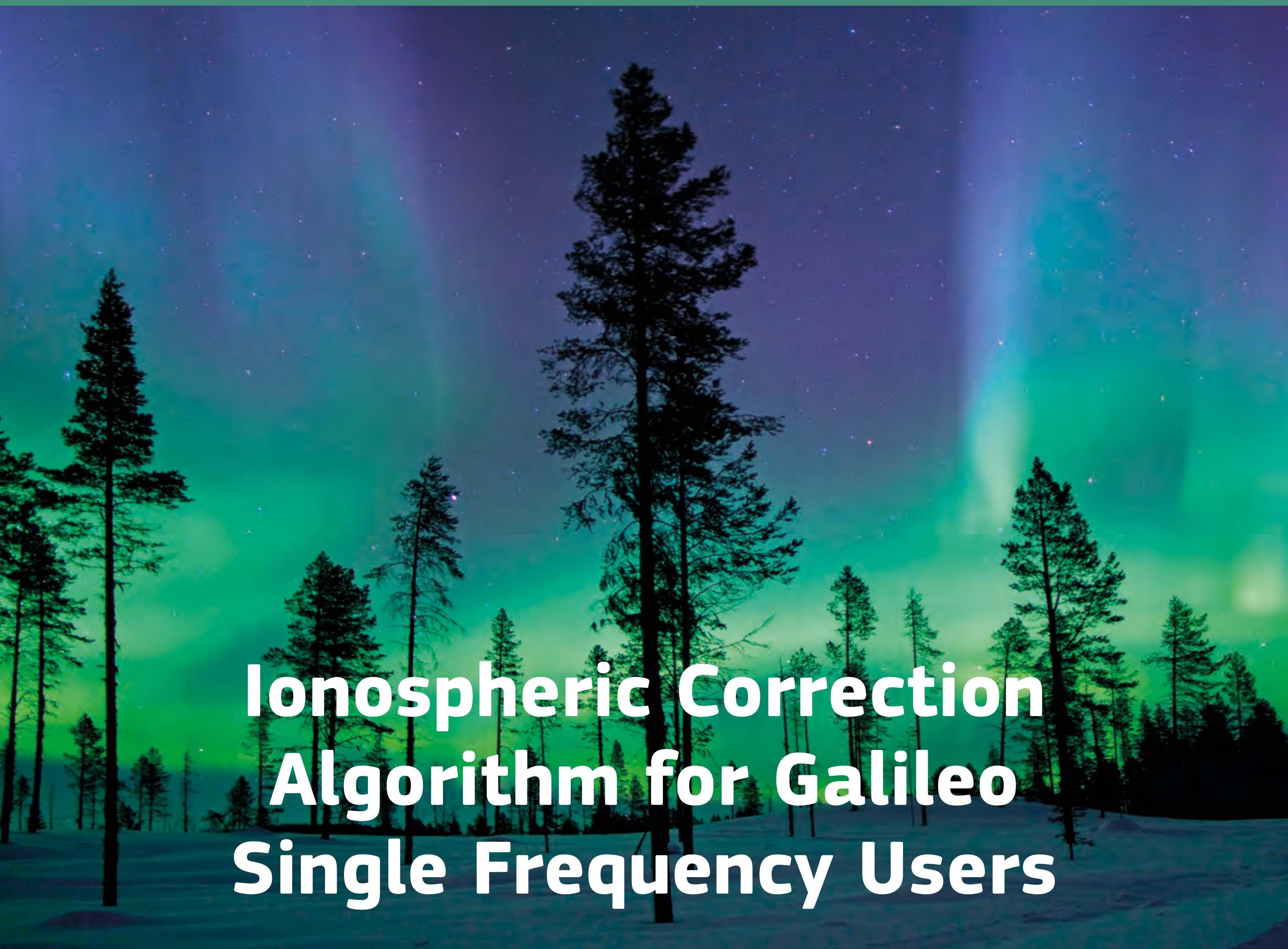
European GNSS (Galileo) Open Service



Ionospheric Correction Algorithm for Galileo Single Frequency Users

GALILEO

NAVIGATION
SOLUTIONS
POWERED BY
EUROPE

Terms of Use and Disclaimers

This document describes the ionospheric model developed for the Galileo satellite navigation system that can be used to determine Galileo single-frequency ionospheric corrections. Its content has been prepared and scrutinized by various groups of specialized scientists. The model has been characterized and thoroughly tested and gives encouraging performance improvements compared to other currently used solutions. Ionosphere's physical behaviour is however such that one cannot produce an algorithm, which will systematically deliver fully satisfactory compensation of ionospheric error under all conditions.

The European Commission, ESA, the author(s) or contributor(s) therefore do not assume any responsibility whatsoever for its use, and do not make any guarantee, expressed or implied, about the quality, reliability, fitness for any particular use or any other characteristic of the algorithm. Under no circumstances shall the European Commission, European Union, ESA, the author(s) or contributor(s) be liable for damages resulting directly or indirectly from the use, misuse or inability to use the algorithm.

Acknowledgements

The *NeQuick* electron density model was developed by the Abdus Salam International Center of Theoretical Physics (ICTP) and the University of Graz. The adaptation of *NeQuick* for Galileo single-frequency ionospheric correction algorithm (*NeQuick G*) has been performed by the European Space Agency (ESA) involving the original authors and other European ionospheric scientists under various ESA contracts. The step-by-step algorithmic description of *NeQuick* for Galileo contained in this document has been a collaborative effort of ICTP, ESA and the European Commission, including JRC.

Document Change Record

Reason for Change	Issue	Revision	Date
First Issue	Issue 1	0	November 2014
Document ready for publication after revision	Issue 1	1	February 2015

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Document Scope

This document complements the Galileo OS SIS ICD [1] by describing in detail the reference algorithm to be implemented at user receivers to compute ionospheric corrections based on the broadcast coefficients in the navigation message for Galileo single-frequency users. The term "Galileo" is used to refer to system established under the European GNSS (Galileo) programme.

It also includes the description of a sample implementation of the *NeQuick* ionospheric model as adapted for Galileo correction algorithm, and data for the verification of independent implementations.

1.2 Background

Galileo is the European global navigation satellite system providing a highly accurate and global positioning service under civilian control. Galileo, and in general current GNSS, are based on the broadcasting of electromagnetic ranging signals in the L frequency band. Those satellite signals suffer from a number of impairments when propagating through the Earth's atmosphere. In this sense, Earth's atmosphere can be subdivided into:

- the troposphere, whose main effect is a group delay on the navigation signal due to water vapour and the gas components of the dry air. This delay, for microwave frequencies, is non-dispersive (independent of frequency).
- the ionosphere, which is the ionised part of the atmosphere, inducing a dispersive group delay that is several orders of magnitude larger than the one from the troposphere. Other ionospheric effects such as scintillations may be also observed.

The ionosphere is a region of weakly ionised gas in the Earth's atmosphere lying between about 50 kilometres up to several thousand kilometres from Earth's surface. Solar radiation is responsible for this ionisation producing free electrons and ions. The ionospheric refractive index (the ratio between the speed of propagation in the media and the speed of propagation in vacuum) is related to the number of free electrons through the propagation path. For this purpose, the Total Electron Content (TEC) is defined as the electron density in a cross-section of 1 m^2, integrated along a slant (or vertical) path between two points (*e.g.* a satellite and a receiver); it is expressed in TEC units (TECU) where 1TECU equals 10^{16}

electrons/m^2. The ionosphere affects radio wave propagation in various ways such as refraction, absorption, Faraday rotation, group delay, time dispersion or scintillations, being most of them related to TEC in the propagation path. These effects are dispersive, as they depend on the signal frequency.

The ionosphere is classically sub-divided in layers characterized by different properties: *D*, *E*, *F*1 and *F*2, the latter being largely responsible for the ionospheric effects which typically affect GNSS applications[i].

Ionospheric electron density and in general ionospheric effects depend on different factors such as time of the day, location, season, solar activity and the interaction between solar activity and the Earth's magnetic field or level of disturbance of the ionosphere, such as those happening during geomagnetic storms. On a large time-scale, solar activity follows a periodic 11-year cycle. The level of solar activity (and hence the solar cycle) is usually represented by solar indices such as the Sun Spot Number (SSN) or the solar radio flux at 10.7 cm (F10.7). The equatorial anomaly regions, located at around ±15-20 degrees on either side of the magnetic equator, usually present the largest TEC values. Mid-latitude regions daytime TEC values are usually less than half the value found in the equatorial anomaly region. Polar and auroral regions present moderate TEC values, but larger variability than in mid-latitudes due to the characteristics of the geomagnetic field.

The ionosphere group delay (delay on the pseudo-range or signal code phase), neglecting higher order terms, may be expressed as:

$$d_{Igr} = \frac{40.3}{f^2} \cdot \int_{path} N \cdot dl = \frac{40.3}{f^2} \cdot STEC \qquad \text{Eq. 1}$$

Where d_{Igr} is the group delay [m], f is frequency [Hz], *N* is electron density [electrons/m^3], *STEC* is Slant Total Electron Content [electrons/m^2], and path is the propagation path between receiver and satellite. This effect introduces ranging errors of several meters if not corrected. Higher order terms usually account for differences at millimetre level and may be neglected for code ranging. The effect

---

[i] Historically, the division arose from the successive plateaus of electron density (Ne) observed on records of the time delay (i.e., virtual height) of radio reflections as the transmitted signal was swept through frequency. The *E* layer was the first to be detected and was so labeled as being the atmospheric layer reflecting the *E* vector of the radio signal. Later the lower *D* and higher *F* layers were discovered. Thus the four main ionospheric regions can be associated with different governing physical processes, and this physics (rather than simple height differentiation) is the basis for labeling the ionospheric regions as a *D*, *E*, *F1*, or *F2* [16].

on the carrier phase has the same magnitude as the code delay, but opposite sign, meaning that the carrier phase is advanced while propagating through the ionosphere.

Ionospheric group delay is dispersive in nature and its effect can be mitigated using combinations of signals at two frequencies. For single-frequency receivers, GNSS systems often rely on correction models driven by broadcast data. For example, in GPS, the Ionospheric Correction Algorithm (ICA) [10] uses 8 broadcast coefficients to describe the ionosphere which is represented as a two-dimensional thin-shell model (all the VTEC is assumed to be concentrated in a two-dimensional shell at a given height, relying on an analytical mapping or obliquity function to convert between VTEC and STEC depending on the elevation angle). This model is very efficient in terms of computational complexity and it typically removes over 50% of the ionospheric error particularly at mid-latitudes. Other example are Satellite Based Augmentation Systems, such as EGNOS or WAAS, which also rely on a thin shell model for correction represented with grid points distributed over the coverage area and broadcasting continuously the estimated vertical delay (and related error bound) in those grid points. They achieve great correction accuracy at the expense of large bandwidth required in their messages. Occasionally, during high solar activity periods or during geomagnetic storm periods, they may suffer from mapping function errors and spatial resolution, particularly at low latitudes and low elevations. This model is defined by the SBAS MOPS [2].

Galileo has been designed to provide various civil frequency combinations in order to mitigate the effects of the ionosphere using dual-frequency combinations. Single frequency receivers will be able to counteract the errors introduced by the ionospheric propagation delay using the Galileo single-frequency ionospheric correction algorithm described within this document, which is based on a three dimensional representation of the electron density using an adaptation of the *NeQuick* ionospheric electron density model for quasi-real-time corrections and driven by three broadcast coefficients in the navigation message.

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 monthly mean electron density from analytical profiles, depending on the solar activity-related input values: R_{12} (12-month smoothed sunspot number) or *F10.7* (previously defined), month, geographic latitude and longitude, height and UT. A global VTEC map obtained with *NeQuick* for R_{12}=150, 13h UT and the month of April with a grid resolution of 2.5x2.5 degrees in latitude and longitude is illustrated in Figure 1. The first version (*NeQuick 1*) of this model was adapted by ITU-R for Total Electron Content (TEC) estimation used for radiowave propagation predictions. The climatological monthly mean model has continued its development with updated formulations and a new version *NeQuick 2* is currently recommended in ITU-R Recommendation P.531 [2].

The *NeQuick* model has been adapted for real-time Galileo single-frequency ionospheric corrections (for convenience, it will be referred to as *NeQuick G*) in order to derive real-time predictions based on a single input parameter, the Effective Ionisation Level, *Az*, which is determined using three coefficients broadcast in the navigation message. This version of *NeQuick* is the one recommended for implementation in user equipment consistent with the broadcast coefficients, as opposed to the still evolving monthly mean model available from ITU.

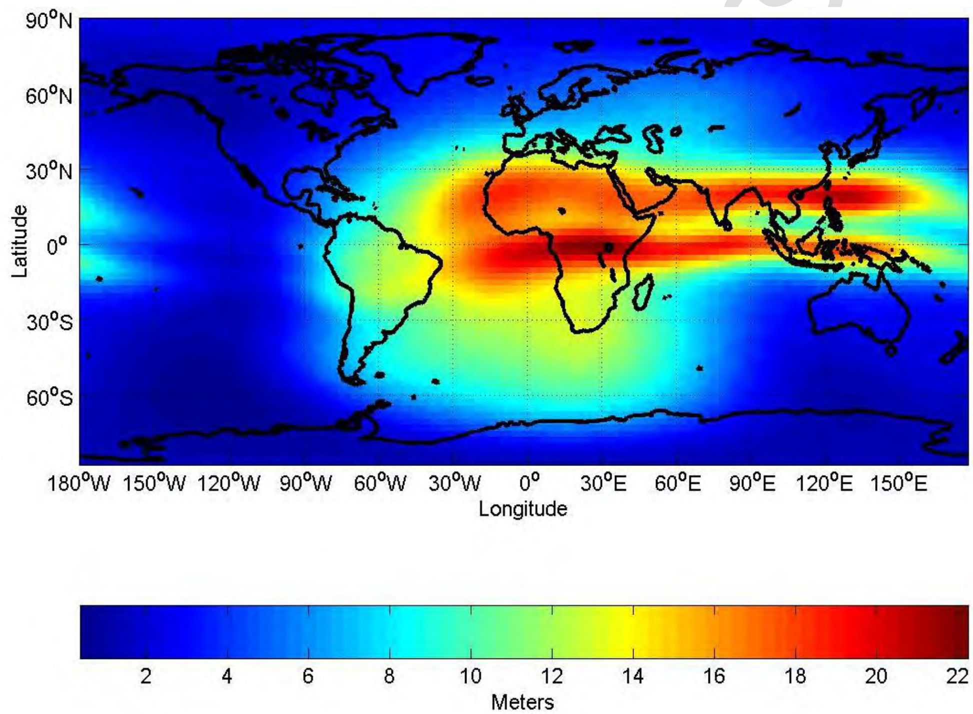


Figure 1. Example of a global VTEC map obtained with *NeQuick*

2. Single Frequency Ionospheric Correction Algorithm

2.1 Overview

Receivers operating in single frequency mode may use the single frequency ionospheric correction algorithm described in the following pages to estimate the ionospheric delay on each satellite link.

As specified in the Galileo OS SIS ICD [1], the Effective Ionisation Level, *Az*, is determined from three ionospheric coefficients (broadcast within the navigation message) as follows:

$$Az = a_{i0} + a_{i1} \times MODIP + a_{i2} \times (MODIP)^2 \qquad \text{Eq. 2}$$

where (a_{i0}, a_{i1}, a_{i2}) are the three broadcast coefficients and *MODIP* is Modified Dip Latitude at the location of the user receiver. MODIP is expressed in degrees and a table grid of MODIP values versus geographical location is provided together with *NeQuick G* model. The receiver then calculates the integrated Slant Total Electron Content along the path using *NeQuick G* and converts it to slant delay using Eq. 1.

2.2 Step-by-step procedure

In order to implement the ionospheric algorithm for Galileo single frequency receivers the following steps shall be followed:

for each satellite-receiver link

Obtain estimates of receiver position $(\varphi,\lambda,h)_i$, satellite position $(\varphi,\lambda,h)^j$ and time (time of day and month)

Obtain receiver $MODIP_U$ using φ_i, λ_i.

Obtain *Effective Ionisation Level* Az_U using eq. (2) with $MODIP_U$ and broadcast coefficients (a_{i0}, a_{i1}, a_{i2})

Call *NeQuick G* STEC integration routine for path $(x,y,z)^j$ to $(x,y,z)_i$,

for each integration point in the path

Call *NeQuick* routine to obtain electron density with Az_U, time of day and month

end

Integrate STEC for all points in the path

Obtain correction by converting STEC to code delay using Eq. 1 for the corresponding frequency

Apply correction to selected link

End

2.3 Inputs and Outputs

In order to evaluate TEC values the receiver needs as input:

Parameter	Description	Unit
a_{i0}	Effective Ionisation Level 1st order parameter	sfu**
a_{i1}	Effective Ionisation Level 2nd order parameter	sfu**/deg
a_{i2}	Effective Ionisation Level 3rd order parameter	sfu**/deg^2
φ_1	Geodetic latitude from receiver	deg
λ_1	Geodetic longitude from receiver	deg
h_1	Geodetic height from receiver	meters
φ_2	Geodetic latitude from satellite	deg
λ_2	Geodetic longitude from satellite	deg
h_2	Geodetic height from satellite	meters
UT	UT time	hours
mth	Month (numerical value, January = 1, ...)	dimensionless

Table 1. Input Parameters

** Note that "sfu" (solar flux unit) is not a SI unit but can be converted as: 1 sfu = 10^{-22} W/(m^2*Hz)

Remark: Receiver and satellite positions estimated values are given in WGS-84 ellipsoidal coordinates: geodetic latitude, geodetic longitude and ellipsoidal height.

The output of the algorithm is STEC in TECU that can be converted to ionospheric delay using Eq. 1.

2.3.1 Galileo navigation message relevant to single-frequency ionospheric algorithm

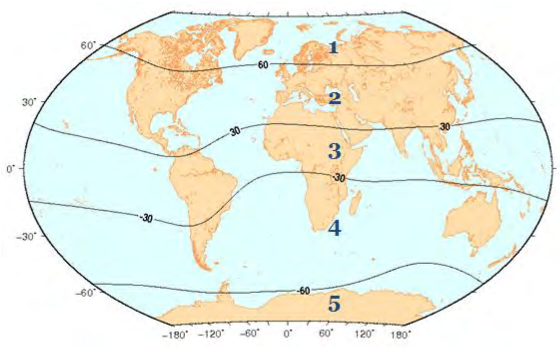
As described in the Galileo OS SIS ICD [1], the following parameters are broadcast in the Galileo navigation message (the parameters are sent within both F/NAV and I/NAV):

- Three Effective Ionisation Level coefficients (a_{i0}, a_{i1} and a_{i2}).
- Five Ionospheric Disturbance Flags for Regions 1 to 5 (SF_1, SF_2, SF_3, SF_4 and SF_5).

As detailed in the OS SIS ICD [1] the ionospheric correction parameters are transmitted within F/NAV Page Type 1 and I/NAV Word Type 5.

2.4 MODIP Regions

Depending on the severity and general characterisation of ionospheric effects, five regions are defined based on their MODIP (related to geomagnetic field). The five regions are presented below:



Region 1	60deg < *MODIP* ≤ 90deg
Region 2	30deg < *MODIP* ≤ 60deg
Region 3	-30deg ≤ *MODIP* ≤ 30deg
Region 4	-60deg ≤ *MODIP* < -30deg
Region 5	-90deg ≤ *MODIP* < -60deg

Figure 2. MODIP regions associated to different ionospheric characteristics

Fields have been reserved in the Galileo OS SIS ICD [1] to potentially broadcast specific information about the state of the ionosphere in each of these regions (see Ionospheric Disturbance Flags above). These parameters are not used in the current version of the model presented in this document.

2.5 NeQuick G ionospheric electron density model

NeQuick model has been adapted for Galileo single-frequency ionospheric corrections in order to derive real-time predictions based on a single input parameter, the Effective Ionisation Level. *NeQuick* is a "profiler" that makes use of three profile anchor points: *E* layer peak (at a fixed height of 120 km), *F*1 peak, *F*2 peak, where *E*, *F*1 and *F*2 are different layers of the ionosphere, as previously introduced. To model the anchor points the model employs "ionosonde parameters" *foE*, *foF*1, *foF*2 (critical frequencies) and *M*(3000)*F2* (transmission factor).

The model is constituted by two major components:

a) The bottom side model for the height region below the peak of the *F*2-layer, which consists on the superposition of three Epstein layers which peak at the anchor points. This is a modified version of the "Di Giovanni-Radicella" model based on the ionospheric characteristics *foE*, *foF*1, *foF*2 and *M*(3000)*F2*. For *foE* derivation, a modified formulation of that due to John Titheridge is selected and *foF*1 is selected as being equal to 1.4×*foE* during daytime and zero during night-time, respectively. For the calculation of *foF*2 and *M*(3000)*F*2, the CCIR maps (provided as *ccirXX.asc* files) are used.

b) The topside model for the height region above the *F*2-layer peak. The topside of *NeQuick* is a semi-Epstein layer with a height dependent thickness parameter *B* through a new parameter *H*. A correction factor adjusts vertical TEC values to take into account exosphere electron density in a simple manner.

2.5.1 The Epstein function

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

$$\mathrm{Epst}(X,Y,Z,W) = \frac{X \cdot \exp\left(\frac{W-Y}{Z}\right)}{\left(1+\exp\left(\frac{W-Y}{Z}\right)\right)^2} \qquad \text{Eq. 3}$$

where, for the purpose of ionospheric profile, in general *X* denotes peak amplitude, *Y* denotes peak height, *Z* describes thickness around the peak, and *W* is the height dependent variable.

2.5.2 Constants used

For the calculation of the slant TEC, some constant parameters are used and they are summarized in Table 2.

Symbol	Constant description	Value	Units
DR	Degree to radiant conversion factor	$\pi/180$	rad/deg
RD	Radiant to degree conversion factor	$180/\pi$	rad/deg
χ_0	Zenith angle at night-day transition	86.23	deg
R_E	Earth mean radius	6371.2	km

Table 2. Constants definition

2.5.3 Complementary files

2.5.3.1 MODIP grid

The MODIP grid allows to estimate MODIP μ [deg] at a given point (φ,λ) by interpolation of the relevant values contained in the support file *modipNeQG _wrapped.asc*. The file is provided in the electronic version of this document in Annex C. 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 at a given point (φ,λ) interpolating with the 4x4-points grid surrounding the desired element (φ,λ).

The file *modipNeQG_wrapped.asc*. contains the values of MODIP μ (expressed in degrees) 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.

2.5.3.2 CCIR files

The CCIR files are used inside *NeQuick* to compute *foF*2 and *M*(3000)*F*2 as described in [4]. These coefficients are stored in the ccirXX.asc files and include the spherical harmonic coefficients representing the development of monthly median *foF*2 and *M*(3000)*F*2 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) to obtain the values corresponding to the required solar activity.

Each file ccirXX.asc contains 2858 values sequentially organized 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 is month + 10. The content of the file is included in Annex C.

2.5.4 Auxiliary parameters

To compute the *NeQuick* electron density, several auxiliary parameters are preliminarily evaluated through specific modules. In the following sections the formulation of each of these modules resulting in the auxiliary parameters is given.

2.5.4.1 Local Time

Compute local time LT (in hours and decimals) for the location considered.

Inputs:

longitude λ [deg], Universal Time UT [hours and decimals].

Output:

local time LT [hours and decimals].

$$\mathrm{LT} = \mathrm{UT} + \frac{\lambda}{15} \qquad \text{Eq. 4}$$

2.5.4.2 Read modipNeQG_wrapped.asc values in an array

stModip: $\mathrm{stModip}_{i,j}$ *i*=1,…,39; *j*=-1,…,37; Eq. 5

2.5.4.3 Compute MODIP

Inputs:

latitude φ [deg], longitude λ [deg], array stModip (of MODIP values).

Output:

MODIP μ [deg]

The selection of the interpolation grid-points is done by computing:

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

If *l*<0 use

$$l = l + 36 \qquad \text{Eq. 7}$$

If *l* >33 use

$$l = l - 36 \qquad \text{Eq. 8}$$

Compute

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

$$x = a - \mathrm{int}(a) \qquad \text{Eq. 10}$$

$$i = \mathrm{int}(a) - 2 \qquad \text{Eq. 11}$$

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

$$z_{j,k} = \text{stModip}_{i+j,l+k} \quad \text{Eq. 12}$$

For k=1,4 compute

$$z_k = z_x(z_{1,k}, z_{2,k}, z_{3,k}, z_{4,k}, x) \quad \text{Eq. 13}$$

using the interpolation function described in 2.5.7.1.

Finally compute

$$b = \frac{\lambda + 180}{10} \quad \text{Eq. 14}$$

$$y = b - \text{int}(b) \quad \text{Eq. 15}$$

and, using the interpolation function described in 2.5.7.1, calculate

$$\mu = z_x(z_1, z_2, z_3, z_4, y) \quad \text{Eq. 16}$$

2.5.4.4 Effective Ionization Level Az

Compute the Effective Ionization Level *Az* at the given receiver location (having MODIP μ) as a function of the coefficients (a_0, a_1, a_2) broadcast in the navigation message. Note that *Az* is not updated with MODIP along the ray. Instead, for each ray, *Az* is fixed to the one corresponding to MODIP at the receiver location.

Inputs:

Ionospheric coefficients (a_0, a_1, a_2), MODIP μ [deg].

Output:

Effective Ionisation Level *Az*.

If $a_0 = a_1 = a_2 = 0$,

$$Az = 63.7, \quad \text{Eq. 17}$$

else

$$Az = a_0 + a_1\mu + a_2\mu^2 \quad \text{Eq. 18}$$

The user should verify that *Az* is within range [0,400], as described in section 3.3.

Remark: This parameter is equivalent to F10.7 in climatological *NeQuick*.

2.5.4.5 *Effective Sunspot Number (R_{12} like)*

Compute the Effective Sunspot Number Az_R as a function of the Effective Ionisation Level Az.

$$Az_R = \sqrt{167273 + (Az - 63.7) \cdot 1123.6} - 408.99 \quad \text{Eq. 19}$$

Remark: This parameter is equivalent to R_{12} in climatological *NeQuick*.

2.5.4.6 *Solar declination*

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

Inputs:

month mth, Universal Time UT [hours]

Outputs:

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

Compute day of year at the middle of the month:

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

Compute time [days]:

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

Compute the argument:

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

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

Finally compute sine and cosine of solar declination:

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

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

2.5.4.7 *Solar zenith angle*

Compute solar zenith angle χ [deg] for the given location.

Inputs:

latitude φ [deg], local time LT [hours], $\sin(\delta_{Sun})$, $\cos(\delta_{Sun})$

Output:

solar zenith angle χ [deg].

Compute

$$\cos(\chi) = \sin(\varphi \cdot DR) \cdot \sin(\delta_{Sun}) + \cos(\varphi \cdot DR) \cdot \cos(\delta_{Sun}) \cdot \cos\left(\frac{\pi}{12}(12 - LT)\right)$$ Eq. 26

$$\chi = RD \cdot \mathrm{atan2}\left(\sqrt{1 - \cos^2(\chi)}, \cos(\chi)\right)$$ Eq. 27

2.5.4.8 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].

Inputs:

solar zenith angle χ [deg], χ_0 [deg]

Output:

effective solar zenith angle χ_{eff} [deg].

Being

$$\chi_0 = 86.23°$$ Eq. 28

Then

$$\chi_{eff} = \frac{\chi + [90 - 0.24 \cdot \exp(20 - 0.2 \cdot \chi)] \cdot \exp\big(12(\chi - \chi_0)\big)}{1 + \exp\big(12(\chi - \chi_0)\big)}$$ Eq. 29

2.5.5 Model parameters

In the following sections model peak parameter and auxiliary parameter values are calculated.

2.5.5.1 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.

Inputs:

latitude φ [deg], Effective Ionisation Level *Az*,

effective solar zenith angle χ_{eff} [deg], month mth.

Output:

foE [MHz].

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. 30

If mth=3,4,9,10 then *seas*=0 Eq. 31

If mth=5,6,7,8 then *seas*=1 Eq. 32

Introduce the latitudinal dependence:

$$ee = \exp(0.3 \cdot \varphi) \quad \text{Eq. 33}$$

$$seasp = seas \cdot \frac{ee - 1}{ee + 1} \quad \text{Eq. 34}$$

$$foE = \sqrt{(1.112 - 0.019 \cdot seasp)^2 \cdot \sqrt{Az} \cdot \left[\cos\left(\chi_{eff} \cdot DR\right)\right]^{0.6} + 0.49} \quad \text{Eq. 35}$$

The *E* layer maximum density *NmE* [10^{11} m^{-3}] as a function of *foE* [MHz] is computed as:

$$NmE = 0.124 \cdot foE^2 \quad \text{Eq. 36}$$

2.5.5.2 foF1 and NmF1

The *F*1 layer critical frequency *foF*1 [MHz] as a function of the solar zenith angle χ and the solar zenith angle at day night transition χ_0 is computed as:

Inputs:

E layer critical frequency *foE* [MHz], solar zenith angle χ [deg], solar zenith angle at day night transition χ_0 [deg]

Output:

foF1 [MHz]

$foF1 = 1.4 * foE$, if $foE \geq 2.0$ MHz

$foF1 = 0$, if $foE < 2.0$ MHz Eq. 37

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

The *F* layer maximum density *NmF1* [10^{11} m^{-3}] as a function of *foF1* [MHz] is computed as:

$$NmF1 = 0.124 \cdot (foE + 0.5)^2, \quad \text{if } foF1 \leq 0 \text{ and } foE > 2 \quad \text{Eq. 38}$$

$$NmF1 = 0.124 \cdot foF1^2, \quad \text{otherwise} \quad \text{Eq. 39}$$

2.5.5.3 foF2 and NmF2; M(3000)F2

2.5.5.3.1 Read ccirXX.asc values

Input:

Month mth

Outputs:

*F*2, *Fm*3

Select the file name to read:

$$XX = \text{mth} + 10 \quad \text{Eq. 40}$$

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

coefficients for *foF*2

$$F2: \quad f2_{i,j,k} \quad i=1,2; \quad j=1,\dots,76; \quad k=1,\dots,13 \quad \text{Eq. 41}$$

coefficients for *M*(3000)*F*2

$$Fm3: \quad fm3_{i,j,k} \quad i=1,2; \quad j=1,\dots,49; \quad k=1,\dots,9 \quad \text{Eq. 42}$$

2.5.5.3.2 Interpolate ITU-R coefficients for Az_R

Compute *AF*2, the array of interpolated coefficients for *foF*2 and *Am*3, the array of interpolated coefficients for *M*(3000)*F*2.

Inputs:

$F2$, $Fm3$, Az_R

Outputs:

$AF2$, $Am3$

Compute the array of interpolated coefficients for *foF*2:

$$AF2: \quad af2_{j,k} \quad j=1,\ldots,76; \quad k=1,\ldots,13 \qquad \text{Eq. 43}$$

*AF*2 elements are calculated by linear combination of the elements of *F*2:

$$af2_{j,k} = f2_{1,j,k}\left(1 - \frac{Az_R}{100}\right) + f2_{2,j,k}\frac{Az_R}{100} \qquad j=1,\ldots,76; \quad k=1,\ldots,13 \qquad \text{Eq. 44}$$

Compute the array of interpolated coefficients for *M*(3000)*F*2:

$$Am3: \quad am3_{j,k} \quad j=1,\ldots,49; \quad k=1,\ldots,9 \qquad \text{Eq. 45}$$

*Am*3 elements are calculated by linear combination of the elements of *Fm*3:

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

2.5.5.3.3 Compute Fourier time series for *foF*2 and *M*(3000)*F*2

Inputs:

Universal Time UT [hours], arrays of interpolated ITU-R coefficients *AF*2, *Am*3

Outputs:

*CF*2, *Cm*3, vectors of coefficients for Legendre calculation for *foF*2 and *M*(3000)*F*2.

The vector *CF*2 has 76 elements:

$$CF2: \quad cf2_l \quad l=1,\ldots,76 \qquad \text{Eq. 47}$$

The vector *Cm*3 has 49 elements:

$$Cm3: \quad cm3_l \quad l=1,\ldots,49 \qquad \text{Eq. 48}$$

Compute the time argument:

$$T = (15 \cdot \mathrm{UT} - 180) \cdot DR \qquad \text{Eq. 49}$$

For i=1,..,76 calculate the Fourier time series for *foF*2:

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

For *i*=1,..,49 calculate the Fourier time series for *M*(3000)*F*2:

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

2.5.5.3.4 Compute *foF*2 and *M*(3000)*F*2 by Legendre calculation

Inputs:

MODIP μ [deg], latitude φ [deg], longitude λ [deg], vector *CF*2 of the coefficients for Legendre combination for *foF*2, vector *Cm*3 of the coefficients for Legendre combination for *M*(3000)*F*2.

Outputs:

*foF*2 [MHz], *M*(3000)*F*2

Define vectors containing sine and cosine of the coordinates:

M: m_k *k*=1,...,12 Eq. 52

P: p_n *n*=2,...,9 Eq. 53

S: s_n *n*=2,...,9 Eq. 54

C: c_n *n*=2,...,9 Eq. 55

Compute MODIP coefficients

$$m_1 = 1 \qquad \text{Eq. 56}$$

and for *k*=2,...,12

$$m_k = \sin^{k-1}(\mu \cdot DR) \qquad \text{Eq. 57}$$

Compute latitude and longitude coefficients:

for *n*=2,...,9

$$p_n = \cos^{n-1}(\varphi \cdot DR) \qquad \text{Eq. 58}$$

$$s_n = \sin\big((n-1)\cdot\lambda\cdot DR\big)$$ Eq. 59

$$c_n = \cos\big((n-1)\cdot\lambda\cdot DR\big)$$ Eq. 60

Compute *foF*2

Order 0 term:

$$foF2_1 = \sum_{k=1}^{12} cf2_k m_k$$ Eq. 61

having the increased Legendre grades for *foF*2 in a vector:

Q: q_n *n*=1,...,9 Eq. 62

$$Q = (12,12,9,5,2,1,1,1,1)$$ Eq. 63

for computational efficiency, define also:

K: k_n *n*=1,...,9 Eq. 64

$$k_1 = -q_1$$ Eq. 65

and for *n*=2,...,9

$$k_n = k_{n-1} + 2q_{n-1}$$ Eq. 66

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

$$foF2_n = \sum_{k=1}^{q_n}\big(cf2_{k_n+2k-1}c_n + cf2_{k_n+2k}s_n\big)m_k p_n$$ Eq. 67

Finally sum the terms to obtain *foF*2:

$$foF2 = \sum_{n=1}^{9} foF2_n$$ Eq. 68

Compute *M*(3000)*F*2

Order 0 term:

$$M(3000)F2_0 = \sum_{k=1}^{7} cm3_k m_k \qquad \text{Eq. 69}$$

having the increased Legendre grades for *M*(3000)*F*2 in a vector:

R: r_n *n*=1,...,7 Eq. 70

$$R = (7,8,6,3,2,1,1) \qquad \text{Eq. 71}$$

for computational efficiency, define also:

H: h_n *n*=1,...,7 Eq. 72

$$h_1 = -r_1 \qquad \text{Eq. 73}$$

and for n=2,...,7

$$h_n = h_{n-1} + 2r_{n-1} \qquad \text{Eq. 74}$$

for n=2,...7, compute the higher order terms:

$$M(3000)F2_n = \sum_{k=1}^{r_n} \left(cm3_{h_n+2k-1} c_n + cm3_{h_n+2k} s_n\right) m_k p_n \qquad \text{Eq. 75}$$

Finally sum the terms:

$$M(3000) = \sum_{n=1}^{7} M(3000)F2_n \qquad \text{Eq. 76}$$

To compute *NmF*2 use:

$$NmF2 = 0.124 \cdot foF2^2 \qquad \text{Eq. 77}$$

where *NmF*2 is in [10^{11}m^{-3}].

2.5.5.4 hmE

The *E* layer maximum density height *hmE* [km] is defined as a constant:

$$hmE = 120 \qquad \text{Eq. 78}$$

2.5.5.5 hmF1

Compute the $F1$ layer maximum density height $hmF1$ [km]:

Inputs:

$NmF1$ [10^{11} m^{-3}], Dip I [deg].

Output:

$hmF1$ [km].

$$hmF1 = \frac{hmF2 + hmE}{2} \qquad \text{Eq. 79}$$

2.5.5.6 hmF2

Compute the $F2$ layer maximum density height $hmF2$ [km].

Inputs:

foE [MHz], $foF2$ [MHz], $M(3000)F2$.

Output:

$hmF2$ [km].

$$hmF2 = \frac{1490 \cdot M \cdot \sqrt{\frac{0.0196 \cdot M^2 + 1}{1.2967 \cdot M^2 - 1}}}{M + \Delta M} - 176 \qquad \text{Eq. 80}$$

Where

$$M = M(3000)F2 \qquad \text{Eq. 81}$$

$$\Delta M = -0.012 \quad \text{if} \quad foE < 10^{-30} \qquad \text{Eq. 82}$$

$$\Delta M = \frac{0.253}{\rho - 1.215} - 0.012 \quad \text{if} \quad foE \geq 10^{-30} \qquad \text{Eq. 83}$$

and the ratio ρ is computed as:

$$\rho = \frac{\frac{foF2}{foE} \cdot \exp\left(20 \cdot \left(\frac{foF2}{foE} - 1.75\right)\right) + 1.75}{\exp\left(20 \cdot \left(\frac{foF2}{foE} - 1.75\right)\right) + 1} \qquad \text{Eq. 84}$$

2.5.5.7 B2bot, B1top, B1bot, BEtop, BEbot

Compute the thickness parameters *B2bot*, *B1top*, *B1bot*, *BEtop*, *BEbot* [km]

Inputs	Outputs	Definition	
NmF2 [10^{11} m^{-3}] *foF2* [MHz] *M*(3000)*F2*	*B2bot* [km]	$B2bot = \frac{0.385 \cdot NmF2}{0.01 \cdot \exp(-3.467+0.857 \cdot \ln(foF2^2)+2.02 \cdot \ln(M))}$ where $M = M(3000)F2$	Eq. 85
hmF1 [km] *hmF2*[km]	*B1top* [km]	$B1top = 0.3 \cdot (hmF2 - hmF1)$	Eq. 86
hmF1 [km] *hmE* [km]	*B1bot* [km]	$B1bot = 0.5 \cdot (hmF1 - hmE)$	Eq. 87
B1bot [km]	*BEtop* [km]	$BEtop = \max[B1bot, 7]$	Eq. 88
-	*BEbot* [km]	$BEbot = 5$	Eq. 89

2.5.5.8 A1

Compute the *F2* layer amplitude *A1* [10^{11} m^{-3}].

Inputs:

NmF2 [10^{11} m^{-3}].

Output:

A1 [10^{11} m^{-3}].

$$A1 = 4 \cdot NmF2 \quad \text{Eq. 90}$$

2.5.5.9 A2 and A3

Compute the *F1* layer amplitude *A2* [10^{11} m^{-3}] and the *E* layer amplitude *A3* [10^{11} m^{-3}].

Inputs:

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]

Output:

$A2$ [10^{11} m^{-3}], $A3$ [10^{11} m^{-3}].

if $foF1 < 0.5$:

$$A2 = 0 \quad \text{Eq. 91}$$

$$A3 = 4.0 \cdot [NmE - \text{Epst}(A1, hmF2, B2bot, hmE)] \quad \text{Eq. 92}$$

if $foF1 \geq 0.5$,

$$A3 = 4.0 \cdot NmE \quad \text{Eq. 93}$$

Repeat 5 times the iterations below:

$$A2a = 4.0 \cdot [NmF1 - \text{Epst}(A1, hmF2, B2bot, hmF1) - \text{Epst}(A3, hmE, BEtop, hmF1)] \quad \text{Eq. 94}$$

$$A2a = \frac{A2a \cdot \exp(A2a - 0.80 \cdot NmF1) + 0.80 \cdot NmF1}{1 + \exp(A2a - 0.80 \cdot NmF1)} \quad \text{Eq. 95}$$

$$A3a = 4.0 \cdot [NmE - \text{Epst}(A2a, hmF1, B1bot, hmE) - \text{Epst}(A1, hmF2, B2bot, hmE)] \quad \text{Eq. 96}$$

where the function Epst is the one defined in 2.5.1. Then compute

$$A2 = A2a \quad \text{Eq. 97}$$

$$A3 = \frac{A3a \cdot \exp\big(60 \cdot (A3a - 0.005)\big) + 0.05}{1 + \exp\big(60 \cdot (A3a - 0.005)\big)} \quad \text{Eq. 98}$$

2.5.5.10 Shape parameter k

Compute the shape parameter *k*.

Inputs:

mth, *NmF*2 [10^{11} m^{-3}], *hmF*2 [km], *B*2*bot* [km].

Output:

k.

First compute the auxiliary parameter *ka*:

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

$$ka = 6.705 - 0.014 \cdot Az_R - 0.008 \cdot hmF2 \quad \text{Eq. 99}$$

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

$$ka = -7.77 + 0.097 \cdot \left(\frac{hmF2}{B2bot}\right)^2 + 0.153 \cdot NmF2 \quad \text{Eq. 100}$$

Then compute the auxiliary parameter *kb*:

$$kb = \frac{ka \cdot \exp(ka - 2) + 2}{1 + \exp(ka - 2)} \quad \text{Eq. 101}$$

Eventually compute:

$$k = \frac{8 \cdot \exp(kb - 8) + kb}{1 + \exp(kb - 8)} \quad \text{Eq. 102}$$

2.5.5.11 H_0

Compute the topside thickness parameter H_0 [km].

Inputs:

B2bot [km], *k*.

Output:

H_0 [km]

First compute the auxiliary parameter H_a:

$$H_a = k \cdot B2bot \quad \text{Eq. 103}$$

Then compute the auxiliary parameters *x* and *v* as follows:

$$x = \frac{H_a - 150}{100} \quad \text{Eq. 104}$$

$$v = (0.041163 \cdot x - 0.183981) \cdot x + 1.424472 \quad \text{Eq. 105}$$

Eventually compute

$$H_0 = \frac{H_a}{v} \quad \text{Eq. 106}$$

2.5.6 Electron density computation

To compute the electron density $N = N(h, \varphi, \lambda, a_0, a_1, a_2, \text{mth}, \text{UT})$ at a given point (identified by the coordinates h, φ, λ) at a given time (mth, UT) and using a given set of Effective Ionisation Level *Az* derived with the Effective Ionisation Level coefficients (a_{i0}, a_{i1}, a_{i2}) and the MODIP at the receiver location, all *NeQuick* parameters have to be evaluated for the given point. Nevertheless 2 different modules have to be used accordingly to the height considered. In particular

$$\text{if} \quad h \leq hmF2 \qquad \text{Eq. 107}$$

the bottomside electron density has to be computed using the algorithm illustrated in 2.5.6.1, while

$$\text{if} \quad h > hmF2 \qquad \text{Eq. 108}$$

the topside electron density has to be computed using the algorithm illustrated in 2.5.6.2.

2.5.6.1 *The bottomside electron density*

Compute the electron density *N* of the bottomside (case $h \leq hmF2$).

Inputs:

height *h* [km], *A*1 [10^{11} m^{-3}], *A*2 [10^{11} m^{-3}], *A*3 [10^{11} m^{-3}], *hmF*2 [km], *hmF*1 [km], *hmE* [km], *B2bot* [km], *B*1*top* [km], *B*1*bot* [km], *BEtop* [km], *BEbot* [km].

Output:

(bottomside) electron density *N* [m^{-3}].

Select the relevant *B* parameters for the current height:

$$BE = \begin{cases} BEtop & \text{if } h > hmE \\ BEbot & \text{if } h \leq hmE \end{cases} \qquad \text{Eq. 109}$$

$$BF1 = \begin{cases} BF1top & \text{if } h > hmF1 \\ BF1bot & \text{if } h \leq hmF1 \end{cases} \qquad \text{Eq. 110}$$

Compute the exponential arguments for each layer:

$$\alpha_1 = \frac{h - hmF2}{B2bot} \qquad \text{Eq. 111}$$

$$\alpha_2 = \frac{h - hmF1}{BF1} \exp\left(\frac{10}{1 + |h - hmF2|}\right) \qquad \text{Eq. 112}$$

$$\alpha_3 = \frac{h - hmE}{BE} \exp\left(\frac{10}{1 + |h - hmF2|}\right) \quad \text{Eq. 113}$$

For each *i*=1,3 compute:

$$s_i = \begin{cases} 0 & \text{if } |\alpha_i| > 25 \\ A_i \dfrac{\exp(\alpha_i)}{(1 + \exp(\alpha_i))^2} & \text{if } |\alpha_i| \leq 25 \end{cases} \quad \text{Eq. 114}$$

If *h*≥100 km compute the electron density as:

$$N = (s_1 + s_2 + s_3) \times 10^{11} \quad \text{Eq. 115}$$

If *h*<100 km compute also the corrective terms:

$$ds_1 = \begin{cases} 0 & \text{if } |\alpha_1| > 25 \\ \dfrac{1}{B2bot} \dfrac{1 - \exp(\alpha_1)}{1 + \exp(\alpha_1)} & \text{if } |\alpha_1| \leq 25 \end{cases} \quad \text{Eq. 116}$$

$$ds_2 = \begin{cases} 0 & \text{if } |\alpha_2| > 25 \\ \dfrac{1}{BF1} \dfrac{1 - \exp(\alpha_2)}{1 + \exp(\alpha_2)} & \text{if } |\alpha_2| \leq 25 \end{cases} \quad \text{Eq. 117}$$

$$ds_3 = \begin{cases} 0 & \text{if } |\alpha_3| > 25 \\ \dfrac{1}{BE} \dfrac{1 - \exp(\alpha_3)}{1 + \exp(\alpha_3)} & \text{if } |\alpha_3| \leq 25 \end{cases} \quad \text{Eq. 118}$$

and the Chapman parameters:

$$BC = 1 - 10 \frac{\sum_{i=1}^{3} s_i ds_i}{\sum_{i=1}^{3} s_i} \quad \text{Eq. 119}$$

$$z = \frac{h - 100}{10} \quad \text{Eq. 120}$$

Then compute the electron density as:

$$N = (s_1 + s_2 + s_3) \cdot \exp(1 - BC \cdot z - \exp(-z)) \times 10^{11} \quad \text{Eq. 121}$$

2.5.6.2 The topside electron density

Compute the electron density *N* of the topside (case *h* > *hmF*2).

Inputs:

height h [km], $NmF2$ [10^{11} m^{-3}], $hmF2$ [km], H_0 [km].

Output:

(topside) electron density N [m^{-3}].

Define the constant parameters g and r as:

$$g=0.125 \quad \text{Eq. 122}$$

$$r=100 \quad \text{Eq. 123}$$

compute the arguments Δh and z as:

$$\Delta h = h - hmF2 \quad \text{Eq. 124}$$

$$z = \frac{\Delta h}{H_0\left[1 + \frac{rg\Delta h}{rH_0 + g\Delta h}\right]} \quad \text{Eq. 125}$$

then the exponential:

$$e_a = \exp(z) \quad \text{Eq. 126}$$

Eventually

$$N = \begin{cases} \frac{4 \cdot NmF2}{e_a} \times 10^{11} & \text{if } e_a > 10^{11} \\ 4 \cdot NmF2 \frac{e_a}{(1+e_a)^2} \times 10^{11} & \text{if } e_a \le 10^{11} \end{cases} \quad \text{Eq. 127}$$

2.5.7 Auxiliary routines

2.5.7.1 Third order interpolation function $z_x(z_1, z_2, z_3, z_4, x)$

Be $P1=(-1,z_1)$, $P2=(0,z_2)$, $P3=(1,z_3)$, $P4=(2,z_4)$. If $P=(x,z_x)$, to compute the interpolated value z_x at the position x, being $x \in [0,1)$, the following algorithm is applied.

Inputs:

z_1, z_2, z_3, z_4, x

Outputs:

z_x.

If $|x^2| \le 10^{10}$

$$z_x = z_2 \quad \text{Eq. 128}$$

Otherwise compute:

$$\delta = 2x - 1 \quad \text{Eq. 129}$$

$$g_1 = z_3 + z_2 \quad \text{Eq. 130}$$

$$g_2 = z_3 - z_2 \quad \text{Eq. 131}$$

$$g_3 = z_4 + z_1 \quad \text{Eq. 132}$$

$$g_4 = \frac{z_4 - z_1}{3} \quad \text{Eq. 133}$$

$$a_0 = 9g_1 - g_3 \quad \text{Eq. 134}$$

$$a_1 = 9g_2 - g_4 \quad \text{Eq. 135}$$

$$a_2 = g_3 - g_1 \quad \text{Eq. 136}$$

$$a_3 = g_4 - g_2 \quad \text{Eq. 137}$$

$$z_x = \frac{1}{16}(a_0 + a_1\delta + a_2\delta^2 + a_3\delta^3) \quad \text{Eq. 138}$$

2.5.8 TEC calculation

To compute the slant TEC along a straight line between a point P_1 and a point P_2, the *NeQuick* electron density N has to be evaluated on a point P defined by the coordinates $\{h,\varphi,\lambda\}$ 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.

The Earth is assumed to be a sphere with a radius of 6371.2 km, as indicated in Table 2.

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 section 2.5.8.1; otherwise, the slant integration algorithm to be used is the one described in section 2.5.8.2. When performing the TEC computation, the electron density at the point P has to be evaluated as indicated in 2.5.6, while the calculation of the coordinates of the point *P* along the ray-path is described in 2.5.8.1.1, in the case a vertical ray-path is considered, and in 2.5.8.2.6, if a slant ray-path is considered.

2.5.8.1 Vertical TEC calculation

To compute *NeQuick* vertical TEC, first compute all profile parameters *hmE*, *hmF*1, *hmF*2, *A*1, *A*2, *A*3, *B*2*bot*, *B*1*top* *B*1*bot*, *BEtop*, *BEbot*, *NmF*2, H_0, then compute the integration of the electron density (bottomside or topside) as function of height:

$$TEC = \int_{h_1}^{h_2} N(h)\mathrm{d}h \qquad \text{Eq. 139}$$

being:

$$h_1 = r_1 - R_E \qquad \text{Eq. 140}$$

$$h_2 = r_2 - R_E \qquad \text{Eq. 141}$$

2.5.8.1.1 Vertical TEC numerical integration

Inputs:

- Integration endpoints h_1 [km], h_2 [km]
- Target integration accuracy ε - relative difference between two integration steps, recommended maximum value $\varepsilon=10^{-3}$.
- Model parameters A_1 [10^{11} m^{-3}], A_2 [10^{11} m^{-3}], A_3 [10^{11} m^{-3}], *hm*F2 [km], *hm*F1 [km], *hm*E [km], *B2bot* [km], *B*1*top* [km], *B*1*bot* [km], *BEtop* [km], *BEbot* [km], *NmF*2 [m^{-3}], H_0 [km]

Output:

TEC [TECU]

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

$$N(h) = \begin{cases} \text{bottomside}\, N & \text{if } h \le hmF2 \\ \text{topside}\, N & \text{if } h > hmF2 \end{cases} \qquad \text{Eq. 142}$$

N(*h*) is computed using the algorithms described in 2.5.6.

Start the calculation using 8 points:

$$n = 8 \quad \text{Eq. 143}$$

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 integration speed at the expense of accuracy):

Calculate the integration intervals:

$$\Delta_n = \frac{h_2 - h_1}{n} \quad \text{Eq. 144}$$

$$g = 0.5773502691896 \cdot \Delta_n \quad \text{Eq. 145}$$

$$y = g_1 + \frac{\Delta_n - g}{2} \quad \text{Eq. 146}$$

$$GN_2 = \frac{\Delta_n}{2} \cdot \sum_{i=0}^{n-1} [N(y + i\Delta_n) + N(y + i\Delta_n + g)] \quad \text{Eq. 147}$$

Double the number of points:

$$n = 2n \quad \text{Eq. 148}$$

and define

$$GN_1 = GN_2 \quad \text{Eq. 149}$$

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

$$\text{if} \quad |GN_1 - GN_2| > \varepsilon |GN_1| \quad \text{Eq. 150}$$

then continue increasing the number of points, redefine GN_1 and repeat again.

When the test fails, the required accuracy has been reached, and the value of the integral is obtained by:

$$TEC = \left(GN_2 + \frac{GN_2 - GN_1}{15}\right) \times 10^{-13} \quad \text{Eq. 151}$$

2.5.8.2 *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.

2.5.8.2.1 Geometrical configuration

To simplify the formulation we assume that if α is an angle in [deg], $\tilde{\alpha}$ is the same angle in [rad]:

$$\tilde{\alpha} = \alpha \cdot DR \qquad \text{Eq. 152}$$

2.5.8.2.2 Zenith angle computation

Figure 3 indicates the geometry involved in the computation of the zenith angle ζ at P_1. Calculate:

$$\cos(\tilde{\delta}) = \sin(\tilde{\varphi}_1)\sin(\tilde{\varphi}_2) + \cos(\tilde{\varphi}_1)\cos(\tilde{\varphi}_2)\cos(\tilde{\lambda}_2 - \tilde{\lambda}_1) \qquad \text{Eq. 153}$$

$$\sin(\tilde{\delta}) = \sqrt{1 - \cos^2(\tilde{\delta})} \qquad \text{Eq. 154}$$

$$\tilde{\zeta} = \operatorname{atan2}\left(\sin(\tilde{\delta}), \cos(\tilde{\delta}) - \frac{r_1}{r_2}\right) \qquad \text{Eq. 155}$$

being δ 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 $(-\pi,\pi]$.

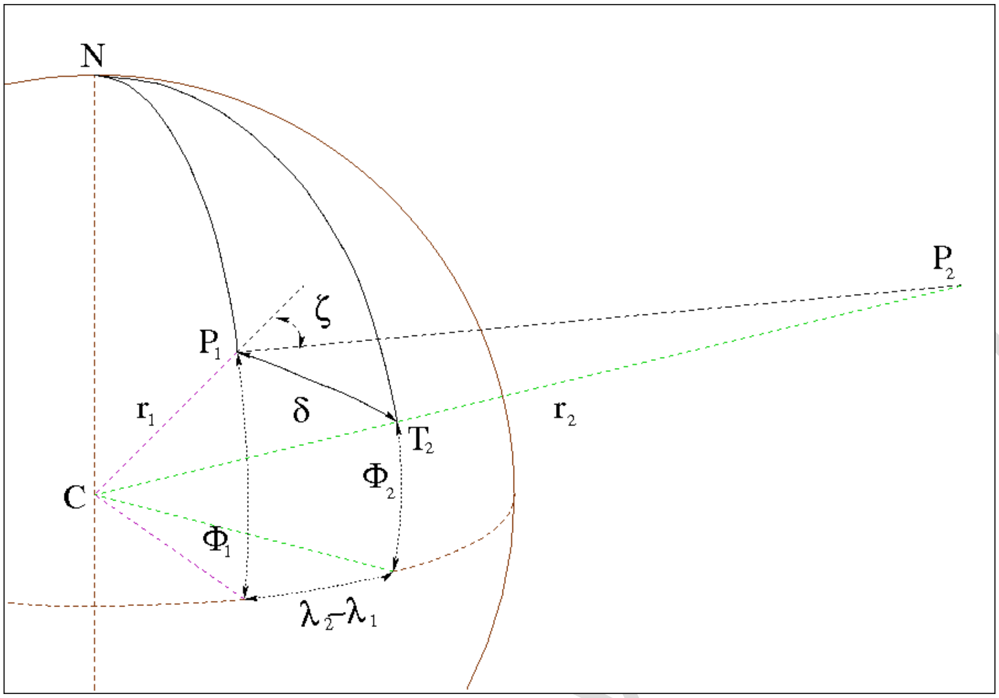


Figure 3. Geometry of zenith angle computation.

2.5.8.2.3 Ray-perigee computation

Figure 4 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:

$$r_p = r_1 \sin(\tilde{\zeta}) \quad \text{Eq. 156}$$

Calculate φ_p:

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

$$\varphi_p = \begin{cases} \zeta & \text{if } \varphi_1 > 0 \\ -\zeta & \text{if } \varphi_1 < 0 \end{cases} \quad \text{Eq. 157}$$

otherwise use

$$\sin(\tilde{\sigma}) = \frac{\sin(\tilde{\lambda}_2 - \tilde{\lambda}_1)\cos(\tilde{\varphi}_2)}{\sin(\tilde{\delta})} \quad \text{Eq. 158}$$

$$\cos(\tilde{\sigma}) = \frac{\sin(\tilde{\varphi}_2) - \cos(\tilde{\delta})\sin(\tilde{\varphi}_1)}{\sin(\tilde{\delta})\cos(\tilde{\varphi}_1)} \quad \text{Eq. 159}$$

$$\tilde{\delta}_p = \frac{\pi}{2} - \tilde{\zeta} \quad \text{Eq. 160}$$

$$\sin(\tilde{\varphi}_p) = \sin(\tilde{\varphi}_1)\cos(\tilde{\delta}_p) - \cos(\tilde{\varphi}_1)\sin(\tilde{\delta}_p)\cos(\tilde{\sigma}) \quad \text{Eq. 161}$$

$$\cos(\tilde{\varphi}_p) = \sqrt{1 - \sin^2(\tilde{\varphi}_p)} \quad \text{Eq. 162}$$

$$\tilde{\varphi}_p = \operatorname{atan2}(\sin(\tilde{\varphi}_p), \cos(\tilde{\varphi}_p)) \quad \text{Eq. 163}$$

Calculate λ_p:

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

$$\tilde{\lambda}_p = \begin{cases} \tilde{\lambda}_2 + \pi & \text{if } \tilde{\zeta} \geq 0 \\ \tilde{\lambda}_2 & \text{if } \tilde{\zeta} < 0 \end{cases} \quad \text{Eq. 164}$$

otherwise use

$$\sin(\tilde{\lambda}_1 - \tilde{\lambda}_p) = -\frac{\sin(\tilde{\sigma})\sin(\tilde{\delta}_p)}{\cos(\tilde{\varphi}_p)} \quad \text{Eq. 165}$$

$$\cos(\tilde{\lambda}_1 - \tilde{\lambda}_p) = \frac{\cos(\tilde{\delta}_p) - \sin(\tilde{\varphi}_1)\sin(\tilde{\varphi}_p)}{\cos(\tilde{\varphi}_1)\cos(\tilde{\varphi}_p)} \quad \text{Eq. 166}$$

$$\lambda_p = [\operatorname{atan2}(\sin(\tilde{\lambda}_1 - \tilde{\lambda}_p), \cos(\tilde{\lambda}_1 - \tilde{\lambda}_p)) + \tilde{\lambda}_1] \cdot RD \quad \text{Eq. 167}$$

being σ the azimuth of P_2 seen from P_1 and δ_p the Earth angle between P_1 and the ray-perigee P_P.

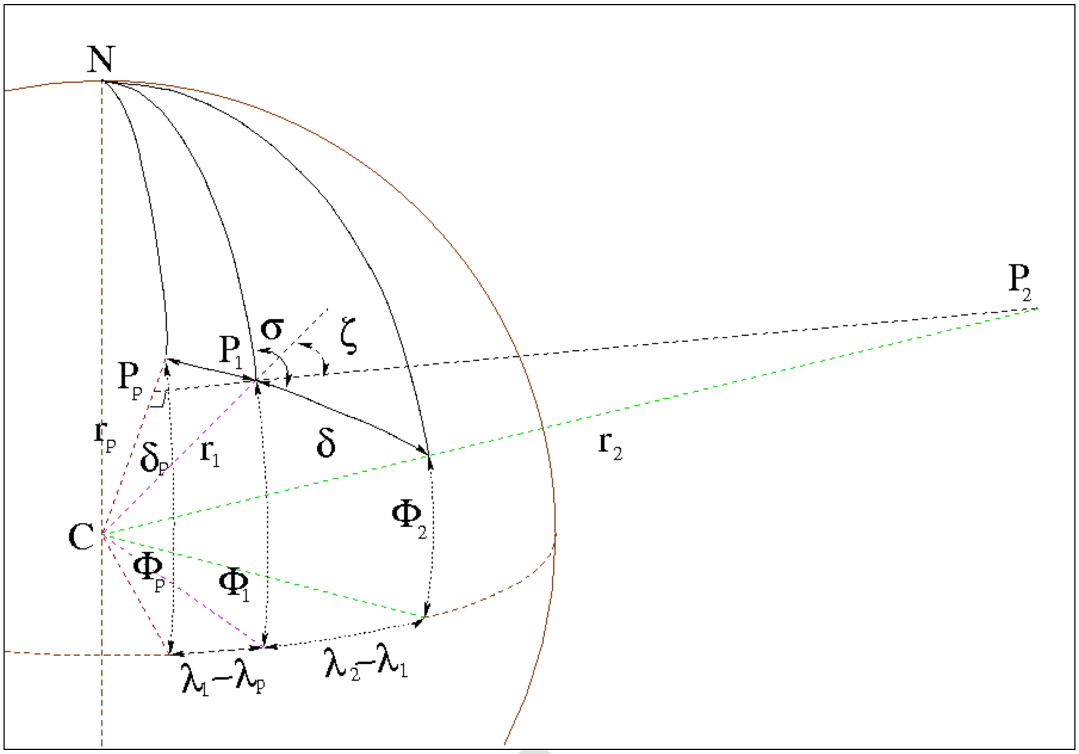


Figure 4. Geometry of ray perigee computation

2.5.8.2.4 Great circle properties

Compute the great circle angle ψ from ray-perigee to satellite:

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

$$\psi = |\varphi_2 - \varphi_p| \quad \text{Eq. 168}$$

otherwise use

$$\cos(\tilde{\psi}) = \sin(\tilde{\varphi}_p)\sin(\tilde{\varphi}_2) + \cos(\tilde{\varphi}_p)\cos(\tilde{\varphi}_2)\cos(\tilde{\lambda}_2 - \tilde{\lambda}_p) \quad \text{Eq. 169}$$

$$\sin(\tilde{\psi}) = \sqrt{1 - \cos^2(\tilde{\psi})} \quad \text{Eq. 170}$$

$$\tilde{\psi} = \text{atan2}(\sin(\tilde{\psi}), \cos(\tilde{\psi})) \quad \text{Eq. 171}$$

Compute sine and cosine of azimuth σ of satellite as seen from ray-perigee P_p:

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

$$\sin(\tilde{\sigma}_p) = 0 \quad \text{Eq. 172}$$

$$\cos(\tilde{\sigma}_p) = \begin{cases} -1 & \text{if } \varphi_p > 0 \\ 1 & \text{if } \varphi_p < 0 \end{cases} \quad \text{Eq. 173}$$

otherwise use

$$\sin(\tilde{\sigma}_p) = -\frac{\cos(\tilde{\varphi}_2)\sin(\tilde{\lambda}_2 - \tilde{\lambda}_p)}{\sin(\tilde{\psi})} \quad \text{Eq. 174}$$

$$\cos(\tilde{\sigma}_p) = -\frac{\sin(\tilde{\varphi}_2) - \sin(\tilde{\varphi}_p)\cos(\tilde{\psi})}{\cos(\tilde{\varphi}_p)\sin(\tilde{\psi})} \quad \text{Eq. 175}$$

2.5.8.2.5 Integration endpoints:

Indicating with s_1 and s_2 the distances of P_1 and P_2 respectively from the ray perigee compute:

$$s_1 = \sqrt{{r_1}^2 - {r_p}^2} \quad \text{Eq. 176}$$

$$s_2 = \sqrt{{r_2}^2 - {r_p}^2} \quad \text{Eq. 177}$$

2.5.8.2.6 Coordinates along the integration path: $c(h_s, \varphi_s, \lambda_s)$

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

Inputs:

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

Outputs:

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:

$$h_s = \sqrt{s^2 + {r_p}^2} - R_E \quad \text{Eq. 178}$$

being R_E the Earth mean radius.

Calculate great circle parameters:

$$\tan \tilde{\delta}_s = \frac{s}{r_p} \quad \text{Eq. 179}$$

$$\cos(\tilde{\delta}_s) = \frac{1}{\sqrt{1 + \tan^2(\tilde{\delta}_s)}} \quad \text{Eq. 180}$$

$$\sin(\tilde{\delta}_s) = \tan(\tilde{\delta}_s)\cos(\tilde{\delta}_s) \quad \text{Eq. 181}$$

Calculate φ_s:

$$\sin(\tilde{\varphi}_s) = \sin(\tilde{\varphi}_p)\cos(\tilde{\delta}_s) + \cos(\tilde{\varphi}_p)\sin(\tilde{\delta}_s)\cos(\tilde{\sigma}_p) \quad \text{Eq. 182}$$

$$\cos(\tilde{\varphi}_s) = \sqrt{1 - \sin^2(\tilde{\varphi}_s)} \quad \text{Eq. 183}$$

$$\varphi_s = \text{atan2}(\sin(\tilde{\varphi}_s), \cos(\tilde{\varphi}_s)) \cdot RD \quad \text{Eq. 184}$$

Calculate λ_s:

$$\sin(\tilde{\lambda}_s - \tilde{\lambda}_p) = \sin(\tilde{\delta}_s)\sin(\tilde{\sigma}_p)\cos(\tilde{\varphi}_p) \quad \text{Eq. 185}$$

$$\cos(\tilde{\lambda}_s - \tilde{\lambda}_p) = \cos(\tilde{\delta}_s) - \sin(\tilde{\varphi}_p)\sin(\tilde{\varphi}_s) \quad \text{Eq. 186}$$

$$\lambda_s = \left[\text{atan2}\left(\sin(\tilde{\lambda}_s - \tilde{\lambda}_p), \cos(\tilde{\lambda}_s - \tilde{\lambda}_p)\right) + \tilde{\lambda}_p\right] \cdot RD \quad \text{Eq. 187}$$

2.5.8.2.7 Slant TEC numerical integration

To compute slant TEC along a ray-path defined by its perigee coordinates, direction and end-point, a numerical integration algorithm is used. In *NeQuick* 1, a Gauss integration is used and is described as follows.

Inputs

- 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]
- *Az* coefficients: a_0, a_1, a_2
- Month mth
- *UT* [hours]

Output

- Slant TEC [TECU]

One possible numerical algorithm for slant TEC calculation is the following.

In the case of integration from ground to satellite (h_1<1000 km and h_2>2000 km) it is convenient to divide the integration path in three parts defining intermediate points s_a, s_b:

$$s_a = \sqrt{(R_E + 1000)^2 - {r_p}^2} \quad \text{Eq. 188}$$

$$s_b = \sqrt{(R_E + 2000)^2 - {r_p}^2} \quad \text{Eq. 189}$$

We have that $(R_E + 1000)^2 = 54334589.44$ and $(R_E + 2000)^2 = 70076989.44$.

The slant TEC becomes therefore:

$$TEC = \int_{s_1}^{s_a} N(s)\mathrm{d}s + \int_{s_a}^{s_b} N(s)\mathrm{d}s + \int_{s_b}^{s_2} N(s)\mathrm{d}s \quad \text{Eq. 190}$$

To compute each integral, the algorithm described in section 2.5.8.2.8 can be used as

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

Eq. 191

where the parameter ε indicates the integration accuracy. Here we assume:

- ε=0.001 for the integration between s_0 and s_a
- ε=0.01 for the integrations between s_a and s_b and between s_b and s_2

2.5.8.2.8 Gauss algorithm

Inputs:

- distances from the ray perigee of the first integration endpoint: g_1 [km]
- distances from the ray perigee of the second integration endpoint: g_2 [km]
- Target integration accuracy ε - relative difference between two integration steps, recommended maximum value $\varepsilon=10^{-3}$
- Ray-perigee parameters: r_p [km], $\sin(\tilde{\varphi}_p)$, $\cos(\tilde{\varphi}_p)$, $\sin(\tilde{\sigma}_p)$, $\cos(\tilde{\sigma}_p)$, λ_p [deg]
- *Az* coefficients: a_0, a_1, a_2
- Month mth
- UT [hours]

Output:

- TEC [TECU]

To be able to compute *NeQuick* 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 2.5.8.2.6:

$$\left(h(s),\varphi(s),\lambda(s)\right)=c\left(s,\sin\left(\tilde{\varphi}_p\right),\cos\left(\tilde{\varphi}_p\right),\sin\left(\tilde{\sigma}_p\right),\cos\left(\tilde{\sigma}_p\right),\lambda_p\right) \qquad \text{Eq. 192}$$

and being a_0, a_1, a_2, mth, UT fixed during the integration, in the following a simplified notation is used:

$$f(s):=N(h(s),\varphi(s),\lambda(s),a_0,a_1,a_2,\text{mth},\text{UT}) \qquad \text{Eq. 193}$$

Start the calculation using 8 points:

$$n = 8 \quad \text{Eq. 194}$$

Repeat the following computations until the target integration accuracy (ε) is obtained as follows:

Calculate the integration intervals:

$$\Delta_n = \frac{g_2 - g_1}{n} \quad \text{Eq. 195}$$

$$g = 0.5773502691896 \cdot \Delta_n \quad \text{Eq. 196}$$

$$y = g_1 + \frac{\Delta_n - g}{2} \quad \text{Eq. 197}$$

$$GN_2 = \frac{\Delta_n}{2} \cdot \sum_{i=0}^{n-1} [f(y + i\Delta_n) + f(y + i\Delta_n + g)] \quad \text{Eq. 198}$$

Double the number of points:

$$n = 2n \quad \text{Eq. 199}$$

and define

$$GN_1 = GN_2 \quad \text{Eq. 200}$$

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

$$\text{if} \qquad |GN_1 - GN_2| > \varepsilon |GN_1| \quad \text{Eq. 201}$$

then continue increasing the number of points, redefine GN_1 and repeat again steps above.

When the test fails, the required accuracy has been reached, and the value of the integral is obtained by:

$$TEC = \left(GN_2 + \frac{GN_2 - GN_1}{15} \right) \times 10^{-13} \quad \text{Eq. 202}$$

2.5.8.3 Alternative computational efficient TEC integration method

In Section 9.2.6 within Annex F, an alternative more computationally efficient integration method for calculating TEC along rays based on Kronrod G_7-K_{15}

adaptive quadrature method is presented. This method involves sampling values at 15 points and calculating the integration from them.

2.5.9 Clarification on coordinates used in *NeQuick*

The MODIP table grid file used by *NeQuick* is calculated using the IGRF model for the Earth's magnetic field. Strictly speaking, the coordinates derived from such a model are defined as Corrected Geomagnetic Coordinates (CGM), as defined in Annex B.

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.

Thereby, when *NeQuick* applies equations 204 and 205 given in Annex B for *I* and *MODIP* using CGM coordinates instead of the dipole geomagnetic coordinates, those concepts should be referred as Corrected Magnetic Dip (*I'*) and Corrected Modified Dip Latitude (*MODIP'*) respectively. 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.

2.6 Differences between NeQuick G and NeQuick 1 and NeQuick 2

This section summarizes, for information, the main algorithmic differences between the *NeQuick* implementation within *NeQuick G* and the one within ITU-R *NeQuick* 1 and *NeQuick* 2.

The most important difference between *NeQuick G* and versions 1 and 2 is related to the driving input parameter: for *NeQuick G* the driving parameter is the Effective Ionisation Level "*Az*", based on optimisation of global observations for real-time usage and broadcast in Galileo navigation message through three parameters; for *NeQuick 1* and *2,* the input is the monthly mean or 12-month running Solar Flux (intended for climatological usage).

2.6.1 Summary of differences with *NeQuick* 1

The main differences between *NeQuick G* and the ITU-R *NeQuick* 1 are:

- The *MODIP* file provided with the *NeQuick G* (see Annex D) and the *diplats* file provided with the ITU-R Fortran code correspond to different generation of the International Geomagnetic Reference Field (IGRF) model, being the one in *NeQuick G* newer;

- The location within the algorithm of the mapping from DIPLATS to *MODIP* has a small effect on the computed *MODIP* value for the location. The impact on computed STEC value can reach large values (maximum of ~5 TECU but rms of ~0.15 TECU);

- The change that has the greatest impact on computed STEC values is the calculation of Epstein amplitudes, E and F1 layer bottom and top thickness, and peak height. This change is accurately described in [5]. The reason to change the calculation of peak plasma frequency for F1 layer is also introduced in [5];

- The change to numerical integration method affects all STEC values to some extent (below 0.1 TECU rms). The rationale for this change is described within [6];

2.6.2 Summary of differences with *NeQuick* 2

The main differences between *NeQuick G* and *NeQuick 2,* included in ITU-R Recommendation P.681-13, are the following:

- The revision of the topside shape parameter *k*:

$$B_{top}^{F_2} = k \cdot B_{bot}^{F_2}$$

$$k = 3.22 - 0.0538 \cdot f_0F_2 - 0.00664 \cdot h_mF_2 + 0.113 \frac{h_mF_2}{B_{bot}^{F_2}} + 0.00257 \cdot R_{12}$$

$$k \geq 1$$

- ITU-R version in [3] incorporates a *MODIP* file obtained with an IGRF geomagnetic field consistent with the field when the CCIR files (used internally in the model in the same manner as for *NeQuick* 1) were generated. That means that the *MODIP* file is consistent with the *diplats* file from *NeQuick* 1 and it is based on IGRF from the 1970s whereas *NeQuick G* is based on a *MODIP* file from a recent IGRF generation. *NeQuick 2* authors often use also *MODIP* files from newer IGRF generations for other applications such as data assimilation.

3. Implementation Guidelines for User Receivers

This section describes practical guidelines for the implementation of the single-frequency ionospheric model previously described within Galileo user receivers.

3.1 Zero-valued coefficients and default Effective Ionisation Level

When all the Effective Ionisation Level ionospheric broadcast coefficients are set to zero:

$$a_{i0} = a_{i1} = a_{i2} = 0$$

the coefficients shall NOT be used for single-frequency ionospheric correction. In those cases, a default value shall be used for correction in the receiver:

$$a_{i0} = 63.7;\ a_{i1} = a_{i2} = 0$$

This default value represents the lowest Solar Flux value in average conditions that *NeQuick* is expected to operate on. In terms of the analytical expression relating twelve-month running Sun Spot Number and Solar Flux (see Eq. 206 within Section 5.2), it corresponds to a Sun Spot Number of 0. This value is considered adequate when no other solution is available, being still able to correct for a significant contribution of the ionospheric delay error.

3.2 Applicability and coherence of broadcast coefficients

In nominal conditions, a Galileo user receiver decoding the Galileo navigation message from N Galileo satellites simultaneously, with N>1, would receive at a given epoch, the same broadcast ionospheric coefficients for all N satellites. However, when the ionospheric parameters are updated, given that the message up-link to different satellites is not necessarily synchronised due to satellite visibility to Up-Link Stations (ULS), it may happen that some satellites broadcast the recently updated coefficients while others still broadcast the previous batch of coefficients. In such situation, any given user decoding the navigation message at any given moment does not have means to identify, which coefficients are newer and which are older. Also, it has no means to decide which coefficients to apply for correction of pseudo-ranges of different satellites (either each satellite applying corrections broadcast on its own navigation message or one single set of coefficients used for correcting all satellites). Nevertheless, the correction capability will still be achieved on statistical sense, independently of the adopted solution.

3.3 Effective Ionisation Level boundaries

Due to the fact that the Effective Ionisation Level is represented simply by a second-order degree polynomial as a function of MODIP, it may exceptionally happen that for some MODIP values, the Effective Ionisation Parameter becomes out of range. The operational range for the Effective Ionisation Parameter is between 0 and 400 s.f.u. and the following condition for out of range values should be used:

Given a user receiver U with

$$MODIP = \mu$$

and broadcast coefficients

$$(a_{i0}, a_{i1}, a_{i2})$$

Calculate Effective Ionisation Level: $Az_U = a_{i0} + a_{i1} \times \mu + a_{i2} \times \mu^2$

if $Az_U < 0$

$$Az_U = 0$$

if $Az_U > 400$

$$Az_U = 400$$

3.4 Integration of NeQuick G into higher level software

NeQuick G requires data from 13 files, as indicated in 2.5.3 and Appendix C. Although the convenience of integrating *NeQuick* G 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* G in the main target software at initialisation. All the rest can be separated into a library.

3.5 Computation rate of ionospheric corrections

The ionosphere variations in nominal conditions are fairly slow and for the majority of applications it is not required to re-compute the delay correction at high rates. For example, in most cases, for stationary receivers or pedestrian users, it may suffice to re-compute the corrections every 30 seconds.

4. Annex A – Applicable and Reference Documents

4.1 Applicable Documents

[1] *European GNSS (Galileo) Open Service Signal In Space Interface Control Document (OS SIS ICD)*, Issue 1.2, European Union, 2015

4.2 Reference Documents

[2] *Minimal Operational Performance Standards for GPS/WAAS Airborne Equipment*, RTCA-DO 229 C, Nov. 2001.

[3] ITU-R, *Ionospheric propagation data and prediction methods required for the design of satellite services and systems*, Rec. ITU-R P. 531-11, January 2012.

[4] ITU-R, *Choices if indices for long-term ionospheric predictions*, Rec. ITU-R P.371-8, 1999.

[5] Leitinger, R.; Zhang, M.-L.; Radicella, S.M., *An improved bottomside for the ionospheric electron density model NeQuick*, Ann. Geophys., Vol. 48, No. 3, p. 525-534, 2005.

[6] Knezevich, M.; Radicella, S.M., *Development of an ionospheric NeQuick model algorithm for GNSS receivers*, in NAVITEC 2004, Noordwijk, 2004.

[7] *Corrected Geomagnetic Coordinates*, [online], NASA Vitmo models, http://omniweb.gsfc.nasa.gov/vitmo/cgmm_des.html, 12 September 2013

[8] Rawer, K., *Meteorological and Astronomical influences on Radio Wave Propagation*, B. Landmark (ed.), Chapter 11. Propagation of Decameter Waves (H.F. Band), p. 221, Academic Press New York, 1963

[9] *International Geomagnetic Reference Field*, [online], IAGA Geomagnetic Field Modeling, http://www.ngdc.noaa.gov/IAGA/vmod/igrf.html, 12 September 2013

[10] Klobuchar, J., *An Ionospheric Time-Delay Algorithm for Single-Frequency GPS Users,* IEEE Transactions on Aerospace and Electronic Systems, AES-23 (3), 325–331, 1987

[11] R. Prieto-Cerdeira, R. Orus-Perez, E. Breeuwer, R. Lucas-Rodriguez, M. Falcone, *The European Way: Assessment of NeQuick Ionospheric Model for Galileo Single-Frequency Users*, GPS World, vol. 25, no. 6, pp. 53–58, 2014.

[12] Mosert de Gonzalez, M.; Radicella, S.M., *On a characteristic point at the base of F2 layer in the ionosphere*, Adv. Space Res., Vol. 10, Iss. 11, 1990

[13] Di Giovanni, G.; Radicella, S.M., *An analytical model of the electron density profile in the ionosphere*, Adv. Space Res., Vol. 10, No. 11, p. 27-30, 1990.

[14] Radicella, S.M.; Zhang, M.-L., *The improved DGR analytical model of electron density height profile and total electron content in the ionosphere*, Ann. Geofis, Vol. XXXVIII, No. 1, p. 35-41, 1995

[15] Radicella, S.M., Leitinger, R., *The evolution of the DGR approach to model electron density profiles*, Adv. Space Res., Vol. 27, No. 1, p. 35-40, 2001.

[16] http://utd500.utdallas.edu/ionosphere.htm

5. Annex B – Acronyms and Definitions

5.1 Acronyms

EC	European Commission
ESA	European Space Agency
FOC	Full Operational Capability
GIOVE	Galileo In Orbit Validation Element
GNSS	Global Navigation Satellite System
GSA	European GNSS Agency
ICTP	Abdus Salam International Center of Theoretical Physics
IOV	In Orbit Validation
IGS	International GNSS Service
MODIP	Modified Dip Latitude
NSL	Nottingham Scientific Ltd
OS	Open Service
RMS	Root Mean Square
SISICD	Signal In Space Interface Control Document
SFIono	Galileo Single Frequency Ionospheric Algorithm
s.f.u.	Solar Flux Units
STEC	Slant Total Electron Content
TEC	Total Electron Content
UERE	User Equivalent Ranging Error
VTEC	Vertical Total Electron Content

5.2 Definitions

CCIR files: numerical grid maps which describe the regular variation of the ionosphere. They are used to derive other variables such as critical frequencies and transmission factors. In *NeQuick*, *foF2* and *M(3000)F2* are derived from R_{12} and the CCIR files.

Corrected Geomagnetic Coordinates (CGM): coordinates relative to Earth's magnetic field when this is not approximated from a dipole but instead calculated from a model, usually the International Geomagnetic Reference Field (IGRF) model [7].

Critical frequency (*fo*): maximum frequency for which an electromagnetic wave is reflected at vertical incidence. It is usually referred to a particular layer, e.g., *foE*, *foF1* and *foF2*. At frequencies larger than the critical frequency, the radio wave penetrates the layer. It is expressed in Hz.

Effective Ionisation Level (*Az*): index that represents solar activity in the same fashion as *F10.7*. It is used to drive *NeQuick* model for a daily use in Galileo, instead of the monthly mean original behaviour when using *F10.7*. It is also expressed in s.f.u. This parameter is valid for the whole world. The *Az* parameter is a continuous function of the *MODIP* at the receiver location. *Az* is expressed by three coefficients:

$$Az = a_{i0} + a_{i1} \times MODIP + a_{i2} \times (MODIP)^2 \qquad \text{Eq. 203}$$

The three coefficients, a_{i0}, a_{i1} and a_{i2} are transmitted to the users in the Galileo navigation broadcast message.

Geomagnetic Coordinates: coordinates relative to Earth's magnetic field when this is approximated by a dipole. For the dipole approximation, the centred dipole axis cuts the Earth's surface at the south and north dipole poles. The dipole axis is inclined at about 11deg to the axis of rotation [7]. The plane through Earth's centre perpendicular to the dipole axis is called the dipole equator. The geomagnetic coordinates or dipole coordinates (dipole latitude and dipole longitude) are reckoned relative to the dipole axis, dipole poles and dipole equator. The relation between dipole and geographic coordinates can be found in [7].

Height of maximum electron density (*hm*): it is the height at which a layer has its peak of electron density, *e.g.*, *hmF2*. It is expressed in units of meters.

Ionosphere: part of the upper atmosphere where sufficient ionization can exist to affect the propagation of radio waves and lying between about 50 km and several Earth radii [7]. It consists of several regions or layers of ionisation (D, E, F1, F2 and top-side or plasmasphere). The level of ionization, which is caused by solar radiation, has diurnal, season and 11-year solar cycle variations and is dependent strongly on geographical locations and geomagnetic activity [7].

***M* factor (or *MUF* factor or transmission factor):** it is the Maximum Usable Frequency divided by the critical frequency for a given distance and layer, for instance, $M(3000)F2=MUF(3000)F2/foF2$. It is also related to the height of the maximum electron density in the layer.

Magnetic dip or magnetic inclination (*I*): it represents the angle of the geomagnetic field relative to the horizontal plane at a particular position. It is defined as [7]:

$$I = \tan^{-1}(2\times\tan(\phi)) \quad \text{Eq. 204}$$

where ϕ is the geomagnetic latitude (or dipole latitude).

Maximum Electron Density (*Nm*): it is the maximum electron density referred to an ionospheric layer, *e.g.*, *NmF2*. It is expressed in units of el m^{-2}.

Maximum Usable Frequency (*MUF*): highest frequency by which a radio wave can propagate between two points at a given distance by ionospheric propagation, independent of power. For instance, *MUF*(3000)F2 refers to the maximum usable frequency at F2 layer to reach a distance of 3000 km. It is expressed in Hz.

Modified Dip Latitude (*MODIP* or μ): defined by the following expression given by Rawer [8]:

$$\tan(\text{MODIP}) = I / \text{sqrt}(\cos\phi) \quad \text{Eq. 205}$$

where I is the magnetic dip and ϕ is geographic latitude. MODIP, I and ϕ are usually expressed in degrees.

Radio noise flux at 10.7 cm (*F10.7* or Φ): measure of the solar radio noise flux at a wavelength of 10.7 cm. Together with the Sun Spot Number, it is a typical index to represent solar ionization level in the ionosphere. It is expressed in solar flux units s.f.u. with 1 s.f.u. = 10^{-22} W m^{-2} Hz^{-1}.

Φ usually represents monthly mean values although daily values exist. Φ_{12} is the 12-month running radio noise flux calculated in a similar way as R_{12} (see Sun Spot Number). The recommended relationship between R_{12} and Φ_{12} is [4]:

$$\Phi_{12} = 63.7 + 0.728\times R_{12} + 8.9\times10^{-4}\times R_{12}^{2} \quad \text{Eq. 206}$$

Sun Spot Number (*SSN* or *R*): solar index that represents the occurrence of sunspots (notable dark spots on the solar surface). The daily Sun Spot Number is calculated as:

$$R = k\times(10\times g + s) \quad \text{Eq. 207}$$

where g is the number of sunspot groups, s is the number of observed individual sunspots and k is a correction factor which takes into account equipment and observer characteristics.

R_i, where i=[1,12], is the mean of the daily sunspot numbers for a single month, and R_{12} is the 12-month running sunspot number for a given month, calculated as:

$$R_{12} = [\ \textstyle\sum_{i=n-5:n+5} R_i + (R_{n+6} + R_{n-6})/2\ ]\ /12 \qquad \text{Eq. 208}$$

Total Electron Content (*TEC*): electron density integrated along a slant (or vertical) path between a satellite and a receiver. It is expressed in TEC units (TECU) where 1TECU=10^{16} el m^{-2}.

6. Annex C – Complementary Files (CCIR and MODIP)

This appendix includes all the complementary files required by *NeQuick* G. They are 12 CCIR files and one MODIP grid file. For the details on formatting and how to read these files see section 2.5.3.

Files:

ccir11.asc ccir12.asc ccir13.asc ccir14.asc ccir15.asc ccir16.asc

ccir17.asc ccir18.asc ccir19.asc ccir20.asc ccir21.asc ccir22.asc

modipNeQG_wrapped.asc

7. Annex D – *NeQuick* G Performance Results

In this section the performance of the *NeQuick* Galileo Single Frequency algorithm, referred as *NeQuick* G, is evaluated. Also, the performance of the model is compared with that of the GPS Ionospheric Correction Algorithm (ICA) algorithm, referred to as Klobuchar model.

Due to the complex nature of the ionospheric layer, different methods have been proposed to model it, such as empirical models, numerical maps, analytical and physical models. *NeQuick* G and Klobuchar belong to the empirical models class, which is based on the parameterization of a large amount of data collected for a long period of time.

NeQuick G is designed to reach a correction capability of at least 70% of the ionospheric code delay (RMS), with a lower STEC residual error bound of 20 TECU) for any location, time of day, season and solar activity, excluding periods where the ionosphere is largely disturbed due to, for instance, geomagnetic storms. Such performance has been assessed successfully using GPS data only and GPS+GIOVE data during GIOVE Experimentation.

Klobuchar model is defined as an SLM (Single Layer Model), because the ionosphere (i.e. its TEC) is assumed as concentrated in an infinitesimal layer, placed at an average altitude of 350 km from the Earth's surface [10], while *NeQuick* G model uses the peaks of the three main ionospheric regions (E, F1, and F2) as anchor points, as explained in section 2.5.

The Klobuchar model provides a different estimation of the day and night time ionospheric delay along the SIP vertical direction, starting from a set of eight coefficients (transmitted in the GPS navigation message). Night time corrections are assumed equal to a globally constant value of 5 ns (~ 1.5 m) for L1 carrier. Diurnal vertical delay, $\mathrm{T}_{\mathrm{iono}}^{\mathrm{V}}$ [sec], is modeled as a cosine function, while the delay along the LOS ($\mathrm{T}_{\mathrm{iono}}$ [sec]) is computed using a mapping function.

As an example of the behavior of the two models as a function of the time of day, the delay computed using Klobuchar and *NeQuick G* are plotted as a function of the satellite elevation and of UTC in Figure 5. For this example, in order to have a direct comparison between the two models, the delays computed using Klobuchar and *NeQuick* are compared with respect to the delay estimated using Global Ionospheric Map (GIM). The plots have been computed for a station in latitude [deg] 40.8234, longitude [deg] 14.2161, altitude [m] 122.6590 m, using GPS satellite PRN 11 and for day 16 of year 2010 characterized by quiet geomagnetic activity.

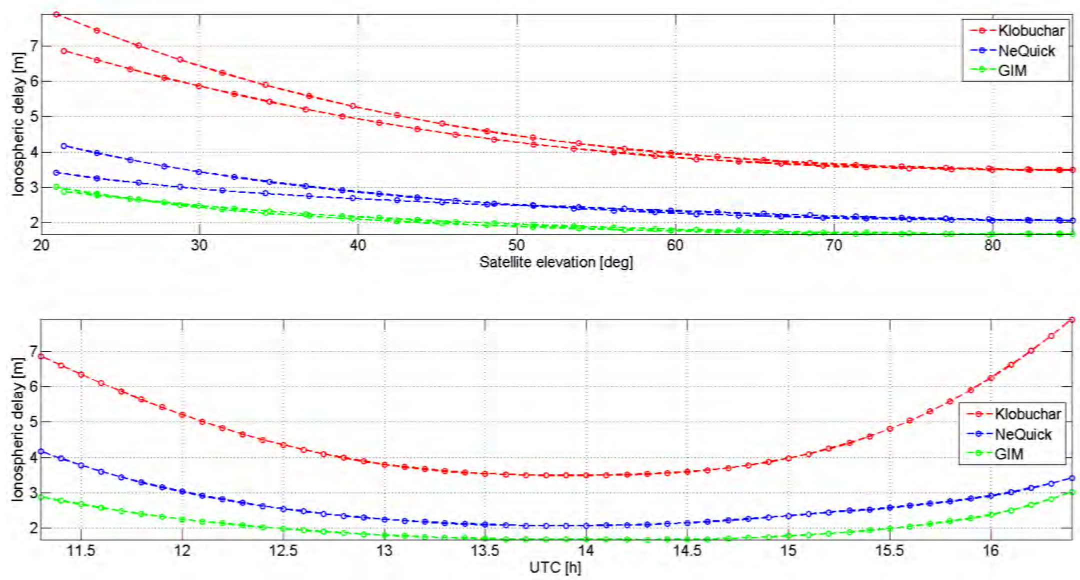


Figure 5. Ionospheric Delay vs. satellite elevation (upper) and of UTC (lower)

The first results using the correction algorithm with broadcast parameters from Galileo satellites have been performed during IOV in the period April 2013 to March 2014, including a more active secondary peak of solar activity during solar maximum of solar cycle 24 [11]. The Galileo broadcast data used for this test are *Az* coefficients broadcast by the four Galileo IOV satellites. It is important to remember that during this assessment, the IOV infrastructure is reduced with respect to the target Full Operational Capability, including for the ionospheric parameters: 4 IOV satellites (no other GNSS satellites are used in the estimation) and a reduced number of stations.

A performance assessment is performed against reference STEC estimated using dual-frequency observables from GPS on stations from the International GNSS Service (IGS); many stations distributed around the world were selected for the correction capability performance assessment. This result on 6 to 9 satellites for any epoch and with more than 120 stations per day assures a good global coverage for the test. As a reference for comparative purposes, in some cases, the results are compared to those obtained with the GPS ICA correction model (Klobuchar model) using the broadcast parameters from GPS satellites.

The daily RMS error and correction capability for each station was computed, resulting in most days reaching expected performance. An example is presented in Figure 6 including data for a disturbed and a quiet day. It is observed that even for the disturbed day example, the correction capability is above 70% except for some stations in the equatorial regions.

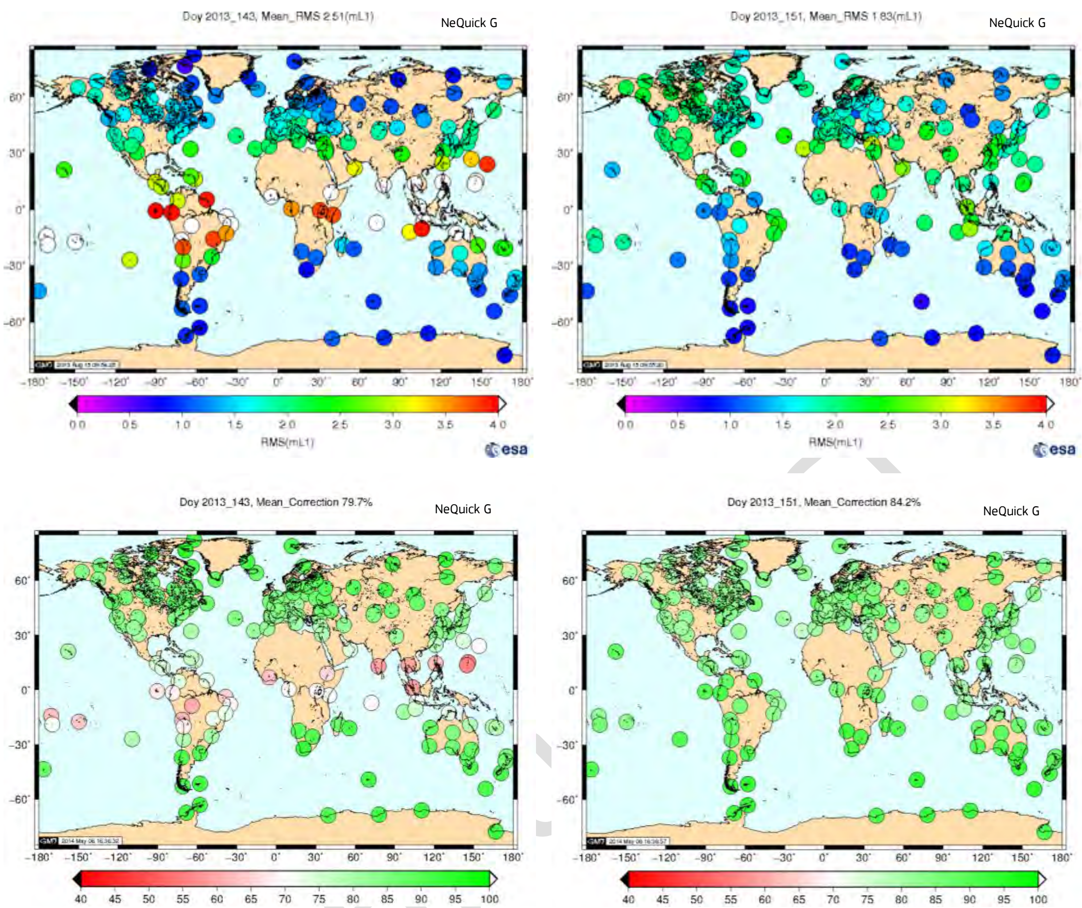


Figure 6. Performance of the Galileo Single Frequency Ionospheric correction when using the E11 satellite broadcast: (left) disturbed day, (right) quiet day, (top) RMS error in meters of L1, (bottom) correction capability in %

The evolution of the RMS residual error both for Galileo *NeQuick* G and GPS ICA from April 2013 to March 2014 are presented in Figure 7. In this plot, ionospheric activity at the equinoxes is clearly observed in the degradation of performance, and increased solar activity from October 2013 to March 2014 is also evident. The residual error of the Galileo correction model is already at the level of the expected capability for the full constellation. It also shows better performance as compared to the GPS Klobuchar model, especially at equatorial latitudes.

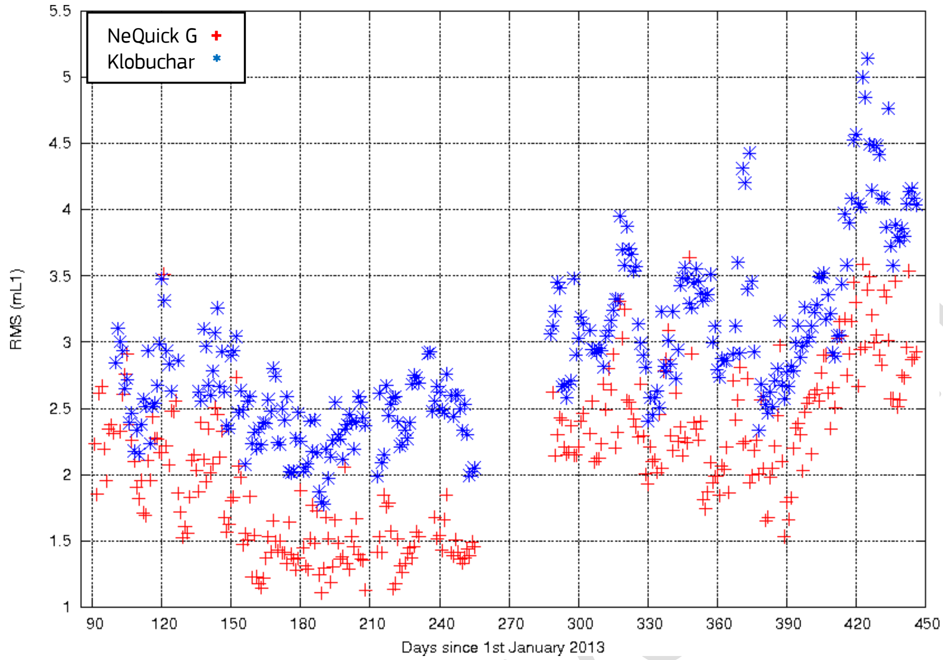
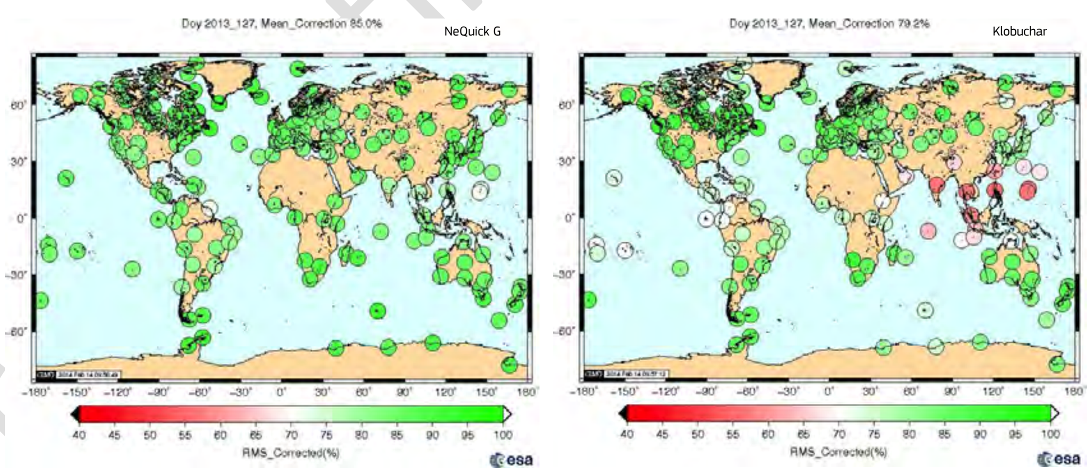


Figure 7. Global daily RMS ionospheric residual error [meters$_{L1}$] after correction with Galileo *NeQuick* G (red) and GPS ICA (blue) from April 2013 to March 2014

The level of correction capability for each station for the Galileo *NeQuick* G model with respect to GPS Klobuchar model are presented in Figure 8 for a quiet day in May 2013 and an active day during the Spring equinox in 2014.



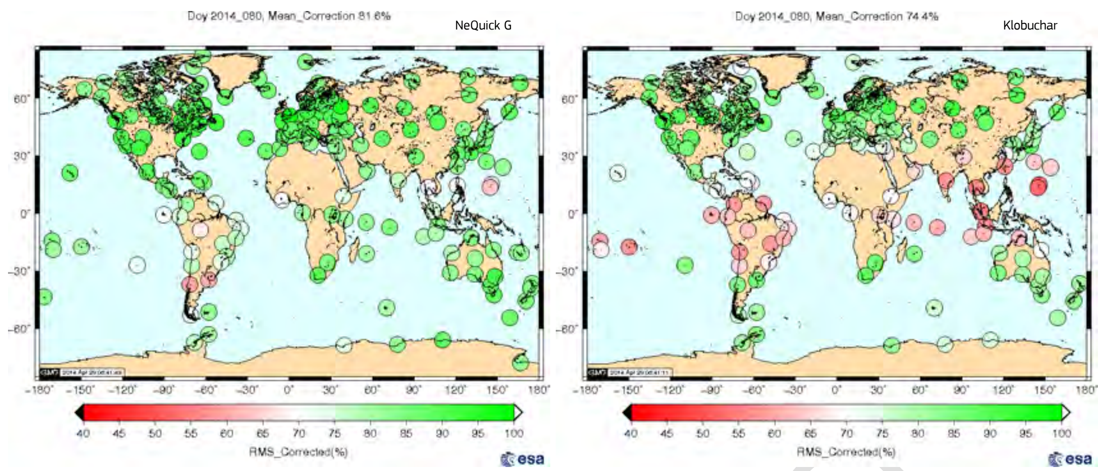


Figure 8. RMS correction capability (%, with a lower bound of 20 TECU) of Galileo *Nequick* G (left) and GPS ICA (right) correction models for day 127, 2013 (top) and day 80, 2014 (bottom)

In terms of residual single-frequency contribution to UERE after correction with single-frequency ionospheric algorithm, the results are presented in Figure 9 during the period between April and July 2013 and classified by MODIP region. The RMS error is plot as a function of the elevation angle, for all MODIP bands (SF1 to SF5) and the period April to July 2013 using the ionospheric coefficients broadcast by Galileo IOV satellites. The Galileo FOC UERE budget is included as reference.

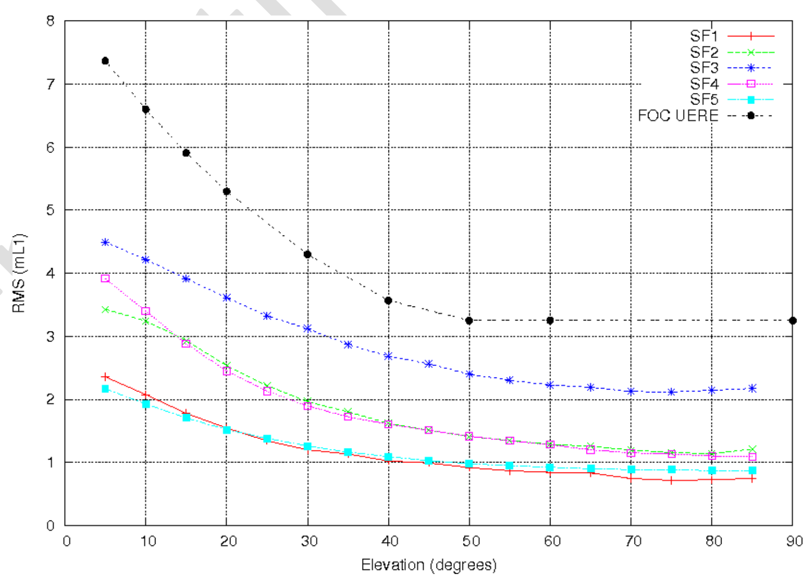


Figure 9. Residual single-frequency RMS error contribution to UERE after correction with single-frequency ionospheric algorithm [meters$_{L1}$]

8. Annex E – Input/Output Verification Data

The validation datasets provided in this appendix are valid for the current Galileo Single Frequency Ionospheric Model *NeQuick* G version.

8.1 Az coefficients (high solar activity)

a_0	a_1	a_2
236.831641	-0.39362878	0.00402826613

Input								Output
Month	Time (UTC)	Lon (station)	Lat (station)	Height (station)	Lon (sat)	Lat (sat)	Height (sat)	STEC (TECU)
4	0	297.66	82.49	78.11	8.23	54.29	20281546.18	20.40
4	0	297.66	82.49	78.11	-158.03	24.05	20275295.43	53.45
4	0	297.66	82.49	78.11	-30.86	41.04	19953770.93	25.91
4	4	297.66	82.49	78.11	-85.72	53.69	20544786.65	18.78
4	4	297.66	82.49	78.11	-130.77	54.40	20121312.46	20.00
4	4	297.66	82.49	78.11	140.68	35.85	19953735.00	37.81
4	8	297.66	82.49	78.11	-126.28	51.26	20513440.10	21.31
4	8	297.66	82.49	78.11	84.26	54.68	20305726.79	27.72
4	8	297.66	82.49	78.11	-96.21	37.33	19956072.48	24.13
4	12	297.66	82.49	78.11	81.09	35.20	20278071.03	49.30
4	12	297.66	82.49	78.11	175.57	51.89	19995445.72	26.61
4	12	297.66	82.49	78.11	4.25	53.43	20107681.66	29.21
4	16	297.66	82.49	78.11	14.89	32.88	20636367.33	37.95
4	16	297.66	82.49	78.11	-70.26	50.63	20043030.82	23.93
4	16	297.66	82.49	78.11	-130.60	49.21	20288021.34	26.97
4	20	297.66	82.49	78.11	-52.46	24.28	19831557.96	48.45
4	20	297.66	82.49	78.11	-165.78	35.06	20196268.24	41.71
4	20	297.66	82.49	78.11	168.73	52.58	20288372.95	27.79
4	0	307.19	5.25	-25.76	-89.48	-29.05	20081457.33	240.59
4	0	307.19	5.25	-25.76	-46.73	-24.08	19975517.42	152.19
4	0	307.19	5.25	-25.76	-99.26	34.47	20275286.46	204.92
4	4	307.19	5.25	-25.76	-46.61	54.84	20258938.89	124.39
4	4	307.19	5.25	-25.76	-85.72	53.68	20544786.61	140.16
4	4	307.19	5.25	-25.76	-18.13	14.17	20267783.18	95.52
4	8	307.19	5.25	-25.76	7.14	-19.55	20226657.45	26.47
4	8	307.19	5.25	-25.76	-48.38	-31.04	20069586.93	21.34
4	8	307.19	5.25	-25.76	-58.59	21.93	20008556.82	21.68
4	12	307.19	5.25	-25.76	-102.83	-40.74	20153844.84	169.86
4	12	307.19	5.25	-25.76	-0.60	10.75	20272829.17	178.43
4	12	307.19	5.25	-25.76	-120.35	11.00	20283503.35	146.95
4	16	307.19	5.25	-25.76	-70.26	50.63	20043030.72	198.43
4	16	307.19	5.25	-25.76	-72.73	-9.78	19936049.27	149.02
4	16	307.19	5.25	-25.76	-66.77	2.37	19986966.89	133.16
4	20	307.19	5.25	-25.76	-1.57	-7.90	20373709.74	255.31
4	20	307.19	5.25	-25.76	0.44	50.83	19975412.45	292.41
4	20	307.19	5.25	-25.76	10.94	44.72	20450566.19	336.74

8.2 Az coefficients (medium solar activity)

a_0	a_1	a_2
121.129893	0.351254133	0.0134635348

Input								Output
Month	Time (UTC)	Lon (station)	Lat (station)	Height (station)	Lon (sat)	Lat (sat)	Height (sat)	STEC (TECU)
4	0	40.19	-3.00	-23.32	76.65	-41.43	20157673.93	18.26
4	0	40.19	-3.00	-23.32	-13.11	-4.67	20194168.22	35.84
4	0	40.19	-3.00	-23.32	26.31	-39.04	20671871.64	17.18
4	4	40.19	-3.00	-23.32	79.33	-55.34	20679595.44	36.00
4	4	40.19	-3.00	-23.32	107.19	-10.65	19943686.06	76.77
4	4	40.19	-3.00	-23.32	56.35	47.54	20322471.38	38.01
4	8	40.19	-3.00	-23.32	7.14	-19.55	20226657.34	69.17
4	8	40.19	-3.00	-23.32	51.96	-1.90	20218595.37	59.53
4	8	40.19	-3.00	-23.32	89.22	-40.56	20055109.63	101.26
4	12	40.19	-3.00	-23.32	90.78	-28.26	20081398.25	127.83
4	12	40.19	-3.00	-23.32	35.75	-14.88	20010521.91	81.34
4	12	40.19	-3.00	-23.32	81.09	35.20	20278071.09	113.92
4	16	40.19	-3.00	-23.32	14.89	32.88	20636367.52	91.07
4	16	40.19	-3.00	-23.32	2.04	11.23	20394926.95	96.70
4	16	40.19	-3.00	-23.32	22.79	-35.87	20125991.19	71.45
4	20	40.19	-3.00	-23.32	54.11	3.15	20251696.28	48.06
4	20	40.19	-3.00	-23.32	95.06	17.94	20246498.07	77.64
4	20	40.19	-3.00	-23.32	-1.81	-52.00	20332764.38	50.10
4	0	115.89	-31.80	12.78	119.90	-8.76	19941513.27	24.84
4	0	115.89	-31.80	12.78	165.14	-13.93	20181976.57	43.94
4	0	115.89	-31.80	12.78	76.65	-41.43	20157673.77	19.90
4	4	115.89	-31.80	12.78	107.19	-10.65	19943685.24	46.25
4	4	115.89	-31.80	12.78	79.33	-55.34	20679595.29	46.10
4	4	115.89	-31.80	12.78	64.90	-17.58	20177185.06	64.59
4	8	115.89	-31.80	12.78	127.35	23.46	19837695.71	88.58
4	8	115.89	-31.80	12.78	89.22	-40.56	20055109.56	40.62
4	8	115.89	-31.80	12.78	148.31	-29.93	20109263.99	40.82
4	12	115.89	-31.80	12.78	90.78	-28.26	20081398.25	14.48
4	12	115.89	-31.80	12.78	133.47	-24.87	19975574.41	13.63
4	12	115.89	-31.80	12.78	166.97	-3.87	20196778.56	27.69
4	16	115.89	-31.80	12.78	124.09	-14.31	20100697.90	6.96
4	16	115.89	-31.80	12.78	154.31	-45.19	20116286.17	7.48
4	16	115.89	-31.80	12.78	-167.50	-43.24	20095343.13	13.87
4	20	115.89	-31.80	12.78	131.65	-31.56	20066111.12	4.36
4	20	115.89	-31.80	12.78	115.68	-52.78	20231909.06	4.35
4	20	115.89	-31.80	12.78	50.87	-50.69	20186511.77	6.97

8.3 Az coefficients (low solar activity)

a_0	a_1	a_2
2.580271	0.127628236	0.0252748384

Input								Output
Month	Time (UTC)	Lon (station)	Lat (station)	Height (station)	Lon (sat)	Lat (sat)	Height (sat)	STEC (TECU)
4	0	141.13	39.14	117.00	165.14	-13.93	20181976.50	36.44
4	0	141.13	39.14	117.00	85.59	36.64	20015444.79	14.24
4	0	141.13	39.14	117.00	119.90	-8.76	19941513.27	23.54
4	4	141.13	39.14	117.00	107.19	-10.65	19943685.88	77.49
4	4	141.13	39.14	117.00	38.39	51.98	20457198.52	29.28
4	4	141.13	39.14	117.00	-130.77	54.40	20121312.41	23.02
4	8	141.13	39.14	117.00	179.50	51.35	19967933.94	13.62
4	8	141.13	39.14	117.00	97.28	21.46	19941941.52	24.28
4	8	141.13	39.14	117.00	84.26	54.68	20305726.98	15.90
4	12	141.13	39.14	117.00	62.65	54.77	20370905.24	16.33
4	12	141.13	39.14	117.00	115.63	-1.28	20165065.92	11.05
4	12	141.13	39.14	117.00	81.09	35.20	20278071.22	14.25
4	16	141.13	39.14	117.00	124.09	-14.31	20100698.19	8.12
4	16	141.13	39.14	117.00	-130.60	49.21	20288020.98	6.45
4	16	141.13	39.14	117.00	161.97	13.35	20265041.53	4.69
4	20	141.13	39.14	117.00	84.18	36.59	19953853.27	6.06
4	20	141.13	39.14	117.00	54.67	51.65	20511861.27	8.28
4	20	141.13	39.14	117.00	-136.92	46.53	20309713.36	10.78
4	0	204.54	19.80	3754.69	165.14	-13.93	20181976.58	94.98
4	0	204.54	19.80	3754.69	179.32	9.92	20274303.54	60.87
4	0	204.54	19.80	3754.69	-144.16	-15.44	20007317.84	72.83
4	4	204.54	19.80	3754.69	-130.77	54.40	20121312.45	30.77
4	4	204.54	19.80	3754.69	-99.26	37.44	20066769.88	36.18
4	4	204.54	19.80	3754.69	-85.72	53.69	20544786.60	37.25
4	8	204.54	19.80	3754.69	178.35	-7.05	20372509.81	1.62
4	8	204.54	19.80	3754.69	-125.97	2.30	20251559.90	0.12
4	8	204.54	19.80	3754.69	179.50	51.35	19967934.29	0.56
4	12	204.54	19.80	3754.69	158.88	-12.61	20145417.20	1.77
4	12	204.54	19.80	3754.69	-146.53	22.03	20069033.97	0.64
4	12	204.54	19.80	3754.69	-153.30	-39.75	20672066.87	2.99
4	16	204.54	19.80	3754.69	-140.58	51.70	20455387.61	2.16
4	16	204.54	19.80	3754.69	-167.50	-43.24	20095343.11	3.11
4	16	204.54	19.80	3754.69	-164.50	27.08	20494802.61	1.22
4	20	204.54	19.80	3754.69	-172.71	-20.37	20225145.06	24.53
4	20	204.54	19.80	3754.69	-136.92	46.53	20309713.37	13.14
4	20	204.54	19.80	3754.69	-82.52	20.64	19937791.48	38.20

9. Annex F – *NeQuick* G Detailed Processing Model

This Section presents the Detailed Processing Model of the *NeQuick* version used for Galileo single frequency ionospheric correction: *NeQuick* G.

9.1 External Interfaces

9.1.1 Introduction

The main function for the *NeQuick* G model is named *NeQuick*. The *NeQuick* function has 3 interface parameters and a return value. It has one input structure that contains all input information necessary to execute the *NeQuick* function, one output variable that returns the computed STEC value, and one input/output structure that contains information that is useful to store between calls to the *NeQuick* function in order to optimise the processing. The return value is numerical value (enum), for which 0 indicates no error (E_OK) and 1 indicates a problem has occurred (E_ERROR). The input and output parameters are described in the following sections. Specific data structures internal to *NeQuick* are defined in section 9.3.

9.1.1.1 Inputs

In order to compute the slant TEC along a ray path, the *NeQuick* G model requires information on the properties of that ray path (start and end points), information on the properties of the ionosphere and geomagnetic model, and an indication of the time at which the values are required. This information is passed to the *NeQuick* function in the form of two data structures, each containing multiple parameters. The reason for combining the data in this way was simply to limit the number of parameters in the *NeQuick* function call and fulfil the required coding standards. The inputs are described in the following table.

Name	Type	Size	Range	Units	Description
pstNeQuickInputData	NeQuickInputData_st	1	Not defined	N/A	Input data required for NeQuick function. This includes position of ray start and end points, current time, MODIP grid and CCIR maps.

Table 3. NeQuick Function Input Data

9.1.1.2 Outputs

The NeQuick function outputs a single value, which is the computed TEC value along the ray path to the satellite. This output is described in the following table.

Name	Type	Size	Range	Units	Description
pdSTEC	double	1	Not defined	Electrons/m^2	Computed STEC value for ray from GSS to Sat

Table 4. NeQuick Function Output Data

9.1.1.3 Input/Output

Although not strictly necessary for the computations within the *NeQuick* function, for situations where the *NeQuick* function is called multiple times with very similar ray properties (e.g. month and R12 value) it is useful to store certain ray properties between function calls so that they do not have to be recomputed each time. In this way the processing is optimised and computation time is reduced. The current CCIR parameters are output from the *NeQuick* function and can be passed back in to the function on the next call in order to preserve these values. They are described in the following table.

Name	Type	Size	Range	Units	Description
pstCurrCCIR	CurrentCCIR_st	1	Not defined	N/A	Information required for computing f0F2 and M3000F2 coefficients for given time and R12 value

Table 5. NeQuick Function Output Data

It should be noted that prior to the first call of the *NeQuick* function the current CCIR values should be initialised to out of range values so that they will not unintentionally be used within the *NeQuick* function on the first call.

9.2 Modules

9.2.1 Introduction

This section contains a description of the internal processing within one implementation of the *NeQuick* G model. For each function, an explanation of the inputs and output variables is provided, along with a description of the internal processing within the function.

9.2.2 Function Overview

There are 24 separate functions. The hierarchy of the different functions is illustrated in Figure 10.

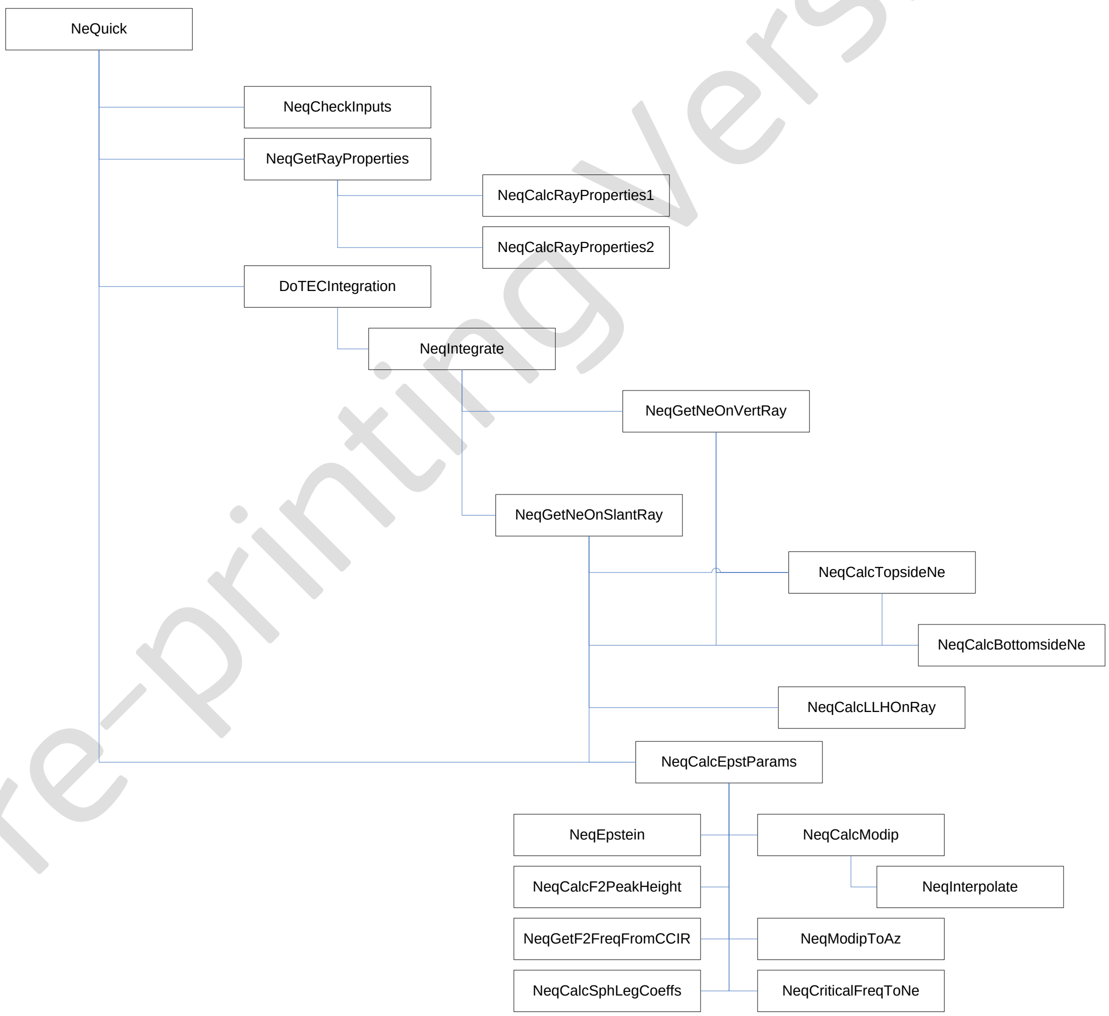


Figure 10. Overview of NeQuick Function Hierarchy

In addition to those functions shown in the figure, there are 3 utility functions (NeqSquared, NeqClipExp and NeqJoin), which are general computation functions that are used by a number of the main *NeQuick* functions. Each of these functions is described in more detail in the following sections. To ease the understanding of the detailed processing model, the sections are divided by module.

9.2.3 NeQuick.c Module

9.2.3.1 Function "NeQuick" (Main Function)

Purpose

This is the main *NeQuick* function that returns the total electron content (TEC) between two points. It calculates STEC between the points or VTEC if point 2 is almost directly above point 1. Point 1 is the sensor station. Point 2 is the satellite. The value is calculated from the month, time and Az value, as well as the two positions.

Interfaces

Calls: NeqCheckInputs, NeqGetRayProperties, NeqCalcEpstParams and DoTECIntegration

The external inputs and outputs to the NeQuick function are described in section 9.1.

Internal Processing

Initialise *pstIntegrateData* structure with *pstNeQuickInputData* and p*stCurrCCIR*

Call function "*NeqCheckInputs(pstNeQuickInputData)*" to check that inputs are within ranges

if_1 inputs are ok
    Set stP1 = Receiver position (Latitude/Longitude in degrees, Height in km)
    Set stP2 = Sat position (Latitude/Longitude in degrees, Height in km)

    Call function "*NeqCalcModip*" to calculate Modip at the Receiver position stP1: *ModipRx*.
    Call function "*NeqModiptoAz*" to calculate Az_U using *ModipRx*
        Assign Az_U to *pstNeQuickInputData->dAzBase*

    Check that *Az is within ranges:*
        if $a_{i0} = a_{i1} = a_{i2} = 0$, $Az_U = 63.7$,
        if $Az_U<0$, $Az_U=0$
        if $Az_U>400$, $Az_U=400$

    Call function "*NeqGetRayProperties*" to calculate ray properties and to check if ray is valid

    if_2 ray is valid
        Calculate slant distance of each point
        $stP1.dS = \sqrt{stP1.dR^2 - stRay.dR^2}$
        $stP2.dS = \sqrt{stP2dR^2 - stRay.dR^2}$
        If_3 point 1 (receiver) is below the Earth's surface, stP*1.dH < 0*
            Set point 0 to be at Earth's surface, *stP0.dH = 0*
        Else_3
            Set point 0 equal to point 1, set *stP0.dH = stP1.dH*

end if$_3$
Set radius of point 0, *stP0.dR = stP0.dH + Re*
(where *Re* = 6371.2km) $stP0.dS = \sqrt{stP0.dR^2 - stRay.dR^2}$
Initialise *pactual* (the 'current' position)

$$stPactual.dH = stP2.dH$$

$$stPactual.dLat = stP1.dLat$$

$$stPactual.dLng = stP1.dLng$$

Calculate sine and cosine of delta (solar declination)

$$amrad = (0.9856 * \left(mth * 30.5 - 15 + \frac{18-ut}{24}\right) - 3.289) * DEGTORAD$$

$$sdelta = 0.39782 * \sin(amrad + (1.916 * \sin(amrad) + 0.02 * \sin(2 * amrad) + 282.634) * DEGTORAD))$$

$$cdelta = \sqrt{1 - sdelta^2}$$

(where mth = pstNeQuickInputData.siMonth and *ut* = pstNeQuickInputData.dUT)

If$_3$ vertical ray (point 2 directly above point 1, *i.e.* Ray Perigee radius < 0.1)
  Call function "*NeqCalcEpstParams*" to calculate Ionosphere parameters
  Set bVertical flag = TRUE
End **if$_3$**
Update *pstIntegrateData* structure with stP1 and stP2 location for integration.
Call function "*DoTECIntegration*" to perform TEC integration along ray (*dTEC)*

Else$_2$
  Return an error

else$_1$
  Return an error

End **if$_1$**

pdSTEC = 1000*dTEC (convert internal TEC value to correct units for output from *NeQuick* function, factor of 1000 since integration is done based on heights in km)

9.2.3.2 NeQuick internal function "NeqCheckInputs"

Purpose

This function checks some of the input data to *NeQuick* to determine if values are within range and will allow a valid TEC value to be computed or not. Note that MODIP values, CCIR maps and Kronrod tolerances are not checked because in *NeQuick* G these values should already be checked before being passed to *NeQuick*.

Interfaces

Called by: *NeQuick*

Calls: -

Inputs:

Name	Type	Size	Range	Units	Description
pstNeQuickInputData	NeQuickInputData_st	1	Not defined	N/A	Input data required for *NeQuick* function. This includes position of ray start and end points, current time, MODIP grid and CCIR maps.

Table 6. NeqCheckInputs Function Input Data

Outputs: -

Return value: bError

Internal Processing

Check latitude of receiver is between -90 and 90 degrees.

Check latitude of satellite is between -90 and 90 degrees.

Check the month is between 1 and 12.

Check the Time is between 0 and 24hrs.

Check the number of coefficients is not less than 1.

Check the array with the coefficients is not null.

If any checked fail, set the return bError value to TRUE.

9.2.3.3 NeQuick internal function "DoTECIntegration"

Purpose

This function checks whether the ray is vertical or slant, and where the start and end points are located in regards to the different integration points, before passing the appropriate information to the integration function.

Interfaces

Called by: *NeQuick*

Calls: NeqIntegrate

Inputs:

Name	Type	Size	Range	Units	Description
pstIntegrateData	IntegrateData_st	1	Not defined	N/A	Information required for *NeQuick* integration.
bVert	Boolean	1	False, True	N/A	Flag indicating whether ray is vertical or not
stP0	SPoint_st	1	Not defined	N/A	Positional information for initial point 0.
pdNmax	double	1	Not defined	electrons /m³	Maximum Ne (F2 peak)
pstLayers	LayerProperties_st	1	Not defined	N/A	Current Ionospheric properties

Table 7. DoTECIntegration Function Input Data

Outputs:

Name	Type	Size	Range	Units	Description
pdTEC	Double	1	Not defined	Electrons *km/m³	Total electron content

Table 8. *NeQuick* Function Output Data

Internal Processing

Set integration tolerance in case Kronrod G17-K15 integration is used.

(Default tolerance values, 0.001 below 1000 km and 0.01 above 1000 km). Increasing tolerance increases integration speed at the expense of accuracy.)

Get height of points P0 (initial point), P1 (receiver) and P2 (satellite)

If$_1$ bVert = TRUE
    HeightP0 = stP0.dH
    HeightP1 = stP1.dH
    HeightP2 = stP2.dH
Else$_1$
    HeightP0 = stP0.dS
    HeightP1 = stP1.dS
    HeightP2 = stP2.dS
End **if$_1$**
(where *stP1* and *stP2* are obtained from *pstIntegrateData* structure)
Check if ray path crosses either of the integration break points and split up integration accordingly (it is assumed that P1 is always lower than P2)
If$_1$ stP2.dH <= 1000km (i.e. start and end point both below 1st break point)
    Call NeqIntegrate with start point of HeightP0, end point of HeightP2.
Else$_1$
    Set slant distance of 1st integration breakpoint S1a at 1000km.
    If$_2$ stP2.dH <= 2000km (end point below 2nd break point)
        If$_3$ stP1.dH >= 1000km (start point above 1st break point)

Start and end points are both between 1st and 2nd break points.
Call NeqIntegrate with start point of HeightP1, end point of HeightP2.

Else$_3$

Ray path crosses 1st integration break point.
Call NeqIntegrate with start point of HeightP0, end point S1a.
Call NeqIntegrate with start point of S1a, end point of HeightP2.
Sum the TEC values from the two NeqIntegrate calls

End **if$_3$**

Else$_2$

If$_3$ stP1.dH >= 2000km (start point above 2nd break point)
Start and end points are both above 2nd break point.
Call NeqIntegrate with start point of HeightP1, end point of HeightP2.

Else$_3$

Set slant distance of 2nd integration breakpoint S1b at 2000km
If$_4$ stP1.dH >= 1000km (start point above 1st break point)
Ray path crosses 2nd integration break point.
Call NeqIntegrate with start point of HeightP1, end point of S1b.
Call NeqIntegrate with start point of S1b, end point of HeightP2.
Sum the TEC values from the two NeqIntegrate calls

Else$_4$

Ray path crosses 1st and 2nd integration break points.
Call NeqIntegrate with start point of HeightP0, end point of S1a.
Call NeqIntegrate with start point of S1a, end point of S1b.
Call NeqIntegrate with start point of S1b, end point of HeightP2.
Sum the TEC values from the three NeqIntegrate calls

End **if$_4$**

End **if$_3$**

End **if$_2$**

End **if$_1$**

9.2.4 NeqCalcModipAz.c Module

9.2.4.1 NeQuick internal function "NeqCalcModip"

Purpose

This function uses the current latitude and longitude to calculate the corresponding modified dip latitude (MODIP) value. The MODIP grip should be 'pre-wrapped' at edges and poles.

Interfaces

Called by: NeQuick, NeqCalcEpstParams

Calls: NeqInterpolate

Inputs:

Name	Type	Size	Range	Units	Description
dLat	double	1	-90 to 90	deg	Latitude of point
dLng	double	1	-180 to 180	deg	Longitude of point
pstModip	MODIP_st	1	Not defined	N/A	Structure containing grid of modified dip latitude values.

Table 9. NeqCalcModip Function Input Data

Return value: dReturn – calculated MODIP value at given location

Internal Processing

```
Check Latitude is within ±90 degrees
if1 dLat <= -90
        Set computed Modip value dReturn = -90
Else if1 dLat >= 90
        Set computed Modip value dReturn = 90
Else

        Define properties of MODIP grid (lat step = 5 deg, lon step = 10 deg)
        constants describing grid lngp=36,dlatp=5,dlngp=10

        Obtain lon grid square (sj) and position in that square (dj)

        lng1=(dLng + 180) / dlngp
        sj = floor(lng1) - 2
        dj=lng1 – floor(lng1)

        Adjust for sign and wrap to grid if required
        If2 (sj < 0)
                sj=sj + lngp
        end if2
        if2 (sj > (lngp - 3))
                sj=sj - lngp;
        end if2

        Obtain lat grid square (si) and position in that square (di)

        lat1=(dLat + 90) / dlatp + 1
        si=floor(lat1 - 1e-6) - 2
        di = lat1 - si - 2

        Interpolate across lat grid to obtain values at 4 lon points on lat line
        For2 k=1;k <= 4;++k
                For3 j=1;j <= 4;++j
                        z1[j - 1]=pstModip[si + j][sj + k + 1]
                end for3
                z[k - 1]= NeqInterpolate (z1,di)
        end for2

        Interpolate for lon value using these 4 points using NeqInterpolate (z,dj);
        Set computed Modip value dReturn
End if1
```


9.2.4.2 NeQuick internal function “NeqInterpolate”

Purpose

This function performs third order interpolation. It is used when calculating the modified dip latitude value. Input z[4] is -1,0,1,2 point values, x is position to interpolate to.

Interfaces

Called by: NeqCalcModip

Calls: none

Inputs:

Name	Type	Size	Range	Units	Description
pdZ	double	[4]	Not defined	N/A	Array containing the -1, 0, 1 and 2 point values
dDeltaX	double	1	Not defined	N/A	Position to interpolate to (offset from 0pt to 1pt)
pstModip	MODIP_st	1	Not defined	N/A	Structure containing grid of modified dip latitude values.

Table 10. NeqInterpolate Function Input Data

Return value: dIntZ – the interpolated value at the point

Internal Processing

if₁ dDeltaX is small,

return = zero point, pdZ[1]₁

else₁

Interpolate

$g_0 = pdZ[2] + pdZ[1]$
$g_1 = pdZ[2] - pdZ[1]$
$g_2 = pdZ[3] + pdZ[0]$
$g_3 = (pdZ[3] - pdZ[0]) / 3$

$a_0 = 9 * g_0 - g_2$
$a_1 = 9 * g_1 - g_3$
$a_2 = g_2 - g_0$
$a_3 = g_3 - g_1$

Δx=2*dDeltaX – 1

return = $\frac{1}{16}\sum_{j=0}^{3} a_j \Delta x^j$

end if₁

9.2.4.3 NeQuick internal function “NeqModipToAz”

Purpose

This function calculates Az from the provided coefficients and modified dip latitude value (μ). If only one non-zero coefficient (a_0) is provided then Az = a_0 (no dependency with MODIP). This function evaluates:

$$Az = \sum_i a_i \mu^i$$

If Az < 0 then Az=0

Interfaces

Called by: NeQuick

Calls: none

Inputs:

Name	Type	Size	Range	Units	Description
dModip	double	1	-90 to 90	deg	Modified dip latitude of point
siNumCoeff	double	1	>=1	N/A	Number of Az coefficients
pdCoeff	double	[numCoeffs]	>0	flux units/deg^j	Az coefficients

Table 11. NeqCalcModip Function Input Data

Return value: dFlx – the computed Az value at the given modified dip latitude

Internal Processing

return $Az = \sum_i a_{i-1}\mu^{i-1}$(or zero if Az<0),

where:

i = siNumCoeff

a_i = pdCoeff

μ = dModip

9.2.5 NeqGetRayProperties.c Module

9.2.5.1 NeQuick internal function "NeqGetRayProperties"

Purpose

This function obtains the properties of the ray and checks if it is a valid ray. 'Ray' is straight line passing through p1 and p2. *pstRay->dLat, pstRay->dLng* and *pstRay->dR* are co-ordinates of ray perigee, i.e. point on ray closest to centre of Earth.

Interfaces

Called by: main function

Calls: NeqCalcRayProperties1, NeqCalcRayProperties2

Inputs/Outputs:

Name	Type	Size	Range	Units	Description
pstP1	SPoint_st	1	Not defined	N/A	Positional information for Point 1 (receiver)
pstP2	SPoint_st	1	Not defined	N/A	Positional information for Point 2 (satellite)

Table 12. NeqGetRayProperties Function Input/Output Data

Outputs:

Name	Type	Size	Range	Units	Description
pstRay	SPoint_st	1	Not defined	N/A	Positional information for ray perigee
pdZeta	double	1	-90 to 90	deg	Zenith angle of point 2 seen from point 1
pdSinSig	double	1	-1 to 1	N/A	Sine of ray azimuth
pdCosSig	double	1	-1 to 1	N/A	Cosine of ray azimuth

Table 13. NeqGetRayProperties Function Output Data

Return value: bError – Set to FALSE if no error and TRUE if there is an error.

Internal Processing

Check if ray is vertical

if$_1$ point 2 is directly above point 1, |*pstP2->dLat – pstP1->dLat* |<10^{-5} and | *pstP2->dLng – pstP1->dLng* |<10^{-5}

Set point 2 longitude to be **exactly** the same as point 1 longitude:

pstP2->dLng = pstP1->dLng

end if$_1$

Calculate ranges of points 1 and 2 from the centre of the earth

$$\mathrm{pstP1\text{->}\,dR} = \mathrm{pstP1\text{->}\,dH} + Re$$

$$\mathrm{pstP2\text{->}\,dR} = \mathrm{pstP2\text{->}\,dH} + Re$$

(*Re* = 6371.2km)

call function "NeqCalcRayProperties1" to calculate properties of the ray itself.

if$_1$ invalid ray, i.e. |*pdZeta*| > *90.0* and *pstRay->dR* < *Re*

Set return value bError to TRUE

end if$_1$

if$_1$ ray is not vertical, *pstRay->dR >= 0.1*

Call function "NeqCalcRayProperties2" to calculate additional ray properties.

end if$_1$

Return bError = FALSE

9.2.5.2 NeQuick internal function "NeqCalcRayProperties1"

Purpose

This function calculates the properties of the ray. It does not calculate as many properties if the ray is vertical, as they are not needed. 'Ray' is straight line passing through p1 and p2. *pstRay->dLat, pstRay->dLng* and *pstRay->dR* are co-ordinates of ray perigee, i.e. point on ray closest to centre of Earth.

Interfaces

Called by: NeqGetRayProperties

Calls: none

Inputs/Outputs:

Name	Type	Size	Range	Units	Description
pstP1	SPoint_st	1	Not defined	N/A	Positional information for Point 1 (receiver)
pstP2	SPoint_st	1	Not defined	N/A	Positional information for Point 2 (satellite)

Table 14. NeqCalcRayProperties1 Function Input/Output Data

Outputs:

Name	Type	Size	Range	Units	Description
pstRay	SPoint_st	1	Not defined	N/A	Positional information for ray perigee
pdZeta	double	1	-90 to 90	deg	Zenith angle of point 2 seen from point 1

Table 15. NeqCalcRayProperties1 Function Output Data

Return value: -

Internal Processing

Check if ray is vertical

if$_1$ point 2 is directly above point 1, |*pstP2->dLat – pstP1->dLat* |<10^{-5} and | *pstP2->dLng – pstP1->dLng* |<10^{-5}

Set the ray latitude and longitude to be the same as point 1, pstR*ay->dLat = pstP1->dLat* and pstR*ay->dLng = pstP1->dLng*
Set the ray to have no slant, *pstRay->dR = 0* and *pdZeta = 0*

else$_1$

Calculate (and store) sine and cosine of point 1 and point 2 latitudes,

pstP1->dSinLat, pstP1->dCosLat, pstP2->dSinLat, pstP2->dCosLat

Calculate temporary variables*: CosDl12, SinDl12, CosDel*, *SinDel*

$CosDl12 = \cos((pstP2 \rightarrow dLng - pstP1 \rightarrow dLng) * DEGTORAD)$

$SinDl12 = \sin((pstP2 \rightarrow dLng - pstP1 \rightarrow dLng) * DEGTORAD)$

$CosDel = pstP1 \rightarrow dSinLat * pstP2 \rightarrow dSinLat + pstP1 \rightarrow dCosLat * pstP2 \rightarrow dCosLat * CosDl12$

$SinDel = \sqrt{1 - CosDel^2}$

Calculate (and store) *pdZeta* (zenith angle of p2, seen from p1)

$pdZeta = \mathrm{atan}_2\left(SinDel, CosDel - \frac{pstP1 \rightarrow dR}{pstP2 \rightarrow dR}\right)$

Calculate temporary variables, *sdelp*, *cdelp*, *sphp*, *cphp*

$$SinSigp = \frac{SinDl12 * pstP2 -> dCosLat}{SinDel}$$

$$CosSigp = \frac{pstP2 -> dSinLat - CosDel * pstP1 -> dSinLat}{SinDel * pstP1 -> dCosLat}$$

$$Delp = \frac{\pi}{2} - pdZeta$$

$$SinDelp = \sin(Delp)$$

$$CosDelp = \cos(Delp)$$

$$SinPhp = pstP1 -> dSinLat * CosDelp - pstP1 -> dCosLat * SinDelp * CosSigp$$

$$CosPhp = \sqrt{1 - SinPhp^2}$$

Calculate ray perigee latitude

$$pstRay -> dLat = \mathrm{atan}_2(SinPhp, CosPhp) * RADTODEG$$

(NB this returns identical result to *asin(SinPhp)*)

Calculate ray perigee longitude

$$SinLamp = \frac{-1 * SinSigp * SinDelp}{CosPhp}$$

$$CosLamp = \frac{CosDelp - pstP1 -> dSinLat * SinPhp}{pstP1 -> dCosLat * CosPhp}$$

$$pstRay -> dLng = \mathrm{atan}_2(SinLamp, CosLamp) * RADTODEG + pstP1 -> dLng$$

Calculate radius of ray perigee

$$pstRay -> dR = pstP1 -> dR * \sin(pdZeta)$$

Convert zeta to degrees, *pdZeta = pdZeta * RD*

end if$_1$

9.2.5.3 NeQuick internal function "NeqCalcRayProperties2"

Purpose

This function calculates the sine and cosine of end point latitudes and azimuth. It is only called for slanted rays.

Interfaces

Called by: NeqGetRayProperties

Calls: none

Inputs:

Name	Type	Size	Range	Units	Description
pstRay	SPoint_st	1	Not defined	N/A	Positional information for ray perigee

Table 16. NeqCalcRayProperties2 Function Input Data

Inputs/Outputs:

Name	Type	Size	Range	Units	Description
pstP2	SPoint_st	1	Not defined	N/A	Positional information for Point 2 (satellite)

Table 17. NeqCalcRayProperties2 Function Input/Output Data

Outputs:

Name	Type	Size	Range	Units	Description
pstP1	SPoint_st	1	Not defined	N/A	Positional information for Point 1 (receiver)
pdSinSig	double	1	-1 to 1	N/A	Sine of ray azimuth
pdCosSig	double	1	-1 to 1	N/A	Cosine of ray azimuth

Table 18. NeqCalcRayProperties2 Function Output Data

Return value: -

Internal Processing

Calculate sine and cosine of end point latitudes (using ray perigee latitude for point 1)

$$pstP1->dSinLat = \sin(pstRay->dLat * DEGTORAD)$$
$$pstP1->dCosLat = \cos(pstRay->dLat * DEGTORAD)$$
$$pstP2->dSinLat = \sin(pstP2->dLat * DEGTORAD)$$
$$pstP2->dCosLat = \cos(pstP2->dLat * DEGTORAD)$$

Calculate difference in longitude of ray end points

$$DeltaLong = (pstP2->dLng - pstRay->dLng) * DEGTORAD$$

Check if latitude of lower end point is ±90 degrees – would cause divide by zero error later on

if$_1$ $||pstRay\text{->}dLat| - 90| < 10^{-10}$

Set sine of azimuth, pd*SinSig = 0*

If$_2$ positive latitude, *pstRay->dLat > 0*

Set cosine of azimuth, pd*CosSig = -1*

else$_2$

Set cosine of azimuth, pd*CosSig = 1*

end if$_2$

else$_1$

Calculate sine and cosine of angular distance between ends of ray (psi)

$$CosPsi = pstP1->dSinLat * pstP2->dSinLat +$$
$$pstP1->dCosLat * pstP2->dCosLat * \cos(DeltaLong)$$
$$SinPsi = \sqrt{1 - CosPsi^2}$$

Calculate sine and cosine of azimuth

$$pdSinSig = \frac{pstP2 \rightarrow dCosLat * \sin(DeltaLong)}{SinPsi}$$

$$pdCosSig = \frac{pstP2 \rightarrow dSinLat - pstP1 \rightarrow dSinLat * CosPsi}{SinPsi * pstP1 \rightarrow dCosLat}$$

end if$_1$

9.2.6 NeqIntegrate.c Module

9.2.6.1 NeQuick internal function "NeqIntegrate"

Purpose

Integration function for calculating TEC along rays using Kronrod G_7-$K_{15adaptive}$ quadrature method. 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.

Interfaces

Called by: DoTECIntegration

Calls: Self (**recursive**), NeqGetNeOnVertRay, NeqGetNeOnSlantRay

Inputs:

Name	Type	Size	Range	Units	Description
pstIntegrateData	IntegrateData_st	1	Not defined	N/A	Data required for computation of integrated TEC value
dH1	double	1	Not defined	km	Height of point 1
dH2	double	1	Not defined	km	Height of point 2
siCurrentLevel	Integer	1	>=0	N/A	Current level of integration recursion

Table 19. NeqIntegrate Function Input Data

Outputs:

Name	Type	Size	Range	Units	Description
pstPactual	SPoint_st	1	Not defined	N/A	Positional information for current integration point.

Table 20. NeqIntegrate Function Output Data

Inputs/Outputs:

Name	Type	Size	Range	Units	Description
pdNmax	double	1	Not defined	electrons/m^3	Maximum Ne (F2 peak)
pstLayers	LayerProperties_st	1	Not defined	N/A	Current Ionospheric properties

Table 21. NeqIntegrate Function Input/Output Data

Return value: dReturn – calculated TEC value

Internal Processing

Set Kronrod integration coefficients (G_7-K_{15})

Set weights for K15 sample points

```
wi[15]={  0.022935322010529224963732008058970,
          0.063092092629978553290700663189204,
          0.104790010322250183839876322541518,
          0.140653259715525918745189590510238,
          0.169004726639267902826583426598550,
          0.190350578064785409913256402421014,
          0.204432940075298892414161999234649,
          0.209482141084727828012999174891714,
          0.204432940075298892414161999234649,
          0.190350578064785409913256402421014,
```

0.16900472663926790282658342659855 0,
0.14065325971552591874518959051023 8,
0.10479001032225018383987632254151 8,
0.06309209262997855329070066318920 4,
0.02293532201052922496373200805897 0 }

Set weights for G7 sample points

wig[7]={ 0.12948496616886969327061143267908 2,
0.27970539148927666790146777142378 0,
0.38183005050511894495036977548897 5,
0.41795918367346938775510204081632 7,
0.38183005050511894495036977548897 5,
0.27970539148927666790146777142378 0,
0.12948496616886969327061143267908 2 }

Set at what points the samples are used in integration process

xi[15]= { -0.99145537112081263920685469752632 9,
-0.94910791234275852452618968404785 1,
-0.86486442335976907278971278864092 6,
-0.74153118559939443986386477328078 8,
-0.58608723546769113029414483825873 0,
-0.40584515137739716690660641207696 1,
-0.20778495500789846760068940377324 5,
0,
0.20778495500789846760068940377324 5,
0.40584515137739716690660641207696 1,
0.58608723546769113029414483825873 0,
0.74153118559939443986386477328078 8,
0.86486442335976907278971278864092 6,
0.94910791234275852452618968404785 1,
0.99145537112081263920685469752632 9 }

Calculate the midpoint, *hh* and the half difference, *h2*

h2=(dH2-dH1)/2

hh=(dH2+dH1)/2

Initialise integration results *intk* and *intg*, and G7 counter Gind

Loop through the G15 and K7 integration points

for$_1$ *i=0;i<15;++i*

*x=h2*xi[i]+hh*

if$_2$ *ray is vertical*

y=NeqGetNeOnVerticalRay at x

else$_2$

y=NeqGetNeOnSlantRay at x

end if$_2$

Accumulate on to the k15 total

*intk= intk + y * wi[i]*

if$_2$ *this is a G7 point (every other point – i.e. modulus of i/2==1)*

*intg = intg + y * wig[Gind]*

Gind = Gind + 1

end if₂

end for₁

Complete the calculation of the integration results
*Intk = intk * h2*
*Intg = intg * h2*

Check if the result is within tolerance
if₁*(fabs((intk - intg)/intk) <= tolerance || fabs(intk - intg) <= tolerance)*
Result is within tolerance so set return value = intk
else if₁ current level = MaxRecurse (max recursion level reached)
Can do not further integration
Set return value = intk as best guess
else₁
Result is not within tolerance
Split portion into two equal halves (from dH1 to dH1+h2 and from dH1+h2 to dH2 with h2=(dH2-dH1)/2) and call NeqIntegrate on each new portion
Sum the return values from the two NeqIntegrate calls and set as return value
end if₁

9.2.7 NeqGetNeOnVertRay.c Module

9.2.7.1 NeQuick internal function "NeqGetNeOnVerticalRay"

Purpose

This function returns electron density at a specified point along a vertical ray.

Interfaces

Called by: NeqIntegrate

Calls: NeqCalcTopsideNe, NeqCalcBottomsideNe

Inputs:

Name	Type	Size	Range	Units	Description
dH	double	1	Not defined	km	Height of point
pstLayers	LayerProperties_st	1	Not defined	N/A	Current Ionospheric properties

Table 22. NeqGetNeOnVertRay Function Input Data

Inputs/Outputs:

Name	Type	Size	Range	Units	Description
pdNmax	double	1	Not defined	electrons/m^3	Maximum Ne (F2 peak)

Table 23. NeqGetNeOnVertRay Function Input/Output Data

Return value: dReturn – Calculated electron content at specified height

Internal Processing

if$_1$ specified height is above F2 peak, dH > *PeakHeight[F2]*
Call function "*NeqCalcTopsideNe*" to compute dReturn
else$_1$
Call function "*NeqCalcBottomsideNe*" to compute dReturn
end if$_1$
Return dReturn

9.2.8 NeqGetNeOnSlantRay.c Module

9.2.8.1 NeQuick internal function "NeqGetNeOnSlantRay"

Purpose

This function returns electron density at the specified point along a slanted ray.

Interfaces

Called by: NeqIntegrate

Calls: NeqCalcLLHOnRay, NeqCalcEpstParams, NeqCalcTopsideNe, NeqCalcBottomsideNe

Inputs:

Name	Type	Size	Range	Units	Description
dS	double	1	Not defined	km	Distance of the point along the ray
pstNeQuickInputData	NeQuickInputData_st	1	Not defined	N/A	Input data required for NeQuick Function
pstGeom	GeometryData_st	1	Not defined	N/A	Geometry data for ray

Table 24. NeqGetNeOnSlantRay Function Input Data

Outputs:

Name	Type	Size	Range	Units	Description
pdNmax	double	1	Not defined	electrons/m^3	Maximum Ne (F2 peak)
pstPactual	SPoint_st	1	Not defined	N/A	Positional information for current integration point.
pstLayers	LayerProperties_st	1	Not defined	N/A	Current Ionospheric properties

Table 25. NeqGetNeOnSlantRay Function Output Data

Inputs/Outputs:

Name	Type	Size	Range	Units	Description
pstCurrCCIR	CurrentCCIR_st	1	Not defined	N/A	Information required for computing F0F2 and M3000F2 coefficients for given time and R12 value.

Table 26. NeqGetNeOnSlantRay Function Input/Output Data

Return value: dReturn – electron content value at specified point

Internal Processing

Call function "*NeqCalcLLHOnRay*" to adjust position information for current position along ray, *pstPactual.*

all function "*NeqCalcEpstParams*" to recalculate ionosphere information now that the latitude and longitude have changed.

if$_1$ current height is above F2 peak height, pstP*actual->dH > PeakHeight[F2]*
Call function "*NeqCalcTopsideNe*" to compute dReturn
else$_1$
Call function "*NeqCalcBottomsideNe*" to compute dReturn
end if$_1$
Return dReturn

9.2.8.2 *NeQuick internal function "NeqCalcLLHOnRay"*

Purpose

This function sets the latitude, longitude and height of the current position along the ray *Pactual* according to the specified slant position *s* (the distance along the slanted ray).

Interfaces

Called by: NeqGetNeOnSlantRay

Calls: none.

Inputs:

Name	Type	Size	Range	Units	Description
dS	double	1	Not defined	km	Distance of the point along the ray
pstRay	SPoint_st	1	Not defined	N/A	Information for ray perigee
pstP1	SPoint_st	1	Not defined	N/A	Information for Point 1
dSinSig	double	1	-1 to 1	N/A	Sine of ray azimuth
dCosSig	double	1	-1 to 1	N/A	Cosine of ray azimuth

Table 27. NeqCalcLLHOnRay Function Input Data

Outputs:

Name	Type	Size	Range	Units	Description
pstPactual	SPoint_st	1	Not defined	N/A	Positional information for current integration point.

Table 28. NeqCalcLLHOnRay Function Output Data

Return value: -

Internal Processing

Calculate trig of angle at centre of earth between lines to ray perigee and point on ray (Del):

$$TanDel = \frac{dS}{pstRay->dR}$$

$$CosDel = \frac{1}{\sqrt{1+TanDel^2}}$$

$$SinDel = TanDel * CosDel$$

Calculate latitude

$$arg = pstP1->dSinLat * CosDel + pstP1->dCosLat * SinDel * dCosSig$$

$$pstPactual \rightarrow dLat = \mathrm{atan2}\left(arg, \sqrt{1-arg^2}\right) \cdot RD$$

(NB asin(arg) would give same result)

Calculate longitude

$$CLong = \mathrm{atan2}(SinDel \cdot dSinSig \cdot pstP1 \rightarrow dCosLat, CosDel - pstP1 \rightarrow dSinLat \cdot arg) \cdot RD$$

$$pstPactual->dLng = CLong + pstRay->dLng$$

Calculate height

$$pstPactual->dH = \sqrt{dS^2 + pstRay->dR^2} - Re$$

(with Re=6371.2)

9.2.9 NeqCalcEpstParams.c Module

9.2.9.1 NeQuick internal function "NeqCalcEpstParams"

Purpose

This function calculates the values of ionospheric properties for the current latitude, longitude, time, etc.

The properties calculated are:

- M3000
- f0E, f0F1, f0F2
- PeakHeight
- Amp
- TopThick
- BotThick

The following are also calculated but not used outside this function:

- Az, R12,
- Nm[E], Nm[F1], Nm[F2]

Interfaces

Called by: NeQuick and NeqGetNeOnSlantRay.

Calls: NeqCalcModip, NeqModipToAz, NeqGetF2FreqFromCCIR, NeqCriticalFreqToNe, NeqCalcF2PeakHeight, NeqCalcSphLegCoeffs, NeqEpstein NeqClipExp, NeqJoin, NeqSquared

Inputs:

Name	Type	Size	Range	Units	Description
pstNeQuickInputData	NeQuickInputData_st	1	Not defined	N/A	Input data required for NeQuick Function
pstPactual	SPoint_st	1	Not defined	N/A	Positional information for current integration point.
dSinDelta	double	1	-1 to 1	N/A	Sine of angle of declination of sun
dCosDelta	double	1	-1 to 1	N/A	Cosine of angle of declination of sun

Table 29. NeqCalcEpstParams Function Input Data

Outputs:

Name	Type	Size	Range	Units	Description
pdNmax	double	1	Not defined	electrons/m^3	Maximum Ne (F2 peak)
pstLayers	LayerProperties_st	1	Not defined	N/A	Current Ionospheric properties

Table 30. NeqCalcEpstParams Function Output Data

Inputs/Outputs:

Name	Type	Size	Range	Units	Description
pstCurrCCIR	CurrentCCIR_st	1	Not defined	N/A	Information required for computing F0F2 and M3000F2 coefficients for given time and R12 value.

Table 31. NeqCalcEpstParams Function Input/Output Data

Return value: -

Internal Processing

Calculate MODIP at current longitude and latitude.

modip = NeqCalcModip

Retrieve Az at user Receiver position: Az = pstNeQuickInputData->dAzBase

Calculate R12 sunspot number from solar flux (Az):

$$R12 = \sqrt{1123.6 \cdot (Az - 63.7) + 167273} - 408.99$$

Initialise pdNmax to -1.0 to force NeqCalcTopsideNe function to call NeqCalcBottomsideNe

Check if month or solar flux has changed since spherical Legendre coefficients were last computed

```
If1 month or R12 have changed
        Load working matrices with CCIR coefficients
        RR2 = R12/100
        RR1 = 1-RR2
        Blend high and low activity cases in ration RR2:RR1 as
        (NB: assume index starts at 1,if index starts at 0, change accordingly)
        for1 i=1:13
                for2 j=1:76
                        CurrCCIR.pdF0F2((j-1)*13 + i) = CCIR.pdF2(siMonth,i,j,1) * RR1 +
                        CCIR.pdF2(siMonth,i,j,2)* RR2;
                end for2
        end for1

        for1 i=1:9
                for2 j=1:49
                        CurrCCIR.pdM3000F2( (j-1)*9 + i) = CCIR.pdM3000(siMonth,i,j,1)* RR1 +
                        CCIR.pdM3000(siMonth,i,j,2)*RR2;
                end for2
        end for1

        Set current R12 and Month

        Call NeqCalcSphLegCoeffs with current time and blended CCIR information
        else If1 time has changed
                Call NeqCalcSphLegCoeffs with current time and blended CCIR information
end If1
```

Calculate $\sin^n$(modip) array for n=0 to 11, and cosine of latitude

Get F2 layer data from CCIR
For f0F2, call NeqGetF2FreqsFromCCIR function with foF2 working arrays
For M3000, call NeqGetF2FreqsFromCCIR function with M3000 working arrays

Calculate local time and map back into 24 hour period
xlt = ut + pstPactual->dLng / 15.0
if$_1$*(xlt < 0)*
        xlt=xlt + 24.0
else if$_1$*(xlt >= 24)*
        xlt=xlt - 24.0
end if$_1$

Calculate Solar Zenith angle
CosChi=sin(*pstPactual->dLat* * DegreesToRadians) *dSinDelta + cos(*pstPactual->dLat* * DegreesToRadians) * dCosDelta * cos(pi * (12 - xlt) / 12)
chi=atan2(sqrt(1 - CosChi * CosChi),CosChi) * RadiansToDegrees
chi0 = 86.23

Set season flag (seas):
Jan-Feb: -1
Mar-Apr: 0
May-Aug: 1
Sep-Oct: 0
Nov-Dec: -1

Estimate foE and f0F1

- The model for foE adopted is based on the solar zenith angle law. An exponential transition between "day" and "night" is used which ensures differentiability.
- For daytime the model takes foF1 = 1.4*foE, for night-time foF1= 0, using the same exponential day-night transition as for foE. In addition foF1 is reduced by 15% if too close to foF2.

ee=**NeqClipExp**(0.3 * lat)
seas= seas*(ee-1)/(ee+1)
chin=**NeqJoin**(90.0 - 0.24 * **NeqClipExp**(20.0 - 0.2 * chi),chi,12,chi-chi0)
sfac=(1.112 - 0.019 * seas) * **sqrt**(**sqrt**(Az))
fa=sfac * **NeqClipExp**(log(cos(chin * DR)) * 0.3)

Calculate E peak plasma frequency
f0E=**sqrt**(fa * fa + 0.49)

Calculate F1 peak plasma frequency and set to zero if negligible
f0F1=**NeqJoin**(1.4*f0E,0,1000.0,f0E-2) (Titheridge's Formula f0F1 = 1.4 f0F2)
f0F1=**NeqJoin**(0,f0F1,1000.0,f0E-f0F1)
f0F1=**NeqJoin**(f0F1,0.85*f0F1,60.0,0.85*f0F2-f0F1)
if$_1$ *(f0F1 < 1e-6)*
        f0F1=0
end if$_1$

Calculate peak electron densities from critical frequencies for the F2, F1 and E layers.
Nm[F2] = **NeqCriticalFreqToNe**(*foF2*)
Nm[F1] = **NeqCriticalFreqToNe**(*foF1*)

Nm[E] = **NeqCriticalFreqToNe***(foE)*
(F2, F1 and E correspond to indices 0, 1 and 2 respectively)

Calculate height of electron density peaks for the layers. NB F2 peak is calculated each time, E layer peak is fixed at 120km and the F1 peak is set halfway between them.
PeakHeight[F2]=peakh()
PeakHeight[E]=120km
PeakHeight[F1]=(PeakHeight[F2]+ PeakHeight[E])/2

Calculate density gradient at base of F2 layer (10^9 m^-3 km^-1) (see [12])
*NdHmx=0.01 * exp(-3.467 + 0.857 * log(f0F2 * f0F2) + 2.02 ** **log***(M3000))*

Calculate Bottom-side thickness parameters (see [13]
*BotThick[F2]=0.385 * Nm[F2] / NdHmx*
TopThick[F1]=0.3(PeakHeight[F2] – PeakHeight[F1])*
BotThick[F1]=.5(PeakHeight[F1] – PeakHeight[E])*
TopThick[E]=BotThick[F1]
if$_1$*(TopThick[E] < 7)*
    TopThick[E]=7
end if$_1$
BotThick[E]=5

Calculate Epstein function amplitudes
The construction of the vertical profile is based on "anchor" points related to the ionospheric characteristics of the main layers routinely scaled from the ionograms: foF2, M(3000)F2, foF1 and foE. The basic equations of the model are:

$$N(h) = NF2(h) + NF1(h) + NE(h)$$

$$N(h) = \frac{4Nm*F2}{(1+\exp(\frac{h-hmF2}{B_2}))^2}\exp(\frac{h-hmF2}{B_2}) + $$
$$+\frac{4Nm*F1}{(1+\exp(\frac{h-hmF1}{B_1}))^2}\exp(\frac{h-hmF1}{B_1}) + \frac{4Nm*E}{(1+\exp(\frac{h-hmE}{B_E}))^2}\exp(\frac{h-hmE}{B_E})$$

where:

$$Nm*F2 = NmF2 - 0.1NmF1$$
$$Nm*F1 = NmF1 - NF2(hmF1)$$
$$Nm*E = NmE - NF1(hmE = 120km) - NF2(hmE = 120km)$$

The values of Nm are derived from the critical frequencies read in the ionograms. The peak height of the F2 layer hmF2 is calculated from M(3000)F2 and the ratio foF2/foE, the F1 peak height hmF1 is modelled in terms of NmF1 and the geomagnetic dip of the location and the E peak height is fixed at 120 km.

The algorithm provides continuity to the function N(h) taking into account the exponential nature of the equations describing the model, using as an auxiliary function FEpst defined by:
FEpst(X,Y,Z,W)=X*fexp((W-Y)/Z)/(1+fexp((W-Y)/Z))^2

*Amp[F2]=4 * Nm[F2]*
*Amp[F1]=4 * Nm[F1]*
*Amp[E]=4 * Nm[E]*

```
if_1(f0F1 < 0.5)
        Amp[F1]=0
        Amp[E]=4 * (Nm[E]
                - NeqEpstein(Amp[F2], PeakHeight[F2], BotThick[F2], PeakHeight[E]))
else_1
        for_2(i=0;i<5;++i){
           Amp[F1]=4*(Nm[F1]
                - NeqEpstein(Amp[F2],PeakHeight[F2],BotThick[F2], PeakHeight[F1])
                - NeqEpstein(Amp[E], PeakHeight[E],TopThick[E], PeakHeight[F1]))
           Amp[F1]=NeqJoin(Amp[F1],0.8*Nm[F1],1, Amp[F1] - 0.8*Nm[F1])
           Amp[E]=4*(Nm[E]
                - NeqEpstein(Amp[F1], PeakHeight[F1],BotThick[F1], PeakHeight[E])
                - NeqEpstein(Amp[F2], PeakHeight[F2],BotThick[F2], PeakHeight[E]))
        end for_2
end if_1
Amp[E]=NeqJoin(Amp[E],0.05,60.0, Amp[E] - 0.005)

Calculate shape factor for topside F2 region (see [14])
if_1 (mth > 3 && mth < 10), April to September
        b2k=6.705 - 0.014 * R12 - 0.008*PeakHeight[F2]
else_1 October to May
        b2k=-7.77 + 0.097 * pow(PeakHeight[F2]/ BotThick[F2],2) + 0.153 * Nm[F2]
end if_1
b2k=NeqJoin(b2k,2,1,b2k - 2.0)
b2k=NeqJoin(8,b2k,1,b2k - 8.0)

Adjust the vertical TEC value to take into account exosphere electron density ([15])
TopThick[F2]=b2k * BotThick[F2]
x=(TopThick[F2] - 150.0) / 100.0
v=(0.041163 * x - 0.183981) * x + 1.424472
TopThick[F2] = TopThick[F2] / v
```

9.2.9.2 NeQuick internal function "NeqCalcSphLegCoeffs"

Purpose

This function calculates the spherical Legendre coefficients which are used in the calculations for foF2 or M(3000)F2 frequencies calculated from CCIR map file data.

Interfaces

Called by: NeqCalcEpstParams

Calls: none

Inputs:

Name	Type	Size	Range	Units	Description
dUT	double	1	[0,24)	hours	Time (UTC) at which STEC value is required

Table 32. NeqCalcSphLegCoeffs Function Input Data

Inputs/Outputs:

Name	Type	Size	Range	Units	Description
pstCurrCCIR	CurrentCCIR_st	1	Not defined	N/A	Information required for computing F0F2 and M3000F2 coefficients for given time and R12 value.

Table 33. NeqCalcSphLegCoeffs Function Input/Output Data

Return value: -

Internal Processing

Compute the longitude of the sun *t* at the current time
*t =(dUT * 15° - 180) * DR*

Calculate the sine and cosine of the fundamental and the harmonics using sin(nA)=sin[(n-1)A + A] with sin(A+B) = sin(A)cos(B) + cos(A)sin(B), and cos(nA)=sin[(n-1)A + A] with cos(A+B) = cos(A)cos(B) - sin(A)sin(B)

$sinHarm_1$ = sin(t)
$cosHarm_1$ = cos(t)
for1 each additional harmonic, i = 2, i<=numMaxHarm (where numMaxHarm=6)
    $sinHarm_i = sinHarm_{i-1} * cosHarm_1 + cosHarm_{i-1} * sinHarm_1$
    $cosHarm_i = cosHarm_{i-1} * cosHarm_1 - sinHarm_{i-1} * sinHarm_1$
end for1

Calculate coefficients for spherical Legendre function using following equation:

$$c_i = pstCurrCCIR_{(i-1)N+1} + \sum_{k=1}^{numHarm} \left(pstCurrCCIR_{(i-1)N+2k} sinHarm_k + pstCurrCCIR_{(i-1)N+2k+1} cosHarm_k\right)$$

Where:

- *i* is the index of the coefficients (*i*=1,...,76 for FoF2 and *i*=1,...,49 for M(3000)F2),
- *N* is the short number of coefficients (*N*=13 for FoF2 and *N*=9 for M(3000)F2),
- *numHarm*=6 for foF2 and *numHarm*=4 for M(3000)F2

9.2.9.3 NeQuick internal function "NeqGetF2FreqFromCCIR"

Purpose

This function returns foF2 or M(3000)F2 calculated from CCIR map file data.

Interfaces

Called by: NeqCalcEpstParams

Calls: none

Inputs:

Name	Type	Size	Range	Units	Description
dCosLat	double	1	-1 to 1	N/A	Cosine of the current latitude
dLng	double	1	-180 to 180	deg	Current longitude
pdLegCoeffs	double	[76]	Not defined	N/A	Current set of spherical Legendre coefficients
pdSinModipToN	double	[12]	Not defined	N/A	Array containing sin(modip)^N terms
siMode	Integer	1	0, 1	N/A	Flag indicating which frequency is to be calculated (FoF2 or M3000F2)

Table 34. NeqGetF2FreqFromCCIR Function Input Data

Return value: dResult – calculated FoF2 or M3000F2 value at current point

Internal Processing

harm = number of harmonics in expansion, nq[] = constants used in spherical Legendre expansion, k1 = size of nq[], m = rows in CCIR[], mm = cols in CCIR[], m3 = total elements in CCIR[]

If$_1$ siMode = 0 (F0F2)
    Set constants used in spherical Legendre expansion
    nq = { 11,11,8,4,1,0,0,0,0}
    Set k1 (size of nq) = 9
Else$_1$
    Set constants used in spherical Legendre expansion
    nq = { 6,7,5,2,1,0,0}
    Set k1 (size of nq) = 7
End$_1$

Compute output value as sum of spherical Legendre functions:

$$foF2 \text{ or } M(3000)F2 = \sum_{j=0}^{nq_0} c_j \sin^j \mu + \sum_{k=1}^{k1-1} \sum_{i=0}^{nq_k} \sin^i \mu \cdot \cos^k \phi \cdot (c_{R+2i} \cos k\lambda + c_{R+2i+1} \sin k\lambda)$$

Where:

$$R = \left[2 \sum_{l=0}^{k-1} (nq_l + 1) \right] - (nq_0 + 1)$$

and:

- $\sin^i \mu$ = *pdSinModipToN*$_i$, $\cos \phi$=*dCosLat*, λ=*dLng*
- c_i comes from the input set of current spherical Legendre coefficients pdLegCoeffs

Note1: $\sin^i \mu$ *terms are set to zero if found to be <= 10*$^{-30}$

Note2: *when setting the constants to be used in spherical Legendre expansion, notice that the above defined nq vectors differ by one unit in each vector component from the corresponding ones provided in Eq. 63 and Eq. 71 since the nq vector above is 0 base.*

9.2.9.4 NeQuick internal function "NeqCriticalFreqToNe"

Purpose

From critical frequency, calculates the associated electron density, using

$N[m^{-3}] = 0.124*10^{11}*f[MHz]^2$

(NB output is not scaled by 10^{11} here, it is scaled in NeqCalcBottomsideNe and NeqCalcTopsideNe).

Interfaces

Called by: NeqCalcEpstParams

Calls: none

Input:

Name	Type	Size	Range	Units	Description
dF0	double	1	>0	MHz	The peak plasma frequency for the layer

Table 35. NeqCriticalFreqToNe Function Input Data

Return value: The calculated electron density.

Internal Processing

return 0.124 * dF0 * dF0

9.2.9.5 NeQuick internal function "NeqCalcF2PeakHeight"

Purpose

This function calculates F2 layer peak height *hm[F2]* from *foE*, *foF2* and *M3000*. It is based on the method of Dudeney(1983), but modified such that the ratio foF2/foE is clipped at 1.75 using NeqJoin. Note that the clipping is 'soft', the 1st derivative is continuous but note the clipped value can be slightly below 1.75 at the join (but note analysis indicate >1.73). Also, Dudeney uses a figure of 1470 rather than 1490 in the numerator of hmF2 and a figure of 1.296 rather than 1.2967 in the denominator of MF.

Interfaces

Called by: NeqCalcEpstParams

Calls: NeqJoin

Inputs:

Name	Type	Size	Range	Units	Description
dM3000	double	1	>0	N/A	The ratio of the maximum usable frequency at a distance of 3000 km to the F2 layer critical frequency, foF2
dF0E	double	1	>0	Hz	The peak plasma frequency for the E layer
dF0F2	double	1	>0	Hz	The peak plasma frequency for the F2 layer

Table 36. NeqCalcF2PeakHeight Function Input Data

Return value: Height at which electron density peaks in F2 layer.

Internal Processing

Compute $MF = dM3000 * \sqrt{\frac{0.0196 * dM3000^2 + 1}{1.2967 * dM3000^2 - 1}}$

(note: no divide by zero check needed as previous clipping ensures M3000 >= 1)

if₁ dF*0E >=1e*$^{-30}$*, i.e. dF0E* non-zero

$r = \frac{dF0F2}{dF0E}$, soft clipped for r<1.75, using NeqJoin:

$r2 = \frac{re^{20(r-1.75)} + 1.75}{e^{20(r-1.75)} + 1}$ (NB exp() protected using NeqClipExp())

$\Delta M = \frac{0.253}{r2 - 1.215} - 0.012$

(note: no divide by zero check needed as r2 > ~1.75)

else₁

$\Delta M = -0.012$ (limit as r becomes very large)

end if₁

Compute peak height in F2 later as:

$$PeakHeightF2 = \frac{1490 * MF}{\mathrm{dM3000} + \Delta M} - 176 \qquad [km]$$

(note: no divide by zero check as dM3000>=1 and minimum possible ΔM is -0.012)

return *PeakHeightF2*

9.2.9.6 NeQuick internal function “NeqEpstein”

Purpose

Evaluates Epstein layer function (but without the normalisation factor =4).

Interfaces

Called by: NeqCalcEpstParams

Calls: NeqClipExp, NeqSquared

Input:

Name	Type	Size	Range	Units	Description
dNmax	double	1	Not defined	m^3	The peak density for the layer
dHmax	double	1	Not defined	km	The height of the layer electron density peak
dB	double	1	Not defined	N/A	The layer thickness parameter
dH	double	1	Not defined	km	The height at which the electron density is required

Table 37. NeqCalcF2PeakHeight Function Input Data

Return value: A quarter of the electron density at the point.

Internal Processing

ExpTerm = **NeqClipExp***((dH - dHmax) / dB)*
Return = d*Nmax * ExpTerm / (1 + ExpTerm)^2*

NB the full equation is:

$$N(h) = \frac{4.Nm.e^{\frac{h-Hm}{B}}}{\left(e^{\frac{h-Hm}{B}} + 1\right)^2}$$

However, this function only calculates a 1/4 of it because the factor of 4 is completed once other mathematical formulas have been done to decrease computation time.

9.2.10 NeqCalcTopSide.c Module

9.2.10.1 NeQuick internal function "NeqCalcTopsideNe"

Purpose

This function calculates electron content at the specified height, in the top part of the ionosphere above the F2 electron density peak point.

The function uses topside expression derived from [15], although the expression in the paper has an error with the brackets.

Interfaces

Called by: NeqGetNeOnVertRay and NeqGetNeOnSlantRay

Calls: NeqCalcBottomsideNe, NeqClipExp, NeqSquared

Inputs:

Name	Type	Size	Range	Units	Description
dUT	double	1	[0,24)	hours	Time (UTC) at which STEC value is required

Table 38. NeqCalcTopSide Function Input Data

Inputs/Outputs:

Name	Type	Size	Range	Units	Description
pstCurrCCIR	CurrentCCIR_st	1	Not defined	N/A	Information required for computing F0F2 and M3000F2 coefficients for given time and R12 value.

Table 39. NeqCalcTopSide Function Input/Output Data

Return value: dReturn – the computed electron density at the given height in the top part of the ionosphere above the F2 electron density peak.

Internal Processing

Calculate temporary variable *ee*

$$ee = NeqClipExp\left(\frac{dH - PeakHeight[F2]}{TopThick[F2] * \left(1 + \frac{rfac * g * (dH - PeakHeight[F2])}{rfac * TopThick[F2] + g * (dH - PeakHeight[F2])}\right)}\right)$$

where

- $g = 0.125$
- $rfac$=100
- And PeakHeight[F2] and TopThick[F2] are contained in the pstLayers input data structure.

if$_1$ *ee* > 10^{11}, deal with limit when ee very large

$$ep = \frac{4}{ee}$$

else$_1$

$$ep = \frac{4 * ee}{(1 + ee)^2}$$

end if$_1$

if$_1$ pd*Nmax* has not been calculated for this location,

call function "*NeqCalcBottomsideNe*" to calculate pdNmax at PeakHeight[F2] (crossover point)

end if$_1$

Return = *pdNmax * ep*

9.2.11 NeqCalcBottomsideNe.c Module

9.2.11.1 NeQuick internal function "NeqCalcBottomsideNe"

Purpose

This function calculates electron content at the specified height *dHH,* in the bottom part of the ionosphere below the F2 peak height. The function sums semi-Epstein Layers with a modification to reduce excessive Ne around F2 peak and 100km.

Interfaces

Called by: NeqGetNeOnVertRay, NeqGetNeOnSlantRay and NeqCalcTopsideNe

Calls: NeqClipExp, NeqSquared

Inputs:

Name	Type	Size	Range	Units	Description
dHH	double	1	Not defined	km	The height at which the electron density is required
pstLayers	LayerProperties_st	1	Not defined	N/A	Current Ionospheric properties

Table 40. NeqCalcBottomSide Function Input Data

Return value: dReturn – the computed electron density at the given height in the bottom part of the ionosphere below the F2 electron density peak.

Internal Processing

Set value of *B[F2] = BotThick[F2]*

if$_1$ above F1 peak, *dHH > PeakHeight[F1]*
Set *B[F1] = TopThick[F1]*
else$_1$
Set *B[F1] = BotThick[F1]*
end if$_1$

if$_1$ above E peak, *dHH > PeakHeight[E]*
B[E] = TopThick[E]
else$_1$
B[E] = BotThick[E]
end if$_1$
(where PeakHeight, BothThick and TopThick are contained in pstLayers input data)

if$_1$ height is below 100km (*dHH < 100*)
for$_2$ each ionospheric section *j=E,F1,F2*
if$_3$ F2 section, *j = F2*
Calculate arg from PeakHeight[F2],

$$arg = \frac{h_0 - PeakHeight[F2]}{B[F2]}$$ where h_0 = *100km*

else$_3$

Calculate arg from *PeakHeight[F2]* and *PeakHeight[j]* ,

$$arg = \frac{h_0 - PeakHeight[j]}{B[j]} * \exp\left(\frac{f1}{1 + f2 * |h_0 - PeakHeight[F2]|}\right)$$

$$h_0 = 100$$

Where: $f1 = 10$

$$f2 = 1$$

end if$_3$

if$_3$ size of arg is large, $|arg| > 25$

Values of s_j and ds_j are zero, set $s_j = 0$ and $ds_j = 0$

else$_3$

Calculate s_j and ds_j,

$$s_j = \frac{Amp[j]\exp(arg)}{(1 + \exp(arg))^2}$$

$$ds_j = \frac{1 - \exp(arg)}{B[j](1 + \exp(arg))}$$

end if$_3$

end for$_2$

(where Amp[j] is included in pstLayers input data)

Calculate return result,

$$N_e = aN_0 * NeqClipExp(1 - bf * z - NeqClipExp(-z))$$

Where

- $aN_0 = 10^{11}\sum_j s_j$
- $bf = 1 - \frac{Hd \cdot \sum_j (ds_j * s_j)}{\sum_j s_j}$
- $z = \frac{dHH - h_0}{Hd}$
- $Hd = 10$

else$_1$ (above 100km)

for$_2$ each ionospheric section *j=E,F1,F2*

if$_3$ F2 section, *j = F2*

Calculate arg from *PeakHeight[F2]*,

$$arg = \frac{dHH - PeakHeight[F2]}{B[j]}$$

else$_3$

Calculate arg from *PeakHeight[F2]* and *PeakHeight[j]*,

$$arg = \frac{dHH - PeakHeight[j]}{B[j]} * \exp\left(\frac{f1}{1 + f2 * |dHH - PeakHeight[F2]|}\right)$$

where $f1 = 10$, $f2 = 1$

end if$_3$

if$_3$ size of arg is large, $|arg| > 25$

Value of s_j is zero, set $s_j = 0$

else₃

Calculate s_j,

$$s_j = \frac{Amp[j]\exp(arg)}{(1+\exp(arg))^2}$$

end if₃

end for₂

Calculate return result, (scaled to correct for Amp[] scaling)

$$N_e = 10^{11}\sum_j s_j$$

end if₁

9.2.12 NeqUtils.c Module

9.2.12.1 NeQuick internal function "NeqJoin"

Purpose

Allows smooth joining of functions f1 and f2 (i.e. continuous first derivatives) at origin. Alpha determines width of transition region. Calculates value of joined functions at x.

Interfaces

Called by: NeqCalcEpstParams, NeqCalcF2PeakHeight

Calls: NeqClipExp

Inputs:

Name	Type	Size	Range	Units	Description
dF1	double	1	Not defined	N/A	Input term for NeqJoin computation
dF2	double	1	Not defined	N/A	Input term for NeqJoin computation
dAlpha	double	1	Not defined	N/A	Input term for NeqJoin computation
dX	double	1	Not defined	N/A	Input term for NeqJoin computation

Table 41. NeqJoin Function Input Data

Return value: Computed value

Internal Processing

ee=**NeqClipExp**(*dAlpha * dX*)
return = (dF1 * ee + dF2) / (ee + 1);

9.2.12.2 NeQuick internal function "NeqClipExp"

Purpose

A clipped exponential function – always returns valid output.

Interfaces

Called by: NeqCalcEpstParams, NeqEpstein, NeqCalcTopsideNe, NeqCalcBottomsideNe and NeqJoin

Calls: none

Input:

Name	Type	Size	Range	Units	Description
dPower	double	1	Not defined	N/A	Power for exponential function

Table 42. NeqClipExp Function Input Data

Return value: Clipped exponential value

Internal Processing

$$\text{Return} = \begin{cases} 5.5406e34 & \text{dPower} > 80 \\ e^{\text{Power}} & 80 \geq \text{dPower} \geq -80 \\ 1.8049e-35 & \text{dPower} < -80 \end{cases}$$

9.2.12.3 NeQuick internal function "NeqSquared"

Purpose

This calculated the square of a number.

Interfaces

Called by: NeqCalcEpstParams, NeqEpstein, NeqCalcTopsideNe and NeqCalcBottomsideNe

Calls: none

Input:

Name	Type	Size	Range	Units	Description
dValue	double	1	Not defined	N/A	Input value

Table 43. NeqSquared Function Input Data

Return value: Square of the input value.

Internal Processing

Return = dValue * dValue

9.3 NeQuick Function Data Structures

This section contains descriptions of the internal data structures that are used within the NeQuick function.

Name	Type	Size	Range	Units	Description
pstModip	MODIP_st	1	Not defined	N/A	Structure containing grid of modified dip latitude values
pstCCIR	CCIR_st	1	Not defined	N/A	Structure containing CCIR coefficients for computing FoF2 and M(3000)F2
pdKronrodTol	double	[2]	>0	N/A	Tolerances for Kronrod integration
siMaxRecurse	int	1	>0	N/A	Maximum level of recursion allowed in Kronrod integration
pdGssPosLLH	double	[3]	Not defined	rad/rad/m	Receiver position (lat/lon/h)[ii]
pdSatPosLLH	double	[3]	Not defined	rad/rad/m	Satellite position (lat/lon/h)
siMonth	int	1	[1,12]	months	Month during which STEC value is required
dUT	double	1	[0,24)	hours	Time (UTC) at which STEC value is required
siNumCoeff	int	1	>=1	N/A	Number of Az coefficients
pdCoeff	double	[numCoeffs]	>0	flux units/deg^j	Az coefficients
dAzBase	double	1	0 -400	solar flux units	Az value at receiver locations

Table 44. Definition of NeQuickInputData_st Data Structure

[ii] Receiver position and satellite position input values are expected in WGS-84 ellipsoidal coordinates: geodetic latitude, geodetic longitude and ellipsoidal height. Notice that these ellipsoidal coordinates are treated as spherical coordinates within the NeQuick model.

Name	Type	Size	Range	Units	Description
pdModip	double	[39][39]	[-90,90]	deg	Grid of modified dip latitude values. The grid is wrapped around the poles and so the arrangement is as follows: • Row 0: 85 degrees North • Row 1: 90 degrees North • Row 2: 85 degrees North • … • Row 37: 90 degrees South • Row 38: 85 degrees South In a similar way, the columns go from 190 degrees West to 190 degrees East in 10 degree steps

Table 45. Definition of MODIP_st Data Structure

Name	Type	Size	Range	Units	Description
pdF2	double	[12][13][76][2]	Not defined	N/A	CCIR coefficients for computing FoF2, the critical frequency of the F2 layer
pdM3000	double	[12][9][49][2]	Not defined	N/A	CCIR coefficients for computing M(3000)F2, The ratio of the maximum usable frequency at a distance of 3000 km to the F2 layer critical frequency, foF2

Table 46. Definition of CCIR_st Data Structure

Name	Type	Size	Range	Units	Description
dLat	double	1	-90 to 90	deg	Latitude of Point
dLng	double	1	-180 to 180	deg	Longitude of Point
dH	double	1	Not defined	km	Height of Point
dR	double	1	>0	km	Radius of Point
dS	double	1	>0	km	Distance of Point to Ray Parigee
dSinLat	double	1	-1 to 1	N/A	Sine of latitude of Point
dCosLat	double	1	-1 to 1	N/A	Cosine of latitude of Point

Table 47. Definition of SPoint_st Data Structure

Name	Type	Size	Range	Units	Description
dLat	double	1	-90 to 90	deg	Latitude of Point
siMonth	int	1	[1,12]	months	Month during which current STEC value has been computed
dR12	double	1	Not defined	N/A	Current R12 index - twelve-month smoothed relative sunspot number
pdF0F2	double	[988]	Not defined	N/A	Interpolated coefficients for computing FoF2 for current month and R12 conditions
pdM3000F2	double	[441]	Not defined	N/A	Interpolated coefficients for computing M3000F2 for current month and R12 conditions
dUT	double	1	[0,24)	hours	Time (UTC) at which current STEC value has been computed
pdLegCoeffs_F0	double	[76]	Not defined	N/A	Spherical Legendre coefficients for calculating F0F2 for current month and R12 conditions
pdLegCoeffs_M3000	double	[49]	Not defined	N/A	Spherical Legendre coefficients for calculating M(3000)F2 for current month and R12 conditions

Table 48. Definition of CurrentCCIR_st Data Structure

Name	Type	Size	Range	Units	Description
pdAmp	double	3	Not defined	10^11/m^3	Epstein amplitude parameter
pdPeakHeight	double	3	Not defined	km	Epstein peak height parameter
pdBotThick	double	3	Not defined	km	Epstein bottom half-layer thickness parameter
pdTopThick	double	3	Not defined	km	Epstein top half-layer thickness parameter
dM3000	double	1	Not defined		Current M(3000)F2 value
pdF0	double	3	Not defined		Current F0 (peak plasma frequency) for the F2, F1 and E layers respectively

Table 49. Definition of LayerProperties_st Data Structure

Name	Type	Size	Range	Units	Description
stP1	SPoint_st	1	Not defined	N/A	Information for point 1 (receiver)
stP2	SPoint_st	1	Not defined	N/A	Information for point 2 (satellite)
stRay	SPoint_st	1	Not defined	N/A	Information for ray
stPactual	SPoint_st	1	Not defined	N/A	Information for current integration point
dZeta	double	1	-90 to 90	deg	Zenith angle of point 2 seen from point 1
dSinDelta	double	1	-1 to 1	N/A	Sine of angle of declination of sun
dCosDelta	double	1	-1 to 1	N/A	Cosine of angle of declination of sun
dSinSig	double	1	-1 to 1	N/A	Sine of ray azimuth
dCosSig	double	1	-1 to 1	N/A	Cosine of ray azimuth

Table 50. Definition of Geometry_st Data Structure

Name	Type	Size	Range	Units	Description
pstNeQuickInput Data	NeQuic kInputD ata_st	1	Not defined	N/A	Input data to NeQuick Function
pstGeom	Geome tryData _st	1	Not defined	N/A	Geometry data for ray
pstCurrCCIR	Current _st	1	Not defined	N/A	foF2 and M(3000)F2 information for current month and R12
dTolerance	double	1	>0	N/A	Tolerance for Kronrod integration
bVert	Boolea n	1	FALSE, TRUE	N/A	Flag indicating whether ray is vertical or not

Table 51. Definition of IntegrateData_st Data Structure

ET-02-15-003-EN-N


European GNSS (Galileo)
Open Service

Signal In Space
Interface Control Document

GALILEO

NAVIGATION
SOLUTIONS
POWERED BY
E U R O P E