Tuijin Jishu/Journal of Propulsion Technology
ISSN:1001-4055
Vol. 44 No.4 (2023)

Atmosim – An Atmosphere Simulation Package in the Tool Command Language (Tcl)

Morlang F.

Department of ATM Simulation, Institute of Flight Guidance, German Aerospace Center (DLR), Germany

Abstract- DLR's SpaceLiner orbiter concept flight simulator gets a change from its commercial flight simulation software "X-Plane" dedicated solutionto a distributed system of systems high level architecture (HLA) based approach. For that purpose, a Tclatmosphere simulation package has been developed. Its configurable parametrization including the coverage of altitudes up to 10^9 meters is described. Package usage test results in a standalone application show an execution time performance below 300 microseconds. Future options of different human-in-the-loop real-time system of systems integration capabilities are presented and discussed.

Keywords: *aerospace, distributed simulation, HLA, real-time, Tcl/Tk*

1. Introduction

DLR's advanced concept for a suborbital, hypersonic, winged passenger transport vehicle called SpaceLiner, is the use case for realization testing and validation of future space traffic integration concepts. Segregation of future landing-like-an-aircraft commercial space vehicles will need a transition to integration, especially if the number of these types of spacecraft will increase. To perform validation simulations for this scenario, a real-time human-in-the-loop simulation model for the commercial flight simulation software "X-Plane" has been developed (Figure 1). Although this current solution covers a wide range of usage and extension possibilities, constraints referring to proprietary data formats and a dedicated vendor dependency consolidated the conclusion for a needed change to a distributed system of systems (Figure 2). Here, the simulation will berealized as an HLA 4 federation based on the Space Reference Federation Object Model (SpaceFOM). An atmosphere simulation package has been established to facilitate the development of an atmosphere HLA federate application in Tcl.

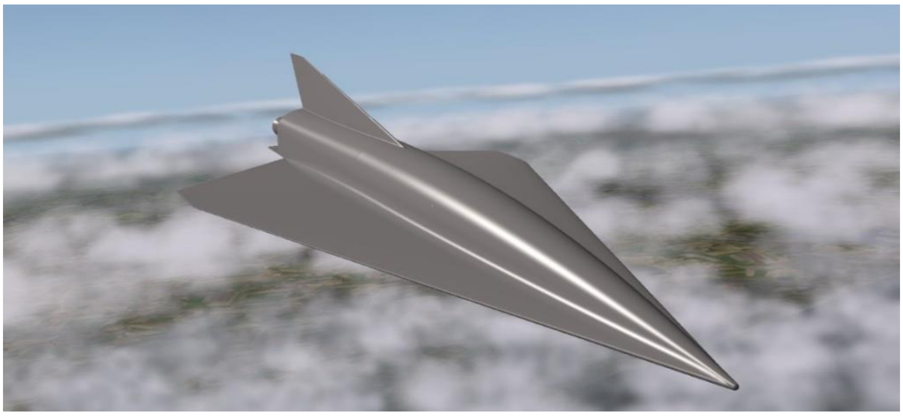

Figure 1:Space Liner simulation screenshot

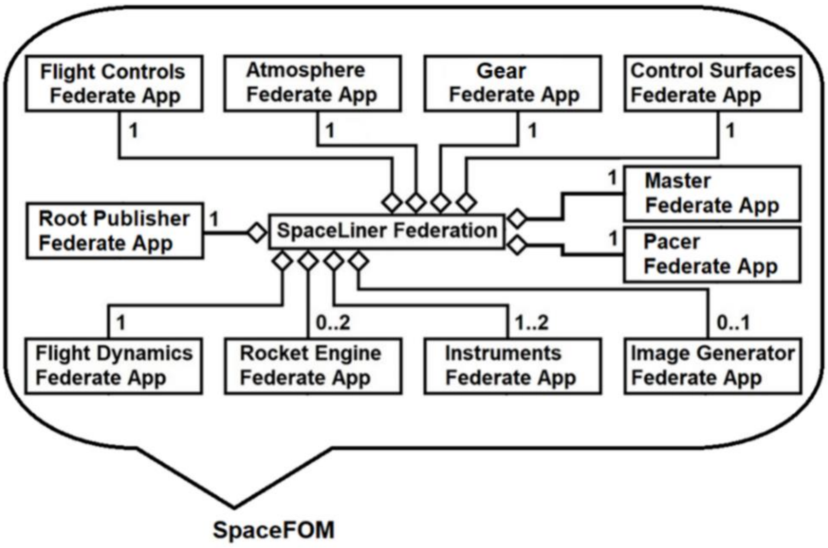

Figure 2:Space Liner system of systems concept[1]

2. Method

The principle behavior of the atmosphere model is shown in Figure 3. From a given temperature at sea level, a given pressure at sea level and a given geopotential altitude as input parameters, the air density, temperature and pressure at that altitude are calculated. In order to implement these calculation possibilities, the atmoSim package provides the following methods:

- Get Temperature(altitude pressure sea level temperature sea level),
- get Pressure (altitude pressure sea level temperature sea level),
- get Density (temperature pressure).

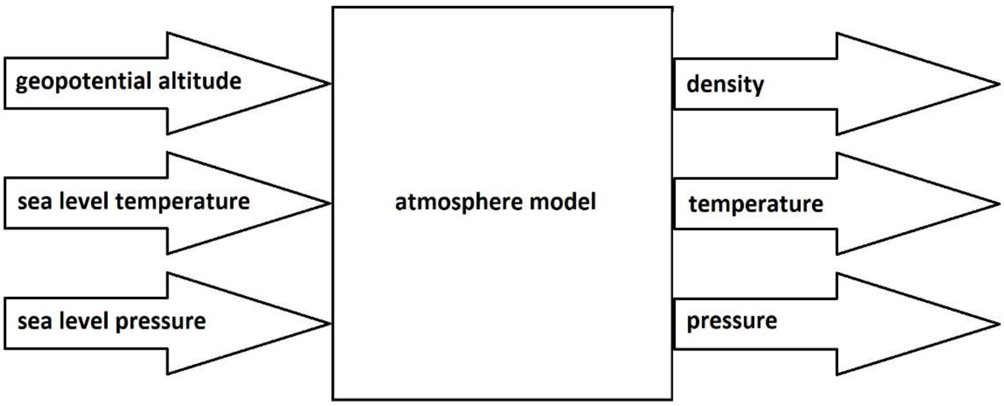

Figure 3:Atmosphere model

The temperature change with geopotential altitude is defined according to Table 1, where the layers id 0 to 7 refer to [2]. Temperature values between the explicit layers are derived with linear interpolation according to equation 1 with T_{ref} and h_{ref} as the layer temperature and layer altitude, -dT/dh = a isthe temperature lapse for that layer.The pressure values are calculated according to equation 2 or equation 3, depending on the existence of a temperature lapse rate a or not.Equations 2 and3 are derived from (25) and (28) in [3]. Air density values result from equation 4.

Table 1:Reference values from [2]

id	h / m	T / °C	p / Pa	a / K / m
0	0	15	101325.00000	0.0065
1	11000	-56.5	22632.10000	0.0
2	20000	-56.5	5474.89000	-0.0010
3	32000	-44.5	868.01900	-0.0028
4	47000	-2.5	110.90600	0.0
5	51000	-2.5	66.93890	0.0028
6	71000	-58.5	3.95642	0.0020
7	80000	-76.5	0.88628	0.0

The parameters in equation 1 to equation 4 are as follows:

a =atmospheric lapse rate in K/m

id =layer no.

g =acceleration of gravity in m/s^2

h =geopotential altitude in m

M =molar mass of air in kg/mol

p_h =pressure at altitude h in Pa

p_{ref} =pressure at layer reference altitude in Pa

R =gas constant = 8.31446261815324 J/molK

ρ_h =air density at altitude h in kg/m^3

T_h =temperature at altitude h in K

T_{ref} =temperature at layer reference altitude in K

$$T_h = T_{ref} - a * (h - h_{ref})$$

Equation 1

$$p_h = p_{ref} * \frac{T_{ref} - a * (h - h_{ref})}{T_{ref}}^{\frac{\frac{1}{a*R}}{\frac{M}{g}}}$$

Equation 2

$$p_h = p_{ref} * e^{\frac{\frac{-g*M}{R}}{T_{ref} * (h - h_{ref})}}$$

Equation 3

$$\rho_h = \frac{p_h * M}{R * T_h}$$

Equation 4

Tuijin Jishu/Journal of Propulsion Technology
ISSN:1001-4055
Vol. 44 No.4 (2023)

For a geopotential altitude 80000 m < h <= 1000000 m the temperature and pressure values are calculated according to [4] (Table 2 and Table 3).

Table 2:Implemented temperature calculation for 80000 m < h <= 1000000 m according to [4]

id	h / m	T / °C
8	80000 < h <= 86000	(356.65 – 2.0 *(h / 1000)) - 273.15
9	86000 < h <= 91000	-86.2827
10	91000 < h <= 110000	(263.1905 – 76.3232 *sqrt[1 – (((h / 1000) – 91) / –19.9429)^2]) - 273.15
11	110000 < h <= 120000	(240 + 12 * ((h / 1000) – 110)) - 273.15
12	120000 < h <= 1000000	(1000 - 640 * exp(-0.01875 * □)) - 273.15, with: □ = ((h / 1000) - 120) * (6356.766 + 120) / (6356.766 + (h / 1000))

Table 3:Implemented pressure calculation for 80000 m < h <= 1000000 m according to [4]

id	h / m	p / Pa
8	80000 < h <= 86000	3.956420 * (214.65 / (214.65 - 2 * ((h / 1000) - 71)))$^{34.1632 / -2}$
9	86000 < h <= 91000	exp(A * (h / 1000)^4 + B * (h / 1000)^3 + C * (h / 1000)^2 + D * (h / 1000) + E) with: A = 0.0 B = 2.159582e-06 C = -4.836957e-04 D = -0.1425192 E = 13.47530
10	91000 < h <= 100000	exp(A * (h / 1000)^4 + B * (h / 1000)^3 + C * (h / 1000)^2 + D * (h / 1000) + E) with: A = 0.0 B = 3.304895e-05 C = -0.009062730 D = 0.6516698 E = -11.03037
11	100000 < h <= 110000	exp(A * (h / 1000)^4 + B * (h / 1000)^3 + C * (h / 1000)^2 + D * (h / 1000) + E)

		with: A = 0.0 B = 6.693926e-05 C = -0.01945388 D = 1.719080 E = -47.75030
12	110000 < h <= 120000	$\exp(A * (h / 1000)^4 + B * (h / 1000)^3 + C * (h / 1000)^2 + D * (h / 1000) + E)$ with: A = 0.0 B = -6.539316e-05 C = 0.02485568 D = -3.223620 E = 135.9355
13	120000 < h <= 150000	$\exp(A * (h / 1000)^4 + B * (h / 1000)^3 + C * (h / 1000)^2 + D * (h / 1000) + E)$ with: A = 2.283506e-07 B = -1.343221e-04 C = 0.02999016 D = -3.055446 E = 113.5764
14	150000 < h <= 200000	$\exp(A * (h / 1000)^4 + B * (h / 1000)^3 + C * (h / 1000)^2 + D * (h / 1000) + E)$ with: A = 1.209434e-08 B = -9.692458e-06 C = 0.003002041 D = -0.4523015 E = 19.19151
15	200000 < h <= 300000	$\exp(A * (h / 1000)^4 + B * (h / 1000)^3 + C * (h / 1000)^2 + D * (h / 1000) + E)$ with: A = 8.113942e-10 B = -9.822568e-07

		C = 4.687616e-04 D = -0.1231710 E = 3.067409
16	300000 < h <= 500000	$\exp(A * (h / 1000)^4 + B * (h / 1000)^3 + C * (h / 1000)^2 + D * (h / 1000) + E)$ with: A = 9.814674e-11 B = -1.654439e-07 C = 1.148115e-04 D = -0.05431334 E = -2.011365
17	500000 < h <= 750000	$\exp(A * (h / 1000)^4 + B * (h / 1000)^3 + C * (h / 1000)^2 + D * (h / 1000) + E)$ with: A = -7.835161e-11 B = 1.964589e-07 C = -1.657213e-04 D = 0.04305869 E = -14.77132
18	750000 < h <= 1000000	$\exp(A * (h / 1000)^4 + B * (h / 1000)^3 + C * (h / 1000)^2 + D * (h / 1000) + E)$ with: A = 2.813255e-11 B = -1.120689e-07 C = 1.695568e-04 D = -0.1188941 E = 14.56718

3. Results

The test configuration of hardware, operation system and software are shown in

Table *4*.

Table 4:Test configuration

Computer	Dell Latitude 7490 32GB memory Intel® Core(TM) i7-8650U CPU @ 2.112 GHz Windows 10 Enterprise (19044)
Tcl/Tk version	Tcl/Tk 8.6.10 (64 bit)

The procedure triples for fetching the relevant data with the associated package methods show an execution time below 300 microseconds for both altitude cases, from ground to 80000 m and from 80000 m to 1000000 m (Figure 4).Tabled data with a 1000 m granularity in [2] were used to check the atmosphere model. This revealed relative errors of up to nearly 30 % (Figure 5). As a consequence, zone corrections in the source code comparable to Figure 6 were made to reduce relative errors to a corridor of +/- 10 % (Figure 7).

Although temperature/pressure/density tabled data for altitudes of 80000 m to 1000000 m could be identified[5], no check of the atmosphere model for this altitude range was realized so far, because the source character of the data requests intensive hand typing for further processing.

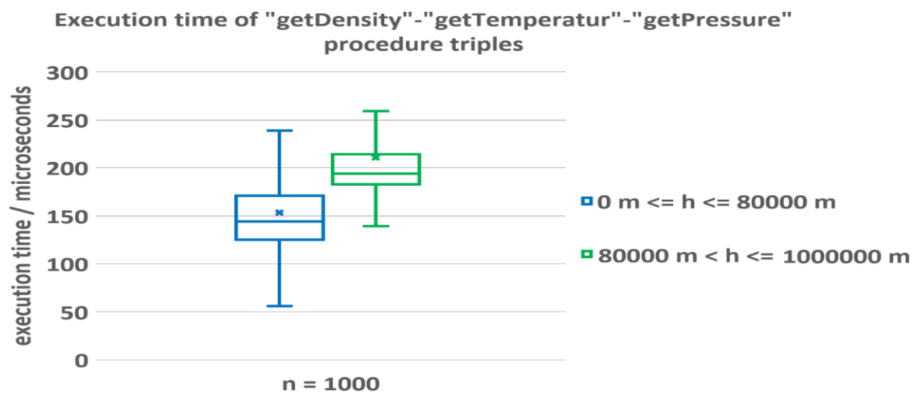

Figure 4:atmoSim performance

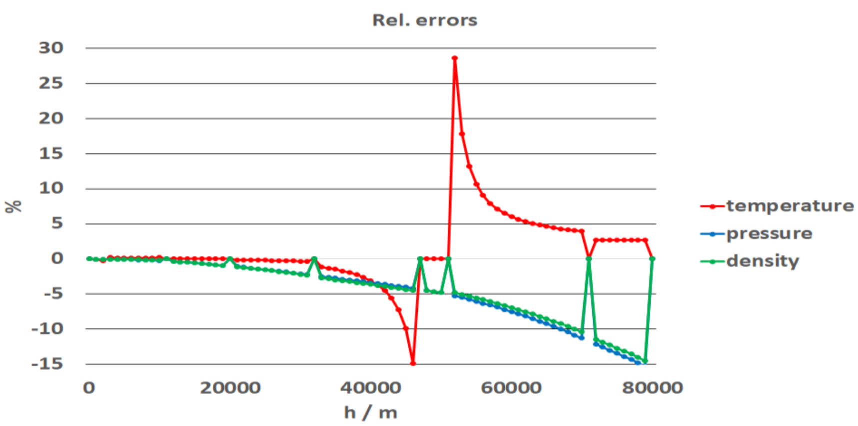

Figure 5:Relative errors

```
method getTemperature {altitude pressuresealevel temperaturesealevel} {
    # ...
    # ..
    # .
    if {$altitude >= 45000 && $altitude <= 46000} {
        set Buffer [my CalculateTemperatureInLayer \
                $altitude $StartAltitude \
                $StartPressure $StartTemperature $Lapse]
        set Corrector [expr {$Buffer / 10.0}]
        return [expr {$Buffer + $Corrector}]
    }
    # ...
    # ..
    # .
}
```

Figure 6:Zone correction code example

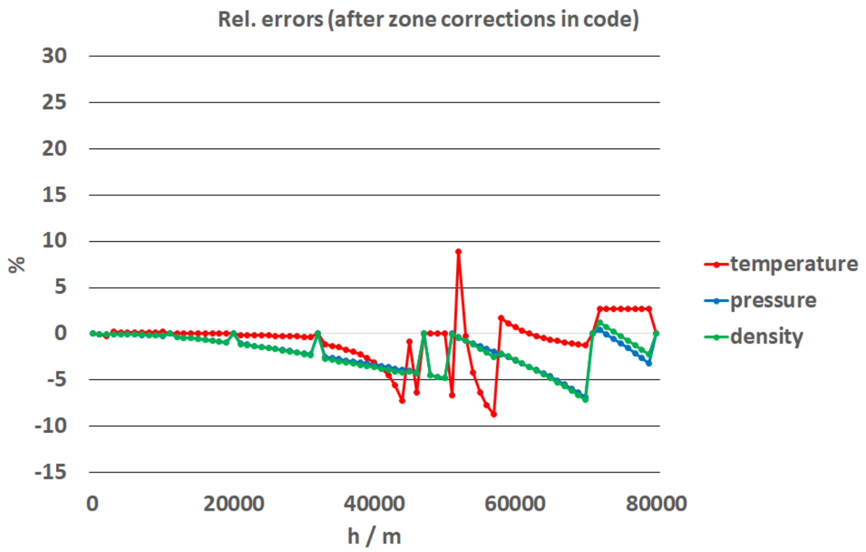

Figure 7:Relative errors after zone corrections

4. Discussion and conclusion

The atmoSim package shows an execution time performance never exceeding 300 microseconds, thus being suitable for the practice in real-time flight simulation use cases. Its accuracy with relative errors below +/- 10% in an altitude range from surface up to 80000 m allows its usage in a wide range of future space traffic integration concept validations, especially when re-entry scenarios are taken into consideration. The exactness of the implemented atmosphere model for altitudes from 80000 m up to 1000000 m still needs a substantiation against reference data but is anticipated to be of sufficient quality, because the guidelines for the altitude dependent function derivations only omit constituent gases that never contribute more than about 0.5% of the total atmosphere composition[4].

Future examinations of the atmoSim package behavior will refer to distributed simulationsin the High-Level Architecture (HLA) context. Here, scenarios will cover the actual HLA Evolved as well as the upcoming HLA 4 standard with its ability to integrate loosely coupled federates via a standardized federate protocol. In this case, the central run-time infrastructure (RTI) component acts as a server over a communications protocol, freeing federate applications from the need to integrate a local RTI component library (Figure 8). Next versions of the package will benefit from alternative calculation procedures in embedded C code for the CriTcl C Runtime in Tcl[6].

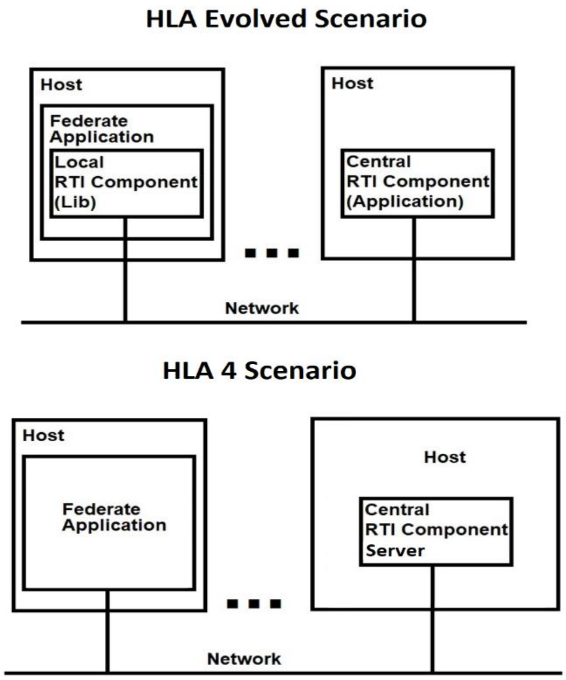

Figure 8:HLA simulation scenarios

Furthermore, air density values will be enhanced by considering air moisture through the replacement of equation 4 by:

$$\rho_h = \frac{p_h}{(R_f * T_h)}$$

Equation 5

with:

$$R_f = R/M/\varepsilon$$

Equation 6

with:

$$\varepsilon = 1 - \varphi * p_d/p_h * (1 - R/M/R_d)$$

Equation 7

Where R_f represents the corrected specific gas constant. R_d is the specific gas constant of watervapor, p_d the saturation vapor pressure and φ characterizes the relative air humidity.

References

[1] Morlang, F., (2021).Loose Coupling in Future Distributed Space Safety Simulations.*Journal of Space Safety Engineering*., Vol. 9, https://doi.org/10.1016/j.jsse.2022.10.002.

[2] International Civil Aviation Organization, (1993)*Manual of the ICAO Standard Atmosphere (extended to 80 kilometres (262 500 feet))* (Third ed.). Doc 7488-CD.

[3] Leslie, F.W. and Justus, C.G., (2011).*The NASA Marshall Space Flight Center Earth Global Reference Atmospheric Model—2010 Version*, [Online]. Available: NASA,https://ntrs.nasa.gov/api/citations/20110012696/downloads/20110012696.pdf. [Accessed Jun.02, 2023].

[4] Braeunig, R.A., (2014).*AtmosphericModels*. [Online].Available: http://www.braeunig.us/space/atmmodel.htm. [Accessed Jun. 02, 2023].

[5] *U.S. Standard Atmosphere, 1976*, (1976). NOAA-S/T 76-1562.

[6] Kupries, A., (2016). C Runtime InTcl.*Proceedings of the 23rd Annual Tcl/TkConference, Houston, TX, USA, November 14-18, 2016.*