

Modeling Heat Transfer in the Space Shuttle Thermal Protection System

NEW MEXICO ADVENTURES IN
SUPERCOMPUTING CHALLENGE

FINAL REPORT

APRIL 7, 2004

TEAM 059

RIO RANCHO MID-HIGH

Project Members:

Chris Morrison

Daniella McCade

Michael Faith

Emily Montana

Teacher:

Debra Loftin

Project Mentor:

Bernie Davenport

Nick Bennett

Table of Contents

Executive Summary	3
Introduction	4
Project Description	
Significance	
Space Shuttle Thermal Protection System	5
HRSI Tile Composition	
Atmospheric Properties	6
Troposphere	
Stratosphere	
Mesosphere	
Thermosphere	
Heat Flow Physics	9
Methods	10
Math Model	12
Algorithm	
Program Verification	15
Other Applications	15
Conclusion	16
Acknowledgments	16
References	17
Appendix A – The Code.....	19

Executive Summary

The purpose of this project is to create a program modeling the heat flow transfer a Shuttle would encounter throughout the liftoff of a mission. The program is written in Starlogo, which characterizes each tile as a patch. Each patch can be given various attributes put in by the user or made into buttons that can be set by the user. The heat capacity of a launch vehicle's ceramic insulation depends on many factors such as, the velocity of the vehicle, the atmospheric density at various points in the launch, and the relationship between friction and temperature at the outer surface of the tile. The model represents the amount of heat that will flow from a particular material (like an HRSI tile) originally to the four adjacent tiles at each time index, per degree of temperature difference per second. For each iteration the values for the heat source and sink are changing.

Ever since humans have traveled into space, safety has been a major concern. In recent years, several Space Shuttles have been destroyed for various reasons. Our goal is to create a computer model that shows the transfer of heat to more accurately determine how much heat the internal environment of the Space Shuttle must deal with. After testing our model several times it did accurately model the expected temperature rise while in the densest parts of the atmosphere and showed a dramatic decrease at greater altitudes.

Introduction

Project Description

The efficiency of the Space Shuttle Thermal Protection System is an important problem to study, because the internal temperature of the shuttle must remain nearly constant. Damage to the tiles can be fatal which was made clear in the recent failure of shuttle tiles that caused the Columbia shuttle to burn up upon reentry on its 2002 mission. We wondered how each tile works to dissipate the heat experienced during a launch and how defective Shuttle tiles would change the temperature of the internal environment of the launch vehicle.

After the disastrous 2002 Shuttle mission Michael Cole, author of a book on the event wrote, "Somehow, the shield was penetrated and hot gas got inside the left wing, near the tires." A CNN's report stated the following findings. First, there were no indications to the Columbia crew or engineers in Mission Control that the mission was in trouble as a result of the foam strike during the ascent. Second, a breach in the Shuttle's Thermal Protection System allowed superheated air to penetrate through the leading edge insulation and progressively melt the aluminum structure of the left wing. Third, the accident was not a random event but rather a result of the spaceflight program's culture, which had as much to do with the accident as the foam did.

The Columbia Accident Investigation Board (CAIB) advanced a number of

theories this one focused on the tile system of the Shuttle and found that the Reinforced Carbon-Carbon RCC tiles bolted on the leading edge of the wing had been damaged. On February 13, 2003 they stated that, “a plasma in the left wheel well may have caused the accident, but also said that heat transfer through the structure, as from a missing tile, would not be sufficient to cause the temperature indications seen in the last minutes of the flight.” NASA evidence collected almost 20 years ago may contradict this statement. This confusion over the amount of heat that could be allowed in due to a missing tile is what started our project.

The cause of the crash, as the CNN report explains, was because the TPS was damaged, progressively leading to the melting of the shuttle’s wing. The friction-created by the, air melted wiring that controlled the shuttle’s movements and as a result Columbia broke apart when its computers were unable to control the space shuttle’s movement (Cole, 2003). This is just one case where a problem with the TPS caused a terrible accident and it seems that if there are any problems with the tiles, it can lead to serious problems for the astronauts.

Space Shuttle Thermal Protection System

Composition

The purpose of the Thermal Protection System (TPS) is to protect the shuttle from the friction-created heat of the atmosphere, which is meant to burn up objects falling to earth. The TPS must also shield the inhabitants and instrumentation of the shuttle from varying temperatures in the 'shade' and 'light' in space, which can rapidly change from 250°F to 3000°F. It is designed as a barrier to melting and freezing during a Space Shuttle mission. The TPS actually consist of four major subsystems which include

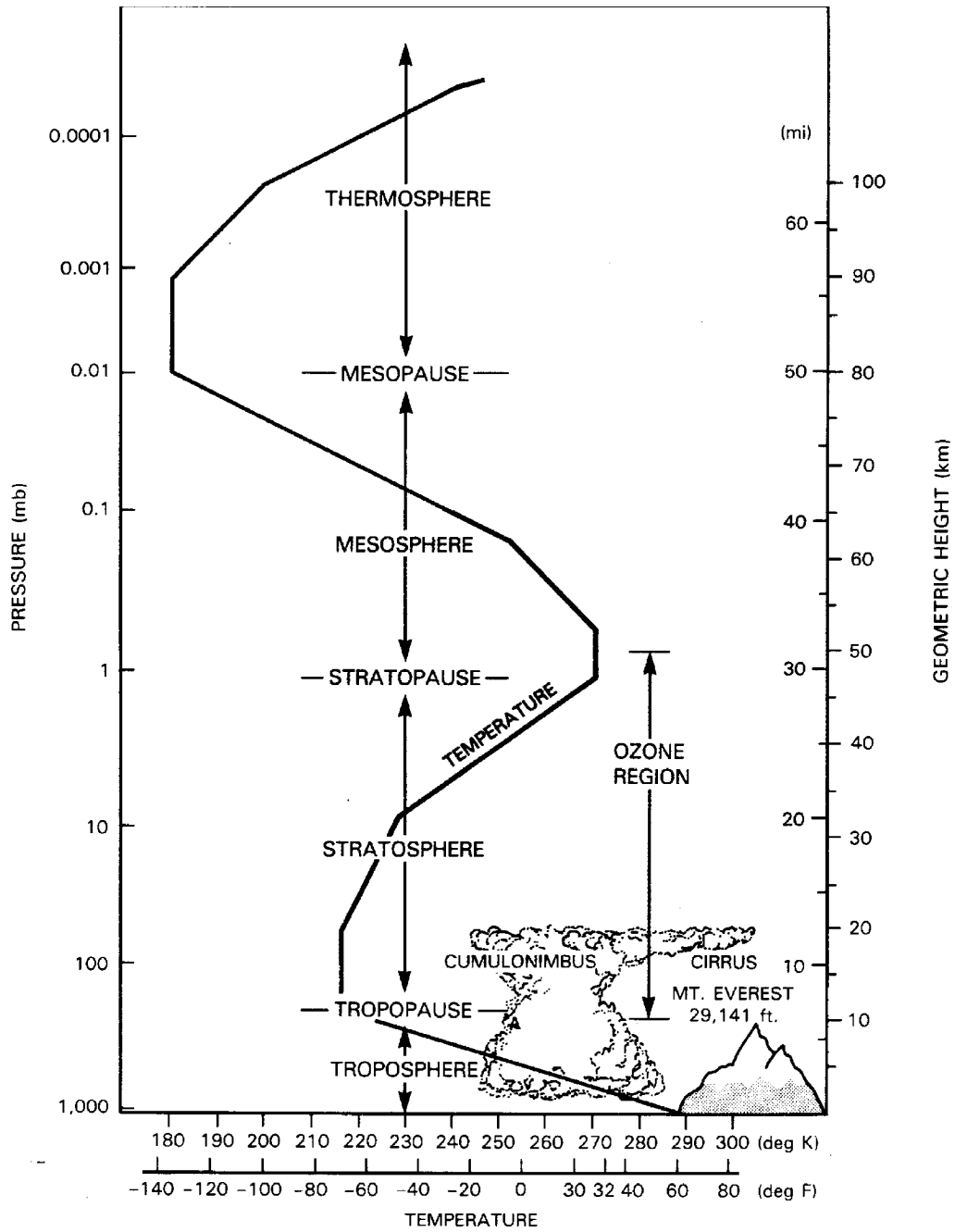
Reinforced-Carbon-Carbon (RCC), Low Temperature Reusable Surface Insulation (LRSI), Flexible Insulation Blankets (FIB), and High Temperature Reusable Surface Insulation (HRSI).

Our program only focuses on these tiles since they are the most common, and here we will discuss some details of their mechanical design specifications. First, it is very important that they are lightweight so that gravity will not be a hindrance. This is achieved by their high porosity, which equals approximately 9lbs/cu.ft. They must also be resilient enough to withstand high G forces in addition to being thermally efficient (Milaninia, Kaveh et al.). They will not crack under the sound of launches (165 decibels), and resist thermal shock damage from repeated fluctuations from such factors as entry, reentry, etc. The silica fibers inside the tiles are not easily flexed, do not freeze, and have a high resilience. They have a low thermal conductivity due to the low coefficient of thermal expansion given for the silica fibers which are used to make the tiles (Milaninia, Kaveh et al.). These fibers are very chemically stable and their flexibility also helps to reduce thermal stress. This property also prevents fiber devitrification or melting at elevated temperatures which are repeated over and over. Finally, in order to avert failure the tiles must also resist ice crystal formation and not be too brittle so that they can be easily formed and shaped to fit the shuttle body (Milaninia Kaveh, et al.). Clearly this very complex material is given simple working parameters in our program and any expansion and contraction is not taken into consideration.

Atmospheric Properties

We realized after creating a simple model of heat flow that the differences in the atmosphere up to about 57 miles would be very important in modeling temperature

fluctuations. A function to indicate the amount of heat building up in the tiles each second at a given velocity and altitude also depends on the change in temperature in each atmospheric layer. We found this information posted at the *Goddard Earth Sciences Data and Information Services Center* site. The diagram below came from The Space Science Division at the Naval Research Lab and helped us to understand what was happening in the atmosphere at the altitudes that the Shuttle reaches during launch.



Troposphere

The troposphere is the inner-most and densest layer of the atmosphere. It extends

14.5 km (5-9 miles) into space and it supports all life on Earth. Also, the majority of the weather that occurs on Earth is created in this layer of the atmosphere. The temperature in the lower troposphere can vary, but as you descend further into the stratosphere, the temperature drops to about 290 to 221K degrees Celsius. This is the layer where almost all of earth's weather occurs, and it is connected to the stratosphere via the tropopause, which is an intermediate layer

(<http://liftoff.msfc.nasa.gov/academy/space/atmosphere.html>).

Stratosphere

Coming immediately after what is called the lower atmosphere (troposphere and tropopause), the stratosphere is recognized for containing the ozone layer, which absorbs ultraviolet radiation. The stratosphere is dry and less dense than the troposphere, and extends 50 km (31 miles) into space

(<http://liftoff.msfc.nasa.gov/academy/space/atmosphere.html>).

Mesosphere

The mesosphere is subsequent to the stratosphere, and reaches out 85 km (53 miles) into the atmosphere. The chemicals in the mesosphere are excited because they absorb the sun's energy, though temperatures can fall as low as 180K or -93°C. The mesosphere and stratosphere (along with their intermediate layers the mesopause and stratopause) make up the middle atmosphere

(<http://liftoff.msfc.nasa.gov/academy/space/atmosphere.html>).

Thermosphere

The thermosphere is the largest of all the layers, extending 600 km (372 miles)

out into space. The temperatures increase the farther out in altitude is measured because of the energy of the sun. Said temperatures can get as high as 2,000K or 1,727°C and the location has caused this layer to be dubbed the upper atmosphere (<http://liftoff.msfc.nasa.gov/academy/space/atmosphere.html>).

Heat Flow Physics

As we searched for information using the Internet and our school library we realized that understanding the physics of heat flow would be very important to our program. We found basic information on how heat moves and a formula for Fourier's law of heat conduction. Another valuable source of information described heat flux rates for launch vehicles.

“As a fundamental law, heat moves from warmer materials to cooler ones until there is no longer a temperature difference between the two” (U.S Department of Energy). There are three ways heat can transfer itself; conduction, convection, and radiation. Conduction spreads heat through different materials by vibration. The molecules close to the source of heat vibrate harshly, and they pass these vibrations to even more adjacent molecules, and on the cycle continues transferring heat energy (U.S. Department of Energy). Radiation is how heat moves through the air from a warmer object to a colder one. Depending on the properties of the object it is transmitted to, radiation can be reflected, absorbed, or transmitted when it hits said object (U.S. Department of Energy). Convection is how heat circulates through gases and liquids. Depending on temperature and density, different fluids can either ‘rise’ or ‘sink’.

The next step was to determine what factors effect the movement of heat. Our

research led us to several equations mostly based on metals. A PowerPoint entitled “Thermal protection Systems for Reusable Launch Vehicles” by Max Blosser was helpful in understanding the key technology for reusable launch vehicles. Improved thermal protection systems are important in extending the life of launch vehicles and will be even more vital as larger vehicles are developed in the future. There are five factors that affect the aerodynamic heating of TPS. First is flow phenomenon such as: free molecular to continuum flow regimes, this basically illustrates that as velocity increases simultaneously with altitude the molecules in the air can behave differently. There is a transition layer at about 330,000 feet where continuum flow changes to rarefied flow where molecules reach free flow. Since we are only considering the Shuttle upon launch we did not incorporate this change because it does not reach these altitudes.

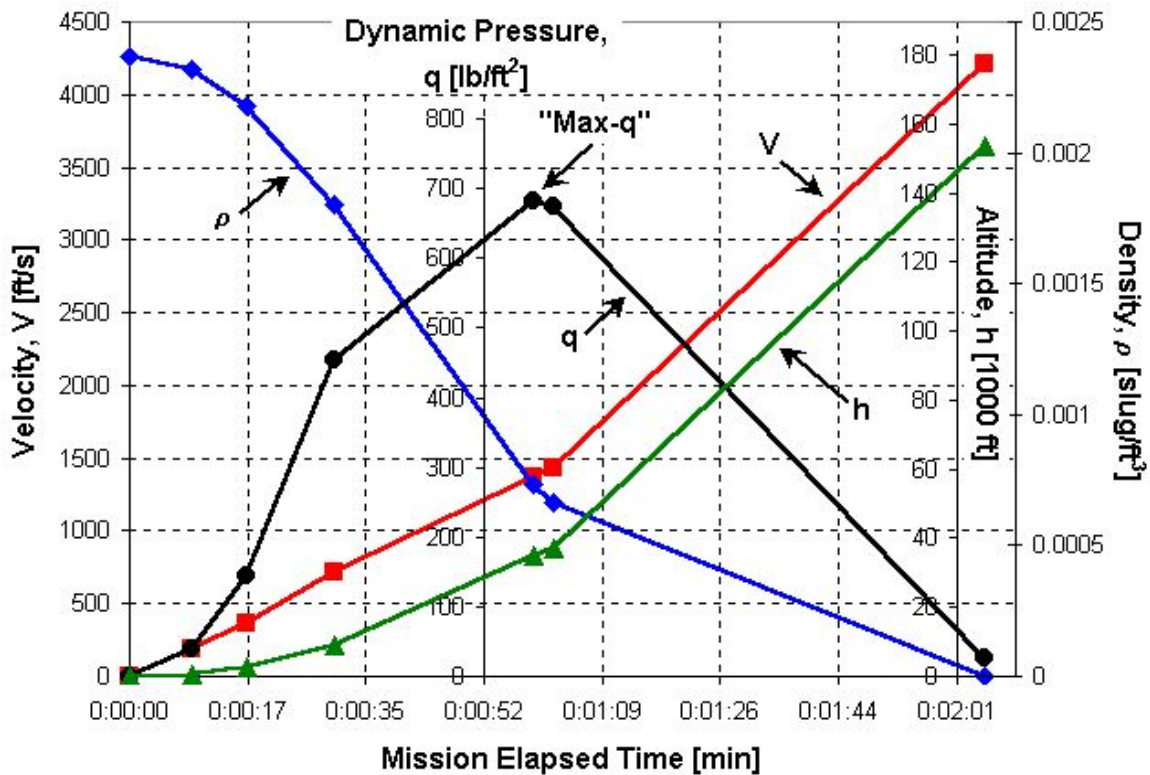
The information we found on Fourier’s Law of heat conduction, in a research summary by P.K. Mohanty and in a document by Steve Baum, basically states that in complex situations the theoretical calculations do not accurately represent the flow of heat in a solid. In its simplest form it states that “heat tends to flow from regions of high concentration of internal energy to regions of low internal energy”, which is represented by the formula $\vec{q} = -C\nabla T$ (Baum). Much of the information relates to the heating of the Earth and was very difficult to fully understand and its accuracy depends on whether the gradient is linear or not.

Methods

In order to complete our program we had to learn as much as possible about heat flow and the Space Shuttle Thermal Protection System. We had to consider the Aerodynamic heating of the TPS which we found relates only to convective heating (at

the Shuttle's velocity) surface smoothness, vehicle geometry and trajectory (both not incorporated into our program at this time), and heating predictions based on air temperature and density in the various levels.

There was a great deal of information on the temperatures of the atmospheric layers but we could not find specific air densities at each altitude, therefore our mentor derived a set decrease based on the air density at sea level which is equal to 0.0023769 slug/ft³. This data and the Simple Exponential Atmospheric Model come from the Gamma Ray Large Area Space Telescope web page posted on the Santa Cruz Institute for Particle Physics Outreach web site. This formula was also used to get an approximation of the ratio of the density at a given altitude to density at sea level.



Doug Jackson posted the above graph at the aerospaceweb.org site, which shows the

relationships of several factors. Our mentor was able to use the information on height (or altitude) and density (in slug/ft³) to create a simplified version of the Shuttle flight profile.

Math Model

At the Kennedy Space Center web site we found an article that described how convective heat transfer of launch vehicles is detected using a Schmidt-Boelter gauge. Our goal is to create a three-dimensional math model, but as of yet we have not found further information on this model. Therefore we did not have a prepared computational model to work from, and relied heavily on our mentor for the algorithm used.

At the aerospace question and answer web site, Doug Jackson states that, “Bernoulli’s equation relates the static pressure and velocity of a fluid at point two.” This is when air density is constant and represents incompressible flow.

Algorithm

Our program uses 4 types of patches. All of these interact and change in the duration of the program. The first is the ambient temperature. Essentially this means the temperature outside the space shuttle. So we gathered research on the different layers of the atmosphere the shuttle encounters (troposphere, stratosphere, mesosphere and thermosphere). We calculated rise in temperature by taking average rise in temperature per kilometer to raise the temperature for each iteration.

Here is the ambient temperature code we based off internet resources

```
ifelse(:altitude <= 13000) [
```

The vehicle is in the troposphere ambient temperature is dropping.

```
    output (initialOutsideTemp - (6.0 * :altitude / 1000.0))  
  ] [
```

```
ifelse (:altitude <= 25000) [
```

The vehicle is in the lower stratosphere ambient temperature is constant.

```
output (initialOutsideTemp - 78)
] [
ifelse (:altitude <= 50000) [
```

The vehicle is in the stratosphere ambient temperature is climbing.

```
output (initialOutsideTemp - 78 + (0.002 * (:altitude - 25000)))
] [
ifelse (:altitude <= 80000) [
```

The vehicle is in the mesosphere ambient temperature is dropping again.

```
output (initialOutsideTemp - 28 - (0.0025 *
(:altitude - 50000)))
] [
```

The vehicle is in the thermosphere outside temperature is climbing again.

```
output (initialOutsideTemp - 103 + (0.006 *
(:altitude - 80000)))
```

The shuttle's altitude is calculated off its velocity therefore we had to find the shuttles differing velocities. We learned that the shuttle reaches mach 1 at time = 120 and that the shuttle reaches to 7280 meters per second to stabilize its orbit. Therefore we averaged the acceleration to 5.417meters per second for the first 120 seconds then from 120-510 the shuttle must accelerate to 7280 so we averaged (time)=120-510 to 18.385's acceleration to 18.385m/s.

Here is the part of the code for velocity

```
getVelocity :time
ifelse (:time <= 120) [
output 5.417 * :time
```

Shuttle acceleration from time 0-120seconds

```
] [  
    ifelse (:time <= 510) [  
        output (650 + 18.385 * (:time - 120))  
    ] [  
        output 7820  
    ]  
]
```

Shuttle acceleration from time 121-510

```
    ]  
end ]
```

Shuttle altitude is a record of the velocity m/s added together.

to getAltitude :time

Here is where we need to calculate how high we have gone at any given time during the flight. Below, we have a simplified version of the shuttle flight profile.

```
    ifelse (:time <= 120) [  
        output (2.7085 * (:time * :time))  
    ] [  
        ifelse (:time <= 196) [  
            output (39000 + 9.1925 * ((:time - 120) * (:time - 120)))  
        ] [  
            output 92000  
        ]  
    ]  
end ]
```

Program Verification

The program was run five times to see if the data generated realistically modeled the change in temperatures that would be expected during launch. At first, the temperature was climbing too high before it reached the thinner layers of the atmosphere. This was corrected by using a more reasonable assumption of atmospheric density. Finally, a button to show the interior temperature and the max interior temperature were

added so that the user could easily view when the internal temperature of the Space Shuttle reached lethal levels. We also realized that our model does not define what each square really represents and that the magnitude of each different indices would have to be adjusted to reflect the actual size chosen. This was a problem that we did not have time to address this year, but would be a good improvement to make in the future.

Other Applications

There are several other applications we believe our program could be modified to address. Clearly, it could be used to model heat flow on other types of launch vehicles, such as the rovers recently sent to Mars. Sliders in the Starlogo program would need to allow the user to adjust them to account for differences in atmospheric density. If the field of patches were arranged to show the exterior shape of the entire Shuttle, then it could be used as a visual aid over the whole shuttle body.

Conclusion

With more time we could have tried to base our program on more accurate information about exactly how the temperature of the tiles would change due to all of the variations in density. Although we were making several important assumptions to be able to write our program we believe that it does a good job of modeling the temperatures during a launch. Shuttle's velocity does not account for how it might change due to fuel loss for example and the program could also be improved by considering other factors that affect the outside air temperature, such as water content. As our programming abilities improve and we take higher level math classes we will be able to work on

problems in a more complex way.

Acknowledgements

There are many people we would like to thank including the administrators who supported us at our school. We would like to thank Nick Bennett for all of his help. If he were not willing to drive all the way across town and fight the traffic up to Rio Rancho we would have never gotten this project finished. We would also like to thank Bernie Davenport who helped us to get started on our research and was a valuable contact. Finally we want to thank Ms. Loftin for being our sponsor as she helped us to stay on track and provided good snacks. Last but not least we would like to thank our parents for letting us stay after school and picking us up when we needed rides.

References

- Baum, Steve. 1997. Fourier's law of heat conduction.
< <http://stommel.tamu.edu/~baum/reid/book1/book/node100.html> >
- Bulavatskyi, V. M. "Nonlinear Boundary-Value Problem for the Heat Mass Transfer Model of W. Fushchych" Symmetry in Nonlinear Math
- “Ceramic”. Dictionary.com. 12-12-03
<<http://dictionary.reference.com/search?q=ceramics&r=67>>
- Chazot, O.. Testing and Catalycity Determination of TPS Materials. VKI.AC.BE. 27 Feb. 2004
<www.vki.ac.be/research/themes/annualsurvey/2002/catalycity_ar0903v1.pdf>
- CNN Report. "Columbia report findings, recommendations." Columbia Accident Investigation Board report. 26 8 2003: 2. © 2003 Cable News Network LP, LLLP. CNN.com/Science & Space. 11/22/03
<<http://edition.cnn.com/2003/TECH/space/08/26/sprj.colu.findings.recs/>>
- CNRS, INCA Wall Interactions and Radiation. National Center for Scientific Research (CNRS). 2/9/04
<<http://www.cnrs.fr/cw/en/pres/compress/INCA/radiation.htm>>
- Cole, Michael. 2003. The Columbia Space Shuttle Disaster From First Liftoff to Tragic Final Flight. Berkeley Heights: Enslow Publishers, Inc., p.41.
- GLAST. Gamma Ray Large Area Space Telescope. Santa Cruz Institute for Particle Physics
<<http://scipp.ucsc.edu/outreach/balloon/glost/environment3.html>>
- Jackson, Doug. 2001. Aerodynamics Questions. Aerospaceweb.org
< <http://www.aerospaceweb.org/question/aerodynamics/q0025.shtml> >
- Lambourne, Robert. The Restless Universe. Open University. 12-12-03
<http://physicalworld.org/restless_universe/html/ru_3_11.html>
- Milaninia, Kaveh, Weber, Dale Andreanna. “High Temperature Reusable Surface Insulation Tiles”
<<http://www.mse.berkeley.edu/classes/matsci102/F01reports/tiles.pdf>>

Mohanty, P. K.. Research Summary (2002-2003). Physics of Complex Systems,
Weizmann Institute. 1/24/04
<<http://www.weizmann.ac.il/home/femohan/ResSum0203.html>>.

Physics. None. 2/23/04
<<http://www.hindawi.dk/books/9775945046/B9775945046000385.pdf>>

U.S. Department of Energy. 8/13/2003. Energy Efficiency and Renewable Energy.
Energy Savers a consumer guide to energy efficiency and renewable energy. Heat
& Cool
<http://www.eere.energy.gov/consumerinfo/heatcool/hc_ssh_heatflow.html>

Appendix A – The Code

```
patches-own [  
    temperature  
    workingTemp  
    leftIndex  
    rightIndex  
]  
  
globals [  
    stepNumber  
    vehicleWidth  
    tileColor  
    hullColor  
    airColor  
    interiorColor  
    settingUp  
    ambientTemperature  
    hullTemperature  
    tileTemperature  
    interiorTemperature  
    centerTemperature  
    maxInteriorTemperature  
    velocity  
    altitude  
]  
  
to startup  
    set settingUp true  
    set stepNumber 0  
    set altitude 0  
    set velocity 0  
    set ambientTemperature initialOutsideTemp  
    set hullTemperature initialOutsideTemp  
    set tileTemperature initialOutsideTemp  
    set interiorTemperature initialOutsideTemp  
    set maxInteriorTemperature interiorTemperature  
    set centerTemperature initialOutsideTemp  
    cleargraphics  
    set vehicleWidth (screen-half-width * 0.9)  
    set tileColor red
```

```

set hullColor violet
set airColor blue
set interiorColor gray
setbg blue
ask-patches [
    set temperature initialOutsideTemp
]
addVehicle
clearplots
convertTempToColor
set settingUp false
end

to addVehicle
; Not only are we going to color the patches according to what they are -
; we're also going to set the "right" and "left" heat transfer indices.
; This will save us a lot of trouble as we iterate.
let [:boundary (round (vehicleWidth / 2))]
ask-patches [
    ifelse ((abs xcor) = :boundary) [
        ; This is the hull
        setpatchcolor hullColor
        ifelse (xcor < 0) [
            set leftIndex hullTileIndex
            set rightIndex interiorHullIndex
        ] [
            set leftIndex interiorHullIndex
            set rightIndex hullTileIndex
        ]
    ] [
        ifelse ((abs xcor) = (:boundary + 1)) [
            ; This is the insulating tile
            setpatchcolor tileColor
            ifelse (xcor < 0) [
                set leftIndex tileExteriorIndex
                set rightIndex hullTileIndex
            ] [
                set leftIndex hullTileIndex
                set rightIndex tileExteriorIndex
            ]
        ] [
            ifelse ((abs xcor) > (:boundary + 1)) [
                ; This is the outside atmosphere, but we also
                need to see if we're immediately next to the tile
                setpatchcolor airColor
                ifelse ((abs xcor) = (:boundary + 2)) [

```

```

; This is the layer right next to the tile
ifelse (xcor < 0) [
    set rightIndex tileExteriorIndex
] [
    set leftIndex tileExteriorIndex
]
] [
indices, based on altituds
; We will be adjusting left and right
]
] [
need to see if we're next to the hull
; This is the interior of the vessel, but we also
setpatchcolor interiorColor
ifelse ((abs xcor) = (:boundary - 1)) [
; This is the layer right next to the hull
ifelse (xcor < 0) [
    set leftIndex interiorHullIndex
    set rightIndex interiorIndex
] [
    set leftIndex interiorIndex
    set rightIndex interiorHullIndex
]
] [
    set leftIndex interiorIndex
    set rightIndex interiorIndex
]
]
]
]
end

to iterate
    set stepNumber (stepNumber + 1)
    let [:temp 0]
    let [:tempList []]
    let [:boundary (round (vehicleWidth / 2))]
    set velocity (getVelocity stepNumber)
    set altitude (getAltitude stepNumber)
    let [:tileHeatAdded (getTileHeatAdded velocity altitude)]
    set ambientTemperature (getAmbientTemperature altitude)
    let [:outsideIndex (getTransferIndex altitude)]
    ; Let's get the temperatures of each patch in a horizontal cross-section of the
screen
    dotimes [:colIndex screen-width] [

```

```

ask-patch-at (:colIndex - screen-half-width - 1) (0) [
  set :temp temperature
]
set :tempList (lput :temp :tempList)
]
; It would probably be more efficient to use a do-times to calculate the new
temp
; for a horizontal cross-section, and then use ask-patches to propogate this
through
; the entire screen, but this way is simple enough.
ask-patches [
  ; Set the working temp to the current temp
  set workingTemp temperature
  if ((abs xcor) > (:boundary + 1)) [
    ; This is the outside atmosphere, but we also need to see if
we're immediately next to the tile
    setpatchcolor airColor
    ifelse ((abs xcor) = (:boundary + 2)) [
      ; This is the layer right next to the tile
      ifelse (xcor < 0) [
        set leftIndex :outsideIndex
      ] [
        set rightIndex :outsideIndex
      ]
    ] [
      set leftIndex :outsideIndex
      set rightIndex :outsideIndex
    ]
  ]
  ; So now, we know the working temp, and we know how much heat is
being transferred to the left and the right
  ; All that's left to do is calculate!
  ; First, calculate the transfer on the left-hand side
  if (xcor > (0 - screen-half-width)) [
    set workingTemp (workingTemp + 0.0025 * leftIndex * ((item
(xcor + screen-half-width) :tempList) - temperature))
  ]
  ; Now, calculate the transfer on the right-hand side
  if (xcor < screen-half-width) [
    set workingTemp (workingTemp + 0.0025 * rightIndex *
((item (xcor + screen-half-width + 2) :tempList) - temperature))
  ]
  ; Now, set the temperature to the working temperature -
  ; except for the extreme edges of the screen, which will get the
ambient temperature,
  ; and the tile, which will have heat added, based on velocity and

```

```

altitude
  ifelse ((abs xcor) = screen-half-width) [
    set temperature ambientTemperature
  ] [
    if ((abs xcor) = (:boundary + 1)) [
      set workingTemp (workingTemp + :tileHeatAdded)
    ]
    set temperature workingTemp
  ]
]
convertTempToColor
updateMonitors
end

```

```

to convertTempToColor
  let [:boundary (round (vehicleWidth / 2))]
  ask-patches [
    ifelse ((abs xcor) = :boundary) [
      ; We are looking at a piece of the hull
      scale-pc hullColor temperature 400 200
    ] [
      ifelse ((abs xcor) = (:boundary + 1)) [
        ; We are looking at a piece of the tile
        scale-pc tileColor temperature 1000 0
      ] [
        ifelse ((abs xcor) > (:boundary + 1)) [
          ; We are looking at a patch of outside air
          scale-pc hullColor temperature 500 0
        ] [
          ; We are looking at a patch of inside air
          scale-pc interiorColor temperature 350 250
        ]
      ]
    ]
  ]
end

```

```

to updateMonitors
  let [:boundary (round (vehicleWidth / 2))]
  ask-patch-at :boundary 0 [
    set hullTemperature temperature
  ]
  ask-patch-at (:boundary + 1) 0 [
    set tileTemperature temperature
  ]
  ask-patch-at (:boundary - 1) 0 [

```

```

        set interiorTemperature temperature
        if (interiorTemperature > maxInteriorTemperature) [
            set maxInteriorTemperature interiorTemperature
        ]
    ]
    ask-patch-at 0 0 [
        set centerTemperature temperature
    ]
end

to getVelocity :time
    ; Here is where we need to calculate how fast we are going at any given time
during
    ; the flight. Below, we have a simplified version of the shuttle flight profile.
    ifelse (:time <= 120) [
        output 5.417 * :time
    ] [
        ifelse (:time <= 510) [
            output (650 + 18.385 * (:time - 120))
        ] [
            output 7820
        ]
    ]
end

to getAltitude :time
    ; Here is where we need to calculate how high we have gone at any given time
during
    ; the flight. Below, we have a simplified version of the shuttle flight profile.
    ifelse (:time <= 120) [
        output (2.7085 * (:time * :time))
    ] [
        ifelse (:time <= 196) [
            output (39000 + 9.1925 * ((:time - 120) * (:time - 120)))
        ] [
            output 92000
        ]
    ]
end

to getDensityIndex :altitude
    ; This is a simple formula that gives an approximation of the ratio of density
at
    ; a given altitude, to density at sea level. This is the "simple exponential
    ; atmospheric model".
    output (exp ((0.0 - :altitude) / 8200.0))

```

```

end

to getTileHeatAdded :velocity :altitude
  ; Here is where we need a function that will tell us much heat (in degrees
  Kelvin)
  ; builds up in the tiles, each second, at a given velocity and altitude. For now,
  ; we will use a very simple approximation.

  ; Assuming that friction is approximately linear with velocity and density,
  ; and that the heat increase per second is a linear function of this friction,
  ; approximate this with some constant factor, multiplied by the velocity, and
  by
  ; the density index.
  output (((tileDrag / 1000.0) * (:velocity * (getdensityIndex :altitude)))
end

to getAmbientTemperature :altitude
  ; Here, we need to compute the ambient air temperature (in Kelvin) at any
  given
  ; altitude.
  ifelse (:altitude <= 13000) [
    ; The vehicle is in the troposphere; ambient temperature is dropping.
    output (initialOutsideTemp - (6.0 * :altitude / 1000.0))
  ] [
    ifelse (:altitude <= 25000) [
      ; The vehicle is in the lower stratosphere; ambient temperature
      is constant.
      output (initialOutsideTemp - 78)
    ] [
      ifelse (:altitude <= 50000) [
        ; The vehicle is in the stratosphere; ambient
        temperature is climbing.
        output (initialOutsideTemp - 78 + (0.002 * (:altitude -
        25000)))
      ] [
        ifelse (:altitude <= 80000) [
          ; The vehicle is in the mesosphere; ambient
          temperature is dropping again.
          output (initialOutsideTemp - 28 - (0.0025 *
          (:altitude - 50000)))
        ] [
          ; the vehicle is in the thermosphere; outside
          temperature is climbing again.
          output (initialOutsideTemp - 103 + (0.006 *
          (:altitude - 80000)))
        ]
      ]
    ]
  ]
end

```

```
    ]
  ]
end
```

to getTransferIndex :altitude

```
  ; Here, we need to compute how fast the outside layer next to the tiles will
  ; dissipate heat, as we get higher (in reality, it would be more accurate to look
  ; at how the tiles radiate heat, but this is a bit easier to handle). At sea level,
  ; this will simply be interiorIndex; as altitude becomes effectively infinite, this
  ; will rise to 1 (all heat dissipates). As a first approximation, let's simply
  ; scale between those two values, based on the air density.
  output (100 - ((100.0 - interiorIndex) * (getDensityIndex :altitude)))
end
```