Jupiter Conjunction Atmospheric Upgrade for Orbiter 2010

By DR_AERONAUTICS

Description

This atmospheric upgrade for Jupiter is designed for orbonauts like myself, who enjoy performing re-entries and flight into the upper atmospheres of gas giants. As there is so big a price to pay fuel-wise for so little practical reason, I would imagine there's not many of you. However if you tried this stunt you likely noticed the Surface MFD's thermal readout reading a constant 288K. Thus the vanilla orbiter config file defines the following for Jupiter's atmosphere:

T=288K

P=2014*e^(-z/h) in kPa, where z is a constant

Rho= 1.3293*e^-(g0/(R*T)*z) kg/m^3, where g0 and T are constant

The result is an atmosphere that is generic in orbiter. It takes only a few minutes to define, but is totally lacking distinct layers. This makes it much more difficult to recreate any future Jupiter missions or demonstrate concepts of said missions with any reasonable accuracy. Such examples might include (but are not limited to) attempting aerocapture at 400 kilometers per a NASA design document only to find thyself incinerated, and attempting to re-enter with a delta glider and seeing higher than realistic heating due to a stratosphere at +15C. This is precisely what encouraged me to spend the hours involved in coding up a proper atmosphere.

We are lucky to have a spaceflight agency with enough scientists to mandate a Jupiter orbiter also carry along a hardened atmospheric suicide probe and drop it into the soupy well that is Jupiter's atmosphere. The Galileo probe entered the atmosphere of Jupiter on December 7, 1995, decelerating at a peak 207 g, heating to over 15,000C, and loosing half its heat shield to ablation. It collected atmospheric data we need for this module across a span of 1161 kilometers, before the probe overheated, the parachute melted, and its contents evaporated to become a permanent part of Jupiter. Using the data the probe gave us, I was able to identify 6 unique layers, meaning this module does the work of 6 config files, each one also defining temperature.

Process

A major hurtle for literally years was the ability for me to find released raw data from the probe. What I could dig up were generalized graphics that only offered up to 12 data points often clustered in one region of the atmosphere, preventing me from identifying any layer trends. Finally I came across the raw ASCII data recorded for the mission on the Planetary Data System website: https://pds.nasa.gov/ds-view/pds/viewProfile.jsp?dsid=GP-J-ASI-3-ENTRY-V1.0

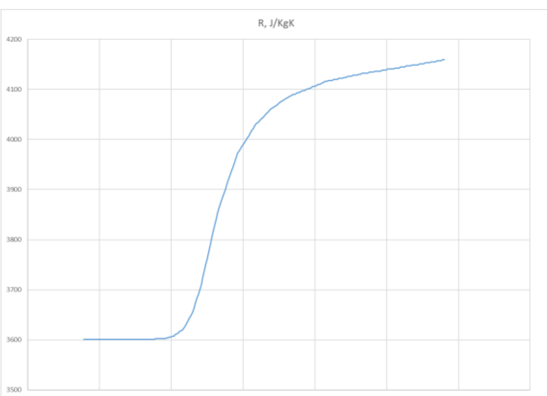
The data we are interested in came from the ASI instrument, literally Atmospheric Structure Instrument, convenient as we are trying to create the structure of an atmosphere. I collected data from 4 tables that provided me with temperature, pressure, and density throughout the mission: descent.tab, UpperATM.tab, LowerATM.tab, and ELCoeffs.tab to identify cloud layers (which only 1 can be simulated in Orbiter).

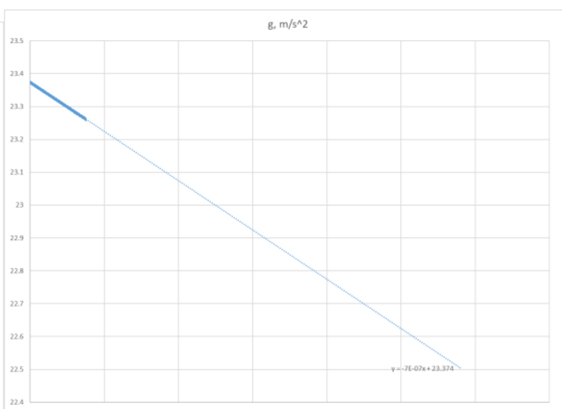
The ascii data were imported onto an excel spreadsheet and converted to an excel table. The data was then compiled onto a single table where the abbreviated header is shown below.

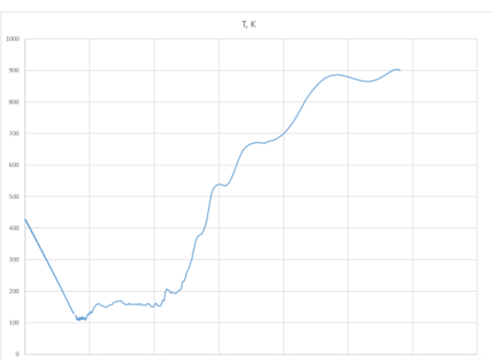
Time	Altitude	Pressure	Temperat	Density	Molecular	CP OVER (	R	V_REL	FPA	Lattitude	Longitude	Gravity	Descent V	Aerotherr
SECONDS	KM	BARS	KELVINS	KG/M^3	AMU	N/A	J/KG K	KM/S	degrees	Degrees	Degrees	M/S^2	M/S	KELVINS/I

At this point to keep things harder to make careless mistakes I translated the 0 altitude datum from the 1-bar level to the probe failure altitude. This allowed me to work in Orbiter compliant altitudes, as altitudes below 0 km are not simulated. The translation was exactly +132540m, which is the exact distance from the explicitly defined Jupiter altitude-0 and the last reported reliable packet containing all the information I need.

Next I needed to determine some data trends. This module runs fundamentally from specific gas constant, R, acceleration due to gravity, g, and measured temperature, T. All other parameters are calculated using these 3 base values. The ideal solution to providing these inputs to the module was to plot the data from the Galileo probe and identify specific trend lines that could express these values at any altitude. Shown below are the plots of R, g, and T. All x axis values are in meters.







From the data, the following are the module implemented trendlines:

$$R(z) = 1 + e^{-0.00004*(z-531000)} + 3600$$

Interestingly R seems to fit a logistic growth equation. I don't know enough about atmospheric science to make a conclusion but my guess is it has something to do with the Jovian turbopause (where the atmosphere separates out into its compounds and stops being a mixture).

$$g(z) = -7E - 7 * z + 23.374$$

Yes, I know theory says this is a 1/r^2 law. But I really want the data to fit what was provided by Galileo's probe (to within its margin of error).

For temperature, out comes 6 equations for the 6 layers I found:

$$For\ troposphere, T0(z) = -0.002 * z + 431.13$$

$$For\ tropopause, T1(z) = 8E - 5 * z + 98.104$$

$$For\ stratosphere, T2(z) = 0.0012 * z - 110.4$$

$$For\ stratopause, T3(z) = -2E - 7 * z + 158.15$$

$$For\ thermosphere, T4(z) = 7.8139e^{7E-6*z}$$

$$For\ exosphere, T5(z) = -1E - 9 * z^2 + 0.0027 * z - 696.59$$

You'll notice there is no mesosphere. I didn't find one and this is consistent with planetary atmospheric scientists' positions.

I also needed one final equation to allow for aerocapture. This one is theoretical, as Galileo's probe did not sample way up here. I used the assumption on Earth, which is constant temperature as the atmosphere merges into space. Surprisingly enough, Earth is the planet with the closest behaving atmosphere to Jupiter's, at least above the probe failure altitude. Temperature on its own is irrelevant here, it's largely to support the density calculation.

$$T6(z) = 1090.2$$

Here is the table mapping each equation to altitude:

Temperature Layers			
Troposphere	0	160000	0
Tropopause	160000	188700	1
Stratosphere	188700	229000	2
Stratopause	229000	418000	3
Thermosphere	418000	576000	4
Exosphere	576000	1161000	5
	From	To	ID

From that point, we have what we need to determine p and rho.

For every altitude, the pressure equation was run for the entire layer, and the outputs at the top of the layer defined the equation coefficients for the next layer. Here is a table of coefficients used in the calculations:

Z0	0
Z1	160110
Z2	186164
Z3	223754
Z4	429585
Z5	617656
Z6	1161000
Z7	5000000
T0	431.13
T1	110.93
T2	112.992
T3	158.04
T4	158.0641
T5	589.3995
T6	1090.2
P0	2234216
P1	27820.4
P2	6211.941
P3	1020.965
P4	0.234516
P5	0.005049
P6	9.74E-05
L0	-0.002
L1	8.00E-05
L2	0.0012
L3	-2.00E-07
L4	1.50E-03
L5	0.00071
L6	0
g3	23.21734
g4	23.07335
g5	22.9468
g6	22.5613
R6	4122

For every altitude except the transition,

$$p(z) = P\# * (1 + (z - Z\#) * L\#/T\#)^{-g/(R*L\#)}$$

Where # represents 0-5 as shown in the table. H may be taken by reading the altitudes in the 2nd column of the mapping table. Now one might identify this equation as the ideal gas law, and then recall that Jupiter's atmosphere is not an ideal gas, specifically it is nearly entirely supercritical below 13 bar. However, if I run the ideal gas law 160 kilometers to the top of the troposphere, it produces an error from Galileo's probe of 1.7%. This is a little outside the margin of error of the probe's instruments, but perhaps the pressure differential between the belts and the zones might vary by this much?

In the space transition region, pressure is always $p(z) = P6 * e^{-\frac{g}{R*T6}*(z-Z6)}$

Density is a simple relationship between everything calculated so far. At all altitudes, $\rho = \frac{p}{R*T}$

Now some changes I made to the Jupiter.cfg file:

SidRotPeriod- I'm not sure why it was set to 3.5 hours. This value makes atmospheric entry FAR easier than it would be in real life. The Planetary Society has a nice graphic that shows wind speeds only vary by up to 200 m/s, with a more typical value being 100 m/s. This is so little when compared with Jupiter's rotational speed its not worth an offset. So I returned it to its true rigid body period of 9H50M.

AtmPressure0- adjusted to what Galileo sent in its last packet.

AtmDensity0- adjusted to what Galileo sent in its last packet.

AtmGasConstant- adjusted to what Galileo sent in its last packet.

AtmGamma- adjusted to what Galileo sent in its last packet.

AtmAltLimit- raised to 5000 km to simulate the exosphere transition to space and enable atmospheric driftdown of low hanging orbits.

AtmHazeShift- adjusted to neatly overlie the cloud layer.

CloudAlt- was 234km above the base, or about 0.00067 bar. Nothing can condense to form clouds way up here. The regime of the upper cloud deck of ammonium, forming the white zones, is from 0.3 to 0.7 bar. The middle cloud deck of ammonium hydrosulfide, blanketing the planet and visible as the orange belts, ranges from 1.3 to 2.2 bar. As it covers the entire planet, the layer was adjusted to the 1.7 bar line and acts as the ammonium hydrosulfide layer. The really neat thing about this is the brighter clouds carry an optical illusion at high altitudes that makes them look higher than the deeper ammonium hydrosulfide clouds, as they actually are in Jupiter.

CloudRotPeriod- adjusted so that it moves at 100 m/s relative to the rigid body of Jupiter. Note that because Jupiter is NOT a rigid body, its cloud deck appearance evolves over time. So you'll see a slightly different surface every day.

Well that about covers it. I hope you enjoy flying in your realistically upgraded Jupiter, being mindful that if you try to descend below 22 bar, your mission will end, as Galileo’s Probe did.

If you have questions or want to learn more you can reach out on the forums.

Be sure to check out my Youtube Channel: https://www.youtube.com/user/4656nick/