Technical Document 1641
September 1989

Specification for HF Maximum Usable Frequency (MUF) Model

T. N. Roy

Approved for public release; distribution is unlimited.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b. RESTRICTIVE MARKINGS:

2a. SECURITY CLASSIFICATION AUTHORITY:

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE:

3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S): NOSC TD 1641

5. MONITORING ORGANIZATION REPORT NUMBER(S):

6a. NAME OF PERFORMING ORGANIZATION: Naval Ocean Systems Center

6b. OFFICE SYMBOL (if applicable): Code 542

7a. NAME OF MONITORING ORGANIZATION:

6c. ADDRESS (City, State and ZIP Code): San Diego, CA 92152-5000

7b. ADDRESS (City, State and ZIP Code):

8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Naval Oceanographic Office

8b. OFFICE SYMBOL (if applicable):

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:

8c. ADDRESS (City, State and ZIP Code): NSTL Station
Bay St. Louis, MS 39522

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	AGENCY ACCESSION NO.
OMN		54-MP10	DN309 030

11. TITLE (include Security Classification): SPECIFICATION FOR HF MAXIMUM USABLE FREQUENCY (MUF) MODEL

12. PERSONAL AUTHOR(S):

13a. TYPE OF REPORT: Specification Document

13b. TIME COVERED: FROM Mar 1989 TO Sep 1989

14. DATE OF REPORT (Year, Month, Day): September 1989

15. PAGE COUNT: 40

16. SUPPLEMENTARY NOTATION:

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

maximum usable frequency
high frequency
electromagnetic propagation
communications modeling
ionospheric propagation

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This report documents a high-frequency (HF) maximum usable frequency (MUF) model called MINIMUF-85. Developed at the Naval Ocean Systems Center, this semiempirical model permits simplified HF propagation MUF predictions to be performed in near-real-time. The model predicts the MUF for arbitrary input values of receiver/transmitter position, date, time (UT), and solar sunspot number. Several algorithms are contained in the model to describe the dependence of the MUF on solar activity, geomagnetic latitude, time of year, and time of day. An appendix contains the computer FORTRAN 77 code required to implement the fundamental features of this model.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a. NAME OF RESPONSIBLE PERSON: T. N. Roy

22b. TELEPHONE (include Area Code): (619) 553-3068

22c. OFFICE SYMBOL: Code 542

CONTENTS

1.0 INTRODUCTION

Any communication system operating in the high-frequency (HF) spectrum between 2 and 32 MHz is subject to certain physical limitations on the propagation of its signal. These limitations determine propagation boundaries which are unique and definable for any given point in time and over any path. The upper boundary is called the maximum usable frequency (MUF). The instantaneous MUF is a direct function of ionospheric electron density and is an absolute frequency limit, since the ionosphere is not capable of supporting propagation by the refraction of higher frequencies for that path. Ionospheric electron density is driven both by solar activity and by seasonal and diurnal variations. A model developed to predict MUF must be able to deal with these variations.

In 1978 a semiempirical model for MUF prediction, called MINIMUF-3.5, was developed at the Naval Ocean Systems Center (NOSC). It was used on small mobile propagation forecast (PROPHET) terminals (Ref. 1 and 2). Values for the parameters used in the model were determined by using HF oblique sounder propagation data. A data base of HF oblique sounder data was used in 1981 to evaluate the accuracy of this model (Ref. 3).

In 1986 MINIMUF-3.5 was improved. Called MINIMUF-85, the new model contained improvements in the f_oF2 algorithms, high sunspot number predictions, an *M*-factor algorithm, and a polar region algorithm (Ref. 4). The accuracy of this improved model was evaluated in 1987 by using an expanded data base of HF oblique sounder data (Ref. 5). During this period several user's manuals were also published. These manuals contain specific information on how to use the HF propagation prediction models developed at NOSC (Ref. 6 and 7).

2.0 MUF MODEL INPUT AND OUTPUT PARAMETERS

MUF model input and output parameters and parameter limits are listed in Table 2-1. Monthly median sunspot values can be found in the Solar Indices Bulletin, published by the National Geophysical Data Center, Solar-Terrestrial Physics Division, located in Boulder, Colorado.

Table 2-1. MUF Model Input and Output Parameters.

Parameter Description	Parameter Limits
INPUT	
Transmitter Latitude	$-\pi/2$ to $\pi/2$ radians (90 deg south to 90 deg north latitude)
Transmitter West Longitude	0 to 2π radians (0 deg to 360 deg west longitude)
Receiver Latitude	$-\pi/2$ to $\pi/2$ radians (90 deg south to 90 deg north latitude)
Receiver West Longitude	0 to 2π radians (0 deg to 360 deg west longitude)
Month	1 to 12
Day	1 to 31
Hour	0 to 23 (Universal time)
Minute	0 to 59 (Universal time)
Julian Day	1 to 366
Year	Example: 89 for 1989
Monthly Median Sunspot Number	0 to 300
OUTPUT	
Maximum Usable Frequency	2 to 50 MHz
Distance Between Transmitter and Receiver	0 to π radians
Latitude of the Path Midpoint	$-\pi/2$ to $\pi/2$ radians
West Longitude of the Path Midpoint	0 to 2π radians
Latitude of a Point 1000 km from Receiver	$-\pi/2$ to $\pi/2$ radians
West Longitude of a Point 1000 km from Receiver	0 to 2π radians
Latitude of a Point 1000 km from Transmitter	$-\pi/2$ to $\pi/2$ radians
West Longitude of a Point 1000 km from Transmitter	0 to 2π radians
Azimuth from Receiver to Transmitter	0 to 2π radians

3.0 MUF MODEL COMPUTER PROGRAM, SUBROUTINES, AND FUNCTIONS

A program called CMUF passes input parameters to the subroutine MUF85 and receives the calculated MUF and geographic path information from this subroutine. The MUF85 subroutine calls the subroutines PATH and RAZGC and the functions FOF2 and SYGN.

The subroutine PATH computes eight values of geographic path information for a given propagation path. The computation assumes a spherical earth with a radius of 6371 km. Required inputs for this routine are transmitter and receiver latitude and west longitude in radians. The following information, in radians, is returned by this subroutine: distance between transmitter and receiver, midpoint latitude, midpoint west longitude, latitude of a point 1000 km from the transmitter and receiver, west longitude of a point 1000 km from the transmitter and receiver, and azimuth from receiver to transmitter. Subroutines GCRAZ and RAZGC are called by PATH.

Subroutine GCRAZ computes the great circle range and azimuth between two points on the earth's surface in radians. Latitude and west longitude of the two points, in radians, are required as input. This computation assumes a spherical earth and recognizes the degenerate cases of a point at the North or South Pole and coincident points.

Subroutine RAZGC computes the latitude and west longitude in radians of a point a specified distance and azimuth from a given point on the earth's surface. Latitude, west longitude, distance, and azimuth, in radians, of the given point to the new point are required as inputs. This computation assumes a spherical earth and recognizes the degenerate cases of the given point at the North or South Pole and when distance is zero.

Function FOF2 provides a correction to the calculation of the *F*2-layer critical frequency for polar latitudes by using the Chiu polar model (Ref. 8). Inputs to this function are the critical frequency in megahertz, local mean time at the control point, month, day, hour (UT), minute (UT), Julian day and year, geographic latitude and west longitude of the control point in radians, magnetic latitude of the control point in radians, and sunspot number (SSN). This function calculates the corrected *F*2-layer critical frequency, in megahertz, which is passed to the MUF85 subroutine.

Function SYGN returns the value of zero if the input value is zero, -1 if the input value is less than zero, and +1 if the input value if greater than zero.

4.0 MUF MODEL ALGORITHMS

The expression for the MUF used in MINIMUF-85 is given by

$$\mathrm{MUF} = A_2\,(\mathrm{SSN}) \cdot A_3\,(\mathrm{month}) \cdot A_4\,(\mathrm{time}) \cdot M \cdot f_o F2 \tag{1}$$

where M is the obliquity, or M-factor, which reflects the dependence of the MUF on the transmission path length. The parameter $f_o F2$ is the critical or penetration frequency at vertical incidence for the $F2$ layer. The parameters A_2, A_3, and A_4 contain the sunspot dependence, the seasonal dependence, and the time dependence in the M-factor.

The equation for the A_2 parameter is

$$A_2\,(\mathrm{SSN}) = 1.3022 - (0.00156)R \tag{2}$$

where R is the monthly median sunspot number. The A_2 parameters provide a linear decrease as a function of sunspot number.

The equation for the A_3 parameter is

$$\begin{aligned} A_3\,(\mathrm{month}) = {} & 0.9925 + 0.011 \sin(m) + 0.087 \cos(m) \\ & - 0.043 \sin(2m) + 0.003 \cos(2m) \\ & - 0.013 \sin(3m) - 0.022 \cos(3m) \\ & + 0.003 \sin(4m) + 0.005 \sin(5m) \\ & + 0.018 \cos(6m) \end{aligned} \tag{3}$$

where $m = 2\pi$, (month)/12 and month = 1, 2, 3... 12. This seasonally dependent factor attempts to account for unusually high electron density during winter months, which allows higher frequencies to propagate on a given transmission path.

The equation for the A_4 parameter is

$$A_4\,(\mathrm{time}) = 1.11 - 0.01\, T_{\mathrm{local}} \tag{4}$$

which adequately fits daytime data and for night

$$\begin{aligned} A_4\,(\mathrm{time}) = {} & 1.0195 - 0.06 \sin(2T) - 0.037 \cos(2T) \\ & + 0.018 \sin(4T) - 0.003 \cos(4T) \\ & + 0.025 \sin(6T) + 0.018 \cos(6T) \\ & + 0.007 \sin(8T) - 0.005 \cos(8T) \\ & + 0.006 \sin(10T) + 0.017 \cos(10T) \\ & - 0.009 \sin(12T) - 0.004 \cos(12T) \end{aligned} \tag{5}$$

where $T = T_{local} - T_{sunset}$, which represents elapsed time in hours after sunset.

The expression for the M-factor is given by

$$M = \{1 + 2.5[\sin(2.5\ \psi k)]^{3/2}\} \cdot G_1(\Delta T) \cdot G_2(L_o) \cdot G_3(L_1, L_2) \quad (6)$$

where ψ is the great-circle arc length, in radians, of the transmission path, L_o is the latitude of the path midpoint, L_1 and L_2 are the transmitter and receiver latitudes, and ΔT is the duration of daylight at the path midpoint in hours. The k parameter is equal to 1.0 for path lengths less than or equal to 4000 km and 0.5 for greater path lengths. The value of 2.5 ψk is limited to a maximum of $\pi/2$ for path lengths greater than 8000 km.

The first factor of Eq. 6 contains the range dependence of the M-factor and was obtained by curve-fitting to exact results for a parabolic layer at a height of 290 km with a ratio of semithickness to base height equal to 0.4. The G_1 factor gives recognition to the approximately 50% increase in F_2 layer heights observed at high (northern) latitudes during the summer at or near "midnight sun" conditions. The equation for the G_1 parameter is

$$G_1 = 1 - 0.1 \exp[(\Delta T - 24)/3] \quad (7)$$

which has the effect of a 10% reduction in MUF under full midnight sun conditions, with G_1 recovering rapidly as the midpath latitude moves toward the equator.

The G_2 parameter is designed to produce further, seasonally independent, reductions in MUF for high latitude paths. Since propagation data indicate a fairly abrupt onset of this reduction for absolute values of latitude greater than or equal to 45 degrees, a step function is used which is equal to 1 for absolute values of latitude less than 45 degrees and to 0.8 (20% reduction) for absolute values of latitude greater than or equal to 45 degrees.

The G_3 parameter is a correction factor for transequatorial paths to approximate the well-known MUF increases on such paths. The factor applies another 20% correction as a step function equal to 1.0 if the transmitter and receiver latitudes are the same sign and equal to 1.2 if the transmitter and receiver are of opposite sign.

The expression for the critical frequency, f_oF2, is given by

$$f_oF2 = [A_o + A_1(\mathrm{SSN}) \cdot (\cos \chi_{eff})^{1/2}]^{1/2} \quad (8)$$

where A_o is equal to 6 and

$$A_1(\mathrm{SSN}) = 22.23 + (0.814)R \quad (9)$$

where R is the monthly median sunspot number. The A_1 parameter provides a saturation effect in the behavior of the critical frequency as a function of sunspot number.

In Eq. 8, χ_{eff} is an "effective" solar zenith angle. Cos χ_{eff} is modeled as the lagged response of a dynamic linear system "driven" by the instantaneous value of cos χ. By using an effective value of the zenith angle, recognition is given to the fact that the $F2$ layer, unlike the E and D layers, does not show a relatively simple $\cos^n \chi$ diurnal dependence on χ. The dynamic behavior of the $F2$ layer is more complicated because of various other dependencies. In keeping with the uncomplicated nature of the f_oF2 model, defining an effective χ allows relatively accurate modeling without explicitly including these other dependencies. To model cos χ_{eff} at night, the following quantity is constructed:

$$(\cos\chi_{eff})_{night} = (\cos\chi_{eff})_{sunset} \cdot \exp[-(T - T_{sunset})/\tau_N] \quad (10)$$

In Eq. 10, τ_N is a nighttime relaxation time, equal to 2 hours, which is taken to be a constant independent of season and geographical location. $T - T_{sunset}$ is the elapsed time in hours since sunset.

However, during the day, a different relaxation time, τ_D, is assumed. In contrast to τ_N, τ_D does depend on location and position. The daytime relaxation time is assumed to be a function of the actual noontime solar zenith angle ($\cos\chi_{noon}$) and can be expressed as

$$\tau_D = \tau_o (\cos\chi_{noon})^{P_2} \quad (11)$$

where τ_o and P_2 are constants, equal to 9.7 and 9.6 hours, respectively, and are independent of season or location. The value of τ_D is not allowed to fall below 0.1. Note that during summer at equatorial and moderate latitudes, $\tau_D \to \tau_o$, whereas in the winter at high latitudes, $\tau_D \ll \tau_o$.

The time dependence of the actual $\cos\chi$ in terms of its value at noon, $\cos\chi_{noon}$, is

$$\cos\chi \approx \cos\chi_{noon} \cdot \sin\left[\frac{\pi(T - T_{dawn})}{\Delta T}\right] \quad (12)$$

where ΔT is the daytime duration given by

$$\Delta T = T_{sunset} - T_{dawn} \quad (13)$$

and noon occurs when $T = T_{dawn} + \Delta T/2$.

Next, $\cos\chi_{eff}$ is assumed to represent the response of a linear first-order system driven by the actual $\cos\chi$:

$$\tau_D \frac{d}{dt}(\cos\chi_{eff}) + \cos\chi_{eff} = \cos\chi \quad (14)$$

By using Eq. 12 for $\cos\chi$ in Eq. 14, we obtain the daytime $\cos\chi_{eff}$:

$$(\cos\chi_{eff})_{day} = \frac{\cos\chi_{noon}}{1+\beta^2}\left\{\sin(\alpha) + \beta\left(\exp\left[-\frac{(T - T_{dawn})}{\tau_D}\right] - \cos(\alpha)\right)\right\} \quad (15)$$

where

$$\beta = \frac{\pi\tau_D}{\Delta T} \quad (16)$$

and

$$\alpha = \frac{\pi(T - T_{dawn})}{\Delta T} \quad (17)$$

At sunset ($T - T_{dawn} = \Delta T$) and $(\cos\chi_{eff})_{sunset}$ is

$$(\cos \chi_{\text{eff}})_{\text{sunset}} = \frac{\cos \chi_{\text{noon}}}{1 + \beta^2} \left\{ \beta \left[1 + \exp\left(\frac{-\Delta T}{\tau_D} \right) \right] \right\} \tag{18}$$

To avoid discontinuities in cos χ_{eff} just after sunrise, $(\cos \chi_{\text{eff}})_{\text{day}}$ is not allowed to fall below its value just before sunrise.

$$(\cos \chi_{\text{eff}})_{\text{day}} = \max\{(\cos \chi_{\text{eff}})_{\text{sunset}} \cdot \exp[(\Delta T - 24)/\tau_N], \\ (\cos \chi_{\text{eff}})_{\text{day}}\} \tag{19}$$

The basic f_oF2 model is therefore given in Eq. 8, 10, and 18 for nighttime (i.e., $T_{\text{sunset}} < T < T_{\text{dawn}}$) and Eq. 8, 15, and 19 for daytime ($T_{\text{dawn}} < T < T_{\text{sunset}}$).

Simple analytical approximations for the times of local noon, sunrise, and sunset and for the noon value of the solar zenith angle are presented in the following equations. To an acceptable degree of accuracy, for the Universal time of local noon the approximation

$$T_{\text{noon}} = 12 \left(\frac{W}{\chi} + 1 \right) + 0.13 [\sin(Y_1) + 1.2 \sin(2Y_1)] \tag{20}$$

where

W = longitude west of Greenwich [0,2π radians]

$Y_1 = 0.0172\,(D + 10)$

$D = 30.4\,(M1 - 1) + D1$

$D1$ = day values which range from 1 to 31

$M1$ = month values which range from 1 to 12

In terms of the subsidiary variable Y_2, defined by

$$Y_2 = 0.409 \cos(Y_1) \text{ [radians]} \tag{21}$$

and

$$\cos \chi_{noon} = |\cos(L + Y_2)| \tag{22}$$

where

L = north latitude in radians ($-\pi/2$, $\pi/2$)

The duration of the daytime, ΔT, is approximated by

$$\Delta T = \frac{24}{\pi} \arccos \left(\frac{-0.26 + \sin(Y_2)\sin(L)}{\cos(Y_2)\cos(L)} \right) \tag{23}$$

where the factor -0.26 approximates the difference between sunrise (or sunset) at the surface of the earth and at $2F_2$ layer heights. From Eq. 23

$$T_{dawn} = T_{noon} - \Delta T/2 \tag{24}$$

and

$$T_{sunset} = T_{noon} + \Delta T/2 \tag{25}$$

where

T_{noon} is given in Eq. 20.

The fact that high-latitude ionospheric behavior differs sharply from that at lower latitudes has been recognized for some time. This sharp difference required the addition of a routine to MINIMUF-85 specifically tailored to model the behavior of the MUF at high latitudes. This is accomplished by using a folding function, f, to merge a polar model with the model of lower latitude behavior.

The folding function determines when polar effects (particle precipitation) become dominant. It is a function of geomagnetic latitude and sunspot number and makes an abrupt transition from 0 to 1 between geomagnetic latitudes of 60 degrees to 75 degrees. When the folding function is near 1, particle precipitation effects will dominate, and when the folding function is near 0, solar zenith angle is the major factor in causing ionization. In between, there exists a narrow transition region where both sources of free electrons are significant. The equation for merging the two models is

$$N_{total} = (1 - f)\, N_{MINIMUF} + f N_{polar} \tag{26}$$

where N is the electron density calculated from the expression

$$f_oF2\,(\text{MHz}) = 2.85 N^{1/2}\ [\text{electrons/cm}^3] \tag{27}$$

The total electron density at the control point is then converted back to f_oF2 by using Eq. 26, and the MUF is obtained by multiplying the value of f_oF2 by the range-dependent portion of the M-factor.

The equation for the folding function is

$$f = \exp(-X^6) \tag{28}$$

where

$$X = [2.2 + (0.2 + R/1000) \sin(L_1)] \cos(L_1) \tag{29}$$

where

R = monthly median sunspot number

and

L_1 = magnetic latitude of the control point in radians.

The equation for the polar model of electron density for northern geomagnetic latitudes is

$$N_{polar} = (2.0 + 0.012R)(1.0 + 0.3V)W \tag{30}$$

where

$$V = \sin(\pi_{month}/12) \tag{31}$$

and

$$W = \exp\left(-1.2\left\{\cos\left[L_1 - 0.41015\cos\left(\frac{\pi T}{12}\right)\right] - \cos(L_1)\right\}\right) \tag{32}$$

with

L_1 = magnetic latitude of the control point in radians

and

T = the local mean time at the control point in hours

The expression for electron density for southern geomagnetic latitudes is

$$\begin{aligned} N_{polar} = &\left\{2.5 + R/50 + U[0.5 + S(1.3 + 0.002R)]\right. \\ &\left. + (1.3 + 0.005R)\cos\left[\frac{\pi T}{12} - \pi(1 + B)\right]\right\} \\ &\cdot [1 + 0.4(1 - V^2)]\exp(-VS) \end{aligned} \tag{33}$$

where

$$U = \cos(2\pi_{month}/12) \tag{34}$$

$$S = \cos^4(D_4/2 - \pi/20) \tag{35}$$

$$\begin{aligned} B = &V\{[\sin(D_4/2) - \sin(D_4)]/2 - \sin^8(D_4/2)\} \\ &- (1 + V)U\sin(D_4/Z)\exp[-4\sin^2(D_4/2)] \end{aligned} \tag{36}$$

and

$$Z = |\sin(D_4)|^{1/2} \tag{37}$$

where

$$D_4 = \sin^{-1}(D_3) \tag{38}$$

where

D_3 = the maximum of D_2 or -1.0

and

D_2 = the minimum of D_1 or 1.0

with

$$D_1 = \cos(L_o) \sin(D_o - 1.2043) \cos(L_1) \tag{39}$$

where

L_o = geographic latitude in radians

and

D_o = geographic west longitude in radians

The equations in the preceding sections yield f_oF2 at a specified latitude, L, and west longitude, W. In an actual application, the latitude and longitude of the receiver and transmitter are given quantities, and L and W are to be evaluated at specific control points along the great-circle propagation path.

From spherical trigonometry, one obtains the following expressions for the great-circle arc length, ψ (in radians), connecting two points defined by the (latitude, longitude) pairs (L_1, W_1) and (L_2, W_2):

$$\psi = \text{arc cos}\,[\sin(L_1)\sin(L_2) + \cos(L_1)\cos(L_2)\cos(W_2 - W_1)] \tag{40}$$

If one travels a fraction, K, of the distance from point 1 to point 2, the north latitude, L_K, and west longitude, W_K, of this location are determined from

$$L_K = \text{arc sin} \left\{ \frac{\sin[\psi(1 - K)]\sin(L_1) + \sin(\psi K)\sin(L_2)}{\sin(\psi)} \right\} \tag{41}$$

and

$$W_K = \text{arc cos} \left\{ \frac{\cos(L_1)\cos(W_1)\sin[\psi(1 - K)] + \cos(L_2)\cos(W_2)\sin(\psi K)}{\cos(L_K)\sin(\psi)} \right\} \tag{42}$$

The calculated MUF is the minimum value evaluated at specific control points along the great-circle propagation path. In MINIMUF-85 these control points are located at the path midpoint for path lengths less than or equal to 4000 km, 2000 km from either terminus for path lengths greater than 4000 km, but less than or equal to 6000 km and at 1/4, 1/2, and 3/4 of the path length for paths greater than 6000 km and less than or equal to 8000 km in length. For path lengths greater than 8000 km and less than or equal to 12000 km, the control points are located at 1/6, 1/3, 1/2, 2/3, and 5/6 of the path length.

In MINIMUF-85 the geomagnetic latitude dependence accounts for critical frequency separation between ordinary and extraordinary ionospheric propagation by adding one-half the gyrofrequency to the $f_o F2$ for latitudes greater than 55°N geomagnetic. The gyrofrequency for an earth-centered dipole field is given by

$$f_H = 0.3789 \left[1 + 3 \sin^2 (\theta)\right]^{1/2} - 0.5 \text{ [MHz]} \tag{43}$$

where

θ = latitude of the midpoint of the propagation path in magnetic coordinates (radians)

The geomagnetic latitude θ is given by

$$\sin (\theta) = \sin (\phi) \sin (\phi_o) + \cos (\phi) \cos (\phi_o) \cos (\lambda - \lambda_o) \tag{44}$$

where

ϕ = latitude of the midpoint of the propagation path (radians)

λ = longitude of the midpoint of the propagation path (radians)

ϕ_o = latitude of the North Magnetic Pole (1.3666 radians or 78.3°N)

λ_o = longitude of the North Magnetic Pole (1.2043 radians or 69°W).

Substituting the values of ϕ_o and λ_o, Eq. 44 becomes

$$\sin (\theta) = 0.9792 \sin (\phi) + 0.2028 \cos (\phi) \cos (\lambda - 1.2043) \tag{45}$$

5.0 MUF MODEL TEST CASES

The following tables of test case results are provided as an aid in determining the proper operation of the MUF model algorithms. Table 5-1 lists the results of exercising the MUF model for the range of season values. A sunspot value of 75 was used to generate these values. Transmitter latitude was 0.5712 radian, transmitter longitude was 2.0450 radians, receiver latitude was 0.5304 radian, and receiver longitude was 1.5645 radians. Seasonal values were calculated for the 15th day of January, April, July, and October.

Table 5-1. MUF model season results (MHz).

Time (UT)	Season (month number)			
	Winter (1)	Spring (4)	Summer (7)	Fall (10)
0000	22.28	28.08	24.22	26.85
0400	11.39	21.97	21.59	16.56
0800	11.29	17.71	16.71	14.52
1200	8.94	14.25	13.71	13.29
1600	28.10	25.68	21.32	32.63
2000	29.05	29.08	24.21	33.49

Table 5-2 lists the results of exercising the MUF model for a solar cycle. Locations of the transmitter and receiver were the same as those in Table 5-1. The date used was 15 January.

Table 5-2. MUF model solar cycle results (MHz).

Time (UT)	Solar cycle (sunspot number)				
	Minimum) (10)	Rise and Decline (45)	Near Maximum (75)	Maximum (105)	High Maximum (150)
0000	16.23	20.01	22.28	23.96	25.66
0400	10.37	11.00	11.39	11.67	11.89
0800	11.27	11.31	11.29	11.22	11.05
1200	9.37	9.14	8.94	8.72	8.36
1600	19.96	25.06	28.10	30.34	32.62
2000	20.42	25.83	29.05	31.41	33.83

Tables 5-3 through 5-7 list the results of exercising the MUF model for various locations of the transmitter and receiver. The date used for the following tests was 15 January at 1200 UT and the sunspot number was 75.0.

Table 5-3. MUF model results (MHz) for transmitter at 75 degrees north latitude (1.3090 radians) and 150 degrees west longitude (2.6180 radians).

Receiver, North Latitude, deg (radians)	Receiver, West Longitude, deg (radians)					
	0 (0.0000)	60 (1.0472)	120 (2.0944)	180 (3.1416)	240 (4.1888)	300 (5.2360)
75 (1.3090)	17.40	13.81	6.63	5.98	10.55	13.60
35 (0.6109)	18.78	15.13	8.08	7.54	8.73	9.85
0 (0.0000)	18.73	14.61	11.19	10.78	10.55	11.13
-35 (-0.6109)	18.46	12.39	11.59	12.03	11.46	11.89
-75 (-1.3090)	11.04	10.54	11.66	12.40	11.50	11.61

Table 5-4. MUF model results (MHz) for transmitter at 35 degrees north latitude (0.6109 radian) and 150 degrees west longitude (2.6180 radians).

Receiver, North Latitude, deg (radians)	Receiver, West Longitude, deg (radians)					
	0 (0.000)	60 (1.0472)	120 (2.0944)	180 (3.1416)	240 (4.1888)	300 (5.2360)
75 (1.3090)	9.67	9.11	7.67	8.01	9.57	9.97
35 (0.6109)	9.62	10.28	10.75	11.44	12.54	9.17
0 (0.0000)	10.31	11.78	12.68	14.18	14.04	10.67
-35 (-0.6109)	13.40	15.44	16.49	17.50	16.75	15.71
-75 (-1.3090)	18.17	17.43	17.35	17.77	18.02	18.46

Table 5-5. MUF model results (MHz) for transmitter at 0 degrees north latitude (0.0 radians) and 150 degrees west longitude (2.6180 radians).

Receiver, North Latitude, deg (radians)	Receiver, West Longitude, deg (radians)					
	0 (0.0000)	60 (1.0472)	120 (2.0944)	180 (3.1416)	240 (4.1888)	300 (5.2360)
75 (1.3090)	10.57	11.73	10.58	11.20	10.32	11.20
35 (0.6109)	9.96	11.91	12.53	14.15	14.64	10.79
0 (0.0000)	15.35	15.35	18.25	22.25	22.67	22.67
-35 (-0.6109)	17.39	16.47	15.04	19.26	23.38	21.42
-75 (-1.3090)	19.95	20.72	19.90	18.36	17.60	18.30

Table 5-6. MUF model results (MHz) for transmitter at -35 degrees north latitude (-0.6109 radian) and 150 degrees west longitude (2.6180 radians).

Receiver, North Latitude, deg (radians)	Receiver, West Longitude, deg (radians)					
	0 (0.0000)	60 (1.0472)	120 (2.0944)	180 (3.1416)	240 (4.1888)	300 (5.2360)
75 (1.3090)	11.57	11.79	11.60	12.30	11.18	11.48
35 (0.6109)	18.77	16.75	15.13	17.37	17.73	26.67
0 (0.0000)	18.21	18.37	14.51	19.90	26.58	20.62
-35 (-0.6109)	17.90	19.67	14.93	21.04	25.79	17.22
-75 (-1.3090)	18.65	16.23	15.31	15.42	16.30	18.74

Table 5-7. MUF model results (MHz) for transmitter at -75 degrees north latitude (-1.3090 radians) and 150 degrees west longitude (2.6180 radians).

Receiver, North Latitude, deg (radians)	Receiver, West Longitude, deg (radians)					
	0 (0.0000)	60 (1.0472)	120 (2.0944)	180 (3.1416)	240 (4.1888)	300 (5.2360)
75 (1.3090)	20.95	10.80	11.62	11.93	11.22	11.32
35 (0.6109)	25.18	26.95	14.86	18.11	17.81	18.35
0 (0.0000)	23.67	26.64	16.12	22.51	17.65	19.56
-35 (-0.6109)	22.98	23.73	16.08	16.97	16.43	18.34
-75 (-1.3090)	25.91	22.32	11.82	11.16	15.46	17.93

6.0 REFERENCES

1. Rose, R.B., J.N. Martin, P.H. Levine, "MINIMUF-3: A Simplified HF MUF Prediction Algorithm," Naval Ocean Systems Center, San Diego, Calif., NOSC TR 186, February 1978.
2. Rose, R.B., J.N. Martin, "MINIMUF-3.5, Improved Version of MINIMUF-3, A Simplified HF MUF Prediction Algorithm," Naval Ocean Systems Center, San Diego, Calif., NOSC TD 201, October 1978.
3. Sailors, D.B., W.K. Moision, R.P. Brown, "Accuracy of High Frequency Maximum Usable Frequencies (MUF) Prediction," Naval Ocean Systems Center, San Diego, Calif., NOSC TR 695, September 1981.
4. Sailors, D.B., R.A. Sprague, W.H. Rix, "MINIMUF-85: An Improved HF MUF Prediction Algorithm," Naval Ocean Systems Center, San Diego, Calif., NOSC TR 1121, July 1986.
5. Roy T.N., D.B. Sailors, "HF Maximum Usable Frequency (MUF) Model Uncertainty Assessment," Naval Ocean Systems Center, San Diego, Calif., NOSC TR 1184, June 1987.
6. LaBahn, R.W., "Operational Users Manual for Advanced PROPHET on MS-DOS-Based Microcomputer Systems," Naval Ocean Systems Center, San Diego, Calif., NOSC TD 848, August 1985.
7. Sailors, D.B., "Operational User's Manual for Tactical Decision Aids for HF Communication," Naval Ocean Systems Center, San Diego, Calif., NOSC TD 1065, July 1987.
8. Chiu, Y.T., "An Improved Phenomenological Model of Ionospheric Density," *Journal of Atmospheric and Terrestrial Physics*, Volume 37, pp. 1563–1570, 1975.

Appendix

FORTRAN 77 PROGRAM AND SUBROUTINE LISTINGS FOR MUF MODEL

The MUF calculation program and subroutines that follow are written in FORTRAN 77. The parameters passed to the subroutines and those returned are described in the comments portion of the routines.

```
      program cmuf
cp***********************************************************************
c     This program calculates the classical maximum usable frequency
c     (MUF) given the values specified.
c
c     Parameters input:
c        tlat:      transmitter latitude (radians), + north, - south
c        tlon:      transmitter longitude (radians), + west, - east
c        rlat:      receiver latitude (radians), + north, - south
c        rlon:      receiver longitude (radians), + west, - east
c        itime(1): month
c        itime(2): day
c        itime(3): hour (UT)
c        itime(4): minute (UT)
c        itime(5): julian day
c        itime(6): year
c        ssn:       monthly median sunspot number
c
c     Parameters returned:
c        muf:  classical MUF value in MHz
c        cpnt: geographic path information in radians
c
c     Subroutines used:  muf85
c
cz***********************************************************************
      real cpnt(8)
      real tlat
      real tlon
      real rlat
      real rlon
      real ssn
      real muf
      integer itime(6)
c
c     Initialize MUF and geographic path information variables
c     prior to MUF85 subroutine call
c
      muf=0.0
      do 100 i=1,8
         cpnt(i)=0.0
100   continue
c
c     Enter MUF calculation parameters
c
      tlat=0.37385
      tlon=2.76024
      rlat=0.57125
      rlon=2.0450
      itime(1)=1
      itime(2)=1
      itime(3)=0
      itime(4)=0
      itime(5)=1
      itime(6)=89
      ssn=100.0
c
c     Calculate classical MUF using MINIMUF-85
c
```

```
call muf85 (tlat, tlon, rlat, rlon, itime, cpnt, ssn, muf)
end
```

```
      subroutine muf85 (tlat, tlon, rlat, rlon, itime, cpnt, ssn,cmuf)
cp*************************************************************************
c     subroutine muf85
c
c     call muf85(tlat,tlon,rlat,rlon,itime,ssn,cmuf)
c
c     Updated Nov. 1985
c     An improved version of muf35 which includes sunspot,season and
c     diurnal dependence in the M-factor plus an improved FOF2 model
c     which includes a polar region modification.
c
c     This routine computes the classical maximum useable frequency
c     (cmuf) for a given propagation path.  The required input is:
c
c           tlat:  transmitter latitude in radians
c           tlon:  transmitter west longitude in radians
c           rlat:  receiver latitude in radians
c           rlon:  receiver west longitude in radians
c           itime: six element array containing the (1)month,(2)day
c                  (3)hour,(4)minute,(5)julian day, and (6)year
c           ssn:   sunspot number
c
c        parameters returned:
c           cpnt:  geographic path information in radians
c           cmuf:  classical muf in megahertz.
c
c        called by subroutine or function:  mufluf
c
c        subroutines and functions called:  fof2
c                                                   path
c                                                   razgc
c                                                   sygn
c
c        method:  Uses the functional form for the foF2 and M-factor
c                 described in NOSC TR-186 (MINIMUF-3: A simplified
c                 HF MUF Prediction Algorithm, Rose R.B.,J.N. Martin
c                 and P.H. Levine, 1978). Includes improvements as
c                 described in NOSC TR-1121 (MINIMUF-85: An improved
c                 HF MUF Prediction Algorithm, Sailors,D.B.,R.A.
c                 Sprague and W.H. Rix).
cz*************************************************************************
      integer itime(6)
      integer khop
      integer kkhop
      real    cpnt(8)
      real    mfctk
      real    cplat
      real    lmt
      real    ltnoon
      real    daylen
      real    mfctor
      real    cpmlat
      real    sn(6)
      real    cn(6)
      real    tlat
      real    tlon
      real    rlat
      real    rlon
```

```
      real     ssn
      real     tday
      real     qn
      real     arg
      real     cfssn
      real     mfssn
      real     mfmth
      real     pthlen
      real     azim
      real     pl4000
      real     cmuf
      real     cplon
      real     xk1
      real     xhop
      real     ak1
      real     plkhop
      real     smg
      real     gyro
      real     tsol
      real     soldec
      real     mfdrnl
      real     cosxef
      real     tsunrs
      real     tsunst
      real     cnxef
      real     tdrlx
      real     tday1
      real     tnite
      real     tnite1
      real     ag
      real     ag1
      real     ag2
      real     ag3
      real     ag4
      real     ag5
      real     ag6
      real     a14
      real     a24
      real     a34
      real     a44
      real     beta
      real     relax
      real     dayrlx
      real     alpha
      real     trnxef
      real     cpmuf
      data     pi/3.14159265/,twopi/6.2831853/,halfpi/1.57079632/,
     &          dtr/0.017453293/,rtd/57.2957795/,r0/6371./
c
      tday = float( itime(3) ) + float( itime(4) )/60.0
c
c   mfdrnl: contains the diurnal dependence of the M-factor
c   mfmth:  calculated using a 6th order fourier series which is
c             a function of month in the new M-factor
c   mfssn:  a linear function of ssn in the M-factor
c   cfssn:  a linear function of ssn in the critical frquency
c             expression
c
```

```
c    M-factor seasonal dependence calculation
c
        do n = 1,6,1
          qn = float(2*n)
          arg = pi*qn*itime(1)/12.0
          sn(n) = sin(arg)
          cn(n) = cos(arg)
        end do
        mfmth = .9925+.011*sn(1)+.087*cn(1)-.043*sn(2)
     &       +.003*cn(2)-.013*sn(3)-.022*cn(3)
     &       +.003*sn(4)+.005*sn(5)+.018*cn(6)
c
c    foF2 sunspot number dependence calculation
c
        cfssn = .814*ssn+22.23
c
c    M-factor sunspot number dependence calculation
c
        mfssn = 1.3022-.00156*ssn
c
c    control point calculation
c
        call path(tlat,tlon,rlat,rlon,cpnt)
        pthlen = cpnt(1)
        azim = cpnt(8)
c
c    control point changes for muf85
c
            pl4000 = 1.59*cpnt(1)
            if(pl4000 .lt. 1.0)pl4000=1.0
            mfctk = 1.0/pl4000
            if(mfctk .ne. 1.0) mfctk = .5
            khop = int(cpnt(1)/.62784)+1
            kkhop = khop
            if(cpnt(1) .gt. 0.94174)kkhop = 2*khop-1
c
        cmuf=100.0
        do 800 k1 = 1,kkhop
c
c    cplat, cplon = latitude and west longitude of control points
c    control point method for muf85
c
        if(cpnt(1) .gt. .94174) then
            xk1 = k1
            xhop = khop
            ak1 = xk1/(2.0*xhop)
        else
            ak1 = 1.0/(2.0*pl4000)+float(k1-1)*(.9999-1.0/pl4000)
        end if
c
        plkhop = pthlen*ak1
c
          call razgc( rlat, rlon, plkhop, azim, cplat, cplon )
c
c    lmt = local mean time in hours at the control point
c    cpmlat = geomagnetic latitude at the control point
c
          if ( cplon .ge. 0.0 )  then
```

```
          lmt = cplon
        else
          lmt = cplon + twopi
        end if
        lmt = tday - lmt*rtd/15.0
        if ( lmt .lt. 0.0 )  then
          lmt = lmt + 24.0
        else if ( lmt .ge. 24.0 )  then
          lmt = lmt - 24.0
        end if
c
c   calculation of geomagnetic latitude at the control point
c
        smg = 0.9792*sin(cplat) + 0.2028*cos(cplat)*cos(cplon - 1.2043)
        smg = amax1( amin1( smg, 1.0 ), -1.0 )
        cpmlat = asin( smg )
c
c   geomagnetic latitude foF2 dependence calculation
c   gyro frequency for latitude > 55 degrees
c
        if ( abs( cpmlat ) .lt. 0.95993 )  then
          gyro = 0.0
        else
          gyro = 0.3789*sqrt( 1.0 + 3.0*smg*smg) - 0.5
        end if
c
c   calculation of local noon time
c   tsol = 2*pi*date/365.25
c   soldec = -solar declination
c   ltnoon = time of local noon
c
        tsol = 0.0172*(10.0 + float(itime(1)-1)*30.4 + itime(2))
        soldec = 0.409*cos(tsol)
c
        ltnoon = 3.82*cplon+12.0+0.13*(sin(tsol)+1.2*sin(2.0*tsol))
        if ( ltnoon .gt. 24.0 )  then
          ltnoon = ltnoon - 24.0
        else if ( ltnoon .le. 0.0 )  then
          ltnoon = ltnoon + 24.0
        end if
c
c   M-factor range dependence calculation
c   mfctor = M-factor = muf/f0F2
c
        mfctor = amin1( 2.5*pthlen*mfctk, halfpi )
        mfctor = sin( mfctor )
        mfctor = 1.0 + 2.5*mfctor*sqrt( mfctor )
c
c   daylen = length of daylight
c   tsunrs = time of sunrise
c   tsunst = time of sunset
c   tdrlx = daytime relaxation time
c   night time relaxation time = 2 hours
c
        if ( cos( cplat + soldec ) .gt. -0.26 )  go to 600
c
c   no daylight on path at any time during the day
c
```

```
          cosxef = 0.0
          daylen = 0.0
          mfdrnl = 1.0
          go to 700
600       continue
c
c    delta T = length of daylight calculation T(sunset)-T(dawn)
c
          daylen = (-0.26+sin(soldec)*sin(cplat))/(cos(soldec)*cos(cplat)
     &              +1.0e-3)
          daylen = amax1( amin1 ( daylen, 1.0 ), -1.0 )
          daylen = 12.0 - asin( daylen )*7.6394
c
c    T(dawn) calculation
c
          tsunrs = ltnoon - daylen/2.0
          if ( tsunrs .lt. 0.0 )  tsunrs = tsunrs + 24.0
c
c    t(sunset) calculation
c
          tsunst = ltnoon + daylen/2.0
          if ( tsunst .gt. 24.0 )  tsunst = tsunst - 24.0
c
c    calculation of the cosine noon time solar zenith angle
c
          cnxef = abs( cos( cplat + soldec ) )
c
c    calculation of the day time relaxation time
c
          tdrlx = 9.7*( amax1( cnxef, .1 ) )**9.6
          tdrlx = amax1( tdrlx, 0.1 )
          tday1 = tday
        if (( tsunst .lt. tsunrs .and. (tday-tsunst)*(tsunrs-tday) .gt.
     &       0.0 ) .or. ( tsunst .ge. tsunrs .and. (tday-tsunrs)*
     &       (tsunst-tday) .le. 0.0 ))then
c
c    night time at control point
c    M-factor night time dependence calculation
c    6th order fourier series night time factor for M-factor,
c    based on hours after sunset, tnite
c
             if ( tsunst .gt. tday )  tday1 = tday1 + 24.0
             tnite = tday1 - tsunst
             tnite1 = 14.0*tnite/(24.0-daylen)
             ag = pi *(tnite1+1.0)/15.0
             ag1 = 2.0*ag
             ag2 = 4.0*ag
             ag3 = 6.0*ag
             ag4 = 8.0*ag
             ag5 = 10.0*ag
             ag6 = 12.0*ag
             a14 = 1.0195 -.06*sin(ag1)-.037*cos(ag1)+.018*sin(ag2)
             a24 = -.003*cos(ag2)+.025*sin(ag3)+.018*cos(ag3)
             a34 = .007*sin(ag4)-.005*cos(ag4)+.006*sin(ag5)
             a44 = .017*cos(ag5)-.009*sin(ag6)-.004*cos(ag6)
             mfdrnl = a14+a24+a34+a44
c
             beta = pi*tdrlx/daylen
```

```
c
c   (T(sunset)-T)/night relaxation time
c
            relax = (tsunst-tday1)/2.0
            relax = amin1( amax1( relax, -75.0 ), +75.0 )
c
c   - delta T/day relaxation time
c
            dayrlx = -daylen/tdrlx
            dayrlx = amin1( amax1( dayrlx, -75.0 ), +75.0 )
c
c   calculation of sunset cosine effective solar zenith angle
c
            cosxef = cnxef*(beta*(exp(dayrlx)+1.0))*exp(relax)/
     &                  (1.0+beta*beta)
      else
c
c   day time at the control point
c
           if ( tsunrs .gt. tday )  tday1 = tday1 + 24.0
c
c   M-factor day time dependence calculation
c
           mfdrnl = 1.11-.01 * lmt
c
           alpha = pi*( tday1 - tsunrs )/daylen
           beta = pi*tdrlx/daylen
c
c   (T(dawn)-T)/day relaxation time
c
           relax = ( tsunrs - tday1 )/tdrlx
           relax = amin1( amax1( relax, -87.0 ), +87.0 )
           dayrlx = -daylen/tdrlx
           dayrlx = amin1( amax1( dayrlx, -87.0 ), +87.0 )
c
c   calculation of day cosi..e effective solar zenith angle
c
           cosxef = cnxef*(sin(alpha)+beta*(exp(relax)-cos(alpha)))/
     &                  (1.0+beta*beta)
c
c   sunrise transition
c
           trnxef = cnxef*( beta*( exp( dayrlx ) + 1.0 ) )
     &                  *exp( ( daylen - 24.0 )/2.0 )/( 1.0 + beta*beta )
           cosxef = amax1( cosxef, trnxef )
c
      end if
c
c   cpmuf = muf at the control point
c   calculation of foF2
c
700     cpmuf = sqrt(6.0 + cfssn * sqrt(cosxef)) + gyro
c
c   MUF high latitude season dependence calculation
c
        cpmuf = cpmuf*( 1.0 - 0.1*exp( ( daylen - 24.0 )/3.0 ) )
c
c   MUF transequatorial path dependence calculation
```

```
c
          cpmuf = cpmuf*( 1.0 + ( 1.0 - sygn( tlat )*sygn( rlat ) )*0.1 )
c
c    MUF high latitude path dependence calculation
c
          cpmuf = cpmuf*( 1.0 - 0.1*( 1.0 + sygn( abs( sin( cplat ) )
     &        - cos( cplat ) ) ) )
c
c    fof2 function corrects for polar region foF2
c    result is cpmuf if control point not in polar region
c
          if ( abs( cpmlat ) .ge. 0.95993 )  then
            cpmuf = mfctor*fof2( cpmuf,lmt,itime,cplat,cplon,cpmlat,ssn )
          else
            cpmuf = cpmuf*mfctor
          end if
c
c    MUF sunspot, season and diurnal dependence
c
        cpmuf = cpmuf*mfssn*mfmth*mfdrnl
          cmuf = amin1( cmuf, cpmuf )
800     continue
c
c    MUF minimum value 2MHz, maximum value 50 MHz
c
        cmuf = amin1( amax1( cmuf, 2.0 ), 50.0 )
c
        return
        end
```

```
      subroutine path (tlat, tlon, rlat, rlon, cpnt)
cp*********************************************************************
c       subroutine path
c
c       call path(tlat,tlon,rlat,rlon,cpnt)
c
c       This routine computes the range, azimuth and control point
c       coordinates for a given propagation path.  The method assumes
c       a spherical earth with a radius of 6371 km.
c
c       Parameters input:
c               tlat: transmitter latitude in radians
c               tlon: transmitter west (positive) longitude in radians
c               rlat: receiver latitude in radians
c               rlon: receiver west (positive) longitude in radians
c
c     This subroutine returns geographic path information in an 8 word
c     real array (cpnt).
c
c     Parameters returned:
c       cpnt(1): distance between the receiver and transmitter in
c                radians
c       cpnt(2): latitude of midpoint in radians
c       cnpt(3): west longitude in radians
c       cpnt(4): latitude of point 1000 km from the receiver in radians
c       cpnt(5): west longitude of point 1000 km from receiver in radians
c       cpnt(6): latitude of point 1000 km from transmitter in radians
c       cpnt(7): west longitude of point 1000 km from transmitter
c                in radians
c       cpnt(8): azimuth from receiver to transmitter in radians
c
c      * cpnt(4) through cpnt(7) will not be computed for paths less
c                than 1000 km (0.15696 radians) in length.
c
c       Subroutines and functions used:  gcraz
c                                        razgc
c
c       Common blocks:  none
c
cz*********************************************************************
c
      real cpnt(8)
      real rlat
      real rlon
      real tlat
      real tlon
      real pl
c
c     Get range and azimuth between points 1 and 2
c
      call gcraz( rlat, rlon, tlat, tlon, cpnt(1), cpnt(8) )
c
c     Get mid-point coordinates
c
      pl = cpnt(1)/2.0
      call razgc( rlat, rlon, pl, cpnt(8), cpnt(2), cpnt(3) )
c
c     Is path length >= 1000 km?
```

```
c
      if ( cpnt(1) .ge. 0.156961231 ) then
c
c     Get coordinates of 1000 km points
c
         pl = 0.156961231
         call razgc( rlat, rlon, pl, cpnt(8), cpnt(4), cpnt(5) )
         pl = cpnt(1) - 0.156961231
         call razgc( rlat, rlon, pl, cpnt(8), cpnt(6), cpnt(7) )
         return
      else
         return
      end if
      end
```

```
      subroutine razgc( lat1, lon1, range, azim, lat2, lon2 )
cp*********************************************************************
c       subroutine razgc
c
c          call razgc(lat1,lon1,range,azim,lat2,lon2)
c
c          This routine computes the latitude and west(positive) longi-
c          tude (lat2, lon2) of a point a specified range from a given
c          point on the earth's surface.  Also required for input
c          is the azimuth (azim) to the new point in radians.  This
c          method assumes a spherical earth (6371.0 km) and recognizes
c          the degenerate cases of the given point being at the north
c          or south pole.  For the degenerate cases, azim should be 0
c          or pi and lon2 is undefined.  However, azim is not checked,
c          and lon2 is arbitrarily set equal to lon1.  This routine
c          recognizes the degenerate case when range is set to zero.
c          All coordinates are in radians.
c
c          Parameters input: lat1, lon1, range, azim
c
c          Parameters returned: lat2, lon2
c
c          Subroutines and functions used:   none
c
c          Common blocks:  none
c
c          Method: Uses law of cosines for sides on spherical triangle
c                  defined by (lat1,lon1), north pole and point defined
c                  by azim and range.
cz*********************************************************************
c
      real lat1
      real lon1
      real lat2
      real lon2
      real s1
      real c1
      real cr
      real ca
      real cg
      real a
      real g
      real sa
c
      data  pi/3.141592654/,twopi/6.283185308/,halfpi/1.570796327/
     &        rtd/57.29577951/,dtr/0.017453293/
c
c     Test for degenerate cases
c        1)Given point is north or south pole:
c
      if ( abs( lat1 - halfpi ) .le. 1.0e-5 ) then
c
c     The given point is the north pole
c
         lat2 = halfpi - range
         lon2 = lon1
         return
      else
```

```
            if ( abs( lat1 + halfpi ) .le. 1.0e-5 ) then
c
c       The given point is the south pole
c
                lat2 = range - halfpi
                lon2 = lon1
                return
            end if
        end if
c
c           2)Coincident points:
c
        if ( range .eq. 0.0 ) then
c
c       Point 2 coincident with point 1
c
            lat2 = lat1
            lon2 = lon1
            return
        end if
c
c       General case
c
        s1 = sin( lat1 )
        c1 = cos( lat1 )
        cr = cos( range )
        ca = s1*cr + c1*sin( range )*cos( azim )
        ca = amin1( amax1( ca, -1.0 ), +1.0 )
        a = acos( ca )
c
c       Test if destination ends up on the poles
c
        if( abs(a).le.1.0e-5 ) then
            lat2 = halfpi
            lon2 = lon1
            return
        else
            if( abs(a-pi) .le. 1.0e-5 ) then
                lat2 = -halfpi
                lon2 = lon1
                return
            end if
        end if
c
c       Get destination coordinates
c
        cg = ( cr - s1*ca )/( c1*sin( a ) )
        cg = amin1( amax1( cg, -1.0 ), +1.0 )
        g = acos( cg )
        lat2 = halfpi - a
        sa = sin( azim )
        if ( sa .ge. 0.0 )  lon2 = amod( lon1 - g, twopi )
        if ( sa .lt. 0.0 )  lon2 = amod( lon1 + g, twopi )
        return
        end
```

```
      subroutine gcraz( lat1, lon1, lat2, lon2, range, azim )
cp*********************************************************************
c      subroutine gcraz
c
c         call gcraz (lat1,lon1,lat2,lon2,range,azim)
c
c         This routine computes the great circle range and azimuth
c         between two points on the earth's surface.  lat1 and lon1
c         are the coordinates of point 1 , lat2 and lon2 are the
c         coordinates of point 2.  Both longitudes are west longitudes.
c         West longitudes are positive throughout the Muf85 algorithm.
c         Latitudes are positive if north and negative if south.
c         The output is range, the distance between the two points
c         in radians and azim, the azimuth from one point to the other
c         in radians.  This method assumes a spherical earth and
c         recognizes the degenerate cases of point 1 at the north
c         or south pole or points 1 and 2 coincident.  All coordinates
c         are in radians.
c
c         Parameters input: lat1, lon1, lat2, lon2
c
c         Parameters returned: range, azim
c
c         Subroutines and functions used:  none
c
c         Common blocks:  none
c
c         Method: Uses law of cosines for sides on spherical triangle
c                 defined by (lat1,lon1),(lat2,lon2) and north pole.
c
cz*********************************************************************
c
      real lat1
      real lon1
      real lat2
      real lon2
      real range
      real azim
      real s1
      real c1
      real s2
      real c2
      real cr
      real ca
c
      data pi/3.141592654/,twopi/6.283185308/,halfpi/1.570796327/,
     &     dtr/0.017453293/,rtd/57.29577951/
c
c     Test for degenerate cases
c        1)Point 1 at north or south pole:
c
      if ( abs( lat1 - halfpi ) .le. 1.0e-5 ) then
c
c     Point 1 is at the north pole
c
         range = halfpi - lat2
         azim = pi
         return
```

```
      else
          if ( abs( lat1 + halfpi ) .le. 1.0e-5 ) then
c
c     Point 1 is at the south pole
c
              range = halfpi + lat2
              azim = 0.0
              return
          end if
      end if
c
c        2)Coincident points:
c
      if ( abs( lat1 - lat2 ) .le. 1.0e-5 .and.
     &      abs( lon1 - lon2 ) .le. 1.0e-5 ) then
c
c     Points 1 and 2 are coincident
c
          range = 1.0e-8
          azim = 0.0
          return
      end if
c
c     General case
c
      s1 = sin( lat1 )
      c1 = cos( lat1 )
      s2 = sin( lat2 )
      c2 = cos( lat2 )
      cr = s1*s2 + c1*c2*cos( lon1 - lon2 )
      cr = amin1( amax1( cr, -1.0 ), +1.0 )
      range = acos( cr )
      ca = ( s2 - s1*cr )/( c1*sin( range ) )
      ca = amin1( amax1( ca, -1.0 ), +1.0 )
      azim = acos( ca )
      if ( sin( lon1 - lon2 ) .lt. 0.0 )  azim = twopi - azim
      return
      end
```

```
      function fof2( ff2, lmt, itime, lat, lon, mlat, ssn )
cp*******************************************************************
c     function fof2
c
c       x = fof2( ff2, lmt, itime, lat, lon, mlat, ssn )
c
c       This function corrects the f2-layer critical frequency
c       computed by muf85 for polar latitudes using the Chiu model.
c
c       Parameters input:
c         ff2:   critical frequency from muf35 in mhz
c         lmt:   local mean time at lat,lon in hours
c         itime: integer array containing month, day,
c                hour, minute, julian day, and year
c         lat:   geographic latitude in radians
c         lon:   geographic west longitude in radians
c         mlat:  magnetic latitude in radians
c         ssn:   sunspot number
c
c       Parameters returned:
c         fof2     the f2-layer critical frequency in mhz - real
c
c       Subroutines and functions called:  none
c
c       Common blocks referenced:          none
c
c       Method:  uses the high latitude electron density model
c                developed by B.K. Ching and Y.T. Chiu (Ching,B.K.,
c                Y.T. Chiu, J. Atmos. Terr. Phys., 35, 1615,(1973))
c                and later improved by Chiu (Chiu Y.T., J. Atmos.
c                Terr. Phys., 37, 1563, (1975)). The peak electron
c                density is then converted to foF2 and used to cor-
c                rect the value input from muf85. A folding function is
c                used in transition latitudes to provide continuity
c                of the transition.
c
cz*******************************************************************
      integer  itime(6)
      real  lat
      real  lmt
      real  lon
      real  mlat
      real  mlon
      real  phi
      real  tmo
      real  cmlat
      real  x
      real  ssn
      real  ff
      real  gg
      real  t
      real  v
      real  y
      real  ys
      real  z
      real  w
      real  plr
      real  u
```

```
      real  za
      real  am
      real  b
      real  ys4
c
      data           pi /3.1415926/
c
      phi = lmt*pi/12.0
      tmo = itime(1) + ( itime(2) + itime(3)/24.0 + itime(4)/1440.0 )/30.0
     &        - 0.5
      cmlat = cos( mlat )
      mlon = cos( lat )*sin( lon - 1.2043 )/cmlat
      mlon = amax1( amin1( mlon, 1.0 ), -1.0 )
      mlon = asin( mlon )
      x = ( 2.2 + ( 0.2 + ssn/1000.0 )*sin( mlat ) )*cmlat
      ff = exp( -( x**6 ) )
      gg = 1.0 - ff
      t = pi*tmo/12.0
      v = sin( t )
      if ( mlat .ge. 0.0 )  then
        w = exp( -1.2*( cos( mlat - 0.41015*cos( phi ) ) - cmlat ) )
        plr = ( 2.0 + 0.012*ssn )*w*( 1.0 + 0.3*v )
      else
        u = cos( t+t )
        y = sin( mlon/2.0 )
        ys = cos( mlon/2.0 - pi/20.0 )
        z = sin( mlon )
        za = sqrt( abs( z ) )
        am = 1.0 + v
        b = v*( ( y - z )/2.0 - y**8 ) - am*u*( z/za )*exp( -4.0*y*y )
        ys4 = ys**4
        plr = (2.5 + ssn/50.0 + u*( 0.5 + ( 1.3 + 0.002*ssn )*ys4 )
     &            + ( 1.3 + 0.005*ssn )*cos( phi - pi*( 1.0 + b ) ))
     &            * ( 1.0 + 0.4*( 1.0 - v*v ) )*exp( -v*ys4 )
      end if
c
      fof2 = gg*ff2*ff2/8.12 + 0.66*ff*plr
      if (fof2 .gt. 0.0)  then
       fof2 = 2.85*sqrt(fof2)
      else
       fof2 = ff2
      end if
      return
      end
```

```
      function sygn( y )
cp*********************************************************************
c
c     real function sygn
c
c     x= sygn( y )
c
c     This function returns the value 0 if y=0, -1 if y is less than 0
c     or 1 if y is greater than 0.
c
c     Subroutines and functions used: none
c
c     Common blocks: none
c
c     Parameters input: y
c
cz*********************************************************************
c
      real y
c
      if(y)100,200,300
100   sygn=-1.0
      go to 1000
200   sygn=0.0
      go to 1000
300   sygn=1.0
1000  return
      end
```