhibit the necessary symmetry that under application of the Central Limit Theorem of classical statistics allows for the Gaussian assumption (Wilks, 1995).

Those assumptions being met, this case requires $\underline{\mathbf{d}}$ to be the measured data vector (i.e., brightness temperatures), **G** to be the applicable data kernel matrix, and $\underline{\mathbf{m}}$ to be the vector of geophysical parameters attempting to be obtained (i.e., physical temperatures). **G** is constructed with available *a priori* information so **G** and **d** are known and **m** is solved by

$$\underline{\mathbf{m}} = \mathbf{G}^{-1}\,\underline{\mathbf{d}}\ . \tag{23}$$

and it is now apparent why this technique is referred to in the literature as the "inversion technique", owing to the matrix inversion of the data kernel.

The question arises as to the discrete nature of equations (22) and (23), since geophysical parameters are generally continuous. In a simple two-dimensional case, let G = G(x,y) and m = m(x) be continuous functions over a fixed interval (Twomey, 1977)

$$d(y) = \int_a^b G(x, y)m(x)dx \,. \tag{24}$$

To discretize introduces some error, ε, called the quadrature error,

$$d(y) + \varepsilon \cong \sum_{j=a}^{b} G_{ij} m(x_j) \tag{25}$$

which leads to the linear approximation, equation (22). Twomey shows that the errors associated with inverting the **G** matrix eigenvalues are far larger than the quadrature error, such that in most cases, ε is ignored (Twomey, 1977). Other sources of error are the random noise error, systematic error due to uncertainty in parameters or model bias, and "null-space error" due to the finite resolution of the observing sensor (Rodgers, 1990). For the use of this study, it is assumed the regression techniques minimize the random and systematic errors to the greatest extent possible. The null-space error is held in common by both algorithms and is therefore not a player in the comparison. Other errors particular to this treatment are discussed in Chapter 5.

Typically, the application of equation (23) uses the "statistical inversion method" to optimize the measurements (Ulaby, 1986). This method works by minimizing the variance of the unknown function, **m**. It involves a description of the estimated departure from some *a priori* mean value (usually a climatologically derived value), <x>,

$$\mathbf{m}' = \mathbf{m} - \langle \mathbf{m} \rangle \tag{26}$$

to implement in the linear model in the form

$$\mathbf{m}' = \mathbf{D}\,\mathbf{T_m}' \quad . \tag{27}$$

**D** is a matrix of regression coefficients (often simply called the D-matrix) and $\mathbf{T_m}'$ is the vector of observations with error taken into account (Ulaby, 1986). This is essentially the method taken by the SFCTMP model, which is introduced in the Cloud Modeling section.

DMSP Overview

The Defense Meteorological Satellite Program (DMSP), which operates the Department of Defense (DoD) polar orbiting satellites -- two of which are directly maintained by the USAF -- was declassified in 1972. Through the years, different suites of instruments have flown onboard the DMSP. Currently, two of the main operational sensors are the Special Sensor Microwave Imager (SSM/I) and the Operational Line Scanner (OLS), the latter of which provides the graphical swaths of visible and infrared scenes commonly used in operational meteorology. Both SSM/I and OLS also provide input to the AFGWC SFCTMP model.

The SSM/I sensor was developed as a joint Navy/AF project and is a passive microwave radiometer, receiving brightness temperatures at four frequencies (19.3, 22.2, 37.0, and 85.5 GHz). As mentioned in the previous chapter, each frequency features linear polarizations, vertical and horizontal, in each channel; except for 22.2 GHz, which is only vertically polarized (Hollinger, 1983). The suite of frequencies was chosen based

on the empirical applications each brings to the model. Imaging practices follow these guidelines for type of detection (Chen, 1985):

Table 1. Microwave Imaging Frequencies

| Spectral Band Center Frequency | Image Type Detected |
|---|---|
| 19.3 GHz | Surface Features, Mapping |
| 22.2 GHz | Atmospheric Water Content |
| 35.0 GHz | Surface Imaging |
| 37.0 GHz | Surface Ice, Oil |
| 90.0 GHz | Surface Imaging |

Therefore, it is apparent from the image types that the SSM/I operating frequencies were chosen judiciously, with emphasis on the sensor's ability to "probe the bottom" of the earth's atmosphere. The selection also had to account for avoiding ground interference from ground emissions.

The SSM/I data flow from satellite to ground software application follows this path: Scene brightness temperatures from the satellite "footprint" are received at the radiometer ($T_{AP}$ from equation (17)) and converted to output voltages and transmitted. AFGWC receives this data and converts it to a format of "sensor counts" and transfers it to gridpoints corresponding to the standard Real-Time Nephanalysis (RTNEPH) 1/8 mesh grid shown in Figure 4 (Kopp, 1994). The data used for this study was forwarded from AFGWC in this format. These sensor counts constitute the sensor data record, or SDR. Applying different algorithms produces environmental parameters of interest, or the environmental data record, or EDR. Examples of these parameters are the Navy

applications of retrieved wave height and sea surface temperatures. USAF interest has focused on the use of land surface temperatures to provide input to the RTNEPH (a global cloud model) in data sparse regions (Kopp, 1994). Briefly stated, that process takes the SSM/I brightness temperatures, runs them through the TMPSMI algorithm for gridded surface temperatures, takes that output through the SFCTMP, and then feeds into RTNEPH.

Cloud Modeling in the USAF

The currently used USAF cloud model is the RTNEPH ("neph" comes from the Greek word "nephos", for cloud). This model attempts to assimilate cloud data in real-time from all available sources. To do this, six major processors are required to merge conventional observations with raw multi-frequency satellite data to produce a timely, coherent process (Hamill, 1992).

One of these major programs is the Surface Temperature Model (SFCTMP), the current version of which has been operational since 1991, which is tasked with producing global surface (skin) and near-surface (shelter) temperatures. These temperatures then feed into the RTNEPH to make cloud/no cloud decisions and to provide infrared background thresholds (Kopp, 1995), as well as provide first guess initialization to other numerical weather prediction models. The model runs every three hours and requires surface weather observations as input. Conventional observing is not available worldwide, nor are the existing sites evenly spaced over the land masses, so the SSM/I was implemented to fill the gaps in the observing network. This is achieved by weighting

the influence of the SSM/I pass heavily in data sparse areas, and less heavily where conventional observing sites are dense (Kopp, 1994).

The task at hand is to quantify how well the SSM/I sensor and the TMPSMI algorithm does the job of producing usable surface temperatures, and to demonstrate whether it performs better than the CAL/VAL algorithm with identical input.

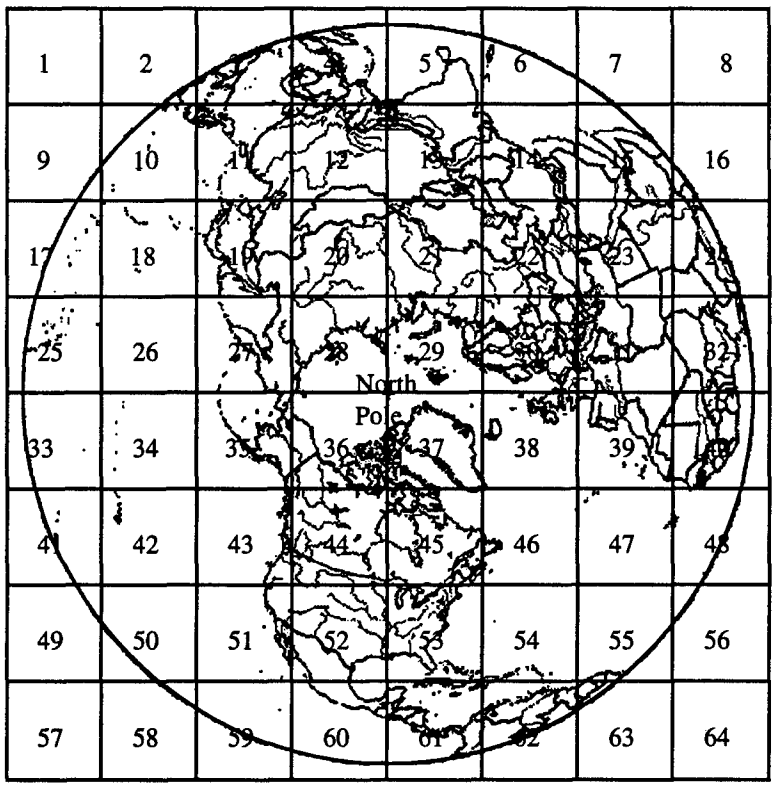


Figure 4: The Northern Hemisphere RTNEPH Grid Scheme

# III. Methodology

## Introduction

This chapter outlines the procedures followed to collect and analyze the data to address the question formulated in Chapter 1: How well does the AFGWC SFCTMP model perform? Specifically, the test concerns the TMPSMI algorithm within the SFCTMP model, which computes the temperature field from the SSM/I measured inputs. Comparisons are also made between CAL/VAL output and surface observations. Resulting performance conclusions will draw from each model's ability to match the temperature observed at the geographically closest observing site.

This methodology traces the path the data takes from the satellite through the respective models up to the point of operational implementation in the cloud models. The basic phases are 1) download, 2) grid transformation, 3) model discussion, and 4) model calculation/output.

## Download Phase

This part describes the process of the polar orbiting satellite in "beaming down" the seven measured brightness temperatures, hereafter referred to as the channel data. Polar orbiters complete a pass every 101 minutes (Hamill, 1992) and the swath width of the SSM/I radiometer is 1,394 km (Hollinger and Lo, 1983). Therefore, covering the CONUS region, the area of focus in this study, requires at least two passes with some overlap at higher latitudes and, conversely, data gaps at lower latitudes. This is noticeable even in CONUS latitudes. The scans sweep continuously along the orbit track by rotating

the antenna which features a constant incident angle of 53.1°, greatly simplifying the formulation of the model equations by eliminating any angle dependence.

Because special manual procedures were required, a single daily download was conducted at 0600 UTC for this study. This means the data has time stamps corresponding to the time of the last over pass, typically ranging back to 0000 UTC. This download consisted of the data from only one DMSP, satellite F13.

The download to tape occurred in the AFGWC process at the point between the mainframe processing and implementation in the modeling operations. This processing includes gridding, which fixes the resolution as described in the next section.

Grid Transformation

The SSM/I operation during the continuous scan is described as follows:

> The radiometer outputs are sampled differently on alternate scans. During the scene portion of the scans, the five lower frequency channels are each sampled over 64 equal 1.6° intervals and the two 85.5 GHz channels are each sampled over 128 equal 0.8° intervals or approximately each 11 km along the scan . . . thus the five lower channels are sampled on an approximate 25 km grid along the scan and along the track. The two 85.5 GHz channels are sampled at one half this spacing both cross and along track. (Hollinger, 1983)

For processing at AFGWC, this data is transferred to the RTNEPH standard grid, a polar stereographic map upon which is imposed 64 "neph boxes", each consisting of 64 X 64 analysis points (see Figure 4) per hemisphere. As noted in Chapter 2, this is also known as the eighth mesh grid. To select the CONUS area only and reduce the amount of information being manipulated, neph boxes 43-45 and 51-53 were pared from the data.

In doing the grid mapping, the SSM/I data loses resolution, degrading to 47.625 km, true at 60° latitude. This is fairly coarse, but is necessary due to computing hardware restraints. The mapping procedure also results in oversampling at low latitudes and undersampling at higher latitudes. This means the channel values in the tropics must be shared between a few gridpoints, while in the arctic there are not enough satellite samples for the number of gridpoints, so some are lost (Hamill, 1992).

While being gridded, the data is also transformed from sensor count format to Kelvin brightness temperatures. The received data from the download thus consists of seven brightness channel files, each numbering 262,144 values per hemisphere representing that frequency and polarization combination's contribution to the overall scene brightness, a file of the same number of date/time stamps, and a similar file to identify the contributing satellite for each day of the study.

The date/time stamp file proved crucial to the methodology as it gave the actual measurement time tagged by the satellite at which all the channels were measured for each individual gridpoint. This is important because although the download happened at 0600 UTC, the measurement times ranged back to whenever F13 had last sampled that gridspace. If this information were compromised, a valid comparison with a surface observation would be impossible.

Model Discussion

This section compares and contrasts the competing algorithms within this study, TMPSMI and CAL/VAL. The background to the development of these models is given in Chapter 1 of this study, however the method by which each produces a gridpoint surface

temperature has not been discussed to this point. The basic function of each will be presented and the scientific merit of the differing approaches will be inspected.

The CAL/VAL Model. CAL/VAL was developed first and operates on an empirical basis. The intent is to classify each SSM/I block by one of eight soil types, as listed in Table 2.

To develop empirical values, control regions which represented the land type were chosen and measurements at the SSM/I frequencies and polarizations were compiled. For example, some of these regions were the Amazon and Congo jungles, the Amazon basin, the Appalachian forest, the Central Plains of the U.S., the Mojave Desert, and the Sahara Desert. This method attempts to capture the emissivities of surface and vegetation type so brightness temperatures can be identified by its relation to the physical temperature.

During the course of the study, statistically significant differences became apparent for certain channel combinations which took advantage of polarization characteristics, i.e., if the scene predominantly featured tall vertically oriented trees, horizontal polarizations would dominate. The relations formed the ranges of values used in the final code, with added "flags" which act to guard against obvious mistyping (naming a block dry, arable soil when large bodies of water are present is an example of that kind of error) (McFarland, 1991).

This method contains two main weaknesses: 1) the innate variability of nearly all the SSM/I scenes; and 2) large "gray" areas where channel subtraction methods contribute values outside the useable range. The result of both weaknesses leads to a very

large number of undetermined land typing, which is a failure of the model to perform its function.

Table 2. CAL/VAL Land Surface Classifications

| Classification | Type |
|---|---|
| 0 | Undetermined Soil Type |
| 1 | Dry Arable Soil |
| 2 | Moist Arable Soil |
| 3 | Semi-Desert Soil |
| 4 | Desert Soil |
| 5 | Wet Soil |
| 6 | Dense Vegetation |
| 7 | Composite Water and Vegetation |
| 8 | Less Dense Vegetation |

The TMPSMI Model. TMPSMI attempts to do a similar land typing scheme using coefficients from a general linear model to approximate the actual brightness scene. This differs from CAL/VAL in that it takes coefficients and applies them to all seven brightness channels. The code was also written with planned progression of up to 25 different surface types, however at the time of this study only eight were operational due to a limitation of RTNEPH (Kopp, 1994). These are shown in Table 3, with the nominal classifications in the first column, the numbers assigned in the TMPSMI code in the second column, and the land type description in column three. The second column illustrates which types are "turned off" (that is, assigned zero) and which types are

grouped to form the set of eight limited by RTNEPH. This allows for a more balanced comparison between the models.

The weakness of TMPSMI is the same as for every general linear model, its ability to model the equation with the least amount of error. However, there exists the hope of continuously improving the tuning of its performance; something CAL/VAL cannot do.

Table 3. TMPSMI Land Surface Classifications

| Classification | Coding (if used) | Type |
|---|---|---|
| 01-06 | 0 | 6 Types of Air Over Ocean Surface |
| 07 | 3 | Snow-Covered Land |
| 08 | 3 | Deep Snow-Covered Land |
| 09 | 0 | Snow and Forest |
| 10 | 4 | Glacial Snow |
| 11 | 0 | Wet, Melting Snow |
| 12 | 0 | Sea Ice, First Year |
| 13 | 0 | Sea Ice, Multi-Year |
| 14 | 2 | Soil, Arable |
| 15 | 0 | Soil, Desert |
| 16 | 7 | Flooded Land |
| 17 | 0 | Soil, Frozen |
| 18 | 1 | Vegetated Land, Cold |
| 19 | 1 | Densely Vegetated Land |
| 20 | 6 | Sparsely Vegetated Land |
| 21 | 1 | Moderately Vegetated Land |
| 22 | 0 | Rain Over Land |
| 23 | 8 | Desert, Wet Surface |
| 24 | 5 | Desert, Morning |
| 25 | 0 | Desert, Afternoon |
| 26-30 | 0 | 5 Spares |
| 31 | 0 | Coast |

Model Calculations and Output

Copies of the FORTRAN 77 versions of both TMPSMI and CAL/VAL were obtained from AFGWC (see Appendices A and B). Both algorithms were run and the resulting gridded temperature fields were output to a file which also included the nearest station observation. A schematic which portrays this process is included as Figure 5.

The CAL/VAL routine is an uncomplicated code and ran smoothly in the research environment, using the basic land surface typing scheme described previously (McFarland, 1991), but produced a surprisingly high number of "undetermined land type" outputs. The effect for an undetermined land type is to not execute a shelter temperature computation, just enter "0.0" to the field at that grid point. These outputs were expected for any grid point over water, but too many land grids did not hold a shelter temperature at completion of the run. Obviously, this reduced the number of comparisons for which CAL/VAL could participate.

Comparing the data required synchronizing the data in time and space. Several FORTRAN codes were created to filter and manipulate the channels without losing the grid point's position and time stamp. Based on position, the observing station network dictated a grid point's usefulness, i.e., if the grid point fell within a certain proximity of the station, it was included in the statistics. The definition of a good proximity for this study was within one grid space in any direction from the observing station (Figure 6.).

Data from August and October 1996 were used. The Air Force Combat Climatology Center (AFCCC) provided the first dataset of hourly surface observations from 75 military airfields (Figure 7) throughout the month of August 1996. The

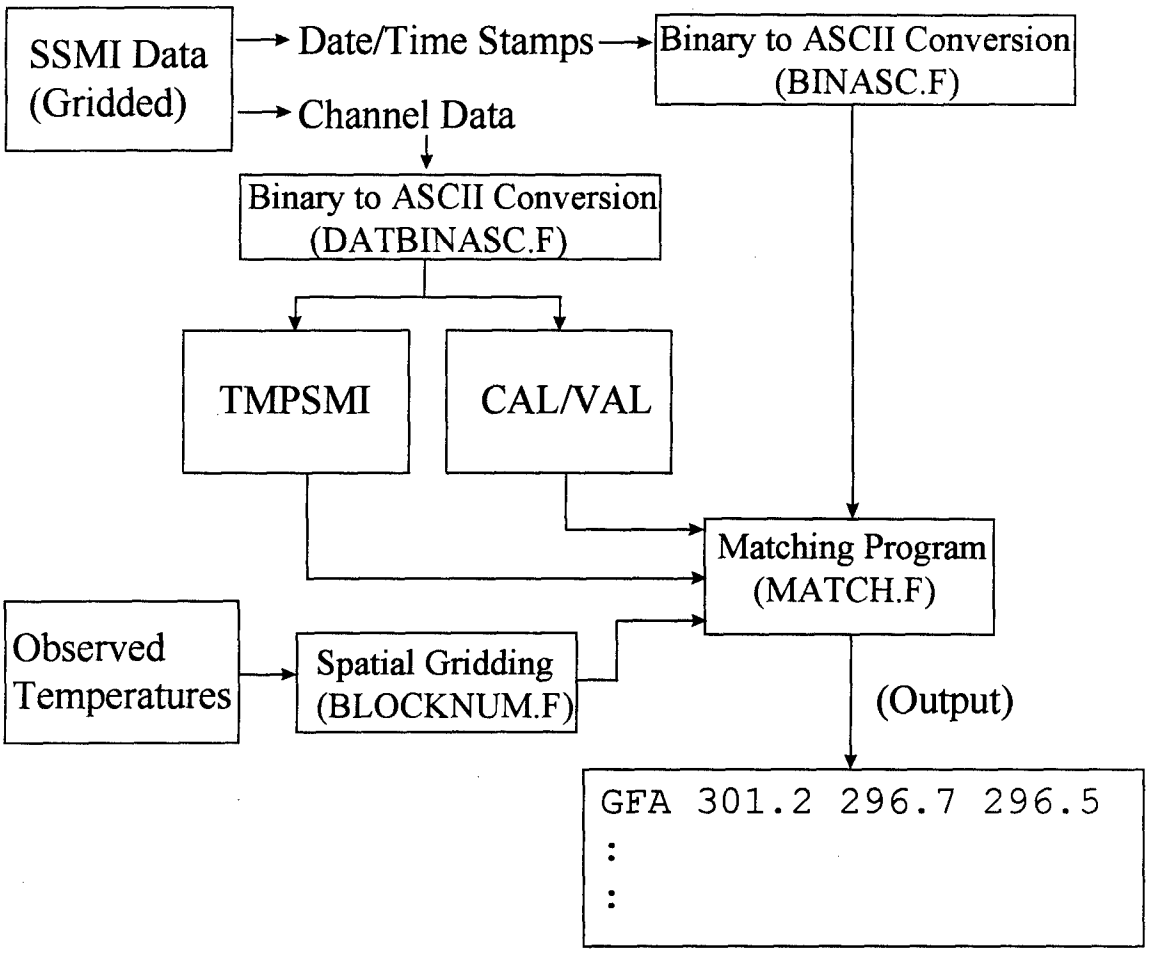


Figure 5. Schematic of Methodology Process. The output format is as follows: ICAO of observing station, the observed air temperature, the TMPSMI output, and then the CAL/VAL output.

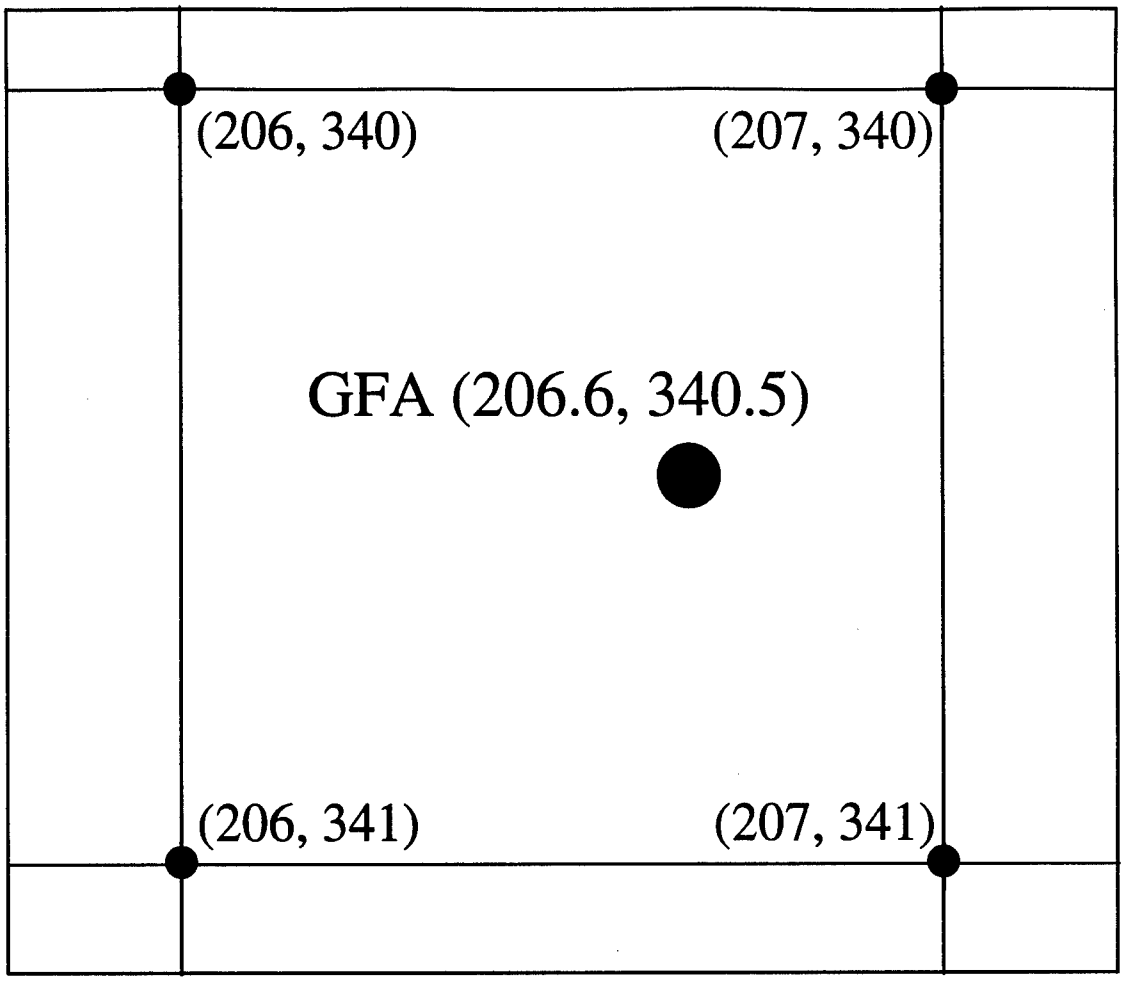


Figure 6. Super Neph gridding of points surrounding Malmstrom AFB.

observations corresponding to the October timeframe were manually collected from the GEMPAK dissemination system currently in use in the Weather Laboratory, Department of Engineering Physics, AFIT. From these datasets, temporal and spatial selections were accomplished. This consisted of creating a file of certain date/times and loading only those grids whose time stamp met that criteria. Allowance was made for a range of minutes to be rounded up or down to the cardinal hour. For example, if the grid was stamped with a time of 2350 UTC, it would be gathered in the file marked for 0000 UTC, not 2300 UTC.

For spatial sorting, using only those grid points which fell within a certain radius, or grid spacing in this case, of the observation station does not pose any physical problem. The surface temperature field is continuous and fairly slowly varying over horizontal distances, except in the region of an intense frontal system. In general, temperatures within a single grid space of the station can be considered a statistically dependent variable. Therefore, the grid points within one grid space of each station were identified, and if not previously time filtered out of consideration, were written to the files for comparative study. A copy of the output of a typical file is shown in Table 4, showing the comparative temperatures for gridpoints which surround the station.

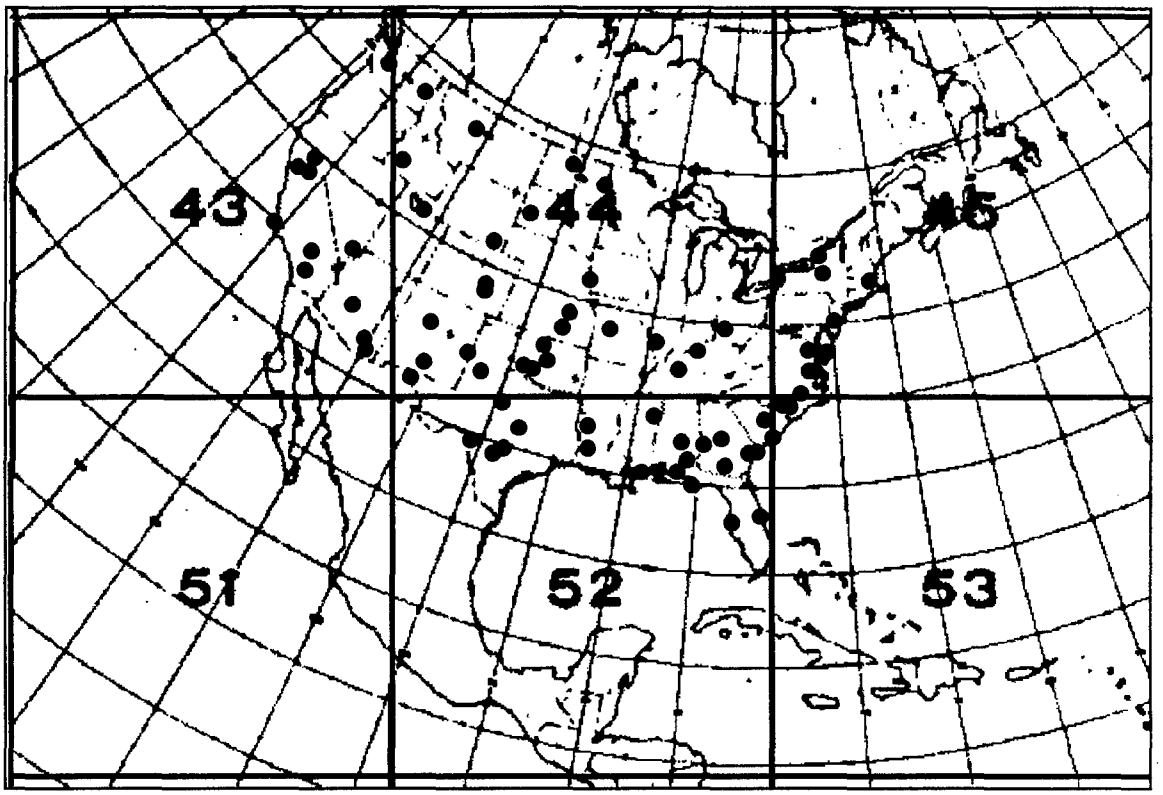


Figure 7: Nephboxes and CONUS Military Weather Observing Sites.

Table 4. Example of Data Output

| ICAO | Observed (K) | TMPSMI (K) | CAL/VAL (K) | Stn I | Stn J | I | J |
|---|---|---|---|---|---|---|---|
| GFA | 301.2 | 296.7 | 296.5 | 206.6 | 340.5 | 206 | 340 |
| GFA | 301.2 | 297.4 | 296.7 | 206.6 | 340.5 | 207 | 340 |
| GFA | 301.2 | 291.9 | 0.0 | 206.6 | 340.5 | 206 | 341 |
| GFA | 301.2 | 295.6 | 0.0 | 206.6 | 340.5 | 207 | 341 |
| MIB | 297.2 | 291.3 | 291.7 | 222.6 | 345.8 | 222 | 345 |
| MIB | 297.2 | 290.1 | 294.2 | 222.6 | 345.8 | 223 | 345 |
| MIB | 297.2 | 290.1 | 291.6 | 222.6 | 345.8 | 222 | 346 |
| MIB | 297.2 | 290.2 | 291.1 | 222.6 | 345.8 | 223 | 346 |

Gridpoints in Table 4 correspond to the "Super NEPH" grid, which is a 512 by 512 gridding instead of the parsing by neph box then gridding (as shown in Figure 4). Note that for grids (206, 341) and (207, 341) CAL/VAL was unable to produce a surface temperature. At the final analysis, both models produced roughly the same number of "good" gridpoints, usually failing at different geographical locations.

The TMPSMI algorithm required the most effort to adapt to local processing, as roughly two-thirds of the attendant subroutines prescribed input and output procedures within the AFGWC modeling system. To make it run on a Sun workstation instead the code had to be "fooled" into running as if at home. After disabling the many error flags used to monitor the automated run, TMPSMI produced its contribution to the study.

The resulting data file no longer carried the time stamp, as all the corresponding temperatures were matched up and the time dependence was eliminated. The ICAO identifiers were retained simply as a means to more easily locate the gridpoints in space. Once all the useable gridpoints were collected, the statistical analysis began.

# IV. Results and Data Analysis

## Introduction

Two periods of data were calculated for days in August 1996, representing the summer season, and October 1996, representing the fall season. Each set contains observed temperatures at military airfields and the CAL/VAL and TMPSMI calculated temperatures at surrounding gridpoints. This matching process is crucial to the examination since a single day download may consist of brightness measurements from several various hours due to the limited swath width of the SSM/I.

To ensure the ground observations are matched in time and space to the satellite measurement at every grid point proved to be no small task. Several FORTRAN programs filtered the data by following this sequence:

1. The satellite time stamp is read for each grid point. This consists of the Julian day, hour, and minute of the grid point (or element) scan.

2. The location of the stations is converted to the RTNEPH grid and the surrounding gridpoints are identified.

3. Each surrounding grid point is associated with an observed temperature from the station it surrounds, at the date/time the satellite associated with that grid point.

4. The models are run and the model temperatures at the identified grid points are gathered. An observed temperature is matched if at least one nonzero model temperature is available; if there is not a contribution, that gridpoint is eliminated.

The final output file then contains the station identifier (ICAO), the observed temperature, the TMPSMI calculated temperature, the CAL/VAL calculated temperature, and the (I,J) gridpoint of the calculated variables.

August Results

The run of the August data resulted in 1,382 matches. The initial plot of the raw temperatures is graphed in Figure 8. A closer examination shows that both models consistently calculated values below the expected surface temperature, with the TMPSMI showing the lower temperatures during the hottest month of the year.

Descriptive statistics indicate both samples differ from the observed (Table 5).

Table 5: August Statistics

| | **Mean (K)** | **Variance ($K^2$)** | **Std Deviation (K)** |
|---|---|---|---|
| **Observed** | 299.715 | 44.733 | 6.688 |
| **TMPSMI** | 293.400 | 62.116 | 7.881 |
| **CAL/VAL** | 293.910 | 25.290 | 5.029 |

Since the mean temperature differs by only 0.5 K, it leads one to ponder whether there is any statistical difference in the populations from which the samples were drawn. By definition the normal distribution family is defined by its mean and its variance, therefore the variability becomes the discriminator.

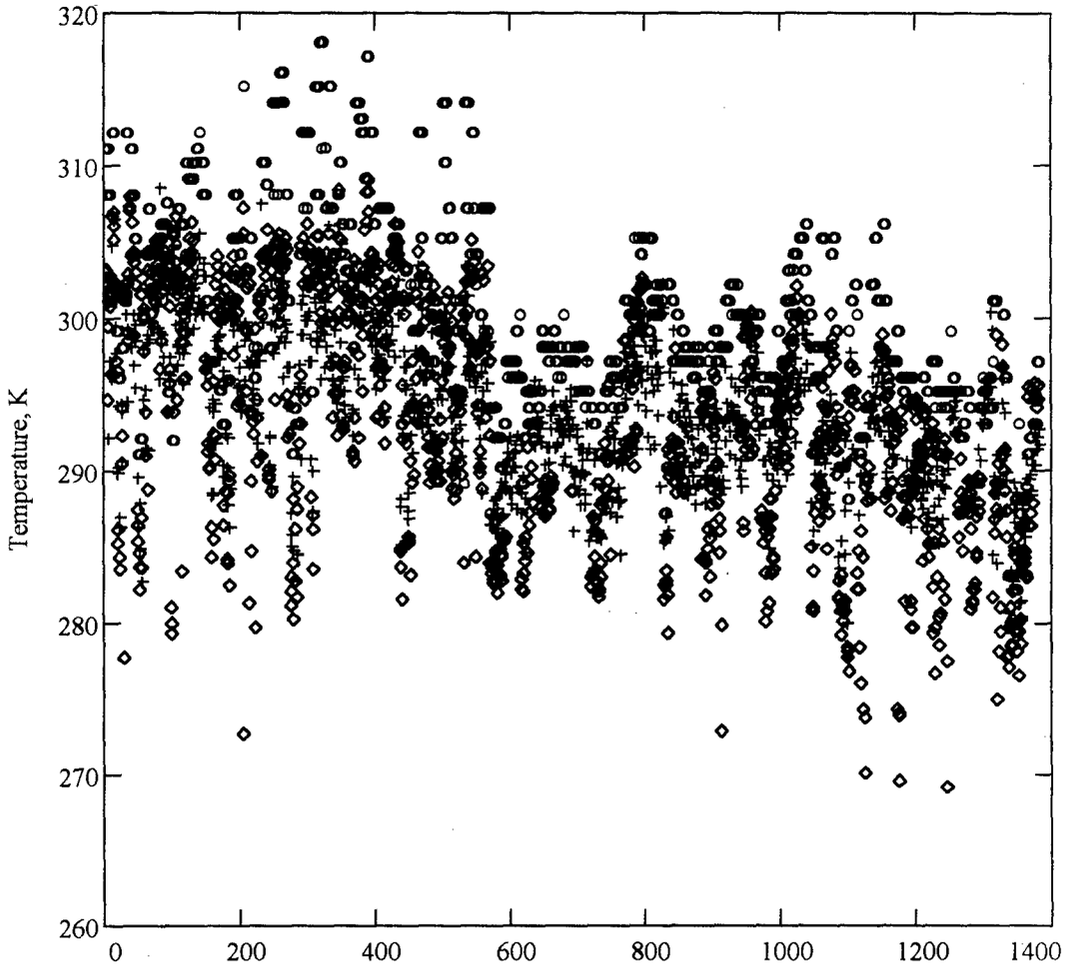


Figure 8. Comparison plot of August temperatures. The circles are the observed temperatures, the diamonds indicate TMPSMI output values, and the crosses indicate the CAL/VAL output values.

The procedure chosen analyzes the three samples by creating a two sample comparison. This is accomplished by subtracting both model outputs from the respective observed temperatures.

$$TMPSMI - Observed = TMPDiff \tag{28}$$

$$CAL/VAL - Observed = CALVDiff \tag{29}$$

This compares both model outputs relative to the truth data. The same number of samples from both model output vectors was subtracted from its corresponding observation temperatures to produce two difference vectors, TMPDiff and CALVDiff. By defining the resultant vectors this way we can move to a statistical T test to decide if there is a statistical difference between the original populations without losing the reference frame provided by the ground "truth"; in this case, the surface observations.

This test is a paired T test, due to the pairing of the data with the observed and the resulting independency between samples. It is used because of its robustness, even when the variances are not equal, as long as they are not too disparate. It also works for distributions which approximate the normal, which these do under the Central Limit Theorem (Devore, 1995).

The results of the test, conducted at a 95% confidence interval and with 1,076 degrees of freedom, showed there is indeed a significant statistical difference between the August populations. This result indicates there is then a difference in the output of the two models worth pursuing. If the T test had shown no difference, the comparison study would be moot. The theoretical difference distributions are displayed in Figure 9, along with a total difference distribution created by subtracting TMPDiff and CALVDiff,

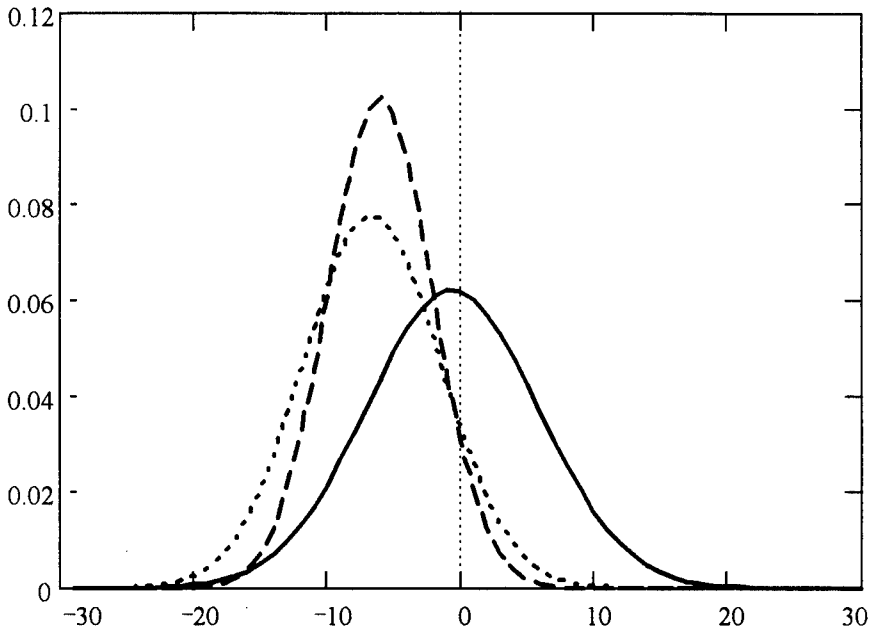


Figure 9. Probability Density Functions of August theoretical distributions. The solid line is the distribution of the differences (the mean is centered on -0.618), the dotted line is the distribution of TMPSMI departures from observed, and the dashed line is the distribution of CAL/VAL departures from observed.

$$TMPDiff - CALVDiff = Diff \quad (30)$$

which graphically illustrates the disparity between model performances. If the T test had failed, the mean of Diff would be closer to zero. Notice that the CAL/VAL difference distribution is centered closer to the observed and has a tighter variance, but as noted in Table 5, the mean is still quite less than the observed mean.

October Results

The measurements taken during the month of October 1996 portray a significantly different depiction (Table 6), possibly indicating seasonal sensitivities in each model's capability. The October dataset had many fewer comparison points, as only eight calendar days of data were available.

Table 6: October Statistics

| | **Mean (K)** | **Variance ($K^2$)** | **Std Deviation (K)** |
|---|---|---|---|
| **Observed** | 285.796 | 83.442 | 9.135 |
| **TMPSMI** | 278.239 | 53.762 | 7.332 |
| **CAL/VAL** | 283.905 | 59.227 | 7.696 |

The analysis for those days produced only 627 data points, which are displayed in Figure 10. For this case, the cold bias mentioned for the August dataset is not as apparent. It is possible that for a transition season, when even the observed temperatures

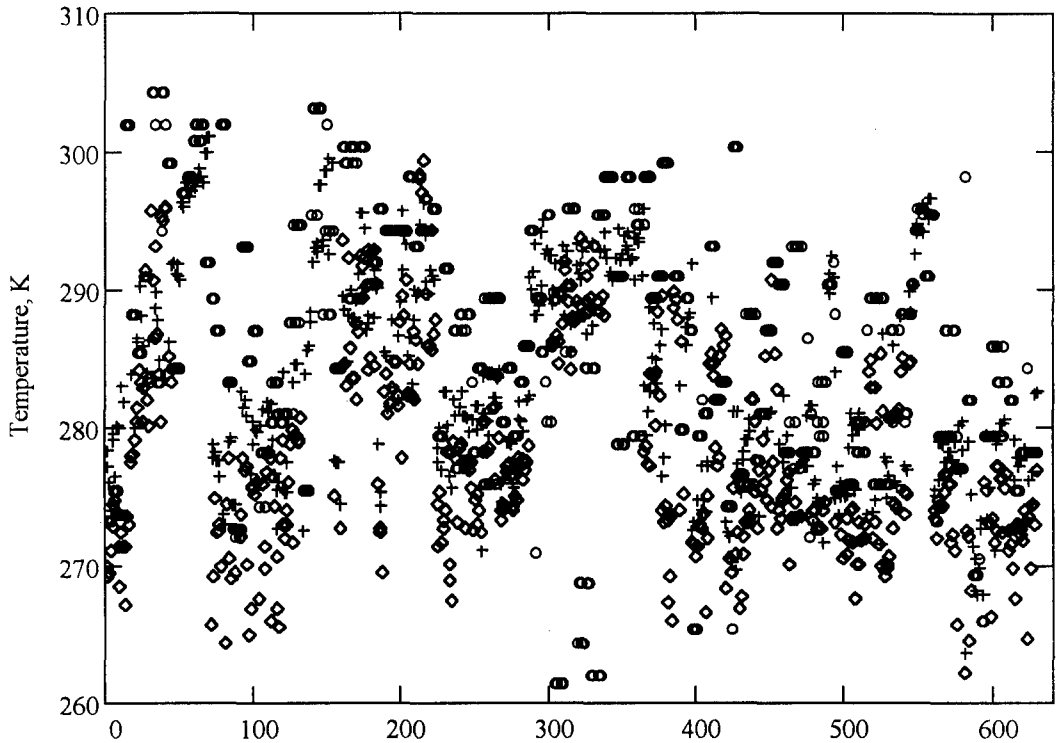


Figure 10. Comparison plot of October temperatures. The circles are the observed temperatures, the diamonds indicate TMPSMI output values, and the crosses indicate the CAL/VAL output values. Any data gaps indicate a zero value for the model output.

exhibit a large variance (see Table 6), that bias does not hold. Also, CAL/VAL produces a mean very close to the observed mean. However, the trends noted in the August data are not present in the October data. For example, the correlation coefficients are significantly worse in the October data, the mean absolute error is higher, and some of the CAL/VAL temperatures exceed the observed, something which did not occur even once in the August data. The additional comparative statistics are shown in Table 7. The variables shown are the Pearson Correlation Coefficient relative to the observed temperatures, calculated by

$$\rho(x,y) = \frac{\mathrm{cov}(x,y)}{\sigma_x \cdot \sigma_y} \quad , \tag{31}$$

where cov(x,y) is the covariance, a statistical measure of how strongly two variables relate to each other, and the $\sigma_x$ and $\sigma_y$ are the standard deviations of the variables in question. The Mean Absolute Error (MAE) is calculated by

$$MAE = \frac{1}{n}\sum_{k=1}^{n}\left|temp_k - observed_k\right| \ , \tag{32}$$

where n is the number of samples, k is the summation index, $temp_k$ is the $k^{th}$ model output temperature, and $observed_k$ is the $k^{th}$ observed temperature (Devore, 1995).

The data indicates a very good correlation between the model outputs in the August measurements, and a noticeable drop off in correlation for the October measurements. The MAE does compare between the data sets, with CAL/VAL performing better than TMPSMI in both months.

Table 7: Comparative Statistics

| | Correlation | Coefficients | Mean Abs | Error (K) |
|---|---|---|---|---|
| | TMPSMI | CAL/VAL | TMPSMI | CAL/VAL |
| August Data | 0.770 | 0.812 | 7.120 | 6.255 |
| October Data | 0.369 | 0.518 | 9.179 | 6.110 |

Following the same analysis procedure, the T statistic again differentiates between the model output populations. The graphical portrayal of the difference populations is shown in Figure 11. Obviously, the difference between the two samples supports the T test result because the mean is not centered on zero. It is also apparent that CAL/VAL produced a much more accurate temperature field than did TMPSMI.

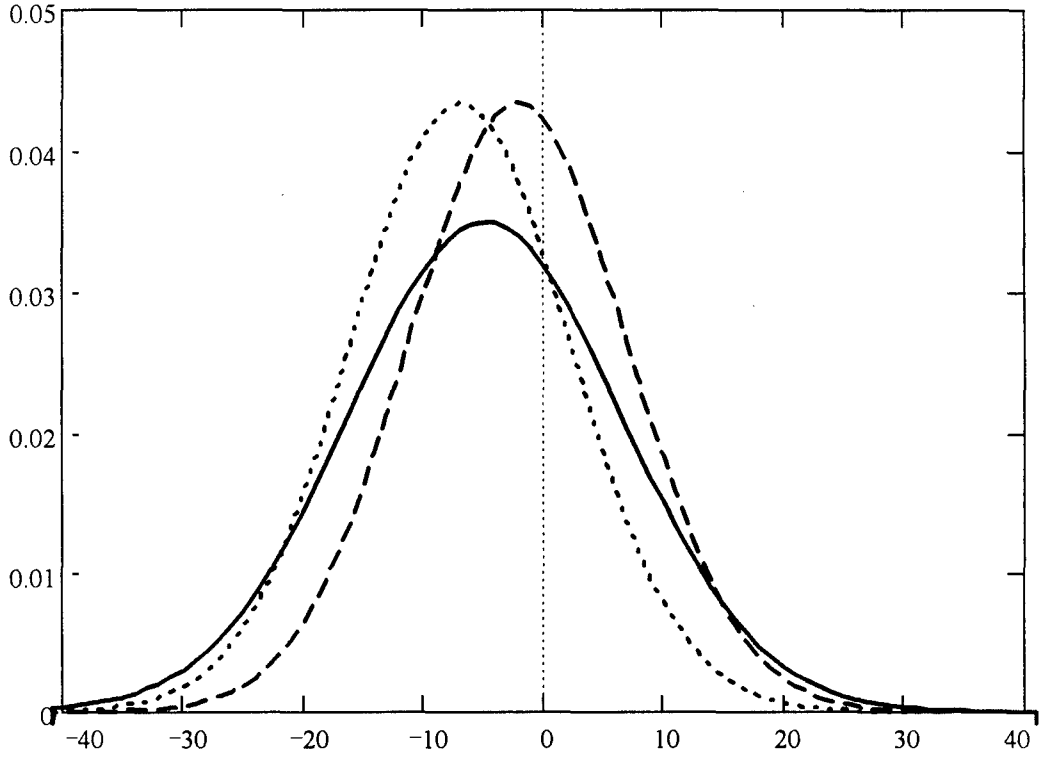


Figure 11. Probability density functions of October theoretical distributions. The solid line is the distribution of the differences (the mean is centered on -4.798), the dotted line is the distribution of TMPSMI departures from observed, and the dashed line is the distribution of CAL/VAL departures from observed.

## V. Conclusions and Recommendations

### Conclusions

Based on the data analyzed, the CAL/VAL model proved to be consistently more accurate, less so for the warmer season. The tighter variance pointed toward a more reliable output, however this may not necessarily mean CAL/VAL is the better model.

If the larger variance of TMPSMI is due to a tendency to attempt a calculation for more land points, then this model may be more useful in providing global coverage. At this time the CAL/VAL may outperform TMPSMI, however TMPSMI incorporates all seven channels on every calculation in the form of a general linear model, while CAL/VAL applies empirical relationships. The idea that eventually the equations optimize, associated with the theory of the general linear model, suggests TMPSMI is a dynamic model, whereas CAL/VAL is a more static case. The question which arises is has TMPSMI optimized or has it reached peak performance?

For the August dataset, figuring a potential of four surrounding points for each of 75 stations, we expect 300 points per day to be the upper bound. If either model operated at perfect efficiency, for the thirteen available August days the resulting number of points would be 3,900. Of course, some of these points correspond to coastal stations and would be over water. Also, data gaps caused by polar orbiter coverage further decreases the total. As such, the August dataset produced 1,382 grid points at which at least one of the models provided a temperature calculation. The breakout between models was 1,108

by TMPSMI versus 1,077 by CAL/VAL. A difference of 31 may not seem large but when extended to a global scale may indicate TMPSMI is more robust than CAL/VAL.

For October, having only eight days of data means the upper bound on points would be 2,400. The total was only 627, with TMPSMI providing 469 and CAL/VAL providing 477. Since this run contained more CAL/VAL points, the previous assertion regarding TMPSMI's robustness may not be valid. However, based on Table 5 (previous chapter), the comparative statistics cast doubt on the validity of the October dataset and lend weight to conclusions drawn from the August data.

At the termination of the research, it was revealed the observed surface temperatures used in the October dataset had not been properly quality controlled in the GEMPAK process. This supports the previously stated doubts regarding those results and indicates a re-analysis is required.

Recommendations

Before any final assessment can be forwarded, a true multi-seasonal examination is required. This study used data available under the time constraints given, allowing an inspection of performance over the CONUS representing a single hot season and a transition season. Further study may be necessary to rule out any annual biases inherent to each model. An effort is currently underway to inspect a single cold season.

Also, to state one model is preferred over another based on a CONUS test and then extrapolating that thesis to a global scale could introduce erroneous results. The surface types not inspected were the jungle, rainforest and polar ice settings. Of course,

obtaining truth data for these land types poses a formidable task. TMPSMI claims the ability to process more land types and should be further tested on this point.

The scope of this study was to inspect the deterministic output of each model in a comparison study. The land typing schemes are so similar the results are not greatly different. A more detailed inspection of each model's regression equations could now be undertaken for the purpose of improving the output to correct for the apparent cold bias both models display and to determine if the TMPSMI regression coefficients are optimized.

More detail should also be given to the effects of various sources of error which were not sufficiently treated here. The sources which may be affecting the individual temperature derivations most are:

1) Instrument noise error; specifically, how it is treated in the regression equations?
2) Contamination error; what effect does the sun's contribution make? This points toward a type of diurnal study.
3) Spatial errors; this paper's assumption was that one gridspace did not impact the comparison greatly.

Additionally, while the other seasons are being examined, a dependable collection of the October surface temperatures should be collected and the model performances re-tested.

Therefore, this study recommends to the sponsoring agency the goal of tightening the TMPSMI variance and correction of the cold bias, and the extension of surface types

tested to include those excluded by a CONUS sample, with the results of this study as a benchmark for progress measurement.

# Appendix A: TMPSMI Code

```
      PROGRAM TMPSMI
C  This version of TMPSMI is modified to run on Sun workstations  within
the AFIT network domain.  It C  has been set to process only  Northern
Hemisphere, CONUS nephboxes.

      INTEGER HEM
      INTEGER I,IE
      INTEGER IRSSCO(9,8)
      INTEGER LTT(32768)
      INTEGER STYDIS(8,3)
      INTEGER TUNDAT(128)
      INTEGER TYPEID(0:31)
      CHARACTER FILE*13,DATE*7
      LOGICAL SMIERR
      LOGICAL TUNERR

      PARAMETER(IE=32678)
      EQUIVALENCE (TUNDAT(1),IRSSCO)
      EQUIVALENCE (TUNDAT(73),STYDIS)
      EQUIVALENCE (TUNDAT(97),TYPEID)

      SMIERR = .FALSE.
      TUNERR = .FALSE.
C      ******************************************************************
C      READ CONDITION WORD TO DETERMINE HEMISPHERE
C      ******************************************************************
C      SET THE HEMISPHERE TO NORTHERN FOR THESIS APPLICATION
      HEM = 1

      PRINT *, 'Enter date in DDMMMYY format, e.g., 01AUG96'
      READ(*, FMT=30) DATE
30     FORMAT(A7)

      FILE='./data/lttbas'

      OPEN(UNIT=10, FILE=FILE, STATUS='UNKNOWN', FORM=
     &   'FORMATTED', ACCESS='SEQUENTIAL')

C     ****************************************************************
C     LOAD TUNING ARRAYS
C     ****************************************************************
      CALL SMITUN(TUNDAT,TUNERR,HEM,TYPEID,STYDIS,IRSSCO)
C     ****************************************************************
C     READ SFCTMP TRANSLATION TABLE IF SUCCESSFUL READ OF TUNING ARRAYS
C     ****************************************************************
      IF (.NOT. TUNERR) THEN
           PRINT*,'Reading LTT'
           READ(10,50) (LTT(I),I=1,IE)
  50    FORMAT(I2)


C       ***************************************************************
C       PROCESS BOX OF SSM/I
C       ***************************************************************
           PRINT*, 'CALLING SMIBOX FROM MAIN'
```

```
          CALL SMIBOX(HEM,STYDIS,IRSSCO,TYPEID,LTT,DATE)
      ENDIF
C      ****************************************************************
C      IF ERROR LOADING TUNING ARRAYS OR READING SFCTMP TRANSLATION TABLE
C     THEN FILL ALL SSM/I TEMPS FOR  ENTIRE HEMISPHERE TO ZERO AND WRITE
C     TO FILE
C      ****************************************************************

      IF ((TUNERR) .OR. (SMIERR))  THEN

          PRINT *,'TMPSMI DID NOT RUN PROPERLY - SFCTMP*SSMI IS EMPTY'

      ELSE

      PRINT *
      PRINT *, ' **** NORMAL EXECUTION OF TMPSMI FINISHED **** '

      ENDIF

      STOP
      END

      SUBROUTINE BOXCLC(HEM,NEFBOX,GEODAT,SMIDAT,STYDIS,LTT,
      &              IRSSCO,TYPEID,SMIBAK,SMIHR)

      INTEGER GEODAT(64,64)
      INTEGER HEM
      INTEGER I8
      INTEGER IRSSCO(9,8)
      INTEGER J8
      INTEGER NEFBOX
      INTEGER LTT(32678)
      INTEGER SMIDAT(64,64,8)
      INTEGER SMIHR(64,64)
      INTEGER SMITYP
      INTEGER STYDIS(8,3)
      INTEGER TYPEID(0:31)
      REAL SMIBAK(64,64)

C      ****************************************************************
C      INITIALIZE SMIHR ARRAY
C      ****************************************************************
      CALL COPY(1,1,SMIHR,1,4096)

C      ****************************************************************
C      LOOP THROUGH ALL POINTS IN THE BOX
C      ****************************************************************
      DO 100 J8 = 1 , 64
          DO 50 I8 = 1 , 64
               SMITYP = 0
C      ************************************************************
C      IS GEOGRAPHY AT PT I,J ON LAND (2) ICE (3), OR COASTLINE (4)
C      ************************************************************

               IF ((GEODAT(I8,J8) .GE. 2) .AND.
      &                    (GEODAT(I8,J8) .LE. 4)) THEN
                    CALL CLCGEO(I8,J8,STYDIS,SMIDAT,LTT,HEM,
      &                  NEFBOX,SMITYP)
```

```
C         ***********************************************************
C         IF A GOOD READ ON SFC TYPE THEN CALC BRIGHTNESS TEMP
C         ***********************************************************
                    IF (SMITYP .NE. -1) THEN

                    CALL CLCTMP(I8,J8,IRSSCO,TYPEID,SMITYP,SMIDAT,
     &                   SMIBAK)

                    ELSE

C           *********************************************************
C           ELSE FILL SSM/I TEMP AS ZERO
C           *********************************************************
                         SMIBAK(I8,J8) = 0.0

                    ENDIF
                ELSE

C         ***********************************************************
C         IF GEOGRAPHY POINT IS ON WATER THEN SET SSM/I TEMP TO ZERO
C         ***********************************************************
                    SMIBAK(I8,J8) = 0.0

                ENDIF
 50  CONTINUE
 100  CONTINUE

     RETURN
     END


     SUBROUTINE BOXDAT(HEM,I,J,NEFBOX,SMIDAT,DATE,DATERR)

     CHARACTER FILE19H*60, FILE19V*60,FILE22V*60,FILE37H*60
     CHARACTER FILE37V*60, FILE85H*60, FILE85V*60
     CHARACTER*7 DATE
     REAL H19(64,64,64), H37(64,64,64), H85(64,64,64), V19(64,64,64)
     REAL V22(64,64,64), V37(64,64,64), V85(64,64,64)
     INTEGER HEM
     INTEGER NEFBOX
     INTEGER SMIDAT(64,64,8)
     INTEGER I, J, IE, JE
     LOGICAL DATERR
     PARAMETER(IE=64, JE=64)

C     ***************************************************************
C     INITIALIZE DATERR
C     ***************************************************************
     DATERR = .FALSE.

C    *******************************************************
C    CALL SMIFLD TO RETRIEVE SSM/I DATA
C    *******************************************************
     FILE19H = './data/'//DATE//'19H.DAT'
     FILE19V = './data/'//DATE//'19V.DAT'
     FILE22V = './data/'//DATE//'22V.DAT'
     FILE37H = './data/'//DATE//'37H.DAT'
     FILE37V = './data/'//DATE//'37V.DAT'
     FILE85H = './data/'//DATE//'85H.DAT'
```

```
      FILE85V = './data/'//DATE//'85V.DAT'
      PRINT*,'FILE19H=',FILE19H

C   Open the respective datafiles for the channel Tbs

      OPEN(UNIT=50, FILE=FILE19V, ACCESS='SEQUENTIAL')
      OPEN(UNIT=60, FILE=FILE19H, ACCESS='SEQUENTIAL')
      OPEN(UNIT=70, FILE=FILE22V, ACCESS='SEQUENTIAL')
      OPEN(UNIT=80, FILE=FILE37V, ACCESS='SEQUENTIAL')
      OPEN(UNIT=90, FILE=FILE37H, ACCESS='SEQUENTIAL')
      OPEN(UNIT=100, FILE=FILE85V, ACCESS='SEQUENTIAL')
      OPEN(UNIT=110, FILE=FILE85H, ACCESS='SEQUENTIAL')
      PRINT*,'LOADING DATA'

      DO J=1, JE
          DO I=1, IE
              READ(50, FMT=120) V19(I,J,NEFBOX),I,J,NEFBOX
              SMIDAT(I,J,2)=V19(I,J,NEFBOX)
              READ(60, FMT=120) H19(I,J,NEFBOX),I,J,NEFBOX
              SMIDAT(I,J,3)=H19(I,J,NEFBOX)
              READ(70, FMT=120) V22(I,J,NEFBOX),I,J,NEFBOX
              SMIDAT(I,J,4)=V22(I,J,NEFBOX)
              READ(80, FMT=120) V37(I,J,NEFBOX),I,J,NEFBOX
              SMIDAT(I,J,5)=V37(I,J,NEFBOX)
              READ(90, FMT=120) H37(I,J,NEFBOX),I,J,NEFBOX
              SMIDAT(I,J,6)=H37(I,J,NEFBOX)
              READ(100, FMT=120) V85(I,J,NEFBOX),I,J,NEFBOX
              SMIDAT(I,J,7)=V85(I,J,NEFBOX)
              READ(110, FMT=120) H85(I,J,NEFBOX),I,J,NEFBOX
              SMIDAT(I,J,8)=H85(I,J,NEFBOX)
          ENDDO
      ENDDO

120    FORMAT(F5.1,1X,I2,1X,I2,1X,I2)
130    FORMAT(I3)

200    CONTINUE


      RETURN
      END

      SUBROUTINE BOXGEO(HEM,NEFBOX,GEODAT,GEOERR)

      INTEGER GEODAT(64,64)
      INTEGER HEM
      INTEGER I
      INTEGER J
      INTEGER NEFBOX

      LOGICAL GEOERR

C      ******************************************************************
C      INITIALIZE GEOERR
C      ******************************************************************
      GEOERR = .FALSE.

C      ******************************************************************
C      COPY GEOGRAPHY DATA TO ARRAY -- Set to 2 (Over Land Point)
```

```
C     ****************************************************************
      DO 100 J = 1 , 64
          DO 50 I = 1 , 64
              GEODAT(I,J) = 2

 50       CONTINUE

 100 CONTINUE

      RETURN
      END

      SUBROUTINE CLCGEO(I8,J8,STYDIS,SMIDAT,LTT,HEM, NEFBOX,SMITYP)

      INTEGER DISC(3)
      INTEGER HEM
      INTEGER I
      INTEGER I8
      INTEGER J
      INTEGER J8
      INTEGER LOCTIM
      INTEGER MAXDSC(3)
      INTEGER NEFBOX
      INTEGER OFFSET
      INTEGER LTT(32768)
      INTEGER SMIDAT(64,64,8)
      INTEGER SMITYP
      INTEGER STYDIS(8,3)
      REAL SUM(3)
      DATA MAXDSC/63,31,15/

      SMITYP = 0

C     ****************************************************************
C     COMPUTE DISCRIMINANTS BY A WEIGHTED SUM OF SSM/I BRIGHTNESS
C     TEMPS PLUS A CONSTANT.  DISCRIMINANTS ARE SCALED TO THE
C     INTERVALS 1-[0,63] 2-[0,31] 3-[0,15]
C     ****************************************************************
      DO 150 J = 1 , 3

C     ************************************************************
C     SET SUM EQUAL TO SCALED CONSTANT
C     ************************************************************
          SUM(J) = FLOAT(STYDIS(1,J)) / 1000.0

C     ************************************************************
C     DO BRIGHTNESS TEMPS FOR EACH CHANNEL
C     ************************************************************
          DO 50 I = 2 , 8

C       ************************************************************
C       ADD IN COEFFICIENT * BRIGHTNESS TEMPS
C       ************************************************************
                SUM(J)= SUM(J)+(FLOAT(STYDIS(I,J)*SMIDAT(I8,J8,I))/1.0E6)

C       ************************************************************
C     IF SMIDAT ARRAY HAS BAD DATA AND CHANNEL IS VALID, SET
C       SFCTYP TO ERROR  (-2 equals missing data)
C       ************************************************************
```

```
          IF ((SMIDAT(I8,J8,I) .EQ. -2)  .AND.  (STYDIS(I,J) .NE. 0))
          THEN
                    SMITYP = -1
                    GOTO 100
          ENDIF

 50       CONTINUE
 100 CONTINUE

     IF (SMITYP .EQ. 0) THEN

C      ************************************************************
C      SET VALUE OF DISCRIMINANT
C      ************************************************************
          DISC(J) = NINT(SUM(J))

C      ************************************************************
C      CHECK THAT THE DISCRIMINANT IS WITHIN BOUNDS AND WITHIN
C      INTERVAL SET ABOVE
C      ************************************************************
          IF ((DISC(J) .LT. 0).OR. (DISC(J) .GT. MAXDSC(J))) SMITYP = -1

     ENDIF

 150  CONTINUE

C     ****************************************************************
C     IF CALCULATIONS ABOVE ARE GOOD THEN CALCULATE INDEX INTO
C     PACKED TRANSLATION TABLE
C     ****************************************************************
     IF (SMITYP .EQ. 0) THEN
          OFFSET = DISC(1) + (DISC(2) * 64) + (DISC(3) * 2048)

C    ************************************************************
C    RETRIEVE SURFACE TYPE
C    ************************************************************
          SMITYP = LTT(OFFSET)

C    ************************************************************
C    IF SFC TYPE IS DESERT (15) CALL GEOTIM TO CALC SSM/I LOCAL TIME
C    ************************************************************
          IF (SMITYP .EQ. 15) THEN
               CALL GEOTIM(SMIDAT(I8,J8,8),HEM,I8,J8,LOCTIM,NEFBOX)

C      ************************************************************
C      IF LOCTIM < 0 SET SMITYP TO UNKNOWN
C      ************************************************************
               IF (LOCTIM .LT. 0) THEN
                    SMITYP = -1

C        ***********************************************************
C        OTHERWISE SET SFC TYPE TO AM DESERT OR PM DESERT
C        ***********************************************************
               ELSE IF (LOCTIM .LT. 1200) THEN
                    SMITYP = 24
               ELSE
                    SMITYP = 25
```

```
            ENDIF
        ENDIF
    ENDIF

    RETURN
    END

    SUBROUTINE CLCTIM(SMIDAT,HEM,SMIHR,HRMIN)

    INTEGER DAY
    INTEGER HEM
    INTEGER HRMIN
    INTEGER IDY
    INTEGER IMO
    INTEGER ISTAT
    INTEGER SMIDAT
    INTEGER SMIHR
    INTEGER YEAR

    SMIHR = 0

C     ****************************************************************
C     UNPACK SSM/I TIME STAMP INTO DAY, YEAR, AND HOUR/MINUTE
C     ****************************************************************
    DAY = SMIDAT / 1000000
    YEAR = MOD(SMIDAT,1000000) / 10000
    HRMIN = MOD(SMIDAT,10000)

C     ****************************************************************
C     RETRIEVE MONTH AND DAY OF MONTH
C     ****************************************************************
    IF (SMIDAT .NE. 0) THEN
        PRINT*,'Enter day of download'
        READ(*,*) IDY
        PRINT*,'Enter the month'
        READ(*,*) IMO
        ISTAT = 0

        IF (ISTAT .NE. 0) THEN
            PRINT *,'ERROR IN SUB CLCTIM IN TMPSMI'
            PRINT *,'COULD NOT READ JDDATE, ISTAT = ',ISTAT
            PRINT *,'NO SSM/I TEMPS AVAILABLE FOR HEM = ',HEM
            HRMIN = 1
            SMIHR = 1
        ELSE
        ENDIF

    ELSE
        SMIHR = 1
        HRMIN = 1
    ENDIF

    RETURN
    END

    SUBROUTINE CLCTMP(I8,J8,IRSSCO,TYPEID,SMITYP,SMIDAT,SMIBAK)
```

```
      INTEGER I
      INTEGER I8
      INTEGER IRSSCO(9,8)
      INTEGER J8
      INTEGER SMIDAT(64,64,8)
      INTEGER SMITYP
      INTEGER TYPEID(0:31)
      LOGICAL ERRFLG
      REAL SMIBAK(64,64)
      REAL SUM

      ERRFLG = .FALSE.

C      ******************************************************************
C      IF VALID SURFACE TYPE THEN PROCESS ELSE SET SUM TO ZERO
C      ******************************************************************
      IF (TYPEID(SMITYP) .GT. 0) THEN

C     ***************************************************************
C     SET SUM = SCALED CONSTANT
C     ***************************************************************
      SUM = FLOAT(IRSSCO(1,TYPEID(SMITYP))) / 1000.0

C     ***************************************************************
C     CALCULATE BRIGHTNESS TEMP FOR EACH CHANNEL
C     ***************************************************************
      DO 100 I = 2 , 8

C       ************************************************************
C       ADD COEFFICIENT * BRIGHTNESS TEMP TO SUM
C       ************************************************************
          SUM = SUM + (FLOAT(IRSSCO(I,TYPEID(SMITYP)) *SMIDAT(I8,J8,I))
     &    / 1.0E4)

C       ************************************************************
C       IF SMIDAT ARRAY HAS BAD DATA AND CHANNEL IS VALID, SET ERR FLAG
C       ************************************************************
          IF ((SMIDAT(I8,J8,I) .EQ. -2) .AND. (IRSSCO(I,TYPEID(SMITYP))
     &    .NE. 0)) ERRFLG = .TRUE.

 100 CONTINUE
          ELSE
               SUM = 0
          ENDIF

C      ******************************************************************
C      IF ERROR FLAG IS SET OR TEMPS ARE OUT OF BOUNDS, SET TEMP TO ZERO
C      ELSE WRITE SUM (TEMP) TO SMIBAK
C      ******************************************************************
      IF ((ERRFLG) .OR. (SUM .LT. 200.0) .OR. (SUM .GT. 325.0)) THEN
          SMIBAK(I8,J8) = 0.0
      ELSE
          SMIBAK(I8,J8) = SUM
      ENDIF

      RETURN
      END
```

```
      SUBROUTINE GEOTIM (DTGRP,HEM,I8,J8,LOCTIM,NEFBOX)

      INTEGER DTGRP
      INTEGER GMTHRS
      INTEGER I8
      INTEGER ICON
      INTEGER IERR
      INTEGER IGLF
      INTEGER IGMF
      INTEGER HEM
      INTEGER J8
      INTEGER LOCHRS
      INTEGER LOCTIM
      INTEGER NBOX
      INTEGER NEFBOX
      REAL RI
      REAL RJ
      REAL RLAT
      REAL RLON

      DATA IGMF /4/
      DATA IGLF /0/
      DATA ICON /2/
      DATA NBOX /0/

C      ******************************************************************
C      CONVERT I,J COORDINATES TO LATITUDE AND LONGITUDE.  LONGITUDE IS
C      RETURNED AS A VALUE FROM 0 TO 360 INCREASING WESTWARDS.
C      ******************************************************************
      RI = (FLOAT(MOD((NEFBOX - 1),8) * 64) + I8) + 0.5
      RJ = (FLOAT(((NEFBOX - 1) / 8) * 64) + J8) + 0.5
      PRINT*,'CALLING GRDTLL'
      CALL GRDTLL (NBOX,RI,RJ,RLAT,RLON,HEM,IGMF,IGLF,ICON,IERR)

C      ******************************************************************
C      IF THERE IS AN ERROR CALLING GRDTLL, SET LOCTIM TO -1
C      ******************************************************************
      IF (IERR .NE. 0) THEN
          LOCTIM = -1

C      ******************************************************************
C     OTHERWISE, CALCULATE SSM/I LOCAL TIME
C      ******************************************************************
      ELSE

C     ***************************************************************
C     UNPACK GMT HOURS FROM SSM/I DATE/TIME GROUP
C     ***************************************************************
          GMTHRS = MOD(DTGRP,10000) / 100

C     ***************************************************************
C     CALCULATE LOCAL TIME IN HOURS BY ADDING ONE HOUR TO GMT
C     FOR EACH 15 DEGREES EASTWARDS
C     ***************************************************************
          LOCHRS = GMTHRS + INT(RLON / 15.0)
          IF (LOCHRS .GT. 23) LOCHRS = LOCHRS - 24

C     ***************************************************************
C     ASSEMBLE LOCAL TIME USING LOCHRS AND MINUTES FROM DATE/TIME GROUP
```

```
C     ***************************************************************
          LOCTIM = (100 * LOCHRS) + MOD(DTGRP,100)

      ENDIF

      RETURN
      END


      SUBROUTINE SMIBOX(HEM,STYDIS,IRSSCO,TYPEID,LTT,DATE)

      INTEGER GEODAT(64,64)
      INTEGER HEM
      INTEGER I
      INTEGER IBOX
      INTEGER IGRID
      INTEGER IRSSCO(9,8)
      INTEGER ISMI(64,64)
      INTEGER J
      INTEGER JBOX
      INTEGER JGRID
      INTEGER NEFBOX, NEPH(11)
      INTEGER NLDCNT
      INTEGER NNOLND(17)
      INTEGER LTT(32768)
      INTEGER PNTCNT
      INTEGER SNOLND(17)
      INTEGER SMIDAT(64,64,8)
      INTEGER SMIHR(64,64)
      INTEGER STYDIS(8,3)
      INTEGER TYPEID(0:31)
      CHARACTER*17 FILEOUT
      CHARACTER*7 DATE

      LOGICAL DATERR
      LOGICAL GEOERR

      REAL SMIBAK(64,64)

      DATA NNOLND/2,7,8,9,41,42,49,50,55,56,57,58,59,63,0,0,0/
      DATA SNOLND/2,3,8,9,10,12,16,17,20,23,27,49,53,57,62,63,0/
      DATA NEPH /43,44,45,0,0,0,0,0,51,52,53/

      DATERR = .FALSE.
      GEOERR = .FALSE.
      NLDCNT = 1
      PNTCNT = 0

C     ****************************************************************
C     BEGIN BOX LOOP AND INITIALIZE BOX SSMI COUNTS TO ZERO
C     ****************************************************************
      DO 200 NEFBOX = 43 , 53
          IF (NEPH(NLDCNT) .EQ. NEFBOX) THEN

C     ************************************************************
C     SET NEFBOX COLUMN AND ROW
C     ************************************************************
                JBOX = (NEFBOX + 7) / 8
                IBOX = NEFBOX - ((JBOX - 1) * 8)
```

```
C       ************************************************************
C       READ GEOGRAPHY TABLE
C       ************************************************************
              CALL BOXGEO(HEM,NEFBOX,GEODAT,GEOERR)

C       ************************************************************
C       IF BOXGEO RETURNS NO ERRORS THEN READ SSM/I DATABASE
C       ************************************************************
                  IF (.NOT. GEOERR) THEN
                  CALL BOXDAT(HEM,IBOX,JBOX,NEFBOX,SMIDAT,DATE,DATERR)

C        ***********************************************************
C        IF BOXDAT RETURNS NO ERRORS THEN CALCULATE SSM/I TEMP-
C        PERATURES AND INTERPOLATE TO THE CURRENT SFCTMP CYCLE
C        ***********************************************************
                       IF (.NOT. DATERR) THEN
                       CALL BOXCLC(HEM,NEFBOX,GEODAT,SMIDAT,STYDIS,LTT,
     &                                IRSSCO,TYPEID,SMIBAK,SMIHR)
                       ENDIF
                  ENDIF

C       ************************************************************
C       IF ANY ERRORS OCCUR IN THESE SUBROUTINES THEN FILL SSM/I
C       TEMPS IN CURRENT BOX AS ZERO
C       ************************************************************
              IF (GEOERR .OR. DATERR)
     &                 CALL RCOPY(0.0,1,SMIBAK,1,4096)

                  DO 120 J=1,64
                       DO 110 I=1,64
                            ISMI(I,J) = NINT(10*SMIBAK(I,J))
                            IF (SMIBAK(I,J) .NE. 0.0)PNTCNT= PNTCNT + 1

 110                   CONTINUE
 120              CONTINUE

                  NLDCNT = NLDCNT + 1
              ELSE
                  NLDCNT = NLDCNT + 1
                  CALL RCOPY(0.0,1,SMIBAK,1,4096)
              ENDIF

          PRINT *,'NEFBOX = ',NEFBOX,' # OF POINTS PROCESSED TO THIS',
     &          ' POINT = ',PNTCNT
          PRINT *
          FILEOUT='./data/'//DATE//'OUT'

C     Write to output file

      OPEN(UNIT=200,FILE=FILEOUT,ACCESS='SEQUENTIAL',FORM='FORMATTED')

              DO 140 J=1,64
                  DO 130 I=1,64
                       WRITE(200,150) SMIBAK(I,J), I, J, NEFBOX

130               CONTINUE
140           CONTINUE

150    FORMAT(F5.1,1X,I2,1X,I2,1X,I2)
```

```
C               PRINT *,'ERROR IN SUBROUTINE SMIBOX WRITING TO FILE'
C          ENDIF

200   CONTINUE
      RETURN
      END


      SUBROUTINE SMITUN(TUNDAT,TUNERR,HEM,TYPEID,STYDIS,IRSSCO)

      INTEGER HEM
      INTEGER ISTAT
      INTEGER TUNDAT(128)
      INTEGER TYPEID(0:31), STYDIS(8,3), IRSSCO(9,8)
      INTEGER I,J,K
      LOGICAL TUNERR

C     Load the Tuning arrays here
      OPEN(UNIT=20, FILE='TYPEID', ACCESS ='SEQUENTIAL')
      OPEN(UNIT=30, FILE='STYDIS', ACCESS ='SEQUENTIAL')
      OPEN(UNIT=40, FILE='IRSSCO', ACCESS ='SEQUENTIAL')

      ISTAT=0
      K=1

      DO J=1,8
           DO I=1,9
                READ(40,100, END=50) IRSSCO(I,J)
                WRITE(TUNDAT(K),100) IRSSCO(I,J)
                ISTAT=ISTAT+1
                K=K+1
           ENDDO
      ENDDO
 50   CONTINUE

      DO J=1,3
           DO I=1,8
                READ(30,100, END=60) STYDIS(I,J)
                WRITE(TUNDAT(K),100) STYDIS(I,J)
                ISTAT=ISTAT+1
                K=K+1
           ENDDO
      ENDDO
 60   CONTINUE

      DO I=0,31
           ISTAT=ISTAT+1
           READ(20,110) TYPEID(I)
           WRITE(TUNDAT(K),100) TYPEID(I)
           K=K+1
      ENDDO
 70   CONTINUE
      print*, 'K=',K,' and ISTAT=',ISTAT

100    FORMAT(I7)
110    FORMAT(I1)

      IF (ISTAT .NE. 128) THEN
```

```
      PRINT *,'ERROR IN SUBROUTINE SMITUN IN TMPSMI'
      PRINT *,'COULD NOT READ TUNECO, ISTAT = ',ISTAT
      PRINT *,'NO SSM/I TEMPS AVAILABLE FOR HEM = ',HEM
      TUNERR = .TRUE.
    ENDIF

    RETURN
    END


    SUBROUTINE GRDTLL (NBOX,RI,RJ,RLAT,RLON,IHEM,IGMF,IGLF,ICON,IERR)

    CHARACTER*35 ERRMSG

    INTEGER ICON
    INTEGER IERR
    INTEGER IGLF
    INTEGER IGMF
    INTEGER IHEM
    INTEGER NBOX
    INTEGER NBXCOL
    INTEGER NBXROW

    REAL ABSI
    REAL ABSJ
    REAL CENTER (4)
    REAL CONVRI (3)
    REAL CONVRJ (3)
    REAL COORMX (4)
    REAL FAC
    REAL GRDSBX (4)
    REAL POLRAD (4)
    REAL RDTODG
    REAL RI
    REAL RJ
    REAL RLAT
    REAL RLIM
    REAL RLON

    DATA CENTER / 33., 65., 129., 257. /
    DATA CONVRI  / -9., -4., -18. /
    DATA CONVRJ  / -7., -4., -14. /
    DATA COORMX / 65., 129., 257., 513. /
    DATA GRDSBX / 8.0, 16.0, 32.0, 64.0 /
    DATA POLRAD / 31.204359, 62.408718, 124.81744, 249.634871 /
    DATA RDTODG / 57.295780 /

C   SET ABSI & ABSJ FOR ABSOLUTE COORDINATES; RESET LATER IF RELATIVE
C   -----VALIDATE USER ARGUMENTS AND CONVERT TO SUPERGRID IF NEEDED----
-
C   CHECK GRID MESH (IGMF); IF OUT OF RANGE, SET ERROR FLAG
C   IF GRID CONVERSION NEEDED:
C        CHECK GRID SYSTEM (IGLF); IF OUT OF RANGE, SET ERROR FLAG
C        CHECK NBOX; IF NOT ZERO, SET ERROR FLAG
C        COMPARE IGLF TO IGMF; IF INCOMPATIBLE, SET ERROR FLAG
C        CONVERT TO SUPERGRID

    IERR = 0
    ERRMSG = ' '
```

```
      ABSI = RI
      ABSJ = RJ

      IF ((IGMF .LT. 1) .OR. (IGMF .GT. 4)) THEN
           IERR = 1
           ERRMSG = 'GRID RES (IGMF) NOT 1 - 4'
      ELSE IF (IGLF .NE. 0) THEN
           IF ((IGLF .LT. 0) .OR. (IGLF .GT. 3)) THEN
                IERR = 2
                ERRMSG = 'GRID SYS (IGLF) NOT 0 - 3'
           ELSE IF (NBOX .NE. 0) THEN
                IERR = 3
                ERRMSG = 'NBOX MUST BE 0 IF IGLF IS NOT 0'
           ELSE IF ((IGLF .LT. 3) .AND. (IGMF .NE. 1)) THEN
                IERR = 4
                ERRMSG = 'IGMF MUST BE 1 IF IGLF IS 1 OR 2'
           ELSE IF ((IGLF .EQ. 3) .AND. (IGMF .NE. 2)) THEN
                IERR = 5
                ERRMSG = 'IGMF MUST BE 2 IF IGLF IS 3'
           ELSE
                ABSI = RI - CONVRI (IGLF)
                ABSJ = RJ - CONVRJ (IGLF)
           ENDIF

C     IF GRID CONVERSION NOT NEEDED:
C          CHECK NBOX; IF OUT OF RANGE, SET ERROR FLAG
C               IF NBOX = 0, RI & RJ ARE ABSOLUTE COORDINATES
C               IF NBOX > 0, RI & RJ ARE RELATIVE COORDINATES
C                    VALIDATE; IF OUT OF RANGE, SET ERROR FLAG
C                    DETERMINE BOX ROW AND COLUMN FROM THE BOX NUMBER
C          NOTE:  BOX ROW AND COLUMN ARE IN THE RANGE 0-7

      ELSE
           IF ((NBOX .LT. 0) .OR. (NBOX .GT. 64)) THEN
                IERR = 6
                ERRMSG = 'NBOX NOT IN RANGE 0 - 64'
           ELSE IF (NBOX .GT. 0) THEN
                RLIM = (GRDSBX (IGMF))
                IF ((RI .LT. 1.) .OR. (RI .GT. RLIM)) THEN
                     IERR = 7
                     ERRMSG = 'BOX RELATIVE RI NOT IN RANGE'
                ELSE IF ((RJ .LT. 1.) .OR. (RJ .GT. RLIM)) THEN
                     IERR = 8
                     ERRMSG = 'BOX RELATIVE RJ NOT IN RANGE'
                ELSE
                     NBXROW = (NBOX - 1) / 8
                     NBXCOL = NBOX - 1 - NBXROW * 8
                     ABSI = RI + (FLOAT (NBXCOL) * GRDSBX (IGMF))
                     ABSJ = RJ + (FLOAT (NBXROW) * GRDSBX (IGMF))
                ENDIF
           ENDIF
      ENDIF

C     IF NO ERRORS SO FAR:
C          VALIDATE ABSOLUTE COORDINATES; IF OUT OF RANGE, SET ERROR FLAG
C          VALIDATE ICON; IF OUT OF RANGE, SET ERROR FLAG
C     CONVERT ABSOLUTE COORDINATES TO X/Y COORDINATES RELATIVE TO THE
C     POLE
C     CALCULATE THE POINT (FAC)
C     CALCULATE THE LATITUDE
```

```
      IF (IERR .EQ. 0) THEN
           IF ((ABSI .LT. 0.5) .OR. (ABSI .GT. COORMX(IGMF))) THEN
                IERR = 9
                ERRMSG = 'ABSOLUTE I COORD OUTSIDE GRID'
           ELSE IF ((ABSJ .LT. 0.5) .OR. (ABSJ .GT. COORMX(IGMF))) THEN
                IERR = 10
                ERRMSG = 'ABSOLUTE J COORD OUTSIDE GRID'
           ELSE IF ((ICON .LT. 1) .OR. (ICON .GT. 4)) THEN
                IERR = 11
                ERRMSG = 'ICON NOT IN RANGE 1 - 4'
           ELSE
                ABSI = ABSI - CENTER(IGMF)
                ABSJ = CENTER(IGMF) - ABSJ
                FAC = SQRT(ABSI**2 + ABSJ**2)
                RLAT = 90.0 - RDTODG * 2.0 * ATAN(FAC/POLRAD(IGMF))

C      -----------------NOTES ON USE OF ABSI AND ABSJ----------------
C          IF ABSI > 0   POINT IS LEFT OF THE Y AXIS
C          IF ABSI = 0   POINT IS ON THE Y AXIS
C          IF ABSI < 0   POINT IS RIGHT OF THE Y AXIS
C          IF ABSJ > 0   POINT IS ABOVE THE X AXIS
C          IF ABSJ = 0   POINT IS ON THE X AXIS
C          IF ABSJ < 0   POINT IS BELOW THE X AXIS
C      ------------------------------------------------------------------
----------------
C          IF ABSI IS 0, RLON IS + OR - 90.0, DEPENDING ON ABSJ.
C     OTHERWISE, RLON IS THE ANGLE FROM THE X AXIS TO THE POINT.
C     DIVISION BY ABSOLUTE VALUE OF ABSI ALLOWS ABSJ TO DETERMINE
C     THE SIGN (WHETHER POINT IS ABOVE OR BELOW THE X AXIS).

                IF (ABSI .EQ. 0) THEN
                     RLON = 90.0
                     IF (ABSJ .LT. 0.0)  RLON = -90.0
                ELSE
                     RLON = RDTODG * ATAN (ABSJ / ABS (ABSI) )
                     IF (ABSI .LT. 0.0)  RLON = 180 - RLON
                ENDIF

C     ---- MODIFY LATITUDE AND LONGITUDE ACCORDING TO HEMISPHERE ----
C     NORTHERN:  LATITUDE OK; REFERENCE LONGITUDE TO 10E (POSITIVE)
C     SOUTHERN:  MAKE LAT NEGATIVE; REFERENCE LON TO 10E (POSITIVE)

                IF (IHEM .EQ. 1) THEN
                     RLON = RLON + 10.0
                ELSE IF (IHEM .EQ. 2) THEN
                     RLAT = -RLAT
                     RLON = 10.0 - RLON
                ELSE
                     IERR = 12
                     ERRMSG = 'IHEM NOT IN RANGE 1 - 2'
                ENDIF
           ENDIF
      ENDIF

C     IF NO ERRORS, ENSURE RLON HAS CORRECT SIGN AND IS IN RANGE
C     IF POSITIVE WEST (ICON = 3 OR 4), REVERSE SIGN ON LONGITUDE
C     IF ICON = 1 OR 3, RANGE IS -180 TO +180
C     IF ICON = 2 OR 4, RANGE IS +0 TO +360
C     MAKE RLON POSITIVE UNLESS IT'S -0
```

```
      IF (IERR .EQ. 0) THEN
           IF (ICON .EQ. 3 .OR. ICON .EQ. 4) THEN
                RLON = -RLON
           ENDIF
           IF (ICON .EQ. 1 .OR. ICON .EQ. 3) THEN
                IF (RLON .LE. -180.0) THEN
                     RLON = RLON + 360.0
                ELSE IF (RLON .GT. +180.0) THEN
                     RLON = RLON - 360.0
                ENDIF
           ELSE IF (ICON .EQ. 2 .OR. ICON .EQ. 4) THEN
                IF (RLON .LT. 0.0 .AND. RLON .NE. -0.0) THEN
                     RLON = RLON + 360.0
                ELSE IF (RLON .GE. 360.0) THEN
                     RLON = RLON - 360.0
                ENDIF
           ENDIF

C     IF ERRORS OCCURRED, PRINT OUT USER ARGUMENTS
      ELSE
           WRITE (6,800) IERR, ERRMSG
           WRITE (6,820) NBOX, RI, RJ, RLAT
           WRITE (6,840) RLON, IHEM, IGMF, IGLF, ICON
      ENDIF

      RETURN

  800 FORMAT (' ** GRDTLL: ', 9('** ERROR **'), / ' ** GRDTLL:  ERROR',
     &          ' DETECTED -- IERR = ',I2, ' -- ', A35, T110, '**')
  820 FORMAT (' ** GRDTLL:  NBOX = ', I2, 10X, 'RI =', E14.7, 9X,
     &         'RJ =', E14.7, 9X, 'RLAT =', E14.7, 9X, T110, '**')
  840 FORMAT (' ** GRDTLL:  RLON =', E14.7, 10X, 'IHEM = ', I1, 10X,
     &         'IGMF = ', I1, 10X, 'IGLF = ', I1, 10X, 'ICON = ', I1,
     &         T110, '**')
      END


      SUBROUTINE RCOPY(VALUE,INIT,VAR,INC,FIN)

      INTEGER INDEX
      INTEGER INIT
      INTEGER INC
      INTEGER FIN
      REAL VALUE
      REAL VAR(FIN)

      DO 100 INDEX = INIT,FIN,INC
           VAR(INDEX) = VALUE
100   CONTINUE

      RETURN
      END


      SUBROUTINE COPY(VALUE,INIT,VAR,INC,FIN)

      INTEGER INDEX
      INTEGER INIT
```

```
      INTEGER INC
      INTEGER FIN
      INTEGER VALUE
      INTEGER VAR(FIN)

      DO 100 INDEX = INIT,FIN,INC
          VAR(INDEX) = VALUE
100   CONTINUE

      RETURN
      END
```

# Appendix B: CAL/VAL Code

```
*   Calval Algorithm
Program CALVAL
      implicit none

      REAL H19(64,64,64), H37(64,64,64), H85(64,64,64), V19(64,64,64)
      REAL V22(64,64,64), V37(64,64,64), V85(64,64,64), APD(64,64,64)
      REAL DEGK(64,64,64), LAT, LON
      INTEGER TYPE(64,64,64), ERR, NEPHBOX(6)
      INTEGER I,J, N, K, IE, JE, NE, ILEN
      CHARACTER DATE*7,  FILENAME*60
      CHARACTER DATFILE*23
      CHARACTER FILE19H*60, FILE19V*60,FILE22V*60,FILE37H*60
      CHARACTER FILE37V*60, FILE85H*60, FILE85V*60
      PARAMETER(IE=64, JE=64, NE=64)
      DATA NEPHBOX /43,44,45,51,52,53/

      PRINT *, 'Enter date in DDMMMYY format, e.g., 01AUG96'
      READ(*, FMT=50) DATE
50    FORMAT(A7)

*****67**0*********0*********0*********0*********0*********0*********0*2
* VARIABLES:
*   NAME                              DESCRIPTION
*   =====  ==================================================
*   APD    MEAN POLARIZATION FOR 19GHz AND 37GHz CHANNELS
*   DEGK   DEGREES KELVIN
*   H19    HORIZONTAL 19GHz CHANNEL BRIGHTNESS TEMPERATURE
*   H37    HORIZONTAL 37GHz CHANNEL BRIGHTNESS TEMPERATURE
*   H85    HORIZONTAL 85GHz CHANNEL BRIGHTNESS TEMPERATURE
*   TYPE   0 = UNDETERMINED SOIL TYPE
*          1 = DRY ARABLE SOIL
*          2 = MOIST ARABLE SOIL
*          3 = SEMI-DESERT SOIL
*          4 = DESERT SOIL
*          5 = WET SOIL
*          6 = DENSE VEGETATION
*          7 = COMPOSITE WATER AND VEGETATION
*          8 = LESS DENSE VEGETATION
*   V19    VERTICAL 19GHz CHANNEL BRIGHTNESS TEMPERATURE
*   V22    VERTICAL 22GHz CHANNEL BRIGHTNESS TEMPERATURE
*   V37    VERTICAL 37GHz CHANNEL BRIGHTNESS TEMPERATURE
*   V85    VERTICAL 85GHz CHANNEL BRIGHTNESS TEMPERATURE
*   N      NUMBER OF RTNEPH BOXES
*   K      NEPH BOX INDEX NUMBER
************************************************************************

      DATA TYPE /262144*0/
      ILEN=IE*JE*4

      FILE19H = './data/'//DATE//'19H.DAT'
      FILE19V = './data/'//DATE//'19V.DAT'
      FILE22V = './data/'//DATE//'22V.DAT'
      FILE37H = './data/'//DATE//'37H.DAT'
      FILE37V = './data/'//DATE//'37V.DAT'
      FILE85H = './data/'//DATE//'85H.DAT'
      FILE85V = './data/'//DATE//'85V.DAT'
```

```
      PRINT*, FILENAME
      DATFILE = './data/'//DATE//'TEMP.DAT'
*  Open the respective datafiles for the channel Tbs

      OPEN(UNIT=20, FILE=FILE19H, ACCESS='SEQUENTIAL')
      OPEN(UNIT=30, FILE=FILE19V, ACCESS='SEQUENTIAL')
      OPEN(UNIT=40, FILE=FILE22V, ACCESS='SEQUENTIAL')
      OPEN(UNIT=50, FILE=FILE37H, ACCESS='SEQUENTIAL')
      OPEN(UNIT=60, FILE=FILE37V, ACCESS='SEQUENTIAL')
      OPEN(UNIT=70, FILE=FILE85H, ACCESS='SEQUENTIAL')
      OPEN(UNIT=80, FILE=FILE85V, ACCESS='SEQUENTIAL')

*  Open the output file
      OPEN(UNIT=11, FILE=DATFILE, STATUS='UNKNOWN',
     &  FORM='FORMATTED',ACCESS='SEQUENTIAL',IOSTAT=ERR)
            IF (ERR .NE. 0) THEN
                  PRINT*, 'ERROR IN WRITE'
            ENDIF

*  READ IN THE RESPECTIVE CHANNEL DATA, OVER CONUS ONLY
*
      DO N = 1,6
            DO J=1,JE
                  DO I=1,IE
                  READ(20, FMT=100) H19(I,J,NEPHBOX(N)),I,J,K
                  READ(30, FMT=100) V19(I,J,NEPHBOX(N)),I,J,K
                  READ(40, FMT=100) V22(I,J,NEPHBOX(N)),I,J,K
                  READ(50, FMT=100) H37(I,J,NEPHBOX(N)),I,J,K
                  READ(60, FMT=100) V37(I,J,NEPHBOX(N)),I,J,K
                  READ(70, FMT=100) H85(I,J,NEPHBOX(N)),I,J,K
                  READ(80, FMT=100) V85(I,J,NEPHBOX(N)),I,J,K

100   FORMAT(F5.1,1X,I2,1X,I2,1X,I2)

                  ENDDO
            ENDDO
      ENDDO

      DO N=1,6
            DO J=1,JE
                  DO I=1,IE
            APD(I,J,NEPHBOX(N)) = (V19(I,J,NEPHBOX(N))
     &      +  V37(I,J,NEPHBOX(N))
     &      - H19(I,J,NEPHBOX(N)) - H37(I,J,NEPHBOX(N)))  / 2.0

200   CONTINUE

* CHECK IF THE SURFACE TYPE IS DRY ARABLE SOIL (TYPE = 1)

      IF ( ((V22(I,J,NEPHBOX(N))-V19(I,J,NEPHBOX(N))) .LE. 4.0) .AND.
      &     (APD(I,J,NEPHBOX(N)) .LE. 9.8)          .AND.
      &     (APD(I,J,NEPHBOX(N)) .GT. 4.0)          .AND.
      &     ((V37(I,J,NEPHBOX(N))-V19(I,J,NEPHBOX(N))) .GE. -6.5) .AND.
      &     ((V85(I,J,NEPHBOX(N))-V37(I,J,NEPHBOX(N))) .LT. 0.5)  .AND.
      &     ((V85(I,J,NEPHBOX(N))-V37(I,J,NEPHBOX(N))) .GE. -5.0) .AND.
      &     ((H85(I,J,NEPHBOX(N))-H37(I,J,NEPHBOX(N))) .LT. 4.2 )) THEN
                  TYPE(I,J,NEPHBOX(N)) = 1
*
```

```
* CHECK IF THE SURFACE TYPE IS MOIST ARABLE SOIL (TYPE = 2)
*
      ELSEIF ( ((V22(I,J,NEPHBOX(N))-V19(I,J,NEPHBOX(N))).LE. 4.0).AND.
     &      (APD(I,J,NEPHBOX(N)) .LT. 19.7)        .AND.
     &      (APD(I,J,NEPHBOX(N)) .GT. 4.0)         .AND.
     &      ((V37(I,J,NEPHBOX(N))-V19(I,J,NEPHBOX(N))) .GE. -6.5) .AND.
     &      ((V85(I,J,NEPHBOX(N))-V37(I,J,NEPHBOX(N))) .GE. 0.5)  .AND.
     &      ((V85(I,J,NEPHBOX(N))-V37(I,J,NEPHBOX(N))) .LT. 4.0)  .AND.
     &      ((H85(I,J,NEPHBOX(N))-H37(I,J,NEPHBOX(N))) .LT. 4.2  )) THEN
                 TYPE(I,J,NEPHBOX(N)) = 2
*
* CHECK IF THE SURFACE TYPE IS SEMI-DESERT SOIL (TYPE = 3)
*
      ELSEIF ( ((V22(I,J,NEPHBOX(N))-V19(I,J,NEPHBOX(N))).LE. 4.0) .AND.
     &      (APD(I,J,NEPHBOX(N)) .LT. 19.7)        .AND.
     &      (APD(I,J,NEPHBOX(N)) .GT. 9.8)         .AND.
     &      ((V85(I,J,NEPHBOX(N))-V37(I,J,NEPHBOX(N))) .LT. 0.5)  .AND.
     &      ((H37(I,J,NEPHBOX(N))-H19(I,J,NEPHBOX(N))) .LT. -1.8) .AND.
     &      ((H85(I,J,NEPHBOX(N))-H37(I,J,NEPHBOX(N))) .LT. 6.0 )) THEN
                 TYPE(I,J,NEPHBOX(N)) = 3
*
* CHECK IF THE SURFACE TYPE IS DESERT SOIL (TYPE = 4)
*
      ELSEIF ( ((V22(I,J,NEPHBOX(N))-V19(I,J,NEPHBOX(N))) .LE. 2.0).AND.
     &      (APD(I,J,NEPHBOX(N)) .GE. 19.7)        .AND.
     &      (V19(I,J,NEPHBOX(N)) .GT. 268.0)       .AND.
     &      ((H85(I,J,NEPHBOX(N))-H37(I,J,NEPHBOX(N))) .GT. -1.0 )) THEN
                 TYPE(I,J,NEPHBOX(N)) = 4
*
* CHECK IF THE SURFACE TYPE IS WET SOIL (TYPE = 5)
*
      ELSEIF ( ((V22(I,J,NEPHBOX(N))-V19(I,J,NEPHBOX(N))) .LE. 4.0).AND.

     &      (APD(I,J,NEPHBOX(N)) .GE. 6.4)         .AND.
     &      ((V37(I,J,NEPHBOX(N))-V19(I,J,NEPHBOX(N))) .GE. -6.5) .AND.
     &      ((V85(I,J,NEPHBOX(N))-V37(I,J,NEPHBOX(N))) .GE. 0.5)  .AND.
     &      ((H85(I,J,NEPHBOX(N))-H37(I,J,NEPHBOX(N))) .GE. 4.2)) THEN
                 TYPE(I,J,NEPHBOX(N)) = 5
*
* CHECK IF THE SURFACE TYPE IS DENSE VEGETATION (TYPE = 6)
*
      ELSEIF ( ((V22(I,J,NEPHBOX(N))-V19(I,J,NEPHBOX(N))) .LE. 4.0).AND.
     &      (APD(I,J,NEPHBOX(N)) .LE. 1.9)         .AND.
     &      ((V85(I,J,NEPHBOX(N))-V37(I,J,NEPHBOX(N))) .GE. -1.0) .AND.
     &      ((H85(I,J,NEPHBOX(N))-H37(I,J,NEPHBOX(N))) .LT. 4.5)  .AND.
     &      (V19(I,J,NEPHBOX(N)) .GT. 262.0)       ) THEN
                 TYPE(I,J,NEPHBOX(N)) = 6
*
* CHECK IF THE SURFACE TYPE IS COMPOSIT WATER AND VEGETATION (TYPE = 7)
*
      ELSEIF ( ((V22(I,J,NEPHBOX(N))-V19(I,J,NEPHBOX(N))) .LE. 4.0).AND.
     &      (APD(I,J,NEPHBOX(N)) .LT. 6.4)         .AND.
     &      ((V85(I,J,NEPHBOX(N))-V37(I,J,NEPHBOX(N))) .GE. -1.0) .AND.
     &      ((H85(I,J,NEPHBOX(N))-H37(I,J,NEPHBOX(N))) .GE. 4.5)  .AND.
     &      (V37(I,J,NEPHBOX(N))-H37(I,J,NEPHBOX(N)) .GT. 257.0) ) THEN
                 TYPE(I,J,NEPHBOX(N)) = 7
*
* CHECK IF THE SURFACE TYPE IS LESS DENSE VEGETATION (TYPE = 8)
```

```
*
      ELSEIF ( ((V22(I,J,NEPHBOX(N))-V19(I,J,NEPHBOX(N))) .LE. 4.0).AND.
     &      (APD(I,J,NEPHBOX(N)) .LE. 4.0)          .AND.
     &      (APD(I,J,NEPHBOX(N)) .GT. 1.9)          .AND.
     &      ((V85(I,J,NEPHBOX(N))-V37(I,J,NEPHBOX(N))) .GE. -1.0) .AND.
     &      ((H85(I,J,NEPHBOX(N))-H37(I,J,NEPHBOX(N))) .LT. 4.5)  .AND.
     &      (V19(I,J,NEPHBOX(N)) .GT. 262.0)        ) THEN
                TYPE(I,J,NEPHBOX(N)) = 8
      ENDIF
*
* SURFACE TEMP FOR MOIST ARABLE SOIL OR WET SOIL (TYPES 2 OR 5)
*
       IF (TYPE(I,J,NEPHBOX(N)) .EQ. 2 .OR. TYPE(I,J,NEPHBOX(N)) .EQ. 5)
     &  THEN
     & DEGK(I,J,NEPHBOX(N)) = 23.16 - 0.1873*H19(I,J,NEPHBOX(N))
     &  + 0.5221*V22(I,J,NEPHBOX(N)) - 0.6271*V37(I,J,NEPHBOX(N)) +
     &  1.2320*V85(I,J,NEPHBOX(N))
       ENDIF
*
* SURFACE TEMP FOR DRY ARABLE SOIL, SEMI-DESERT, OR DESERT (TYPES 1, 3,
* OR 4)
*
       IF (TYPE(I,J,NEPHBOX(N)) .EQ. 1 .OR. TYPE(I,J,NEPHBOX(N))
     & .EQ. 3 .OR. TYPE(I,J,NEPHBOX(N)) .EQ. 4) THEN
     &  DEGK(I,J,NEPHBOX(N)) = 72.68 - 0.4598*H19(I,J,NEPHBOX(N))
     &    + 0.5984*V22(I,J,NEPHBOX(N))
     &    + 0.8828*V37(I,J,NEPHBOX(N)) - 0.2623*V85(I,J,NEPHBOX(N))
       ENDIF

*
* SURFACE TEMP FOR DENSE VEGETATION, OR COMPOSITE WATER AND VEGETATION
* (TYPES 6 OR 7)
*
      IF (TYPE(I,J,NEPHBOX(N)) .EQ. 6 .OR. TYPE(I,J,NEPHBOX(N))
     &  .EQ. 7) THEN
            DEGK(I,J,NEPHBOX(N)) = 29.94 - 1.2784*H19(I,J,NEPHBOX(N))
     &   + 0.8800*V22(I,J,NEPHBOX(N))
     &   + 0.5933*V37(I,J,NEPHBOX(N)) +  0.7229*V85(I,J,NEPHBOX(N))
       ENDIF
*
* SURFACE TEMP FOR LESS DENSE VEGETATION (TYPE 8)
*
       IF (TYPE(I,J,NEPHBOX(N)) .EQ. 8) THEN
            DEGK(I,J,NEPHBOX(N)) =  6.97 - 0.6226*H19(I,J,NEPHBOX(N))
     &   + 0.2716*V22(I,J,NEPHBOX(N))
     &   - 0.1297*V37(I,J,NEPHBOX(N)) + 1.4820*V85(I,J,NEPHBOX(N))
       ENDIF

                ENDDO
            ENDDO
      ENDDO

      DO N = 1, 6
            DO J=1,JE
                DO I=1,IE
                      CALL CONVRT(I,J,K,LAT,LON)

                      WRITE(11,FMT=300) DEGK(I,J,NEPHBOX(N)),I,J,
```

```
      &  NEPHBOX(N),TYPE(I,J,NEPHBOX(N)), LAT,LON

300   FORMAT(F6.2,1X,I2,1X,I2,1X,I2,3X,I2,1X,F6.2,1X,F7.2)
                ENDDO
            ENDDO
      ENDDO
      END

      SUBROUTINE CONVRT(I,J,K,LAT,LON)
C     ARGUMENTS:
      INTEGER*4 FLAG
      INTEGER*4 POLAR

      REAL*4 LON, LAT        ! LONGITUDE, LATITUDE
      REAL*4 A, B            ! 1/8TH-MESH POLAR STEREOGRAPHIC (A,B)
C
C     THIS SUBROUTINE RETURNS THE LAT/LONG ASSOC. WITH THE SUPERNEPH
C     GRIDPOINTS A,B (512 BY 512 HEMISPHERIC BOX)
      FLAG = 2
C     CONVERT THE I,J,K INDICES TO A,B
      IF (K .EQ. 43) THEN
            A = I + 128
            B = J + 320
      ELSEIF (K .EQ. 44) THEN
            A = I + 192
            B = J + 320
      ELSEIF (K .EQ. 45) THEN
            A = I + 256
            B = J + 320
      ELSEIF (K .EQ. 51) THEN
            A = I + 128
            B = J + 384
      ELSEIF (K .EQ. 52) THEN
            A = I + 192
            B = J + 384
      ELSEIF (K .EQ. 53) THEN
            A = I + 256
            B = J + 384
      ENDIF

      STATUS = POLAR(LON, LAT, A, B, FLAG )

      IF (FLAG .EQ. 1) THEN
            PRINT 50, '(LAT,LON) COORDS (', LAT, ',', LON, ')'
      ELSEIF (FLAG .EQ. 2) THEN
      ENDIF

50      FORMAT ( ' ', 1X, A19, F9.5, A1, F10.5, A1)
60      FORMAT ( ' ', 1X, A15 ,F6.1, A1 ,F6.1, A1)
      RETURN
      END

C*******************************************************
      INTEGER*4 FUNCTION POLAR( LON, LAT, A, B, FLAG )
C
C     ROBERT P. D'ENTREMONT   AER   11 NOVEMBER 1993
C
C        THIS FUNCTION CONVERTS FROM/TO POLAR STEREOGRAPHIC GRID
```

```
C     COORDINATES FROM LATITUDE-LONGITUDE COORDINATES.  THE GRID
C     RESOLUTION IS FIXED AT 1/8TH MESH SIZE, AND THE LOCATION
C     IS FIXED TO NORTHERN HEMISPHERE POINTS ONLY.  REFERENCE
C     LONGITUDE IS 80 DEG W (BOTTOM Y-AXIS).
C
C     11-NOV-1993     V1.0   FOR RTNEPH APPLICATIONS
C
C     ARGUMENTS:
C     ----------
C
C        REAL*4 LON   LONGITUDE (EAST +VE)
C        REAL*4 LAT   LATITUDE (NORTH +VE)
C        REAL*4 I     POLAR 1/8TH-MESH COLUMN NUMBER
C        REAL*4 J     POLAR 1/8TH-MESH ROW NUMBER
C
C        INTEGER*4 FLAG = 1 FOR (LON,LAT) TO (I,J) TRANSFORMATION,
C                       = 2 FOR (I,J) TO (LON,LAT) TRANSFORMATION
C
      INTEGER*4 FLAG, QUAD
      INTEGER*4 COMPLET

      REAL*4 LON, LAT,          ! LONGITUDE, LATITUDE
      & A, B,                     ! 1/8TH-MESH POLAR STEREOGRAPHIC (I,J)
      & PI / 3.1415926535 /,      ! PI
      & PI_180 / .01745329 /,     ! # RADIANS PER DEGREE
      & RE / 6371.2213 /,         ! EARTH RADIUS (KM)
      & RO,
      & HEMI                      ! HEMISPHERE (0=NORTH, 1=SOUTH)

      REAL*4 PHI, LAMBDA          ! TEMPORARY LAT, LON VARIABLES

      REAL*4 A_NP / 257. /,       ! NORTH POLE 1/16TH-MESH COLUMN NUMBER
      &       B_NP / 257. /       ! NORTH POLE 1/16TH-MESH ROW NUMBER

      REAL*4 RESOLUTION / 47.6250 /    ! KM PER 1/8TH-MESH BOX AT 60 N

      POLAR = 1
      COMPLET = 1
C
C     CHECK ON INPUT BOUNDS
C
      IF( FLAG .EQ. 1 ) THEN
            IF( LON .LT. -180. .OR. LON .GT. 180. ) COMPLET = 0
            IF( LAT .LT.    0. .OR. LAT .GT.  90. ) COMPLET = 0
      ELSE IF( FLAG .EQ. 2 ) THEN
            IF( A .LT. 1. .OR. A .GT. 513. ) COMPLET = 0
            IF( B .LT. 1. .OR. B .GT. 513. ) COMPLET = 0
      END IF

      POLAR = COMPLET

      DO WHILE( POLAR .NE. 0 )
C
      IF( FLAG .EQ. 2 ) THEN  ! CONVERT TO (LON,LAT)
C
C     COMPUTE THE QUADRANT-RELATIVE LONGITUDE
C
                  DELTA_A = A - A_NP                  ! X
```

```
            DELTA_B = B - B_NP                    ! Y

      IF( DELTA_A .NE. 0 ) THEN
      THETA = ABS( ATAN( DELTA_B / DELTA_A ) )    ! TAN**-1(Y/X)
      ELSE
            THETA = PI / 2.
      END IF
C
C     DETERMINE THE QUADRANT OF THETA ON THE POLAR STEREOGRAPHIC PLANE
C
      IF( DELTA_A .GE. 0 .AND. DELTA_B .LE. 0 ) THEN
            THETA = THETA * 180. / PI                    ! QUADRANT I
      ELSE IF( DELTA_A .LT. 0 .AND. DELTA_B .LE. 0 ) THEN
            THETA = 180. - ( THETA * 180. / PI )         ! QUADRANT II
      ELSE IF( DELTA_A .LT. 0 .AND. DELTA_B .GT. 0 ) THEN
            THETA = 180. + ( THETA * 180. / PI )         ! QUADRANT III
      ELSE IF( DELTA_A .GE. 0 .AND. DELTA_B .GT. 0 ) THEN
            THETA = 360. - THETA * 180. / PI             ! QUADRANT IV
      END IF
C
C     THETA NOW RANGES FROM 0 TO 360 DEG.  CONVERT FROM RELATIVE
C     LONGITUDE TO ABSOLUTE LONGITUDE.  REFERENCE LONGITUDE IS
C     10 E (+ X-AXIS)
C
      LON = THETA + 10.
      IF( LON .GT. 180. ) LON = LON - 360.
C
C     COMPUTE DISTANCE BETWEEN THE NORTH POLE AND THE POINT (A,B)
C     (UNITS ARE 1/8TH-MESH POINTS)
C
      RHO = SQRT( (A-A_NP)*(A-A_NP) + (B-B_NP)*(B-B_NP) ) * RESOLUTION
      LAT = 90. - 2. * ( 180. / PI ) * ATAN( RHO / 11888.44785 )
      COMPLET = 1                 ! SUCCESSFUL COMPLETION
      END IF

      POLAR = 0                    ! END THE DO-WHILE LOOP AFTER
                                   !  THE FIRST TIME THROUGH
      END DO

      POLAR = COMPLET              ! SET RETURN VALUE TO COMPLETION FLAG
      IF( POLAR .EQ. 0 ) THEN      ! POINT OFF NORTHERN HEMISPHERE
            A = -1.0
            B = -1.0
      END IF

      RETURN
      END
```

## Bibliography


Chen, H. S. Space Remote Sensing Systems. Orlando: Academic Press, 1985.

Deuel, Greg. "Special Sensor Microwave Imager (SSM/I) Sensor Document". WWWeb, http://eosims.msfc.nasa.gov:572 (24 Jun 96).

Devore, Jay L. Probability and Statistics for Engineering and the Sciences. Belmont, CA: Wadsworth Publishing, 1995.

Hamill, Thomas M. and others. "A Description of the Air Force Real-Time Nephanalysis Model", Weather and Forecasting, American Meteorological Society, 7: 288-306 (June 1992).

Hollinger, J. P. and R. C. Lo. SSM/I Project Summary Report. Naval Research Laboratory: Washington, D.C., 20 April 1983.

Hollinger, J. P. DMSP Special Sensor Microwave/Imager Calibration/Validation, Final Report Vol II. Naval Research Laboratory: Washington, D.C., 20 May 1991.

Kidder, Stanley Q. and Thomas H. Vonder Haar. Satellite Meteorology. San Diego: Academic Press, 1995.

Kopp, Thomas J. and others. "Application of SSM/I to a Real-Time Automated Satellite Nephanalysis Model". 7th Conference of Satellite Meteorology and Oceanography: 247-250 (1994).

Kopp, Thomas J. The Air Force Global Weather Central Surface Temperature Model. AFGWC/TN-95/004. Offutt AFB NE: AFGWC, December 1995.

McFarland, Marshall J. and others. Land Parameter Algorithm Validation and Calibration, DMSP SSM/I CAL/VAL, Final Report, Vol II, Chapter 9. Naval Research Laboratory: Washington, D.C., 20 May 1991.

Menke, William. Geophysical Data Analysis: Discrete Inverse Theory. San Diego: Academic Press, 1989.

Rees, W. G. Physical Principles of Remote Sensing. New York: Cambridge University Press, 1990.

Rodgers, Clive D. "Characterization and Error Analysis of Profiles Retrieved From Remote Sounding Measurements". Journal of Geophysical Research, Vol 95, No D5: 5587-5595 (1990).

Twomey, S. Introduction to the Mathematics of Inversion in Remote Sensing and Indirect Measurements. Amsterdam, The Netherlands: Elsevier Scientific Publishing Company, 1977.

Ulaby, Fawwaz T. and others. Microwave Remote Sensing: Active and Passive, Vol 1. Reading MA: Addison-Wesley Publishing Company, 1981.

Wilks, Daniel S. Statistical Methods in the Atmospheric Sciences. San Diego: Academic Press, 1995.

# Vita

Capt Ronald L. Comoglio [redacted] He graduated from Shades Valley High School, Birmingham , Alabama, in 1984 and entered undergraduate studies at the University of Alabama in Tuscaloosa, Alabama. He graduated with a Bachelor of Science in Electrical Engineering and received his commission in December 1988. His first assignment was at the Florida State University, Tallahassee, Florida, where he completed the Basic Meteorology Program and received certification for entry to the USAF's Air Weather Service.

In April 1990, Capt Comoglio was assigned as a Staff Meteorologist to the Aeronautical System Division, Wright-Patterson AFB, Ohio, where he provided atmospheric consulting support to the F-15E, F-16, and C-17 System Program Offices and the 4950th Test Wing.

In July 1992, Capt Comoglio assumed the duties of Wing Weather Officer to the 401st (later 31st) Fighter Wing, where he conducted weather support for the beddown of two F-16 squadrons and for the varied aircraft participating in the NATO air operation DENY FLIGHT, which he continued until his departure in July 1995. During that time, Capt Comoglio also held the positions of Weather Flight Commander and Staff Weather Officer to 16th Air Force. After returning from overseas duty, Capt Comoglio entered the School of Engineering, Air Force Institute of Technology.

Capt Comoglio is married to the former Rebekah Beth Gardner, formerly of Colorado Springs, Colorado, and they have three children: Heather, Jonathan and Chiara.

[redacted]

# REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

| 1. AGENCY USE ONLY (Leave blank) | 2. REPORT DATE | 3. REPORT TYPE AND DATES COVERED |
|---|---|---|
| | March 1997 | |

**4. TITLE AND SUBTITLE**

A VALIDATION STUDY OF THE SSM/I TEMPERATURE ALGORITHM AND COMPARISON WITH THE CAL/VAL LAND SURFACE TEMPERATURES

**5. FUNDING NUMBERS**

**6. AUTHOR(S)**

Captain Ronald L. Comoglio, USAF

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

AFIT/ENP
Maj Clifton E. Dungey, USAF

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

HQ AFGWC/SYSM (Dr Thomas J. Kopp)
106 Peacekeeper Drive STE 2N3
Offutt AFB, NE 68113-4039

**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES**

**12a. DISTRIBUTION / AVAILABILITY STATEMENT**

For Public Release; Distribution Unlimited

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

The USAF operational cloud model, the Real-Time Nephanalysis (RTNEPH), requires infrared and visual scene scans as well as input from the Special Sensor Microwave Imager (SSM/I), featured on the Defense Meteorological Satellite Program (DMSP) polar orbiters. Additionally, SSM/I provides input to the Air Force Global Weather Center (AFGWC) surface temperature model (SFCTMP). The global surface temperatures are required for identifying which pixel of a scene is cloud and which is ground and as a first guess input to initialize NWP models. The SSM/I provides brightness temperatures sampled at four discrete frequencies and dual polarizations to produce surface temperatures. Two algorithms are operationally available: one currently in use at AFGWC called TMPSMI, which feeds RTNEPH and SFCTMP; the other an earlier model created by the U. S. Navy during the calibration/validation of the SSM/I sensor, aptly named the CAL/VAL temperature algorithm. Using SSM/I brightness temperatures for the CONUS from days in August and October 1996, this study runs both algorithms and compares the output with observations collected at U.S. reporting sites. The results indicate both models produce similar output, exhibiting a cold bias with respect to the observed surface temperature. Furthermore, the CAL/VAL temperatures approached the observed temperature values more closely than did the SFCTMP output.

**14. SUBJECT TERMS**

MICROWAVE, RADIOMETER, SSM/I, SATELLITE, DMSP, SURFACE TEMPERATURE, CLOUD MODELING

**15. NUMBER OF PAGES**

85

**16. PRICE CODE**

| 17. SECURITY CLASSIFICATION OF REPORT | 18. SECURITY CLASSIFICATION OF THIS PAGE | 19. SECURITY CLASSIFICATION OF ABSTRACT | 20. LIMITATION OF ABSTRACT |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | |