Ionospheric Correction User Algorithm using NeQuick-NavIC (NeQuick-N) Model for NavIC single frequency users

Version 1.0

Contents

List of figures

List of Tables

1 Introduction

1.1 Objective and Scope

The document describes the ionosphere correction user algorithm for single frequency NavIC users based on NeQuick-N model. This algorithm aids the single frequency users in the NavIC service area to improve the user position accuracy through the set of broadcasted coefficients. The user will use these broadcast parameters in the NeQuick-N model to obtain the ionosphere delay for his line of sight.

Note: The user algorithm of NeQuick-N is similar to Galileo's ionosphere correction user algorithm (NeQuick-G). The software implementation of the NeQuick-G model is owned by the European Union and was released to the public as open source software under EUPL license. Users are free to use it, respecting the provisions of the open source license.

1.2 Overview

NeQuick is a three-dimensional and time dependent ionospheric electron density model. It is based on an empirical climatological representation of the ionosphere, which predicts the monthly mean electron density from analytical profiles, depending on the solar activity related input (F10.7), month, geographic latitude and longitude, height and time [1].

NeQuick-N model has been adapted from the basic NeQuick model for the NavIC single frequency users in order to derive the ionosphere corrections (along the user Line of Sight-LOS) based on the Effective Ionization level (Az) [2] [3][4][5][6]. To accommodate the day-to-day variations in the ionosphere for the user, the F10.7 is replaced by Az in NeQuick-N. Az is determined from the three broadcast ionospheric coefficients (a_0, a_1, a_2) as follows:

$$Az = a_0 + a_1 \times MODIP + a_2 \times MODIP^2 \qquad \text{Eq. 1}$$

where, MODIP is the Modified Dip Latitude at the location of the user receiver and a_0, a_1, a_2 are the three broadcast coefficients within the given longitude coverage.

The detailed algorithm required to compute the ionosphere correction for the user using the derived Az is provided in the following sections of the document. The algorithm requires few input files to compute the ionosphere corrections. The information about these files are also provided in this document.

1.3 Input and Output

To evaluate the ionosphere delay along the satellite-receiver LOS, various inputs are required. The input/broadcast parameters are listed below:

Table 1 : Input Parameter description for NeQuick-N

Notation	Parameter description	Unit	Input Source
a_0	Effective Iononization level 1st order	sfu**	Broadcast
a_1	Effective Iononization level 2nd order	sfu**/degree	Broadcast

a_2	Effective Iononization level 3rd order	sfu**/degree2	Broadcast
$Modip_{max}$	Maximum MODIP coverage	Degree	Broadcast
$Modip_{min}$	Minimum MODIP coverage	Degree	Broadcast
$MLon_{max}$	Maximum Longitude coverage	Degree	Broadcast
$MLon_{min}$	Minimum Longitude coverage	Degree	Broadcast
IDF	The Ionospheric Disturbance Flag	-	Broadcast
φ_1	Geodetic latitude of the receiver	Degree	User Computed
λ_1	Geodetic longitude of the receiver	Degree	User Computed
h_1	Geodetic height of the receiver	Meter	User Computed
φ_2	Geodetic latitude of the satellite	Degree	Derived from S/c Ephemeris
λ_2	Geodetic longitude of the satellite	Degree	Derived from S/c Ephemeris
h_2	Geodetic height of the satellite	Meter	Derived from S/c Ephemeris
HOD	Hours of Day	Hours in Universal Time	Derived from Receiver Time
Mth	Month (numerical value)	Dimensionless	Derived from Receiver Time

Remarks:

1. ** Note that sfu (Solar Flux Unit) is not a SI unit but can be converted as: 1sfu=10^{-22} W/(m^2*Hz).
2. Receiver and satellite position are in WGS-84 ellipsoidal coordinates
3. DR is degree to radian conversion factor which is equal to $\pi/180$.
4. RD is the radian to degree conversion factor which is $180/\pi$.
5. 1 TEC unit (TECU) = 10^{16} electrons/m^2.

The output of the algorithm is Slant Ionosphere Delay (STEC - Slant Total Electron Content) along the LOS in electron/m^2 which can be converted to Total ionosphere delay (Tiono) in meters using following equation:

$$Tiono = \frac{40.3}{f^2} STEC \qquad \text{Eq. 2}$$

Here "f" denotes the frequency at which ionosphere delay is to be computed.

The broadcast messages and data contents for NeQuick-N ionosphere parameters are given in NavIC Signal In Space Interface for standard positioning service in L1 frequency Document (NavIC SIS ICD for L1 frequency users)

In NavIC, the Tiono is computed for L5 frequency. For the user operating on a particular frequency f_k, the Tiono must be multiplied by scale factor ($\boldsymbol{\gamma}_{\mathbf{L5k}}$) defined as:

$$\gamma_{L5k} = {f_{L5}^2}\Big/{f_k^2} \qquad \text{Eq. 3}$$

1.4 Complementary Files

1.4.1 MODIP Grid

The MODIP (μ (deg.) at user location is computed from the MODIP grid values using interpolation Sec. 2.1. These MODIP grid values are provided in the support file modipNeQN_wrapped.asc. More details related to the MODIP file is provided in Appendix B.

1.4.2 CCIR files

The CCIR coefficients are used inside NeQuick-N to compute foF2 and M(3000)F2. These coefficients are stored in the ccirxx.asc files. More details related to CCIR files are provided in Appendix A.

1.5 Short description of the NeQuick-N electron density model

NeQuick-N is a profiler that models the ionosphere at three different layers. These three layers are: E, F1 and F2 where, E layer peak at a fixed height of 120 km from the surface of the Earth [2]. To model the anchor points of these layers, the model employs ionosonde parameters in terms of critical frequencies for each layer and transmission factor. The critical frequencies are denoted as foE, foF1, and foF2 for E, F1 and F2 layers respectively, while, the transmission factor is denoted as M(3000)F2 [3] [4]. The computation of these parameters requires external inputs (CCIR files), which will be discussed in detail in Appendix A.

The model is constituted by two major components, bottom side and top side ionosphere models. A brief introduction about the both components is provided in the below sections.

1.5.1 Bottom-side and Top-side model definition

Bottom-side model: It is for the region up to the F2-layer peak (E, F1 and F2). This is a modified version of [5] based on the ionospheric characteristics of foE, foF1, foF2 and M(3000)F2 [2]. For foE derivation, a modified formulation of [6] is adopted and foF1 is selected as being equal to 1.4× foE during daytime and zero during night-time, respectively [7].

Topside model: It is for the region above the F2-layer peak. The topside of NeQuick-N is a semi-Epstein layer with a height dependent thickness parameter B through a new parameter H0, which forms the correction factor that adjusts the vertical TEC (Total Electron Content) values, considering the exosphere electron density in a simple manner [7].

2 NeQuick-N User Algorithm

To compute the NeQuick electron density, several parameters are evaluated through specific modules. In the following sections the formulation of each of these modules are provided.

2.1 Compute MODIP

Input: latitude φ (deg), longitude λ(deg), stModip

Output: MODIP μ (deg)

Read and Store MODIP Grid values into structure:

Read the values from modipNeQN_wrapped.asc into an array as shown below.

stModip: $stModip_{i,j}$

i = 0,....,38; j = -1, ..., 37;

If $\varphi = 90^o$, then $\mu = 90^o$ Eq. 4

If $\varphi = -90^o$, then $\mu = -90^o$ Eq. 5

If $-90^o < \varphi < 90^o$, then

$$l = int\left(\frac{\lambda+180}{10}\right) - 2 \quad \text{Eq. 6}$$

$$l = l + 36\ ;\ if\ l < 0 \quad \text{Eq. 7}$$

$$l = l - 36\ ;\ if\ l > 33 \quad \text{Eq. 8}$$

$$a = \frac{\varphi+90}{5} + 1 \quad \text{Eq. 9}$$

$$x = a - int(a)$$

$$i = int(a) - 2 \quad \text{Eq. 10}$$

For k = 1,4; for j = 1,4 build $z_{j,k}$ as:

$$z_{j,k} = stModip_{i+j,l+k}$$

For k = 1,4 compute

$$z_k = z_x(z_{1,k}, z_{2,k}, z_{3,k}, z_{4,k}, x) \quad \text{Eq. 11}$$

using the interpolation function described in Sec 3.2.

Finally compute:

$$b = \frac{\lambda+180}{10} \quad \text{Eq. 12}$$

$$y = b - int(b) \quad \text{Eq. 13}$$

And, using the interpolation function described in Third order Interpolation Function Zx(Z_1, Z_2, Z_3, Z_4, x) described in Sec. 3.2, calculate,

$$\mu = z_x(z_1, z_2, z_3, z_4, y) \quad \text{Eq. 14}$$

2.2 Compute effective Sunspot number

Compute the Effective Sunspot number Az_R as a function of the Effective ionization level Az

$$Az_R = \sqrt{167273 + (Az - 63.7) * 1123.6} - 408.99 \quad \text{Eq. 15}$$

Where, Az, is the effective ionization level (Sec. 2.2.1).

2.2.1 Compute effective ionization level Az

Compute the Effective ionization level Az for the given receiver location (having MODIP μ) as a function of the coefficients (a_0, a_1, a_2) broadcasted for the user's longitude coverage area in the navigation message. Note that Az is not updated with MODIP along the ray (LOS). Instead, for each ray, Az is fixed for a given MODIP.

Input: Ionospheric coefficients (a_0, a_1, a_2), MODIP μ(deg).

Output: Effective ionization level Az

If $a_0 = a_1 = a_2 = 0$,

then

Az = 63.7

else

$$Az = a_0 + a_1 * \mu + a_2 * \mu^2 \quad \text{Eq. 16}$$

If Az > 400, then Az=400 Eq. 17

If Az<0, then Az=0 Eq. 18

2.3 Compute Local Time

Compute the local time LT (in hours with decimals) of the user location.

Input: longitude λ (deg) and Hours of the Day (HOD).

Output: local time LT (Hours).

$$LT = HOD + \frac{\lambda}{15} \quad \text{Eq. 19}$$

2.4 Compute Solar Declination

Compute $\sin(\delta_{Sun})$, $\cos(\delta_{Sun})$, the sine and cosine of the solar declination

Input: month (mth), Hours of the Day (HOD) (hours, in decimal)

Output: $\sin(\delta_{Sun})$, $\cos(\delta_{Sun})$

Compute day of year at the middle of the month:

$$d_y = 30.5 * mth - 15 \quad \text{Eq. 20}$$

Compute time (days),

$$t = d_y + \frac{18-HOD}{24} \quad \text{Eq. 21}$$

Compute the arguments as,

$$a_m = (0.9856 * t - 3.289) * DR \quad \text{Eq. 22}$$

$$a_l = a_m + [1.916 * \sin(a_m) + 0.020 * sin(2 * a_m) + 282.634] * DR \quad \text{Eq. 23}$$

Finally compute sine and cosine of solar declination

$$sin(\delta_{Sun}) = 0.39782 * sin(a_l) \quad \text{Eq. 24}$$

$$cos(\delta_{Sun}) = \sqrt{1 - sin^2(\delta_{Sun})} \quad \text{Eq. 25}$$

2.5 Compute effective Solar Zenith Angle

Compute the effective solar zenith angle χ_{eff} (deg) as a function of the solar zenith angle χ (deg) and the solar zenith angle at day night transition χ_0(deg).

Input: Solar zenith angle χ (deg) and χ_0 (deg).

Output: effective solar zenith angle χ_{eff} (deg)

Set $\chi_0 = 86.23292796211615^o$

$$\chi_{eff} = NeqJoin(90.0 - 0.24 * NeqClipExp(20.0 - 0.2 * \chi), \chi, 12, \chi - \chi_0) \quad \text{Eq. 26}$$

Where, functions NeqJoin and NeqClipExp are defined in Sec. 3.3 and 3.4 respectively.

2.5.1 Compute Solar zenith angle χ for the given location

Input: latitude (φ in deg), local time (LT in hours), $\sin(\delta_{Sun})$, $\cos(\delta_{Sun})$

Output: solar zenith angle χ(deg).

Compute,

$$\cos(\chi) = \sin(\varphi * DR) * \sin(\delta_{Sun}) + \cos(\varphi * DR) * \cos(\delta_{Sun}) * cos\left(\frac{\pi}{12} * (12 - LT)\right) \quad \text{Eq. 27}$$

$$\chi = \left(atan2\left(\sqrt{1 - cos^2(\chi)}, cos(\chi)\right)\right) * RD \quad \text{Eq. 28}$$

2.6 Compute foE and NmE

To compute the E layer critical frequency foE(MHz) at a given location, in addition to the effective solar zenith angle χ_{eff}, a season dependent parameter has to be computed.

Input: latitude φ(deg), Effective Ionization Level Az, effective solar zenith angle χ_{eff} (deg), month mth.

Output: foE(MHz), NmE (10^{11} m^{-3})

Define the "seas" parameter as a function of the month of the year as follows:

If mth = 1,2,11, 12

then

seas = -1 Eq. 29

If mth = 3, 4, 9, 10

then

seas = 0 Eq. 30

If mth = 5,6,7,8

then

seas = 1 Eq. 31

Introduce the latitudinal dependence,

$$ee = NeqClipExp(0.3 * \boldsymbol{\varphi})$$ Eq. 32

$$seasp = seas * \frac{(ee-1)}{(ee+1)}$$ Eq. 33

$$foE = \sqrt{(1.112 - 0.019 * seasp)^2 * \sqrt{Az} * \left[cos\left(\chi_{eff} * DR\right)\right]^{0.6} + 0.49}$$ Eq. 34

Where, function NeqClipExp is defined in Sec.3.4.

The E layer maximum density NmE ($10^{11}m^{-3}$) as a function of foE (MHz) is computed as:

$$NmE = 0.124 * foE^2$$ Eq. 35

2.7 Compute foF1 and NmF1

Input: E layer critical frequency foE(MHz), fof2 (MHz) [Sec. 2.6]

Output: foF1(MHz), NmF1(10^{11} m^{-3})

The F1 layer critical frequency *foF1* (MHz) is computed as:

$$if\ foE \geq 2.0\ MHz\ then$$

$$foF1 = 1.4 * foE$$ Eq. 36

$$if\ foE < 2.0\ MHz\ then$$

$$foF1 = 0.0$$ Eq. 37

foF1 is reduced by 15% if too close to *foF2*

if *foF1* < 10^{-6},

then, *foF1* = 0 Eq. 38

The implementation of this calculation is given below in Eq. 39 which takes into account the need to ensure continuity and derivability over the full range of *foF1*.

$$foF1 = NeqJoin(1.4 * foE, 0, 1000.0, foE - 2)$$

$$foF1 = NeqJoin(0, foF1, 1000.0, foE - foF1)$$

$$foF1 = NeqJoin(foF1, 0.85 * foF1, 60.0, 0.85 * foF2 - foF1) \quad \text{Eq. 39}$$

where function NeqJoin is defined in Sec.3.3.

The F1 layer maximum density NmF1 (10^{11}m^{-3}) as a function of foF1 (MHz) is computed as:

if *foF1* ≤ 0 and *foE* > 2,

then,

$$\mathrm{NmF1} = 0.124 * (\mathrm{foE} + 0.5)^2;$$

else,

$$\mathrm{NmF1} = 0.124 * \mathrm{foF1}^2 \quad \text{Eq. 40}$$

2.8 Computation of foF2 and NmF2; M(3000)F2

2.8.1 Read ccirxx.*asc* values

Input:

Month(mth)

Output: F2, Fm3

Select the file name to read:

XX = mth+10

(e.g. ccir21.asc for november) and store the file content in the two arrays of coefficients:

coefficients for foF2

F2:

$f2_{i,j,k}$

i=1,2;

j=1,…,76;

k=1,…,13 Eq. 41

coefficients for M(3000)F2

Fm3:

$fm3_{i,j,k}$

i=1,2;

j=1,…,49;

k=1,…,9 Eq. 42

2.8.2 Interpolate ITU-R coefficients for Az_R

Compute Af2, the array of interpolated coefficients for foF2 and Am3, the array of interpolated coefficients for M(3000)F2

Input: F2, Fm3, Az_R

Output: AF2, Am3

Compute the array of interpolated coefficients for foF2

Af2: $af2_{j,k}$

j=1,…,76; k=1,…,13 Eq. 43

Af2 elements are calculated by linear combination of the elements of F2:

$$af2_{j,k} = f2_{1,j,k} * \left(1 - \frac{Az_R}{100}\right) + f2_{2,j,k} * \left(\frac{Az_R}{100}\right)$$

j=1,…,76; k=1,…,13 Eq. 44

Compute the array of interpolated coefficients for M(3000)F2:

Am3: $am3_{j,k}$

j=1,…, 49; k=1,…,9 Eq. 45

Am3 elements are calculated by linear combination of the elements of Fm3:

$$am3_{j,k} = fm3_{1,j,k} * \left(1 - \frac{Az_R}{100}\right) + fm3_{2,j,k} * \left(\frac{Az_R}{100}\right)$$; j=1,…, 49; k=1,…,9 Eq. 46

2.8.3 Compute Fourier Time Series For foF2 and M(3000)F2

Input: HOD (hours), arrays of interpolated ITU-R coefficients AF2, Am3

Output: CF2, Cm3, vectors of coefficients for Legendre calculation for foF2 and M(3000)F2

The vector CF2 has 76 elements:

cF2: $cf2_{l}$; i=1,…,76 Eq. 47

The vector Cm3 has 49 elements:

Cm3: $Cm3_{l}$ i=1,…,49 Eq. 48

Compute the time argument:

T = (15*HOD-180) *DR Eq. 49

For i=1,..,76, Calculate the Fourier time series for foF2:

$$cf2_i = af2_{i,1} + \sum_{k=1}^{6}\left[af2_{i,2k} * \sin(k * T) + af2_{i,2k+1} * \cos(k * T)\right] \quad \text{Eq. 50}$$

for i=1,..,49 calculate the Fourier time series for M(3000)F2:

$$cm3_i = am3_{i,1} + \sum_{k=1}^{4}\left[am3_{i,2k} \sin(k * T) + am3_{i,2k+1} * \cos(k * T)\right] \quad \text{Eq. 51}$$

2.8.4 Compute foF2 and M(3000)F2

Input: MODIP μ(deg), latitude φ(deg), longitude λ (deg), vector CF2 of the coefficients for Legendre combination for foF2, vector Cm3 of the coefficients for Legendre combination for M(3000)F2

Output: foF2 (MHz), M(3000)F2

Define vectors containing sine and cosine of the coordinates:

M: m_k k = ,…,12

P: p_n n = 2,…,9

S: s_n n = 2,…,9

C: c_n n = 2,…,9

Compute MODIP coefficients:

$m_1 = 1$

and for k=2,…,12

$$m_k = sin^{k-1}(\mu * DR) \quad \text{Eq. 52}$$

Compute latitude and longitude coefficients

for n=2,…,9

$$p_n = cos^{n-1}(\varphi * DR) \quad \text{Eq. 53}$$

$$s_n = sin\big((n-1) * \lambda * DR\big) \quad \text{Eq. 54}$$

$$c_n = cos\big((n-1) * \lambda * DR\big) \quad \text{Eq. 55}$$

2.8.4.1 Compute foF2

Order 0 term:

$$foF2_1 = \sum_{k=1}^{12} cf2_k m_k \quad \text{Eq. 56}$$

Having the increased Legendre grades for foF2 in a vector:

Q: qn; n = 1, ..., 9 Eq. 57

Q = (12, 12, 9, 5, 2, 1, 1, 1, 1) Eq. 58

For computational efficiency, define also:

K: k_n; n = 1,...,9 Eq. 59

$$k_1 = -q_1$$ Eq. 60

and for n=2,…,9

$$k_n = k_{n-1} + 2q_{n-1}$$ Eq. 61

for n=2,…,9 compute the higher order terms:

$$foF2_n = \sum_{k=1}^{q_n}\left(cf2_{k_n+2k-1}c_n + cf2_{k_n+2k}s_n\right) * m_k * p_n$$ Eq. 62

Finally sum the terms to obtain foF2:

$$foF2 = \sum_{n=1}^{9} foF2_n$$ Eq. 63

2.8.4.2 Compute M(3000)F2

0 order term:

$$M(3000)F2_1 = \sum_{k=1}^{7} cm3_k m_k$$ Eq. 64

Having the increased Legendre grades for M(3000)F2 in a vector:

R: r_n n = 1,…,7 Eq. 65

R = (7,8,6,3,2,1,1) Eq. 66

For computational efficiency, define also:

H: h_n n = 1,…,7 Eq. 67

$$h_1 = -r_1$$ Eq. 68

And for n = 2, ..., 7

$$h_n = h_{n-1} + 2r_{n-1}$$ Eq. 69

for n=2,…7, compute the higher order terms

$$M(3000)F2_n = \sum_{k=1}^{r_n}\left(cm3_{h_n+2k-1} * c_n + cm3_{h_n+2k} * s_n\right) * m_k * p_n$$ Eq. 70

finally sum the terms:

$$M(3000) = \sum_{n=1}^{7} M(3000)F2_n$$ Eq. 71

To compute NmF2, use

$$NmF2 = 0.124 * foF2^2$$ Eq. 72

where NmF2 is in $10^{11}m^{-3}$.

2.8.5 Compute hmF1

Compute the F1 layer maximum density height hmF1 (km):

Input: hmF2 (km), hmE(km)

Output: hmF1 (km)

$$\mathrm{hmF1} = (\mathrm{hmF2} + \mathrm{hmE})/2$$ Eq. 73

2.8.6 Compute hmE

The E layer maximum density height hmE(km) is defined as a constant:

$$\mathrm{hmE} = 120 \quad \text{Eq. 74}$$

2.8.7 Compute hmF2

Compute the F2 layer maximum density height hmF2 (km).

Input: foE(mhz), foF2 (mhz), M(3000)F2

output: hmF2 (km)

$$hmF2 = \frac{1490 * M * \sqrt{\frac{0.0196 * M^2 + 1}{1.2967 * M^2 - 1}}}{(M + \Delta M)} - 176 \quad \text{Eq. 75}$$

Where,

$$\mathrm{M} = \mathrm{M(3000)F2} \quad \text{Eq. 76}$$

If foE<10^{-30}, then $\Delta M = -0.012$ Eq. 77

$$if\ foE \geq 10^{-30}, \text{then}\ \ \Delta M = \frac{0.253}{\rho - 1.215} - 0.012 \quad \text{Eq. 78}$$

and the ratio ρ is computed as:

$$\rho = \frac{\frac{foF2}{foE} * exp\left(20 * \left(\frac{foF2}{foE} - 1.75\right)\right) + 1.75}{exp\left(20 * \left(\frac{foF2}{foE} - 1.75\right)\right) + 1} \quad \text{Eq. 79}$$

2.9 Compute B2bot, B1top, B1bot, BEtop, BEbot

Input	Output	Formulation	Eq. No.
NmF2(10^{11} m^{-3}) foF2(MHz) M(3000)F2	B2bot(km)	$B2bot = \frac{0.385 * NmF2}{0.01 * exp(-3.467 + 0.857 * \ln(foF2^2) + 2.02 * \ln(M))}$ Where, M = M(3000)F2	Eq. 80
hmF1(km) hmF2(km)	B1top(km)	$B1top = 0.3 * (hmF2 - hmF1)$	Eq. 81
hmF1(km) hmE(km)	B1bot(km)	$B1bot = 0.5 * (hmF1 - hmE)$	Eq. 82
B1bot(km)	BEtop(km)	$BEtop = max[B1bot, 7]$	Eq. 83
	BEbot(km)	BEbot = 5	Eq. 84

2.10 Compute A1

Compute the F2 layer amplitude A1 (10^{11} m^{-3}).

Input: NmF2 (10^{11} m^{-3})

Output: A1 (10^{11} m^{-3})

$$A1 = 4 * NmF2 \quad \text{Eq. 85}$$

2.11 Compute A2 and A3

Compute the F1 layer amplitude A2 (10^{11} m^{-3}) and the E layer amplitude A3 (10^{11} m^{-3}).

Input: NmE(10^{11} m^{-3}), NmF1 (10^{11} m^{-3}), A1(10^{11} m^{-3}), hmF2 (km), hmF1 (km), hmE(km), BEtop(km), B1bot(km), B2bot(km), foF1 (MHz)

Output: A2 (10^{11} m^{-3}), A3 (10^{11} m^{-3})

If foF1 < 0.5, then, A2 =0 Eq. 86

$$A3 = 4.0 * \left(NmE - Epst(A1, hmF2, B2bot, hmE)\right)$$

If foF1 ≥ 0.5:

$$A3a = 4.0 * NmE \quad \text{Eq. 87}$$

Repeat 5 times the iterations below:

$$A2a = 4.0\ [NmF1 - Epst(A1, hmF2, B2bot, hmF1) - Epst(A3a, hmE, BEtop, hmF1)]$$

Eq. 88

$$A2a = \frac{A2a * \exp(A2a - 0.80 * NmF1) + 0.80 * NmF1}{1 + \exp(A2a - 0.80 * NmF1)} \quad \text{Eq. 89}$$

Alternatively, A2a can be written as

$$A2a = NeqJoin(A2a, 0.8 * NmF1, 1, A2a - 0.8 * NmF1)$$

$$A3a = 4.0 * (NmE - Epst(A2a, hmF1, B1bot, hmE) - Epst(A1, hmF2, B2bot, hmE))$$

Eq. 90

where the function Epst() is defined in Sec.3.1. Then compute,

$$A2 = A2a \quad \text{Eq. 91}$$

$$A3 = NeqJoin(A3a, 0.05, 60.0, A3a - 0.005) \quad \text{Eq. 92}$$

where, the functions Epst and NeqJoin are defined in Sec. 3.1 and Sec. 3.3 respectively.

2.11.1 Compute H0

Compute the topside thickness parameter H0(km).

Input: B2bot (km), Shape Parameter (K)

Output: *H0*(km)

First compute the auxiliary parameter Ha:

$$Ha = K * B2bot$$ Eq. 93

Compute the auxiliary parameters x and v as follows:

$$x = (Ha - 150.0)/100$$ Eq. 94

$$v = (0.041163 * x - 0.183981) * x + 1.424472$$ Eq. 95

Then compute

$$H0 = Ha/v$$ Eq. 96

2.11.2 Compute Shape Parameter K

Compute the shape parameter K.

Input: mth, NmF2 (10^{11} m^{-3}), hmF2 (km), B2bot(km), Az_r

Output: K

First, compute the auxiliary parameter ka:

If mth =4, 5, 6, 7, 8, 9, then

$$\text{ka} = 6.705 - 0.014 * \text{AzR} - 0.008 * \text{hmF2}$$ Eq. 97

if mth =1,2,3,10,11,12

$$ka = -7.77 + 0.097 * \left(\frac{hmF2}{B2bot}\right)^2 + 0.153 * NmF2$$ Eq. 98

compute the auxiliary parameter kb:

$$kb = NeqJoin(ka, 2, 1, ka - 2.0)$$ Eq. 99

Then compute:

$$K = NeqJoin(8, kb, 1, kb - 8.0)$$ Eq. 100

Where, the function NeqJoin is defined in Sec. 3.3

2.12 Electron Density Computation

To compute the electron density N = N(φ, λ, h, a_0, a_1, and a_2, mth, HOD) for a given user location (identified by the coordinates φ, λ, h) at a given time (mth, HOD) and using a given set of broadcast NeQuick-N coefficients for user's longitude coverage area, all NeQuick-N parameters have to be compute for the user location. The selection of the a_0, a_1, and a_2 for the given user location is carried out based on the maximum and minimum Longitude coverage given in the NavIC message for NeQuick-N. Then, Effective ionisation level (Az) is derived with the broadcast NeQuick-N coefficients (a_0, a_1, a_2) and the MODIP(μ) at the receiver location. Nevertheless 2 different modules have to be used accordingly to the height considered. In particular, if h ≤ hmF2, the bottom-side electron density as described in Sec.2.12.1, should be computed and if h > hmF2, the topside electron density has to be computed as described in Sec.2.12.2.

2.12.1 Compute Bottom-side Electron Density

Compute the electron density N of the bottom-side (Case: h≤ hmF2).

Input: Height h(Km), A1 (10^{11} m^{-3}), A2 (10^{11} m^{-3}), A3 (10^{11} m^{-3}), hmF2 (km), hmF1 (km), hmE(km), B2bot(km), B1top(km), B1bot(km), BEtop(km), BEbot(km).

Output: (bottom-side) electron density N(m^{-3}).

Select the relevant B parameters for the current height:

$$BE = \begin{cases} BEtop & if\ h > hmE \\ BEbot & if\ h \le hmE \end{cases} \quad \text{Eq. 101}$$

$$BF1 = \begin{cases} B1top & if\ h > hmF1 \\ B1bot & if\ h \le hmF1 \end{cases} \quad \text{Eq. 102}$$

Compute the exponential arguments for each layer:

$$\alpha_1 = \frac{h-hmF2}{B2bot} \quad \text{Eq. 103}$$

$$\alpha_2 = \frac{h-hmF1}{BF1} * exp\left(\frac{10}{1+|h-hmF2|}\right) \quad \text{Eq. 104}$$

$$\alpha_3 = \frac{h-hmE}{BE} * exp\left(\frac{10}{1+|h-hmF2|}\right) \quad \text{Eq. 105}$$

Note: If h <100 km, use the value h = 100 km in above equations.

For i =1 to 3, Compute:

$$s_i = \begin{cases} 0 & if\ |\alpha_i| > 25 \\ A_i * \frac{\exp(\alpha_i)}{(1+\exp(\alpha_i))^2} & if\ |\alpha_i| \le 25 \end{cases} \quad \text{Eq. 106}$$

If h < 100 km compute also the corrective terms:

$$ds_1 = \begin{cases} 0 & if\ |\alpha_1| > 25 \\ \frac{1}{B2bot} * \frac{1-\exp(\alpha_1)}{1+\exp(\alpha_1)} & if\ |\alpha_1| \le 25 \end{cases} \quad \text{Eq. 107}$$

$$ds_2 = \begin{cases} 0 & if\ |\alpha_2| > 25 \\ \frac{1}{BF_1} * \frac{1-\exp(\alpha_2)}{1+\exp(\alpha_2)} & if\ |\alpha_2| \le 25 \end{cases} \quad \text{Eq. 108}$$

$$ds_3 = \begin{cases} 0 & if\ |a_3| > 25 \\ \frac{1}{BE} * \frac{1-\exp(a_3)}{1-\exp(a_3)} & if\ |a_3| \le 25 \end{cases} \quad \text{Eq. 109}$$

and the Chapman parameters:

$$BC = 1 - 10 * \frac{\sum_{i=1}^{3} s_i * ds_i}{\sum_{i=1}^{3} s_i} \quad \text{Eq. 110}$$

$$z = \frac{h-100}{10}$$ Eq. 111

Then compute the electron density as:

$$N = (s_1 + s_2 + s_3) * \exp(1 - BC * z - \exp(-z)) * 10^{11}$$ Eq. 112

If h ≥ 100 km compute the electron density as:

$$N = (s_1 + s_2 + s_3) * 10^{11}$$ Eq. 113

2.12.2 Compute the topside electron density

Compute the electron density N of the topside (case h> hmF2).

Input: height h (km), NmF2 (10^{11}m^{-3}), hmF2 (km), H0(km)

Output: (topside) electron density N(m^{-3})

Define the constant parameters g and r as:

g =0.125 Eq. 114

r =100 Eq. 115

Compute the arguments Δh and Z as:

$$\Delta h = h - hmF2$$ Eq. 116

$$Z = \frac{\Delta h}{H_0 * \left[1 + \frac{r * g * \Delta h}{r * H_0 + g * \Delta h}\right]}$$ Eq. 117

Compute the exponential:

$$e_a = NeqClipExp(Z)$$ Eq. 118

Then compute:

$$N = \begin{cases} \frac{4 * NmF2}{e_a} \times 10^{11} & if\ e_a > 10^{11} \\ 4 * NmF2 * \frac{e_a}{(1+e_a)^2} \times 10^{11} & if\ e_a \leq 10^{11} \end{cases}$$ Eq. 119

2.13 TEC Calculation

To compute the slant TEC along a straight line between a point P_1 and a point P_2, the NeQuick-N electron density N has to be evaluated on a point P defined by the coordinates {φ, λ, h} along the ray-path. It is a choice depending on receiver computation capabilities to identify the number of points where N is to be evaluated, in order to obtain a sufficient accuracy for a subsequent integration, leading to slant TEC. This may be driven directly by the integration routine. For computational efficiency, if the latitude and the longitude of P_1 and P_2 are close to each other (if ray perigee radius r_p < 0.1 km), the vertical integration algorithm has to be used, as described in Sec.2.13.1; otherwise, the slant integration algorithm to be used is the one described in section Sec. 2.13.2. The ray perigee radius (r_p) can be computed using Sec. 2.13.2.3. When performing the TEC computation, the electron density at

the point P has to be evaluated as indicated in Sec. 2.12, while the calculation of the coordinates of the point P along the ray-path is described in Sec. 2.13.1.1, in the case a vertical ray-path is considered, and in Sec. 2.13.2.5, if a slant ray-path is considered.

2.13.1 Vertical TEC Calculation

To compute NeQuick-N vertical TEC, first compute all profile parameters hmE, hmF1, hmF2, A1, A2, A3, B2bot, B1top B1bot, BEtop, BEbot, NmF2, H0, then compute the integration of the electron density (bottom-side or topside) as function of height:

$$TEC = \int_{h1}^{h2} N(h)dh \qquad \text{Eq. 120}$$

where,

$$h_1 = r_1 - R_{EM} \qquad \text{Eq. 121}$$

$$h_2 = r_2 - R_{EM} \qquad \text{Eq. 122}$$

where r_1 and r_2 are the range of the points P_1 and P_2 from the centre of the Earth, respectively. The earth is assumed to be a sphere with a radius (R_{EM}) of 6371.2 km.

2.13.1.1 Vertical TEC Numerical Integration

Input:

- Integration endpoints: h_1(km), h_2(km)
- Model parameters: A1 ($10^{11}m^{-3}$), A2 ($10^{11}m^{-3}$), A3 ($10^{11}m^{-3}$), hmf2 (km), hmf1 (km), hmE(km), B2bot(km), B1top(km), B1bot(km), BEtop(km), BEbot(km), NmF2 (m-3), H0(km)

Output: TEC (TECU)

As φ, λ, and all model parameters are fixed during the integration, the following a simplified notation is used:

$$N(h) = \begin{cases} bottomside\ N & if\ h \le hmF2 \\ topside\ N & if\ h > hmF2 \end{cases} \qquad \text{Eq. 123}$$

N(h) is computed using the algorithms described in Sec.2.12

For integration, Gauss algorithm is followed (Sec. 2.13.1.2). User can use Gauss-Kronrod (G7-K15) Adaptive Quadrature Method as an alternate. The G7-K15 algorithm is explained in Sec. 3.5.

2.13.1.2 Gauss Algorithm

Start the calculation using 8 points (n =8).

Repeat the following computations until the target integration accuracy (ϵ) is obtained (default tolerance (ϵ) values are 0.001 below 1000 km and 0.01 above 1000 km. Increasing tolerance increases the integration speed at the expense of accuracy).

Calculate the integration intervals:

$$\Delta_n = \frac{h_2 - h_1}{n} \quad \text{Eq. 124}$$

$$g = 0.5773502691896 * \Delta_n \quad \text{Eq. 125}$$

$$y = g_1 + \frac{\Delta_n - g}{2} \quad \text{Eq. 126}$$

$$GN_2 = \frac{\Delta_n}{2} . \sum_{i=0}^{n-1} [N(y + i\Delta_n) + N(y + i\Delta_n + g)] \quad \text{Eq. 127}$$

Double the number of points, i.e., n = 2n and define $GN_1 = GN_2$

By repeating the steps above, it is now possible to compare the two values obtained to see if the target integration accuracy ϵ is achieved:

If $|GN_1 - GN_2| > \varepsilon |GN_1|$, Then continue increasing the number of points, redefine GN_1 and repeat again; Else, the required accuracy has been reached, and the value of the integral is obtained by:

$$TEC = \left(GN_2 + \frac{GN_2 - GN_1}{15}\right) \times 10^{-13} \quad \text{Eq. 128}$$

2.13.2 Slant TEC Calculation

To compute the electron density at a point P along the slant ray-path defined by the points P_1 and P_2, the following specific geometrical configuration is considered.

2.13.2.1 Geometrical Configuration

To simplify the formulation, we assume that if α is an angle in (deg), α̃ is the same angle in (rad):

$$\tilde{\alpha} = \alpha * DR \quad \text{Eq. 129}$$

2.13.2.2 Zenith Angle Computation

Input: Location of point P_1 (φ_1, λ_1) and P_2(φ_2, λ_2)

Output: Zenith angle

Figure 1 indicates the geometry involved in the computation of the zenith angle ζ at P_1.

Calculate:

$$\cos(\tilde{\delta}) = \sin(\tilde{\varphi}_1) * \sin(\tilde{\varphi}_2) + \cos(\tilde{\varphi}_1) * \cos(\tilde{\varphi}_2) * \cos(\tilde{\lambda}_2 - \tilde{\lambda}_1) \quad \text{Eq. 130}$$

$$\sin(\tilde{\delta}) = \sqrt{1 - cos^2(\tilde{\delta})} \quad \text{Eq. 131}$$

$$\tilde{\zeta} = atan2\left(\sin(\tilde{\delta}), \cos(\tilde{\delta}) - \frac{r_1}{r_2}\right) \quad \text{Eq. 132}$$

δ is the earth angle on the great circle connecting the receiver (P_1) and the satellite (P_2). The symbol atan2(y, x) indicates the function that computes the arctangent of y/x with a range of (-π,π)

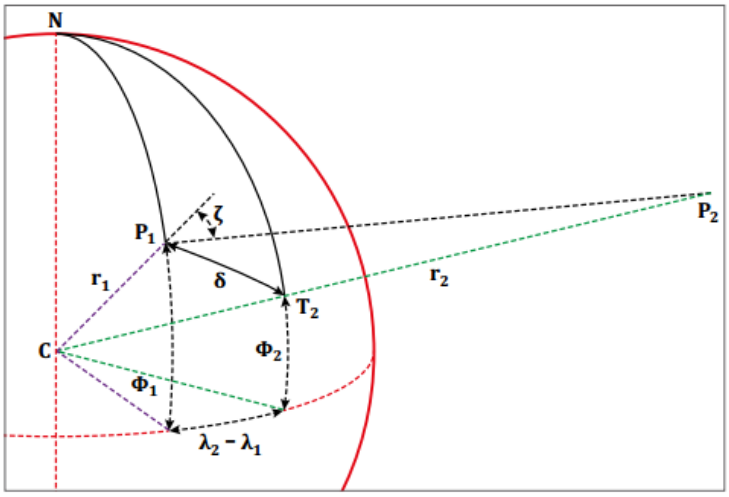

Figure 1 Geometry of zenith angle computation

2.13.2.3 RAY-PERIGEE COMPUTATION

Input: Zenith angle (ζ), location of receiver $P_1(\varphi_1, \lambda_1)$ and satellite $P_2(\varphi_2, \lambda_2)$, the earth angle on the great circle connecting the receiver (P_1) and the satellite (P_2) i.e., δ.

Output: ray perigee properties (r_p, φ_p, λ_p)

Figure 2 indicates the geometry involved in the computation of the coordinates of the ray-perigee P_P: ray perigee radius r_p(km), ray perigee latitude φ_p(deg) and ray perigee longitude λ_p(deg).

Calculate r_p:

$$r_p = r_1 * \sin(\tilde{\zeta}) \quad \text{Eq. 133}$$

calculate ϕ_p:

if $\left| |\varphi_1| - 90° \right| < 10^{-10}$ use

$$\varphi_p = \begin{cases} \zeta \ \ if\ \varphi_1 > 0 \\ -\zeta \ \ if\ \varphi_1 < 0 \end{cases} \quad \text{Eq. 134}$$

$$\tilde{\lambda}_p = \begin{cases} \tilde{\lambda}_2 + \pi \ \ if\ \tilde{\varsigma} \geq 0 \\ \tilde{\lambda}_2 \ \ if\ \tilde{\varsigma} < 0 \end{cases} \quad \text{Eq. 135}$$

Other wise use,

$$sin(\tilde{\sigma}) = \frac{\sin(\tilde{\lambda}_2 - \tilde{\lambda}_1) * \cos(\tilde{\varphi}_2)}{\sin(\tilde{\delta})} \quad \text{Eq. 136}$$

$$cos(\tilde{\sigma}) = \frac{\sin(\tilde{\varphi}_2) - \cos(\tilde{\delta}) * sin(\tilde{\varphi}_1)}{\sin(\tilde{\delta}) * cos(\tilde{\varphi}_1)} \quad \text{Eq. 137}$$

$$\tilde{\delta}_p = \frac{\pi}{2} - \tilde{\zeta} \quad \text{Eq. 138}$$

$$sin(\tilde{\varphi}_p) = sin(\tilde{\varphi}_1) * cos(\tilde{\delta}_p) - cos(\tilde{\varphi}_1) * sin(\tilde{\delta}_p) * \cos(\tilde{\sigma}) \quad \text{Eq. 139}$$

$$cos(\tilde{\varphi}_p) = \sqrt{1 - \sin^2(\tilde{\varphi}_p)} \quad \text{Eq. 140}$$

$$\tilde{\varphi}_p = atan2\left(\sin\left(\tilde{\varphi}_p\right), \cos\left(\tilde{\varphi}_p\right)\right) \quad \text{Eq. 141}$$

$$sin\left(\tilde{\lambda}_p - \tilde{\lambda}_1\right) = -\frac{\sin(\tilde{\sigma}) * \sin\left(\tilde{\delta}_p\right)}{\cos\left(\tilde{\varphi}_p\right)} \quad \text{Eq. 142}$$

$$cos\left(\tilde{\lambda}_p - \tilde{\lambda}_1\right) = \frac{\cos\left(\tilde{\delta}_p\right) - \sin(\tilde{\varphi}_1) * sin(\tilde{\varphi}_p)}{\cos(\tilde{\varphi}_1) * cos(\tilde{\varphi}_p)} \quad \text{Eq. 143}$$

$$\lambda_p = \left[atan2\left(sin\left(\tilde{\lambda}_p - \tilde{\lambda}_1\right), cos\left(\tilde{\lambda}_p - \tilde{\lambda}_1\right)\right) + \tilde{\lambda}_1\right] * RD \quad \text{Eq. 144}$$

σ is the azimuth of P_2 seen from P_1 and δp the earth angle between P_1 and the ray-perigee P_P.

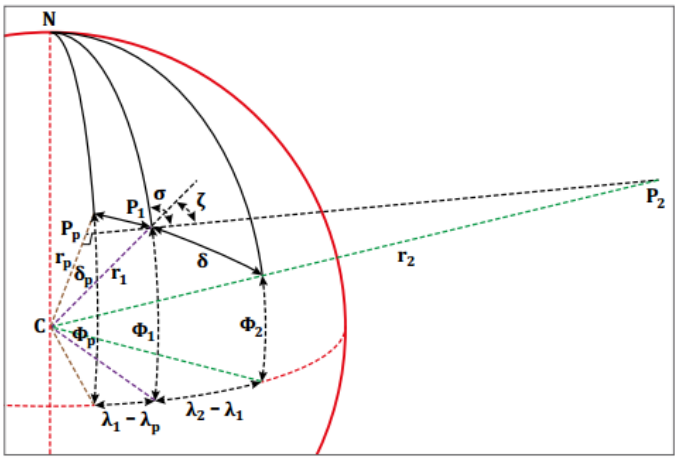

Figure 2 Geometry of ray perigee computation

2.13.2.4 GREAT CIRCLE PROPERTIES

Compute the great circle angle ψ from ray-perigee to satellite and sine and cosine of azimuth (σ) of satellite as seen from ray-perigee P_p. It is used for slanted rays only.

Input: ray perigee location (φ_p, λ_p), and satellite's location(φ_2,λ_2)

Output: ψ, sin(σ_p), cos(σ_p)

if $\left|\,|\varphi_p| - 90°\right| < 10^{-10}$, then,

$$\psi = |\varphi_2 - \varphi_2| \quad \text{Eq. 145}$$

$$sin\left(\tilde{\sigma}_p\right) = 0 \quad \text{Eq. 146}$$

$$\cos\left(\tilde{\sigma}_p\right) = \begin{cases} -1 \;\; if \;\; \tilde{\varphi}_p > 0 \\ 1 \;\; if \;\; \tilde{\varphi}_p < 0 \end{cases} \quad \text{Eq. 147}$$

Else,

$$cos\left(\tilde{\psi}\right) = \sin\left(\tilde{\varphi}_p\right) * \sin(\tilde{\varphi}_2) + \cos\left(\tilde{\varphi}_p\right) * \cos(\tilde{\varphi}_2) * \cos\left(\tilde{\lambda}_2 - \tilde{\lambda}_p\right) \quad \text{Eq. 148}$$

$$\sin\left(\tilde{\psi}\right) = \sqrt{1 - cos^2(\tilde{\psi})} \quad \text{Eq. 149}$$

$$\tilde{\psi} = atan2\left(\sin\left(\tilde{\psi}\right), \cos\left(\tilde{\psi}\right)\right) \quad \text{Eq. 150}$$

$$\sin(\tilde{\sigma}_p) = \frac{\cos(\tilde{\varphi}_2)*\sin(\tilde{\lambda}_2-\tilde{\lambda}_p)}{\sin(\tilde{\psi})} \quad \text{Eq. 151}$$

$$\cos(\tilde{\sigma}_p) = \frac{\sin(\tilde{\varphi}_2)-\sin(\tilde{\varphi}_p)*\cos(\tilde{\psi})}{\cos(\tilde{\varphi}_p)*\sin(\tilde{\psi})} \quad \text{Eq. 152}$$

Let s_1 and s_2 be the distances of P_1 and P_2 respectively from the ray perigee, compute:

$$s_1 = \sqrt{r_1^2 - r_p^2} \quad \text{Eq. 153}$$

$$s_2 = \sqrt{r_2^2 - r_p^2} \quad \text{Eq. 154}$$

2.13.2.5 Coordinates Along the Integration Path: CIP(φ_S, Λ_s, h_S)

With s(km) being the distance of a point P from the ray perigee P_P, the ray perigee coordinates (r_p, φ_p, λ_p) and the sine and cosine of the azimuth of the satellite ($\sin(\sigma_p)$, $\cos(\sigma_p)$) as seen from the ray-perigee, the coordinates of the point P are calculated by the function "CIP" as follows.

Input: Distance s (km), ray perigee coordinates (r_p, φ_p, λ_p), sine and cosine of azimuth of satellite as seen from ray-perigee ($\sin(\tilde{\sigma_p})$, $\cos(\tilde{\sigma_p})$)

Output: coordinates of point P: h_s (km), φ_s (deg), λ_s (deg)

To compute the geocentric coordinates of any point P (having distance s from the ray perigee P_P) along the integration path, the following formulae have to be applied:

Calculate h_s:

$$h_s = \sqrt{s^2 + r_p^2} - R_{EM} \quad \text{Eq. 155}$$

Where R_{EM} is the Earth's mean radius.

Calculate the great circle parameters:

$$\tan(\tilde{\delta}_s) = \frac{s}{r_p} \quad \text{Eq. 156}$$

$$\cos(\tilde{\delta}_s) = \frac{1}{\sqrt{1+tan^2(\tilde{\delta}_s)}} \quad \text{Eq. 157}$$

$$\sin(\tilde{\delta}_s) = \tan(\tilde{\delta}_s) * \cos(\tilde{\delta}_s) \quad \text{Eq. 158}$$

Calculate φ_s:

$$sin(\tilde{\varphi}_s) = \sin(\tilde{\varphi}_p) * \cos(\tilde{\delta}_s) + \cos(\tilde{\varphi}_p) * \sin(\tilde{\delta}_s) * \cos(\tilde{\sigma}_p) \quad \text{Eq. 159}$$

$$cos(\tilde{\varphi}_s) = \sqrt{1 - sin^2(\tilde{\varphi}_s)} \quad \text{Eq. 160}$$

$$\varphi_s = atan2(sin(\tilde{\varphi}_s), cos(\tilde{\varphi}_s)) * RD \quad \text{Eq. 161}$$

Calculate λ_s:

$$sin(\tilde{\lambda}_s - \tilde{\lambda}_p) = \sin(\tilde{\delta}_s) * sin(\tilde{\sigma}_p) * \cos(\tilde{\varphi}_p) \quad \text{Eq. 162}$$

$$cos(\tilde{\lambda}_s - \tilde{\lambda}_p) = \cos(\tilde{\delta}_s) - \sin(\tilde{\varphi}_p) * sin(\tilde{\varphi}_s) \quad \text{Eq. 163}$$

$$\lambda_s = \left[atan2\left(sin(\tilde{\lambda}_s - \tilde{\lambda}_p), cos(\tilde{\lambda}_s - \tilde{\lambda}_p)\right) + \tilde{\lambda}_p\right] * RD \quad \text{Eq. 164}$$

2.13.2.6 Slant TEC Numerical Integration

To compute slant TEC along a ray-path (straight line passing from 2 points) defined by its perigee coordinates, direction and end-point, a numerical integration algorithm is used.

Input: h_1, height of point P_1(km)

φ_1, latitude of point P_1(deg)

λ_1, longitude of point P_1 (deg)

h_2, height of point P_2(km)

φ_2, latitude of point P_2(deg)

λ_2, longitude of point P_2(deg)

Broadcast coefficients: a_0, a_1, a_2

month mth

HOD (UT(hours))

Output: Slant TEC (TECU)

In the case of integration from ground to satellite (h1<1000 km and h2>2000 km), the integration path is divided into three parts defining intermediate points s_a, s_b:

$$s_a = \sqrt{(R_{EM} + 1000)^2 - r_p^2} \quad \text{Eq. 165}$$

$$s_b = \sqrt{(R_{EM} + 2000)^2 - r_p^2} \quad \text{Eq. 166}$$

We have that $(R_{EM} + 1000)^2 = 54334589.44$ and $(R_{EM} + 2000)^2 = 70076989.44$.

The slant TEC becomes therefore:

$$TEC = \int_{s_1}^{s_a} N(s)ds + \int_{s_a}^{s_b} N(s)ds + \int_{s_b}^{s_2} N(s)ds \quad \text{Eq. 167}$$

The above equation involves integration of the electron density N(s) along the ray path. There are two algorithms proposed for integration:

i. Gauss Algorithm
ii. Kronrod G7-K15 Adaptive Quadrature Method (Sec. 3.5)

User can use any of the integration algorithms.

To be able to compute NeQuick-N electron density, in all the following computations, it is necessary to calculate the coordinates of the point P along the ray-path using the algorithm illustrated in Sec. 2.13.2.5.

$$(h(s), \varphi(s), \lambda(s)) = CIP\left(s,\ sin(\tilde{\varphi}_p), cos(\tilde{\varphi}_p), sin(\tilde{\sigma}_p), cos(\tilde{\sigma}_p), \lambda_p\right) \quad \text{Eq. 168}$$

Here function CIP is defined in Sec. 2.13.2.5.

To compute each integral, the algorithm described in Sec. 2.13.3 can be used as:

$$\int_{g_1}^{g_2} N(s)ds = GN\left(g_{1,}g_2, \varepsilon, r_p, \sin\left(\tilde{\varphi}_p\right), \cos\left(\tilde{\varphi}_p\right), \sin\left(\tilde{\sigma}_p\right), \cos\left(\tilde{\sigma}_p\right), \lambda_p, a_0, a_1, a_2, mth, HOD\right)$$

Eq. 169

where, the parameter ϵ indicates the integration accuracy. Here, ϵ =0.001 for the integration between s_1 and s_a and ϵ =0.01 for the integrations between s_a and s_b and between s_b and s_2.

GN represents function to compute integration using Gauss algorithm (GNg) (Sec. 2.13.3) or Kronrod G7-K15 Adaptive Quadrature Method (GNk) (Sec 3.5). g_1 and g_2 are the distances from the ray perigee of the first and second integration end point and with a_0, a_1, a_2, mth, HOD fixed during the integration, in the following a simplified notation is used:

$$f(s) := N\left(h(s), \varphi(s), \lambda(s), a_{0,}a_1, a_2, mth, HOD\right)$$

2.13.3 Gauss Algorithm (GNg)

Start the calculation using 8 points, n = 8 and repeat the following computations until the target integration accuracy (ϵ) is obtained as follows:

calculate the integration intervals:

$$\Delta_n = \frac{g_2 - g_1}{n}$$ Eq. 170

$$g = 0.5773502691896 * \Delta_n$$ Eq. 171

$$y = g_1 + \frac{\Delta_n - g}{2}$$ Eq. 172

$$GN_2 = \frac{\Delta_n}{2} * \sum_{i=0}^{n-1}[f(y + i * \Delta_n) + f(y + i * \Delta_n + g)]$$ Eq. 173

Double the number of points, n =2n and define:

$GN_1 = GN_2$ Eq. 174

Repeating steps above it is now possible to compare the two values obtained to see if the target accuracy is achieved:

If $|GN_1 - GN_2| > \varepsilon |GN_1|$, then increase the number of points, redefine GN_1 and repeat again the steps above; Else, the required accuracy has been reached, and the value of the integral is obtained by:

$$TEC = \left(GN_2 + \frac{GN_2 - GN_1}{15}\right) \times 10^{-13} \text{ TECU}$$ Eq. 175

3 Auxiliary Routines

3.1 The Epstein Function (Epst(x,y,z,w))

The Epstein function is used as a basis analytical function in NeQuick-N for the construction of the ionospheric layers and its analytical expression is given by:

$$Epst(X,Y,Z,W) = \frac{X\ exp\left(\frac{W-Y}{Z}\right)}{\left(1+exp\left(\frac{W-Y}{Z}\right)\right)^2} \quad \text{Eq. 176}$$

where,

X denotes peak amplitude,

Y denotes peak height,

Z describes thickness around the peak and

W is the height dependent variable.

3.2 Third Order Interpolation Function Z_x (Z_1, Z_2, Z_3, Z_4, X)

Let P1 = (-1, z_1), P2 = (0, z_2), P3 = (1, z_3), P4 = (2, z_4). If P = (x, z_x), to compute the interpolated value z_x at the position x, with xϵ[0,1], the following algorithm is applied.

Input: z_1, z_2, z_3, z_4, x

Output: z_x

If $|2x| < 10^{-10}$, then $z_x = z_2$ Eq. 177

Else, compute:

$\delta = 2x - 1$ Eq. 178

$g_1 = z_3 + z_2$ Eq. 179

$g_2 = z_3 - z_2$ Eq. 180

$g_3 = z_4 + z_1$ Eq. 181

$g_4 = \frac{z_4 - z_1}{3}$ Eq. 182

$a_0 = (9 * g_1) - g_3$ Eq. 183

$a_1 = (9 * g_2) - g_4$ Eq. 184

$a_2 = g_3 - g_1$ Eq. 185

$a_3 = g_4 - g_2$ Eq. 186

$z_x = \frac{1}{16} * (a_0 + a_1 * \delta + a_2 * \delta^2 + a_3 * \delta^3)$ Eq. 187

3.3 Function for joining of functions f1 and f2 (NeqJoin (f1, f2, alpha, x))

This function allows smooth joining of functions f1 and f2 ((i.e. continuous first derivatives) at origin. alpha determines width of transition region. It calculates value of joined functions at x. It uses the function NeqClipExp which is defined in Sec.3.4 .

$ee = NeqClipExp(alpha * x)$ Eq. 188

$NeqJoin = {(f1 * ee + f2)}/{(ee + 1)}$ Eq. 189

3.4 The clipped exponential function (*NeqClipExp(x)*)

It is a clipped exponential function which always returns valid output.

$$NeqClipExp = \begin{Bmatrix} 5.5406 * 10^{34} & if\ x > 80 \\ \exp(x) & if\ 80 \geq x \geq -80 \\ 1.8049 * 10^{-35} & if\ x < -80 \end{Bmatrix} \quad \text{Eq. 190}$$

3.5 Kronrod G7-K15 Adaptive Quadrature Method (GNk)

An alternative more computationally efficient integration method for calculating TEC along rays based on kronrod G7 -k15 adaptive quadrature method can also be used for integration. This method involves sampling values at 15 points and calculating the integration from them. At the same time it misses out half of the points to see what difference it makes and therefore the likely error contained in the result, before deciding whether to accept the result, or to split the portion into two and try again in order to improve accuracy. Note that this method is recursive but has appropriate safeguards in the form of the recursion limit passed in from configuration. The algorithm is as follows:

In this method, tolerance (ϵ) values are 0.001 below 1000 km and 0.01 above 1000 km (Increasing tolerance increases the integration speed at the expense of accuracy). The value of Max recursion level can be set based on receiver processing power; it can be 100 in this document.

set weights for K15 sample points

wi1 =0.022935322010529224963732008058970

wi2 = 0.063092092629978553290700663189204

wi3 = 0.104790010322250183839876322541518

wi4 = 0.140653259715525918745189590510238

wi5 = 0.169004726639267902826583426598550

wi6 = 0.190350578064785409913256402421014

wi7 = 0.204432940075298892414161999234649

wi8 = 0.209482141084727828012999174891714

$$wi[15] = [\text{wi1}, \text{wi2}, \text{wi3}, \text{wi4}, \text{wi5}, \text{wi6}, \text{wi7}, \text{wi8}, \text{wi7}, \text{wi6}, \text{wi5}, \text{wi4}, \text{wi3}, \text{wi2}, \text{wi1}]$$

Set weights for G7 sample points

wig1 = 0.129484966168869693270611432679082,

wig2 = 0.279705391489276667901467771423780,

wig3 = 0.381830050505118944950369775488975,

wig4 = 0.417959183673469387755102040816327,

$wig[7]$
$=$ [wig1, wig2, wig3, wig4, wig3, wig2, wig1]

Set at what points the samples are used in integration process

xi0 = 0

xi1 = 0.20778495500789846760068940377324 5

xi2 = 0.40584515137739716690660641207696 1

xi3 = 0.58608723546769113029414483825873 0

xi4 = 0.74153118559939443986386477328078 8

xi5 = 0.86486442335976907278971278864092 6

xi6 = 0.94910791234275852452618968404785 1

xi7 = 0.99145537112081263920685469752632 9

$xi[15] = [-\text{xi7}, -\text{xi6}, -\text{xi5}, -\text{xi4}, -\text{xi3}, -\text{xi2}, -\text{xi1}, \text{xi0}, \text{xi1}, \text{xi2}, \text{xi3}, \text{xi4}, \text{xi5}, \text{xi6}, \text{xi7}]$

set $intk = 0$ and $intg = 0$

set $gind = 1$

set $dh_1 = h_1$(height at point P_1), and $dh_2 = h_2$ (height at point P_2)

Calculate the midpoint (hh) and the half difference (hd_2)

$$hd_2 = {(dh_2 - dh_1)}/{2} \quad \text{Eq. 191}$$

$$hh = {(dh_2 + dh_1)}/{2} \quad \text{Eq. 192}$$

calculate

for i = 1 to 15

$$x = hd_2 * xi[i] + hh \quad \text{Eq. 193}$$

if ray is vertical

$$y = \text{compute electron density along vertical ray at x} \quad \text{Eq. 194}$$

(see Sec. 2.13.1)

if ray is slant

$$y = \text{compute electron density along slant ray at x} \quad \text{Eq. 195}$$

(see Sec. 2.13.2)

$$intk = intk + y * wi[i] \quad \text{Eq. 196}$$

if(i is even number) (for G7 point, i.e mod(i,2) == 0)

$intg = intg + y * wig[gind]$ Eq. 197

$gind = gind + 1$ Eq. 198

Once the for loop is over, calculate

$intk = intk * hd_2$ Eq. 199

$intg = intg * hd_2$ Eq. 200

Check if the error is within tolerance

If $abs\left(\frac{intk-intg}{intk}\right) \leq tolerance$ or $abs(intk - intg) \leq tolerance$

Then

$GNk = intk$ Eq. 201

Else if $current\ recursion\ level = Max\ recursion\ level$ (max $recursion\ level\ reached$)

Then

$GNk = intk$ Eq. 202

Else (i.e. the error is more than tolerance and maximum recursion level is not reached)

Then split the height into two equal halves and compute the integration at each portion as

For first portion

$dh_1 = h_1$

$dh_2 = h_1 + hd_2$

Repeat Eqs. 191 to Eq. 200, which will give *intk1*

For second portion

$dh_1 = h_1 + hd_2$

$dh_2 = h_2$

Repeat Eqs. 191 to Eq. 200, which will give *intk2*

Finally,

$GNk = intk1 + intk2$

$TEC = (GNk) \times 10^{-13}$TECU

Eq. 203

Appendix A. CCIR files

The CCIR files are used inside NeQuick to compute foF2 and M(3000)F2. These coefficients are stored in the ccirxx.asc files and include the spherical harmonic coefficients representing the development of monthly median foF2 and M(3000)F2 all over the world.

The coefficients correspond to low (Sun Spot number=0) and high (sun spot number=100) solar activity conditions. For other Sun Spot number activity, the coefficients must be interpolated (or extrapolated) as mentioned in Sec. 2.8.2 to obtain the values corresponding to the required solar activity [15- 19].

Each file ccirxx.asc contains 2858 values sequentially organised as follows: {$f2_{1,1,1}$, $f2_{1,1,2}$, ..., $f2_{1,1,13}$, $f2_{1,2,1}$, $f2_{1,2,2}$, ... $f2_{1,2,13}$, $f2_{1,76,1}$, $f2_{1,76,2}$, …, $f2_{1,76,13}$, $f2_{2,1,1}$, …, $f2_{2,1,2}$, …, $f2_{2,1,13}$, $f2_{2,2,1}$, $f2_{2,2,2}$, …, $f2_{2,2,13}$, …, $f2_{2,76,1}$, $f2_{2,76,2}$, …, $f2_{2,76,13}$, $fm3_{1,1,1}$, $fm3_{1,1,2}$, …, $fm3_{1,1,9}$, $fm3_{1,2,1}$, $fm3_{1,2,2}$, …, $fm3_{1,2,9}$, …, $fm3_{1,49,1}$, $fm3_{1,49,2}$, …, $fm3_{1,49,9}$, $fm3_{2,1,1}$, $fm3_{2,1,2}$, …, $fm3_{2,1,9}$, $fm3_{2,2,1}$, $fm3_{2,2,2}$, …, $fm3_{2,2,9}$, …, $fm3_{2,49,1}$, $fm3_{2,49,2}$, …, $fm3_{2,49,9}$}. (The notation is explained in the definition of the F2 and Fm3 arrays).

The CCIR file naming convention is ccirxx.asc where each xx means month + 10, for eg: January month file is ccir11.asc. These files will be available at the link provided in the SIS ICD of the L1 signals.

Appendix B. MODIP file

The MODIP grid allows estimating MODIP (μ, degree) at a given point (φ, λ) by interpolation of the relevant values contained in the support file modipNeQN_wrapped.asc. It is recommended to preload this grid in the main executable containing the NeQuick integration routine. This grid is later used within NeQuick to compute MODIP (Refer: Sec. 2.1) at a given point (φ, λ) interpolating with the 4×4 -points grid surrounding the desired element (φ, λ) [15 -19].

The file modipNeQN_wrapped.asc contains the values of MODIP (μ, degree) on a geocentric grid from 90°S to 90°N with a 5-degree step in latitude and from 180°W to 180°E with a 10-degree in longitude. For computational purposes, it is wrapped around including as first column the values of 170°E (i.e. 190°W) and in the last column the values of 170°W(i.e. 190°E), also there is an extra first and last rows phased 180 degrees in longitude to wrap the poles around.

The modipNeQN_wrapped.asc file will be available at the link provided in the SIS ICD of the L1 signals.

Appendix C. Coordinates Used In NeQuick-N

The Modip table grid file used by NeQuick-N is calculated using the IGRF (International Geomagnetic Reference Field) model for the Earth's magnetic field. The coordinates derived from such a model are defined as corrected Geomagnetic coordinates (CGM), as defined in the Galileo ionosphere user algorithm document [18].

Typical geomagnetic coordinates are those derived from a dipole approximation of the Earth's magnetic field. In this sense, parameters that depend on dipole latitude, such as the magnetic dip I or MODIP, were defined based on geomagnetic coordinates and not CGM.

NeQuick computes I and MODIP using CGM coordinates instead of the dipole geomagnetic coordinates [18]. This distinction is usually not found on NeQuick references and is given here solely for the user's knowledge, having no impact on the performance of the model.

Appendix D. Integration of NeQuick-N into Higher Level Software

NeQuick-N requires data from 13 files, as indicated in Appendix A and Appendix B of this document. Although the convenience of integrating NeQuick-N as a library with clear interfaces is convenient for many reasons, the loading of those 13 files for each Slant Delay computation introduces an unnecessary burden in terms of computation time. Therefore, it is recommended to integrate the pre-loading of those files outside NeQuick-N in the main target software at initialization.

Appendix E. Computation rate of Ionospheric Corrections

Ionosphere variations in the nominal conditions are slow. Therefore, the computation of ionosphere correction for the stationary user is not required at high rates and may suffice to compute the corrections every 10 seconds for stationary user.

References

[1] J. Titheridge, "Modelling the peak of the ionospheric E-layer," *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. v [1]ol 62, no. issue 2, pp. P:93-11 , 2000.

[2] ITU-R, "Ionospheric propagation data and prediction methods required for the design of satellite services and systems," *rec. ITU-R* , pp. p. 531-11,, January 2012.

[3] R. Leitinger, M.-L. Zhang and S. Radicella, "An improved bottomside for the ionospheric electron density model NeQuick," *ann. Geophys,* Vols. vol. 48, no. 3, pp. p. 525-534, 2005.

[4] M. Knezevich and R. S.M, "Development of an ionospheric NeQuick model algorithm for GNSS receivers," *naviTEc2004* , no. noordwijk, 2004..

[5] ITU-R, "Choices of indices for long-term ionospheric predictions," *ITU-R,* pp. p.371-8, 1999.

[6] k. Rawer, "Propagation of Decameter waves (h.f. band)," in *Meteorological and Astronomical influences on Radio Wave Propagation,* vol. chapter 11, landmark (ed.), Academic press new york, 1963, p. chapter 11. p. 221.

[7] J. Klobuchar, "An Ionospheric Time-Delay Algorithm for Single-Frequency GPS Users," *IEEE Transactions on Aerospace and electronic systems,* Vols. Aes-23 (3),, p. 325–331, 1987.

[8] "http://www.ngdc.noaa.gov/IAgA/vmod/igrf.html," International Geomagnetic Reference Field, (online), 12th September 2013. [Online].

[9] "NavIC Signal In Space ICD for Standard Positioning Service in L1 frequency". 2023, version 1.0, Indian Space Research Organisation, ISRO, India.

[10] M. Mosert de Gonzalez and S. Radicella, "On a characteristic point at the base of F2 layer in the ionosphere," in *Adv. Space res., vol. 10, iss. 11,* 1990.

[11] R. Di giovanni.G, "An analytical model of the electron density profile in the ionosphere," *Adv. Space res.,* Vols. 10,no. 11, pp. p. 27-30, 1990.

[12] Z. A. G. Radicella.S.M, "The improved DGR analytical model of electron density height profile and total electron content in the ionosphere," Vols. vol. xxxviii, no. 1, pp. p-35-41, 1995.

[13] L. Radicella.S.M, " The evolution of the DGR approach to model electron density profiles," *Adv. Space Res*, vol. 27, pp. p. 35-40, 2001.

[14] C. G. Coordinates, "http://omniweb.gsfc.nasa.gov/vitmo/cgmm_des.html," naSa vitmo models, 12th september 2013. [Online].

[15] O.-p. E. R.-r. M. Prieto-cerdeira, "Assessment of NeQuickIonospheric Model for Galileo Single-Frequency Users: The European Way," *GPS World,* Vols. vol. 25, no. 6, p. pp. 53–58, 2014.

[16] S. M. Radicella, "The NeQuick model genesis, uses and evolution," *Annals of Geophysics*, vol. vol.52, 2009.

[17] M. Maheshwari and N. Srini, "Estimation of Galileo like ionosphere coefficients using IRNSS data for equatorial region," in *IEEE, URSI AP-RASC* , 2019.

[18] "Ionospheric Correction Algorithm For Galileo Single Frequency Users,", European GNSS (Galileo) Open Service, September 2016.

[19] M. Maheshwari and N. Srini, "IRNSS ionosphere modelling for single frequency receiver using data ingestion technique," in *IEEE, 4th International Conference for Convergence in Technology (I2CT)*, 2018.