NASA/TM-20240013383

Code Migration and Optimization for Calculating Atmospheric Attenuation in RF Link Budgets Within Glenn Research Center Communication Analysis Suite

Brady M. Phelps and Bryan W. Welch
Glenn Research Center, Cleveland, Ohio

January 2025

NASA STI Program Report Series

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NTRS Registered and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION.
  Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM.
  Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT.
  Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION.
  Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION.
  Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION.
  English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

NASA/TM-20240013383

Code Migration and Optimization for Calculating Atmospheric Attenuation in RF Link Budgets Within Glenn Research Center Communication Analysis Suite

Brady M. Phelps and Bryan W. Welch
Glenn Research Center, Cleveland, Ohio

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

January 2025

Level of Review: This material has been technically reviewed by technical management.

This report is available in electronic form at https://www.sti.nasa.gov/ and https://ntrs.nasa.gov/

NASA STI Program/Mail Stop 050
NASA Langley Research Center
Hampton, VA 23681-2199

Code Migration and Optimization for Calculating Atmospheric Attenuation in RF Link Budgets Within Glenn Research Center Communication Analysis Suite

Brady M. Phelps* and Bryan W. Welch
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Summary

This technical report encompasses development completed on the Glenn Research Center Communication Analysis Suite (GCAS) as it relates to converting code from the original MATLAB® (The MathWorks, Inc.) GCAS code base to C++. It focuses on a majority of the functions, structures, and intermediate code developed for radiofrequency (RF) links calculations. The report serves as a comprehensive record of the current progress and outlines the technical challenges overcome during this research. Throughout this project, numerous functions and structures were migrated end to end, with validation test cases conducted in accordance with NASA's verification and validation efforts for standard models and simulations. This report also provides high-level documentation of the completed work; detailed low-level documentation has been compiled to provide a line-by-line explanation in alignment with NASA's goals. This development has largely been focused on atmospheric effects on calculating RF link budgets for both terrestrial and space links. Some developments outside of atmospheric functions are also covered throughout this report; these are largely related to mathematical operations and MATLAB® structure conversions.

Nomenclature

2D	two-dimensional
3D	three-dimensional
GCAS	Glenn Research Center Communication Analysis Suite
ITU	International Telecommunication Union
JPL	Jet Propulsion Laboratory
RF	radiofrequency
SPICE	Spacecraft, Planet, Instrument, C-matrix, Events
StdAtm	Standard Atmospheres function

Introduction

The Glenn Research Center Communication Analysis Suite (GCAS) is a tool used to perform dynamic simulations of space-based communication systems. This tool supports line-of-sight connectivity between nodes, dynamic link performance assessment between nodes, statistical assessments of coverage for an architecture of nodes, network loading study assessments of a set of service providers to provide

*NASA Office of STEM Engagement Summer 2024 Intern, Ohio University undergraduate.

communication services to missions, as well as static (time-independent) link budget assessments using common methodologies, processes, and data structures. The tool, which was developed for internal use at the NASA Glenn Research Center, contains a user interface to configure the user's desired analysis and return the results of the analysis in standardized plots. GCAS provides a three-dimensional (3D) visualization tool to represent the scenario that the user models, while connecting nodes visible to each other if the user requests that type of connection information; it can even show the dynamic link performance via a color bar associated with the link performance metric requested. The orbital dynamics are built upon the Jet Propulsion Laboratory's (JPL's) Navigation and Ancillary Information Facility SPICE (Spacecraft, Planet, Instrument, C-matrix, Events) Toolkit and data sets. The link budget calculations use International Telecommunication Union (ITU) models to derive the effect of the Earth's atmosphere on the communication link's signal strength. Because the GCAS tool is being converted from MATLAB® to C++, this report discusses some of the various model aspects and challenges observed throughout the process.

Low-Complexity Data Structures

During the conversion process, several simple data structures were created to match the level of abstraction found in the MATLAB® code. This section provides technical documentation on these data structures, which are used by other functions throughout the migrated code. Figure 1 and Figure 2 illustrate these structures and their public members. The names of the structures are on the left side of the figures; the features on the right side of the structures represent the public member names and their data types.

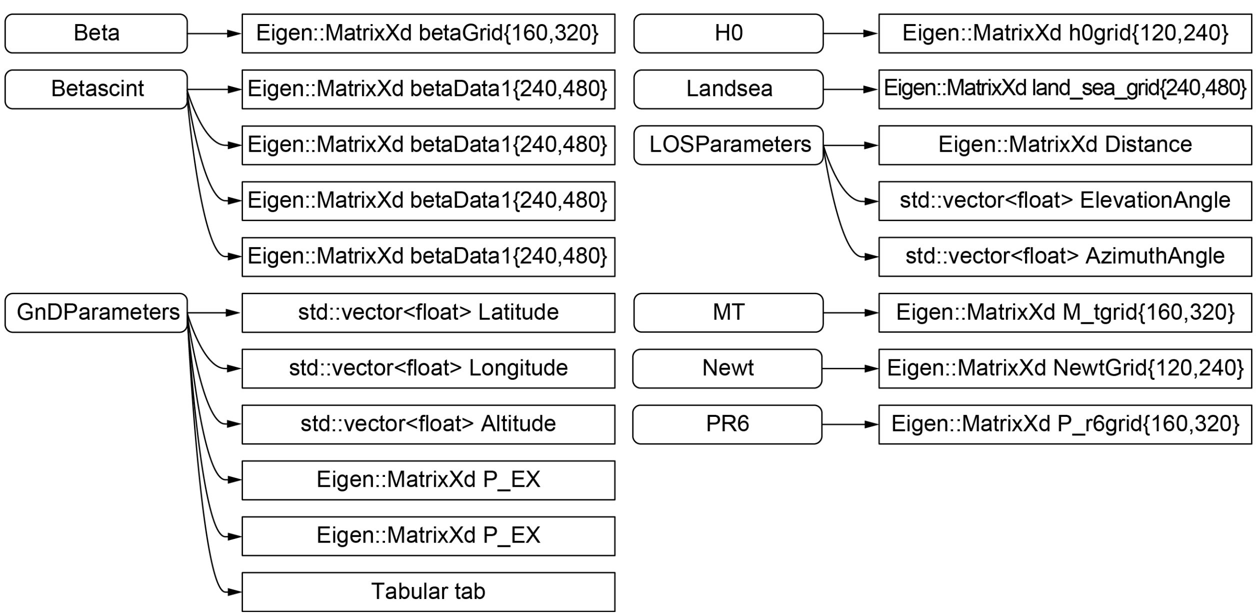

Figure 1.—Low-complexity data structures used in ATM (atmospheric) calculations.

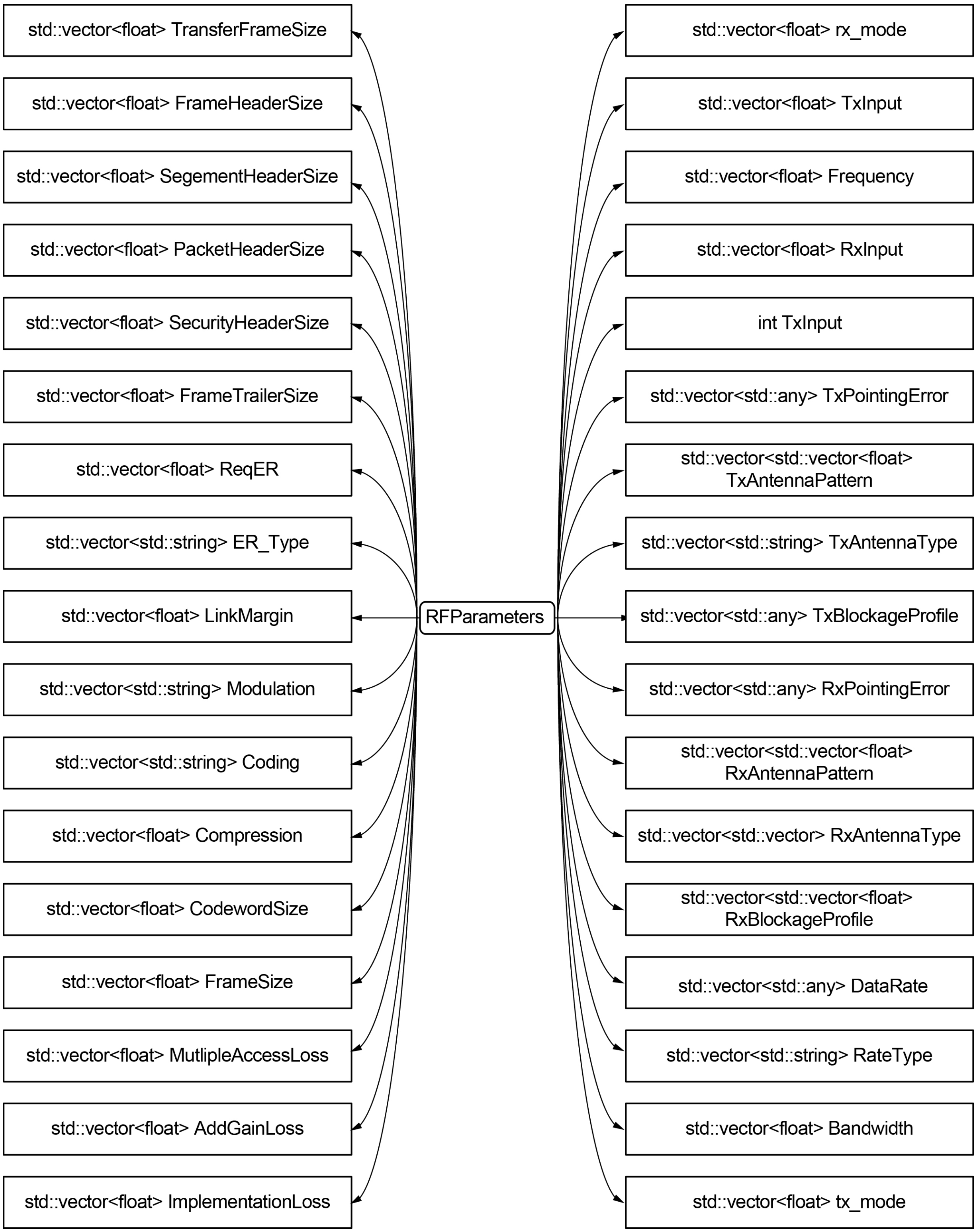

Figure 2.—RFParameters structure, used to store information about specific radiofrequency (RF) instances.

Atmospheric Functions

Numerous atmospheric functions were developed to model varying atmospheric effects, conditions, and scenarios for calculating RF link budgets. This section provides descriptions of these functions, along with the challenges encountered and the rationale behind their development during the migration process. Additionally, all functions developed in this section have accompanying verification and validation testing with verification data from corresponding ITU-R sections.

Standard Atmospheres

The Standard Atmospheres (StdAtm) function calculates the temperature, pressure, and water vapor density at a given location on the Earth for a specified season. This data is used to determine gaseous atmospheric attenuation, which occurs when RF electromagnetic waves pass through the atmosphere along Earth-space paths. A percentage of these waves are absorbed (attenuated) by the atmosphere, with denser atmospheric conditions leading to greater attenuation. Specifically, attenuation increases with lower temperatures, higher pressures, or higher water vapor densities. The calculations and code for temperature, pressure, and water vapor density are based on the piece-wise functions outlined in Sections 2 to 4 of Reference 1. These equations were translated into code and integrated into the converted StdAtm and StdAtm_V2 functions to handle single and multiple altitudes.

Key

T = temperature, K
h = height, km
P = pressure, hPa
ρ = water vapor, g/m^3

Low-Latitude Annual Reference Atmosphere

Low latitudes do not have seasonal variations. For latitudes lower than 22°.

$$T(h) = 300.4222 - 6.3533h + 0.005886h^2 \text{ for } 0 \le h < 17 \tag{1}$$

$$T(h) = 194 + (h - 17) \times 2.533 \text{ for } 17 \le h < 47 \tag{2}$$

$$T(h) = 270 \text{ for } 47 \le h < 52 \tag{3}$$

$$T(h) = 270 - (h - 52) \times 3.0714 \text{ for } 52 \le h < 80 \tag{4}$$

$$T(h) = 184 \text{ for } 80 \le h \le 100 \tag{5}$$

$$P(h) = 1012.0306 - 109.0338h + 3.6316h^2 \text{ for } 0 \le h \le 10 \tag{6}$$

$$P(h) = P_{10} \exp\left[-0.147(h - 10)\right] \text{ for } 10 < h \le 72 \tag{7}$$

$$P(h) = P_{72} \exp\left[-0.165(h - 72)\right] \text{ for } 72 < h \le 100 \tag{8}$$

$$\rho(h) = 19.6542 \exp\left[-0.2313h - 0.1122h^2 + 0.01351h^3\right] \text{ for } 0 \le h \le 15 \tag{9}$$

$$\rho(h) = 0 \text{ for } h > 15 \tag{10}$$

Mid-Latitude Reference Atmosphere

For latitudes between 22° and 45°:

Summer

$$T(h) = 294.9838 - 5.2159h - 0.07109h^2 \text{ for } 0 \le h < 13 \quad (11)$$

$$T(h) = 215.15 \text{ for } 13 \le h < 17 \quad (12)$$

$$T(h) = 215.15 \exp[(h - 170.008128)] \text{ for } 17 \le h < 47 \quad (13)$$

$$T(h) = 275 \text{ for } 47 \le h < 53 \quad (14)$$

$$T(h) = 275 + \{1 - \exp[(h - 53)0.06]\} \times 20 \text{ for } 53 \le h < 80 \quad (15)$$

$$T(h) = 175 \text{ for } 80 \le h \le 100 \quad (16)$$

$$P(h) = 1012.8186 - 111.5569h + 3.8646h^2 \text{ for } 0 \le h \le 10 \quad (17)$$

$$P(h) = P_{10} \exp[-0.147(h - 10)] \text{ for } 10 < h \le 72 \quad (18)$$

$$P(h) = P_{72} \exp[-0.165(h - 72)] \text{ for } 72 < h \le 100 \quad (19)$$

$$\rho(h) = 14.3542 \exp\left[-0.4147h - 0.02290h^2 + 0.001007h^3\right] \text{ for } 0 \le h \le 15 \quad (20)$$

$$\rho(h) = 0 \text{ for } h > 15 \quad (21)$$

Winter

$$T(h) = 272.7241 - 3.6217h - 0.1759h^2 \text{ for } 0 \le h < 10 \quad (22)$$

$$T(h) = 218 \text{ for } 10 \le h < 33 \quad (23)$$

$$T(h) = 218 + (h - 33)3.3571 \text{ for } 33 \le h < 47 \quad (24)$$

$$T(h) = 265 \text{ for } 47 \le h < 53 \quad (25)$$

$$T(h) = 265 - (h - 53) \times 2.0370 \text{ for } 53 \le h < 80 \quad (26)$$

$$T(h) = 210 \text{ for } 80 \le h \le 100 \quad (27)$$

$$P(h) = 1018.8627 - 124.2954h + 4.8307h^2 \text{ for } 0 \le h \le 10 \quad (28)$$

$$P(h) = P_{10} \exp[-0.147(h - 10)] \text{ for } 10 < h \le 72 \quad (29)$$

$$P(h) = P_{72} \exp[-0.155(h - 72)] \text{ for } 72 < h \le 100 \quad (30)$$

$$\rho(h) = 3.4742 \exp\left[-0.2697h - 0.03604h^2 + 0.0004489h^3\right] \text{ for } 0 \le h \le 10 \quad (31)$$

$$\rho(h) = 0 \text{ for } h > 10 \quad (32)$$

High-Latitude Reference Atmosphere

For high latitudes (greater than 45°):

Summer

$$T(h) = 286.8374 - 4.7805h - 0.1402h^2 \text{ for } 0 \le h < 10 \quad (33)$$

$$T(h) = 225 \text{ for } 10 \le h < 23 \quad (34)$$

$$T(h) = 225\exp[(h-23)0.008317] \text{ for } 23 \le h < 48 \quad (35)$$

$$T(h) = 277 \text{ for } 48 \le h < 53 \quad (36)$$

$$T(h) = 277 - (h-53)\times 4.0769 \text{ for } 53 \le \text{h} < 79 \quad (37)$$

$$T(h) = 171 \text{ for } 79 \le h \le 100 \quad (38)$$

$$P(h) = 1008.0278 - 113.2494h + 3.9408h^2 \text{ for } 0 \le h \le 10 \quad (39)$$

$$P(h) = P_{10}\exp[-0.140(h-10)] \text{ for } 10 < h \le 72 \quad (40)$$

$$P(h) = P_{72}\exp[-0.165(h-72)] \text{ for } 72 < h \le 100 \quad (41)$$

$$\rho(h) = 8.988\exp[-0.3614h - 0.005402h^2 - 0.001955h^3] \text{ for } 0 \le h \le 15 \quad (42)$$

$$\rho(h) = 0 \text{ for } h > 15 \quad (43)$$

Winter

$$T(h) = 257.4345 + 2.3474h - 1.5479h^2 + 0.08473h^3 \text{ for } 0 \le h < 8.5 \quad (44)$$

$$T(h) = 217.5 \text{ for } 8.5 \le h < 30 \quad (45)$$

$$T(h) = 217.5 + (h-30)\times 2.125 \text{ for } 30 \le h < 50 \quad (46)$$

$$T(h) = 260 \text{ for } 50 \le h < 54 \quad (47)$$

$$T(h) = 260 - (h-54)\times 1.667 \text{ for } 54 \le h \le 100 \quad (48)$$

$$P(h) = 1010.8828 - 122.2411h + 4.554h^2 \text{ for } 0 \le h \le 10 \quad (49)$$

$$P(h) = P_{10}\exp[-0.147(h-10)] \text{ for } 10 < h \le 72 \quad (50)$$

$$P(h) = P_{72}\exp[-0.150(h-72)] \text{ for } 72 < h \le 100 \quad (51)$$

$$\rho(h) = 1.2319\exp[0.07481h - 0.0981h^2 + 0.00281h^3] \text{ for } 0 \le h \le 10 \quad (52)$$

$$\rho(h) = 0 \text{ for } h > 10 \quad (53)$$

Equations (1) to (53) represent the piece-wise functions found in Reference 1 to calculate temperature, pressure, and water vapor for attenuation calculations.

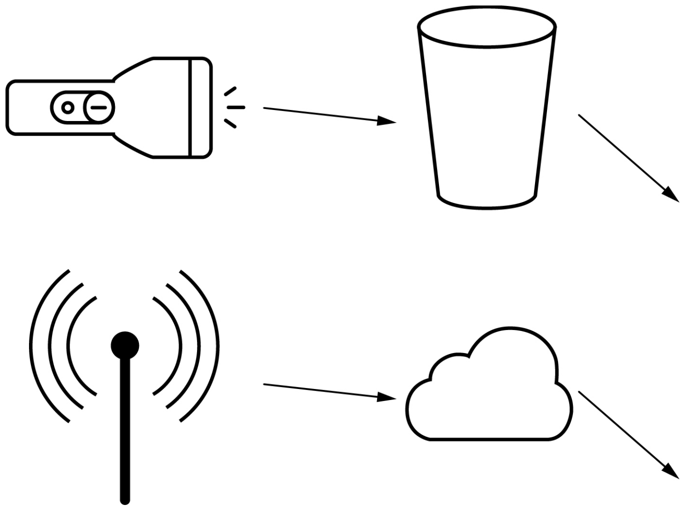

Figure 3.—Refractivity of RF signals through atmosphere works much like shining flashlight through water.

Radio Refractivity Index

The calcNwet function calculates the wet radio refractive index, which measures the refraction caused by molecules in the atmosphere. As RF waves travel through the atmosphere, they are refracted by these molecules in a manner similar to light bending when passing through water (see Figure 3). The calcNwet function performs these calculations based on atmospheric temperature, humidity, and pressure.

- Temperature: Higher temperatures decrease refractivity by reducing atmospheric density.
- Pressure: Higher pressure increases refractivity by condensing particles and making the atmosphere denser.
- Humidity: Higher humidity also increases atmospheric density, thereby increasing refractivity.

The equations used in these calculations are derived from Reference 2 and are integrated into the migrated code to determine the radio refractive index and its behavior. These functions describe both surface and vertical profile characteristics and provide global maps of refractivity parameter distributions and their statistical variations.

Key

T = absolute temperature, K
t = temperature, °C
e = water vapor pressure, hPa
H = relative humidity, percent
e_s = saturation vapor pressure, hPa, at t

$$N_{Wet} = 72\frac{e}{T} + 3.75\times 10^5 \frac{e}{T^2} \qquad (54)^1$$

[1]Equation (4) from Reference 2.

$$e = \frac{H \times e_s}{100} \quad (55)^2$$

$$e_s = EF \times a \times \exp\left[\frac{\left(b - \frac{t}{d}\right) \times t}{t + c}\right] \quad (56)^3$$

$$e = \frac{\rho T}{216.7} \quad (57)^4$$

Rain Height

The rain height (rainHeight) function has been fully converted and validated with test cases. The function estimates the rain height in kilometers for propagation prediction, which is crucial for calculating attenuation along slant paths (Figure 4), such as Earth-to-space or space-to Earth links. The higher the rain is, the greater the loss the link will undergo. The higher the rain is, the longer the link will spend in the rain, which causes more loss (Figure 5). The function takes a latitude, longitude, and a data file of average rain heights as inputs to compute the expected rain height. Rain and precipitation affect RF attenuation through the scattering and absorption of RF waves as they travel through the precipitation. The amount of attenuation also relies on several other factors, such as frequency and altitude.

$$h_R = h_0 + 0.36 \quad (58)^5$$

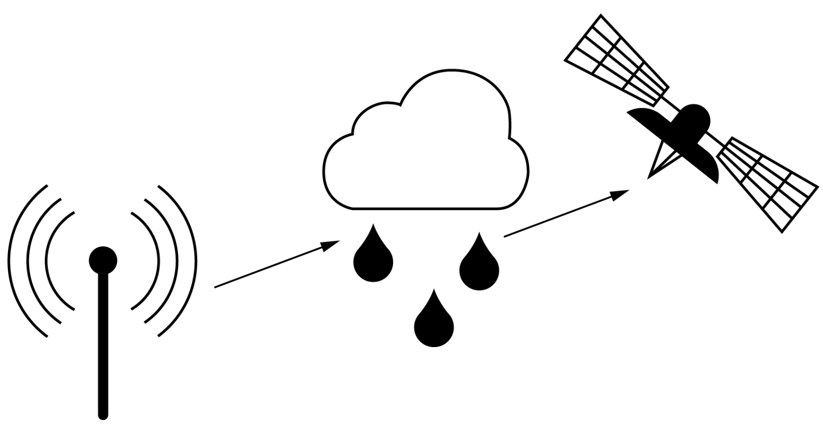

Figure 4.—Rain height plays critical factor in calculating RF attenuation for slant paths. Slant paths connect terrestrial and space nodes.

[2]Equation (8) from Reference 2.
[3]Equation (9) from Reference 2.
[4]Equation (10) from Reference 2.
[5]Equation (1) from Reference 3, used to calculate rain height using mean annual rain height data from above sea level.

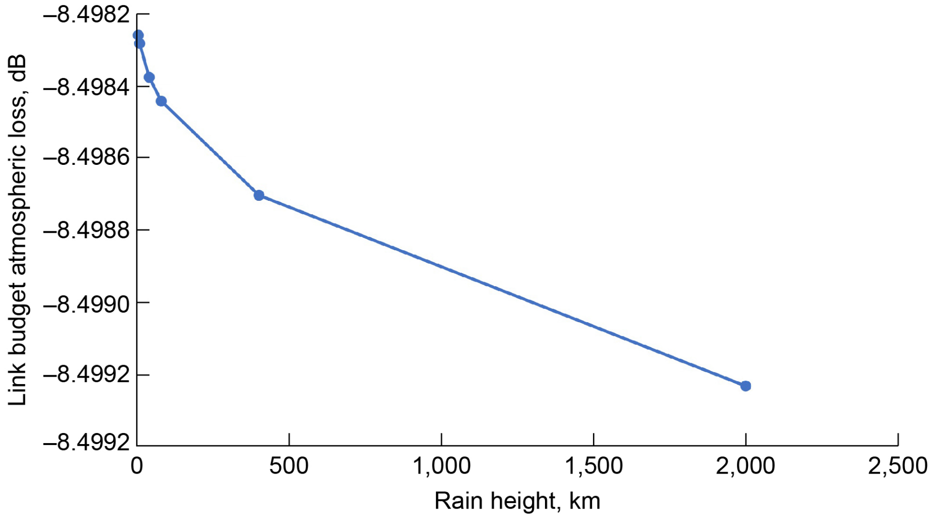

Figure 5.—Higher rain induces greater atmospheric loss on link. Some data points from this test are not feasible in real life (cloud layer ends at 30 km), but data points were used beyond 30 km to better showcase relationship between rain height and link budget loss.

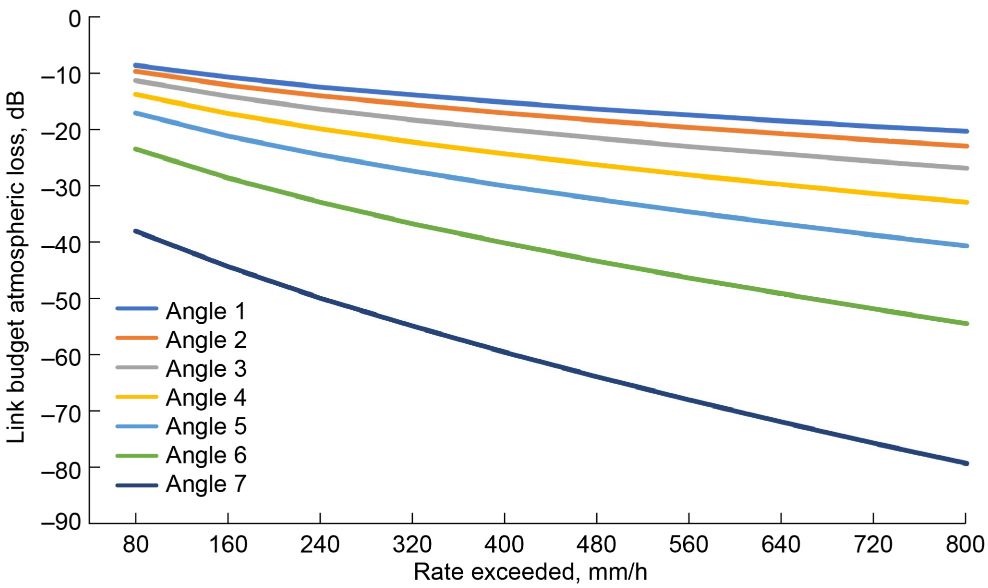

Figure 6.—Effects of increasing rain rate on link budget loss. As RF signals pass through precipitation, attenuation increases with rate of rain. Angles in this figure are decreasing from 1 to 7. As RF signals traverse closer to Earth's surface, attenuation increases due to increased atmospheric thickness.

Rain Rate

The rain rate function (rainRate) function was completely converted with test case validation. The function calculates the rain rate (mm/h) exceeded for a given percentage of an average year at a given location. As the rain rate increases, so does the attenuation of RF signals (Figure 6). To run this function, it requires a latitude, longitude, percentage of an average year that rain rate is predicted to exceed, and three data files for mean annual rainfall, ratio of convectivity to total rainfall amount, and probability of rainy 6-h periods. Equations for this function were developed from the equations provided in Reference 4, Annex 1.

Additionally, while testing data and converting code for both the rain rate and rain height functions, a bug was discovered in the original GCAS relating to rain calculations. A logical error in the code prevented rain attenuation from being calculated. As a result, this bug is now fixed in the official version of GCAS.

$$P_0(Lat,Lon) = P_{r6}(Lat,Lon)\left(1 - e^{-0.0079\left(\frac{M_S(Lat,Lon)}{P_{r6}(Lat,Lon)}\right)}\right) \quad (59)^6$$

$$R_p(Lat,Lon) = \frac{-B + \sqrt{B^2 - 4AC}}{2A} \quad (60a)^7$$

$$A = ab \quad (60b)$$

$$B = a + c\,\ln\left(\frac{p}{P_0(Lat,Lon)}\right) \quad (60c)$$

$$C = \ln\left(\frac{p}{P_0(Lat,Lon)}\right) \quad (60d)$$

$$a = 1.09 \quad (60e)$$

$$b = \frac{(M_c(Lat,Lon) + M_s(Lat,Lon))}{21797 P_0} \quad (60f)$$

$$c = 26.02b \quad (60g)$$

Specific Attenuation

The Specific Attenuation (SpAtt) function was converted from MATLAB® to C++ and is designed to calculate specific attenuation due to atmospheric gases. It computes the specific attenuation (dB/km) of an RF signal up to 1,000 GHz caused by Earth's atmosphere, using parameters such as temperature, air pressure, and water vapor density, in accordance with Reference 5.

The function performs the following steps:

- Spectral resonance lines: It first calculates the spectral resonance lines of oxygen and water vapor. These resonance lines are necessary for understanding the absorption characteristics of the atmosphere.
- Nonresonant and additional factors: The function incorporates additional factors to refine the calculation, including
  - Debye spectrum: For frequencies lower than 10 GHz, it accounts for the nonresonant Debye spectrum of oxygen.
  - Nitrogen attenuation: For frequencies higher than 100 GHz, it considers pressure-induced nitrogen attenuation.
  - Wet continuum: It includes a wet continuum to account for excess water vapor absorption observed experimentally.

[6] Equation (1) from Reference 4.
[7] Equation (2) from Reference 4.

- Layer-based calculations: The function calculates the specific attenuation for 922 individual atmospheric layers, representing the atmosphere from the ground up to 100 km in altitude.

The following equations detail the specific formulas that were converted to C++.

Key

S_i = strength of i^{th} oxygen or water vapor line
F_i = oxygen or water vapor line shape factor
P = dry air pressure, hPa
e = water vapor partial pressure, hPa
T = temperature, K
$\theta = 300/T$

$$N''_{oxygen}(f) = \sum_{i(oxygen)} S_i F_i + N''_D(f) \tag{61a}$$[8]

$$N''_{water\ vapor}(f) = \sum_{i(water\ vapor)} S_i F_i \tag{61b}$$

$$S_i = a_1 \times 10^{-7} p\theta^3 \exp[a_2(1-\theta)] \text{ for oxygen} \tag{62a}$$[9]

$$S_i = b_1 \times 10^{-1} e\theta^{3.5} \exp[b_2(1-\theta)] \text{ for water vapor} \tag{62b}$$

$$F_i = \frac{f}{f_i}\left[\frac{\Delta f - \delta(f_i - f)}{(f_i - f)^2 + \Delta f^2}\right] \tag{63}$$[10]

$$\Delta f = a_3 \times 10^{-4}\left(p\theta^{(0.8-a_4)} + 1.1e\theta\right) \text{ for oxygen} \tag{64a}$$[11]

$$\Delta f = b_3 \times 10^{-4}\left(p\theta^{b_4} + b_5 e\theta^{b_6}\right) \text{ for water vapor} \tag{64b}$$

$$\Delta f = \sqrt{\Delta f^2 + 2.25 \times 10^{-6}} \text{ for oxygen} \tag{65a}$$[12]

$$\Delta f = \sqrt{0.217\Delta f^2 + \frac{2.1316 \times 10^{-12} f_i^2}{\theta}} \text{ for water vapor} \tag{65b}$$

---

[8]Equations (61a) and (61b) are Equations (2a) and (2b) from Reference 5. These equations calculate the imaginary parts of the frequency-dependent complex reactivities.
[9]Equations (62a) and (62b) are Equation (3) from Reference 5. These equations calculate the imaginary parts of the frequency-dependent complex reactivities.
[10]Equation (5) from Reference 5. This equation calculates the line shape factor.
[11]Both Equations (64a) and (64b) represent Equation (6a) from Reference 5. These equations are used to find the oxygen or water line width.
[12]Both Equations (65a) and (65b) represent Equation (6b) from Reference 5. These equations are used in conjunction with Equation (6a) from the same source, and these specifically modify line width to account for Zeeman splitting of oxygen lines and Doppler broadening of water vapor lines.

$$\delta = (a_5 + a_6\theta)\times 10^{-4}(p+e)\theta^{0.8} \text{ for oxygen}$$
$$\delta = 0 \text{ for water vapor} \qquad (66)^{13}$$

$$N''_d(f) = fp\theta^2\left[\frac{6.14\times 10^{-5}}{d\left[1+\left(\frac{f}{d}\right)^2\right]} + \frac{1.4\times 10^{-12}p\theta^{1.5}}{1+1.9\times 10^{-5}f^{1.5}}\right] \qquad (67)^{14}$$

$$d = 5.6\times 10^{-4}(p+e)\theta^{0.8} \qquad (68)^{15}$$

Additionally, the specific attenuation function relies on another smaller function converted from MATLAB® to C++. This function is known as spectralData and was converted from a MATLAB® function to a spectralData structure and a createSpectral function to properly load in the spectral data. This function loads the spectral data for the absorption lines of oxygen and water vapor respectively. Data is obtained from Tables 1 and 2 of Reference 5.

Refractivity

The refractivity (Refr) function was divided into two separate functions, Refr and Refr2, in C++. Both functions have been validated with test cases and differ in the arguments they accept and their return types. The Refr function calculates the refractivity for a single point in the atmosphere, while Refr2 calculates refractivity for multiple points. These functions determine refractivity based on temperature, pressure, and water vapor density, following the guidelines set forth in Reference 6.

Key

P_d = dry atmospheric pressure, hPa
P = total atmospheric pressure, hPa
E = water vapor pressure, hPa
T = absolute temperature, K

$$n = 1 + N\times 10^{-6} \qquad (69)^{16}$$

$$N = 77.6\frac{P_d}{T} + 72\frac{e}{T} + 3.75\times 10^5\frac{e}{T^2} \qquad (70)^{17}$$

[13]Equation (7) from Reference 5. This equation represents the correction factor that arises due to interference effects in oxygen lines.
[14]Equation (8) from Reference 5. This equation finds the dry air continuum from the nonresonant Debye spectrum of oxygen lower than 10 GHz and a pressure-induced nitrogen attenuation about 100 GHz.
[15]Equation (9) from Reference 5. This equation finds the dry air continuum from the nonresonant Debye spectrum of oxygen lower than 10 GHz and a pressure-induced nitrogen attenuation about 100 GHz.
[16]This equation represents Equation (1) from Reference 6. This equation calculates the radio refractive index.
[17]This equation represents Equation (2) from Reference 6. This equation calculates the radio refractive index.

Lred

A functional data set loader for Lred was created using the Eigen (Ref. 7) and HighFive (Ref. 8) libraries, known as createLred. This loader is used to import various atmospheric settings and scenarios into GCAS. The data set contains more than 8.6 million values and loading it with standard C++ would crash the program due to memory limitations.

A custom parser to automatically convert the MATLAB® loader to C++ code was created initially. This script was highly effective, automating much of the code transcription process. However, the transcribed function encountered significant issues. Specifically, the loading time grew exponentially with the data set size, and the data would exhaust the machine's RAM, causing the program to crash.

To address these challenges, a different approach was adopted. Instead of using raw C++, a dynamic loading strategy was implemented, combined with compressed HighFive versions of the data set. This new method significantly improves both speed and memory efficiency compared to the scripted implementation.

Geoclimatic Factor

Functions pertaining to calculating the geoclimatic factor were also converted. The geoclimatic factor has effects on the ionospheric scintillation and the deep fading of a signal. Ionospheric scintillation occurs when radio waves pass through irregularities in the ionosphere, causing rapid fluctuations in the signal amplitude and phase. These effects tend to be more pronounced in certain geoclimatic regions. The specific region of these Earth-to-space or space-to-Earth links can be found by interpolating the given link positions along the land-sea-grid to find the fraction of the propagation path that crosses a body of water or adjacent coastal areas (Figure 7). Lakes and other bodies of water are not included in the calculations, as their effect is negligible in the final calculations. When a path travels over sea, the sea has higher levels of surface conductivity and fewer obstacles, which results in lower levels of attenuation.

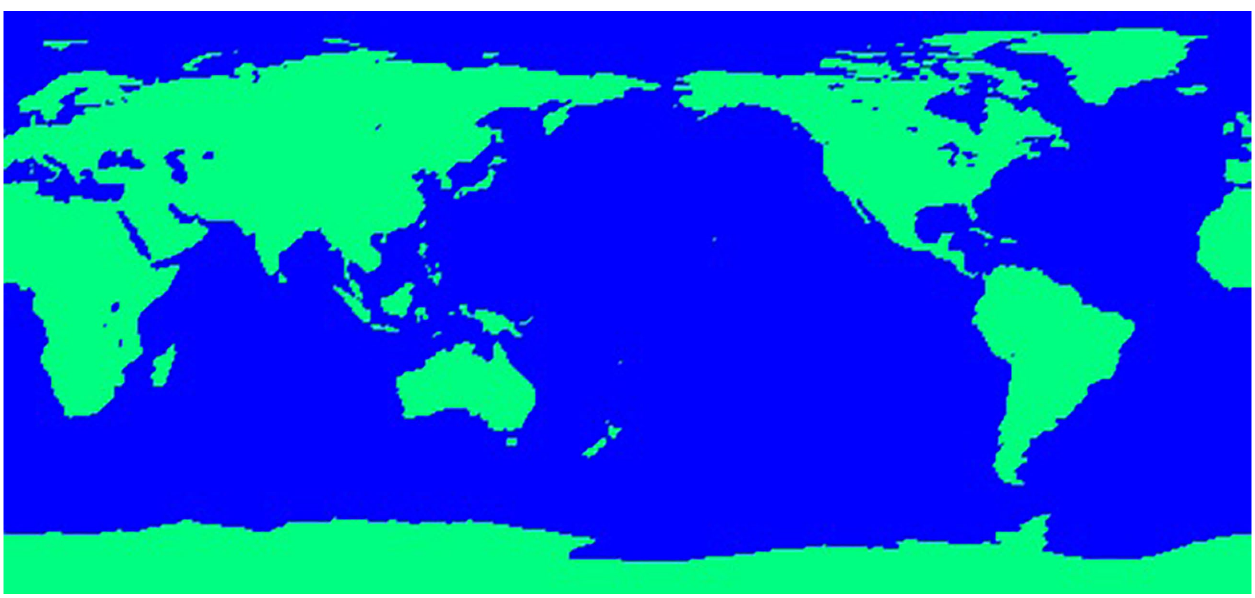

Figure 7.—Land-sea data that constructs global map using ones and zeroes to calculate the fraction of the propagation path that crosses body of water or adjacent coastal areas. Green indicates ones (continents) and blue indicates zeros (water).

Functions related to calculating the constant C_0 were completed with test case validation. The full geoclimatic factor function was also completed but is awaiting testing.

$$K_w = p_L^{1.5} \times 10^{\frac{C_0 + C_{Lat}}{10}} \quad (71)^{18}$$

MATLAB® Functions

Bilinear Interpolation

Bilinear interpolation is a common feature used in the MATLAB® version of GCAS that was not yet implemented in the C++ optimized rewrite. Through this work, a function was created to mimic the behavior of interp2 (bilinear interpolation) from MATLAB®. In this function, a user can feed an array of X values, Y values, function values, and interpolation points to the function. The function then uses the bilinear interpolation equations for each interpolated point and calculates the interpolated value. The function will go through each of the queried points and will establish a bound of the four closest values for each of the points. The data is represented as a two-dimensional (2D) grid of function values, so the function will take the bounds of the closest X and Y values to establish the four closest values on the grid (Figure 8). Then, using these values, standard bilinear formulas are used to calculate the interpolated values. Essentially, the function uses bilinear interpolation to calculate estimated points on a 2D set of data.

Mesh-Grid

An overloaded version of mesh-grid was added to the preexisting Geometry namespace to fit constraints within the RF side of the program. This new function will return a mesh-grid for two sets of starting values, ending values, and incremental steps. This is largely used throughout the atmospheric side of the program to generate latitude and longitude data. This new version of the function is the only implementation of mesh-grid within the C++ project that contains validation test cases.

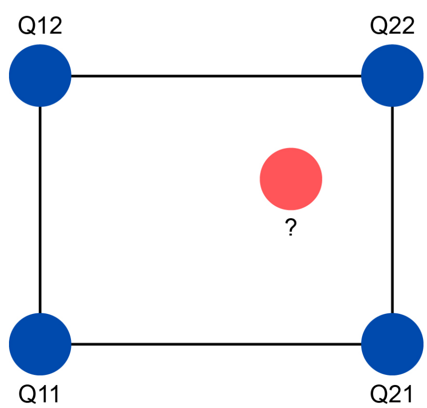

Figure 8.—This figure allows visualizing how interpolation of a data point is based on nearby reference data. Here are found four points closest to value to be estimated. To solve this, one would use four known closest values in interpolation function to solve for "?".

[18]The function for calculating the geoclimatic factor. Equation (50) in Reference 9.

MATLAB® and Eigen Operations

Several functions from MATLAB® could not be directly implemented using Eigen's standard capabilities due to issues with element-wise operations and mismatched row or column dimensions. For example, Eigen does not natively support element-wise power operations. While Eigen does offer some support for element-wise addition and multiplication, problems arose when row dimensions did not match.

To address these challenges, custom functions were developed to handle various dimension mismatch cases and apply the appropriate Eigen operations for each scenario.

The following functions have been converted to handle element-wise operations effectively.

- elementAdd—For special cases where "+ .array()" will not work for adding Eigen matrices.
- elementPow—To implement element-wise power function on matrices.
- elementPowLongB—To implement element-wise power function on matrices when the exponent has more rows than the base.
- elementMult—To implement special instances of element-wise multiplication.
- elementDiv—To implement special instance of element-wise division operations.

Documentation

In addition to this report, comprehensive low-level documentation has been compiled for the converted code. This documentation, generated using Doxygen (Ref. 10), details all data structures and functions that were converted during development (Figure 9 and Figure 10). It facilitates easy navigation both within the code and in the separately compiled documents. The documentation process supports the verification efforts of NASA–STD–7009 (Ref. 11). The documentation is integrated into the codebase itself, providing valuable guidance for future developers working with the completed code.

PR6.hpp	PR6 structure for P_r6grid
RFParameters.hpp	RFparameters - struct containing RF parameters
Tabular.hpp	Tabular structure. Contains lred data, PR6 data, and MT data
calcNwet.hpp	CalcNwet function that calculates the wet radio refractive index (later used to calculate scint)
calculate_RF_AtmAttenuation.hpp	Calculate_RF_AtmAttenuation - determines the RF atmospheric attenuation
createTabFunctions.hpp	Dynamically create tabular data sets
MatLabUtils.hpp	Useful file for MATLAB utilities specific to RF_Links migration
rainHeight.hpp	Estimates the rain height (km) of a given location
rainRate.hpp	RainRate - Rain rate exceeded for a given percentage of an average year Estimates the rain rate (mm/h) exceeded for the percentage, p_ex of an average year at the given latitude (deg) and longitude (deg)
Refr.hpp	Calculates refractivity of a point in the atmosphere given the temperature, pressure, and water vapor density, rho accordin reccomendations (P.453-12)
RF_el_Pre_V3.hpp	Elevation-independent calculations for RF atmospheric attenuation Performs all calculations for atmospheric attenuation independent of angle of elevation to generate input parameters for RF_el.m
scanlinktoolV6_RF_NoRepeater.hpp	In progress
SpAtt_V2.hpp	Specific attenuation due to atmospheric gasses Calculates the specific attenuation (dB/km) of an RF signal up to 1000 Gi Earth's atmosphere for the given temperature, air pressure, and water vapor density (rho) according to International Teleco Union Radiocommunication Sector (ITU-R) Recommendation P.676-11
spectralData.hpp	Object and functions for creating spectral dataset for oxygen and water vapor
StdAtm.hpp	Definition and declaration for StdAtm. Estimated temperature (K), pressure (hPa), and water vapor density, rho (g/m^3), of Earth's atmosphere for a given latitude, altitude, and time of year (summer or winter) according to International Telecomm Union Radiocommunication Sector (ITU-R) Recommendation P.835-5
StdAtm_V2.hpp	Estimated temperature (K), pressure (hPa), and water vapor density, rho (g/m^3), of a point in Earth's atmosphere for a giv altitude, and time of year (summer or winter) according to International Telecommunication Union Radiocommunication S Recommendation P.835-5

Figure 9.—Portion of file documentation overview. Documentation was generated using Doxygen.

Function Documentation

◆ calcNwet()

```
std::pair< double, double > calcNwet ( double temp,
                                       double humidity,
                                       double pressure )
```

calculates the wet radio refractive index (later used to calculate scint)

Parameters

temp	the long term average temperature of the area (C)
humidity	the long term average relative humidity of the area (%)
pressure	the long term average pressure of the area (hPa)

Returns

std::pair<double, double> nwet (first double) - the wet radio refractive index & rho (second double) - the denisty of liquid water

Note

see also ITU-R P.453-13 Annex 1

Figure 10.—Detailed function view for calcNwet. Documentation was generated using Doxygen.

Conclusions

Throughout this development, more than 10,000 lines of codes were written for the contribution of code conversion from MATLAB® to C++ for the Glenn Research Center Communication Analysis Suite (GCAS). This includes converting more than 70 ITU formulas, numerous data structures, additional work for MATLAB® mathematical operation conversion, verification and validation testing for completed functions, and documentation compiled to assist in future development of the project and to align with NASA's goals of clear and thorough technical documentation.

Moving forward with the continued development and conversion of the program, the next steps in converting the atmospheric calculations are the completion of the calculate_RF_AtmAttenuation and RF_el_Pre_V3 files. Heavy development has been completed through this research, and the functions/code converted in this work makes for great progress towards the completion of these functions and the completion of atmospheric calculations for RF links within the C++ version of GCAS.

References

1. International Telecommunication Union: Reference Standard Atmospheres. ITU–R P.835–5, 2012. https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.835-5-201202-S!!PDF-E.pdf Accessed Nov. 18, 2024.
2. International Telecommunication Union: The Radio Refractive Index: Its Formula and Refractivity Data. ITU–R P.453–13, 2017. https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.453-13-201712-S!!PDF-E.pdf Accessed Nov. 18, 2024.
3. International Telecommunication Union: Rain Height Model for Prediction Methods. ITU–R P.839–4, 2013. https://www.itu.int/dms_pubrec/itu-r/rec/p/r-rec-p.839-4-201309-i!!pdf-e.pdf Accessed Nov. 18, 2024.

4. International Telecommunication Union: Characteristics of Precipitation for Propagation Modelling. ITU–R P.837–6, 2012. https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.837-6-201202-S!!PDF-E.pdf Accessed Nov. 20, 2024.
5. International Telecommunication Union: Attenuation by Atmospheric Gases. ITU–R P.676–11, 2016. https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.676-11-201609-I!!PDF-E.pdf Accessed Nov. 20, 2024.
6. International Telecommunication Union: The Radio Refractive Index: Its Formula and Refractivity Data. ITU–R P.453–12, 2016. https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.453-12-201609-S!!PDF-E.pdf Accessed Nov. 20, 2024.
7. Eigen: A C++ Template Library for Linear Algebra. 2021. https://eigen.tuxfamily.org Accessed Aug. 5, 2024.
8. HighFive: HighFive - HDF5 header-only C++ Library. 2024. https://bluebrain.github.io/HighFive/ Accessed Dec. 10, 2024.
9. International Telecommunication Union: Propagation Data and Prediction Methods Required for the Design of Earth-Space Telecommunication Systems. ITU–R P.618–12, 2015. https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.618-12-201507-S!!PDF-E.pdf Accessed Nov. 20, 2024.
10. Doxygen: Documentation System for C++, C, Java, and Other Languages. 2024. https://www.doxygen.nl Accessed Aug. 5, 2024.
11. National Aeronautics and Space Administration: Standard for Models and Simulations. NASA–STD–7009B, 2024.