NOAA Technical Report NESDIS 110

AN ALGORITHM FOR CORRECTION OF NAVIGATION ERRORS IN AMSU-A DATA

Washington, D.C.
December 2002

U.S. DEPARTMENT OF COMMERCE
National Oceanic and Atmospheric Administration
National Environmental Satellite, Data, and Information Service

NOAA TECHNICAL REPORTS

National Environmental Satellite, Data, and Information Service

The National Environmental Satellite, Data, and Information Service (NESDIS) manages the Nation's civil Earth-observing satellite systems, as well as global national data bases for meteorology, oceanography, geophysics, and solar-terrestrial sciences. From these sources, it develops and disseminates environmental data and information products critical to the protection of life and property, national defense, the national economy, energy development and distribution, global food supplies, and the development of natural resources.

Publication in the NOAA Technical Report series does not preclude later publication in scientific journals in expanded or modified form. The NESDIS series of NOAA Technical Reports is a continuation of the former NESS and EDIS series of NOAA Technical Reports and the NESC and EDS series of Environmental Science Services Administration (ESSA) Technical Reports.

A limited number of copies are available by contacting Jessica Pejsa, NOAA/NESDIS, E/RA2, 5200 Auth Road, Room 601, Camp Springs, Maryland 20746, (301) 763-8282. Copies can also be ordered from the National Technical Information Service (NTIS), U.S. Department of Commerce, Sills Bldg., 5285 Port Royal Road, Springfield, VA 22161, (703) 487-4650 (prices on request for paper copies or microfiche, please refer to PB number when ordering). A partial listing of more recent reports appear below:

NESDIS 71 Spectral Radiance-Temperature Conversions for Measurements by AVHRR Thermal Channels 3,4,5. Paul A. Davis, August 1993.

NESDIS 72 Summary of the NOAA/NESDIS Workshop on Development of a Global Satellite/in Situ Environmental Database. Edited by K.P. Gallo and D.A. Hastings, August 1993.

NESDIS 73 Intercomparison of the Operational Calibration of GOES-7 and METEOSAT-3/4. W. Paul Menzel, Johannes Schmetz, Steve Nieman, Leo Van de Berg, Volker Gaertner, and Timothy J. Schmit, September 1993.

NESDIS 74 Dobson Data Re-Evaluation Handbook. R. D. Hudson and W.G. Planet (Eds), October 1993.

NESDIS 75 Detection and Analysis of Fog at Night Using GOES Multispectral. Gary P. Ellrod, February 1994.

NESDIS 76 TOVS Operational Sounding Upgrades: 1990-1992. A. Reale, M. Chalfant, R. Wagoner, T. Gardner and L. Casey, March 1994.

NESDIS 77 NOAA Polar Satellite Calibration: A System Description. Cecil A. Paris, April 1994.

NESDIS 78 Post-Launch Calibration of the Visible and Near Infrared Channels of the Advanced Very High Resolution Radiometer on NOAA-7,-9, and -11 Spacecraft. C. R. Nagaraja Rao and Jianhua Chen, April 1994.

NESDIS 79 Quality Control and Processing of Historical Oceanographic Nutrient Data. Margarita E. Conkright, Timothy P. Boyer and Sydney Levitus, April 1994.

NESDIS 80 Catalogue of Heavy Rainfall Cases of Six Inches or More Over the Continental U.S. During 1993. Richard Borneman and Charles Kadin, August 1994.

NESDIS 81 Quality Control and Processing of Historical Oceanographic Temperature, Salinity, and Oxygen Data. Timothy Boyer and Sydney Levitus, August 1994.

NESDIS 82 An Introduction to the GOES I-M Imager and Sounder Instruments and the GVAR Retransmission Format. Raymond Komajda (Mitre Corp), November 1994.

NESDIS 83 Tropical Cyclone Motion Forecasting Using Satellite Water Vapor Imagery. Vernon F. Dvorak and H. Michael Mogil, December 1994.

NOAA Technical Report NESDIS 110

AN ALGORITHM FOR CORRECTION OF NAVIGATION ERRORS IN AMSU-A DATA

Seiichiro Kigawa
and
Michael P. Weinreb
NOAA/NESDIS/ORA/CRAD
5200 Auth Road, WWB/810
Camp Springs, MD 20746

Washington, DC
December 2002

U.S. DEPARTMENT OF COMMERCE
Donald L. Evans, Secretary

National Oceanic and Atmospheric Administration
Vice Admiral Conrad C. Lautenbacher, Jr., U.S. Navy (Ret.), Under Secretary

National Environmental Satellite, Data, and Information Service
Gregory W. Withee, Assistant Administrator

TABLE OF CONTENTS

Table Captions

Figure Captions

An Algorithm for Correction of Navigation Errors in AMSU-A Data

Seiichiro Kigawa[1] and Michael P. Weinreb
Climate Research and Applications Division
Office of Research and Applications
NESDIS/NOAA
5200 Auth Road, Camp Springs, MD 20746

Abstract

An algorithm for correction of the navigation errors in the Advanced Microwave Sounding Unit-A (AMSU-A) data is developed. The navigation information, consisting of latitudes and longitudes associated with individual AMSU-A field of views, are generated by the NOAA Operational Navigation Software Package and appended in the AMSU-A data sets. Any navigation error can have significant adverse effect on many products derived from the AMSU-A data. The algorithm can correct certain navigation errors and improve the accuracy of geolocation information of the AMSU-A data. Comparison of the imagery of the AMSU-A with that of AMSU-B was used to measure the errors in the AMSU-A geolocation information. Test results using this analysis show that a roll angle error of the order of 1° exists in the NOAA-15 AMSU-A channels 1 and 2. A stand-alone software program, implementing the algorithm described below, was developed to correct the AMSU-A geolocation information. The software requires only the geolocation information of AMSU-A level 1B data and knowledge of the AMSU-A alignment errors, and can be applied both to online processing and offline analysis. The source code was written using two computer languages (Fortran and Visual Basic). Descriptions of the source codes were included for non-expert AMSU-A data users.

[1] Visiting scientist from the Japan Meteorological Agency, Tokyo, Japan

1. Introduction

The Advanced Microwave Sounding Unit-A (AMSU-A) and AMSU-B built for the NOAA-K, L, M, N, and N' series of Polar-orbiting Operational Environmental Satellites (POES) are two new instruments, which are designed to be used together for the sounding of atmospheric temperature and humidity profiles. The first three sets of these instruments have been launched aboard the NOAA-K, -L, and –M, respectively, in May 1998, September 2000, and June 2002. Traditionally, a NOAA spacecraft is designated with an alphabetical letter before launch after which, a numerical series number replaces its letter nomenclature. Therefore, NOAA-K, -L, and –M became NOAA-15, –16, and -17, respectively, as they have been successfully launched.

AMSU-A is a 15-channel total-power microwave radiometer. Each AMSU-A instrument is composed of two separate units: AMSU-A2 with channels 1 and 2 at 23.8 and 31.4 GHz, and AMSU-A1 with twelve channels in the range of 50.3 to 57.3 GHz which are used for temperature sounding from the surface to about 50 km, (i.e., from ~1000 to ~1 mb) plus channel 15 at 89.0 GHz. Channel 3 at 50.3 GHz is also a window channel, as it senses atmospheric temperatures near the surface. A more complete description of the AMSU-A instrument and its radiometric performance was reported elsewhere [1]. Channels 1-3 and 15, collectively referred to as the window channels, aid the retrieval of temperature soundings by providing information to correct the effects due to surface emissivity, atmospheric liquid water, and total precipitable water vapor. The two low frequency channels also provide information on precipitation, sea ice, and snow cover.

AMSU-B carries five channels for humidity sounding. It has two channels at 89 and 150 GHz, respectively, and three channels around the 183 GHz water vapor line, details of which has been described elsewhere [2].

AMSU-A is a cross-track, step-scan instrument and executes one complete revolution every 8-second period. The AMSU-A antenna systems have a nominal field-of-view (FOV) of 3°20' at the half-power points (covering a 50-km diameter footprint at nadir) and execute a cross-track scan with 30 Earth FOVs (within ±48°20' from the nadir direction), one cold calibration FOV, and one warm blackbody FOV per 8-second scan period. Beam positions 1 and 30 are the outermost scan positions of the Earth views, while beam positions 15 and 16 (at ±1.67°from nadir) straddle the nadir.

The user community has reported that the FOV geolocation information in the NOAA-15 AMSU-A data have errors on the order of 1° or less, depending on channels. The geolocation information in the AMSU-A data sets consists of latitudes and longitudes for individual Earth FOVs, as generated by the POES navigation software package. The AMSU-A data have been used extensively in many applications [3-5] at NOAA and

worldwide agencies to generate products for weather forecasting, atmospheric temperature sounding, and hydrological studies. These geolocation errors can have significant adverse effect on the above products, if they are not corrected. For example, an error of 1° in geolocation will result in 15-km error on the Earth's surface at nadir. Such an error would have serious effect on the hydrological products and retrieval of surface products, particularly, near a coastline.

This study was undertaken to develop an algorithm for correcting the geolocation errors in the AMSU-A data. The method described in this report requires only the earth location data (i.e. latitude and longitude for each cell) of AMSU-A level 1B data. Spacecraft position and velocity are calculated internally on condition that the original earth location data of the AMSU-A level 1B data is generated with no attitude error correction (i.e. AMSU-A Data Set Header Record Word 895-896: Constant Roll Attitude Error, 897-898: Constant Pitch Attitude Error, and 899-890: Constant Yaw Attitude Error are zeroes).

2. Equations

This section describes equations for correction of the geolocation errors in the AMSU-A data sets.

2.1 Parameters used in Equations

Table 1 shows the constant parameters used in the equations. Table 2 shows major variables used for the equations.

Parameter	Description	Value
Re	Equatorial radius of the Earth	6378.135 km
f	Flattening of the Earth	1/298.25
Pc	Scan Position at nadir	15.5
Ti	Duration between adjacent IFOVs	202.5 ms
Ts	Scan Interval	8 s
θ_0	Scan Angle at position #0	$15.5 \times 3\frac{1}{3}$ deg
θr	Angle between adjacent IFOVs	$3\frac{1}{3}$ deg

Table 1. Constant Parameters.

Variable	Description	In/Out
P	Scan Position Number	In
θ	Scan Angle	internal
ΔR	Roll Correction Angle	In
ΔP	Pitch Correction Angle	In
ΔY	Yaw Correction Angle	In
φ	Latitude of Earth Location data	In
λ	Longitude of Earth Location data	In
φ'	Corrected Latitude	Out
λ'	Corrected Longitude	Out

Table 2. Major Variables.

2.2 Sub-satellite point

The latitude and longitude of sub-satellite point (SSP) are interpolated using nearest neighbor scan positions. Table 3 shows input parameters to calculate the sub-satellite point of each scan position. Figure 1 shows the geometry of the vectors as expressed in the equations.

$\varphi_{15,n}$	Latitude of scan n and position #15
$\lambda_{15,n}$	Longitude of scan n and position #15
$\varphi_{16,n}$	Latitude of scan n and position #16
$\lambda_{16,n}$	Longitude of scan n and position #16
$\varphi_{15,n+1}$	Latitude of scan n+1 and position #15
$\lambda_{15,n+1}$	Longitude of scan n+1 and position #15
$\varphi_{16,n+1}$	Latitude of scan n+1 and position #16
$\lambda_{16,n+1}$	Longitude of scan n+1 and position #16

Table 3. Input Parameters for SSP calculation.

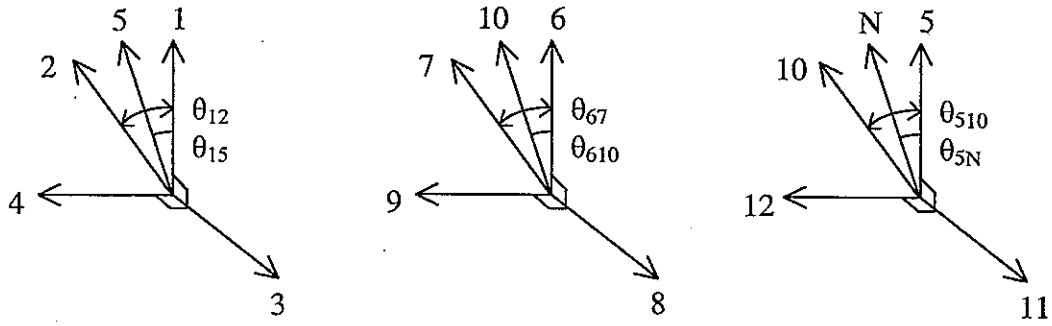


Figure 1. Vector geometry.
(Numbers and symbol show the subscripts of vectors.)

$$e^2 = 2f - f^2 \tag{1}$$

$$\begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix} = \frac{\mathrm{Re}}{\sqrt{1 - e^2 \sin^2 \varphi_{15,n}}} \begin{bmatrix} \cos\phi_{15,n} \cos\lambda_{15,n} \\ \cos\phi_{15,n} \sin\lambda_{15,n} \\ (1 - e^2) \sin\phi_{15,n} \end{bmatrix} \tag{2}$$

$$\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \frac{1}{\sqrt{X_1^2 + Y_1^2 + Z_1^2}} \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix} \tag{3}$$

$$\begin{pmatrix} X_2 \\ Y_2 \\ Z_2 \end{pmatrix} = \frac{\mathrm{Re}}{\sqrt{1 - e^2 \sin^2 \varphi_{16,n}}} \begin{bmatrix} \cos\phi_{16,n} \cos\lambda_{16,n} \\ \cos\phi_{16,n} \sin\lambda_{16,n} \\ (1 - e^2) \sin\phi_{16,n} \end{bmatrix} \tag{4}$$

$$\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} = \frac{1}{\sqrt{X_2^2 + Y_2^2 + Z_2^2}} \begin{pmatrix} X_2 \\ Y_2 \\ Z_2 \end{pmatrix} \tag{5}$$

$$\begin{pmatrix} X_3 \\ Y_3 \\ Z_3 \end{pmatrix} = \begin{pmatrix} y_1 z_2 - z_1 y_2 \\ z_1 x_2 - x_1 z_2 \\ x_1 y_2 - y_1 x_2 \end{pmatrix} \tag{6}$$

$$\begin{pmatrix} x_3 \\ y_3 \\ z_3 \end{pmatrix} = \frac{1}{\sqrt{X_3^{\,2} + Y_3^2 + Z_3^{\,2}}} \begin{pmatrix} X_3 \\ Y_3 \\ Z_3 \end{pmatrix} \tag{7}$$

$$\begin{pmatrix} x_4 \\ y_4 \\ z_4 \end{pmatrix} = \begin{pmatrix} y_3 z_1 - z_3 y_1 \\ z_3 x_1 - x_3 z_1 \\ x_3 y_1 - y_3 x_1 \end{pmatrix} \tag{8}$$

$$\theta_{12} = \cos^{-1}\left(x_1 x_2 + y_1 y_2 + z_1 z_2\right) \tag{9}$$

$$\theta_{15} = \frac{\theta_{12}}{2} \tag{10}$$

$$\begin{pmatrix} x_5 \\ y_5 \\ z_5 \end{pmatrix} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} \cos\theta_{15} + \begin{pmatrix} x_4 \\ y_4 \\ z_4 \end{pmatrix} \sin\theta_{15} \tag{11}$$

$$\begin{pmatrix} X_6 \\ Y_6 \\ Z_6 \end{pmatrix} = \frac{\mathrm{Re}}{\sqrt{1 - e^2 \sin^2 \varphi_{15,n+1}}} \begin{bmatrix} \cos\phi_{15,n+1} \cos\lambda_{15,n+1} \\ \cos\phi_{15,n+1} \sin\lambda_{15,n+1} \\ (1 - e^2) \sin\phi_{15,n+1} \end{bmatrix} \tag{12}$$

$$\begin{pmatrix} x_6 \\ y_6 \\ z_6 \end{pmatrix} = \frac{1}{\sqrt{X_6^{\,2} + Y_6^2 + Z_6^{\,2}}} \begin{pmatrix} X_6 \\ Y_6 \\ Z_6 \end{pmatrix} \tag{13}$$

$$\begin{pmatrix} X_7 \\ Y_7 \\ Z_7 \end{pmatrix} = \frac{\mathrm{Re}}{\sqrt{1 - e^2 \sin^2 \varphi_{16,n+1}}} \begin{bmatrix} \cos\phi_{16,n+1} \cos\lambda_{16,n+1} \\ \cos\phi_{16,n+1} \sin\lambda_{16,n+1} \\ (1 - e^2) \sin\phi_{16,n+1} \end{bmatrix} \tag{14}$$

$$\begin{pmatrix} x_7 \\ y_7 \\ z_7 \end{pmatrix} = \frac{1}{\sqrt{X_7^{\,2} + Y_7^2 + Z_7^{\,2}}} \begin{pmatrix} X_7 \\ Y_7 \\ Z_7 \end{pmatrix} \tag{15}$$

$$\begin{pmatrix} X_8 \\ Y_8 \\ Z_8 \end{pmatrix} = \begin{pmatrix} y_6 z_7 - z_6 y_7 \\ z_6 x_7 - x_6 z_7 \\ x_6 y_7 - y_6 x_7 \end{pmatrix} \tag{16}$$

$$\begin{pmatrix} x_8 \\ y_8 \\ z_8 \end{pmatrix} = \frac{1}{\sqrt{X_8^{\,2} + Y_8^2 + Z_8^{\,2}}} \begin{pmatrix} X_8 \\ Y_8 \\ Z_8 \end{pmatrix} \tag{17}$$

$$\begin{pmatrix} x_9 \\ y_9 \\ z_9 \end{pmatrix} = \begin{pmatrix} y_8 z_6 - z_8 y_6 \\ z_8 x_6 - x_8 z_6 \\ x_8 y_6 - y_8 x_6 \end{pmatrix} \tag{18}$$

$$\theta_{67} = \cos^{-1}\left(x_6 x_7 + y_6 y_7 + z_6 z_7\right) \tag{19}$$

$$\theta_{610} = \frac{\theta_{67}}{2} \tag{20}$$

$$\begin{pmatrix} x_{10} \\ y_{10} \\ z_{10} \end{pmatrix} = \begin{pmatrix} x_6 \\ y_6 \\ z_6 \end{pmatrix} \cos\theta_{610} + \begin{pmatrix} x_9 \\ y_9 \\ z_9 \end{pmatrix} \sin\theta_{610} \tag{21}$$

$$\begin{pmatrix} X_{11} \\ Y_{11} \\ Z_{11} \end{pmatrix} = \begin{pmatrix} y_5 z_{10} - z_5 y_{10} \\ z_5 x_{10} - x_5 z_{10} \\ x_5 y_{10} - y_5 x_{10} \end{pmatrix} \tag{22}$$

$$\begin{pmatrix} x_{11} \\ y_{11} \\ z_{11} \end{pmatrix} = \frac{1}{\sqrt{X_{11}^{\,2} + Y_{11}^{\,2} + Z_{11}^{\,2}}} \begin{pmatrix} X_{11} \\ Y_{11} \\ Z_{11} \end{pmatrix} \tag{23}$$

$$\begin{pmatrix} x_{12} \\ y_{12} \\ z_{12} \end{pmatrix} = \begin{pmatrix} y_{11} z_5 - z_{11} y_5 \\ z_{11} x_5 - x_{11} z_5 \\ x_{11} y_5 - y_{11} x_5 \end{pmatrix} \tag{24}$$

$$\theta_{510} = \cos^{-1}\left[x_5 x_{10} + y_5 y_{10} + z_5 z_{10}\right] \tag{25}$$

$$\theta_{5N} = (P - Pc)\frac{Ti}{Ts}\theta_{510} \tag{26}$$

2.3 Nadir vector (sub-satellite point vector)

$$\begin{pmatrix} x_N \\ y_N \\ z_N \end{pmatrix} = \begin{pmatrix} x_5 \\ y_5 \\ z_5 \end{pmatrix} \cos\theta_{5N} + \begin{pmatrix} x_{12} \\ y_{12} \\ z_{12} \end{pmatrix} \sin\theta_{5N} \tag{27}$$

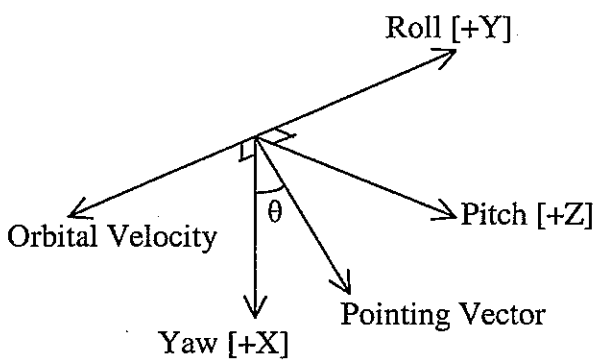


Figure 2 Attitude geometry

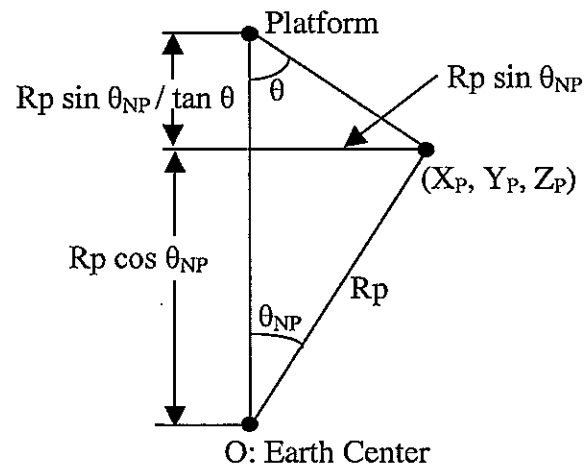


Figure 3 Satellite position geometry

2.4 Yaw vector

Figure 2 shows the definition of roll, pitch, and yaw vectors. The yaw vector of instrument platform is expressed as

$$\begin{pmatrix} x_{yaw} \\ y_{yaw} \\ z_{yaw} \end{pmatrix} = -\begin{pmatrix} x_N \\ y_N \\ z_N \end{pmatrix} \tag{28}$$

2.5 Scene vector

Unit vector (x_p, y_p, z_p) is directed from the earth center to the point of interest (X_p, Y_p, Z_p) that is the point of scan n, and position P, and is given by

$$\begin{pmatrix} X_P \\ Y_P \\ Z_P \end{pmatrix} = \frac{\mathrm{Re}}{\sqrt{1 - e^2 \sin^2 \varphi_{P,n}}} \begin{bmatrix} \cos\phi_{P,n} \cos\lambda_{P,n} \\ \cos\phi_{P,n} \sin\lambda_{P,n} \\ (1 - e^2) \sin\phi_{P,n} \end{bmatrix} \tag{29}$$

$$\begin{pmatrix} x_P \\ y_P \\ z_P \end{pmatrix} = \frac{1}{\sqrt{X_P^{\,2} + Y_P^{\,2} + Z_P^{\,2}}} \begin{pmatrix} X_P \\ Y_P \\ Z_P \end{pmatrix} \tag{30}$$

2.6 Roll vector

The roll vector of the instrument platform is expressed as

$$\theta = -\theta r \cdot P + \theta_0 \tag{31}$$

$$\begin{pmatrix} X_{roll} \\ Y_{roll} \\ Z_{roll} \end{pmatrix} = \begin{pmatrix} y_{yaw} z_P - z_{yaw} y_P \\ z_{yaw} x_P - x_{yaw} z_P \\ x_{yaw} y_P - y_{yaw} x_P \end{pmatrix} \cdot \frac{-\theta}{|\theta|} \tag{32}$$

$$\begin{pmatrix} x_{roll} \\ y_{roll} \\ z_{roll} \end{pmatrix} = \frac{1}{\sqrt{X_{roll}^{\ 2} + Y_{roll}^{\ 2} + Z_{roll}^{\ 2}}} \begin{pmatrix} X_{roll} \\ Y_{roll} \\ Z_{roll} \end{pmatrix} \tag{33}$$

2.7 Pitch vector

The pitch vector of the instrument platform is given by

$$\begin{pmatrix} x_{pitch} \\ y_{pitch} \\ z_{pitch} \end{pmatrix} = \begin{pmatrix} y_{yaw} z_{roll} - z_{yaw} y_{roll} \\ z_{yaw} x_{roll} - x_{yaw} z_{roll} \\ x_{yaw} y_{roll} - y_{yaw} x_{roll} \end{pmatrix} \tag{34}$$

2.8 Satellite position

Satellite position (Xo, Yo, Zo) is computed by the nadir vector, scene vector, and scan angle. Figure 3 shows the geometry of satellite position calculation.

$$\theta_{NP} = \cos^{-1}\left[x_N x_P + y_N y_P + z_N z_P\right] \tag{35}$$

$$\begin{pmatrix} Xo \\ Yo \\ Zo \end{pmatrix} = \left(\frac{\sin\theta_{NP}}{\tan|\theta|} + \cos\theta_{NP} \right) \sqrt{X_P^{\ 2} + Y_P^{\ 2} + Z_P^{\ 2}} \begin{pmatrix} x_N \\ y_N \\ z_N \end{pmatrix} \tag{36}$$

2.9 Correction matrix

Attitude correction matrix is defined as

$$L = \begin{bmatrix} \cos\Delta P & -\sin\Delta P & 0 \\ \sin\Delta P & \cos\Delta P & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\Delta R & 0 & \sin\Delta R \\ 0 & 1 & 0 \\ -\sin\Delta R & 0 & \cos\Delta R \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Delta Y & -\sin\Delta Y \\ 0 & \sin\Delta Y & \cos\Delta Y \end{bmatrix}$$

$$= \begin{bmatrix} \cos\Delta P\cos\Delta R & \cos\Delta P\sin\Delta R\sin\Delta Y - \sin\Delta P\cos\Delta Y & \cos\Delta P\sin\Delta R\cos\Delta Y + \sin\Delta P\sin\Delta Y \\ \sin\Delta P\cos\Delta R & \sin\Delta P\sin\Delta R\sin\Delta Y + \cos\Delta P\cos\Delta Y & \sin\Delta P\sin\Delta R\cos\Delta Y - \cos\Delta P\sin\Delta Y \\ -\sin\Delta R & \cos\Delta R\sin\Delta Y & \cos\Delta R\cos\Delta Y \end{bmatrix} \quad (37)$$

2.10 Corrected pointing vector

Corrected pointing vector in the platform coordinates is given by

$$\begin{pmatrix} x_{cpp} \\ y_{cpp} \\ z_{cpp} \end{pmatrix} = L \begin{pmatrix} \cos\theta \\ 0 \\ \sin\theta \end{pmatrix} \quad (38)$$

The corrected pointing vector in the earth fixed coordinates is expressed as

$$\begin{pmatrix} x_{cp} \\ y_{cp} \\ z_{cp} \end{pmatrix} = \begin{pmatrix} x_{yaw} \\ y_{yaw} \\ z_{yaw} \end{pmatrix} x_{cpp} + \begin{pmatrix} x_{roll} \\ y_{roll} \\ z_{roll} \end{pmatrix} y_{cpp} + \begin{pmatrix} x_{pitch} \\ y_{pitch} \\ z_{pitch} \end{pmatrix} z_{cpp} \quad (39)$$

2.11 Corrected view vector

The corrected view vector (X_p', Y_p', Z_p') directed from the platform to the point of interest is given by

$$a = (1-f)^2(x_{cp}^{\ 2} + y_{cp}^{\ 2}) + z_{cp}^{\ 2} \quad (40)$$

$$b = (1-f)^2(Xo \cdot x_{cp} + Yo \cdot y_{cp}) + Zo \cdot z_{cp} \quad (41)$$

$$c = (1-f)^2(Xo^2 + Yo^2 - \mathrm{Re}^2) + Zo^2 \quad (42)$$

$$k = \frac{-b \pm \sqrt{b^2 - ac}}{a}$$

(smaller absolute value should be employed) (43)

$$\begin{pmatrix} X_p' \\ Y_p' \\ Z_p' \end{pmatrix} = \begin{pmatrix} Xo \\ Yo \\ Zo \end{pmatrix} + k \begin{pmatrix} x_{cp} \\ y_{cp} \\ z_{cp} \end{pmatrix} \tag{44}$$

2.12 Corrected latitude and longitude

The corrected latitude and longitude of the point of interest are given by

$$\varphi' = \tan^{-1}\left(\frac{Z_p'}{(1-f)^2 \sqrt{(X_p')^2 + (Y_p')^2}} \right) \tag{45}$$

$$\lambda' = \tan^{-1}\left(\frac{Y_p'}{X_p'} \right) \tag{46}$$

3. Application

Once the instrumental attitude errors for each AMSU-A channel are measured, then accurate geolocation information for individual channels can be calculated by the equations as described above. There is no guarantee that the errors are constant and stable. If the instrumental attitude errors are defined as a variable in a software code such as a function of time, latitude, or longitude, it is able to correct the errors in case they are not stable.

The equations can be applied to AMSU-B, HIRS, and AVHRR level 1B data, if the constant parameters (Pc, Ti, Ts, θ_0, θr) and the input parameters shown in Table 3 are modified. For AVHRR, a modification may be required, because the angle of Equation (35) will be very small near nadir so that the error of satellite position will increase. In Appendix A, two sample programs in FORTRAN and Visual Basic are listed. The sample programs are,

- Earth_Location_Correction1.0.xls
- Subroutine ELC

4. Attitude Errors

Inter-sensor navigation (registration) is performed to determine the instrumental attitude error of AMSU-A using pattern matching technique. AMSU-B Channel 16 is used as reference and cross correlation method is used for the pattern matching between AMSU-A and AMSU-B images. Figure 4 shows a sample of the pattern matching.

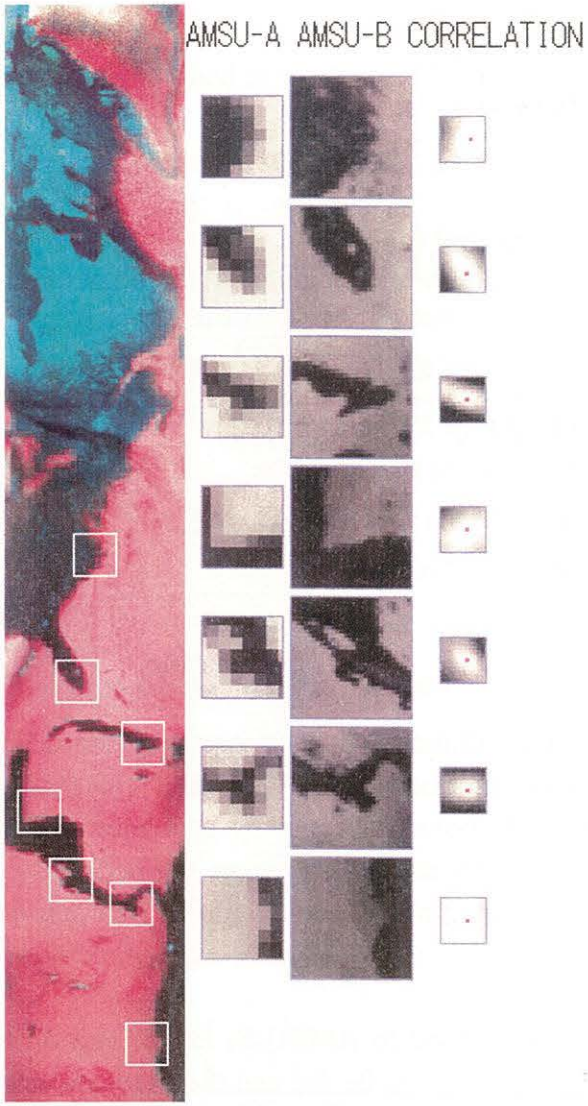


Figure 4. Sample of Pattern matching between AMSU-A and AMSU-B.

- First column: AMSU-A channel 1 superimposed on AMSU-B channel 16 of NOAA-15, 01/30/2001, 1242-1308.

- Second column: AMSU-B windows are bigger than AMSU-A windows to calculate cross correlation coefficients.

- Fourth column: A gray-scaled image of the cross correlation coefficients is shown while a dot shows their peak.

Figure 5 shows the results of the pattern matching of NOAA-15 AMUS-A and B. Line and pixel shift values are shown with the average of 20 measurements a day. Error range shows the mean square error of the average value. Figure 5 suggests that the line and pixel shifts are constant.

The AMSU-A system consists of three subsystems: AMSU-A1-1, AMSU-A1-2, and AMSU-A2. Each subsystem has one scanning antenna that may have small pointing (attitude or alignment) error. The pattern matching is performed on the window channel that shows the land features of a coastline to determine the shift values for each subsystem.

Table 4 shows the shift values calculated with the average of the pattern matching measurements shown in Figure 5.

Since the reference imagery of the pattern matching is AMSU-B channel 16,

1) if a line shift is positive, AMSU-A imagery has an offset to the direction of orbital velocity compared with AMSU-B.
2) if a pixel shift is positive, AMSU-A imagery has an offset to the scan direction compared with AMSU-B.

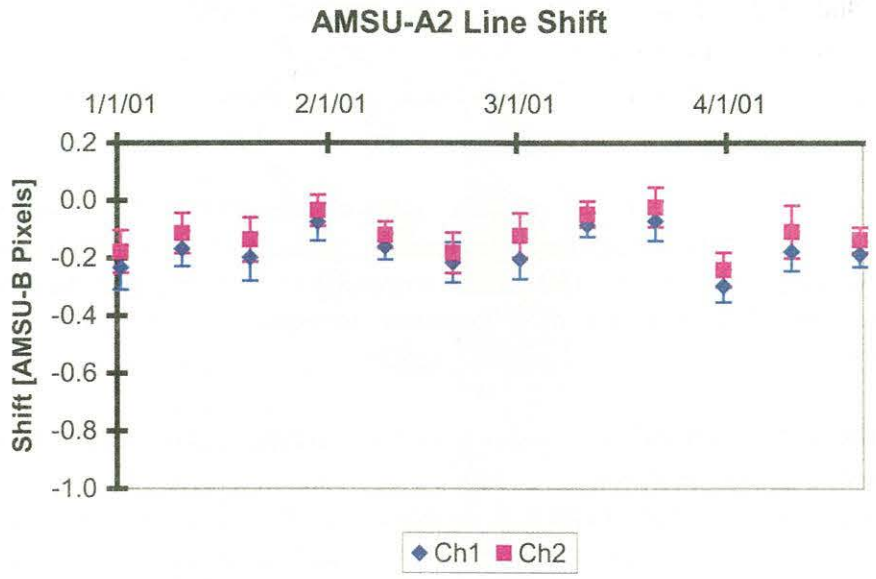
AMSU-A2 [Ch1, 2]		AMSU-A1-2 [Ch3]		AMSU-A1-1 [Ch15]		Note
Line [Along-track]	Pixel [Cross-track]	Line [Along-track]	Pixel [Cross-track]	Line [Along-track]	Pixel [Cross-track]	
-0.15	-0.92	-0.26	0.20	-0.29	0.20	04/23/2001

Table 4. AMSU Image Shift Values expressed by AMSU-B Lines and Pixels.

Assume that the instrumental attitude error of AMSU-B is negligible, then the line and pixel shifts can be directly converted to AMSU-A attitude errors and their correction angles. Table 5 shows the correction angles that are converted from the shifts shown in Table 4. It is assumed that the window channel represents the pointing (attitude) error of the subsystem, because it is difficult to get any accurate registration information of non-window channels using the pattern matching. These can be applied to calculate the corrected earth location data described in section 2. Pitch and roll correction angles should be interpreted as ΔP and ΔR in Table 2 respectively.

AMSU-A2 [Ch1, 2]		AMSU-A1-2 [Ch3-5,8]		AMSU-A1-1 [Ch6,7,9-15]		Note
Pitch	Roll	Pitch	Roll	Pitch	Roll	
-3.1	18	-5.5	-3.8	-6.1	-3.8	04/23/2001

Table 5. AMSU Correction Angles (milli-radians).



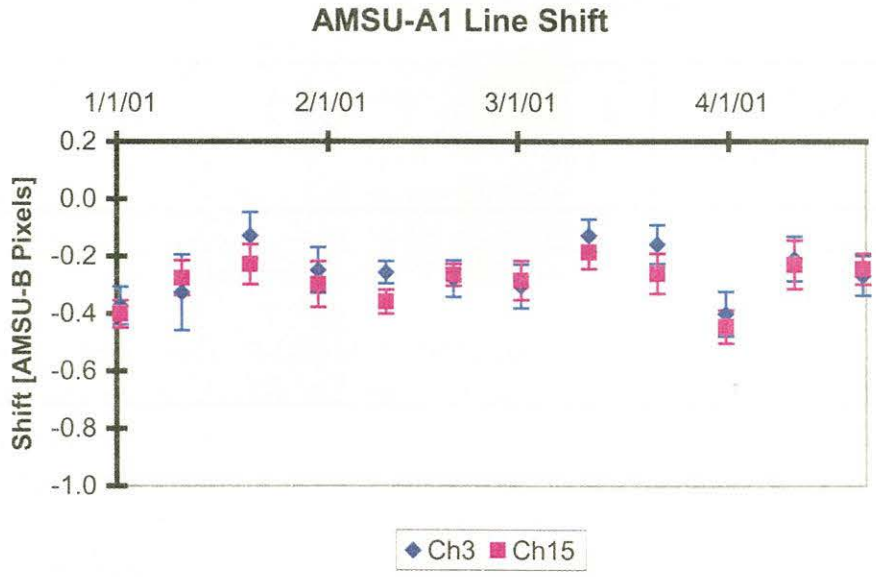
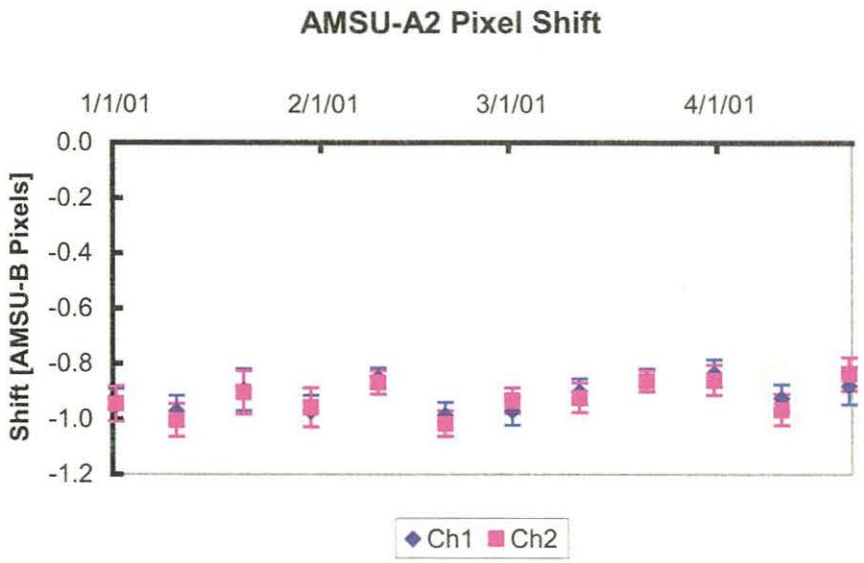


Figure 5a. NOAA-15 AMSU-A Line (along-track) Shift.



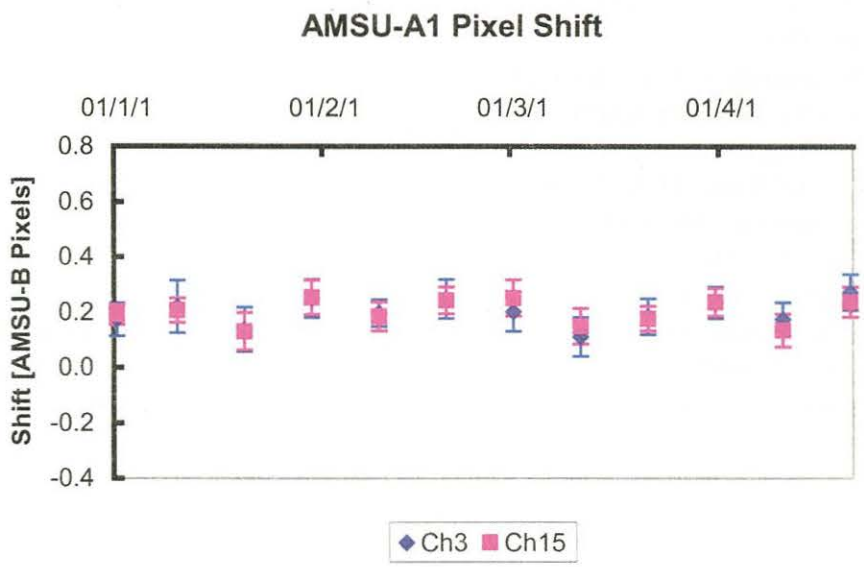


Figure 5b. NOAA-15 AMSU-A Pixel (cross-track) Shift.

5. Conclusion

An algorithm for correction of the navigation errors in the AMSU-A data was developed. These navigation errors, which are although not produced by the AMSU-A instruments, can have significant adverse effect on many products derived from AMSU-A data. The algorithm can correct the navigation errors and improve the accuracy of geolocation information of the AMSU-A data. Test results using the algorithm show that a roll angle error of the order of 1° exists in the NOAA-15 AMSU-A channels 1 and 2. A stand-alone software that can be used off-line to correct the navigation errors was developed and tested. The algorithm as shown in the source codes is being tested for operational use.

References

[1]. T. Mo, "Prelaunch Calibration of the Advanced Microwave Sounding Unit-A for NOAA-K," IEEE Trans. Microwave Theory and Techniques, Vol. 44, pp 1460-1469, 1996.

[2]. R. W. Saunders, T. J. Hewison, S. J. Stringer and N. C. Atkinson, "The Radiometric Characterisation of AMSU-B," IEEE Trans. Microwave Theory and Techniques, Vol. 43, pp. 760-771, 1995.

[3]. M. D. Goldberg, D. S. Crosby, and L. Chou, "The Limb Adjustment of AMSU-A Observations: Methodology and Validation," J. Applied Meteorology, Vol. 40, pp. 70-83, 2001.

[4]. C. Prigent, J. P. Wigneron, W. B. Rossow, and J. R. Pardo-Carrion, "Frequency and Angular Variations of Land Surface Microwave emissivities: Can We Estimate SSM/T and AMSU Emissivities from SSM/I Emissivities?," IEEE Trans. Geoscience and Remote Sensing, Vol. 38, pp. 2373-2386, 2000.

[5]. P. W. Rosenkranz, "Retrieval of Temperature and Moisture Profiles from AMSU-A and AMSU-B Measurements," IEEE Trans. Geoscience and Remote Sensing, Vol. 39, pp. 2429-2435, 2001.

APPENDIX A

Listing of Software

The computer programs that are used to produce the results presented in this report are listed below.

A-1. Earth_Location_Correction1.0.xls

		Attitude Correction [rad]			Pre-correction Data		Post-correction Data		correction		Nadir			
Scan #	Position #	Roll	Pitch	Yaw	Lat.	Lon.	Lat.	Lon.	Lat.	Lon.	Lat.	Lon.	Altitude	Scan Angle
1	1	1.75E-02	0.00E+00	0.00E+00	72.26	-137.13	72.64	-137.21	0.38	-0.08	81.30	-140.89	803.2	48.3
1	2	1.75E-02	0.00E+00	0.00E+00	73.44	-137.43	73.76	-137.51	0.32	-0.08	81.30	-140.97	803.2	45.0
1	3	1.75E-02	0.00E+00	0.00E+00	74.44	-137.73	74.70	-137.80	0.27	-0.07	81.30	-141.05	803.2	41.7
1	4	1.75E-02	0.00E+00	0.00E+00	75.29	-138.02	75.52	-138.09	0.23	-0.07	81.30	-141.13	803.2	38.3
1	5	1.75E-02	0.00E+00	0.00E+00	76.04	-138.31	76.25	-138.38	0.21	-0.07	81.30	-141.21	803.2	35.0

A-2. Visual Basic Macro

```
DefDbl U, X-Z
Sub Earth_Location_Correction()
'
'--- parameter definition ---
'    PI
Dim Pi As Double, Rad As Double
Pi = 3.14159265359
Rad = Pi / CDbl(180)
'    Earth shape parameters
Dim Re As Double, f As Double, ee As Double, RN As Double
Re = 6378.135
f = 1 / 298.25
ee = 2 * f - f * f
'    Instrument parameters
Dim Ti As Double, Ts As Double, Np As Double, Sr As Double, S0 As Double
Ti = 0.2025          'Duration between adjacent IFOVs (sec)
Ts = 8               'Time interval between adjacent scans (sec)
Pc = 15.5            'scan position of nadir
Sr = (3 + 1 / 3)     'scan angle interval between adjacent positions (degree)
S0 = Sr * Pc         'scan angle of position #0 (degree)
'    others
Dim ratio As Double
Dim angle12 As Double, angle67 As Double, angle510 As Double
Dim Latp As Double, Lonp As Double
Dim angleNp As Double, altitude As Double, scan As Double
Dim CM(3, 3) As Double
Dim dR As Double, dP As Double, dY As Double
Dim ka As Double, kb As Double, kc As Double
Dim k1 As Double, k2 As Double, k As Double
Dim Latpc As Double, Lonpc As Double
'
'--- input Earth location data ---
'    count total scans
For I = 1 To 999
    For J = 1 To 30
        If Sheets("AMSU-A").Cells((I - 1) * 30 + J + 2, 1) = "" Then GoTo count_end
    Next J
Next I
count_end:
Total_scan = I - 1
'
'--- correct earth location data ---
'
```

```
For I = 1 To Total_scan
'
    '--- data input offset ---
    doff = 0
    If I = Total_scan Then doff = -1
    '--- scan n ---
    Lat15_1 = Sheets("AMSU-A").Cells((I - 1 + doff) * 30 + 15 + 2, 6) * Rad
    Lon15_1 = Sheets("AMSU-A").Cells((I - 1 + doff) * 30 + 15 + 2, 7) * Rad
    Lat16_1 = Sheets("AMSU-A").Cells((I - 1 + doff) * 30 + 16 + 2, 6) * Rad
    Lon16_1 = Sheets("AMSU-A").Cells((I - 1 + doff) * 30 + 16 + 2, 7) * Rad
    '--- vector 1 ---
    RN = Re / Sqr(1 - ee * Sin(CDbl(Lat15_1)) ^ 2)
    X1 = RN * Cos(CDbl(Lat15_1)) * Cos(CDbl(Lon15_1))
    Y1 = RN * Cos(CDbl(Lat15_1)) * Sin(CDbl(Lon15_1))
    Z1 = RN * (1 - ee) * Sin(CDbl(Lat15_1))
    ux1 = X1 / Sqr(X1 * X1 + Y1 * Y1 + Z1 * Z1)
    uy1 = Y1 / Sqr(X1 * X1 + Y1 * Y1 + Z1 * Z1)
    uz1 = Z1 / Sqr(X1 * X1 + Y1 * Y1 + Z1 * Z1)
    '--- vector 2 ---
    RN = Re / Sqr(1 - ee * Sin(CDbl(Lat16_1)) ^ 2)
    X2 = RN * Cos(CDbl(Lat16_1)) * Cos(CDbl(Lon16_1))
    Y2 = RN * Cos(CDbl(Lat16_1)) * Sin(CDbl(Lon16_1))
    Z2 = RN * (1 - ee) * Sin(CDbl(Lat16_1))
    ux2 = X2 / Sqr(X2 * X2 + Y2 * Y2 + Z2 * Z2)
    uy2 = Y2 / Sqr(X2 * X2 + Y2 * Y2 + Z2 * Z2)
    uz2 = Z2 / Sqr(X2 * X2 + Y2 * Y2 + Z2 * Z2)
    '--- vector 3 ---
    X3 = uy1 * uz2 - uz1 * uy2
    Y3 = uz1 * ux2 - ux1 * uz2
    Z3 = ux1 * uy2 - uy1 * ux2
    ux3 = X3 / Sqr(X3 * X3 + Y3 * Y3 + Z3 * Z3)
    uy3 = Y3 / Sqr(X3 * X3 + Y3 * Y3 + Z3 * Z3)
    uz3 = Z3 / Sqr(X3 * X3 + Y3 * Y3 + Z3 * Z3)
    '--- vector 4 ---
    X4 = uy3 * uz1 - uz3 * uy1
    Y4 = uz3 * ux1 - ux3 * uz1
    Z4 = ux3 * uy1 - uy3 * ux1
    ux4 = X4 / Sqr(X4 * X4 + Y4 * Y4 + Z4 * Z4)
    uy4 = Y4 / Sqr(X4 * X4 + Y4 * Y4 + Z4 * Z4)
    uz4 = Z4 / Sqr(X4 * X4 + Y4 * Y4 + Z4 * Z4)
    '--- angle 1 to 2 ---
    Call Arcos(CDbl(ux1 * ux2 + uy1 * uy2 + uz1 * uz2), angle12)
    '--- vector 5 : nadir ---
    ux5 = ux1 * CDbl(Cos(angle12 * CDbl(0.5))) + ux4 * CDbl(Sin(angle12 * CDbl(0.5)))
    uy5 = uy1 * CDbl(Cos(angle12 * CDbl(0.5))) + uy4 * CDbl(Sin(angle12 * CDbl(0.5)))
    uz5 = uz1 * CDbl(Cos(angle12 * CDbl(0.5))) + uz4 * CDbl(Sin(angle12 * CDbl(0.5)))

     --- scan n+1 ---
    Lat15_2 = Sheets("AMSU-A").Cells((I - 0 + doff) * 30 + 15 + 2, 6) * Rad
    Lon15_2 = Sheets("AMSU-A").Cells((I - 0 + doff) * 30 + 15 + 2, 7) * Rad
    Lat16_2 = Sheets("AMSU-A").Cells((I - 0 + doff) * 30 + 16 + 2, 6) * Rad
    Lon16_2 = Sheets("AMSU-A").Cells((I - 0 + doff) * 30 + 16 + 2, 7) * Rad
    '--- vector 6 ---
    RN = Re / Sqr(1 - ee * Sin(CDbl(Lat15_2)) ^ 2)
    X6 = RN * Cos(CDbl(Lat15_2)) * Cos(CDbl(Lon15_2))
    Y6 = RN * Cos(CDbl(Lat15_2)) * Sin(CDbl(Lon15_2))
    Z6 = RN * (1 - ee) * Sin(CDbl(Lat15_2))
    ux6 = X6 / Sqr(X6 * X6 + Y6 * Y6 + Z6 * Z6)
    uy6 = Y6 / Sqr(X6 * X6 + Y6 * Y6 + Z6 * Z6)
    uz6 = Z6 / Sqr(X6 * X6 + Y6 * Y6 + Z6 * Z6)
    '--- vector 7 ---
    RN = Re / Sqr(1 - ee * Sin(CDbl(Lat16_2)) ^ 2)
    X7 = RN * Cos(CDbl(Lat16_2)) * Cos(CDbl(Lon16_2))
    Y7 = RN * Cos(CDbl(Lat16_2)) * Sin(CDbl(Lon16_2))
    Z7 = RN * (1 - ee) * Sin(CDbl(Lat16_2))
    ux7 = X7 / Sqr(X7 * X7 + Y7 * Y7 + Z7 * Z7)
    uy7 = Y7 / Sqr(X7 * X7 + Y7 * Y7 + Z7 * Z7)
    uz7 = Z7 / Sqr(X7 * X7 + Y7 * Y7 + Z7 * Z7)
    '--- vector 8 ---
    X8 = uy6 * uz7 - uz6 * uy7
    Y8 = uz6 * ux7 - ux6 * uz7
```

```
Z8 = ux6 * uy7 - uy6 * ux7
ux8 = X8 / Sqr(X8 * X8 + Y8 * Y8 + Z8 * Z8)
uy8 = Y8 / Sqr(X8 * X8 + Y8 * Y8 + Z8 * Z8)
uz8 = Z8 / Sqr(X8 * X8 + Y8 * Y8 + Z8 * Z8)
'--- vector 9 ---
X9 = uy8 * uz6 - uz8 * uy6
Y9 = uz8 * ux6 - ux8 * uz6
Z9 = ux8 * uy6 - uy8 * ux6
ux9 = X9 / Sqr(X9 * X9 + Y9 * Y9 + Z9 * Z9)
uy9 = Y9 / Sqr(X9 * X9 + Y9 * Y9 + Z9 * Z9)
uz9 = Z9 / Sqr(X9 * X9 + Y9 * Y9 + Z9 * Z9)
'--- angle 6 to 7 ---
Call Arcos(CDbl(ux6 * ux7 + uy6 * uy7 + uz6 * uz7), angle67)
'--- vector 10 : nadir ---
ux10 = ux6 * CDbl(Cos(angle67 * CDbl(0.5))) + ux9 * CDbl(Sin(angle67 * CDbl(0.5)))
uy10 = uy6 * CDbl(Cos(angle67 * CDbl(0.5))) + uy9 * CDbl(Sin(angle67 * CDbl(0.5)))
uz10 = uz6 * CDbl(Cos(angle67 * CDbl(0.5))) + uz9 * CDbl(Sin(angle67 * CDbl(0.5)))

 --- vector 11 ---
X11 = uy5 * uz10 - uz5 * uy10
Y11 = uz5 * ux10 - ux5 * uz10
Z11 = ux5 * uy10 - uy5 * ux10
ux11 = X11 / Sqr(X11 * X11 + Y11 * Y11 + Z11 * Z11)
uy11 = Y11 / Sqr(X11 * X11 + Y11 * Y11 + Z11 * Z11)
uz11 = Z11 / Sqr(X11 * X11 + Y11 * Y11 + Z11 * Z11)
'--- vector 12 ---
X12 = uy11 * uz5 - uz11 * uy5
Y12 = uz11 * ux5 - ux11 * uz5
Z12 = ux11 * uy5 - uy11 * ux5
ux12 = X12 / Sqr(X12 * X12 + Y12 * Y12 + Z12 * Z12)
uy12 = Y12 / Sqr(X12 * X12 + Y12 * Y12 + Z12 * Z12)
uz12 = Z12 / Sqr(X12 * X12 + Y12 * Y12 + Z12 * Z12)
'--- angle 5 to 10 ---
Call Arcos(CDbl(ux5 * ux10 + uy5 * uy10 + uz5 * uz10), angle510)
'--- correction for each position ---
For P = 1 To 30
    '--- nadir ---
    ratio = CDbl(P - Pc) * Ti / Ts + CDbl(-doff)
    uxN = ux5 * CDbl(Cos(angle510 * ratio)) + ux12 * CDbl(Sin(angle510 * ratio))
    uyN = uy5 * CDbl(Cos(angle510 * ratio)) + uy12 * CDbl(Sin(angle510 * ratio))
    uzN = uz5 * CDbl(Cos(angle510 * ratio)) + uz12 * CDbl(Sin(angle510 * ratio))
    '--- nadir lat&lon ---
    If uxN <> 0 And uyN <> 0 Then
        Latn = Atn(uzN / Sqr(uxN * uxN + uyN * uyN))
    Else
        Latn = 0
    End If
    If uxN <> 0 Then
        Lonn = Atn(uyN / uxN)
        If uxN < 0 And uyN >= 0 Then Lonn = Lonn + Pi
        If uxN < 0 And uyN < 0 Then Lonn = Lonn - Pi
    Else
        If uyN > 0 Then
            Lonn = Pi / 2
        Else
            Lonn = -Pi / 2
        End If
    End If
    Sheets("AMSU-A").Cells((I - 1) * 30 + 2 + P, 12) = Latn / Rad
    Sheets("AMSU-A").Cells((I - 1) * 30 + 2 + P, 13) = Lonn / Rad

    '--- yaw (+Xp) vector ---
    uxyaw = -uxN
    uyyaw = -uyN
    uzyaw = -uzN
    '--- scene vector ---
    scan = (-Sr * CDbl(P) + S0) * Rad
    Latp = Sheets("AMSU-A").Cells((I - 1) * 30 + 2 + P, 6) * Rad
    Lonp = Sheets("AMSU-A").Cells((I - 1) * 30 + 2 + P, 7) * Rad
    RN = Re / Sqr(1 - ee * Sin(CDbl(Latp)) ^ 2)
    Xp = RN * Cos(CDbl(Latp)) * Cos(CDbl(Lonp))
```

```
Yp = RN * Cos(CDbl(Latp)) * Sin(CDbl(Lonp))
Zp = RN * (1 - ee) * Sin(CDbl(Latp))
uxp = Xp / Sqr(Xp * Xp + Yp * Yp + Zp * Zp)
uyp = Yp / Sqr(Xp * Xp + Yp * Yp + Zp * Zp)
uzp = Zp / Sqr(Xp * Xp + Yp * Yp + Zp * Zp)
'--- roll (+Yp) vector ---
Xroll = (uyyaw * uzp - uzyaw * uyp) * -Sgn(scan)
Yroll = (uzyaw * uxp - uxyaw * uzp) * -Sgn(scan)
Zroll = (uxyaw * uyp - uyyaw * uxp) * -Sgn(scan)
uxroll = Xroll / Sqr(Xroll * Xroll + Yroll * Yroll + Zroll * Zroll)
uyroll = Yroll / Sqr(Xroll * Xroll + Yroll * Yroll + Zroll * Zroll)
uzroll = Zroll / Sqr(Xroll * Xroll + Yroll * Yroll + Zroll * Zroll)
'--- pitch (+Zp) vector ---
uxpitch = uyyaw * uzroll - uzyaw * uyroll
uypitch = uzyaw * uxroll - uxyaw * uzroll
uzpitch = uxyaw * uyroll - uyyaw * uxroll
'--- satellite point ---
Call Arcos((uxN * uxp + uyN * uyp + uzN * uzp), angleNp)
Xo = CDbl(Sin(angleNp)/Tan(Abs(scan))+Cos(angleNp))*Sqr(Xp*Xp+Yp*Yp+Zp*Zp)*uxN
Yo = CDbl(Sin(angleNp)/Tan(Abs(scan))+Cos(angleNp))*Sqr(Xp*Xp+Yp*Yp+Zp*Zp)*uyN
Zo = CDbl(Sin(angleNp)/Tan(Abs(scan))+Cos(angleNp))*Sqr(Xp*Xp+Yp*Yp+Zp*Zp)*uzN
altitude = Sqr(Xo * Xo + Yo * Yo + Zo * Zo) - Re
Sheets("AMSU-A").Cells((I - 1) * 30 + 2 + P, 14) = altitude
Sheets("AMSU-A").Cells((I - 1) * 30 + 2 + P, 15) = scan / Rad
'--- corrected pointing vector ---
dRoll = Sheets("AMSU-A").Cells((I - 1) * 30 + 2 + P, 3)
dPitch = Sheets("AMSU-A").Cells((I - 1) * 30 + 2 + P, 4)
dYaw = Sheets("AMSU-A").Cells((I - 1) * 30 + 2 + P, 5)
dR = CDbl(dRoll)
dP = CDbl(dPitch)
dY = CDbl(dYaw)
CM(1, 1) = CDbl(Cos(dP) * Cos(dR))
CM(1, 2) = CDbl(Cos(dP) * Sin(dR) * Sin(dY) - Sin(dP) * Cos(dY))
CM(1, 3) = CDbl(Cos(dP) * Sin(dR) * Cos(dY) + Sin(dP) * Sin(dY))
CM(2, 1) = CDbl(Sin(dP) * Cos(dR))
CM(2, 2) = CDbl(Sin(dP) * Sin(dR) * Sin(dY) + Cos(dP) * Cos(dY))
CM(2, 3) = CDbl(Sin(dP) * Sin(dR) * Cos(dY) - Cos(dP) * Sin(dY))
CM(3, 1) = CDbl(-Sin(dR))
CM(3, 2) = CDbl(Cos(dR) * Sin(dY))
CM(3, 3) = CDbl(Cos(dR) * Cos(dY))
uxcpp = CM(1, 1) * CDbl(Cos(scan)) + CM(1, 3) * CDbl(Sin(scan))
uycpp = CM(2, 1) * CDbl(Cos(scan)) + CM(2, 3) * CDbl(Sin(scan))
uzcpp = CM(3, 1) * CDbl(Cos(scan)) + CM(3, 3) * CDbl(Sin(scan))
uxcp = uxyaw * uxcpp + uxroll * uycpp + uxpitch * uzcpp
uycp = uyyaw * uxcpp + uyroll * uycpp + uypitch * uzcpp
uzcp = uzyaw * uxcpp + uzroll * uycpp + uzpitch * uzcpp
'--- corrected view vector ---
ka = (1 - f) * (1 - f) * (uxcp * uxcp + uycp * uycp) + uzcp * uzcp
kb = (1 - f) * (1 - f) * (Xo * uxcp + Yo * uycp) + Zo * uzcp
kc = (1 - f) * (1 - f) * (Xo * Xo + Yo * Yo - Re * Re) + Zo * Zo
k1 = (-kb + Sqr(kb * kb - ka * kc)) / ka
k2 = (-kb - Sqr(kb * kb - ka * kc)) / ka
If Abs(k1) > Abs(k2) Then
    k = k2
Else
    k = k1
End If
'--- corrected scene point ---
Xpc = Xo + k * uxcp
Ypc = Yo + k * uycp
Zpc = Zo + k * uzcp
Latpc = Atn(Zpc / ((1 - f) * (1 - f) * Sqr(Xpc * Xpc + Ypc * Ypc)))
If Xpc <> 0 Then
    Lonpc = Atn(Ypc / Xpc)
    If Xpc < 0 And Ypc >= 0 Then Lonpc = Lonpc + Pi
    If Xpc < 0 And Ypc < 0 Then Lonpc = Lonpc - Pi
Else
    If Ypc > 0 Then
        Lonpc = Pi / 2
    Else
        Lonpc = -Pi / 2
```

```
            End If
        End If
        Sheets("AMSU-A").Cells((I - 1) * 30 + 2 + P, 8) = Latpc / Rad
        Sheets("AMSU-A").Cells((I - 1) * 30 + 2 + P, 9) = Lonpc / Rad
    Next P
'
Next I
'
'
End Sub
Sub Arcos(cosine As Double, angle)
If cosine <> 0 Then
    angle = CDbl(Atn(Sqr(1 / cosine ^ 2 - 1)))
Else
    angle = CDbl(0)
End If
End Sub
```

A-3. FORTRAN Subroutine

```
      Subroutine arguments
      LAT0 : IN [degrees], latitude of scan n
      LON0 : IN [degrees], longitude of scan n
      LAT1 : IN [degrees], latitude of scan n+1
      LON1 : IN [degrees], longitude of scan n+1
      ROLL : IN [radians], Roll Correction Angle
      PITCH: IN [radians], Pitch Correction Angle
      YAW  : IN [radians], Yaw Correction Angle
      OFFSET:IN
       (=0: LATC and LONC are corrected data of LAT0, LON0)
       (=1: LATC and LONC are corrected data of LAT1, LON1)
      LATC : OUT[radians], corrected latitude
      LONC : OUT[radians], corrected longitude

      SUBROUTINE ELC(LAT0,LON0,LAT1,LON1,ROLL,PITCH,YAW,OFFSET,
                                  LATC,LONC)
C
C--- parameter definition ---
C      IN
      REAL*8 LAT0(30),LON0(30),LAT1(30),LON1(30)
      REAL*8 ROLL(30),PITCH(30),YAW(30)
      INTEGER OFFSET
C      OUT
      REAL*8 LATC(30),LONC(30)
C      INTERNAL
      REAL*8 PI,RAD
      REAL*8 RE,F,EE,RN
      REAL*8 TI, TS, NP, SR, S0
      REAL*8 X1,Y1,Z1,UX1,UY1,UZ1,X2,Y2,Z2,UX2,UY2,UZ2
      REAL*8 X3,Y3,Z3,UX3,UY3,UZ3
      REAL*8 X4,Y4,Z4,UX4,UY4,UZ4
      REAL*8 UX5,UY5,UZ5,X6,Y6,Z6,UX6,UY6,UZ6,RATIO
      REAL*8 X7,Y7,Z7,UX7,UY7,UZ7,X8,Y8,Z8,UX8,UY8,UZ8
      REAL*8 X9,Y9,Z9,UX9,UY9,UZ9
      REAL*8 UX10,UY10,UZ10,X11,Y11,Z11,UX11,UY11,UZ11
      REAL*8 X12,Y12,Z12,UX12,UY12,UZ12
      REAL*8 UXN,UYN,UZN
      REAL*8 ANGLE12,ANGLE67,ANGLE510
      REAL*8 UXYAW,UYYAW,UZYAW
      REAL*8 LATP,LONP,XP,YP,ZP,UXP,UYP,UZP
      REAL*8 XROLL,YROLL,ZROLL,UXROLL,UYROLL,UZROLL
      REAL*8 UXPITCH,UYPITCH,UZPITCH
      REAL*8 ANGLENP,XO,YO,ZO,altitude,SCAN
      REAL*8 CM(3,3)
      REAL*8 DR,DP,DY
      REAL*8 UXCPP,UYCPP,UZCPP,UXCP,UYCP,UZCP
      REAL*8 KA,KB,KC,K1,K2,K,XPC,YPC,ZPC
      INTEGER P
```

```
C     PI
      PI = 3.14159265359D0
      RAD = PI/180.D0
C     Earth shape parameters
      RE = 6378.135D0
      F = 1.D0 / 298.25D0
      EE = 2.D0 * F - F * F
C     Instrument parameters
C      DuRATIOn between adjacent IFOVs (sec)
      TI = 0.2025D0
C      Time interval between adjacent SCANs (sec)
      TS = 8D0
C      SCAN position of nadir
      PC = 15.5D0
C      SCAN angle interval between adjacent positions (degree)
      SR = (3.D0 + 1.D0 / 3.D0)
C      SCAN angle of position #0 (degree)
      S0 = SR * PC
C
C--- correct earth location data ---
C
C     --- scan n ---
C     --- vector 1 ---
      RN = RE / DSQRT(1.D0-EE*DSIN(LAT0(15)*RAD)*DSIN(LAT0(15)*RAD))
      X1 = RN * DCOS(LAT0(15)*RAD) * DCOS((360.D0+LON0(15))*RAD)
      Y1 = RN * DCOS(LAT0(15)*RAD) * DSIN((360.D0+LON0(15))*RAD)
      Z1 = RN * (1.D0-EE) * DSIN(LAT0(15)*RAD)
      UX1 = X1 / DSQRT(X1 * X1 + Y1 * Y1 + Z1 * Z1)
      UY1 = Y1 / DSQRT(X1 * X1 + Y1 * Y1 + Z1 * Z1)
      UZ1 = Z1 / DSQRT(X1 * X1 + Y1 * Y1 + Z1 * Z1)
C     --- vector 2 ---
      RN = RE / DSQRT(1.D0-EE*DSIN(LAT0(16)*RAD)*DSIN(LAT0(16)*RAD))
      X2 = RN * DCOS(LAT0(16)*RAD) * DCOS(LON0(16)*RAD)
      Y2 = RN * DCOS(LAT0(16)*RAD) * DSIN(LON0(16)*RAD)
      Z2 = RN * (1.D0-EE) * DSIN(LAT0(16)*RAD)
      UX2 = X2 / DSQRT(X2 * X2 + Y2 * Y2 + Z2 * Z2)
      UY2 = Y2 / DSQRT(X2 * X2 + Y2 * Y2 + Z2 * Z2)
      UZ2 = Z2 / DSQRT(X2 * X2 + Y2 * Y2 + Z2 * Z2)
C     --- vector 3 ---
      X3 = UY1 * UZ2 - UZ1 * UY2
      Y3 = UZ1 * UX2 - UX1 * UZ2
      Z3 = UX1 * UY2 - UY1 * UX2
      UX3 = X3 / DSQRT(X3 * X3 + Y3 * Y3 + Z3 * Z3)
      UY3 = Y3 / DSQRT(X3 * X3 + Y3 * Y3 + Z3 * Z3)
      UZ3 = Z3 / DSQRT(X3 * X3 + Y3 * Y3 + Z3 * Z3)
C     --- vector 4 ---
      X4 = UY3 * UZ1 - UZ3 * UY1
      Y4 = UZ3 * UX1 - UX3 * UZ1
      Z4 = UX3 * UY1 - UY3 * UX1
      UX4 = X4 / DSQRT(X4 * X4 + Y4 * Y4 + Z4 * Z4)
      UY4 = Y4 / DSQRT(X4 * X4 + Y4 * Y4 + Z4 * Z4)
      UZ4 = Z4 / DSQRT(X4 * X4 + Y4 * Y4 + Z4 * Z4)
C     --- angle 1 to 2 ---
      ANGLE12 = DACOS(UX1 * UX2 + UY1 * UY2 + UZ1 * UZ2)
C     --- vector 5 : nadir ---
      UX5 = UX1 * DCOS(ANGLE12 * 0.5D0) + UX4 * DSIN(ANGLE12 * 0.5D0)
      UY5 = UY1 * DCOS(ANGLE12 * 0.5D0) + UY4 * DSIN(ANGLE12 * 0.5D0)
      UZ5 = UZ1 * DCOS(ANGLE12 * 0.5D0) + UZ4 * DSIN(ANGLE12 * 0.5D0)
C
C     --- scan n+1 ---
C     --- vector 6 ---
      RN = RE / DSQRT(1.D0-EE*DSIN(LAT1(15)*RAD)*DSIN(LAT1(15)*RAD))
      X6 = RN * DCOS(LAT1(15)*RAD) * DCOS(LON1(15)*RAD)
      Y6 = RN * DCOS(LAT1(15)*RAD) * DSIN(LON1(15)*RAD)
      Z6 = RN * (1.D0-EE) * DSIN(LAT1(15)*RAD)
      UX6 = X6 / DSQRT(X6 * X6 + Y6 * Y6 + Z6 * Z6)
      UY6 = Y6 / DSQRT(X6 * X6 + Y6 * Y6 + Z6 * Z6)
      UZ6 = Z6 / DSQRT(X6 * X6 + Y6 * Y6 + Z6 * Z6)
C     --- vector 7 ---
      RN = RE / DSQRT(1-EE*DSIN(LAT1(16))*DSIN(LAT1(16)))
      X7 = RN * DCOS(LAT1(16)*RAD) * DCOS(LON1(16)*RAD)
```

```
      Y7 = RN * DCOS(LAT1(16)*RAD) * DSIN(LON1(16)*RAD)
      Z7 = RN * (1.D0-EE) * DSIN(LAT1(16)*RAD)
      UX7 = X7 / DSQRT(X7 * X7 + Y7 * Y7 + Z7 * Z7)
      UY7 = Y7 / DSQRT(X7 * X7 + Y7 * Y7 + Z7 * Z7)
      UZ7 = Z7 / DSQRT(X7 * X7 + Y7 * Y7 + Z7 * Z7)
C     --- vector 8 ---
      X8 = UY6 * UZ7 - UZ6 * UY7
      Y8 = UZ6 * UX7 - UX6 * UZ7
      Z8 = UX6 * UY7 - UY6 * UX7
      UX8 = X8 / DSQRT(X8 * X8 + Y8 * Y8 + Z8 * Z8)
      UY8 = Y8 / DSQRT(X8 * X8 + Y8 * Y8 + Z8 * Z8)
      UZ8 = Z8 / DSQRT(X8 * X8 + Y8 * Y8 + Z8 * Z8)
C     --- vector 9 ---
      X9 = UY8 * UZ6 - UZ8 * UY6
      Y9 = UZ8 * UX6 - UX8 * UZ6
      Z9 = UX8 * UY6 - UY8 * UX6
      UX9 = X9 / DSQRT(X9 * X9 + Y9 * Y9 + Z9 * Z9)
      UY9 = Y9 / DSQRT(X9 * X9 + Y9 * Y9 + Z9 * Z9)
      UZ9 = Z9 / DSQRT(X9 * X9 + Y9 * Y9 + Z9 * Z9)
C     --- angle 6 to 7 ---
      ANGLE67 = DACOS(UX6 * UX7 + UY6 * UY7 + UZ6 * UZ7)
C     --- vector 10 : nadir ---
      UX10 = UX6 * DCOS(ANGLE67 * 0.5D0) + UX9 * DSIN(ANGLE67 * 0.5D0)
      UY10 = UY6 * DCOS(ANGLE67 * 0.5D0) + UY9 * DSIN(ANGLE67 * 0.5D0)
      UZ10 = UZ6 * DCOS(ANGLE67 * 0.5D0) + UZ9 * DSIN(ANGLE67 * 0.5D0)
C
C     --- vector 11 ---
      X11 = UY5 * UZ10 - UZ5 * UY10
      Y11 = UZ5 * UX10 - UX5 * UZ10
      Z11 = UX5 * UY10 - UY5 * UX10
      UX11 = X11 / DSQRT(X11 * X11 + Y11 * Y11 + Z11 * Z11)
      UY11 = Y11 / DSQRT(X11 * X11 + Y11 * Y11 + Z11 * Z11)
      UZ11 = Z11 / DSQRT(X11 * X11 + Y11 * Y11 + Z11 * Z11)
C     --- vector 12 ---
      X12 = UY11 * UZ5 - UZ11 * UY5
      Y12 = UZ11 * UX5 - UX11 * UZ5
      Z12 = UX11 * UY5 - UY11 * UX5
      UX12 = X12 / DSQRT(X12 * X12 + Y12 * Y12 + Z12 * Z12)
      UY12 = Y12 / DSQRT(X12 * X12 + Y12 * Y12 + Z12 * Z12)
      UZ12 = Z12 / DSQRT(X12 * X12 + Y12 * Y12 + Z12 * Z12)
C     --- angle 5 to 10 ---
      ANGLE510 = DACOS(UX5 * UX10 + UY5 * UY10 + UZ5 * UZ10)
C     --- correction for each position ---
      DO 1000 P = 1, 30
C        --- nadir ---
        RATIO = (DBLE(P) - PC) * TI / TS + DBLE(OFFSET)
        UXN = UX5*DCOS(ANGLE510 * RATIO) + UX12*DSIN(ANGLE510 * RATIO)
        UYN = UY5*DCOS(ANGLE510 * RATIO) + UY12*DSIN(ANGLE510 * RATIO)
        UZN = UZ5*DCOS(ANGLE510 * RATIO) + UZ12*DSIN(ANGLE510 * RATIO)
C         --- yaw (+XP) vector ---
        UXYAW = -UXN
        UYYAW = -UYN
        UZYAW = -UZN
C         --- scene vector ---
        SCAN = (-SR * DBLE(P) + S0) * RAD
        IF(OFFSET.EQ.0) THEN
          RN = RE / DSQRT(1.D0-EE*DSIN(LAT0(P)*RAD)*DSIN(LAT0(P)*RAD))
          XP = RN * DCOS(LAT0(P)*RAD) * DCOS(LON0(P)*RAD)
          YP = RN * DCOS(LAT0(P)*RAD) * DSIN(LON0(P)*RAD)
          ZP = RN * (1.D0-EE) * DSIN(LAT0(P)*RAD)
        ELSEIF(OFFSET.EQ.1) THEN
          RN = RE / DSQRT(1.D0-EE*DSIN(LAT1(P)*RAD)*DSIN(LAT1(P)*RAD))
          XP = RN * DCOS(LAT1(P)*RAD) * DCOS(LON1(P)*RAD)
          YP = RN * DCOS(LAT1(P)*RAD) * DSIN(LON1(P)*RAD)
          ZP = RN * (1.D0-EE) * DSIN(LAT1(P)*RAD)
        ENDIF
        UXP = XP / DSQRT(XP * XP + YP * YP + ZP * ZP)
        UYP = YP / DSQRT(XP * XP + YP * YP + ZP * ZP)
        UZP = ZP / DSQRT(XP * XP + YP * YP + ZP * ZP)
C         --- roll (+YP) vector ---
        XROLL = (UYYAW * UZP - UZYAW * UYP) * -DSIGN(1.D0,SCAN)
```

```
      YROLL = (UZYAW * UXP - UXYAW * UZP) * -DSIGN(1.D0,SCAN)
      ZROLL = (UXYAW * UYP - UYYAW * UXP) * -DSIGN(1.D0,SCAN)
      UXROLL = XROLL/DSQRT(XROLL*XROLL+YROLL*YROLL+ZROLL*ZROLL)
      UYROLL = YROLL/DSQRT(XROLL*XROLL+YROLL*YROLL+ZROLL*ZROLL)
      UZROLL = ZROLL/DSQRT(XROLL*XROLL+YROLL*YROLL+ZROLL*ZROLL)
C      --- pitch (+ZP) vector ---
      UXPITCH = UYYAW * UZROLL - UZYAW * UYROLL
      UYPITCH = UZYAW * UXROLL - UXYAW * UZROLL
      UZPITCH = UXYAW * UYROLL - UYYAW * UXROLL
C      --- satellite point ---
      ANGLENP = DACOS(UXN * UXP + UYN * UYP + UZN * UZP)
      XO =(DSIN(ANGLENP)/DTAN(DABS(SCAN))+DCOS(ANGLENP))*
     .          DSQRT(XP*XP+YP*YP+ZP*ZP)*UXN
      YO =(DSIN(ANGLENP)/DTAN(DABS(SCAN))+DCOS(ANGLENP))*
     .          DSQRT(XP*XP+YP*YP+ZP*ZP)*UYN
      ZO =(DSIN(ANGLENP)/DTAN(DABS(SCAN))+DCOS(ANGLENP))*
     .       DSQRT(XP*XP+YP*YP+ZP*ZP)*UZN
      ALTITUDE = DSQRT(XO * XO + YO * YO + ZO * ZO) - RE
C      --- corrected pointing vector ---
      DR = DBLE(ROLL(P))
      DP = DBLE(PITCH(P))
      DY = DBLE(YAW(P))

      CM(1, 1) = DCOS(DP)*DCOS(DR)
      CM(1, 2) = DCOS(DP)*DSIN(DR)*DSIN(DY)-DSIN(DP)*DCOS(DY)
      CM(1, 3) = DCOS(DP)*DSIN(DR)*DCOS(DY)+DSIN(DP)*DSIN(DY)
      CM(2, 1) = DSIN(DP)*DCOS(DR)
      CM(2, 2) = DSIN(DP)*DSIN(DR)*DSIN(DY)+DCOS(DP)*DCOS(DY)
      CM(2, 3) = DSIN(DP)*DSIN(DR)*DCOS(DY)-DCOS(DP)*DSIN(DY)
      CM(3, 1) = -DSIN(DR)
      CM(3, 2) = DCOS(DR) * DSIN(DY)
      CM(3, 3) = DCOS(DR) * DCOS(DY)
      UXCPP = CM(1, 1) * DCOS(SCAN) + CM(1, 3) * DSIN(SCAN)
      UYCPP = CM(2, 1) * DCOS(SCAN) + CM(2, 3) * DSIN(SCAN)
      UZCPP = CM(3, 1) * DCOS(SCAN) + CM(3, 3) * DSIN(SCAN)
      UXCP = UXYAW * UXCPP + UXROLL * UYCPP + UXPITCH * UZCPP
      UYCP = UYYAW * UXCPP + UYROLL * UYCPP + UYPITCH * UZCPP
      UZCP = UZYAW * UXCPP + UZROLL * UYCPP + UZPITCH * UZCPP
C      --- corrected view vector ---
      KA = (1-F)*(1-F)*(UXCP*UXCP+UYCP*UYCP)+UZCP*UZCP
      KB = (1-F)*(1-F)*(XO*UXCP+YO*UYCP)+ZO*UZCP
      KC = (1-F)*(1-F)*(XO*XO+YO*YO-RE*RE)+ZO*ZO
      K1 = (-KB+DSQRT(KB*KB-KA*KC))/KA
      K2 = (-KB-DSQRT(KB*KB-KA*KC))/KA
      If(DABS(K1).GT.DABS(K2)) Then
          K = K2
      Else
          K = K1
      End If
C      --- corrected scene point ---
      XPC = XO + K * UXCP
      YPC = YO + K * UYCP
      ZPC = ZO + K * UZCP
      LATC(P) = DATAN(ZPC/((1-F)*(1-F)*DSQRT(XPC*XPC+YPC*YPC)))
      If(XPC.NE.0) Then
          LONC(P) = DATAN(YPC/XPC)
          If(XPC.LT.0.AND.YPC.GE.0) LONC(P) = LONC(P) + PI
          If(XPC.LT.0.AND.YPC.LT.0) LONC(P) = LONC(P) - PI
      Else
          If(YPC.GT.0) Then
              LONC(P) = PI / 2
          Else
              LONC(P) = -PI / 2
          End If
      End If
 1000 CONTINUE
C
      RETURN
      END
```