European GNSS (Galileo) Open Service

Ionospheric Correction Algorithm for Galileo Single Frequency Users
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

<table>
<thead>
<tr>
<th>Reason for change</th>
<th>Issue</th>
<th>Revision</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Issue</td>
<td>Issue 1</td>
<td>0</td>
<td>November 2014</td>
</tr>
<tr>
<td>Document ready for publication after revision</td>
<td>Issue 1</td>
<td>1</td>
<td>June 2015</td>
</tr>
</tbody>
</table>
**Table of Contents**

1. **Introduction** .................................................................................................................................................. 1
   1.1. Document Scope .......................................................................................................................................... 1
   1.2. Background .................................................................................................................................................. 1

2. **Single Frequency Ionospheric Correction Algorithm** .................................................................................. 5
   2.1. Overview .................................................................................................................................................... 5
   2.2. Step-by-Step Procedure ........................................................................................................................... 5
   2.3. Inputs and Outputs ...................................................................................................................................... 5
       2.3.1. **Galileo Navigation Message Relevant to Single-Frequency Ionospheric Algorithm** .............. 6
   2.4. **MODIP Regions** .................................................................................................................................... 6
   2.5. **NeQuick G Ionospheric Electron Density Mode** .................................................................................. 7
       2.5.1. The Epstein Function .......................................................................................................................... 7
       2.5.2. Constants Used ..................................................................................................................................... 7
       2.5.3. Complementary Files .......................................................................................................................... 8
           2.5.3.1. MODIP Grid .................................................................................................................................. 8
           2.5.3.2. CCIR Files .................................................................................................................................... 8
       2.5.4. Auxiliary Parameters ........................................................................................................................ 8
           2.5.4.1. Local Time .................................................................................................................................... 8
           2.5.4.2. Read modipNeQG_wrapped.Asc Values in an Array ................................................................. 9
           2.5.4.3. Compute MODIP ......................................................................................................................... 9
           2.5.4.4. Effective Ionization Level Az ..................................................................................................... 9
           2.5.4.5. Effective Sunspot Number (R_{12} Like) .................................................................................... 10
           2.5.4.6. Solar Declination ........................................................................................................................ 10
           2.5.4.7. Solar Zenith Angle ....................................................................................................................... 10
           2.5.4.8. Effective Solar Zenith Angle ....................................................................................................... 10
       2.5.5. Model Parameters ............................................................................................................................... 11
           2.5.5.1. f0E and NmE .................................................................................................................................. 11
           2.5.5.2. f0F1 and NmF1 ............................................................................................................................... 11
           2.5.5.3. f0F2 and NmF2, M(3000)F2 ......................................................................................................... 12
           2.5.5.4. hmE ................................................................................................................................................. 15
           2.5.5.5. hmF1 .............................................................................................................................................. 15
           2.5.5.6. hmF2 .............................................................................................................................................. 15
           2.5.5.7. B2bot, B1top, B1bot, BEtop, BEbot ............................................................................................... 15
           2.5.5.8. A1 .................................................................................................................................................... 16
           2.5.5.9. A2 and A3 .................................................................................................................................... 16
           2.5.5.10. Shape Parameter k ...................................................................................................................... 16
           2.5.5.11. H0 .................................................................................................................................................. 17
       2.5.6. Electron Density Computation ............................................................................................................. 17
           2.5.6.1. The Bottomside Electron Density .................................................................................................. 17
           2.5.6.2. The Topside Electron Density ....................................................................................................... 18
       2.5.7. Auxiliary Routines .................................................................................................................................. 19
           2.5.7.1. Third Order Interpolation Function Z(x; Z_1, Z_2, Z_3, Z_4, X) .................................................... 19
       2.5.8. TEC Calculation ..................................................................................................................................... 20
           2.5.8.1. Vertical TEC Calculation ............................................................................................................... 20
           2.5.8.2. Slant TEC Calculation ................................................................................................................. 21
           2.5.8.3. Alternative Computational Efficient TEC Integration Method .................................................... 27
       2.5.9. Clarification on Coordinates Used in NeQuick ..................................................................................... 27

2.6. Differences Between NeQuick G and NeQuick 1 and NeQuick 2 .................................................................. 27
   2.6.1. Summary of Differences With NeQuick 1 ............................................................................................ 27
   2.6.2. Summary of Differences With NeQuick 2 ............................................................................................ 28

© European Union 2015
Document subject to terms of use and disclaimers p.i
Ionospheric Correction Algorithm for Galileo Single Frequency Users, Issue 1.1, June 2015
3. Implementation Guidelines for User Receivers ................................................................. 29
   3.1. Zero-Valued Coefficients and Default Effective Ionisation Level .................................... 29
   3.2. Applicability and Coherence of Broadcast Coefficients .................................................. 29
   3.3. Effective Ionisation Level Boundaries ........................................................................... 29
   3.4. Integration of NeQuick G into Higher Level Software .................................................... 30
   3.5. Computation Rate of Ionospheric Corrections .................................................................. 30

4. Annex A – Applicable and Reference Documents ............................................................... 31
   4.1. Applicable Documents .................................................................................................. 31
   4.2. Reference Documents .................................................................................................. 31

5. Annex B – Acronyms and Definitions .................................................................................. 33
   5.1. Acronyms ..................................................................................................................... 33
   5.2. Definitions ..................................................................................................................... 33

6. Annex C – Complementary Files (CCIR and MODIP) .......................................................... 37

7. Annex D – NeQuick G Performance Results ....................................................................... 39

8. Annex E – Input/Output Verification Data ............................................................................. 43
   8.1. Az Coefficients (High Solar Activity) ............................................................................. 43
   8.2. Az Coefficients (Medium Solar Activity) ....................................................................... 44
   8.3. Az Coefficients (Low Solar Activity) ............................................................................. 45

   9.1. External Interfaces ........................................................................................................ 47
      9.1.1. Introduction ........................................................................................................... 47
         9.1.1.1. Inputs .............................................................................................................. 47
         9.1.1.2. Outputs ........................................................................................................... 47
         9.1.1.3. Input/Output .................................................................................................... 47
      9.2. Modules ..................................................................................................................... 48
         9.2.1. Introduction ........................................................................................................ 48
         9.2.2. Function Overview ............................................................................................. 48
         9.2.3. NeQuick.c Module .............................................................................................. 49
            9.2.3.1. Function “NeQuick” (Main Function) ............................................................ 49
            9.2.3.2. NeQuick Internal Function “NeqCheckInputs” ............................................. 50
            9.2.3.3. NeQuick Internal Function “DoTECIntegration” .......................................... 50
         9.2.4. NeqCalcModipAz.c Module ............................................................................... 52
            9.2.4.1. NeQuick Internal Function “NeqCalcModip” ................................................ 52
            9.2.4.2. NeQuick Internal Function “NeqInterpolate” ................................................. 53
            9.2.4.3. NeQuick Internal Function “NeqModipToAz” ................................................ 54
         9.2.5. NeqGetRayProperties.c Module .......................................................................... 54
            9.2.5.1. NeQuick Internal Function “NeqGetRayProperties” ..................................... 55
            9.2.5.2. NeQuick Internal Function “NeqCalcRayProperties1” .................................. 55
            9.2.5.3. NeQuick Internal Function “NeqCalcRayProperties2” .................................. 57
         9.2.6. NeqIntegrate.c Module ......................................................................................... 58
            9.2.6.1. NeQuick Internal Function “NeqIntegrate” ....................................................... 58
         9.2.7. NeqGetNeOnVertRay.c Module ............................................................................ 60
            9.2.7.1. NeQuick Internal Function “NeqGetNeOnVerticalRay” ................................... 60
9.2.8. NeqGetNeOnSlantRay.c Module........................................................................................................61
  9.2.8.1. NeQuick Internal Function “NeqGetNeOnSlantRay”...............................................................61
  9.2.8.2. NeQuick Internal Function “NeqCalcLLHOnRay”...................................................................62
9.2.9. NeqCalcEpstParams.c Module.......................................................................................................63
  9.2.9.1. NeQuick Internal Function “NeqCalcEpstParams”.................................................................63
  9.2.9.2. NeQuick Internal Function “NeqCalcSphLegCoeffs”..............................................................66
  9.2.9.3. NeQuick Internal Function “NeqGetF2FreqFromCCIR”.........................................................67
  9.2.9.4. NeQuick Internal Function “NeqCriticalFreqToNe”.............................................................68
  9.2.9.5. NeQuick Internal Function “NeqCalcF2PeakHeight”..............................................................69
  9.2.9.6. NeQuick Internal Function “NeqEpstein”...............................................................................70
9.2.10. NeqCalcTopSide.c Module..........................................................................................................70
  9.2.10.1. NeQuick Internal Function “NeqCalcTopsideNe”.................................................................70
9.2.11. NeqCalcBottomsideNe.c Module...............................................................................................71
  9.2.11.1. NeQuick Internal Function “NeqCalcBottomsideNe”..........................................................71
9.2.12. NeqUtils.c Module......................................................................................................................73
  9.2.12.1. NeQuick Internal Function “NeqJoin”..................................................................................73
  9.2.12.2. NeQuick Internal Function “NeqClipExp”..........................................................................74
  9.2.12.3. NeQuick Internal Function “NeqSquared”..........................................................................74
9.3. NeQuick Function Data Structures.................................................................................................75
List of Figures

Figure 1. Example of a Global VTEC Map Obtained With NeQuick ................................................................. 3
Figure 2. MODIP Regions Associated to Different Ionospheric Characteristics ...................................................... 6
Figure 3. Geometry of Zenith angle Computation ................................................................................................. 22
Figure 4. Geometry of Ray Perigee Computation ................................................................................................. 23
Figure 5. Ionospheric Delay Vs. Satellite Elevation (Upper) and of UTC (Lower) ...................................................... 39
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 % ........................................................................... 40
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 .............................................. 41
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) ................................................................................................................................. 41
Figure 9. Residual Single-Frequency RMS Error Contribution to UERE After Correction with Single-Frequency Ionospheric Algorithm [Meters L1] .................................................................................. 42
Figure 10. Overview of NeQuick Function Hierarchy ............................................................................................ 48
## List of Tables

<table>
<thead>
<tr>
<th>Table No.</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Input Parameters</td>
<td>6</td>
</tr>
<tr>
<td>Table 2</td>
<td>Constants Definition</td>
<td>7</td>
</tr>
<tr>
<td>Table 3</td>
<td>NeQuick Function Input Data</td>
<td>47</td>
</tr>
<tr>
<td>Table 4</td>
<td>NeQuick Function Output Data</td>
<td>47</td>
</tr>
<tr>
<td>Table 5</td>
<td>NeQuick Function Output Data</td>
<td>48</td>
</tr>
<tr>
<td>Table 6</td>
<td>NeqCheckInputs Function Input Data</td>
<td>50</td>
</tr>
<tr>
<td>Table 7</td>
<td>DoTECIntegration Function Input Data</td>
<td>51</td>
</tr>
<tr>
<td>Table 8</td>
<td>NeQuick Function Output Data</td>
<td>51</td>
</tr>
<tr>
<td>Table 9</td>
<td>NeqCalcModip Function Input Data</td>
<td>52</td>
</tr>
<tr>
<td>Table 10</td>
<td>NeqInterpolate Function Input Data</td>
<td>53</td>
</tr>
<tr>
<td>Table 11</td>
<td>NeqCalcModip Function Input Data</td>
<td>54</td>
</tr>
<tr>
<td>Table 12</td>
<td>NeqGetRayProperties Function Input/Output Data</td>
<td>55</td>
</tr>
<tr>
<td>Table 13</td>
<td>NeqGetRayProperties Function Output Data</td>
<td>55</td>
</tr>
<tr>
<td>Table 14</td>
<td>NeqCalcRayProperties1 Function Input/Output Data</td>
<td>56</td>
</tr>
<tr>
<td>Table 15</td>
<td>NeqCalcRayProperties1 Function Output Data</td>
<td>56</td>
</tr>
<tr>
<td>Table 16</td>
<td>NeqCalcRayProperties2 Function Input Data</td>
<td>57</td>
</tr>
<tr>
<td>Table 17</td>
<td>NeqCalcRayProperties2 Function Output Data</td>
<td>57</td>
</tr>
<tr>
<td>Table 18</td>
<td>NeqIntegrate Function Input Data</td>
<td>58</td>
</tr>
<tr>
<td>Table 19</td>
<td>NeqIntegrate Function Output Data</td>
<td>59</td>
</tr>
<tr>
<td>Table 20</td>
<td>NeqIntegrate Function Input Data</td>
<td>59</td>
</tr>
<tr>
<td>Table 21</td>
<td>NeqIntegrate Function Input/Output Data</td>
<td>59</td>
</tr>
<tr>
<td>Table 22</td>
<td>NeqGetNeOnVertRay Function Input Data</td>
<td>60</td>
</tr>
<tr>
<td>Table 23</td>
<td>NeqGetNeOnVertRay Function Output Data</td>
<td>61</td>
</tr>
<tr>
<td>Table 24</td>
<td>NeqGetNeOnSlantRay Function Input Data</td>
<td>61</td>
</tr>
<tr>
<td>Table 25</td>
<td>NeqGetNeOnSlantRay Function Output Data</td>
<td>61</td>
</tr>
<tr>
<td>Table 26</td>
<td>NeqGetNeOnSlantRay Function Input/Output Data</td>
<td>61</td>
</tr>
<tr>
<td>Table 27</td>
<td>NeqCalcLLHOnRay Function Input Data</td>
<td>62</td>
</tr>
<tr>
<td>Table 28</td>
<td>NeqCalcLLHOnRay Function Output Data</td>
<td>62</td>
</tr>
<tr>
<td>Table 29</td>
<td>NeqCalcEpstParams Function Input Data</td>
<td>63</td>
</tr>
<tr>
<td>Table 30</td>
<td>NeqCalcEpstParams Function Output Data</td>
<td>64</td>
</tr>
<tr>
<td>Table 31</td>
<td>NeqCalcEpstParams Function Input/Output Data</td>
<td>64</td>
</tr>
<tr>
<td>Table 32</td>
<td>NeqCalcSphLegCoeffs Function Input Data</td>
<td>67</td>
</tr>
<tr>
<td>Table 33</td>
<td>NeqCalcSphLegCoeffs Function Output Data</td>
<td>67</td>
</tr>
<tr>
<td>Table 34</td>
<td>NeqGetF2FreqFromCCIR Function Input Data</td>
<td>68</td>
</tr>
<tr>
<td>Table 35</td>
<td>NeqCriticalFreqToNe Function Input Data</td>
<td>69</td>
</tr>
<tr>
<td>Table 36</td>
<td>NeqCalcF2PeakHeight Function Input Data</td>
<td>69</td>
</tr>
<tr>
<td>Table 37</td>
<td>NeqCalcF2PeakHeight Function Output Data</td>
<td>70</td>
</tr>
<tr>
<td>Table 38</td>
<td>NeqCalcTopSide Function Input Data</td>
<td>71</td>
</tr>
<tr>
<td>Table 39</td>
<td>NeqCalcTopSide Function Input/Output Data</td>
<td>71</td>
</tr>
<tr>
<td>Table 40</td>
<td>NeqCalcBottomSide Function Input Data</td>
<td>72</td>
</tr>
<tr>
<td>Table 41</td>
<td>NeqJoin Function Input Data</td>
<td>73</td>
</tr>
<tr>
<td>Table 42</td>
<td>NeqClipExp Function Input Data</td>
<td>74</td>
</tr>
<tr>
<td>Table 43</td>
<td>NeqSquared Function Input Data</td>
<td>74</td>
</tr>
<tr>
<td>Table 44</td>
<td>Definition of NeQuickInputData_st Data Structure</td>
<td>75</td>
</tr>
<tr>
<td>Table 45</td>
<td>Definition of MODIP_st Data Structure</td>
<td>75</td>
</tr>
<tr>
<td>Table 46</td>
<td>Definition of CCIR_st Data Structure</td>
<td>76</td>
</tr>
<tr>
<td>Table 47</td>
<td>Definition of SPoint_st Data Structure</td>
<td>76</td>
</tr>
<tr>
<td>Table 48</td>
<td>Definition of CurrentCCIR_st Data Structure</td>
<td>76</td>
</tr>
<tr>
<td>Table 49</td>
<td>Definition of LayerProperties_st Data Structure</td>
<td>77</td>
</tr>
<tr>
<td>Table 50</td>
<td>Definition of Geometry_st Data Structure</td>
<td>77</td>
</tr>
<tr>
<td>Table 51</td>
<td>Definition of IntegrateData_st Data Structure</td>
<td>77</td>
</tr>
</tbody>
</table>
1. Introduction

1.1. Document Scope

This document complements the Galileo OS SIS ICD [Annex A [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 kilometers up to several thousand kilometers 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 1m$^2$, 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, F1 and F2, the latter being largely responsible for the ionospheric effects which typically affect GNSS applications.

1. 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].
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:

$$d_{igr} = \frac{40.3}{f^2} \cdot \int_{path} N \cdot dl = \frac{40.3}{f^2} \cdot STEC$$

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 millimeter level and may be neglected for code ranging. The effect 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) [Annex A [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 [Annex A [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 [Annex A [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](image)
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 [Annex A [1]], the Effective Ionisation Level, $A_Z$, is determined from three ionospheric coefficients (broadcast within the navigation message)

$$A_Z = a_{i0} + a_{i1} \times \text{MODIP} + a_{i2} \times (\text{MODIP})^2$$  \hspace{1cm} \text{Eq. 2}

as follows:

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 $(\phi, \lambda, h)_i$, satellite position $(\phi, \lambda, h)_j$ and time (time of day and month)

Obtain receiver MODIP$_U$ using $\phi_i$, $\lambda_i$.

Obtain Effective Ionisation Level $A_Z_U$ using eq. (2) with MODIP$_U$ and broadcast coefficients $(a_{i0}, a_{i1}, a_{i2})$

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

for each integration point in the path

Call NeQuick routine to obtain electron density with $A_Z_U$ 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:
** Note that ‘sfu’ (solar flux unit) is not a SI unit but can be converted as:

\[ 1 \text{ sfu} = 10^{-22} \text{ W/(m}^2\text{Hz)} \]

**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 [Annex A [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 [Annex A [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:

<table>
<thead>
<tr>
<th>Region</th>
<th>MODIP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>60deg &lt; MODIP ≤ 90deg</td>
</tr>
<tr>
<td>2</td>
<td>30deg &lt; MODIP ≤ 60deg</td>
</tr>
<tr>
<td>3</td>
<td>-30deg ≤ MODIP &lt; 30deg</td>
</tr>
<tr>
<td>4</td>
<td>-60deg ≤ MODIP &lt; -30deg</td>
</tr>
<tr>
<td>5</td>
<td>-90deg ≤ MODIP &lt; -60deg</td>
</tr>
</tbody>
</table>

![Figure 2. MODIP regions associated to different ionospheric characteristics](image)

Fields have been reserved in the Galileo OS SIS ICD [Annex A [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 Mode

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” \(f_0E, f_0F_1, f_0F_2\) (critical frequencies) and \(M(3000)\) \(F_2\) (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 \(f_0E, f_0F_1, f_0F_2\) and \(M(3000)F_2\). For \(f_0E\) derivation, a modified formulation of that due to John Titheridge is selected and \(f_0F_1\) is selected as being equal to \(1.4 \times f_0E\) during daytime and zero during night-time, respectively. For the calculation of \(f_0F_2\) 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:

\[
\text{Epst}(X, Y, Z, W) = \frac{X \cdot \exp\left(\frac{W - Y}{Z}\right)}{\left(1 + \exp\left(\frac{W - Y}{Z}\right)\right)^2}
\]

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.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Constant description</th>
<th>Value</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>DR</td>
<td>Degree to radian conversion factor</td>
<td>(\pi/180)</td>
<td>rad/deg</td>
</tr>
<tr>
<td>RD</td>
<td>Radian to degree conversion factor</td>
<td>(180\pi)</td>
<td>deg/rad</td>
</tr>
<tr>
<td>(X_0)</td>
<td>Zenith angle at night-day transition</td>
<td>86.23</td>
<td>deg</td>
</tr>
<tr>
<td>(R_E)</td>
<td>Earth mean radius</td>
<td>6371.2</td>
<td>km</td>
</tr>
</tbody>
</table>

Table 2. Constants definition
2.5.3. Complementary Files

2.5.3.1. MODIP Grid

The MODIP grid allows to estimate MODIP $\mu$ [deg] at a given point $(\varphi, \lambda)$ 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 $(\varphi, \lambda)$ interpolating with the 4x4-points grid surrounding the desired element $(\varphi, \lambda)$.

The file `modipNeQG_wrapped.asc` contains the values of MODIP $\mu$ (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 step 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 $f_0F2$ and $M(3000)F2$ as described in [Annex A [4]]. These coefficients are stored in the ccirXX.asc files and include the spherical harmonic coefficients representing the development of monthly median $f_0F2$ and $M(3000)F2$ all over the world. The coefficients correspond to low (Sun Spot Number=0) and high (Sun Spot Number=100) solar activity conditions. For other Sun Spot Number activity the coefficients must be interpolated (or extrapolated) to obtain the values corresponding to the required solar activity.

Each file ccirXX.asc contains 2858 values sequentially organized as follows: $[f2_{1,1,1}, f2_{1,1,2}, \ldots, f2_{1,1,13}, f2_{1,2,1}, \ldots, f2_{1,2,13}, \ldots, f2_{1,76,1}, f2_{1,76,2}, \ldots, f2_{1,76,13}, f2_{2,1,1}, \ldots, f2_{2,1,2}, \ldots, f2_{2,1,13}, f2_{2,2,1}, \ldots, f2_{2,2,13}, \ldots, f2_{2,76,1}, f2_{2,76,2}, \ldots, f2_{2,76,13}, \ldots, f3_{1,1,1}, \ldots, f3_{1,1,9}, f3_{1,2,1}, \ldots, f3_{1,2,9}, \ldots, f3_{1,49,1}, f3_{1,49,2}, \ldots, f3_{1,49,9}, f3_{2,1,1}, \ldots, f3_{2,1,9}, f3_{2,2,1}, f3_{2,2,9}, \ldots, f3_{2,49,1}, f3_{2,49,2}, \ldots, f3_{2,49,9}]$.

(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 $\lambda$ [deg], Universal Time $UT$ [hours and decimals].

Output: local time $LT$ [hours and decimals].

\[
LT = UT + \frac{\lambda}{15} \quad \text{Eq. 4}
\]
2.5.4.2. Read modipNeQG_wrapped.Asc Values in an Array

\[ \text{stModip}: \quad \text{stModip}_{i,j} \quad i=1,\ldots,39; \quad j=-1,\ldots,37; \]  

2.5.4.3. Compute MODIP

Inputs:  latitude \( \varphi \) [deg], longitude \( \lambda \) [deg], array \( \text{stModip} \) (of MODIP values).

Output: MODIP \( \mu \) [deg]

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

\[ l = \text{int} \left( \frac{\lambda + 180}{10} \right) - 2 \]  

If \( l < 0 \) use:

\[ l = l + 36 \]  

If \( l > 33 \) use:

\[ l = l - 36 \]

Compute:

\[ a = \frac{\varphi + 90}{5} + 1 \]

\[ x = a - \text{int}(a) \]

\[ i = \text{int}(a) - 2 \]

For \( k=1,4; \) for \( j=1,4 \) build \( z_{j,k} \) as:

\[ z_{j,k} = \text{stModip}_{i+j,i+k} \]  

For \( k=1,4 \) compute

\[ z_k = z_x(z_{1,k},z_{2,k},z_{3,k},z_{4,k},x) \]

using the interpolation function described in 2.5.7.1.

Finally compute

\[ b = \frac{\lambda + 180}{10} \]

\[ y = b - \text{int}(b) \]

and, using the interpolation function described in 2.5.7.1, calculate

\[ \mu = z_x(z_1,z_2,z_3,z_4,y) \]

2.5.4.4. Effective Ionization Level \( Az \)

Compute the Effective Ionization Level \( Az \) at the given receiver location (having MODIP \( \mu \)) 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 \( \mu \) [deg].

Output:  Effective Ionisation Level \( Az \):

If \( a_0 = a_1 = a_2 = 0 \)

\[ Az = 63.7 \]  

else

\[ Az = a_0 + a_1 \mu + a_2 \mu^2 \]

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

Inputs: month $mth$, Universal Time UT [hours]

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

Compute day of year at the middle of the month:

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

Compute time [days]:

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

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:

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

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

2.5.4.7. **Solar Zenith Angle**

Compute solar zenith angle $\chi$ [deg] for the given location.

Inputs: latitude $\varphi$ [deg], local time LT [hours], $\sin(\delta_{\text{Sun}}), \cos(\delta_{\text{Sun}})$

Outputs: solar zenith angle $\chi$ [deg].

Compute

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

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

2.5.4.8. **Effective Solar Zenith Angle**

Compute the effective solar zenith angle $\chi_{\text{eff}}$ [deg] as a function of the solar zenith angle $\chi$ [deg] and the solar zenith angle at day night transition $\chi_0$ [deg].

Inputs: solar zenith angle $\chi$ [deg], $\chi_0$ [deg]

Output: effective solar zenith angle $\chi_{\text{eff}}$ [deg].

Being

$$\chi_0 = 86.23^\circ \quad \text{Eq. 28}$$
Then \[ \chi_{eff} = \frac{\chi + [90 - 0.24 \cdot \exp(20 - 0.2 \cdot \chi)] \cdot \exp(12(\chi - \chi_0))}{1 + \exp(12(\chi - \chi_0))} \] Eq. 29

2.5.5. Model Parameters

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

2.5.5.1. \( f_oE \) and \( NmE \)

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

Inputs: latitude \( \phi \) [deg], Effective Ionisation Level \( Az \), effective solar zenith angle \( \chi_{eff} \) [deg], month \( mth \).

Output: \( f_oE \) [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:

\[ ee = \exp(0.3 \cdot \phi) \] Eq. 33
\[ seasp = seas \cdot \frac{ee - 1}{ee + 1} \] Eq. 34

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

The \( E \) layer maximum density \( NmE \) \([10^{11} \text{ m}^{-3}]\) as a function of \( f_oE \) [MHz] is computed as:

\[ NmE = 0.124 \cdot f_oE^2 \] Eq. 36

2.5.5.2. \( f_oF1 \) and \( NmF1 \)

The \( F1 \) layer critical frequency \( f_oF1 \) [MHz] as a function of the solar zenith angle \( \chi \) and the solar zenith angle at day night transition \( \chi_0 \) is computed as:

Inputs: \( E \) layer critical frequency \( f_oE \) [MHz], solar zenith angle \( \chi \) [deg], solar zenith angle at day night transition \( \chi_0 \) [deg]

Output: \( f_oF1 \) [MHz]

\[ f_oF1 = 1.4 \cdot f_oE \] if \( f_oE \geq 2.0 \text{ MHz} \),
\[ f_oF1 = 0 \] if \( f_oE < 2.0 \text{ MHz} \), Eq. 37

\( f_oF1 \) is reduced by 15% if too close to \( f_oF2 \)

The \( F \) layer maximum density \( NmF1 \) \([10^{11} \text{ m}^{-3}]\) as a function of \( f_oF1 \) [MHz] is computed as:
2.5.5.3. \( \text{foF2 and NmF2; M(3000)F2} \)

2.5.5.3.1. Read CcirXX.Asc Values

Input: Month \( mth \)

Outputs: \( F_2, Fm_3 \)

Select the file name to read: \( XX = mth + 10 \) Eq. 40
(e.g. ccir21.asc for November) and store the file content in the two arrays of coefficients:

- coefficients for \( \text{foF2} \)
  \[ F_2: \quad f_{2i,j,k}^{f_{2}} = \hat{a}_{2i,j,k}^{f_{2}} \left( 1 - \frac{A_{z_R}}{100} \right) + \hat{a}_{2i,j,k}^{f_{2}} \frac{A_{z_R}}{100} \] Eq. 41

- coefficients for \( M(3000)F_2 \)
  \[ Fm_3: \quad f_{3i,j,k}^{m_{3}} = \hat{a}_{3i,j,k}^{m_{3}} \left( 1 - \frac{A_{z_R}}{100} \right) + \hat{a}_{3i,j,k}^{m_{3}} \frac{A_{z_R}}{100} \] Eq. 42

2.5.5.3.2. Interpolate ITU-R Coefficients for \( A_{z_R} \)

Compute \( AF_2 \), the array of interpolated coefficients for \( \text{foF2} \) and \( Am_3 \), the array of interpolated coefficients for \( M(3000)F_2 \)

Inputs: \( F_2, Fm_3, A_{z_R} \)

Outputs: \( AF_2, Am_3 \)

Compute the array of interpolated coefficients for \( \text{foF2} \)

\[ AF_2: \quad a_{f_{2}}^{i,j,k} = \hat{a}_{2i,j,k}^{f_{2}} \left( 1 - \frac{A_{z_R}}{100} \right) + \hat{a}_{2i,j,k}^{f_{2}} \frac{A_{z_R}}{100} \] Eq. 43

\( AF_2 \) elements are calculated by linear combination of the elements of \( F_2 \):

Compute the array of interpolated coefficients for \( M(3000)F_2 \)

\[ Am_3: \quad a_{m_{3}}^{i,j,k} = \hat{a}_{3i,j,k}^{m_{3}} \left( 1 - \frac{A_{z_R}}{100} \right) + \hat{a}_{3i,j,k}^{m_{3}} \frac{A_{z_R}}{100} \] Eq. 45

\( Am_3 \) elements are calculated by linear combination of the elements of \( Fm_3 \):

2.5.5.3.3. Compute Fourier Time Series for \( \text{foF2} \) 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 \( \text{foF2} \) and \( M(3000)F_2 \)

The vector \( CF_2 \) has 76 elements:

\[ CF_2: \quad c_{f_{2}}^{i} = \hat{a}_{f_{2}}^{i} \quad l=1,...,76 \] Eq. 47
The vector Cm3 has 49 elements:
\[ Cm3: \quad cm3_l, \quad l=1,..,49 \]  

Eq. 48

Compute the time argument:
\[ T=(15\cdot UT-180)\cdot DR \]  

Eq. 49

For \( i=1,..,76 \) calculate the Fourier time series for \( foF2 \):
\[ cf2_i = af2_i,1 + \sum_{k=1}^{6} \left[ af2_i,2k \sin(kT) + af2_i,2k+1 \cos(kT) \right] \]  

Eq. 50

For \( i=1,..,49 \) calculate the Fourier time series for \( M(3000)F2 \):
\[ cm3_i = am3_i,1 + \sum_{k=1}^{4} \left[ am3_i,2k \sin(kT) + am3_i,2k+1 \cos(kT) \right] \]  

Eq. 51

### 2.5.5.3.4. Compute \( foF2 \) and \( M(3000)F2 \) by Legendre Calculation

**Inputs:** MODIP \( \mu \) [deg], latitude \( \phi \) [deg], longitude \( \lambda \) [deg], vector \( CF2 \) of the coefficients for Legendre combination for \( foF2 \), vector \( Cm3 \) of the coefficients for Legendre combination for \( M(3000)F2 \)

**Outputs:** \( foF2 \) [MHz], \( M(3000)F2 \)

Define vectors containing sine and cosine of the coordinates:

\[ M: \quad m_k, \quad k=1,..,12 \]  

Eq. 52

\[ P: \quad p_n, \quad n=2,..,9 \]  

Eq. 53

\[ S: \quad s_n, \quad n=2,..,9 \]  

Eq. 54

\[ C: \quad c_n, \quad n=2,..,9 \]  

Eq. 55

Compute MODIP coefficients
\[ m_1 = 1 \]  

Eq. 56

and for \( k=2,..,12 \)
\[ m_k = \sin^{k-1}(\mu \cdot DR) \]  

Eq. 57

Compute latitude and longitude coefficients
for \( n=2,..,9 \)
\[ p_n = \cos^{n-1}(\phi \cdot DR) \]  

Eq. 58

\[ s_n = \sin((n-1) \cdot \lambda \cdot DR) \]  

Eq. 59

\[ c_n = \cos((n-1) \cdot \lambda \cdot DR) \]  

Eq. 60

Compute \( foF2 \)

Order 0 term:
\[ foF2_1 = \sum_{k=1}^{12} cf2_k m_k \]  

Eq. 61

having the increased Legendre grades for \( foF2 \) in a vector:
\[ Q: \quad q_n, \quad n=1,..,9 \]  

Eq. 62

\[ Q=(12,12,9,5,2,1,1,1,1) \]  

Eq. 63
for computational efficiency, define also:

\[ K: \quad k_n = k_{n-1} + 2q_{n-1} \quad \text{Eq. 64} \]

\[ k_1 = -q_1 \quad \text{Eq. 65} \]

and for \( n=2,\ldots,9 \)

\[ k_n = k_{n-1} + 2q_{n-1} \quad \text{Eq. 66} \]

for \( n=2,\ldots,9 \) compute the higher order terms:

\[
foF^2_n = \sum_{k=1}^{q_n} (cF^2_{k+n+2k-1}c_n + cF^2_{k+n+2k}s_n)m_k p_n
\]

Eq. 67

Finally sum the terms to obtain \( foF^2 \):

\[
foF^2 = \sum_{n=1}^{q} foF^2_n
\]

Eq. 68

Compute \( M(3000)F^2 \)

Order 0 term:

\[
M(3000)F^2_0 = \sum_{k=1}^{7} cm^3_k m_k
\]

Eq. 69

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

\[ R: \quad r_n \quad n=1,\ldots,7 \]

Eq. 70

\[ R = (7,8,6,3,2,1,1) \]

Eq. 71

for computational efficiency, define also:

\[ H: \quad h_n = h_{n-1} + 2r_{n-1} \quad \text{Eq. 72} \]

\[ h_1 = -r_1 \quad \text{Eq. 73} \]

and for \( n=2,\ldots,7 \)

\[ h_n = h_{n-1} + 2r_{n-1} \quad \text{Eq. 74} \]

for \( n=2,\ldots,7 \), compute the higher order terms

\[
M(3000)F^2_n = \sum_{k=1}^{r_n} (cm^3_{h+n+2k-1}c_n + cm^3_{h+n+2k}s_n)m_k p_n
\]

Eq. 75

Finally sum the terms:

\[
M(3000) = \sum_{n=1}^{7} M(3000)F^2_n
\]

Eq. 76

To compute \( NmF^2 \) use:

\[ NmF^2 = 0.124 \cdot foF^2^2 \]

Eq. 77

where \( NmF^2 \) is in \([10^{11}]m^{-3}\).
### 2.5.5.4. \( \text{hmE} \)

The E layer maximum density height \( \text{hmE} \) [km] is defined as a constant:

\[
\text{hmE} = 120 \quad \text{Eq. 78}
\]

### 2.5.5.5. \( \text{hmF1} \)

Compute the F1 layer maximum density height \( \text{hmF1} \) [km]:

**Inputs:** \( \text{NmF1} \) \([10^{11} \text{ m}^{-3}]\), Dip \( l \) [deg]

**Output:** \( \text{hmF1} \) [km]

\[
\text{hmF1} = \frac{\text{hmF2} + \text{hmE}}{2} \quad \text{Eq. 79}
\]

### 2.5.5.6. \( \text{hmF2} \)

Compute the F2 layer maximum density height \( \text{hmF2} \) [km].

**Inputs:** \( \text{foE} \) [MHz], \( \text{foF2} \) [MHz], \( M(3000)F2 \)

**Output:** \( \text{hmF2} \) [km]

\[
\text{hmF2} = \frac{1490 \cdot M \cdot \sqrt{0.0196 \cdot M^2 + 1}}{M + \Delta M} - 176 \quad \text{Eq. 80}
\]

Where

\[
M = M(3000)F2 \quad \text{Eq. 81}
\]

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

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

and the ratio \( \rho \) is computed as:

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

### 2.5.5.7. \( B2\text{bot}, B1\text{top}, B1\text{bot}, BE\text{top}, BE\text{bot} \)

Compute the thickness parameters \( B2\text{bot}, B1\text{top}, B1\text{bot}, BE\text{top}, BE\text{bot} \) [km]

<table>
<thead>
<tr>
<th><strong>Inputs</strong></th>
<th><strong>Outputs</strong></th>
<th><strong>Definition</strong></th>
</tr>
</thead>
</table>
| \( \text{NmF2} \) \([10^{11} \text{ m}^{-3}]\) | \( B2\text{bot} \) [km] | \[
B2\text{bot} = \frac{0.385 \cdot \text{NmF2}}{0.01 \cdot \exp(-3.467 + 0.857 \cdot \ln(\text{foF2}) + 2.02 \cdot \ln(M))}
\]

where \( M = M(3000)F2 \)  

<table>
<thead>
<tr>
<th><strong>Inputs</strong></th>
<th><strong>Outputs</strong></th>
<th><strong>Definition</strong></th>
</tr>
</thead>
</table>
| \( \text{hmF1} \) [km] | \( B1\text{top} \) [km] | \[
B1\text{top} = 0.3 \cdot (\text{hmF2} - \text{hmF1})
\]

<table>
<thead>
<tr>
<th><strong>Inputs</strong></th>
<th><strong>Outputs</strong></th>
<th><strong>Definition</strong></th>
</tr>
</thead>
</table>
| \( \text{hmE} \) [km] | \( B1\text{bot} \) [km] | \[
B1\text{bot} = 0.5 \cdot (\text{hmF1} - \text{hmE})
\]

© European Union 2015  
Document subject to terms of use and disclaimers p.i  
Ionospheric Correction Algorithm for Galileo Single Frequency Users, Issue 1.1, June 2015
2.5.5.8. A1

Compute the F2 layer amplitude $A_1\ [10^{11}\ m^{-3}]$.

Inputs: $NmF_2\ [10^{11}\ m^{-3}]$

Output: $A_1\ [10^{11}\ m^{-3}]$

$$A_1 = 4 \cdot NmF_2$$  \hspace{1cm} Eq. 90

2.5.5.9. A2 and A3

Compute the F1 layer amplitude $A_2\ [10^{11}\ m^{-3}]$ and the E layer amplitude $A_3\ [10^{11}\ m^{-3}]$.

Inputs: $NmE\ [10^{11}\ m^{-3}], NmF_1\ [10^{11}\ m^{-3}], A_1[10^{11}\ m^{-3}], hmF_2\ [\text{km}], hmF_1\ [\text{km}], hmE\ [\text{km}], BEtop\ [\text{km}], B_1\text{\_bot}\ [\text{km}], B_2\text{\_bot}\ [\text{km}]$

Output: $A_2\ [10^{11}\ m^{-3}], A_3\ [10^{11}\ m^{-3}]$

if $foF_1 < 0.5$: $A_2 = 0$  \hspace{1cm} Eq. 91

$$A_3 = 4.0 \cdot [NmE - \text{Epst}(A_1, hmF_2, B_2\text{\_bot, hmE})]$$  \hspace{1cm} Eq. 92

if $foF_1 \geq 0.5$: $A_3 = 4.0 \cdot NmE$  \hspace{1cm} Eq. 93

Repeat 5 times the iterations below:

$$A_2a = 4.0 \cdot [NmF_1 - \text{Epst}(A_1, hmF_2, B_2\text{\_bot, hmF_1}) - \text{Epst}(A_3, hmE, BEtop, hmF_1)]$$  \hspace{1cm} Eq. 94

$$A_2a = \frac{A_2a \cdot \exp(A_2a - 0.80 \cdot NmF_1) + 0.80 \cdot NmF_1}{1 + \exp(A_2a - 0.80 \cdot NmF_1)}$$  \hspace{1cm} Eq. 95

$$A_3a = 4.0 \cdot [NmE - \text{Epst}(A_2a, hmF_1, B_1\text{\_bot, hmE}) - \text{Epst}(A_1, hmF_2, B_2\text{\_bot, hmE})]$$  \hspace{1cm} Eq. 96

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

$$A_2 = A_2a$$  \hspace{1cm} Eq. 97

$$A_3 = \frac{A_3a \cdot \exp(60 \cdot (A_3a - 0.005)) + 0.05}{1 + \exp(60 \cdot (A_3a - 0.005))}$$  \hspace{1cm} Eq. 98

2.5.5.10. Shape Parameter $k$

Compute the shape parameter $k$.

Inputs: $mth, NmF_2\ [10^{11}\ m^{-3}], hmF_2\ [\text{km}], B_2\text{\_bot}\ [\text{km}]$

Output: $k$

First compute the auxiliary parameter $ka$:

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

$$ka = 6.705 - 0.014 \cdot A_2 - 0.008 \cdot hmF_2$$  \hspace{1cm} Eq. 99
if \( mth = 1,2,3,10,11,12 \)

\[
ka = -7.77 + 0.097 \cdot \left(\frac{hmF2}{B2bot}\right)^2 + 0.153 \cdot NmF2
\]

Eq. 100

Then compute the auxiliary parameter \( kb \):

\[
kb = \frac{ka \cdot \exp(ka - 2) + 2}{1 + \exp(ka - 2)}
\]

Eq. 101

Eventually compute:

\[
k = \frac{8 \cdot \exp(kb - 8) + kb}{1 + \exp(kb - 8)}
\]

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 \( H_a \):

\[
H_a = k \cdot B2bot
\]

Eq. 103

Then compute the auxiliary parameters \( x \) and \( v \) as follows:

\[
x = \frac{H_a - 150}{100}
\]

Eq. 104

\[
v = (0.041163 \cdot x - 0.183981) \cdot x + 1.424472
\]

Eq. 105

Eventually compute

\[
H_0 = \frac{H_a}{v}
\]

Eq. 106

2.5.6. Electron Density Computation

To compute the electron density \( N = N(h, \phi, \lambda, a_\phi, a_\lambda, mth, UT) \) at a given point (identified by the coordinates \( h, \phi, \lambda \)) at a given time (mth, UT) and using a given set of Effective Ionisation Level \( Az \) derived with the Effective Ionisation Level coefficients \( (a_{i\phi}, a_{i\lambda}) \) 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

\[
\text{if } h \leq hmF2 \quad \text{Eq. 107}
\]

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

\[
\text{if } h > hmF2 \quad \text{Eq. 108}
\]

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], \( A1 [10^{11} \text{ m}^{-3}], A2 [10^{11} \text{ m}^{-3}], A3 [10^{11} \text{ m}^{-3}], hmF2 \) [km], \( hmF1 \) [km], \( hmE \) [km], \( B2bot \) [km], \( B1top \) [km], \( B1bot \) [km], \( BExtop \) [km], \( BEbot \) [km].
Output: (bottomside) electron density $N [m^{-3}]$.

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

$$BE = \begin{cases} BE_{top} & \text{if } h > hmE \\ BE_{bot} & \text{if } h \leq hmE \end{cases}$$

**Eq. 109**

$$BF1 = \begin{cases} BF1_{top} & \text{if } h > hmF1 \\ BF1_{bot} & \text{if } h \leq hmF1 \end{cases}$$

**Eq. 110**

Compute the exponential arguments for each layer:

$$\alpha_1 = \frac{h - hmF2}{B2bot}$$

**Eq. 111**

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

**Eq. 112**

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

**Eq. 113**

For each $i = 1,3$ compute:

$$s_i = \begin{cases} 0 & \text{if } |\alpha_i| > 25 \\ \frac{\exp(\alpha_i)}{(1 + \exp(\alpha_i))^2} & \text{if } |\alpha_i| \leq 25 \end{cases}$$

**Eq. 114**

If $h \geq 100$ km compute the electron density as:

$$N = (s_1 + s_2 + s_3) \times 10^{11}$$

**Eq. 115**

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

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

**Eq. 116**

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

**Eq. 117**

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

**Eq. 118**

and the Chapman parameters:

$$BC = 1 - 10 \frac{\sum_{i=1}^{3} s_i ds_i}{\sum_{i=1}^{3} s_i}$$

**Eq. 119**

$$z = \frac{h - 100}{10}$$

**Eq. 120**

Then compute the electron density as:

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

**Eq. 121**

2.5.6.2. The Topside Electron Density

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

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

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 $|x^2| \leq 10^{10}$

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

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 $h_{mE}$, $h_{mF1}$, $h_{mF2}$, $A_1$, $A_2$, $A_3$, $B_{2bot}$, $B_{1top}$ $B_{1bot}$, $B_{Etop}$, $B_{Ebot}$, $N_{mf2}$, $H_0$, then compute the integration of the electron density (bottomside or topside) as function of height:

$$ TEC = \int_{h_1}^{h_2} N(h)dh $$  \hspace{1cm} Eq. 139

where:

$$ h_1 = r_1 - R_E $$  \hspace{1cm} Eq. 140

$$ h_2 = r_2 - R_E $$  \hspace{1cm} Eq. 141

2.5.8.1.1. Vertical TEC Numerical Integration

Inputs:

- Integration endpoints $h_1$ [km], $h_2$ [km]
- Target integration accuracy $\varepsilon$ - 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$^{-3}$], $h_{mf2}$ [km], $h_{mf1}$ [km], $h_{me}$ [km], $B_{2bot}$ [km], $B_{1top}$ [km], $B_{1bot}$ [km], $B_{Etop}$ [km], $B_{Ebot}$ [km], $N_{mf2}$ [m$^{-3}$], $H_0$ [km]

Output: TEC [TECU]

Being $\varphi$, $\lambda$, and all model parameters fixed during the integration, in the following a simplified notation is used:

$$ N(h) = \begin{cases} 
\text{bottomside } N & \text{if } h \leq h_{mf2} \\
\text{topside } N & \text{if } h > h_{mf2}
\end{cases} $$  \hspace{1cm} Eq. 142

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

Start the calculation using 8 points:

$$ n = 8 $$  \hspace{1cm} Eq. 143
Repeat the following computations until the target integration accuracy ($\varepsilon$) is obtained (default tolerance ($\varepsilon$) 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 $\varepsilon$ is achieved:

if 

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

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 $\alpha$ is an angle in [deg], $\tilde{\alpha}$ is the same angle in [rad]:

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

2.5.8.2.2. Zenith Angle Computation

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

$$\cos(\delta) = \sin(\varphi_1) \sin(\varphi_2) + \cos(\varphi_1) \cos(\varphi_2) \cos(\lambda_2 - \lambda_1) \quad \text{Eq. 153}$$

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

$$\zeta = \text{atan2} \left( \sin(\delta), \cos(\delta) - \frac{r_1}{r_2} \right) \quad \text{Eq. 155}$$
being $\delta$ 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 $(\mp\pi, \pi)$.

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 $\varphi_p$ [deg] and ray perigee longitude $\lambda_p$ [deg].

Calculate $r_p$:

$$r_p = r_1 \sin(\zeta)$$  \hspace{1cm} Eq. 156

Calculate $\varphi_p$:

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

$$\varphi_p = \begin{cases} \zeta & \text{if } \varphi_1 > 0 \\ -\zeta & \text{if } \varphi_1 < 0 \end{cases}$$  \hspace{1cm} Eq. 157

otherwise use

$$\sin(\delta) = \frac{\sin(\lambda_2 - \lambda_1) \cos(\bar{\varphi}_2)}{\sin(\bar{\delta})}$$  \hspace{1cm} Eq. 158

$$\cos(\bar{\delta}) = \frac{\sin(\bar{\varphi}_2) - \cos(\bar{\delta}) \sin(\bar{\varphi}_1)}{\sin(\bar{\delta}) \cos(\bar{\varphi}_1)}$$  \hspace{1cm} Eq. 159

$$\delta_p = \frac{\pi}{2} - \zeta$$  \hspace{1cm} Eq. 160

$$\sin(\bar{\varphi}_p) = \sin(\bar{\varphi}_1) \cos(\delta_p) - \cos(\bar{\varphi}_1) \sin(\delta_p) \cos(\bar{\delta})$$  \hspace{1cm} Eq. 161

$$\cos(\bar{\varphi}_p) = \sqrt{1 - \sin^2(\bar{\varphi}_p)}$$  \hspace{1cm} Eq. 162

$$\bar{\varphi}_p = \text{atan2}(\sin(\bar{\varphi}_p), \cos(\bar{\varphi}_p))$$  \hspace{1cm} Eq. 163

Figure 3. Geometry of zenith angle computation
Calculate $\lambda_p$:

If $|\phi_1 - 90^\circ| < 10^{-10}$ use

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

Eq. 164

otherwise use

$$\sin(\tilde{\lambda}_1 - \tilde{\lambda}_p) = \frac{-\sin(\tilde{\sigma}) \sin(\tilde{\delta}_p)}{\cos(\tilde{\phi}_p)}$$

Eq. 165

$$\cos(\tilde{\lambda}_1 - \tilde{\lambda}_p) = \frac{\cos(\tilde{\delta}_p) - \sin(\tilde{\phi}_1) \sin(\tilde{\phi}_p)}{\cos(\tilde{\phi}_1) \cos(\tilde{\phi}_p)}$$

Eq. 166

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

Eq. 167

being $\sigma$ the azimuth of $P_2$ seen from $P_1$ and $\delta_p$ the Earth angle between $P_1$ and the ray-perigee $P_p$.

![Geometry of ray perigee computation](image)

2.5.8.2.4. Great Circle Properties

Compute the great circle angle $\psi$ from ray-perigee to satellite:

If $|\phi_p| - 90^\circ| < 10^{-10}$ use

$$\psi = |\phi_2 - \phi_p|$$

Eq. 168

otherwise use

$$\cos(\tilde{\psi}) = \sin(\tilde{\phi}_p) \sin(\tilde{\phi}_2) + \cos(\tilde{\phi}_p) \cos(\tilde{\phi}_2) \cos(\tilde{\lambda}_2 - \tilde{\lambda}_p)$$

Eq. 169

$$\sin(\tilde{\psi}) = \sqrt{1 - \cos^2(\tilde{\psi})}$$

Eq. 170

$$\tilde{\psi} = \text{atan2}(\sin(\tilde{\psi}), \cos(\tilde{\psi}))$$

Eq. 171
Compute sine and cosine of azimuth $\sigma$ of satellite as seen from ray-perigee $P_p$:

if $|\varphi_p - 90^\circ| < 10^{-10}$ use

$$\sin(\tilde{\sigma}_p) = 0$$  \hspace{1cm} \text{Eq. 172}
$$\cos(\tilde{\sigma}_p) = \begin{cases} -1 & \text{if } \varphi_p > 0 \\ 1 & \text{if } \varphi_p < 0 \end{cases}$$  \hspace{1cm} \text{Eq. 173}

otherwise use

$$\sin(\tilde{\sigma}_p) = -\frac{\cos(\tilde{\varphi}_2) \sin(\tilde{\lambda}_2 - \tilde{\lambda}_p)}{\sin(\tilde{\psi})}$$  \hspace{1cm} \text{Eq. 174}
$$\cos(\tilde{\sigma}_p) = -\frac{\sin(\tilde{\varphi}_2) - \sin(\tilde{\varphi}_p) \cos(\tilde{\psi})}{\cos(\tilde{\varphi}_p) \sin(\tilde{\psi})}$$  \hspace{1cm} \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:

$$s_1 = \sqrt{r_1^2 - r_p^2}$$  \hspace{1cm} \text{Eq. 176}
$$s_2 = \sqrt{r_2^2 - r_p^2}$$  \hspace{1cm} \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 $P_p$, $(r_p, \varphi_p, \lambda_p)$ the ray perigee coordinates and $\sin \alpha_p$, $\cos \alpha_p$ 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 $(r_p, \varphi_p, \lambda_p)$, 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], $\varphi_s$ [deg], $\lambda_s$ [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 $h_s$:

$$h_s = \sqrt{s^2 + r_p^2} - R_E$$  \hspace{1cm} \text{Eq. 178}

being $R_E$ the Earth mean radius.

Calculate great circle parameters:

$$\tan \delta_s = \frac{s}{r_p}$$  \hspace{1cm} \text{Eq. 179}
$$\cos(\delta_s) = \frac{1}{\sqrt{1 + \tan^2(\delta_s)}}$$  \hspace{1cm} \text{Eq. 180}
$$\sin(\delta_s) = \tan(\delta_s) \cos(\delta_s)$$  \hspace{1cm} \text{Eq. 181}
Calculate $\varphi_s$:
\[
\sin(\varphi_s) = \sin(\varphi_p) \cos(\delta_s) + \cos(\varphi_p) \sin(\delta_s) \cos(\varphi_s) \quad \text{Eq. 182}
\]
\[
\cos(\varphi_s) = \sqrt{1 - \sin^2(\varphi_s)} \quad \text{Eq. 183}
\]
\[
\varphi_s = \text{atan2}(\sin(\varphi_s), \cos(\varphi_s)) \cdot RD \quad \text{Eq. 184}
\]

Calculate $\lambda_s$:
\[
\sin(\lambda_s - \lambda_p) = \sin(\delta_s) \sin(\varphi_p) \cos(\varphi_s) \quad \text{Eq. 185}
\]
\[
\cos(\lambda_s - \lambda_p) = \cos(\delta_s) - \sin(\varphi_p) \sin(\varphi_s) \quad \text{Eq. 186}
\]
\[
\lambda_s = \left[ \text{atan2}(\sin(\lambda_s - \lambda_p), \cos(\lambda_s - \lambda_p)) + \lambda_p \right] \cdot RD \quad \text{Eq. 187}
\]

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]
- $\varphi_1$, latitude of point $P_1$ [deg]
- $\lambda_1$, longitude of point $P_1$ [deg]
- $h_2$, height of point $P_2$ [km]
- $\varphi_2$, latitude of point $P_2$ [deg]
- $\lambda_2$, 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:
\[
TEC = \int_{s_1}^{s_a} N(s)ds + \int_{s_a}^{s_b} N(s)ds + \int_{s_b}^{s_2} N(s)ds \quad \text{Eq. 190}
\]

To compute each integral, the algorithm described in section 2.5.8.2.8 can be used as
where the parameter $\varepsilon$ indicates the integration accuracy. Here we assume:

$\varepsilon = 0.001$ for the integration between $s_0$ and $s_a$

$\varepsilon = 0.01$ 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 $\varepsilon$ - relative difference between two integration steps, recommended maximum value $\varepsilon = 10^{-3}$
- Ray-perigee parameters: $r_p$ [km], $\sin(\bar{\varphi}_p)$, $\cos(\bar{\varphi}_p)$, $\sin(\bar{\sigma}_p)$, $\cos(\bar{\sigma}_p)$, $\lambda_p$ [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:

$$(h(s), \varphi(s), \lambda(s)) = c(s, \sin(\bar{\varphi}_p), \cos(\bar{\varphi}_p), \sin(\bar{\sigma}_p), \cos(\bar{\sigma}_p), \lambda_p)$$  \hspace{1cm} Eq. 192

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

$$f(s) := N(h(s), \varphi(s), \lambda(s), a_0, a_1, a_2, mth, UT)$$  \hspace{1cm} Eq. 193

Start the calculation using 8 points:

$$n = 8$$  \hspace{1cm} Eq. 194

Repeat the following computations until the target integration accuracy ($\varepsilon$) is obtained as follows:

Calculate the integration intervals:

$$\Delta_n = \frac{g_2 - g_1}{n}$$  \hspace{1cm} Eq. 195

$$g = 0.5773502691896 \cdot \Delta_n$$  \hspace{1cm} Eq. 196

$$y = g_1 + \frac{\Delta_n - g}{2}$$  \hspace{1cm} Eq. 197

$$GN_2 = \frac{\Delta_n}{2} \cdot \sum_{i=0}^{n-1} [f(y + i\Delta_n) + f(y + i\Delta_n + g)]$$  \hspace{1cm} Eq. 198
Double the number of points: 
\[ n = 2n \]  
Eq. 199

and define 
\[ GN_1 = GN_2 \]  
Eq. 200

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

if 
\[ |GN_1 - GN_2| > \varepsilon |GN_1| \]  
Eq. 201

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

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

\[ TEC = \left( GN_2 + \frac{GN_2 - GN_1}{15} \right) \times 10^{-13} \]  
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 [Annex A [5]]. The reason to change the calculation of peak plasma frequency for F1 layer is also introduced in [Annex A [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 [Annex A [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 $k$;

• ITU-R version in [Annex A [3]] incorporates a MODIP file obtained with an IGRF geomagnetic field consistent with the field when the CCIR files (used internally in 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.

\[
B_{top}^{F2} = k \cdot B_{bot}^{F2}
\]

\[
k = 3.22 - 0.0538 \cdot f_0 F_2 - 0.00664 \cdot h_m F_2 + 0.113 \frac{h_m F_2}{B_{bot}^{F2}} + 0.00257 \cdot R_{12}
\]

\[
k \geq 1
\]
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 12-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 \( \text{MODIP} \), it may exceptionally happen that for some \( \text{MODIP} \) values, the Effective Ionisation Parameter becomes out of range. The operational range for the Effective Ionisation Parameter is between 0 and 400 sfu and the following condition for out of range values should be used:

Given a user receiver \( U \) with

\[ \text{MODIP} = \mu \]

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 \( Az_U < 0 \) \hspace{1cm} Az_U = 0

if \( Az_U > 400 \) \hspace{1cm} 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


4.2. Reference Documents


5. Annex B – Acronyms and Definitions

5.1. Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC</td>
<td>European Commission</td>
</tr>
<tr>
<td>ESA</td>
<td>European Space Agency</td>
</tr>
<tr>
<td>FOC</td>
<td>Full Operational Capability</td>
</tr>
<tr>
<td>GIOVE</td>
<td>Galileo In Orbit Validation Element</td>
</tr>
<tr>
<td>GNSS</td>
<td>Global Navigation Satellite System</td>
</tr>
<tr>
<td>GSA</td>
<td>European GNSS Agency</td>
</tr>
<tr>
<td>ICTP</td>
<td>Abdus Salam International Center of Theoretical Physics</td>
</tr>
<tr>
<td>IOV</td>
<td>In Orbit Validation</td>
</tr>
<tr>
<td>IGS</td>
<td>International GNSS Service</td>
</tr>
<tr>
<td>MODIP</td>
<td>Modified Dip Latitude</td>
</tr>
<tr>
<td>NSL</td>
<td>Nottingham Scientific Ltd</td>
</tr>
<tr>
<td>OS</td>
<td>Open Service</td>
</tr>
<tr>
<td>RMS</td>
<td>Root Mean Square</td>
</tr>
<tr>
<td>SISICD</td>
<td>Signal In Space Interface Control Document</td>
</tr>
<tr>
<td>SFIono</td>
<td>Galileo Single Frequency Ionospheric Algorithm</td>
</tr>
<tr>
<td>sfu</td>
<td>Solar Flux Units</td>
</tr>
<tr>
<td>STEC</td>
<td>Slant Total Electron Content</td>
</tr>
<tr>
<td>TEC</td>
<td>Total Electron Content</td>
</tr>
<tr>
<td>UERE</td>
<td>User Equivalent Ranging Error</td>
</tr>
</tbody>
</table>

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 *R12* 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 [Annex A [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 sfu. 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:

\[
Az = a_{i0} + a_{i1} \times MODIP + a_{i2} \times (MODIP)^2
\]

**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 [Annex A [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 [Annex A [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 [Annex A [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 [Annex A [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, \( M(3000)F2 = \frac{MUF(3000)F2}{foF2} \). 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 [Annex A [7]]:

\[
I = \tan^{-1}(2 \times \tan(\phi))
\]

Eq. 204

where \( \phi \) 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 \( \text{el m}^{-2} \).

**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 \( \mu \)): defined by the following expression given by Rawer [Annex A [8]]:

\[
\tan(\text{MODIP}) = \frac{l}{\sqrt{\cos(\phi)}}
\]

Eq. 205

where \( l \) is the magnetic dip and \( \phi \) is geographic latitude. MODIP, \( l \) and \( \phi \) are usually expressed in degrees.

**Radio noise flux at 10.7 cm** (F10.7 or \( \Phi \)): 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 sfu with 1 sfu = \( 10^{-22} \text{ W m}^{-2} \text{ Hz}^{-1} \).

\( \Phi \) usually represents monthly mean values although daily values exist. \( \Phi_{12} \) 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 \( \Phi_{12} \) is [Annex A [4]]:

\[
\Phi_{12} = 63.7 + 0.728 \times R_{12} + 8.9 \times 10^{-4} \times R_{12}^2
\]

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:

\[
R = k \times (10 \times g + s)
\]

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.
where \( R \) 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:

\[
R_{12} = \left( \frac{\sum_{i=n-5}^{n+5} R_i + (R_{n+6} + R_{n-6})}{2} \right) / 12
\]  

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} electrons m^{-2}.
6. Annex C – Complementary Files (CCIR and MODIP)

These files can be accessed/saved from the attachments panel of the pdf file reader. There are 12 CCIR files and 1 MODIP grid file.
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 (Subionospheric point) 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, $T_{\text{iono}}^v$ [sec], is modelled as a cosine function, while the delay along the LOS (Line of Sight) ($T_{\text{iono}}$ [sec]) is computed using a mapping function.

As an example of the behaviour 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, 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)](image-url)
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 [Annex A [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.

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

Figure 7. Global daily RMS ionospheric residual error [m/$\lambda_1$] after correction with Galileo NeQuick G (red) and GPS ICA (blue) from April 2013 to March 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)
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.

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.

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 [mL1]
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)

<table>
<thead>
<tr>
<th>$a_0$</th>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>236.831641</td>
<td>-0.39362878</td>
<td>0.0040286613</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Month (UTC)</th>
<th>Time</th>
<th>Station Longitude (deg)</th>
<th>Station Latitude (deg)</th>
<th>Station Height (m)</th>
<th>Satellite Longitude (deg)</th>
<th>Satellite Latitude (deg)</th>
<th>Satellite Height (m)</th>
<th>STEC (TECU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>8.23</td>
<td>54.29</td>
<td>20281546.18</td>
<td>20.40</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>-158.03</td>
<td>24.05</td>
<td>20275295.43</td>
<td>53.45</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>-30.86</td>
<td>41.04</td>
<td>19953770.93</td>
<td>25.91</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>-85.72</td>
<td>53.69</td>
<td>20544786.65</td>
<td>18.78</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>-130.77</td>
<td>54.40</td>
<td>20121312.46</td>
<td>20.00</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>140.68</td>
<td>35.85</td>
<td>19953735.00</td>
<td>37.81</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>-126.28</td>
<td>51.26</td>
<td>20513440.10</td>
<td>21.31</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>84.26</td>
<td>54.68</td>
<td>20305726.79</td>
<td>27.72</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>-96.21</td>
<td>37.33</td>
<td>19956072.48</td>
<td>24.13</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>175.57</td>
<td>51.89</td>
<td>19995445.72</td>
<td>26.61</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>4.25</td>
<td>53.43</td>
<td>20107681.66</td>
<td>29.21</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>14.89</td>
<td>32.88</td>
<td>20636567.33</td>
<td>37.95</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>-70.26</td>
<td>50.63</td>
<td>20043030.82</td>
<td>23.93</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>-130.60</td>
<td>49.21</td>
<td>20288021.34</td>
<td>26.97</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>-52.46</td>
<td>24.28</td>
<td>19831557.96</td>
<td>48.45</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>-165.78</td>
<td>35.06</td>
<td>20196268.24</td>
<td>41.71</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>297.66</td>
<td>82.49</td>
<td>78.11</td>
<td>168.73</td>
<td>52.58</td>
<td>20288372.95</td>
<td>27.79</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>-89.48</td>
<td>-29.05</td>
<td>20081457.33</td>
<td>240.59</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>-46.73</td>
<td>-24.08</td>
<td>19975517.42</td>
<td>152.19</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>-99.26</td>
<td>34.47</td>
<td>20257286.46</td>
<td>204.92</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>-46.61</td>
<td>54.84</td>
<td>20258938.89</td>
<td>124.39</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>-85.72</td>
<td>53.68</td>
<td>20544786.61</td>
<td>140.16</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>-18.13</td>
<td>14.17</td>
<td>20267783.18</td>
<td>95.52</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>-58.59</td>
<td>21.93</td>
<td>20008556.82</td>
<td>21.68</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>-102.83</td>
<td>-40.74</td>
<td>20153844.84</td>
<td>169.86</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>-0.60</td>
<td>10.75</td>
<td>20272829.17</td>
<td>178.43</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>-120.35</td>
<td>11.00</td>
<td>20283503.35</td>
<td>146.95</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>-70.26</td>
<td>50.63</td>
<td>20043030.72</td>
<td>198.43</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>-72.73</td>
<td>-9.78</td>
<td>19936049.27</td>
<td>149.02</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>-66.77</td>
<td>2.37</td>
<td>19986966.89</td>
<td>133.16</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>-1.57</td>
<td>-7.90</td>
<td>20373709.74</td>
<td>255.31</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>0.44</td>
<td>50.83</td>
<td>19975412.45</td>
<td>292.41</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>307.19</td>
<td>5.25</td>
<td>-25.76</td>
<td>10.94</td>
<td>44.72</td>
<td>20450566.19</td>
<td>336.74</td>
</tr>
</tbody>
</table>
## 8.2. Az Coefficients (Medium Solar Activity)

<table>
<thead>
<tr>
<th>(a_0)</th>
<th>(a_1)</th>
<th>(a_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>121.129893</td>
<td>0.351254133</td>
<td>0.0134635348</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Month</td>
<td>Time (UTC)</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
</tr>
</tbody>
</table>
### 8.3. Az Coefficients (Low Solar Activity)

<table>
<thead>
<tr>
<th>$a_0$</th>
<th>$a_1$</th>
<th>$a_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.580271</td>
<td>0.127628236</td>
<td>0.0252748384</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Month (UTC)</td>
<td>Time (deg)</td>
</tr>
<tr>
<td>-------</td>
<td>----------</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstNeQuickIn</td>
<td>NeQuick</td>
<td>1</td>
<td>Not</td>
<td>N/A</td>
<td>Input data required for NeQuick function. This includes position of ray start and end points, current time, MODIP grid and CCIR maps.</td>
</tr>
<tr>
<td>putData</td>
<td>Input</td>
<td></td>
<td>defined</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data_st</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pdSTEC</td>
<td>double</td>
<td>1</td>
<td>Not</td>
<td>Electrons/m²</td>
<td>Computed STEC value for ray from GSS to Sat</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>defined</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstCurrCCIR</td>
<td>CurrentCCIR_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Information required for computing f0F2 and M3000F2 coefficients for given time and R12 value</td>
</tr>
</tbody>
</table>

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.

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 pstCurrCCIR

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 using ModipRx
Assign Az to pstNeQuickInputData → dAzBase

Check that Az is within ranges:
if a1 = a2 = a3 =0, Az = 63.7,
if Az < 0, Az = 0
if Az > 400, Az = 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 = √stP1.dR² − stRay.dR²
stP2.dS = √stP2.dR² − stRay.dR²

If point 1 (receiver) is below the Earth’s surface, stP1.dH < 0
Set point 0 to be at Earth’s surface, stP0.dH = 0

Else
Set point 0 Flow 2 equal to point 1, set stP0.dH = stP1.dH

end if

Set radius of point 0, stP0.dR = stPO.dR + Re
(where Re = 6371.2km) stP0.dS = √stP0.dR² − stRay.dR²

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 * (mth + 30.5 - 15 + 18 - ut/24) - 3.289) * DEGTORAD
sdelta = 0.39782 * sin(amrad) + (1.916 * sin(amrad) + 0.02 * sin(2 * amrad) + 282634) * DEGTORAD

cdelta = √1 - sdelta²

(where mth = pstNeQuickInputData.siMonth and ut = pstNeQuickInputData.dUT)
If vertical ray (point 2 directly above point 1, i.e. Ray Perigee radius < 0.1)
   Call function "NeqCalcEpslParams" to calculate Ionosphere parameters
   Set bVertical flag = TRUE
End if

Update pstIntegrateData structure with stP1 and stP2 location for integration.
Call function "DoTECIntegration" to perform TEC integration along ray (dTEC)

Else
   Return an error
else
   Return an error
End if

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstNeQuickInputData</td>
<td>NeQuickInputData</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Input data required for NeQuick function. This includes position of ray start and end points, current time, MODIP grid and CCIR maps.</td>
</tr>
</tbody>
</table>

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:

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

Table 7. DoTECIntegration Function Input Data

Outputs:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pdTEC</td>
<td>Double</td>
<td>1</td>
<td>Not defined</td>
<td>Electrons/km/m³</td>
<td>Total electron content</td>
</tr>
</tbody>
</table>

Table 8. NeQuick Function Output Data

Internal Processing

Set integration tolerance in case Kronrod $G_7-K_{15}$ 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).

\[
\text{If } 1 \ bVert = \text{TRUE} \\
\quad \text{HeightP0} = \text{stP0.dH} \\
\quad \text{HeightP1} = \text{stP1.dH} \\
\quad \text{HeightP2} = \text{stP2.dH} \\
\text{Else}_1 \\
\quad \text{HeightP0} = \text{stP0.dS} \\
\quad \text{HeightP1} = \text{stP1.dS} \\
\quad \text{HeightP2} = \text{stP2.dS} \\
\text{End if}_1 \\
\]

(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)

\[
\text{If } 1 \ \text{stP2.dH} \leq 1000 \text{km} \ (\text{i.e. start and end point both below 1}^{\text{st}} \text{ break point)} \\
\quad \text{Call NeqIntegrate with start point of HeightP0, end point of HeightP2.} \\
\text{Else}_1 \\
\quad \text{Set slant distance of 1}^{\text{st}} \text{integration breakpoint S1a at 1000km.} \\
\text{If } 2 \ \text{stP2.dH} \leq 2000 \text{km} \ (\text{end point below 2}^{\text{nd}} \text{ break point)} \\
\quad \text{If } 3 \ \text{stP1.dH} \geq 1000 \text{km} \ (\text{start point above 1}^{\text{st}} \text{ break point)} \\
\quad \text{Start and end points are both between 1}^{\text{st}} \text{ and 2}^{\text{nd}} \text{ break points.} \\
\quad \text{Call NeqIntegrate with start point of HeightP1, end point of HeightP2.} \\
\text{Else}_3 \\
\quad \text{Ray path crosses 1}^{\text{st}} \text{ integration break point.} \\
\]
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

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 grid should be ‘pre-wrapped’ at edges and poles.

Interfaces

Called by: NeQuick, NeqCalcEpstParams

Calls: NeqInterpolate

Inputs:

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

Table 9. NeqCalcModip Function Input Data

Return value: dReturn – calculated MODIP value at given location

Internal Processing

Check Latitude is within ±90 degrees

if_1 dLat ≤ -90
    Set computed Modip value dReturn = -90
else if_1 dLat ≥ 90
    Set computed Modip value dReturn = 90
Define properties of MODIP grid (lat step = 5 deg, lon step = 10 deg)

\[ \text{constants describing grid } \text{lngp} = 36, \text{dlatp} = 5, \text{dlngp} = 10 \]

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

\[ \text{lng1} = (\text{dLng} + 180) / \text{dlngp} \]
\[ sj = \text{floor}(\text{lng1}) - 2 \]
\[ dj = \text{lng1} - \text{floor}(\text{lng1}) \]

Adjust for sign and wrap to grid if required

\[ \text{if } (sj < 0) \]
\[ sj = sj + \text{lngp} \]
\[ \text{end if} \]

\[ \text{if } (sj > (\text{lngp} - 3)) \]
\[ sj = sj - \text{lngp}; \]
\[ \text{end if} \]

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

\[ \text{lat1} = (\text{dLat} + 90) / \text{dlatp} + 1 \]
\[ si = \text{floor}(\text{lat1} - 1e-6) - 2 \]
\[ di = \text{lat1} - si - 2 \]

Interpolate across lat grid to obtain values at 4 lon points on lat line

\[ \text{for } k = 1; k <= 4; ++k \]
\[ \text{for } j = 1; j <= 4; ++j \]
\[ z[1][j - 1] = \text{pstModip}[si][sj + k + 1] \]
\[ \text{end for}_j \]
\[ z[k - 1] = \text{NeqInterpolate}(z[1][di]) \]
\[ \text{end for}_k \]

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

\[ \text{End if} \]

9.2.4.2. \textit{NeQuick Internal Function “NeqInterpolate”}

\textbf{Purpose}

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

\textbf{Interfaces}

Called by: NeqCalcModip

Calls: none

\textbf{Inputs:}

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
\textbf{Name} & \textbf{Type} & \textbf{Size} & \textbf{Range} & \textbf{Units} & \textbf{Description} \\
\hline
pdZ & double & [Annex A [4]] & Not defined & N/A & Array containing the -1, 0, 1 and 2 point values \\
\hline
dDeltaX & double & 1 & Not defined & N/A & Position to interpolate to (offset from 0pt to 1pt) \\
\hline
pstModip & MODIP_st & 1 & Not defined & N/A & Structure containing grid of modified dip latitude values. \\
\hline
\end{tabular}
\caption{NeqInterpolate Function Input Data}
\end{table}

Return value: \(dIntZ\) – the interpolated value at the point

\textbf{Internal Processing}
if \(d\Delta X\) is small, return = zero point, \(pdZ[1]_i\)

else

Interpolate

\[
\begin{align*}
g_1 &= pdZ[2] - pdZ[1] \\
g_2 &= pdZ[3] + pdZ[0] \\
g_3 &= (pdZ[3] - pdZ[0]) / 3 \\
\end{align*}
\]

\[
\begin{align*}
a_0 &= 9g_0 - g_2 \\
a_1 &= 9g_1 - g_3 \\
a_2 &= g_2 - g_0 \\
a_3 &= g_3 - g_1 \\
\end{align*}
\]

\[
\Delta x = 2d\Delta X - 1
\]

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

end if

9.2.4.3. **NeQuick Internal Function “NeqModipToAz”**

**Purpose**

This function calculates Az from the provided coefficients and modified dip latitude value (\(\mu\)). 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:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dModip</td>
<td>double</td>
<td>1</td>
<td>-90 to 90 deg</td>
<td></td>
<td>Modified dip latitude of point</td>
</tr>
<tr>
<td>siNumCoeff</td>
<td>double</td>
<td>1</td>
<td>≥1</td>
<td>N/A</td>
<td>Number of Az coefficients</td>
</tr>
<tr>
<td>pdCoeff</td>
<td>double</td>
<td>[numCoeffs]</td>
<td>&gt;0</td>
<td>flux units/deg^-1</td>
<td>Az coefficients</td>
</tr>
</tbody>
</table>

Table 11. NeqCalcModip Function Input Data

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

**Internal Processing**

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

where:

\(i = siNumCoeff\)

\(a_i = pdCoeff\)

\(\mu = dModip\)

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:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstP1</td>
<td>SPoint_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Positional information for Point 1 (receiver)</td>
</tr>
<tr>
<td>pstP2</td>
<td>SPoint_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Positional information for Point 2 (satellite)</td>
</tr>
</tbody>
</table>

Table 12. NeqGetRayProperties Function Input/Output Data

**Outputs:**

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

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 $|\{p_{P2}\rightarrow dL_{at} - p_{P1}\rightarrow dL_{at}\}| < 10^{-5}$ and $|p_{P2}\rightarrow dL_{ng} - p_{P1}\rightarrow dL_{ng}| < 10^{-5}$

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

\[
p_{P2}\rightarrow dL_{ng} = p_{P1}\rightarrow dL_{ng}
\]

end if

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

\[
p_{P1}\rightarrow dR = p_{P1}\rightarrow dH + Re
\]

\[
p_{P2}\rightarrow dR = p_{P2}\rightarrow dH + Re
\]

\[
(Re = 6371.2km)
\]

call function “NeqCalcRayProperties1” to calculate properties of the ray itself.

if invalid ray, i.e. $|pdZeta| > 90.0$ and $p_{stRay}\rightarrow dR < Re$

Set return value bError to TRUE

end if

\[
\text{if } 1 \text{ ray is not vertical, } p_{stRay}\rightarrow dR \geq 0.1
\]

Call function “NeqCalcRayProperties2” to calculate additional ray properties.

end if

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:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstP1</td>
<td>SPoint_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Positional information for Point 1 (receiver)</td>
</tr>
<tr>
<td>pstP2</td>
<td>SPoint_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Positional information for Point 2 (satellite)</td>
</tr>
</tbody>
</table>

Table 14. NeqCalcRayProperties1 Function Input/Output Data

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

Table 15. NeqCalcRayProperties1 Function Output Data

Return value: -

**Internal Processing**

Check if ray is vertical

\[ \text{if} \ p2 \text{ is directly above point 1,} |\text{pstP2→dLat} - \text{pstP1→dLat}| < 10^{-5} \text{ and } |\text{pstP2→dLng} - \text{pstP1→dLng}| < 10^{-5} \]

Set the ray latitude and longitude to be the same as point 1, pstRay→dLat = pstP1→dLat and pstRay→dLng = pstP1→dLng

Set the ray to have no slant, pstRay→dR = 0 and pdZeta = 0

\[ \text{else}_1 \]

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

\[ \text{pstP1→dSinLat, pstP1→dCosLat, pstP2→dSinLat, pstP2→dCosLat} \]

Calculate temporary variables: CosDI12, SinDI12, CosDel, SinDel

\[ \text{CosDI12} = \cos((\text{pstP2→dLng} - \text{pstP1→dLng}) \times \text{DEGTORAD}) \]
\[ \text{SinDI12} = \sin((\text{pstP2→dLng} - \text{pstP1→dLng}) \times \text{DEGTORAD}) \]
\[ \text{CosDel} = \text{pstP1→dSinLat} \times \text{pstP2→dSinLat} + \text{pstP1→dCosLat} \times \text{pstP2→dCosLat} \times \text{CosDI12} \]
\[ \text{SinDel} = \sqrt{1 - \text{CosDel}^2} \]

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

\[ \text{pdZeta} = \frac{\text{atan}_1\left(\frac{\text{SinDel, CosDel} - \text{pstP1→dR}}{\text{pstP2→dR}}\right)}{\text{pstP2→dR}} \]

Calculate temporary variables, sdelp, cdelp, sphp, cphp

\[ \text{SinSigg} = \frac{\text{SinDI12} \times \text{pstP2→dCosLat}}{\text{SinDel}} \]
\[ \text{CosSigg} = \frac{\text{pstP2→dSinLat} - \text{CosDel} \times \text{pstP1→dSinLat}}{\text{SinDel} \times \text{pstP1→dCosLat}} \]
\[ Delp = \frac{\pi}{2} - \text{pdZeta} \]
$$\sin(Delp) = \sin(Delp)$$
$$\cos(Delp) = \cos(Delp)$$

$$\sin(Php) = \frac{\text{PstP1} \rightarrow d\sin(\text{Lat}) \cdot \cos(Delp) - \text{PstP1} \rightarrow d\cos(\text{Lat}) \cdot \sin(Delp) \cdot \cos(Sig)p}{\sqrt{1 - \sin(Php)^2}}$$

Calculate ray perigee latitude
$$\text{PstRay} \rightarrow d\text{Lat} = \arctan(\sin(Php,\cos(Php)) \cdot \text{RADTODEG})$$
(NB this returns identical result to asin(sin(Php))

Calculate ray perigee longitude
$$\sin(Lamp) = \frac{-1 \cdot \sin(Sig)p \cdot \sin(Delp)}{\cos Php}$$
$$\cos(Lamp) = \frac{\cos(Delp) - \text{PstP1} \rightarrow d\sin(\text{Lat}) \cdot \sin Php}{\text{PstP1} \rightarrow d\cos(\text{Lat}) \cdot \cos Php}$$
$$\text{PstRay} \rightarrow d\text{Lng} = \arctan(\sin(Lamp,\cos(Lamp)) \cdot \text{RADTODEG})$$

Calculate radius of ray perigee
$$\text{PstRay} \rightarrow dR = \text{PstP1} \rightarrow dR \cdot \sin(pdZeta)$$

Convert zeta to degrees, $pdZeta = pdZeta \cdot RD$

\textbf{9.2.5.3. NeQuick Internal Function “NeqCalcRayProperties2”}

\textbf{Purpose}

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

\textbf{Interfaces}

Called by: NeqGetRayProperties

Calls: none

\textbf{Inputs:}

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstRay</td>
<td>SPoint_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Positional information for ray perigee</td>
</tr>
</tbody>
</table>

Table 16. NeqCalcRayProperties2 Function Input Data

\textbf{Inputs/Outputs:}

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstP2</td>
<td>SPoint_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Positional information for Point 2 (satellite)</td>
</tr>
</tbody>
</table>

Table 17. NeqCalcRayProperties2 Function Input/Output Data

\textbf{Outputs:}

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstP1</td>
<td>SPoint_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Positional information for Point 1 (receiver)</td>
</tr>
<tr>
<td>pdSinSig</td>
<td>double</td>
<td>1</td>
<td>-1 to 1</td>
<td>N/A</td>
<td>Sine of ray azimuth</td>
</tr>
<tr>
<td>pdCosSig</td>
<td>double</td>
<td>1</td>
<td>-1 to 1</td>
<td>N/A</td>
<td>Cosine of ray azimuth</td>
</tr>
</tbody>
</table>

Table 18. NeqCalcRayProperties2 Function Output Data

\textbf{Return value: -}

\textbf{Internal Processing}

Calculate sine and cosine of end point latitudes (using ray perigee latitude for point 1)
\[
pstP1\rightarrow d\sinLat = \sin(pstRay\rightarrow d\Lat \cdot \text{DEGTORAD})
\]
\[
pstP1\rightarrow d\cosLat = \cos(pstRay\rightarrow d\Lat \cdot \text{DEGTORAD})
\]
\[
pstP2\rightarrow d\sinLat = \sin(pstP2\rightarrow d\Lat \cdot \text{DEGTORAD})
\]
\[
pstP2\rightarrow d\cosLat = \cos(pstP2\rightarrow d\Lat \cdot \text{DEGTORAD})
\]

Calculate difference in longitude of ray end points

\[
\text{DeltaLong} = (pstP2\rightarrow d\Lng - pstRay\rightarrow d\Lng) \cdot \text{DEGTORAD}
\]

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

\[
\text{if}_1 \quad ||pstRay\rightarrow d\Lat - 90|| < 10^{-10}
\]

Set sine of azimuth, \(pd\sinSig = 0\)

\[
\text{if}_2 \quad \text{positive latitude, } pstRay\rightarrow d\Lat > 0
\]

Set cosine of azimuth, \(pd\cosSig = -1\)

\[
\text{else}_2 \quad \text{Set cosine of azimuth, } pd\cosSig = 1
\]

\[
\text{end if}_2
\]

\[
\text{else}_1
\]

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

\[
\text{CosPsi} = pstP1\rightarrow d\sinLat \cdot pstP2\rightarrow d\sinLat + pstP1\rightarrow d\cosLat \cdot pstP2\rightarrow d\cosLat \cdot \cos(\text{DeltaLong})
\]

\[
\text{SinPsi} = \sqrt{1 - \text{CosPsi}^2}
\]

Calculate sine and cosine of azimuth

\[
pd\sinSig = \frac{pstP2\rightarrow d\cosLat \cdot \sin(\text{DeltaLong})}{\text{SinPsi}}
\]

\[
pd\cosSig = \frac{pstP2\rightarrow d\sinLat - pstP1\rightarrow d\sinLat \cdot \text{CosPsi}}{\text{SinPsi} \cdot pstP1\rightarrow d\cosLat}
\]

\[
\text{end if}_1
\]

9.2.6. NeqIntegrate.c Module

9.2.6.1. NeQuick Internal Function “NeqIntegrate”

Purpose

Integration function for calculating TEC along rays using Kronrod \(G_7-K_{15}\) adaptive 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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstIntegrateData</td>
<td>IntegrateData_{st}</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Data required for computation of integrated TEC value</td>
</tr>
<tr>
<td>dH1</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>km</td>
<td>Height of point 1</td>
</tr>
<tr>
<td>dH2</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>km</td>
<td>Height of point 2</td>
</tr>
<tr>
<td>siCurrentLevel</td>
<td>Integer</td>
<td>1</td>
<td>≥0</td>
<td>N/A</td>
<td>Current level of integration recursion</td>
</tr>
</tbody>
</table>

Table 19. NeqIntegrate Function Input Data
Outputs:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstPactual</td>
<td>SPoint_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Positional information for current integration point.</td>
</tr>
</tbody>
</table>

Table 20. NeqIntegrate Function Output Data

Inputs/Outputs:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pdNmax</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>electrons/m³</td>
<td>Maximum Ne (F2 peak)</td>
</tr>
<tr>
<td>pstLayers</td>
<td>LayerProperties_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Current Ionospheric properties</td>
</tr>
</tbody>
</table>

Table 21. NeqIntegrate Function Input/Output Data

Return value: dReturn – calculated TEC value

Internal Processing

Set Kronrod integration coefficients (G-[K]15)

Set weights for K15 sample points

\[w[j15] = \{0.022935322010529224963732008058970, 0.063092092629978553290700663189204, 0.10479001032225018389876322541518, 0.140653259715525918745189590510238, 0.169004726639267902826583426598550, 0.190350578064785409913256402421014, 0.2043294007529892414161999234649, 0.209482141084727828012999174891714, 0.2043294007529892414161999234649, 0.190350578064785409913256402421014, 0.169004726639267902826583426598550, 0.140653259715525918745189590510238, 0.10479001032225018389876322541518, 0.063092092629978553290700663189204, 0.022935322010529224963732008058970\} \]

Set weights for G7 sample points

\[w[i7] = \{0.129484966168869693270611432679082, 0.2797053591489276667901467771423780, 0.38183005050118944950369775488975, 0.417959183673469387755102040816327, 0.38183005050118944950369775488975, 0.2797053591489276667901467771423780, 0.129484966168869693270611432679082\} \]

Set at what points the samples are used in integration process

\[x[i15] = \{-0.9914553711208120812639026854697526329, -0.949107912342758524526189684047851, -0.864864423559769072789712788640926, -0.741531185599394439863864773280788, -0.5860371235467691130294144838258730, -0.40584515137739176906606412076961, -0.207784955007898467600689403773245, 0, 0.207784955007898467600689403773245, 0.40584515137739176906606412076961, 0.5860371235467691130294144838258730, 0.741531185599394439863864773280788\} \]
Calculate the midpoint, \( hh \) and the half difference, \( h2 \)
\[
h2 = \frac{dH2-dH1}{2} \\
hh = \frac{dH2+dH1}{2}
\]

Initialise integration results \( intk \) and \( intg \), and G7 counter \( Gind \)

Loop through the G15 and K7 integration points

\[
\text{for } i = 0; i < 15; ++i \\
\text{ } x = h2 \times x[i] + hh \\
\text{if } \text{ray is vertical} \\
\quad y = \text{NeqGetNeOnVerticalRay at } x \\
\text{else} \\
\quad y = \text{NeqGetNeOnSlantRay at } x \\
\text{end if}
\]

Accumulate on to the k15 total
\[
\text{intk} = \text{intk} + y \times w[i]
\]
\[
\text{if this is a G7 point (every other point – i.e. modulus of } i/2=1) \\
\quad \text{intg} = \text{intg} + y \times w[\text{Gind}] \\
\quad \text{Gind} = \text{Gind} + 1
\]
\[
\text{end if}
\]

\[
\text{end for}
\]

Complete the calculation of the integration results
\[
\text{Intk} = \text{intk} \times h2 \\
\text{Intg} = \text{intg} \times h2
\]

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

### 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:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dH</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>km</td>
<td>Height of point</td>
</tr>
<tr>
<td>pstLayers</td>
<td>LayerProperties_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Current Ionospheric properties</td>
</tr>
</tbody>
</table>

Table 22. NeqGetNeOnVertRay Function Input Data
Inputs/Outputs:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pdNmax</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>electrons/m$^3$</td>
<td>Maximum Ne (F2 peak)</td>
</tr>
</tbody>
</table>

Table 23. NeqGetNeOnVertRay Function Input/Output Data

Return value: dReturn – Calculated electron content at specified height

Internal Processing

if specified height is above F2 peak, dH > PeakHeight[F2]
Call function “NeqCalcTopsideNe” to compute dReturn
else
Call function “NeqCalcBottomsideNe” to compute dReturn
end if
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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dS</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>km</td>
<td>Distance of the point along the ray</td>
</tr>
<tr>
<td>pstNeQuickInputData</td>
<td>NeQuickIn putData_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Input data required for NeQuick Function</td>
</tr>
<tr>
<td>pstGeom</td>
<td>Geometry Data_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Geometry data for ray</td>
</tr>
</tbody>
</table>

Table 24. NeqGetNeOnSlantRay Function Input Data

Outputs:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pdNmax</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>electrons/m$^3$</td>
<td>Maximum Ne (F2 peak)</td>
</tr>
<tr>
<td>pstPactual</td>
<td>SPoint_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Positional information for current integration point.</td>
</tr>
<tr>
<td>pstLayers</td>
<td>LayerProp erties_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Current Ionospheric properties</td>
</tr>
</tbody>
</table>

Table 25. NeqGetNeOnSlantRay Function Output Data

Inputs/Outputs:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstCurrCCIR</td>
<td>Current CCIR_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Information required for computing F0F2 and M3000F2 coefficients for given time and R12 value</td>
</tr>
</tbody>
</table>

Table 26. NeqGetNeOnSlantRay Function Input/Output Data
Return value: \( d\text{Return} \) – electron content value at specified point

**Internal Processing**

Call function “\text{NeqCalcLLHOnRay}” to adjust position information for current position along ray, \( \text{pstPactual} \).

all function “\text{NeqCalcEpstParams}” to recalculate ionosphere information now that the latitude and longitude have changed.

\[
\text{if} \quad \text{current height is above F2 peak height, } \text{pstPactual} \rightarrow d\text{H} > \text{PeakHeight}[F2]
\]

Call function “\text{NeqCalcTopsideNe}” to compute \( d\text{Return} \)

\[
\text{else}
\]

Call function “\text{NeqCalcBottomsideNe}” to compute \( d\text{Return} \)

\[
\text{end if}
\]

Return \( d\text{Return} \)

9.2.8.2. **NeQuick Internal Function “\text{NeqCalcLLHOnRay}”**

**Purpose**

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

**Interfaces**

Called by: NeqGetNeOnSlantRay

Calls: none.

**Inputs:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( dS )</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>km</td>
<td>Distance of the point along the ray</td>
</tr>
<tr>
<td>( \text{pstRay} )</td>
<td>SPoint_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Information for ray perigee</td>
</tr>
<tr>
<td>( \text{pstP1} )</td>
<td>SPoint_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Information for Point 1</td>
</tr>
<tr>
<td>( d\text{SinSig} )</td>
<td>double</td>
<td>1</td>
<td>-1 to 1</td>
<td>N/A</td>
<td>Sine of ray azimuth</td>
</tr>
<tr>
<td>( d\text{CosSig} )</td>
<td>double</td>
<td>1</td>
<td>-1 to 1</td>
<td>N/A</td>
<td>Cosine of ray azimuth</td>
</tr>
</tbody>
</table>

Table 27. NeqCalcLLHOnRay Function Input Data

**Outputs:**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{pstPactual} )</td>
<td>SPoint_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Positional information for current integration point.</td>
</tr>
</tbody>
</table>

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 (\( \text{Del} \)):

\[
Tan\text{Del} = \frac{dS}{\text{pstRay} \rightarrow dR}
\]

\[
Cos\text{Del} = \frac{1}{\sqrt{1 + Tan\text{Del}^2}}
\]

\[
Sin\text{Del} = Tan\text{Del} \times Cos\text{Del}
\]
Calculate latitude

\[
\text{arg} = \text{pst}P_1 \rightarrow \text{dSinLat} \cdot \text{CosDel} + \text{pst}P_1 \rightarrow \text{dCosLat} \cdot \text{SinDel} \cdot \text{dCosSig} \\
\text{pstPactual} \rightarrow \text{dLat} = \text{atan2}\left(\text{arg}, \sqrt{1 - \text{arg}^2}\right) \cdot \text{RD}
\]

(NB \text{asin}(\text{arg}) \text{ would give same result})

Calculate longitude

\[
\text{CLong} = \text{atan2}(\text{SinDel} \cdot \text{dSinSig} \cdot \text{pst}P_1 \rightarrow \text{dCosLat}, \text{CosDel} - \text{pst}P_1 \rightarrow \text{dSinLat} \cdot \text{arg}) \cdot \text{RD} \\
\text{pstPactual} \rightarrow \text{dLng} = \text{CLong} + \text{pstRay} \rightarrow \text{dLng}
\]

Calculate height

\[
\text{pstPactual} \rightarrow \text{dH} = \sqrt{\text{dS}^2 + \text{pstRay} \rightarrow \text{dR}^2 - \text{Re}}
\]

(\text{with Re}=6371.2)

9.2.9. \textbf{NeqCalcEpstParams.c Module}

9.2.9.1. \textit{NeQuick Internal Function “NeqCalcEpstParams”}

\textbf{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]

\textbf{Interfaces}

Called by: \textit{NeQuick} and \textit{NeqGetNeOnSlantRay}.

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

\textbf{Inputs:}

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstNeQuickInPutData</td>
<td>NeQuickInPutData_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Input data required for \textit{NeQuick} Function</td>
</tr>
<tr>
<td>pstPactual</td>
<td>SPoint_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Positional information for current integration point</td>
</tr>
<tr>
<td>dSinDelta</td>
<td>double</td>
<td>1</td>
<td>-1 to 1</td>
<td>N/A</td>
<td>Sine of angle of declination of sun</td>
</tr>
<tr>
<td>dCosDelta</td>
<td>double</td>
<td>1</td>
<td>-1 to 1</td>
<td>N/A</td>
<td>Cosine of angle of declination of sun</td>
</tr>
</tbody>
</table>

Table 29. NeqCalcEpstParams Function Input Data
Outputs:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pdNmax</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>electrons/m³</td>
<td>Maximum Ne (F2 peak)</td>
</tr>
<tr>
<td>pstLayers</td>
<td>LayerProperties_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Current Ionospheric properties</td>
</tr>
</tbody>
</table>

Table 30. NeqCalcEpstParams Function Output Data

Inputs/Outputs:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstCurrCCIR</td>
<td>CurrentCCIR_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Information required for computing F0F2 and M3000F2 coefficients for given time and R12 value.</td>
</tr>
</tbody>
</table>

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

\[ \text{if} \quad \text{month or R12 have changed} \]

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)

\[ \text{for} \quad i = 1:13 \]
\[ \text{for} \quad j = 1:76 \]
\[ \text{CurrCCIR.pdFOF2((j-1) \cdot 13 + i) = CCIR.pdFOF2(siMonth,i,j,1) \cdot RR1 + CCIR.pdFOF2(siMonth,i,j,2) \cdot RR2;} \]
\[ \text{end for} \]
\[ \text{end for} \]
\[ \text{for} \quad i = 1:9 \]
\[ \text{for} \quad j = 1:49 \]
\[ \text{CurrCCIR.pdM3000F2((j-1) \cdot 9 + i) = CCIR.pdM3000(siMonth,i,j,1) \cdot RR1 + CCIR.pdM3000(siMonth,i,j,2) \cdot RR2;} \]
\[ \text{end for} \]
\[ \text{end for} \]

Set current R12 and Month

Call NeqCalcSphLegCoeffs with current time and blended CCIR information

else \[ \text{if} \quad \text{time has changed} \]

Call NeqCalcSphLegCoeffs with current time and blended CCIR information

end \[ \text{if} \]

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

Get F2 layer data from CCIR

For fOF2, 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
\[ x_{lt} = u + \text{pstPactual} \rightarrow \text{dLg} / 15.0 \]

\[
\text{if1} \ (x_{lt} < 0)
\]
\[
x_{lt} = x_{lt} + 24.0
\]

\[
\text{else if1} \ (x_{lt} \geq 24)
\]
\[
x_{lt} = x_{lt} - 24.0
\]

\[
\text{end if1}
\]

Calculate Solar Zenith angle

\[
\text{CosChi} = \sin(\text{pstPactual} \rightarrow \text{dLat} + \text{DegreesToRadians}) \cdot \text{dSinDelta} + \cos(\text{pstPactual} \rightarrow \text{dLat} + \text{DegreesToRadians}) \cdot \text{dCosDelta} + \cos(\pi \cdot (12 - x_{lt}) / 12)
\]

\[
\text{chi} = \text{atan2}(\text{sqrt}(1 - \text{CosChi} \cdot \text{CosChi}), \text{CosChi}) \cdot \text{RadiansToDegrees}
\]

\[
\text{chi0} = 86.23
\]

Set season flag (seas):

- Jan–Feb: −1
- Mar–Apr: 0
- May–Aug: 1
- Sep–Oct: 0
- Nov–Dec: −1

Estimate \( f_{0E} \) and \( f_{0F1} \)

- The model for \( f_{0E} \) 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 \( f_{0F1} = 1.4 \cdot f_{0E} \), for night-time \( f_{0F1} = 0 \), using the same exponential day-night transition as for \( f_{0E} \). In addition \( f_{0F1} \) is reduced by 15% if too close to \( f_{0F2} \).

\[
\text{ee} = \text{NeqClipExp}(0.3 \cdot \text{lat})
\]

\[
\text{seas} = \text{seas} \cdot (\text{ee} - 1) / (\text{ee} + 1)
\]

\[
\text{chin} = \text{NeqJoin}(90.0 - 0.24 \cdot \text{NeqClipExp}(20.0 - 0.2 \cdot \text{chi}, \text{chi}, 12, \text{chi} - \text{chi0})
\]

\[
\text{sfac} = (1.112 - 0.019 \cdot \text{seas}) \cdot \text{sqrt}(	ext{sqrt}(\text{Az}))
\]

\[
\text{fa} = \text{sfac} \cdot \text{NeqClipExp}(\log(\cos(\text{chin} \cdot \text{DR}))) \cdot 0.3
\]

Calculate \( E \) peak plasma frequency

\[
\text{fOE} = \text{sqrt}(\text{fa} \cdot \text{fa} + 0.49)
\]

Calculate \( F1 \) peak plasma frequency and set to zero if negligible

\[
\text{fOF1} = \text{NeqJoin}(1.4 \cdot f_{0E}, 0, 1000.0, f_{0E}, -2) \quad \text{(Titheridge's Formula } f_{0F1} = 1.4 \cdot f_{0F2})
\]

\[
\text{fOF1} = \text{NeqJoin}(0, f_{0F1}, 1000.0, f_{0E} - f_{0F1})
\]

\[
\text{fOF1} = \text{NeqJoin}(f_{0F1}, 0.85 \cdot f_{0F1}, 60.0, 0.85 \cdot f_{0F2} - f_{0F1})
\]

\[
\text{if1} \ (f_{0F1} < 1e^{-6})
\]

\[
f_{0F1} = 0
\]

\[
\text{end if1}
\]

Calculate peak electron densities from critical frequencies for the \( F2, F1 \) and \( E \) layers.

\[
\text{Nm}[F2] = \text{NeqCriticalFreqToNe}[f_{0F2}]
\]

\[
\text{Nm}[F1] = \text{NeqCriticalFreqToNe}[f_{0F1}]
\]

\[
\text{Nm}[E] = \text{NeqCriticalFreqToNe}[f_{0E}]
\]

(\( F2, F1 \) 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.

\[
\text{PeakHeight}[F2] = \text{peakH}()
\]

\[
\text{PeakHeight}[E] = 120km
\]

\[
\text{PeakHeight}[F1] = (\text{PeakHeight}[F2] + \text{PeakHeight}[E]) / 2
\]

Calculate density gradient at base of \( F2 \) layer \((10^9 \text{ m}^{-3} \text{ km}^{-1})\) (see [Annex A [12]])

\[
\text{NdHmx} = 0.01 \cdot \exp(-3.467 + 0.857 \cdot \log(f_{0F2} \cdot f_{0F2}) + 2.02 \cdot \log(M3000))
\]

Calculate Bottom–side thickness parameters (see [Annex A [13]])

\[
\text{BotThick}[F2] = 0.385 \cdot \text{Nm}[F2] / \text{NdHmx}
\]

\[
\text{TopThick}[F1] = 0.3 + (\text{PeakHeight}[F2] - \text{PeakHeight}[F1])
\]

\[
\text{BotThick}[F1] = 0.5 + (\text{PeakHeight}[F1] - \text{PeakHeight}[E])
\]

\[
\text{TopThick}[E] = \text{BotThick}[F1]
\]
if \( \text{TopThick}[E] < 7 \)
\[
\text{TopThick}[E] = 7
\]
end if

\( \text{BotThick}[E] = 5 \)

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: \( \text{foF2}, \text{M}(3000)\text{F2}, \text{foF1} \) and \( \text{foE} \). The basic equations of the model are:

\[
\begin{align*}
\text{N}(h) &= \text{NF2}(h) + \text{NF1}(h) + \text{NE}(h) \\
\text{N}(h) &= 4\text{Nm} F2 \left( 1 + \exp \left( \frac{h - \text{hmF2}}{B2} \right) \right)^2 \\
&\quad + 4\text{Nm} F1 \left( 1 + \exp \left( \frac{h - \text{hmF1}}{B1} \right) \right)^2 \\
&\quad + 4\text{Nm} E \left( 1 + \exp \left( \frac{h - \text{hmE}}{B_e} \right) \right)^2
\end{align*}
\]

Where:

\[
\begin{align*}
\text{Nm} F2 &= \text{NmF2} - 0.1\text{NmF1} \\
\text{Nm} F1 &= \text{NmF1} - \text{NF2}(\text{hmF1}) \\
\text{Nm} E &= \text{NmE} - \text{NF1}(\text{hmE} = 120\text{km}) - \text{NF2}(\text{hmE} = 120\text{km})
\end{align*}
\]

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

The algorithm provides continuity to the function \( \text{N}(h) \) taking into account the exponential nature of the equations describing the model, using as an auxiliary function \( \text{FEpst} \) defined by:

\[
\begin{align*}
\text{FEpst}(X,Y,Z,W) &= X \times \exp((W-Y)/Z)/(1+\exp((W-Y)/Z))^2 \\
\text{Amp}[F2] &= 4 \times \text{Nm}[F2] \\
\text{Amp}[F1] &= 4 \times \text{Nm}[F1] \\
\text{Amp}[E] &= 4 \times \text{Nm}[E] \\
\text{if} \ (\text{foF1} < 0.5) \\
\text{Amp}[F1] &= 0 \\
\text{Amp}[E] &= 4 \times \text{(Nm}[E)]
\end{align*}
\]

- \( \text{NeqEpstein} = \text{Amp}[F2], \text{PeakHeight}[F2], \text{BotThick}[F2], \text{PeakHeight}[E]) \)

else

\[ (i = 0 / 5,++) \]

\[
\begin{align*}
\text{Amp}[F1] &= 4 \times \text{(Nm}[F1)] \\
\text{NeqEpstein} &= \text{Amp}[F2], \text{PeakHeight}[F2], \text{BotThick}[F2], \text{PeakHeight}[E]) \\
\text{NeqEpstein} &= \text{Amp}[E], \text{PeakHeight}[E], \text{BotThick}[E], \text{PeakHeight}[E]) \\
\text{NeqEpstein} &= \text{Amp}[F2], \text{PeakHeight}[F2], \text{BotThick}[F2], \text{PeakHeight}[E])
\end{align*}
\]

end if

\[
\text{Amp}[E] = \text{NeqJoin}(\text{Amp}[E], 0.05, 60.0, \text{Amp}[E] - 0.005)
\]

Calculate shape factor for topside F2 region (see [Annex A [14]])

\[ (\text{mth} > 3 \&\& \text{mth} < 10) \text{, April to September} \]

\[
\begin{align*}
\text{b2k} &= 6.705 - 0.014 \times \text{R12} - 0.008 \times \text{PeakHeight}[F2] \\
\text{else} \text{, October to May} \\
\text{b2k} &= -7.77 + 0.097 \times \text{pow}([\text{PeakHeight}[F2] / \text{BotThick}[F2], 2] + 0.153 \times \text{Nm}[F2])
\end{align*}
\]

end if

\[
\begin{align*}
\text{b2k} &= \text{NeqJoin}(\text{b2k}, 2.1, \text{b2k} - 2.0) \\
\text{b2k} &= \text{NeqJoin}(\text{b2k}, 1.2, \text{b2k} - 0.0)
\end{align*}
\]

Adjust the vertical TEC value to take into account exosphere electron density ([Annex A [15]])

\[
\begin{align*}
\text{TopThick}[F2] &= \text{b2k} \times \text{BotThick}[F2] \\
\text{x} &= (\text{TopThick}[F2] - 150.0) / 100.0 \\
\text{v} &= (0.041163 \times x - 0.183981) \times x + 1.424472 \\
\text{TopThick}[F2] &= \text{TopThick}[F2] / v
\end{align*}
\]

9.2.9.2. NeQuick Internal Function “NeqCalcSphLegCoeffs”
Purpose

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

Interfaces

Called by: NeqCalcEpstParams

Calls: none

Inputs:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dUT</td>
<td>double</td>
<td>1</td>
<td>[0,24)</td>
<td>hours</td>
<td>Time (UTC) at which STEC value is required</td>
</tr>
</tbody>
</table>

Table 32. NeqCalcSphLegCoeffs Function Input Data

Inputs/Outputs:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstCurrCCIR</td>
<td>CurrentCCIR</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Information required for computing ( \text{F0F2} ) and ( \text{M3000F2} ) coefficients for given time and R12 value.</td>
</tr>
</tbody>
</table>

Table 33. NeqCalcSphLegCoeffs Function Input/Output Data

Return value: -

Internal Processing

### Compute the longitude of the sun \( t \) at the current time

\[
t = (dUT + 15° - 180) \times DR
\]

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 \( \cos(A+B) = \cos(A)\cos(B) - \sin(A)\sin(B) \)

\[
sinHarm_1 = sin(t)
\]
\[
cosHarm_1 = cos(t)
\]

for each additional harmonic, \( i = 2, i \leq \text{numMaxHarm} \) (where \( \text{numMaxHarm} = 6 \))

\[
sinHarm_i = sinHarm_{i-1} \times cosHarm_1 + cosHarm_{i-1} \times sinHarm_1
\]
\[
cosHarm_i = cosHarm_{i-1} \times cosHarm_1 - sinHarm_{i-1} \times sinHarm_1
\]

end for

Calculate coefficients for spherical Legendre function using following equation:

\[
c_i = pstCurrCCIR(i-1)N+1 + \sum_{k=1}^{numHarm} (pstCurrCCIR(i-1)N+2ksinHarm_k + pstCurrCCIR(i-1)N+2k+1cosHarm_k)
\]

Where:

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

9.2.9.3. **NeQuick Internal Function “NeqGetF2FreqFromCCIR”**

Purpose

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

Interfaces

Called by: NeqCalcEpstParams

Calls: none
Inputs:

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

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

Internal Processing

\( h_{\text{arm}} = \text{number of harmonics in expansion}, n_q[] = \text{constants used in spherical Legendre expansion}, k_1 = \text{size of } n_q[], m = \text{rows in CCIR[]}, m_m = \text{cols in CCIR[]}, m_3 = \text{total elements in CCIR[]} \)

If \( \text{siMode} = 0 \) (FoF2)

- Set constants used in spherical Legendre expansion
  \( n_q = \{11,11,8,4,1,0,0,0,0\} \)
  - Set \( k_1 \) (size of \( n_q \)) = 9

Else

- Set constants used in spherical Legendre expansion
  \( n_q = \{6,7,5,2,1,0\} \)
  - Set \( k_1 \) (size of \( n_q \)) = 7

End

Compute output value as sum of spherical Legendre functions:

\[
foF2 \text{ or } M(3000)F2 = \sum_{j=0}^{n_q} \sum_{k=1}^{k_1} c_j \sin j \mu + \sum_{j=0}^{n_q} \sin j \mu \cos k \phi \cdot (c_{R+2i} \cos k \lambda + c_{R+2i+1} \sin k \lambda)
\]

Where:

\[
R = \left[ \frac{2}{\sum_{l=0}^{k_1} (n_q_l + 1)} \right] - (n_q_0 + 1)
\]

and

- \( \sin j \mu = \text{pdSinModipToN}_j \), \( \cos \phi = \text{dCosLat} \), \( \lambda = \text{dLng} \)
- \( c_j \) comes from the input set of current spherical Legendre coefficients \( \text{pdLegCoeffs} \)

Note1: \( \sin j \mu \) terms are set to zero if found to be \( \leq 10^{-30} \)

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

9.2.9.4. NeQuick Internal Function “NeqCriticalFreqToNe”

Purpose

From critical frequency, calculates the associated electron density, using

\[ N[\text{m}^{-3}] = 0.124 \cdot 10^{11} \cdot f[\text{MHz}]^2 \]

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

Interfaces

Called by: NeqCalcEpstParams

Calls: none
Input:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dF0</td>
<td>double</td>
<td>1</td>
<td>&gt;0</td>
<td>MHz</td>
<td>The peak plasma frequency for the layer</td>
</tr>
</tbody>
</table>

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.

Called by: NeqCalcEpstParams

Calls: NeqJoin

Inputs:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dM3000</td>
<td>double</td>
<td>1</td>
<td>&gt;0</td>
<td>N/A</td>
<td>The ratio of the maximum usable frequency at a distance of 3000 km to the F2 layer critical frequency, $foF2$</td>
</tr>
<tr>
<td>dF0E</td>
<td>double</td>
<td>1</td>
<td>&gt;0</td>
<td>Hz</td>
<td>The peak plasma frequency for the E layer</td>
</tr>
<tr>
<td>dF0F2</td>
<td>double</td>
<td>1</td>
<td>&gt;0</td>
<td>Hz</td>
<td>The peak plasma frequency for the F2 layer</td>
</tr>
</tbody>
</table>

Table 36. NeqCalcF2PeakHeight Function Input Data

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

Internal Processing

$$MF = dM3000 \times \frac{0.0196 + dM3000^2 + 1}{1.2967 + dM3000^2 - 1}$$

(note: no divide by zero check needed as previous clipping ensures $M3000 \geq 1$)

if $dF0E \geq 1e^{-30}$, i.e. $dF0E$ non-zero

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

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

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

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

else

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

Compute peak height in F2 later as:

$$\text{PeakHeightF2} = \frac{1490 \times MF}{\text{dM3000} + \Delta M} - 176 \ [\text{km}]$$

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

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

Table 37. NeqCalcF2PeakHeight Function Input Data

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:

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

Internal Processing

$$\text{ExpTerm} = \text{NeqClipExp}((dH - dH_{\text{max}}) / dB)$$

Return = dN_{\text{max}} \times \text{ExpTerm} / (1 + \text{ExpTerm})^2

NB the full equation is:

$$N(h) = 4 \times N_m \times e^{\frac{h-H_m}{B}} \left(\frac{e^{\frac{h-H_m}{B}} + 1}{e^{\frac{h-H_m}{B}}}\right)^2$$

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 [Annex A [15]], although the expression in the paper has an error with the brackets.

Interfaces
Called by: NeqGetNeOnVertRay and NeqGetNeOnSlantRay

Calls: NeqCalcBottomsideNe, NeqClipExp, NeqSquared

Inputs:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dUT</td>
<td>double</td>
<td>1</td>
<td>(0,24)</td>
<td>hours</td>
<td>Time (UTC) at which STEC value is required</td>
</tr>
</tbody>
</table>

Table 38. NeqCalcTopSide Function Input Data

Inputs/Outputs:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstCurrCCIR</td>
<td>CurrentCCIR_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Information required for computing F0F2 and M300OF2 coefficients for given time and R12 value.</td>
</tr>
</tbody>
</table>

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 \( ee \)

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

where

- \( g = 0.125 \)
- \( rfac = 100 \)
- \( \text{And PeakHeight}[F2] \) and \( \text{TopThick}[F2] \) are contained in the \( \text{pstLayers} \) input data structure.

if \( ee > 10^{11} \), deal with limit when \( ee \) very large

\[
ep = \frac{4}{ee}
\]

else

\[
ep = \frac{4 \cdot ee}{(1 + ee)^2}
\]

end if

if \( pdNmax \) has not been calculated for this location, call function “NeqCalcBottomsideNe” to calculate \( pdNmax \) at PeakHeight[F2] (crossover point)

end if

Return = \( pdNmax \cdot ep \)

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dHH</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>km</td>
<td>The height at which the electron density is required</td>
</tr>
<tr>
<td>pstLayers</td>
<td>LayerProperties_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Current Ionospheric properties</td>
</tr>
</tbody>
</table>

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 each ionospheric section $j = E,F1,F2$

if 2 F2 section, $j = F2$

Calculate arg from PeakHeight[F2],

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

else 2

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

$\text{arg} = \frac{h_0 - \text{PeakHeight}[j]}{B[j]} \cdot \exp\left(\frac{f_1}{1 + f_2 - h_0 - \text{PeakHeight}[F2]}\right)$

where: $h_0 = 100$

$f_1 = 10$

$f_2 = 1$

end if 2

if 3 size of arg is large, $|\text{arg}| > 25$

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

else 3

Calculate $s_j$ and $ds_j$,

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

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

end if 3

end for 2

(Where $\text{Amp}[j]$ is included in pstLayers input data)

Calculate return result,

$N_e = a N_0 \cdot \text{NeqClipExp}\left(1 - bf * z - \text{NeqClipExp}(-z)\right)$
Where:

- \( a N_0 = 10^{11} \sum_j s_j \)
- \( b f = 1 - \frac{H_d \cdot \sum_j (d s_j \cdot s_j)}{\sum_j s_j} \)
- \( z = \frac{d H_H - h_0}{H_d} \)
- \( H_d = 10 \)  

else 1 (above 100km)

for 2 each ionospheric section \( j = E, F1, F2 \)

if 3 \( F2 \) section, \( j = F2 \)

Calculate arg from PeakHeight[\( F2 \)],

\[ \text{arg} = \frac{d H_H - \text{PeakHeight}[F2]}{B[j]} \]

else 3

Calculate arg from PeakHeight[\( F2 \)] and PeakHeight[\( j \)],

\[ \text{arg} = \frac{d H_H - \text{PeakHeight}[j]}{B[j]} \times \exp\left( f_1 \frac{1}{1 + f_2 \times d H_H - \text{PeakHeight}[F2]}\right) \]

where:  \( f_1 = 10 \)
\( f_2 = 1 \)

end if 3

if 3 size of arg is large, |arg| > 25

Value of \( s_j \) is zero, set \( s_j = 0 \)

else 3

Calculate \( s_j \),

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

end if 3

end for 2

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

\[ N_e = 10^{11} \sum_j s_j \]

e 1

9.2.12. NeqUtils.c Module

9.2.12.1. NeQuick Internal Function “NeqJoin”

Purpose

Allows smooth joining of functions \( f_1 \) and \( f_2 \) (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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dF1</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Input term for NeqJoin computation</td>
</tr>
<tr>
<td>dF2</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Input term for NeqJoin computation</td>
</tr>
<tr>
<td>dAlpha</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Input term for NeqJoin computation</td>
</tr>
<tr>
<td>dX</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Input term for NeqJoin computation</td>
</tr>
</tbody>
</table>

Table 41. NeqJoin Function Input Data
Return value: Computed value

Internal Processing
\[ ee = \text{NeqClipExp}(d\text{Alpha} + dX) \]
\[ \text{return} = (dF1 * ee + dF2) / (ee + 1); \]

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dPower</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Power for exponential function</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dValue</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Input value</td>
</tr>
</tbody>
</table>

Table 43. NeqSquared Function Input Data

Return value: Square of the input value.

Internal Processing
\[ \text{Return} = \text{dValue} * \text{dValue} \]
9.3. NeQuick Function Data Structures

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstModip</td>
<td>MODIP_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Structure containing grid of modified dip latitude values</td>
</tr>
<tr>
<td>pstCCIR</td>
<td>CCIR_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Structure containing CCIR coefficients for computing FoF2 and M(3000)F2</td>
</tr>
<tr>
<td>pdKronrodTol</td>
<td>double</td>
<td>[2]</td>
<td>&gt;0</td>
<td>N/A</td>
<td>Tolerances for Kronrod integration</td>
</tr>
<tr>
<td>siMaxRecurse</td>
<td>int</td>
<td>1</td>
<td>&gt;0</td>
<td>N/A</td>
<td>Maximum level of recursion allowed in Kronrod integration</td>
</tr>
<tr>
<td>pdGssPosLLH</td>
<td>double</td>
<td>[3]</td>
<td>Not defined</td>
<td>rad/rad/m</td>
<td>Receiver position (lat/lon/h)</td>
</tr>
<tr>
<td>pdSatPosLLH</td>
<td>double</td>
<td>[3]</td>
<td>Not defined</td>
<td>rad/rad/m</td>
<td>Satellite position (lat/lon/h)</td>
</tr>
<tr>
<td>siMonth</td>
<td>int</td>
<td>1</td>
<td>[1,12] months</td>
<td>N/A</td>
<td>Month during which STEC value is required</td>
</tr>
<tr>
<td>dUT</td>
<td>double</td>
<td>1</td>
<td>[0,24) hours</td>
<td>N/A</td>
<td>Time (UTC) at which STEC value is required</td>
</tr>
<tr>
<td>siNumCoeff</td>
<td>int</td>
<td>1</td>
<td>≥1</td>
<td>N/A</td>
<td>Number of Az coefficients</td>
</tr>
<tr>
<td>pdCoeff</td>
<td>double</td>
<td>[numCoeffs]</td>
<td>&gt;0</td>
<td>flux units/deg²</td>
<td>Az coefficients</td>
</tr>
<tr>
<td>dAzBase</td>
<td>double</td>
<td>1</td>
<td>0 - 400</td>
<td>solar flux units</td>
<td>Az value at receiver locations</td>
</tr>
</tbody>
</table>

Table 44. Definition of NeQuickInputData_st Data Structure

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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

---

Receivers 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.
### Table 46. Definition of CCIR_st Data Structure

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pdF2</td>
<td>double</td>
<td>[12][13]</td>
<td>Not defined</td>
<td>N/A</td>
<td>CCIR coefficients for computing FoF2, the critical frequency of the F2 layer</td>
</tr>
<tr>
<td>pdM3000</td>
<td>double</td>
<td>[12][9]</td>
<td>Not defined</td>
<td>N/A</td>
<td>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</td>
</tr>
</tbody>
</table>

### Table 47. Definition of SPoint_st Data Structure

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

### Table 48. Definition of CurrentCCIR_st Data Structure

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dLat</td>
<td>double</td>
<td>1</td>
<td>-90 to 90</td>
<td>deg</td>
<td>Latitude of Point</td>
</tr>
<tr>
<td>siMonth</td>
<td>int</td>
<td>1</td>
<td>[1,12] months</td>
<td>months</td>
<td>Month during which current STEC value has been computed</td>
</tr>
<tr>
<td>dR12</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Current R12 index - 12-month smoothed relative sunspot number</td>
</tr>
<tr>
<td>pdF0F2</td>
<td>double</td>
<td>[988]</td>
<td>Not defined</td>
<td>N/A</td>
<td>Interpolated coefficients for computing FoF2 for current month and R12 conditions</td>
</tr>
<tr>
<td>pdM3000F2</td>
<td>double</td>
<td>[441]</td>
<td>Not defined</td>
<td>N/A</td>
<td>Interpolated coefficients for computing M3000F2 for current month and R12 conditions</td>
</tr>
<tr>
<td>dUT</td>
<td>double</td>
<td>1</td>
<td>[0,24] hours</td>
<td>hours</td>
<td>Time (UTC) at which current STEC value has been computed</td>
</tr>
<tr>
<td>pdLegCoeffs_F0</td>
<td>double</td>
<td>[76]</td>
<td>Not defined</td>
<td>N/A</td>
<td>Spherical Legendre coefficients for calculating F0F2 for current month and R12 conditions</td>
</tr>
<tr>
<td>pdLegCoeffs_M3000</td>
<td>double</td>
<td>[49]</td>
<td>Not defined</td>
<td>N/A</td>
<td>Spherical Legendre coefficients for calculating M(3000)F2 for current month and R12 conditions</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Size</td>
<td>Range</td>
<td>Units</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>----------</td>
<td>------</td>
<td>------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pdAmp</td>
<td>double</td>
<td>3</td>
<td>Not defined</td>
<td>$10^{11}/m^3$</td>
<td>Epstein amplitude parameter</td>
</tr>
<tr>
<td>pdPeakHeight</td>
<td>double</td>
<td>3</td>
<td>Not defined</td>
<td>km</td>
<td>Epstein peak height parameter</td>
</tr>
<tr>
<td>pdBotThick</td>
<td>double</td>
<td>3</td>
<td>Not defined</td>
<td>km</td>
<td>Epstein bottom half-layer thickness parameter</td>
</tr>
<tr>
<td>pdTopThick</td>
<td>double</td>
<td>3</td>
<td>Not defined</td>
<td>km</td>
<td>Epstein top half-layer thickness parameter</td>
</tr>
<tr>
<td>dM3000</td>
<td>double</td>
<td>1</td>
<td>Not defined</td>
<td></td>
<td>Current M(3000)F2 value</td>
</tr>
<tr>
<td>pdF0</td>
<td>double</td>
<td>3</td>
<td>Not defined</td>
<td></td>
<td>Current F0 (peak plasma frequency) for the F2, F1 and E layers respectively</td>
</tr>
</tbody>
</table>

Table 49. Definition of LayerProperties_st Data Structure

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

Table 50. Definition of Geometry_st Data Structure

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Size</th>
<th>Range</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pstNeQuickInputData</td>
<td>NeQuickInputData_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Input data to NeQuick Function</td>
</tr>
<tr>
<td>pstGeom</td>
<td>GeometryData_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>Geometry data for ray</td>
</tr>
<tr>
<td>pstCurrCCIR</td>
<td>Current_st</td>
<td>1</td>
<td>Not defined</td>
<td>N/A</td>
<td>foF2 and M(3000)F2 information for current month and R12</td>
</tr>
<tr>
<td>dTolerance</td>
<td>double</td>
<td>1</td>
<td>&gt;0</td>
<td>N/A</td>
<td>Tolerance for Kronrod integration</td>
</tr>
<tr>
<td>bVert</td>
<td>Boolean</td>
<td>1</td>
<td>FALSE, TRUE</td>
<td>N/A</td>
<td>Flag indicating whether ray is vertical or not</td>
</tr>
</tbody>
</table>

Table 51. Definition of IntegrateData_st Data Structure
Europe Direct is a service to help you find answers to your questions about the European Union.

Freephone number (*):

00 800 6 7 8 9 10 11

(*) Certain mobile telephone operators do not allow access to 00 800 numbers or these calls may be billed.