NASA/TM-20210022058

Jupiter Global Reference Atmospheric Model (Jupiter-GRAM): User Guide

H. L. Justh
Marshall Space Flight Center, Huntsville, Alabama

A. M. Dwyer Cianciolo
Langley Research Center, Hampton, Virginia

J. Hoffman
Analytical Mechanics Associates, Hampton, Virginia

September 2021

NASA STI Program Report Series

The NASA STI Program 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 compila-
tions of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part 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 co-sponsored 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

- Help desk contact information: https://www.sti.nasa.gov/sti-contact-form/ and select the "General" help request type.

NASA/TM-20210022058

Jupiter Global Reference Atmospheric Model (Jupiter-GRAM): User Guide

H. L. Justh
Marshall Space Flight Center, Huntsville, Alabama

A. M. Dwyer Cianciolo
Langley Research Center, Hampton, Virginia

J. Hoffman
Analytical Mechanics Associates, Hampton, Virginia

National Aeronautics and
Space Administration

Marshall Space Flight Center
Huntsville, Alabama 35812

September 2021

Acknowledgment

The authors thank the NASA Science Mission Directorate (SMD) for their support of updating and developing new Global Reference Atmospheric Models (GRAMs) including Jupiter-GRAM.

Trademarks

Available from:

NASA STI Information Desk
Mail Stop 148
NASA Langley Research Center
Hampton, VA 23681–2199, USA
757–864–9658

This report is also available in electronic form at
<http://www.sti.nasa.gov>

PREFACE

The NASA Jupiter Global Reference Atmospheric Model (Jupiter-GRAM) was developed by the Natural Environments Branch, Spacecraft and Vehicle Systems Department, Engineering Directorate of NASA Marshall Space Flight Center and the Atmospheric Flight and Entry Systems Branch at NASA Langley Research Center.

Information on obtaining Jupiter-GRAM code and data can be found on the NASA Software Catalog at: https://software.nasa.gov.

For technical, programmatic or policy questions, please contact

Hilary Justh
Planetary GRAM Lead
NASA MSFC EV44
Marshall Space Flight Center, AL 35812
Phone: (256) 544-3694
Fax: (256) 544-3060
E-mail: Hilary.L.Justh@nasa.gov

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ASCII	American Standard Code for Information Interchange
ASI	Atmospheric Structure Instrument
CSS	Cascading Style Sheets
CSV	comma separated value
ERT	Earth-receive time
GPMS	Galileo Probe Mass Spectrometer
GRAM	Global Reference Atmospheric Model
HAD	Helium Abundance Detector
Jupiter-GRAM	Jupiter Global Reference Atmospheric Model
LTST	local true solar time
MSFC	Marshall Space Flight Center
NAIF	Navigation and Ancillary Information Facility
NFR	Net Flux Radiometer
PET	planet event time
SMD	Science Mission Directorate
SPICE	Spacecraft Planet Instrument C-matrix Events

NOMENCLATURE

A frontal area

a_z Galileo probe deceleration

C_D drag coefficient

dp change in pressure

dt change in time

dh change in altitude

g acceleration due to gravity on Jupiter

L_s solar longitude

m Galileo probe mass

p pressure

p_1 initial value of pressure

R gas constant

R_u universal gas constant

T temperature

V_r Galileo probe velocity relative to the atmosphere

w_p probe descent velocity

γ ratio of specific heats

µ mean molecular weight

ρ density

TECHNICAL MEMORANDUM

JUPITER GLOBAL REFERENCE ATMOSPHERIC MODEL (JUPITER-GRAM): USER GUIDE

1. INTRODUCTION

1.1 Background and Overview

Engineers and mission planners designing vehicles that pass through Jupiter's atmosphere require an atmospheric model that calculates the mean values of atmospheric properties. The Jupiter Global Reference Atmospheric Model (Jupiter-GRAM) is an engineering-oriented model that provides this information based on data from Galileo probe observations. Jupiter-GRAM is designed to offer mission planners the flexibility to select input parameters such as time, latitude, and longitude. Jupiter-GRAM outputs mean values for atmospheric density, temperature, and pressure along a user defined path.

Jupiter-GRAM is one option in the GRAM suite that shares a common software core with the other planetary GRAMs while maintaining Jupiter specific models. Additionally, documentation (including this User Guide, a Programmer's Manual, and trajectory code interfaces) has been made available with the software release.

This Technical Memorandum summarizes the atmospheric data model in Jupiter-GRAM and provides a guide for the user to obtain, set up, and run the code in various configurations. Section 2 describes the atmospheric data files and how they are used in Jupiter-GRAM. Section 3 explains the process to obtain the Jupiter-GRAM code, the data files, and how to set up and run the program. Appendices A through E provide additional details regarding the Jupiter-GRAM input and output files. Appendix F provides a history of Jupiter-GRAM revisions.

1.2 Jupiter-GRAM Features

Jupiter-GRAM takes advantage of major code modifications made to the GRAMs to improve efficiencies in implementation, run time, and maintenance. Important features include:

(1) The incorporation of NASA's Navigation and Ancillary Information Facility (NAIF) Spacecraft Planet Instrument C-matrix Events (SPICE) library for ephemeris calculations. Jupiter ephemeris values, such as longitude of the Sun and solar time, are computed using the NAIF SPICE library for greater accuracy. The use of the NAIF SPICE library requires the Jupiter-GRAM user to download the latest SPICE data before using Jupiter-GRAM. Instructions for doing so are provided in section 3.2.

(2) The output is provided in two formats: (1) a comma separated value (CSV) file and (2) a LIST file (LIST.md). The CSV file consolidates the column formatted output files into a single file that can easily be loaded into data centric programs, such as Microsoft Excel or MATLAB®. A detailed list of CSV file parameters and definitions are provided in appendix A. Alternatively, the LIST file can be read using either a standard American Standard Code for Information

Interchange (ASCII) reader or a Markdown syntax for enhanced rendering in a web browser. An example of both LIST file formats is provided in appendix C.

(3) The GRAM Suite contains a library of data models and utilities that includes GRAM atmospheric data for multiple destinations. Refer to the GRAM Programmer's Manual for additional details.

(4) Jupiter-GRAM computes speed of sound based on a thermodynamic parameterization using density, pressure, and γ, the ratio of specific heats, for a given constituent gas mixture. Jupiter-GRAM does not include a constituent gas model and as a result utilizes a default constant value for γ which is based on the value of γ for hydrogen.

2. JUPITER-GRAM ATMOSPHERIC DATA

2.1 Jupiter-GRAM Atmospheric Data Description

Atmospheric density, temperature, and pressure as a function of altitude are characterized by data from Seiff et al.[1] This data was measured from 1029.2 km above to 132.40 km below the 1-bar pressure level by the Galileo probe's Atmospheric Structure Instrument (ASI)[2] as it entered and descended through the atmosphere of Jupiter on December 7, 1995. The ASI measurements occurred in two stages. The first set of ASI measurements were taken by the on-board accelerometers of the decelerations occurring along and normal to the z axis (the probe axis of symmetry in the direction of the probe velocity) during the probe's high-speed entry into the upper atmosphere of Jupiter. These measurements captured data from 1029.2 km to 23.3 km above the 1-bar level. The second set of ASI measurements began seconds prior to parachute deploy and captured pressure, temperature, and acceleration data for the lower atmosphere of Jupiter from 17.291 km above to 132.4 km below the 1-bar level. The Seiff et al.[1] model did not include wind data due to the lack of understanding of the winds at these altitudes, as a result Jupiter-GRAM does not include a wind model. Additionally, due to the lack of data, this version of Jupiter-GRAM does not include a perturbation or constituent gas model. These values will be updated in the model as they are available.

Planetary constants used in Jupiter-GRAM are from the Planetary Data System "Standard Planetary Information, Formulae and Constants" web page: http://atmos.nmsu.edu/jsdap/encyclopediawork.html. The gravitational parameter is from the JPL Solar System Dynamics Outer Planet Gravity Fields web page: https://ssd.jpl.nasa.gov/?gravity_fields_op. Table 1 provides the Jupiter gravity parameter data that are utilized in Jupiter-GRAM.

Table 1. Jupiter gravity parameters.

Jupiter	Label	Units	Value
Gravitational Parameter	GM	km^3/s^2	126686534.1960128
Mean Equatorial Radius	R_e	km	71492.0
Mean Polar Radius	R_p	km	66854.0
J2 harmonic	J_2	km^5/s^2	0.014736
Period		s	35730.0

2.1.1 Lower Atmosphere Jupiter-GRAM Data

Altitudes and decent velocities in the lower atmosphere of Jupiter, from 17.291 km above to 132.4 km below the 1-bar level, were defined by Seiff et al.[1] as functions of time from the ASI measured temperature and pressure. Seiff et al.[1] calculated the Galileo probe altitudes by assuming hydrostatic equilibrium,

$$dp = -\rho g\, dh = -\left(\frac{pg}{RT}\right) dh \qquad (1)$$

where:

dp = change in pressure
ρ = density
g = acceleration due to gravity on Jupiter
dh = change in altitude
p = pressure
R = gas constant
T = temperature

Equation (1) is then integrated using the ASI measured temperatures and pressures.

The Galileo probe descent velocities (w_p) were calculated by Seiff et al.[1] by writing the differentials in (1) as time derivatives:

$$w_p = \frac{dh}{dt} = -\left(\frac{RT}{pg}\right)\left(\frac{dp}{dt}\right) \tag{2}$$

where:

w_p = probe descent velocity
dh = change in altitude
dt = change in time
R = gas constant
T = temperature
p = pressure
g = acceleration due to gravity on Jupiter
dp = change in pressure

Atmospheric density was calculated by Seiff et al.[1] through utilization of the equation of state. The gas composition and gas constant were allowed to vary with the pressure level as long as their values were consistent with the measured He/H_2 abundance ratio of 0.136/0.864 = 0.1574 from the Galileo probe Helium Abundance Detector (HAD)[3], the minor constituent variations measured by the Galileo Probe Mass Spectrometer (GPMS)[4], and the NH_3 abundance profile determined by radio science[5] and the Galileo probe Net Flux Radiometer (NFR)[6].

2.1.2 Upper Atmosphere Jupiter-GRAM Data

Seiff et al.[1] derived the atmospheric densities for the upper atmosphere of Jupiter from the ASI measured accelerations using the following equation based on Newton's second law of motion and the Galileo probe aerodynamics:

$$\rho = 2\left(\frac{m}{C_D A}\right)\left(\frac{a_z}{{V_r}^2}\right) \tag{3}$$

where:

ρ = density
m = Galileo probe mass
C_D = drag coefficient

A = frontal area
a_z = Galileo probe deceleration in the z direction
V_r = Galileo probe velocity relative to the atmosphere

The Galileo probe velocity, V_r, and the altitude, h, were calculated as functions of time as a result of the Galileo probe trajectory reconstruction completed by Seiff et al.[1] that was based on the Galileo probe's measured accelerations and initial conditions.

The atmospheric pressures in the upper atmosphere were calculated by Seiff et al.[1] through the integration of equation (1) with the density values that resulted from equation (3). It should be noted that the initial value of pressure (p_1) at the first Galileo probe data point was unknown and was assumed by Seiff et al.[1] by choosing plausible initial temperatures and then calculating p_1.

The atmospheric temperatures in the upper atmosphere were calculated by Seiff et al.[1] from the densities and pressures using the equation of state:

$$T = \frac{p\mu}{\rho R_u} \tag{4}$$

where:

T = temperature
p = pressure
μ = mean molecular weight
ρ = density
R_u = universal gas constant

This calculation utilized mean molecular weight data from an upper atmosphere composition model that is detailed in Seiff et al.[1]

2.2 Querying Atmosphere Data

The Jupiter-GRAM user-defined path can be generated in multiple ways. The first is to run Jupiter-GRAM in standalone mode which uses an automated increment approach based on inputs specified in the NAMELIST input file for the initial time and position (e.g. *Year, Month, Day, Hour, Seconds, InitialHeight, InitialLatitude,* and *InitalLongitude*) and the deltas (e.g., *DeltaTime, DeltaHeight, DeltaLatitude*, and *DeltaLongitude*). Refer to section 3.3 for input parameter definitions and appendix B for a sample file. In standalone mode, Jupiter-GRAM steps automatically in user-defined increments of altitude, latitude, longitude, and time to generate a constantly incremented profile. Each point in the profile will have a corresponding atmospheric value for density, temperature, and pressure. A second path generation option is to run the model in trajectory evaluation mode where the user provides a trajectory file, specified using *TrajectoryFileName.* The trajectory file contains a specified time history of altitude, latitude, and longitude and removes the constant increment constraint criteria of the previous option. Additional information about trajectory file input can be found in section 2.5. A third method is to incorporate the Jupiter-GRAM code directly into a user's trajectory code. This version of Jupiter-GRAM contains both C and Fortran interfaces. The GRAM libraries can be incorporated directly in the user's trajectory (or orbit propagation) code for atmospheric evaluations along a trajectory or orbital positions. Documentation of the GRAM libraries, interfaces, and examples are provided in the GRAM Programmer's Manual.

Regardless of the path generation option selected, Jupiter-GRAM writes output to two files: a CSV output file and a LIST output file. These output files are detailed in appendices A and C.

2.3 Auxiliary Atmosphere Profile Option

The auxiliary atmosphere profile option provides the user the ability to overwrite the atmosphere model in Jupiter-GRAM with a profile of atmosphere quantities versus altitude (note: constituent data cannot be over-written using this option). This option is controlled by setting input parameters *AuxiliaryAtmosphereFileName, InnerRadius*, and *OuterRadius* in the NAMELIST input file. Each line of the auxiliary atmosphere profile input file must consist of: (1) height, in km, (2) latitude, in degrees, (3) longitude, in degrees, (4) temperature, in K, (5) pressure, in Pa, (6) density, in kg/m^3, (7) eastward wind, in m/s, and (8) northward wind, in m/s. Longitudes are east or west positive, as set by input parameter *EastLongitudePositive*. Standard Jupiter-GRAM input data for temperature, pressure, or density data are used if the auxiliary atmosphere profile inputs for temperature, pressure, or density are zero. Standard Jupiter-GRAM input wind data (currently zeros) are used if both wind components in the auxiliary atmosphere profile file are set to zero.

A weighting factor for the auxiliary atmosphere profile data (*ProfileWeight*), having values between 0 and 1, is applied between the *InnerRadius* and *OuterRadius*. The *InnerRadius* is the latitude-longitude radius (degrees) within which weight for the auxiliary atmosphere profile is 1.0 (e.g. the data in the auxiliary profile is used as provided). The *OuterRadius* is the latitude-longitude radius (degrees) beyond which the weight for the auxiliary atmosphere profile is 0.0 (e.g., the model uses standard Jupiter-GRAM data). Mean conditions are specified by the auxiliary atmospheric profile input file if the desired point is within the *InnerRadius*; mean conditions are given by the standard Jupiter-GRAM data if the desired point is beyond the *OuterRadius*. Linear interpolation of pressure and density occurs at each altitude increment between the *InnerRadius* and *OuterRadius*. An illustration of the fairing that occurs between the *InnerRadius* and *OuterRadius* is provided in figure 6. If *InnerRadius* = 0, then the auxiliary atmosphere profile data are not used. In addition to fairing in latitude and longitude, fairing of the auxiliary atmosphere profile altitude is performed. This only occurs at the beginning and end of the file. The profile weight factor (*ProfileWeight*) for the auxiliary atmosphere profile varies between 0 at the first auxiliary atmosphere profile altitude level and 1 at the second auxiliary atmosphere profile altitude level (and between 1 at the next-to-last auxiliary atmosphere profile altitude level and 0 at the last auxiliary atmosphere profile altitude level). Therefore, care must be taken when selecting the altitude spacing at the beginning and end of the auxiliary atmosphere profile (e.g., selected to be far enough apart in altitude) to ensure that a smooth transition occurs as *ProfileWeight* changes from 0 to 1 near these auxiliary atmosphere profile beginning and end points.

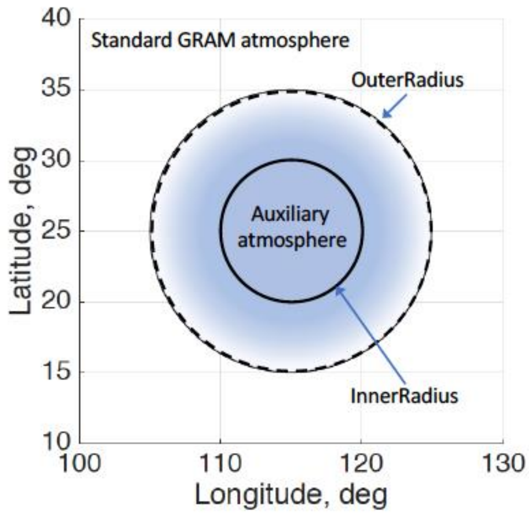

Figure 1. Illustration of two-dimensional auxiliary profile faring implementation with *InnerRadius* = 5° and *OuterRadius* = 10° for a vertical auxiliary profile located at latitude = 25° and longitude = 115°.

2.4 Trajectory File Input

The trajectory file is only utilized when a trajectory, rather than an automatically determined profile, is desired.

To utilize a trajectory file in a Jupiter-GRAM run, simply assign the desired trajectory file name to the NAMELIST variable *TrajectoryFileName*. The trajectory file may contain an unlimited number of individual list-directed (free-field) records, or lines, consisting of four real values:

(1) Time (s) past the start time specified in the NAMELIST input.
(2) Height (km).
(3) Latitude (± 90°, with southern latitudes being negative).
(4) Longitude (± 360°, with positive longitude designated by the input parameter *EastLongitudePositive*).

Any additional information included on each line of the trajectory file (e.g. orbit number, measured density, etc.) is ignored. Trajectory increments in these files do not have to be at small time or space steps. For example, a trajectory file may consist of successive periapsis times and positions for a simulated or observed aerobraking operation. Trajectory files may also contain arrays of locations used for computing height-latitude cross sections or latitude-longitude cross sections.

3. HOW TO RUN JUPITER-GRAM

3.1 How to Obtain the Program

Jupiter-GRAM is available through the NASA Software Catalog: https://software.nasa.gov. The software is offered free of charge. See appendices D and E for summaries of the program and data files available in the downloaded package.

3.2 Running the Program

The Jupiter-GRAM installation includes a set of Windows and Linux 64-bit executable libraries located in the GRAM/Windows and GRAM/Linux folders. The Jupiter-GRAM programs in these folders may be relocated to any folder on the appropriate operating system. For those wishing to build their own executables or those running on another operating system, build instructions are provided in appendix E.

Before running Jupiter-GRAM, the NAIF SPICE data files must be downloaded. These data are available via FTP from ftp://naif.jpl.nasa.gov/pub/naif/generic_kernels. Information about the SPICE data is available from https://naif.jpl.nasa.gov/naif/data.html and help downloading is available from https://naif.jpl.nasa.gov/naif/download_tip.html. NAIF recommends that the entire collection be downloaded, but these files can be rather large. The files required by Jupiter-GRAM are listed in boldface below. They should be downloaded using the same folder structure as on the NAIF site.

```
/spice    (FTP source folder is /generic_kernels)
└──────/lsk    (entire folder, less than 100KB)
        └──────/naif0012.tls    (time data, all GRAMs)
└──────/pck  (entire folder except for a_old_versions, about 27MB)
        └──────/pck00010.tpc    (planetary size/shape data, all GRAMs)
└──────/spk  (massive, consider getting subfolders only)
           └──────/planets    (entire folder except for a_old_versions, about
                                3.3GB)
                └──────/de430.bsp    (Venus-GRAM)
        └──────/satellites    (entire folder except for a_old_versions, about
                               5.8GB)
                └──────/jup310.bsp    (Jupiter-GRAM)
                └──────/mar097.bsp    (Mars-GRAM)
                └──────/nep081.bsp    (Neptune-GRAM)
                └──────/sat375.bsp    (Saturn-GRAM, Titan-GRAM)
                └──────/ura111.bsp    (Uranus-GRAM)
```

The default location of the SPICE data files is in the root folder, /spice, on the current disk. If another location is desired, then be certain to set the *SpicePath* input parameter in the NAMELIST file to the desired location.

To run Jupiter-GRAM, simply double-click the JupiterGRAM.exe file or enter 'JupiterGRAM.exe' from a command prompt. The program will prompt for the path to an input parameter file in NAMELIST format (see section 3.3). The path may be entered as an absolute path or relative to the current folder. Sample input parameter files, ref_input.txt and traj_input.txt, can be found in the /GRAM/Jupiter/sample_inputs folder. Both files are plain text and can be viewed in a text editor, such as WordPad, with no word wrapping. On exit, the program will name the output files generated. In this case, they will be myref_LIST.md and myref_OUTPUT.csv. The myref_OUTPUT.csv file is best viewed using a spreadsheet program such as Microsoft Excel. See appendix C for optional methods for viewing the myref_LIST.md markdown file. Appendix C also shows examples of the myref_LIST.md output. The input parameter file may also be specified on the Jupiter-GRAM command line. The format of this option is 'JupiterGRAM.exe –file ref_input.txt'. The sample_inputs folder contains pregenerated outputs ref_LIST.md and ref_OUTPUT.csv. These files are provided so that users may compare their output with the expected output.

3.3 Program Input

Jupiter-GRAM requires an input file in the format of a Fortran NAMELIST file. Appendix B gives a sample of the NAMELIST format input file for Jupiter-GRAM. All input parameter names are case insensitive. Input parameters whose values are supplied in the input file are given in table 2. (The legacy GRAM input parameters names are still supported and appear in parentheses.)

Table 2. Jupiter-GRAM input parameters.

Input Parameter	Description	Default
	File Path and Names	
SpicePath or SpiceDir	The location of the NAIF SPICE data files. Absolute paths are recommended. Relative paths are acceptable.	/spice
ListFileName (LSTFL)	Name of list formatted file with no file extension. The appropriate file extension will be appended to this name. An example of a LIST file is given in appendix C.	LIST
ColumnFileName (OUTFL)	Name of the column formatted file with no file extension. The appropriate file extension will be appended to this name. A complete description of this file is contained in appendix A.	OUTPUT
TrajectoryFileName (TRAJFL)	(Optional) The trajectory input file name. This file contains time (seconds) relative to start time, height (km), latitude (degrees), and longitude (degrees, see below).	<empty>
	Time Parameters	
TimeFrame (IERT)	Sets the time frame for the start time. 1 for Earth-receive time (ERT) 0 for planet event time (PET)	1
TimeScale (IUTC)	Sets the time scale for the start time. 0 for Terrestrial Dynamical Time (TDT). 1 for Coordinated Universal Time (UTC). 2 for Barycentric Dynamical Time (TDB).	1
Year (MYEAR)	Integer year for the start time. Typically, a 4-digit year. Alternately, years 1970 - 2069 can be input as a 2-digit number.	2000
Month	Integer month (1 through 12) for the start time.	1

Day (MDAY)	Integer day of month for the start time.	1
Hour (IHOUR, IHR)	Integer hour (0 through 23) for the start time in the chosen *TimeScale* and *TimeFrame*.	0
Minute (IMIN)	Integer minute (0 through 59) for the start time in the chosen *TimeScale* and *TimeFrame*.	0
Seconds (SEC)	Real seconds (less than 60.0) for the start time in the chosen *TimeScale* and *TimeFrame*.	0.0
Trajectory Parameters		
EastLongitudePositive (LONEAST)	This flag controls the convention for input and output of longitudes. East positive convention if *EastLongitudePositive* = 1. West positive convention if *EastLongitudePositive* = 0.	1
NumberOfPositions (NPOS)	The number of positions to generate and evaluate, if an automatically generated profile is to be produced. This parameter is ignored if a *TrajectoryFileName* is provided.	21
InitialHeight (FHGT)	Height (km) of the initial position.	0.0
InitialLatitude (FLAT)	Latitude (degrees, north positive) of the initial position.	0.0
InitialLongitude (FLON)	Longitude (degrees) of the initial position. The direction of positive longitudes is determined by the *EastLongitudePositive* parameter.	0.0
DeltaHeight (DELHGT)	Height increment (km) between successive steps in an automatically generated profile (positive upward).	10.0
DeltaLatitude (DELLAT)	Latitude increment (degrees, north positive) between successive steps in an automatically generated profile.	0.0
DeltaLongitude (DELLON)	Longitude increment (degrees) between successive steps in an automatically generated profile. The direction of positive longitudes is determined by the *EastLongitudePositive* parameter.	0.0
DeltaTime (DELTIME)	Time increment (seconds) between steps in an automatically generated profile.	0.0
Auxiliary Atmosphere Parameters		
AuxiliaryAtmosphereFileName (PROFILE)	(Optional) Input file name of the profile data for the auxiliary atmosphere.	<empty>
InnerRadius (PROFNEAR)	(Optional) Latitude-longitude radius (degrees) within which weight for the auxiliary profile is 1.0 (A value of 0.0 implies no auxiliary atmosphere data is present.)	0.0
OuterRadius (PROFFAR)	(Optional) Latitude-longitude radius (degrees) beyond which weight for the auxiliary profile is 0.0.	0.0
Output Parameters		
FastModeOn	Controls the speed and accuracy of ephemeris calculations. 0: More accurate, but slower. 1: Faster, but less accurate.	0

ExtraPrecision	For the new column output format, this parameter adds precision to all outputs.	0

3.4 Program Output

There are two general types of program output provided by Jupiter-GRAM. The first output file is a listing format with the file name specified by input parameter *ListFileName*. This file contains header and descriptor information which is suitable for printing or viewing by an analyst. The list file is output using a Markdown format. Markdown is a lightweight markup language that is designed to be readable in plain text format and offers improved formatting when converted to other file formats (typically html). Markdown viewer apps are available on all platforms. While not yet natively supported, most web browsers offer an extension/add-on that adds the Markdown capability. Markdown viewing options and an example of the list output file format are given in appendix C.

The second output file is in a CSV format with the file name specified by the input parameter *ColumnFileName*. This file contains one header line and one line per output position and is suitable for reading into another program for additional analysis. The precision of the outputs can be increased using the input parameter *ExtraPrecision*. The CSV format can be easily loaded into most spreadsheet programs. It can also be imported into programs, such as MATLAB®, for analysis. A description of each of the output fields in the CSV file format can be found in appendix A.

3.5 Reference Test Run

The Jupiter-GRAM distribution includes sample files ref_input.txt and traj_input.txt for application in a reference test run. To verify the Jupiter-GRAM build, execute *JupiterGRAM.exe* using ref_input.txt as the input parameter file. The files myref_LIST.md and myref_OUTPUT.csv, generated during the test run, should be identical to the supplied ref_LIST.md and ref_OUTPUT.csv files.

3.6 FindDates Utility

Jupiter-GRAM gives the user the option to find the date and time for a particular solar longitude (L_s) and Jupiter local true solar time (LTST) through the FindDates utility. It also computes the Earth date and time of the next closest occurrence to the initial input date and time for which L_s and LTST are the user desired values. The SPICE data are required for this capability. The FindDates capability is contained within the Jupiter-GRAM program and is controlled by the *FindDates* input parameter (see table 3). The utility will return three dates and times: the date and time of the target L_s and the two dates and times of the target LTST that immediately precede and follow the target L_s date. A sample FindDates input file can be found in the sample_inputs file.

Table 3. FindDates input parameters.

Input Parameter	Description	Default
SpicePath or SpiceDir	The location of the NAIF SPICE data files. Absolute paths are recommended. Relative paths are acceptable.	/spice
FindDates	The parameter flags the use of the FindDates auxiliary capability. Use the FindDates capability if *FindDates* = 1. Use Jupiter-GRAM if *FindDates* = 0.	0
EastLongitudePositive (LONEAST)	This flag controls the convention for input and output of longitudes. East positive convention if *EastLongitudePositive* = 1. West positive convention if *EastLongitudePositive* = 0.	1
Time Parameters		
TimeFrame (IERT)	Sets the time frame for the start time. 1 for Earth-receive time (ERT) 0 for planet event time (PET)	1
TimeScale (IUTC)	Sets the time scale for the start time. 0 for Terrestrial Dynamical Time (TDT) 1 for Coordinated Universal Time (UTC) 2 for Barycentric Dynamical Time (TDB)	1
Year (MYEAR)	Integer year for the start time. Typically, a 4-digit year. Alternately, years 1970 - 2069 can be input as a 2-digit number.	2000
Month	Integer month (1 through 12) for the start time.	1
Day (MDAY)	Integer day of month for the start time.	1
Hour (IHOUR, IHR)	Integer hour (0 through 23) for the start time in the chosen *TimeScale* and *TimeFrame*.	0
Minute (IMIN)	Integer minute (0 through 59) for the start time in the chosen *TimeScale* and *TimeFrame*.	0
Seconds (SEC)	Real seconds (less than 60.0) for the start time in the chosen *TimeScale* and *TimeFrame*.	0.0
Position Parameters		
InitialHeight (FHGT)	Height (km) of the initial position.	0.0
InitialLatitude (FLAT)	Latitude (degrees, North positive) of the initial position.	0.0
InitialLongitude (FLON)	Longitude (degrees) of the initial position. The direction of positive longitudes is determined by the *EastLongitudePositive* parameter.	0.0
FindDates Parameters		
TargetLongitudeSun	The desired longitude of the sun in degrees.	0.0
TargetSolarTime	The desired true local solar time in hours (0 to 24).	0.0

APPENDIX A – HEADERS FOR JUPITER-GRAM OUTPUT FILE

Jupiter-GRAM produces a CSV output file (see table 4) suitable for passing to a data-centric program for plotting and further analysis. The field names purposely lack any special characters other than an underscore separating the units. Thus, for some fields, such as Gravity_ms2, the precise units must be inferred, as in m/s^2.

The current version of Jupiter-GRAM does not have a winds model, perturbation model, or constituent gas model. To maintain uniformity with the other GRAM models, wind outputs, perturbation outputs, the number density of the atmosphere, the average molecular weight of the atmosphere, and compressibility factor will be included. As noted below, some of these outputs will be zero.

Table 4. OUTPUT.csv (or as prescribed in the *ColumnFileName* input parameter).

Time_s	Seconds past the start time
Height_km	Height above the reference ellipsoid
Latitude_deg	Geocentric latitude
LongitudeE_deg LongitudeW_deg	East (or west) longitude, as controlled by input value *EastLongitudePositive*
TotalRadius_km	Radial distance from planetary center of mass to the current position (latitude radius plus altitude)
LatitudeRadius_km	Planetary radius at current latitude.
Gravity_ms2	Local acceleration of gravity (m/s^2)
Temperature_K	Mean temperature (K)
Pressure_Nm2	Mean pressure (Pa)
Density_kgm3	Mean density (kg/m^3)
PressureScaleHeight_km	The height range over which pressure decreases by a factor of *e*
DensityScaleHeight_km	The height range over which density decreases by a factor of *e*
SpeedOfSound_ms	The speed of sound (m/s)
PressureAtSurface_Nm2	Pressure at the zero altitude surface (Pa)
SigmaLevel	The ratio of pressure to pressure at the surface.
PressureAltitude_km	Pressure altitude
ReferenceTemperature_K	Temperature of the reference atmosphere
ReferencePressure_Nm2	Pressure of the reference atmosphere (N/m^2)
ReferenceDensity_kgm3	Density of the reference atmosphere (kg/m^3)
ProfileWeight	Weight factor for auxiliary input profile data
LowDensity_kgm3	Mean density - 1 standard deviation (kg/m^3)
HighDensity_kgm3	Mean density + 1 standard deviation (kg/m^3)
PerturbedDensity_kgm3*	Mean density + density perturbation (kg/m^3) *** Note: The current version of Jupiter-GRAM does not have a perturbation model. This value will be equal to the mean density.**
DensityPerturbation_pct*	Density perturbation (kg/m^3) *** Note: The current version of Jupiter-GRAM does not have a perturbation model. This value will be zero.**

DensityStandardDeviation_kgm3*	Standard deviation of the density (kg/m^3) *** Note: The current version of Jupiter-GRAM does not have a perturbation model. This value will be zero.**
PerturbedSpeedOfSound_ms*	The speed of sound at the current perturbed density (m/s) *** Note: The current version of Jupiter-GRAM does not have a perturbation model. This value will be equal to the speed of sound.**
RelativeStepSize	Fraction of minimum step size for accuracy of perturbations (should be > 1 for insured accuracy of perturbations)
DensityDeviation_pct*	Percent deviation of the mean density from the reference density *** Note: The current version of Jupiter-GRAM does not have a perturbation model. This value will be zero.**
LowDensityDeviation_pct*	Percent deviation of the low density from the reference density *** Note: The current version of Jupiter-GRAM does not have a perturbation model. This value will be zero.**
HighDensityDeviation_pct*	Percent deviation of the high density from the reference density *** Note: The current version of Jupiter-GRAM does not have a perturbation model. This value will be zero.**
PerturbedDensityDeviation_pct*	Percent deviation of the perturbed density from the reference density *** Note: The current version of Jupiter-GRAM does not have a perturbation model. This value will be zero.**
EWWind_ms*****	Mean eastward wind component (m/s) *** Note: The current version of Jupiter-GRAM does not have a winds model. This value will be zero.**
NSWind_ms*****	Mean northward wind component (m/s) *** Note: The current version of Jupiter-GRAM does not have a winds model. This value will be zero.**
EWWindPerturbation_ms*****	Eastward wind perturbation (m/s) *** Note: The current version of Jupiter-GRAM does not have a winds model. This value will be zero.**
NSWindPerturbation_ms*****	Northward wind perturbation (m/s) *** Note: The current version of Jupiter-GRAM does not have a winds model. This value will be zero.**
PerturbedEWWind_ms*****	Total (mean plus perturbed) eastward wind (m/s) *** Note: The current version of Jupiter-GRAM does not have a winds model. This value will be zero.**
PerturbedNSWind_ms*****	Total (mean plus perturbed) northward wind (m/s) *** Note: The current version of Jupiter-GRAM does not have a winds model. This value will be zero.**
EWStandardDeviation_ms*****	Standard deviation of eastward wind perturbations (m/s) *** Note: The current version of Jupiter-GRAM does not have a winds model. This value will be zero.**
NSStandardDeviation_ms*****	Standard deviation of northward wind perturbations (m/s) *** Note: The current version of Jupiter-GRAM does not have a winds model. This value will be zero.**
LongitudeOfTheSun_deg	The planetocentric longitude of the sun, L_s
SubsolarLatitude_deg	The latitude of the sub-solar point at the current time

SubsolarLongitudeE_deg SubsolarLongitudeW_deg	The longitude of the sub-solar point at the current time. East positive or west positive as controlled by the input value *EastLongitudePositive*
LocalSolarTime_hr	The local solar time using 24 “hour” intervals
SolarZenithAngle_deg	The solar zenith angle
OneWayLightTime_min	One way light time to/from Earth and the current position
OrbitalRadius_AU	The current orbital radius of the planet
SecondsPerSol	The number of seconds in a local sol (planetary day)
TotalNumberDensity_m3*	Number density of the atmosphere (#/m^3) *** Note: The current version of Jupiter-GRAM does not have a constituent gas model. This value will be zero.**
AverageMolecularWeight*	Average molecular weight of the atmosphere (amu) *** Note: The current version of Jupiter-GRAM does not have a constituent gas model. This value will be zero.**
CompressibilityFactor*	Compressibility factor (or zeta). This quantifies the deviation of a real gas from ideal gas behavior (zeta = 1 for ideal gases). *** Note: The current version of Jupiter-GRAM does not have a constituent gas model. This value will be zero.**
SpecificGasConstant_JkgK	Specific gas constant (J/(kg K))
SpecificHeatRatio*	Specific heat ratio of the gas mixture. *** Note: The current version of Jupiter-GRAM does not have a constituent gas model. This value is set to the value for hydrogen.**

APPENDIX B – EXAMPLE NAMELIST FORMAT INPUT FILE

The following is an example of the NAMELIST format input file required by Jupiter-GRAM. Input data given here are provided as file ref_input.txt. Values given are the default values assigned by the program. Only values that differ from the defaults actually have to be included in the NAMELIST file.

```
$INPUT
  SpicePath              = '\spice'
  ListFileName           = 'myref_LIST'
  ColumnFileName         = 'myref_OUTPUT'
  EastLongitudePositive = 1

  TimeFrame = 1
  TimeScale = 1
  Month     = 3
  Day       = 25
  Year      = 2020
  Hour      = 12
  Minute    = 30
  Seconds   = 0.0

  TrajectoryFileName = 'null'
  NumberOfPositions  = 201
  InitialHeight      = 0.0
  InitialLatitude    = 22.0
  InitialLongitude   = 48.0
  DeltaHeight        = 5.0
  DeltaLatitude      = 0.3
  DeltaLongitude     = 0.5
  DeltaTime          = 500.0

  AuxiliaryAtmosphereFileName = 'null'
  InnerRadius = 0.0
  OuterRadius = 0.0

  FastModeOn        = 0
  ExtraPrecision    = 0

 $END

 Explanation of variables:
  SpicePath         =  Path to NAIF Spice data
  ListFileName      =  List file name
  ColumnFileName      =  Output file name
  EastLongitudePositive =  0 for input and output West longitudes positive
                           1 for East longitudes positive

  TimeFrame = 0 Planet event time (PET)
              1 for time input as Earth-receive time (ERT)

  TimeScale = 0 for Terrestrial (Dynamical) Time (TDT)
              1 for time input as Coordinated Universal Time (UTC)
              2 for Barycentric Dynamical Time (TDB)
  Month     = month of year
  Day       = day of month
  Year      = year (4-digit, or 1970-2069 can be 2-digit)
  Hour      = hour of day (meaning controlled by TimeFrame and TimeScale)
  Minute    = minute of hour (meaning controlled by TimeFrame and TimeScale)
  Seconds   = seconds of minute (meaning controlled by TimeFrame and TimeScale)
```

```
TrajectoryFileName = (Optional) Trajectory input file name
                     If present, then the values below are ignored
NumberOfPositions  = number of positions to evaluate
InitialHeight      = initial height (km)
InitialLatitude    = initial latitude (N positive), degrees
InitialLongitude   = initial longitude, degrees
                     (depends on EastLongitudePositive)
DeltaHeight        = height increment (km) between steps
DeltaLatitude      = latitude increment (deg) between steps
DeltaLongitude     = longitude increment (deg) between steps
                     (depends on EastLongitudePositive)
DeltaTime          = time increment (seconds) between steps.


AuxiliaryAtmosphereFileName = (Optional) auxiliary profile input file name
InnerRadius = Lat-lon radius within which weight for auxiliary profile is 1.0
              (Use InnerRadius = 0.0 for no profile input)
OuterRadius = Lat-lon radius beyond which weight for auxiliary profile is 0.0

FastModeOn       = Flags use of faster ephemeris computations (less accurate)
                    0 Most accurate ephemeris computations are used
                    1 Faster computations with slight loss in accuracy
ExtraPrecision    = For the new column output format, this parameter
                    adds precision to all outputs.
```

APPENDIX C – SAMPLE OUTPUT LIST FILE

Following is a portion of the LIST file output produced by the standard input parameters given in appendix B. The output data given below are provided in the file ref_LIST.md. This file allows users to complete a test run after compiling Jupiter-GRAM on their own computer and to electronically check their output by a file-compare process (e.g. the ‘diff’ command in UNIX or the ‘fc’ command from a Windows Command Prompt). Please note that, due to machine-dependent or compiler-dependent rounding differences, some output values may differ slightly from those shown here. These differences are usually no more than one unit in the last significant digit displayed.

The current version of Jupiter-GRAM does not have a winds model, perturbation model, or constituent gas model. To maintain uniformity with the other GRAM models, wind, perturbation, and gasses outputs will be included. As seen in the sample LIST file output included below, some of these outputs will be zero.

```
| Field             | Value                             | Field                        | Value       |
|-------------------|-----------------------------------|------------------------------|-------------|
| Time Frame        | Earth Receive Time (ERT)          | Initial Random Seed          | 1001        |
| Time Scale        | Coordinated Universal Time (UTC)  | Minimum Relative Step Size   | 0.000       |
| Start Date        | 3/25/2020                         | Density Perturbation Scale   | 1.00        |
| Start Time        | 12:30:00.00                       | EW Wind Perturbation Scale   | 1.00        |
| Julian Day        | 2458934.020833                    | NS Wind Perturbation Scale   | 1.00        |

--------------------
## Record #1
--------------------

| Field                              | Value      | Field                              | Value      |
|------------------------------------|------------|------------------------------------|------------|
| Elapsed Time (s)                   | 0.00       | Elapsed Time (sols)                | 0.00       |
| Height Above Ref. Ellipsoid (km)   | 0.000      | Reference Radius (km)              | 70782.6    |
| Latitude (deg)                     | 22.000     | Local Solar Time (hrs)             | 20.70      |
| Longitude E (deg)                  | 48.00      | Longitude of the Sun (deg)         | 325.53     |
| Pressure Scale Height (km)         | 25.331     | Orbital Radius (AU)                | 5.20       |
| Density Scale Height (km)          | -38.082    | One Way Light Time (min)           | 45.08      |
| Temperature (K)                    | 166.1      | Subsolar Latitude (deg)            | -1.54      |
| Pressure (Pa)                      | 1.000e+05  | Subsolar Longitude E (deg)         | 277.52     |
| Sigma Level                        | 1.000      | Solar Zenith Angle (km)            | 127.02     |
| Pressure Altitude (km)             | -0.000     | Gravity (m/s^2)                    | 23.734     |
| Surface Pressure (Pa)              | 1.000e+05  | Speed of Sound (m/s)               | 900.217    |
| Compressibility Factor (zeta)      | 0.0000     | Specific Gas Constant (J/(kg K))   | 3460.239   |
| Specific Heat Ratio                | 1.410      | Profile Weight                     | 0.000      |

| Density                            | Low        | Average                            | High       |
|------------------------------------|------------|------------------------------------|------------|
| Density (kg/m^3)                   | 1.7399e-01 | 1.7399e-01                         | 1.7399e-01 |
| Density Deviation (%)              | 0.0        | 0.0                                | 0.0        |
| Perturbed Density (kg/m^3)         | 1.7399e-01 | Perturbation (%)                   | 0.0        |
| Perturbed Density Deviation (%)    | 0.00       | Perturbed Speed of Sound (m/s)     | 900.22     |

| Winds                              | Mean          | Perturbation  | Perturbed     |
|------------------------------------|---------------|---------------|---------------|
| Eastward Wind  (m/s)               | 0.0           | -0.0          | 0.0           |
| Northward Wind  (m/s)              | 0.0           | -0.0          | 0.0           |

| Gases                 | Number Density (#/m^3) | Mass (%) | Mole (%) | Avg Mol Wgt | Cp (J/gK)   |
|-----------------------|------------------------|----------|----------|-------------|-------------|
| Total                 | 0.0000e+00             | 0.0      | 0.0      | 0.00        | 0.00        |

--------------------
## Record #2
--------------------

| Field                              | Value      | Field                              | Value      |
|------------------------------------|------------|------------------------------------|------------|
| Elapsed Time (s)                   | 500.00     | Elapsed Time (sols)                | 0.01       |
| Height Above Ref. Ellipsoid (km)   | 5.000      | Reference Radius (km)              | 70764.3    |
| Latitude (deg)                     | 22.300     | Local Solar Time (hrs)             | 21.07      |
| Longitude E (deg)                  | 48.50      | Longitude of the Sun (deg)         | 325.53     |
| Pressure Scale Height (km)         | 23.653     | Orbital Radius (AU)                | 5.20       |
| Density Scale Height (km)          | 35.049     | One Way Light Time (min)           | 45.08      |
```

Temperature (K)	155.5	Subsolar Latitude (deg)	-1.54
Pressure (Pa)	8.151e+04	Subsolar Longitude E (deg)	272.48
Sigma Level	0.815	Solar Zenith Angle (km)	131.63
Pressure Altitude (km)	4.837	Gravity (m/s^2)	23.746
Surface Pressure (Pa)	1.000e+05	Speed of Sound (m/s)	880.947
Compressibility Factor (zeta)	0.0000	Specific Gas Constant (J/(kg K))	3539.539
Specific Heat Ratio	1.410	Profile Weight	0.000

Density	Low	Average	High
Density (kg/m^3)	1.4808e-01	1.4808e-01	1.4808e-01
Density Deviation (%)	0.0	0.0	0.0
Perturbed Density (kg/m^3)	1.4808e-01	Perturbation (%)	0.0
Perturbed Density Deviation (%)	0.00	Perturbed Speed of Sound (m/s)	880.95

Winds	Mean	Perturbation	Perturbed
Eastward Wind (m/s)	0.0	0.0	0.0
Northward Wind (m/s)	0.0	-0.0	0.0

Gases	Number Density (#/m^3)	Mass (%)	Mole (%)	Avg Mol Wgt	Cp (J/gK)
Total	0.0000e+00	0.0	0.0	0.00	0.00

(Snipped for brevity)

Record #200

Field	Value	Field	Value
Elapsed Time (s)	99500.00	Elapsed Time (sols)	2.78
Height Above Ref. Ellipsoid (km)	995.000	Reference Radius (km)	66941.6
Latitude (deg)	81.700	Local Solar Time (hrs)	22.17
Longitude E (deg)	147.50	Longitude of the Sun (deg)	325.63
Pressure Scale Height (km)	163.908	Orbital Radius (AU)	5.20
Density Scale Height (km)	144.770	One Way Light Time (min)	44.94
Temperature (K)	893.9	Subsolar Latitude (deg)	-1.54
Pressure (Pa)	1.184e-04	Subsolar Longitude E (deg)	355.00
Sigma Level	0.000	Solar Zenith Angle (km)	98.21
Pressure Altitude (km)	3369.066	Gravity (m/s^2)	26.103
Surface Pressure (Pa)	1.000e+05	Speed of Sound (m/s)	2288.190
Compressibility Factor (zeta)	0.0000	Specific Gas Constant (J/(kg K))	4154.183
Specific Heat Ratio	1.410	Profile Weight	0.000

Density	Low	Average	High
Density (kg/m^3)	3.1877e-11	3.1877e-11	3.1877e-11
Density Deviation (%)	0.0	0.0	0.0
Perturbed Density (kg/m^3)	3.1877e-11	Perturbation (%)	0.0
Perturbed Density Deviation (%)	0.00	Perturbed Speed of Sound (m/s)	2288.19

Winds	Mean	Perturbation	Perturbed
Eastward Wind (m/s)	0.0	-0.0	0.0
Northward Wind (m/s)	0.0	0.0	0.0

Gases	Number Density (#/m^3)	Mass (%)	Mole (%)	Avg Mol Wgt	Cp (J/gK)
Total	0.0000e+00	0.0	0.0	0.00	0.00

Record #201

Field	Value	Field	Value
Elapsed Time (s)	100000.00	Elapsed Time (sols)	2.80
Height Above Ref. Ellipsoid (km)	1000.000	Reference Radius (km)	66935.4
Latitude (deg)	82.000	Local Solar Time (hrs)	22.54
Longitude E (deg)	148.00	Longitude of the Sun (deg)	325.63
Pressure Scale Height (km)	161.569	Orbital Radius (AU)	5.20
Density Scale Height (km)	148.528	One Way Light Time (min)	44.94
Temperature (K)	896.9	Subsolar Latitude (deg)	-1.54
Pressure (Pa)	1.148e-04	Subsolar Longitude E (deg)	349.96
Sigma Level	0.000	Solar Zenith Angle (km)	98.26
Pressure Altitude (km)	3325.952	Gravity (m/s^2)	26.104
Surface Pressure (Pa)	1.000e+05	Speed of Sound (m/s)	2291.834
Compressibility Factor (zeta)	0.0000	Specific Gas Constant (J/(kg K))	4153.310
Specific Heat Ratio	1.410	Profile Weight	0.000

```
| Density                           | Low        | Average                           | High       |
|-----------------------------------|------------|-----------------------------------|------------|
| Density (kg/m^3)                  | 3.0813e-11 | 3.0813e-11                        | 3.0813e-11 |
| Density Deviation (%)             | 0.0        | 0.0                               | 0.0        |
| Perturbed Density (kg/m^3)        | 3.0813e-11 | Perturbation (%)                  | 0.0        |
| Perturbed Density Deviation (%)   | 0.00       | Perturbed Speed of Sound (m/s)    | 2291.83    |

| Winds                             | Mean          | Perturbation  | Perturbed     |
|-----------------------------------|---------------|---------------|---------------|
| Eastward Wind  (m/s)              | 0.0           | 0.0           | 0.0           |
| Northward Wind  (m/s)             | 0.0           | -0.0          | 0.0           |

| Gases                | Number Density (#/m^3) | Mass (%) | Mole (%) | Avg Mol Wgt | Cp (J/gK)   |
|----------------------|------------------------|----------|----------|-------------|-------------|
| Total                | 0.0000e+00             | 0.0      | 0.0      | 0.00        | 0.00        |

--------------------
## End of data
--------------------
```

The list file is formatted using the Markdown syntax. The file can also be displayed using a Markdown viewer. A sample of the Markdown output is shown below. Most web browsers support Markdown via extensions/add-ons or through online Markdown editors. The ‘Markdown Viewer’ extension is suggested for Chrome and the ‘Markdown Viewer Webext’ works well in Firefox. Installable Markdown viewers are available on all platforms. On Windows, the Notepad++ application has a ‘Markdown++’ plugin which displays Markdown with exports to html or pdf formats. For command line users, Pandoc will convert Markdown (use –f gfm) to a host of familiar rich text formats. The example below used Pandoc to convert Markdown to Open Document format.

Field	**Value**	**Field**	**Value**
Time Frame	Earth Receive Time (ERT)	Initial Random Seed	1001
Time Scale	Coordinated Universal Time (UTC)	Minimum Relative Step Size	0.000
Start Date	3/25/2020	Density Perturbation Scale	1.00
Start Time	12:30:00.00	EW Wind Perturbation Scale	1.00
Julian Day	2458934.020833	NS Wind Perturbation Scale	1.00

Record #1

Field	**Value**	**Field**	**Value**
Elapsed Time (s)	0.00	Elapsed Time (sols)	0.00
Height Above Ref. Ellipsoid (km)	0.000	Reference Radius (km)	70782.6
Latitude (deg)	22.000	Local Solar Time (hrs)	20.70
Longitude E (deg)	48.00	Longitude of the Sun (deg)	325.53
Pressure Scale Height (km)	25.331	Orbital Radius (AU)	5.20
Density Scale Height (km)	-38.082	One Way Light Time (min)	45.08
Temperature (K)	166.1	Subsolar Latitude (deg)	-1.54
Pressure (Pa)	1.000e+05	Subsolar Longitude E (deg)	277.52
Sigma Level	1.000	Solar Zenith Angle (km)	127.02
Pressure Altitude (km)	-0.000	Gravity (m/s^2)	23.734

Surface Pressure (Pa)	1.000e+05	Speed of Sound (m/s)	900.217
Compressibility Factor (zeta)	0.0000	Specific Gas Constant (J/(kg K))	3460.239
Specific Heat Ratio	1.410	Profile Weight	0.000

Density	**Low**	**Average**	**High**
Density (kg/m^3)	1.7399e-01	1.7399e-01	1.7399e-01
Density Deviation (%)	0.0	0.0	0.0
Perturbed Density (kg/m^3)	1.7399e-01	Perturbation (%)	0.0
Perturbed Density Deviation (%)	0.00	Perturbed Speed of Sound (m/s)	900.22

Winds	**Mean**	**Perturbation**	**Perturbed**
Eastward Wind (m/s)	0.0	-0.0	0.0
Northward Wind (m/s)	0.0	-0.0	0.0

Gases	**Number Density (#/m^3)**	**Mass (%)**	**Mole (%)**	**Avg Mol Wgt**	**Cp (J/gK)**
Total	0.0000e+00	0.0	0.0	0.00	0.00

Many of the Markdown viewers allow customization of the table formats using Cascading Style Sheets (CSS). The following CSS snippet will give the table layout a nice look and feel. Search the options of the Markdown viewer for custom CSS.

```
table {
  width: 100%;
  margin-top: 10px;
  border-collapse: collapse; }
table tr {
  border-top: 1px solid silver;
  background-color: white; }
table tr:nth-child(2n) {
  background-color: whitesmoke; }
table tr th {
  font-weight: bold;
  border: 1px solid silver;
  background-color: lightgray;
  text-align: left;
  padding: 2px 8px; }
table tr td {
  border: 1px solid silver;
  text-align: left;
  padding: 1px 8px;}
```

APPENDIX D – SUMMARY OF FILES PROVIDED WITH JUPITER-GRAM

The following are provided with the Jupiter-GRAM distribution:

- Build: A makefile system for building the GRAM Suite.
- MSVS: A Visual Studio solution for building the GRAM Suite (no Fortran).
- Documentation: A User Guide, a Programmer's Manual, and a GRAM Suite change log.
- Windows: Binary executables and libraries (64-bit) for Windows.
- Linux: Binary executables and libraries (64-bit) for Linux.
- common: A framework shared by all GRAM models:
    - include: Header files for the model
    - source: Source code for the model
    - examples: Generic example functions
    - unittest: Source code for unit tests
    - cspice: Headers and libraries for the NAIF SPICE toolkit
    - googletest: Headers and source for the unit test framework
- Jupiter: The model-specific code, examples, and tests for each planet
    - include: Header files for the model
    - source: Source code for the model
    - examples: Examples and the GRAM program for this model
    - unittest: Source code for unit tests
    - sample_inputs: Sample input parameter files and resulting outputs
    - md files: Markdown files used to build the Programmer's Manual
- GRAM: Source files for examples that combine all GRAM models.
- Doxyfile and DoxygenLayout.html: Configuration files used to generate the Programmer's Manual

APPENDIX E – BUILDING JUPITER-GRAM

The Jupiter-GRAM distribution contains 64-bit executables and libraries for Windows in the folder /GRAM/Windows. These binaries were compiled with Microsoft Visual Studio 2017 using the solution /GRAM/MSVS/GRAMs.sln. To rebuild these binaries:

(1) Open the solution in MSVS 2017.
(2) Set the Solution Configuration to *Release*.
(3) Set the Solution Platform to *x64*.
(4) From the Build menu, select *Rebuild Solution*.

The resulting binaries will be found in /GRAM/MSVS/x64/Release. It is possible to use MSVS 2015 to build Jupiter-GRAM. Instructions can be found in the first chapter of the GRAM Programmer's Manual.

To build Jupiter-GRAM on other operating systems or other compilers, a GNU makefile system is provided in the /GRAM/Build folder. The process for building the executables and libraries is:

(1) Set the build environment in makefile.defs.
(2) Enter the command "make clean".
(3) Enter the command "make –j".

The resulting executables will be placed in /GRAM/Build/bin. Libraries will be placed in /GRAM/Build/lib. The makefile system parameters are defined in the file makefile.defs. The current settings work on a Linux platform or under MSYS2 using the GCC compiler suite version 6.3 or later. The key parameters in this file are:

- CXX, CC, FF, LNK
  - The command that invokes the C++ compiler, C compiler, Fortran compiler, and the linker, respectively.
- CXX_FLAGS
  - Must be set to use the C++11 standard.
- C_FLAGS
  - Must be set to use the C99 standard.
- F_FLAGS
  - Must be set to use the Fortran 2003 standard.
- SPICE_LIB
  - Path to the NAIF CSPICE library.

The above processes use pre-built SPICE libraries that were compiled following the cspice instructions (version N0066). These libraries are found in /GRAM/common/cspice/lib. To rebuild these libraries, please refer to the README.txt file that comes with the appropriate CSPICE toolkit. The toolkits can be obtained from https://naif.jpl.nasa.gov/naif/toolkit_C.html.

APPENDIX F – HISTORY OF JUPITER-GRAM VERSION REVISIONS

Table 5. Jupiter-GRAM version revisions.

Version	Date	Comments
2021	9/2021	First release of Jupiter-GRAM.

REFERENCES

1. Seiff, A.; Kirk, D.B.; Knight, T.C.D.; Young, R.E.; Mihalov, J.D.; Young, L.A.; Milos, F.S.; Schubert, G.; Blanchard, R.C.; and Atkinson, D.: “Thermal structure of Jupiter's atmosphere near the edge of a 5-μm hot spot in the north equatorial belt,” *Journal of Geophysical Research*, Vol. 103, No. E10, pp. 22,857-22,889, 1998.

2. Seiff, A.; and Knight, T.C.D.: “The Galileo probe Atmospheric Structure Instrument,” *Space Science Reviews*, Vol. 60, No. 1-4, pp. 203-232, 1992.

3. Von Zahn, U.; Hunten, D.M.; and Lehmacher, G.: “Helium in Jupiter’s Atmosphere: Results from the Galileo probe Helium Interferometer Experiment,” *Journal of Geophysical Research*, Vol. 103, No. E10, pp. 22,815-22,879, 1998.

4. Niemann, H.B.; Atreya, S.K.; Carignan, G.R.; Donahue, T.M.; Haberman, J.A.; Harpold, D.N.; Hartle, R.E.; Hunten, D.M.; Kasprzak, W.T.; Mahaffy, P.R.; Owen, T.C.; and Way, S.H.: “The composition of the Jovian atmosphere as determined by the Galileo probe mass spectrometer,” *Journal of Geophysical Research*, Vol. 103, No. E10, pp. 22,831-22,845, 1998.

5. Folkner, W.M.; Woo, R.; Nandi, S.: “Ammonia abundance in Jupiter’s atmosphere derived from the attenuation of the Galileo probe’s radio signal,” *Journal of Geophysical Research*, Vol. 103, No. E10, pp. 22,847-22,855, 1998.

6. Sromovsky, L.A.; Collard, A.D.; Fry, P.M.; Orton, G.S.; Lemmon, M.T.; Tomasko, M.G.; and Freedman, R.S.: “Galileo probe measurements of thermal and solar radiation fluxes in the Jovian atmosphere,” *Journal of Geophysical Research*, Vol. 103, No. E10, pp. 22,929-22,977, 1998.