Lunar Thermal Analysis Guidebook (L-TAG):
Lunar Surface 3D Terrain Modeling

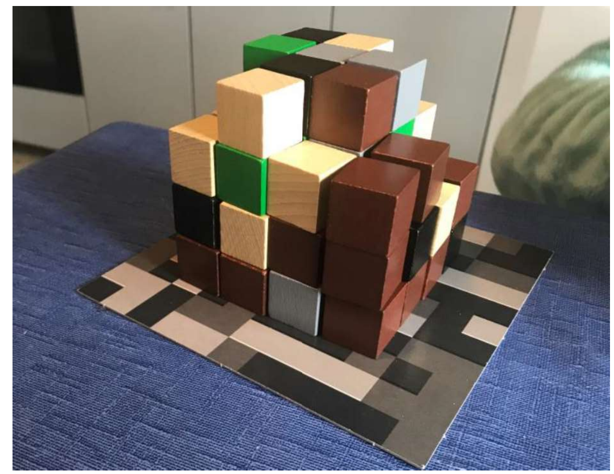
Carlos Gomez, Brian Hamill, Greg Schunk
NASA/Marshall Space Flight Center
&
Lisa Erickson
NASA/Johnson Space Center

Presented By
Lisa Erickson

Thermal & Fluids Analysis Workshop
TFAWS 2022
September 6th-9th, 2022
Virtual Conference

Accessing LRO Digital Terrain Data

- Lunar surface meshes can be constructed using digital elevations models (DEMs) of the surface
- DEMs specify average elevation for squares – bounded by constant LAT/LON lines[1]
- Lunar Surface DEMs are:
  - Derived from Lunar Reconnaissance Orbiter's Lunar Orbiter Laser Altimeter (LOLA) elevation data
  - Provided as Gridded Data Record DEMs (GDRDEMs) – DEMs projected onto a x-y map[1]

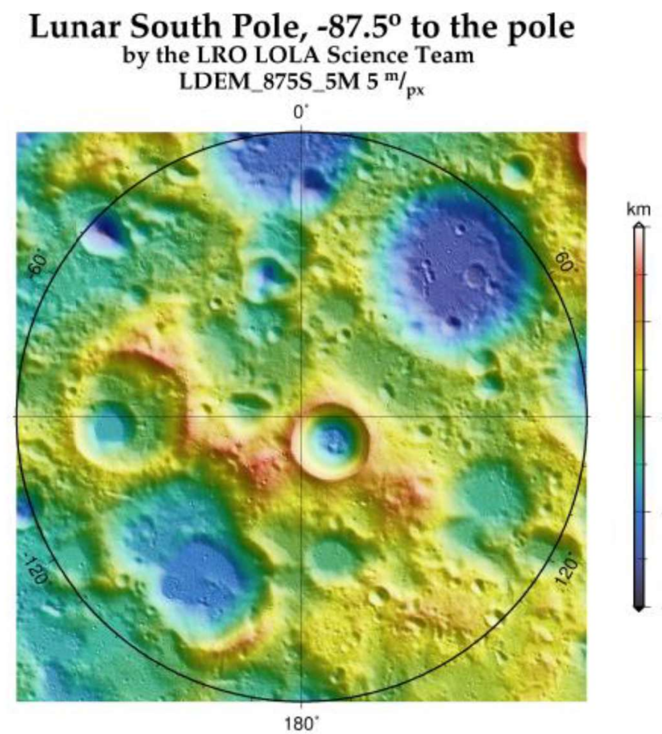


Visualization of DEMs - a block world [2]

[1] "GDRDEM – Gridded Data Record Shape Map" https://ode.rsl.wustl.edu/moon/pagehelp/Content/Missions_Instruments/LRO/LOLA/GDR/GDRDEM.htm
[2] https://commons.wikimedia.org/wiki/File:Minecraft_board_game_blocks_01.jpg

Accessing LRO Digital Terrain Data

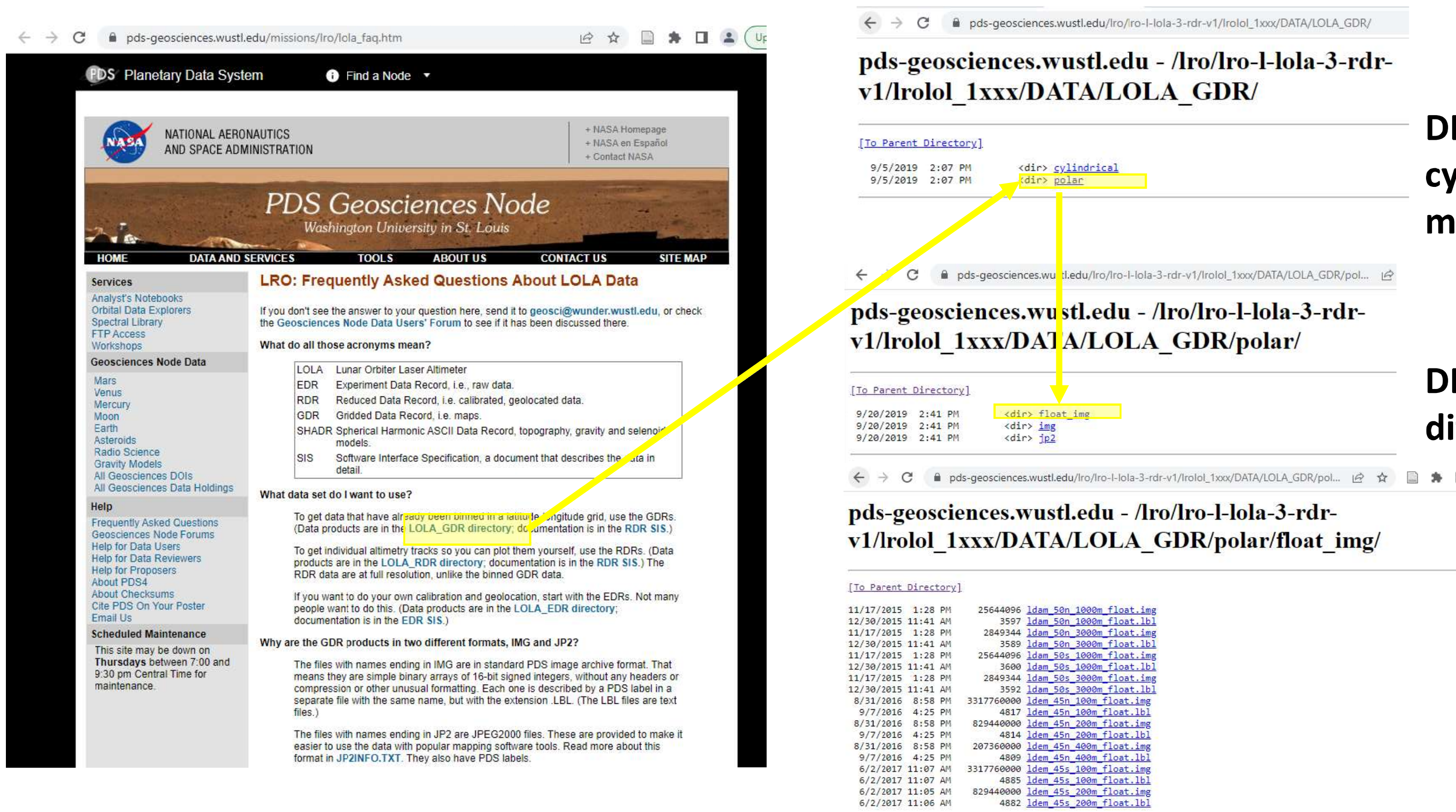
- Lunar surface meshes can be constructed using digital elevations models (DEMs) of the surface
- DEMs specify average elevation for squares – bounded by constant LAT/LON lines[1]
- Lunar Surface DEMs are:
  - Derived from Lunar Reconnaissance Orbiter's Lunar Orbiter Laser Altimeter (LOLA) elevation data
  - Provided as Gridded Data Record DEMs (GDRDEMs) – DEMs projected onto a x-y map[1]



GDRDEM where pixel = elevation[3]

[1] "GDRDEM – Gridded Data Record Shape Map" https://ode.rsl.wustl.edu/moon/pagehelp/Content/Missions_Instruments/LRO/LOLA/GDR/GDRDEM.htm
[2] https://commons.wikimedia.org/wiki/File:Minecraft_board_game_blocks_01.jpg
[3] LOLA Gridded Data Record Shape Map 87.5S 5m https://ode.rsl.wustl.edu/moon/indexproductpage.aspx?product_idgeo=21746070

Where to Find DEMs?

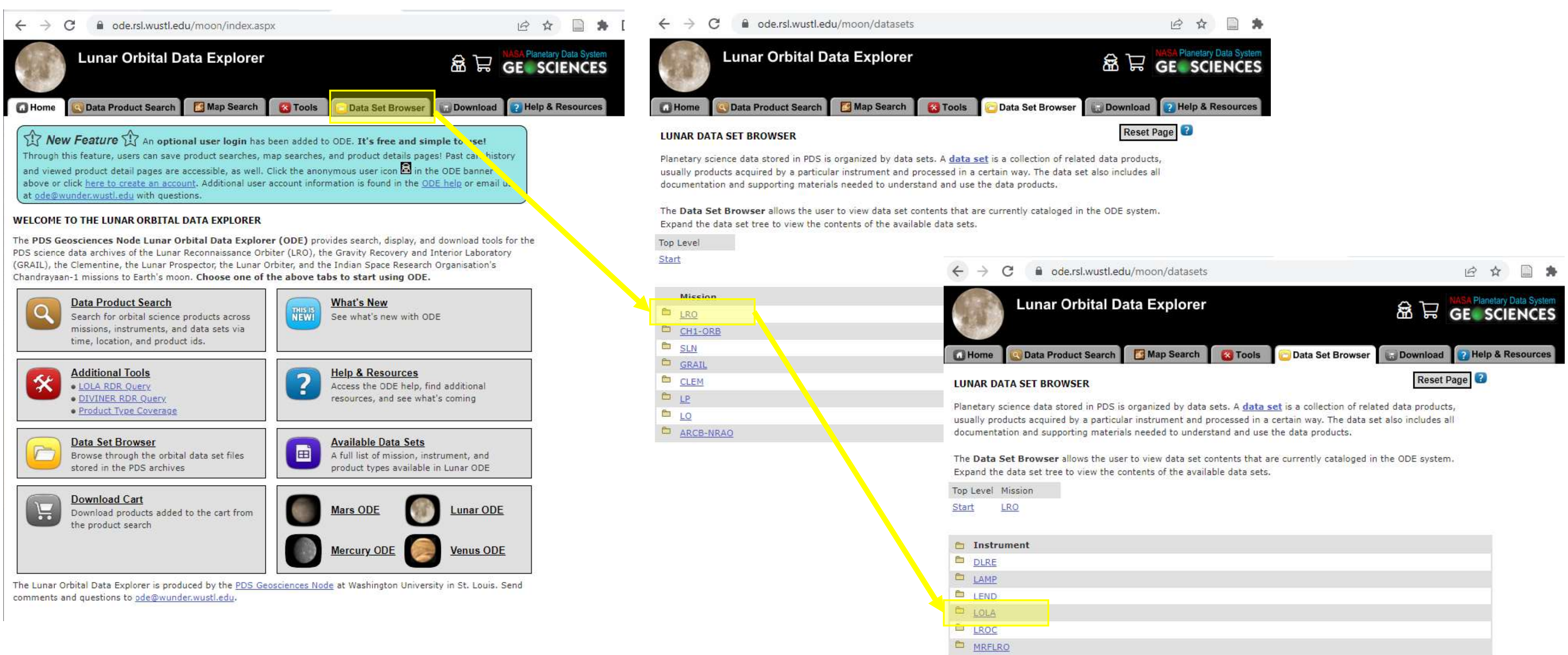


DEMs available with cylindrical and polar map projections

DEMs available in different file formats

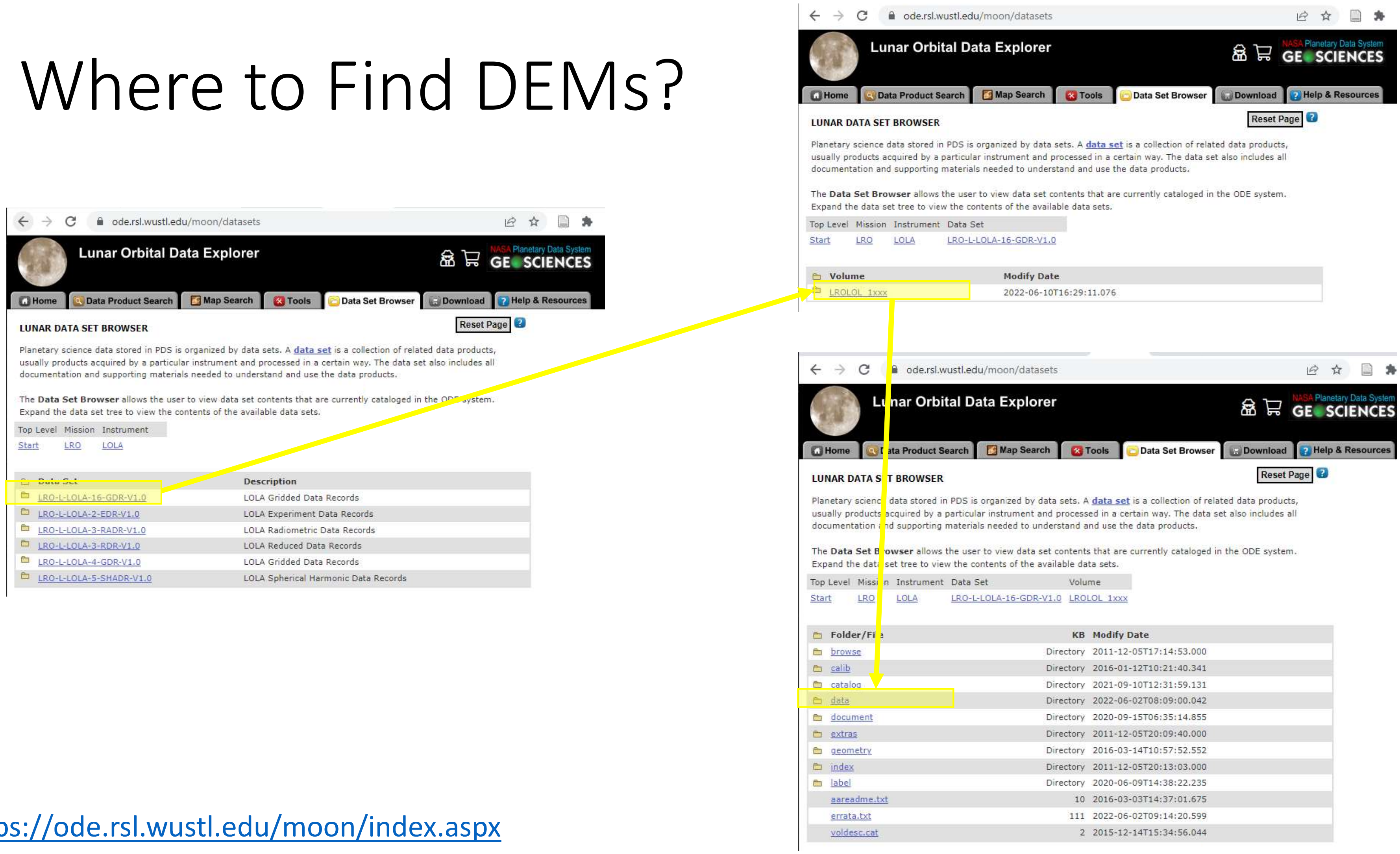
https://pds-geosciences.wustl.edu/missions/lro/lola_faq.htm

Where to Find DEMs?



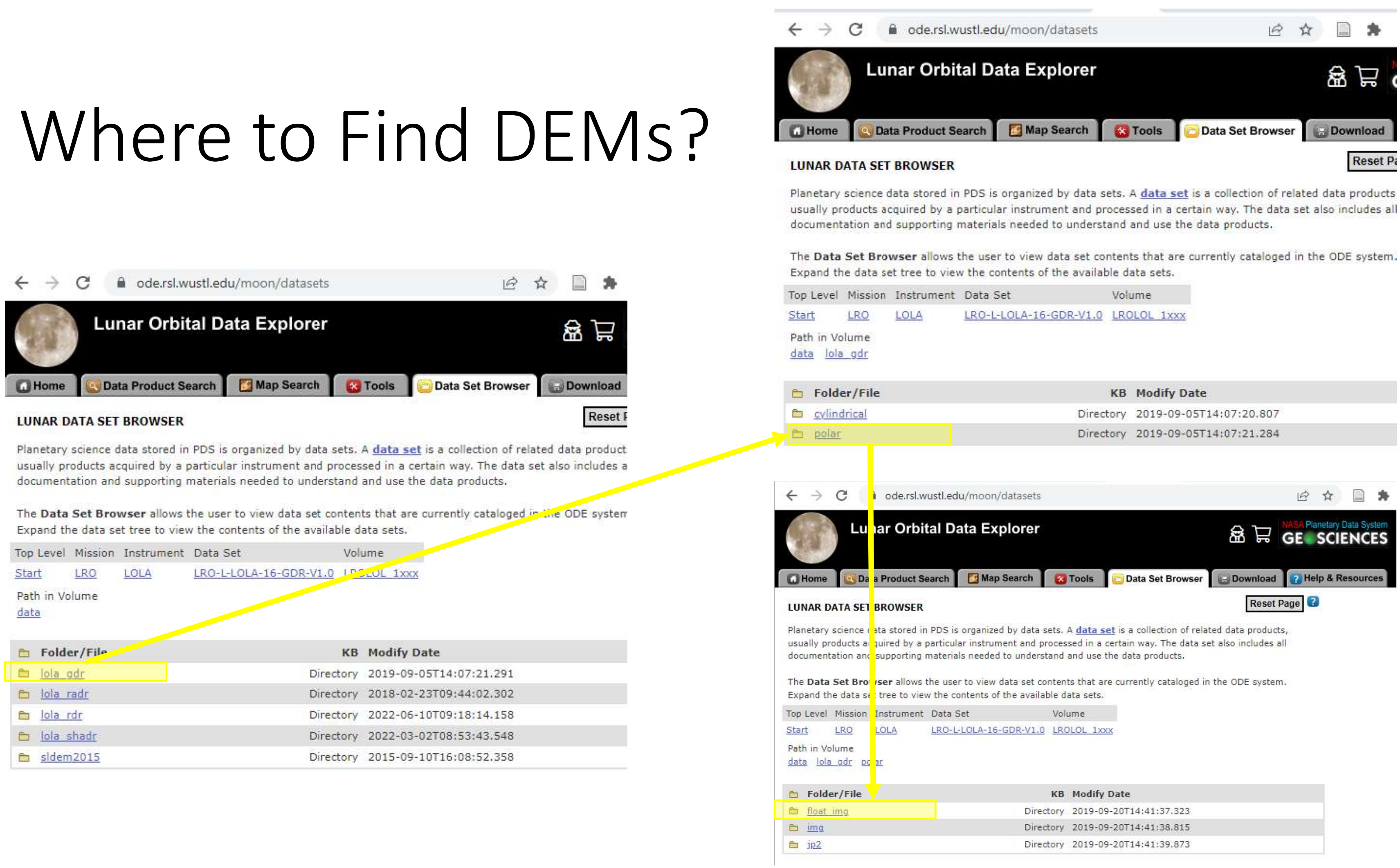
https://ode.rsl.wustl.edu/moon/index.aspx

Where to Find DEMs?



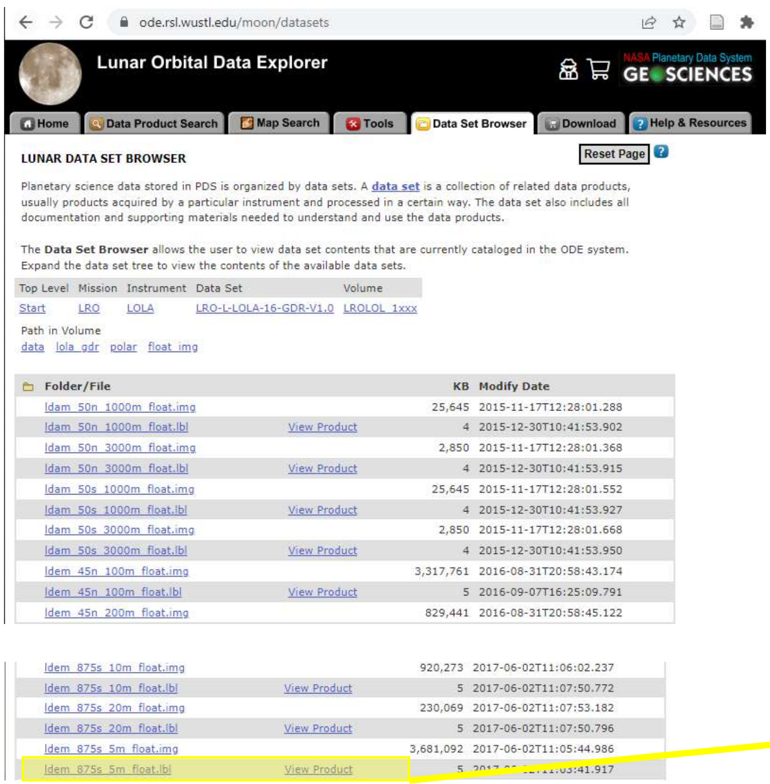
https://ode.rsl.wustl.edu/moon/index.aspx

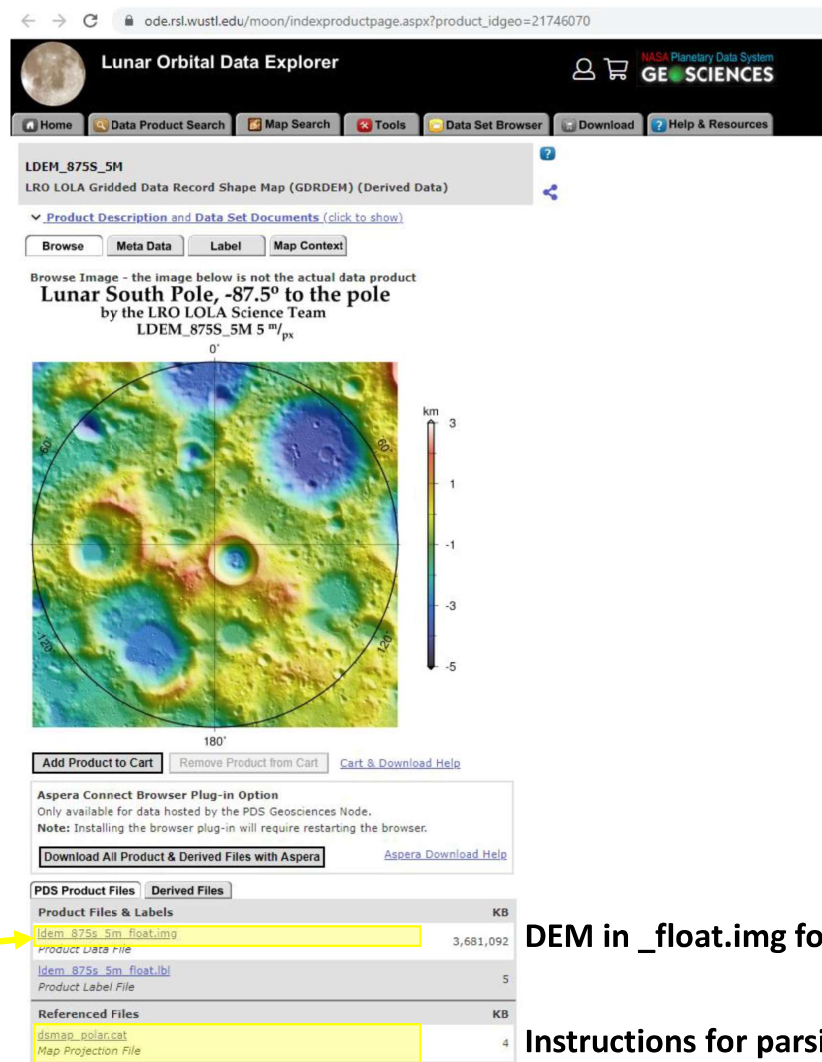
Where to Find DEMs?



https://ode.rsl.wustl.edu/moon/index.aspx

Where to Find DEMs?





DEM in _float.img format

Instructions for parsing DEM data

https://ode.rsl.wustl.edu/moon/index.aspx

Converting LRO DEMs to Mesh

- Two steps:
  1. Convert DEM → Point Cloud (using Python)
  2. Convert Point Cloud → triangular mesh (using MeshLab)
- Note: These steps can be performed multiple ways
  - e.g., A DEM in .tiff format it can be converted using QGIS to a 3D model and meshes can be manipulated in Blender
    - https://www.youtube.com/watch?v=eZZ2MVLZN4o
    - https://www.youtube.com/watch?v=-ubDqXZtdJE

Step 1: Convert DEM → Point Cloud

- Read ldem_xxx_float.img
  - The **ldem_xxx_float.lbl** file gives data structure to assume when uncompressing the data
    - SAMPLE_TYPE = PC_Real means float or ‘f’
    - The total number of samples is LINES*LINE_SAMPLES
  - The **dsmap_polar.cat** states how to parse the file one sample at a time

```
data_file = io.open(input_file,'rb')
raw_data = data_file.read()
datas = struct.unpack_from('f'*LINES*LINE_SAMPLES,raw_data)
i = 1 # sample coordinate
j = 1 # line coordinate
for val in datas: #range(0,len(datas)+1):
# Xmap,Ymap = I,J converted to Cartesian coordinates
#  with 0,0 at center, N = total number of lines
    Xmap = (i - N/2. - 0.5)*MAP_SCALE
    Ymap = (j - N/2. - 0.5)*MAP_SCALE
    # radius from center of map to pixel I,J in meters
    Rmap = (Xmap**2. + Ymap**2.)**0.5
    # elevation from DEM
    Elevation = val*SCALING_FACTOR*1000 # in meters
    # Convert DEM coordinates to spherical coordinates
    # (S. Hemisphere)
    # (LAT, LON, ro) = ([degrees],[degrees],[m])
    LON = 90+math.atan2(Ymap,Xmap)*180/math.pi
    LAT = -90 + 180/math.pi * 2*math.atan(0.5 * Rmap/1737400)
    ro = (val*SCALING_FACTOR+OFFSET)*1000
    i = i + 1
    # Check if finished sampling a line
    if i == LINE_SAMPLES+1:
        j = j+ 1 # increment line coordinate
        i = 1    # reset sample coordinate
    if i*j == LINES*LINE_SAMPLES:
        print(all lines parsed)
```

Sample Python code for reading the DEM

Step 1: Convert DEM → Point Cloud

- Use spherical to cartesian conversion to convert from (LAT, LON, radius) to (x,y,z)

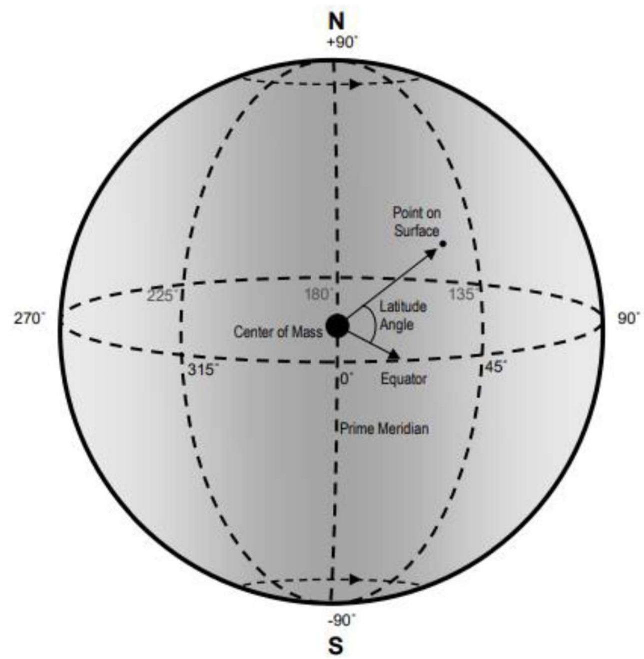


Figure 1. Planetocentric coordinates are expressed as right-handed coordinates with the origin at the center of mass of the body.

https://lunar.gsfc.nasa.gov/library/LunCoordWhitePaper-10-08.pdf

```
# Convert spherical to cartesian coordinates
# (x,y,z) = ([m],[m],[m])
x = ro * math.cos((LAT) * math.pi/180) * math.cos(LON * math.pi/180)
y = ro * math.cos((LAT) * math.pi/180) * math.sin(LON * math.pi/180)
z = ro * math.sin((LAT) * math.pi/180)
```

Sample Python code for Spherical to Cartesian Conversion

Step 1: Convert DEM → Point Cloud

- Reduce size of Point cloud file by selectively outputting points
- Points are outputted to .xyz file with format:

lunar_terrain_40m_boundbox.xyz - Notepad

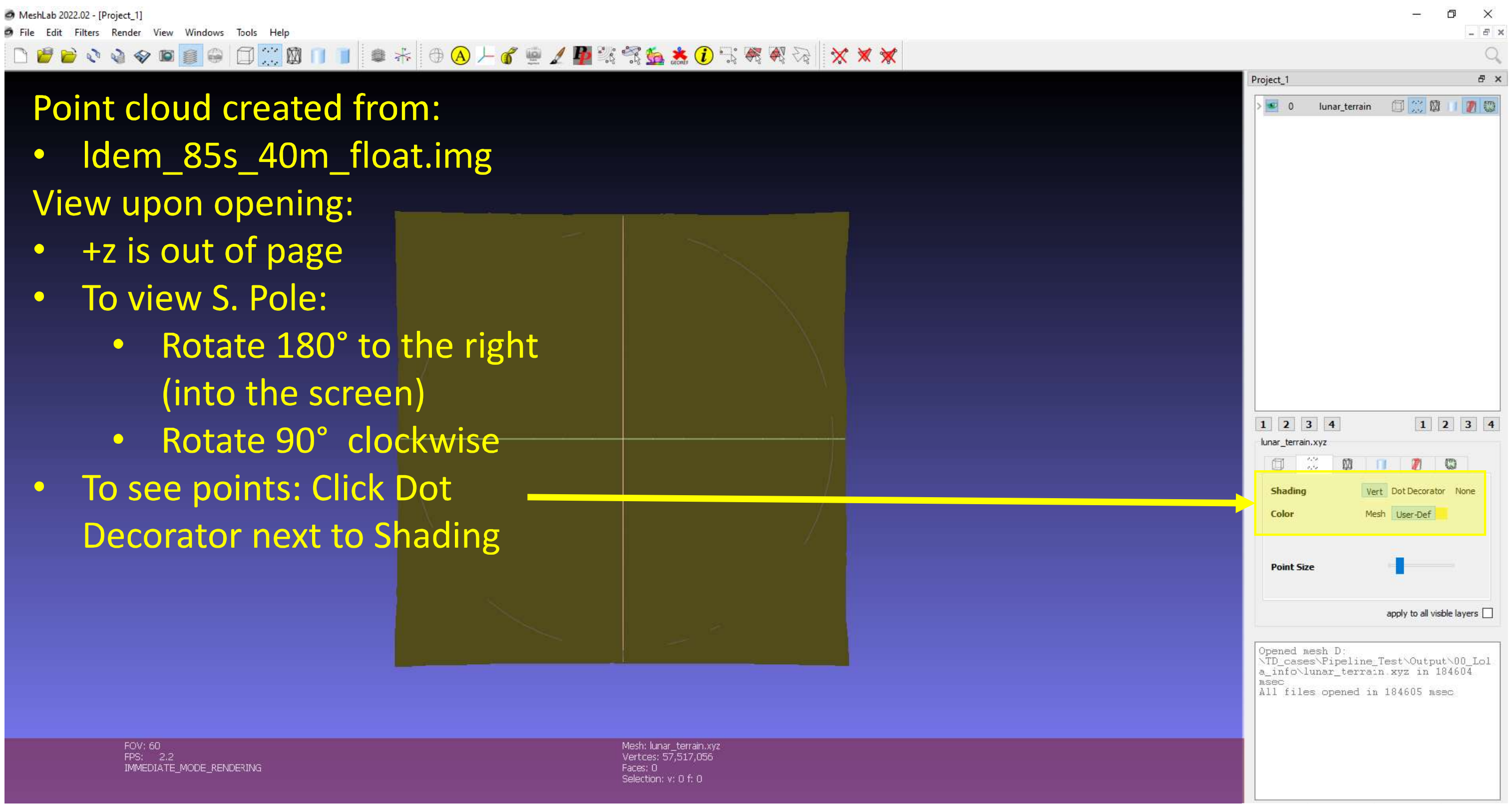
File Edit Format View Help

```
128433.873095 32592.4653867 -1731275.74671
128433.073592 32632.1795374 -1731264.59408
128432.458443 32671.9400888 -1731255.92606
128431.445556 32711.5989535 -1731241.89612
128430.341495 32751.2339385 -1731226.63672
```

```
datas = struct.unpack_from('f'*LINES*LINE_SAMPLES,raw_data)
i = 1 # sample coordinate
j = 1 # line coordinate
for val in datas: #range(0,len(datas)+1):
#  Xmap,Ymap = I,J converted to Cartesian coordinates with 0,0 at center
    Xmap = (i - N/2. - 0.5)*MAP_SCALE
    Ymap = (j - N/2. - 0.5)*MAP_SCALE
    # radius from center of map to pixel I,J in meters
    Rmap = (Xmap**2. + Ymap**2.)**0.5
    # elevation from DEM
    Elevation = val*SCALING_FACTOR*1000 # in meters
    # Convert DEM coordinates to spherical coordinates (S. Hemisphere)
    # (LAT, LON, ro) = ([degrees],[degrees],[m])
    LON = 90+math.atan2(Ymap,Xmap)*180/math.pi
    LAT = -90 + 180/math.pi * 2*math.atan(0.5 * Rmap/1737400)
    ro = (val*SCALING_FACTOR+OFFSET)*1000
    # Convert spherical to cartesian coordinates
    # (x,y,z) = ([m],[m],[m])
    x = ro * math.cos((LAT) * math.pi/180) * math.cos(LON * math.pi/180)
    y = ro * math.cos((LAT) * math.pi/180) * math.sin(LON * math.pi/180)
    z = ro * math.sin((LAT) * math.pi/180)
    output_str = str(x) + " " + str(y) + " " + str(z)+"\n"
    # Only output points within the box
    if (x <= xTL): # xTL = box top left x coordinate
         if (x >= xBR): # xBR = box bottom right x coordinate
            if (y >= yTL): #yTL = box top left y coordinate
                if (y <= yBR): # yTL = box bottom right y coordinate
                    fid_xyz.write(output_str)
    i = i + 1
    # Check if finished sampling a line
    if i == LINE_SAMPLES+1:
        j = j+ 1 # increment line coordinate
        i = 1    # reset sample coordinate
    if i*j == LINES*LINE_SAMPLES:
        print(all lines parsed)
```

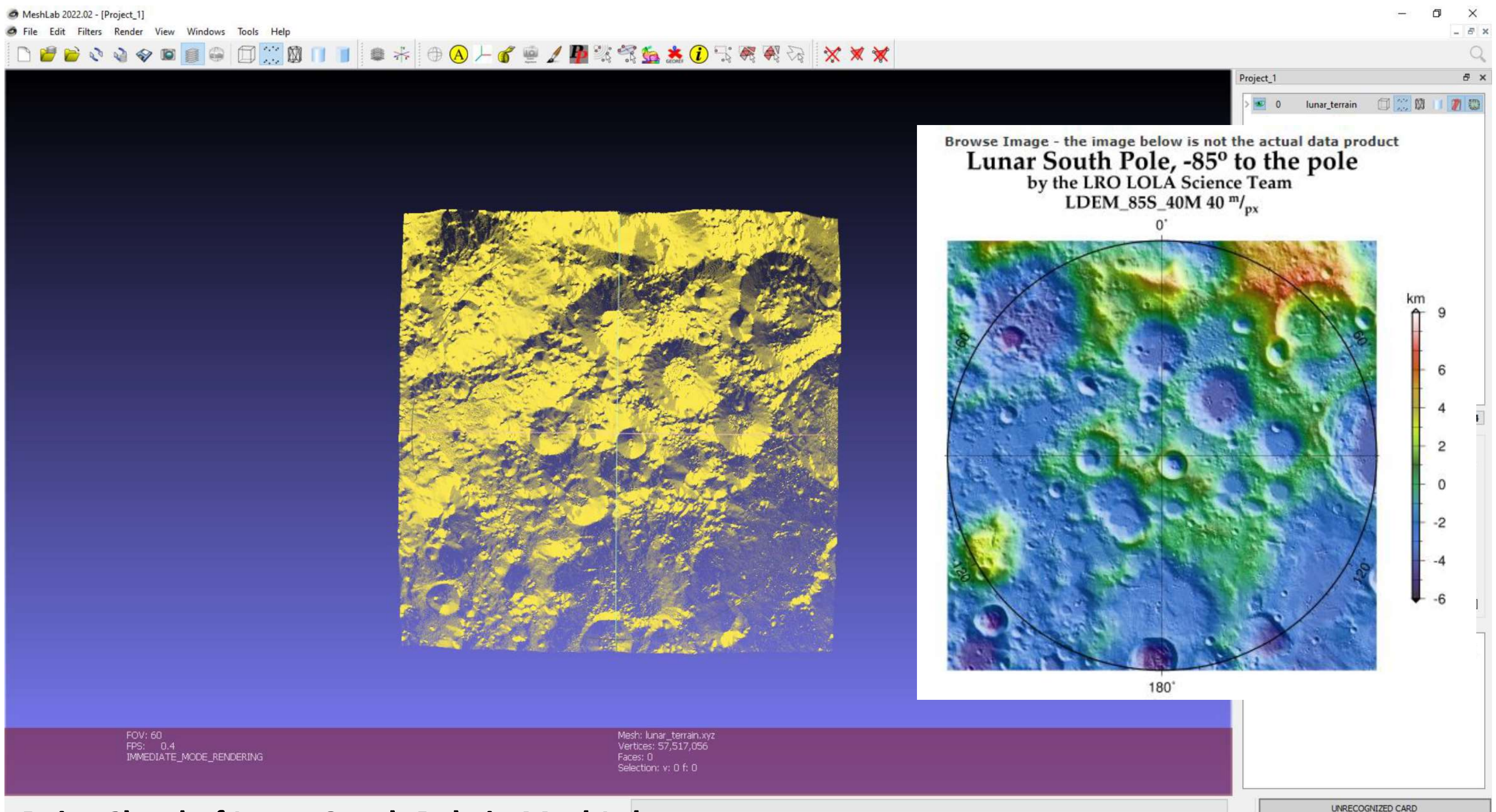
Sample Python code for creating point cloud from DEM

Point Cloud Example: 40m DEM 85S to 90S



Point Cloud of Lunar South Pole in MeshLab

Point Cloud Example: 40m DEM 85S to 90S



Point Cloud of Lunar South Pole in MeshLab

LOLA Gridded Data Record Shape Map 85S 40m https://ode.rsl.wustl.edu/moon/indexproductpage.aspx?product_idgeo=21746067

Step 2: Point Cloud → Triangular mesh

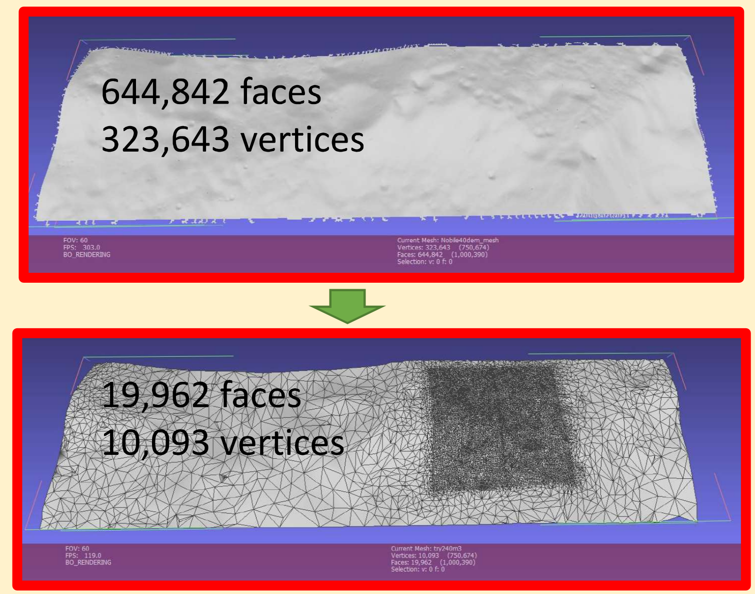
1. Open the lunar_terrain.xyz file in MeshLab.
2. Compute normal of mesh vertices that would be made from the xyz points
   - Filters > Point Set > Compute normal for point sets
   - use default of 10 neighbors
3. Construct the mesh (2 ways to do this)
   - Filters > Remeshing, Simplification, and Reconstruction > Screened Poisson Surface Reconstruction
     - Example settings (gave elements 100m in size from the 40m DEM):
       - Reconstruction Depth: 10 -- Increasing this decreases mesh element size.
       - Min Number of Samples: 1
       - use Default for the rest
   - Filters > Remeshing, Simplification, and Reconstruction > Ball Pivoting
4. Mesh Reduction (simplify/cleanup mesh)
5. Export resulting mesh to an .stl file
   - File> Export Mesh As
     - Unselect Materialize Color Encoding

Mesh reduction

1. If meshed with Poisson Surface Reconstruction remove the extra triangles made when constructing mesh
   - Filters > Selection > Select Faces with edges longer than .... use ~150 or 200
   - Click on triangle with the x over it to remove these faces
2. Clean up edges
   - Filters > Cleaning and Repairing > Remove isolated pieces (wrt Face Num.)
3. Mesh reduction options
   - Filters > Remeshing Simplification and Reconstruction > Simplification: clustering dissemination
     - Best to do this incrementally from outside in, detailed mesh will be in the center
   - Filters > Remeshing Simplification and Reconstruction > Quadratic Edge Collapse Decimation
     - MeshLab will only apply this to whole mesh
   - Remove
4. Locally increasing mesh density
   - Filters > Remeshing Simplification and Reconstruction > Subdivision Surfaces: Midpoint

Mesh reduction

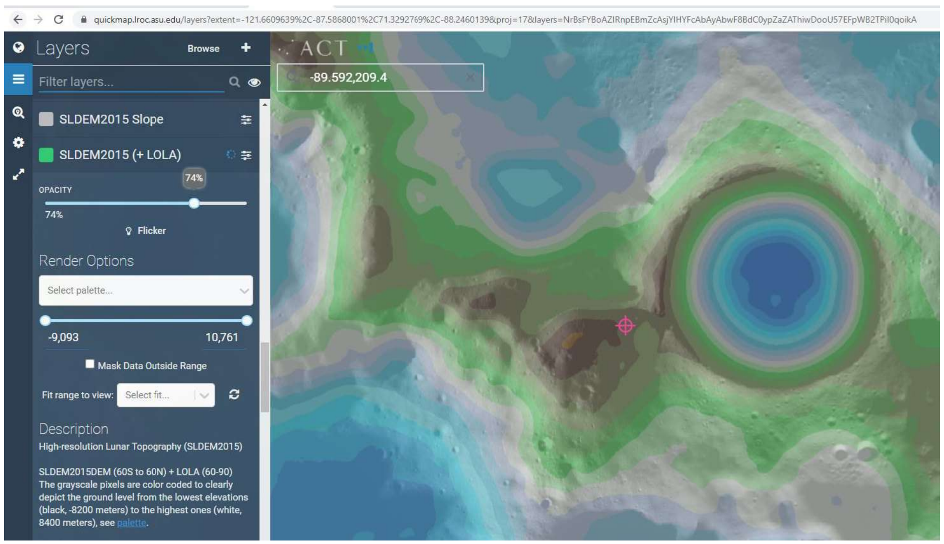
1. If meshed with Poisson Surface Reconstruct made when constructing mesh
    - Filters > Selection > Select Faces with edges longer
    - Click on triangle with the x over it to remove these
2. Clean up edges
    - Filters > Cleaning and Repairing > Remove isolated
3. Mesh reduction options
    - Filters > Remeshing Simplification and Reconstruction > Simplification: clustering dissemination
        - **Best to do this incrementally from inside out, detailed mesh will be in the center**
    - Filters > Remeshing Simplification and Reconstruction > Quadratic Edge Collapse Decimation
        - MeshLab will only apply this to whole mesh
4. Locally increasing mesh density
    - Filters > Remeshing Simplification and Reconstruction > Subdivision Surfaces: Midpoint



Example mesh reduction

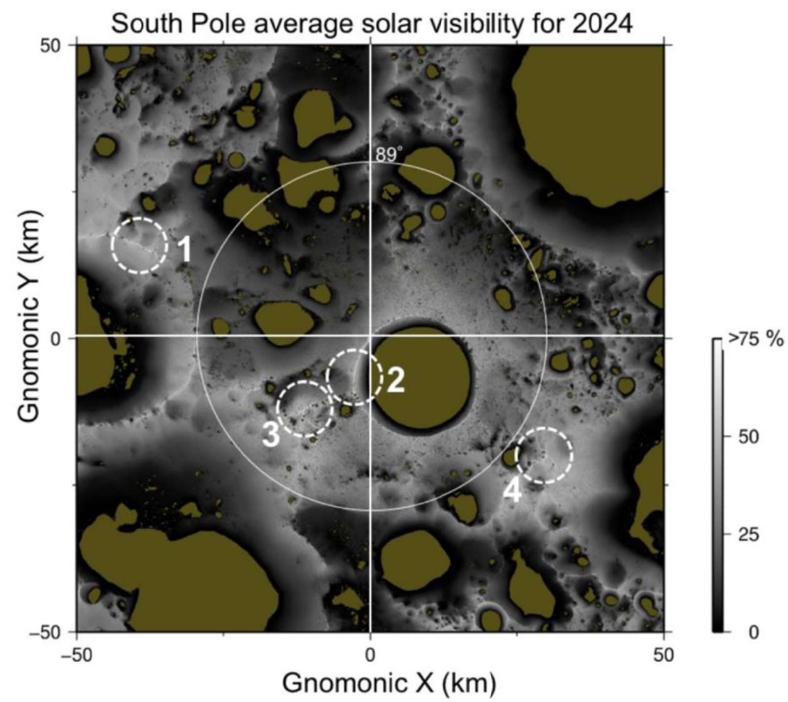
Example: Point Cloud → Triangular mesh

- Area around 89.592S and 209.4E (-150.6W)[1]



Example location shown in "Lunar QuickMap" website

https://quickmap.lroc.asu.edu/query?extent=-121.0589468%2C-88.5101578%2C72.6825193%2C-89.1820632&proj=17&features=209.40000000%2C-89.59200000&selected=0&layers=NrBsFYBoAZIRnpEBmZcAsjYIHYFcAbAyAbwF8BdJAdgTpgDoAOAJknyNICIAnAQxwBzAKYAFPgWEAXKcK4AuLgHcAxgHsCeKdC6Uq5KsGTRkAThhYGUFHGOIE0BtVDtCxA-r1A



Landing Site Regions of Interest on Lunar surface[2]

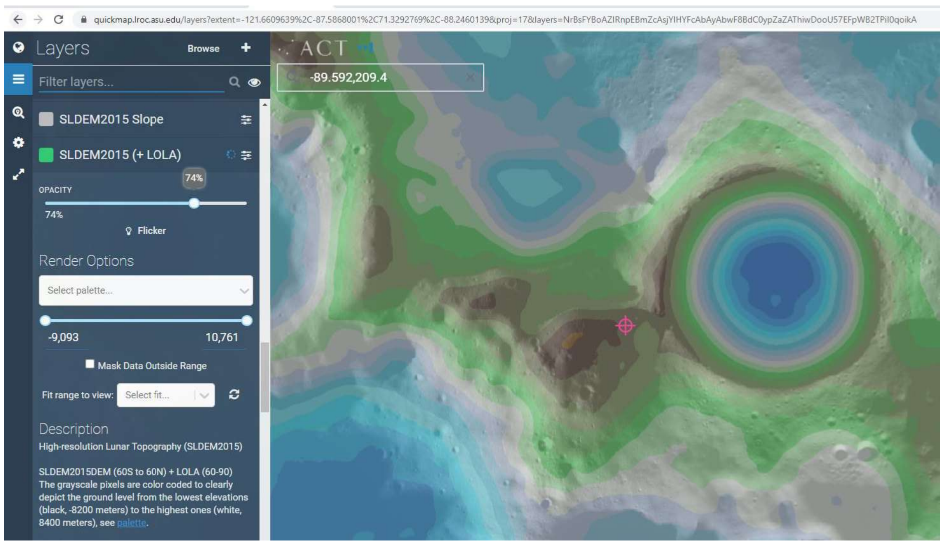
[1] Location considered by "Optimized traverse planning for future polar prospectors based on lunar topography" https://ttu-ir.tdl.org/bitstream/handle/2346/86274/ICES-2020-288.pdf?sequence=1&isAllowed=y
[2] M. Smith, "Human Lunar Exploration" NASA Advisory Council Human Exploration & Operations Committee, May 2019, https://www.nasa.gov/sites/default/files/atoms/files/20190528-nac-heoc-smith-v5b.pdf

Start Demonstration

The next few slides summarize the demonstration which gives an example on how to create a lunar surface mesh from a point cloud in MeshLab.

Example: Point Cloud → Triangular mesh

- Area around 89.592S and 209.4E (-150.6W)



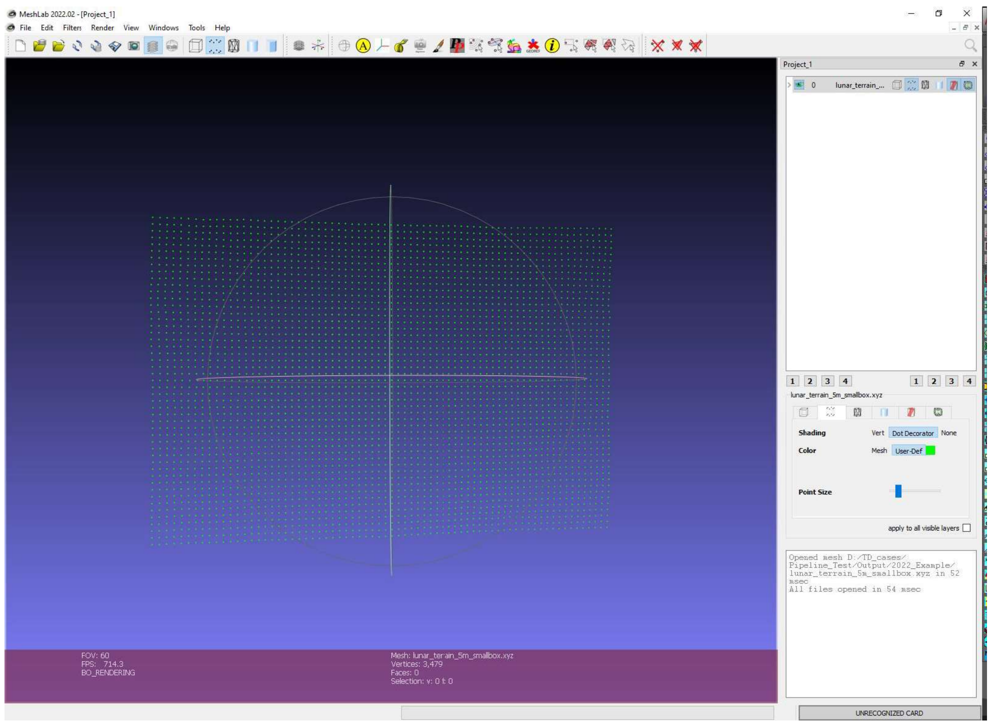
Example location shown in "Lunar QuickMap" website

https://quickmap.lroc.asu.edu/query?extent=-121.0589468%2C-88.5101578%2C72.6825193%2C-89.1820632&proj=17&features=209.40000000%2C-89.59200000&selected=0&layers=NrBsFYBoAZIRnpEBmZcAsjYIHYFcAbAyAbwF8BdJAdgTpgDoAOAJknyNICIAnAQxwBzAKYAFPgWEAXKcK4AuLgHcAxgHsCeKdC6Uq5KsGTRkAThhYGUFHGOIE0BtVDtCxA-r1A

For simplicity limit the box size:
top_left = [-89.59183, -149.59296
top_right = [-89.59775, -151.06213]
bottom_right = [-89.59053, -151.58538
bottom_left = [-89.58441, -150.12331]

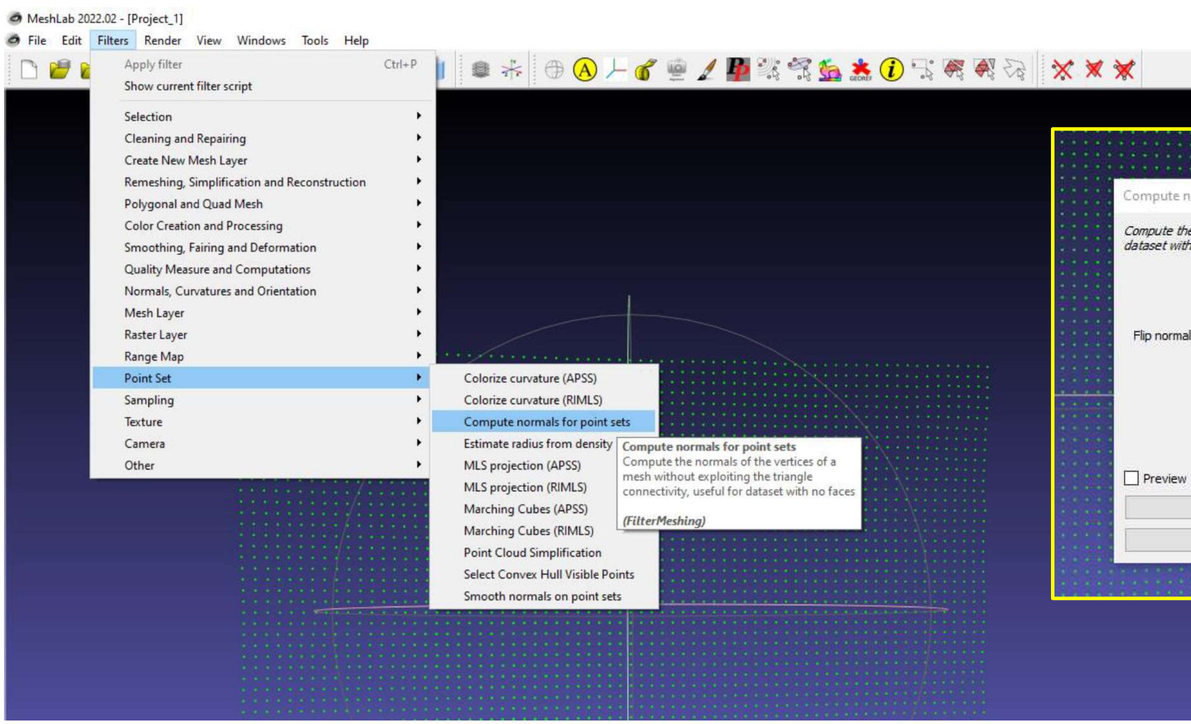
Example: Point Cloud → Triangular mesh

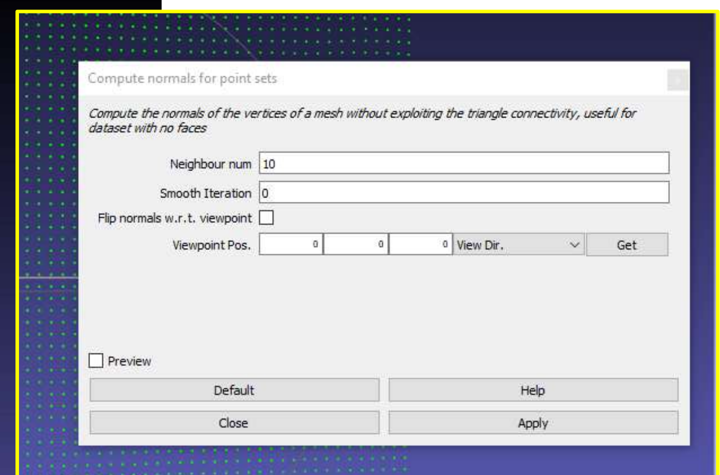
1. Open the lunar_terrain.xyz file in MeshLab.
    - File > Import Mesh > Select .xyz file
    - Note +z points up. For S. Pole locations flip image 180° then rotate 90° clockwise



Example: Point Cloud → Triangular mesh

2. Compute normals of mesh vertices
    - Filters > Point Set > Compute normal for point sets



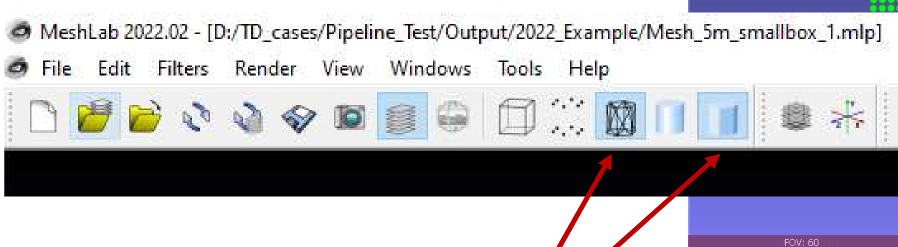


Example: Point Cloud → Triangular mesh

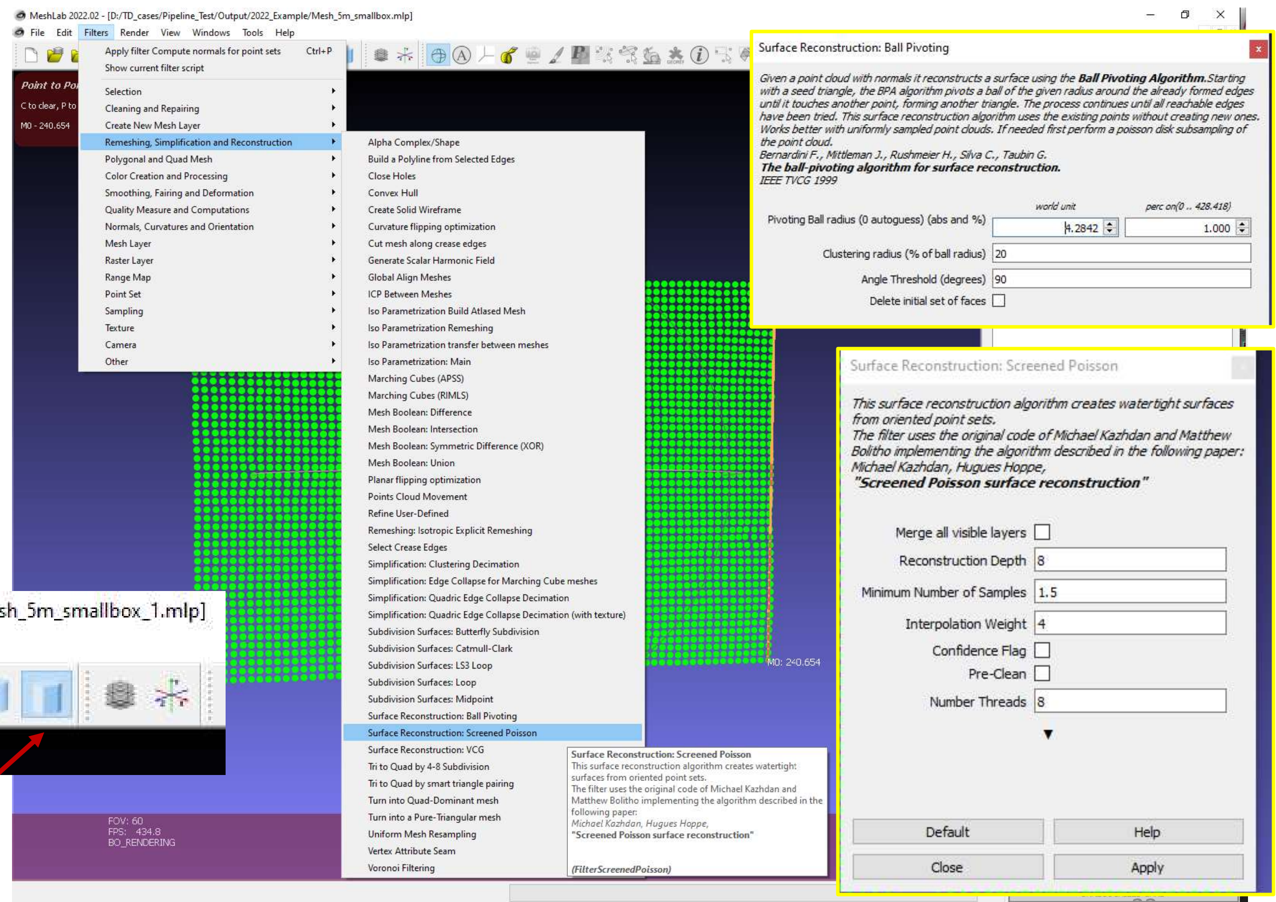
3. Construct the mesh
   - Filters > Remeshing, Simplification, and Reconstruction > Screened Poisson Surface Reconstruction

   <u>OR</u>

   - Filters > Remeshing, Simplification, and Reconstruction > Ball Pivoting

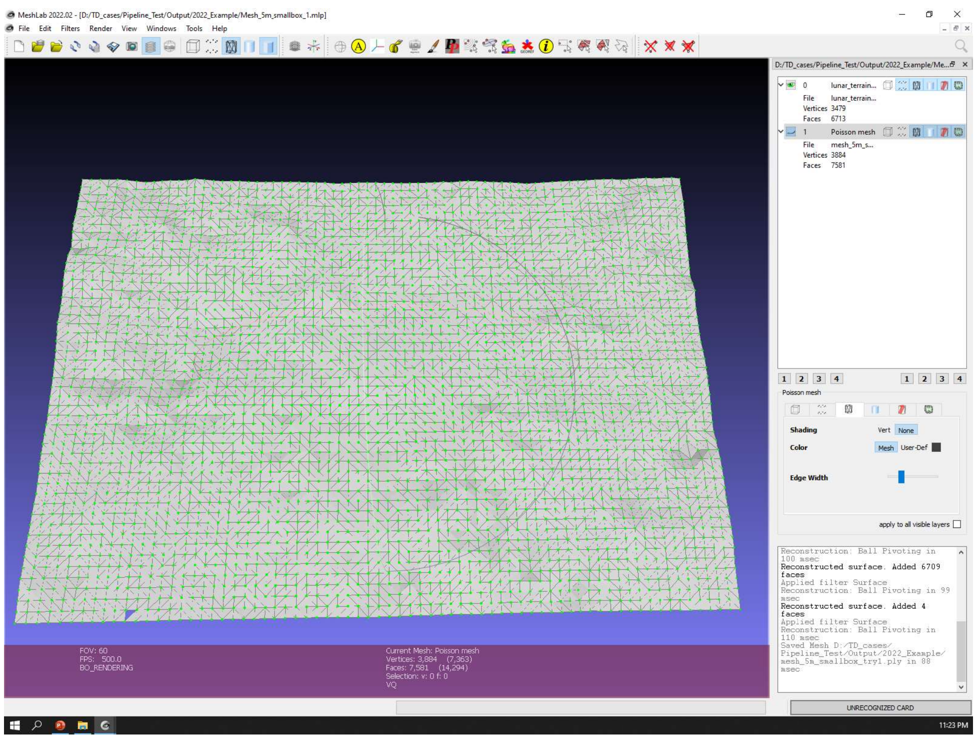


Toggle these buttons to show mesh

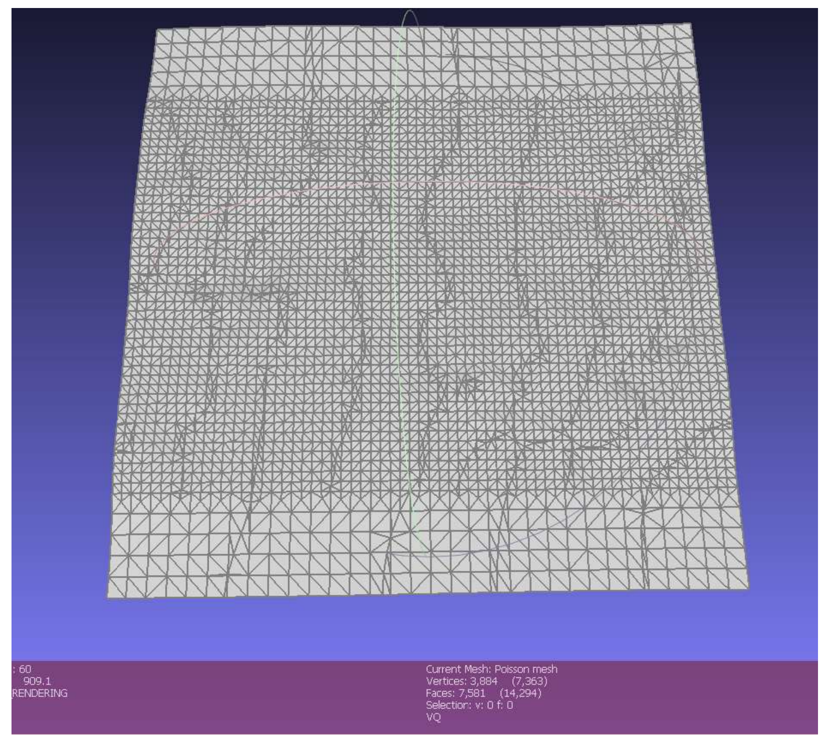


Example: Point Cloud → Triangular mesh

- Ball Pivoting

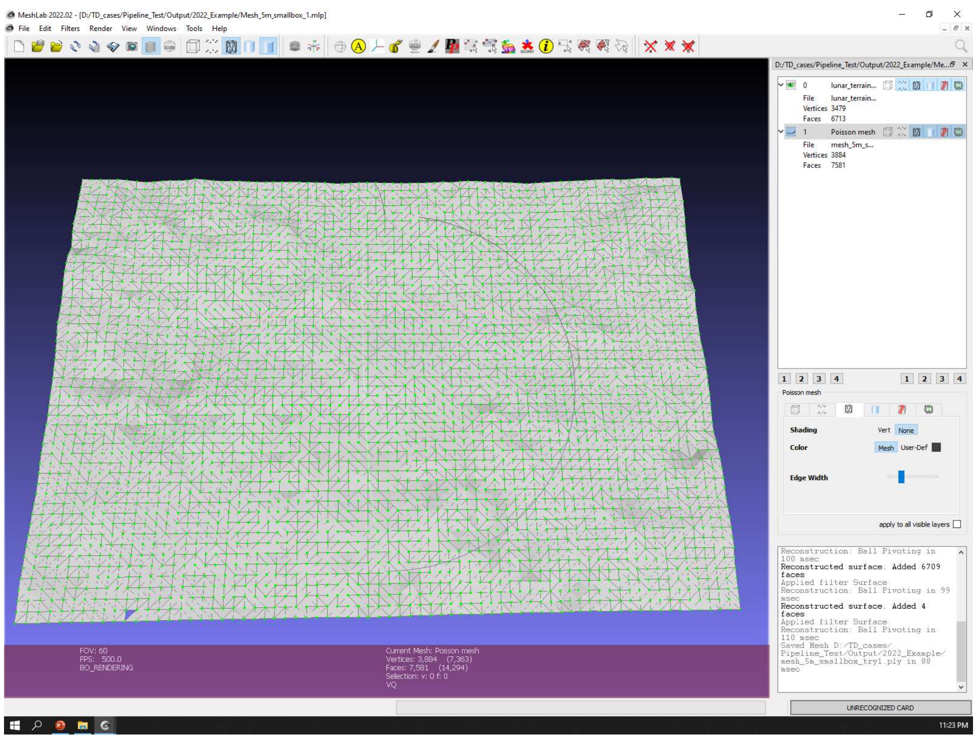


- Screened Poisson Surface Reconstruction

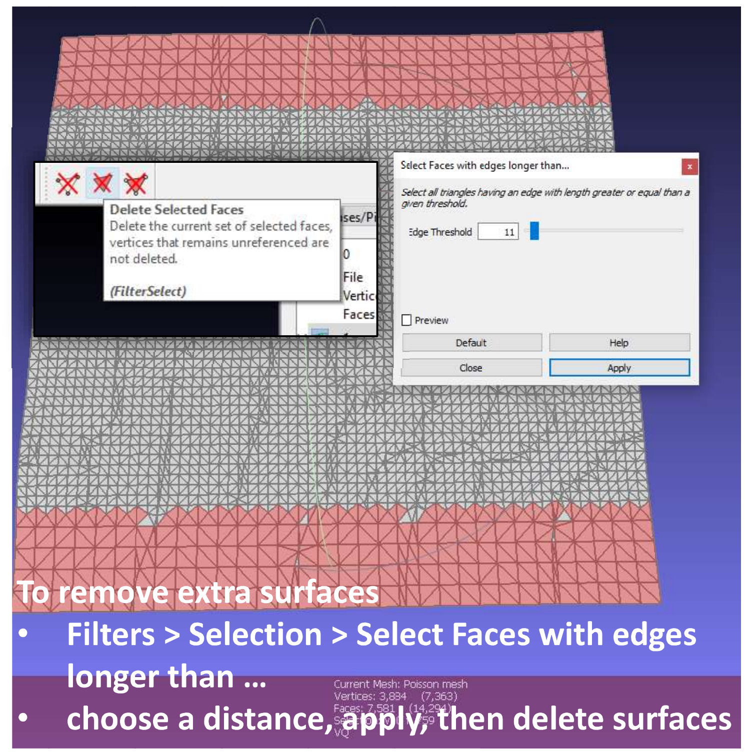


Example: Point Cloud → Triangular mesh

- Ball Pivoting

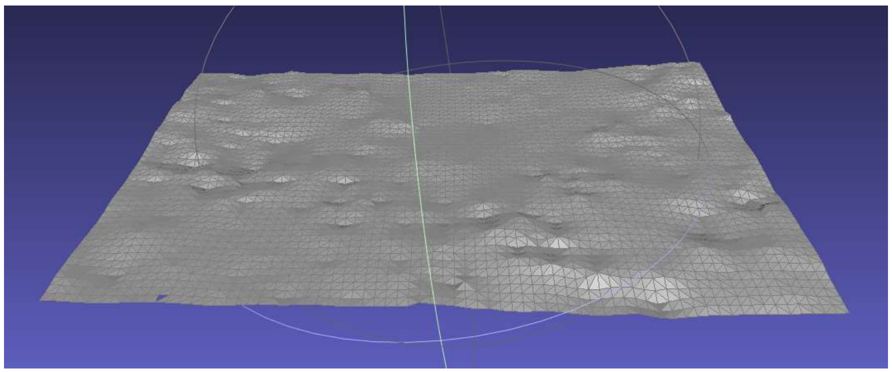


- Screened Poisson Surface Reconstruction

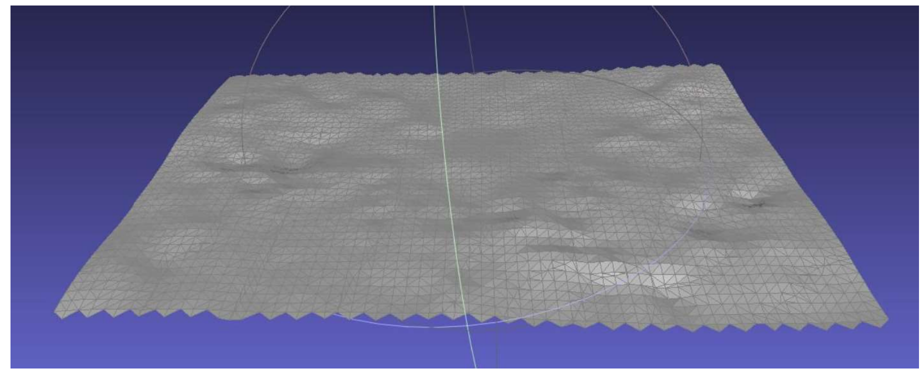


Example: Point Cloud → Triangular mesh

- Ball Pivoting



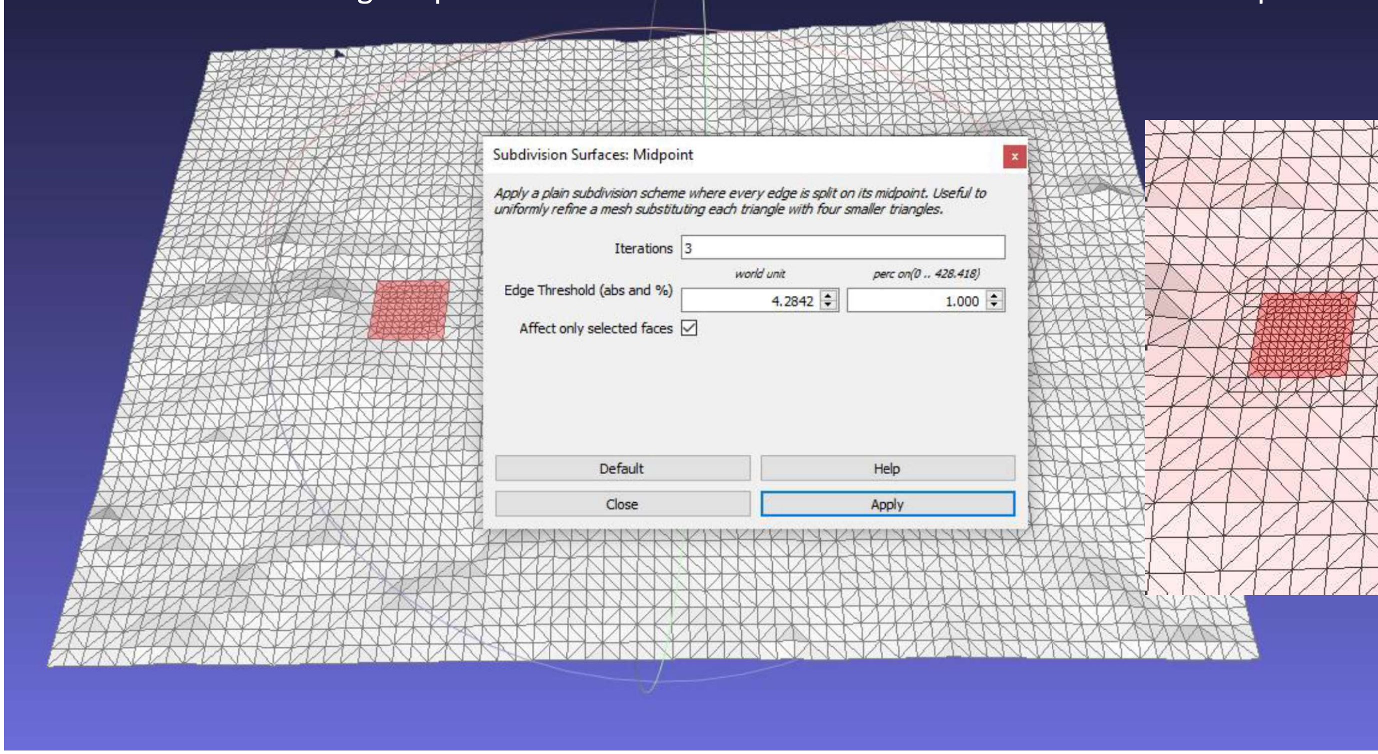
- Screened Poisson Surface Reconstruction

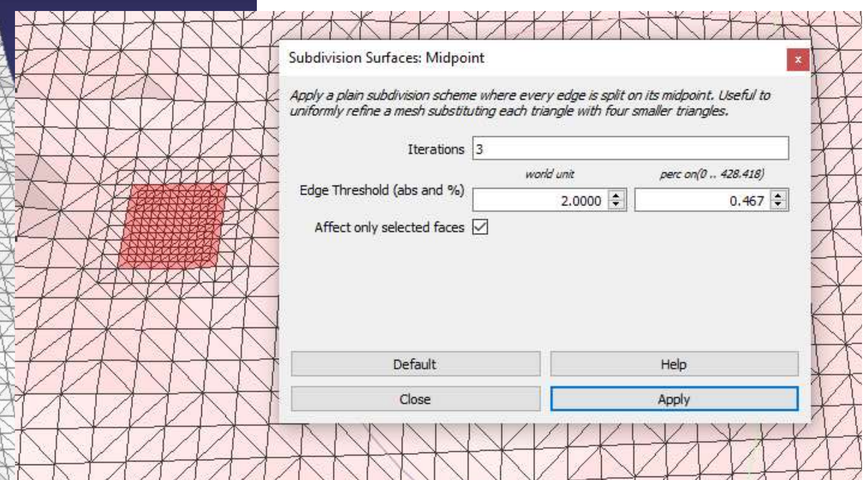


Example: Point Cloud → Triangular mesh

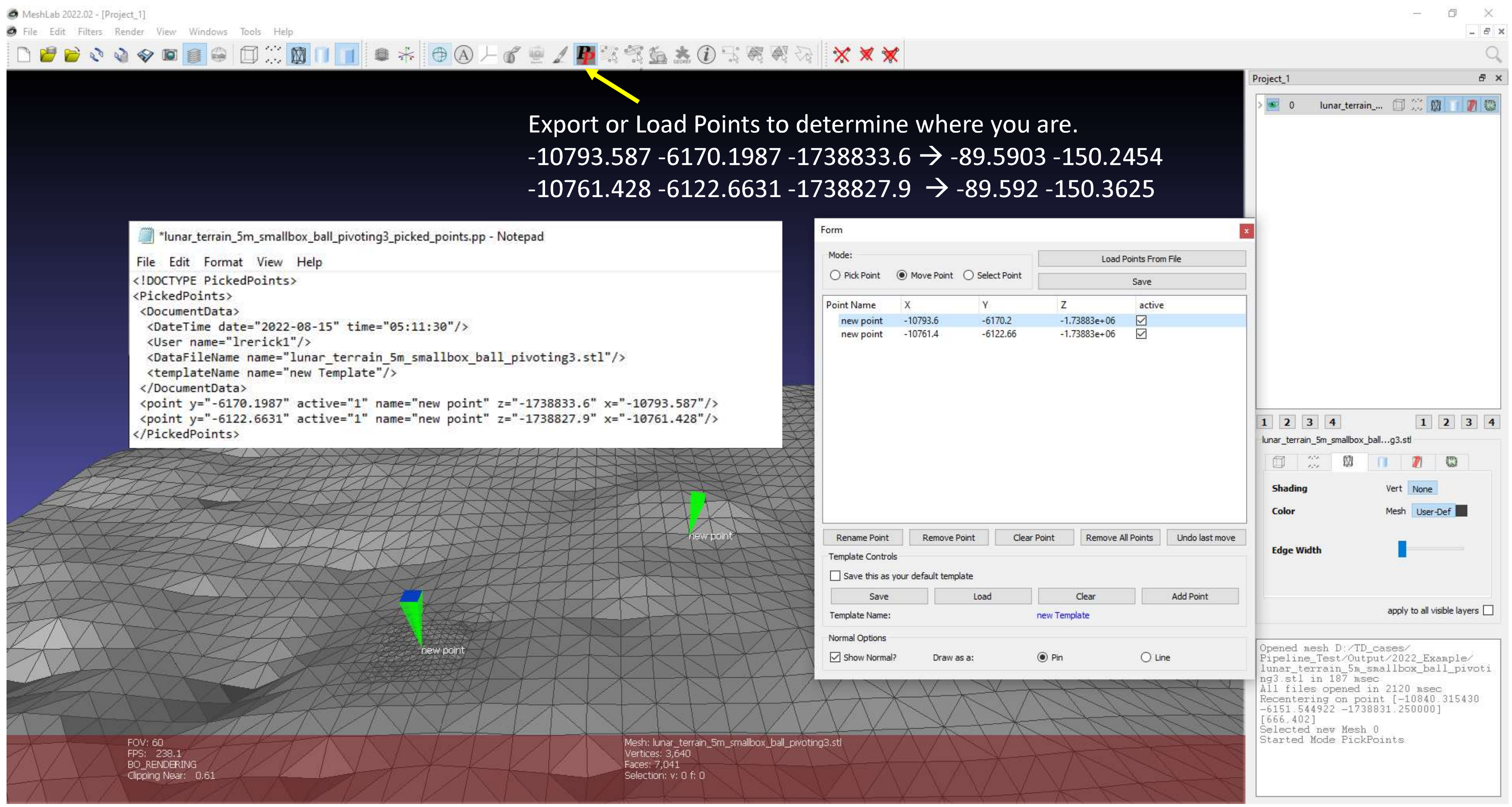
Increase mesh density in a particular area

- Filters > Remeshing Simplification and Reconstruction > Subdivision Surfaces: Midpoint





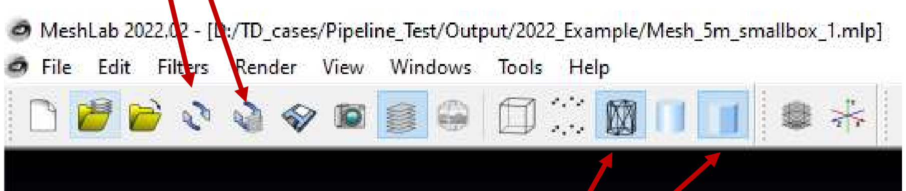
Example: Point Cloud → Triangular mesh



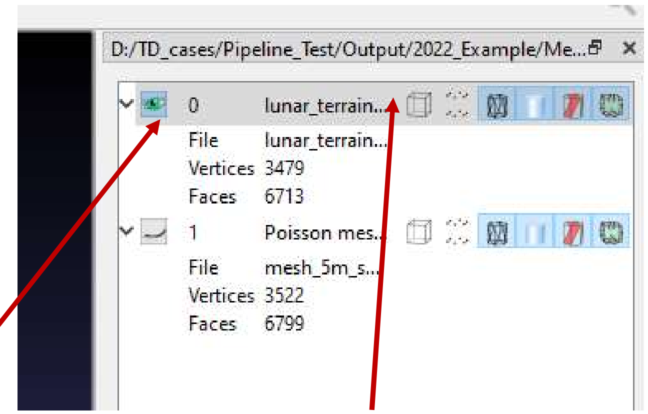
Example: Point Cloud → Triangular mesh

3. Export mesh:
   - File > Export mesh as
   - Select .stl extensions

Reload last saved layer (instead of undo)



Toggle these buttons to show mesh



Show or hide mesh

Operations (e.g., filtering & selection will be performed on highlighted one)

End Demonstration

Import Mesh into Thermal Desktop

- Three steps:
  1. Convert .stl file to a list of mesh vectors (that describe each triangle)
  2. Run an AutoCAD VBA script to create mesh elements from these vectors
     - Requires AutoCAD VBA module to be installed
       - https://knowledge.autodesk.com/support/autocad/downloads/caas/downloads/content/download-the-microsoft-vba-module-for-autocad.html
  3. Convert AutoCAD mesh to TD mesh elements and add surface properties

See: Human Landing System Lunar Thermal Analysis Guidebook Section 9.5.1.3: Importing Meshes https://ntrs.nasa.gov/api/citations/20210010030/downloads/HLS-UG-001%20Lunar%20Thermal%20Analysis%20Guidebook%20Baseline_STI.pdf

Step 1: Convert .stl file to Mesh Vectors

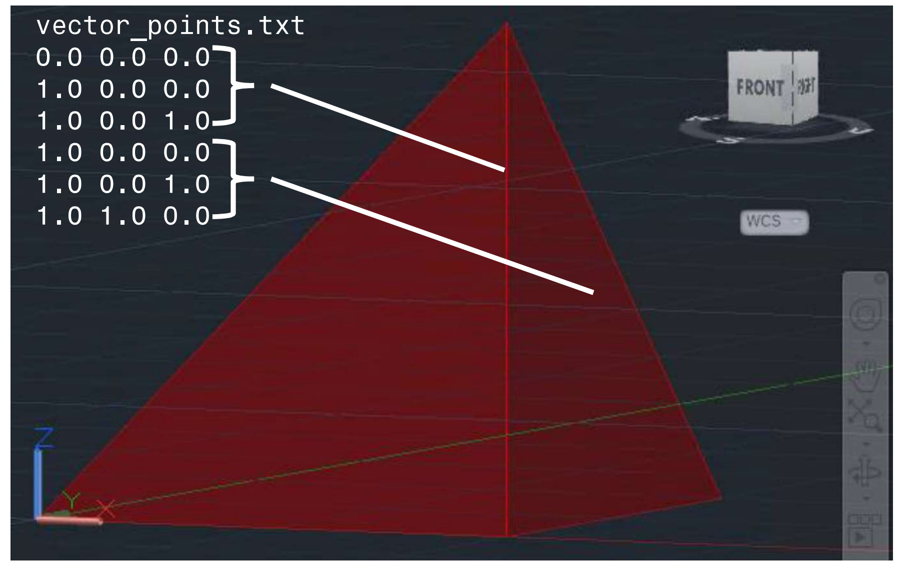
- Each triangular surface is described by 3 vectors
  - Each vector is defined its vertices
- Goal:
  - Write vectors to a .txt file, which will be read by AutoCAD in step 2
- Use Python package numpy-stl to
  - extract vectors from the .stl file
  - output vectors to vector_points.txt

vector_points.txt - Notepad

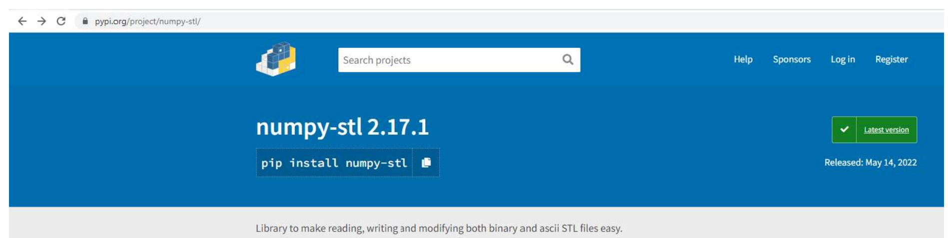
File Edit Format View Help

```
-10676.5205 -6257.787 -1738847.1
-10676.525 -6262.7944 -1738848.0
-10681.525 -6257.787 -1738847.1
```

Format:
x y z for point_1 of Face_n
x y z for point_2 of Face_n
x y z for point_3 of Face_n



AutoCAD Surface Elements



numpy-stl https://pypi.org/project/numpy-stl/

Step 1: Convert .stl file to Mesh Vectors

Example method to create vector_points.txt:

s_mesh = mesh object

s_mesh.vectors contains a list of all mesh vectors

an individual vector can be accessed with the object s_mesh.vectors[i][j][k]

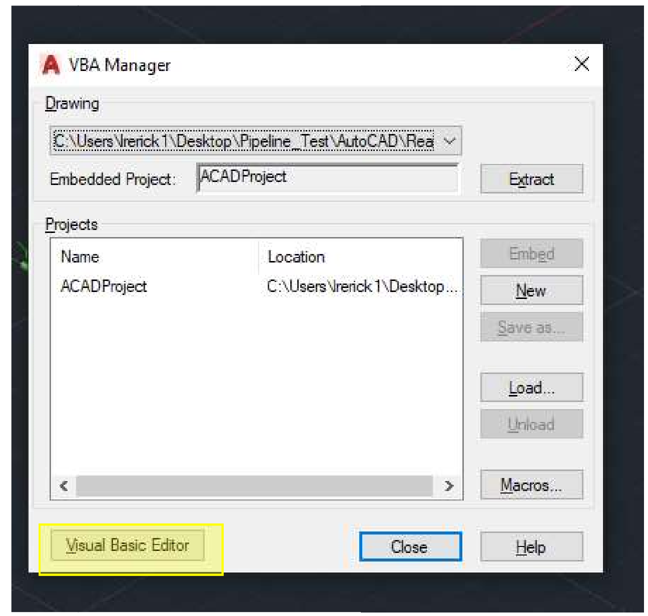
```
# Installed modules.
import numpy as np
from stl import mesh

# Read in the surface mesh.
   s_mesh = mesh.Mesh.from_file(input_stl_full)

# Generate output text file.
    f = open(filepath, 'w')

    # Iterate through vectors of .stl file
    for i in range(0,len(s_mesh.vectors)):
        # Iterate through 3 points of a vector
        for j in range(0,len(s_mesh.vectors[0])):
            # Iterate through x y z of each point
            outstr='' # initialize output string
            for k in range (0,len(s_mesh.vectors[0][0])):
                outstr=outstr+str(s_mesh.vectors[i][j][k])
                if k<(len(s_mesh.vectors[0][0])-1):
                    outstr=outstr+' '
                else:
                    outstr=outstr +'\n'
            # Write to output file
            f.write(outstr)
    f.close()
```

Python code outputting mesh vectors

Step 2: Create AutoCAD surface mesh

- To open VBA editor:
  - Type VBAMAN
  - Click Visual Basic Editor

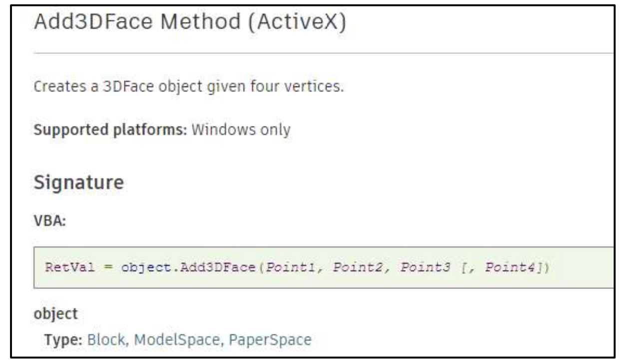
AutoCAD file with VBA code will prompt to enable macros



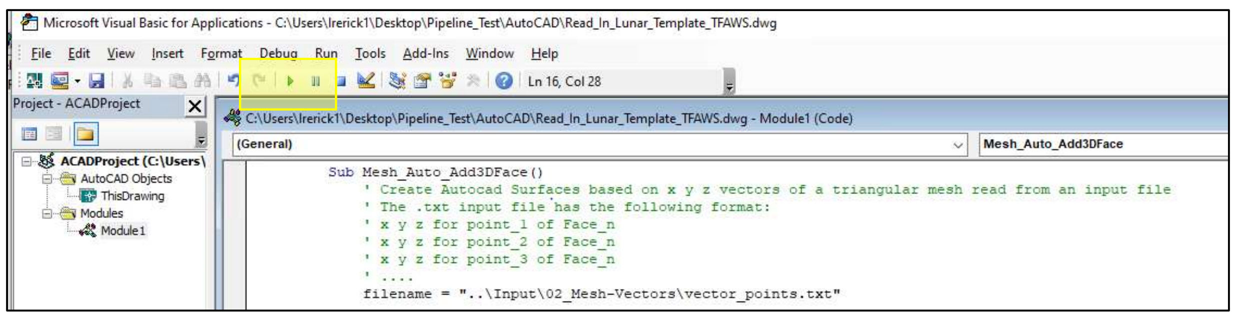
AutoCAD VBA Manager

Step 2: Create AutoCAD surface mesh

- Example code to create surface mesh is given in the Human Landing System Lunar Thermal Analysis Guidebook
  - Search for: Mesh_Auto_Add3DFace
  - Filename = sets input directory
  - Add3DFace(pt0, pt1, pt2, pt2) creates the triangular element
    - For a Quad element would input Add3DFace(pt0, pt1, pt2, pt3)
- In the VBA editor press the green play button to run the script



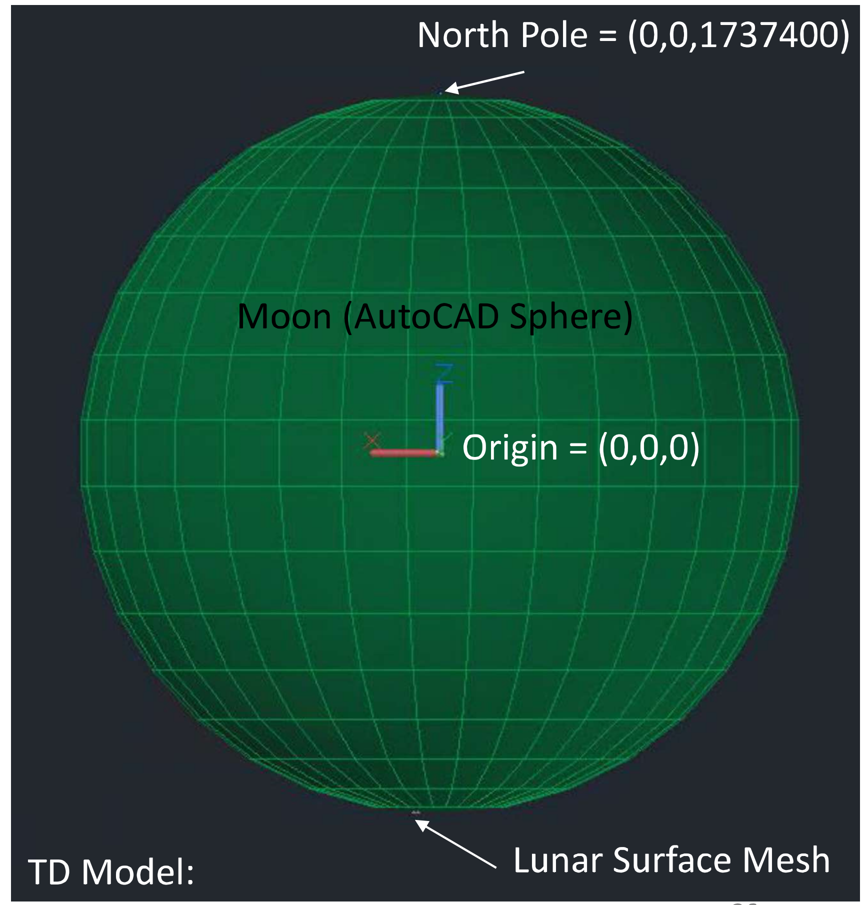
AutoCAD Add3Face Method help:
https://help.autodesk.com/view/OARX/2022/ENU/?guid=GUID-E125323B-7DC2-4174-AEA9-DAAEA0E683D9



AutoCAD VBA editor

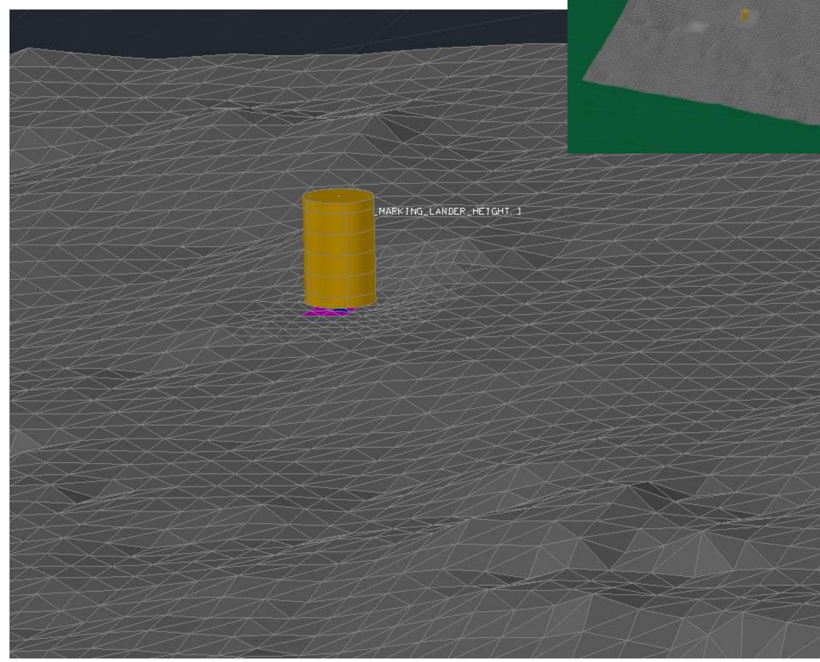
Step 3: Convert AutoCAD mesh to TD mesh

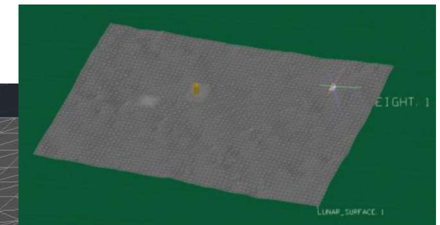
- Highlight the ground the select:
  - Thermal > FD/FEM Network > Convert AutoCAD Surface to Nodes/Elements
- Zoom in, mesh will be far away from origin
  - Show a node label if needed to see it
- Use properties described earlier



Step 4: Placing object on surface

- A simple approach for static objects:
  - Attach your object to an articulator placed at the center of a triangle
  - Calculate normal vector out of triangle's surface
    - $\vec{n} = \overrightarrow{AB} \times \overrightarrow{AC}$
    - Where $\overrightarrow{AB}$ and $\overrightarrow{AC}$ are vectors describing 2 of the triangle's edges
    - Use right hand rule to pick them such that $\vec{n}$ points in the desired direction
  - Calculate point x distance away from triangle's center along normal vector
  - Place a temporary node at that point
  - Grab +z articulator grip point and move it to the temporary node.
- Alternative approach:
  - Place object on a square
  - Attach center of square to articulator
  - Specify articular rotations
    - Order of rotations matching TD articulators:
      - Rotate about z axis
      - Rotate about transformed y axis
      - Rotate about transformed x axis

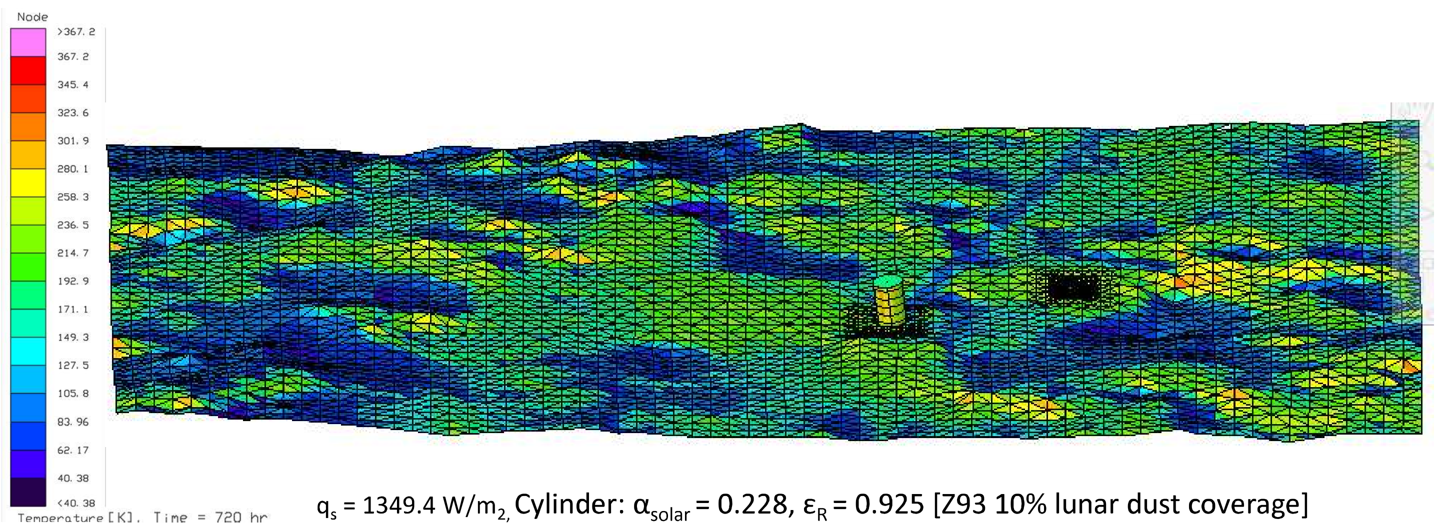




Example: Cylinder 1 m above meshed terrain (h=12 m, r=4 m)

Example: Cylinder on Meshed Terrain

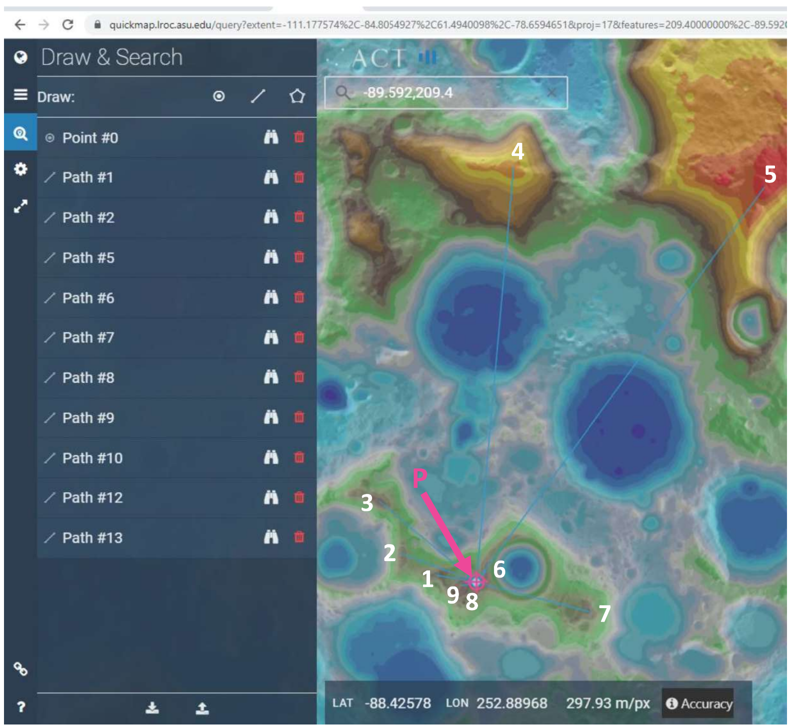
- Area around 89.592S and 209.4E (-150.6W)
- Orbit: 2010-Sep-21 00:00 to 2010-Oct-21 00:00
- Ground is initialized for 5 years prior to run



q_s = 1349.4 W/m_2, Cylinder: α_{solar} = 0.228, ε_R = 0.925 [Z93 10% lunar dust coverage]

Example: Effect of Far-field Terrain

- Used geometry to check if terrain features could shadow terrain at point P



- Predict if an target object at point P could be shadowed by a blocking object at point P_H located θ degrees away.
  - Estimate θ as difference between latitudes, assuming at some point the target and blocking objects are aligned.
- Know radius of moon, r, height of blocking object, H, height of target object H_T (e.g. terrain elevation at P), solar elevation angle, α, and θ.
- Calculate height of shadow at point P, y.
  - Shadow cast by H at point P_H = shadow cast by H_f on a flat plane at a distance d away from target point P.
- Note: Sunlight is imperfectly collimated (±0.5°), to 'correct' +0.5° to max solar elevation angle at P.

By Trig relations:

$z = H\sin(\alpha)\tan(\alpha)$

$h_f = h_p - c - z$ estimated flat plane height

$d = r\sin(\theta)$ distance from P_H to P

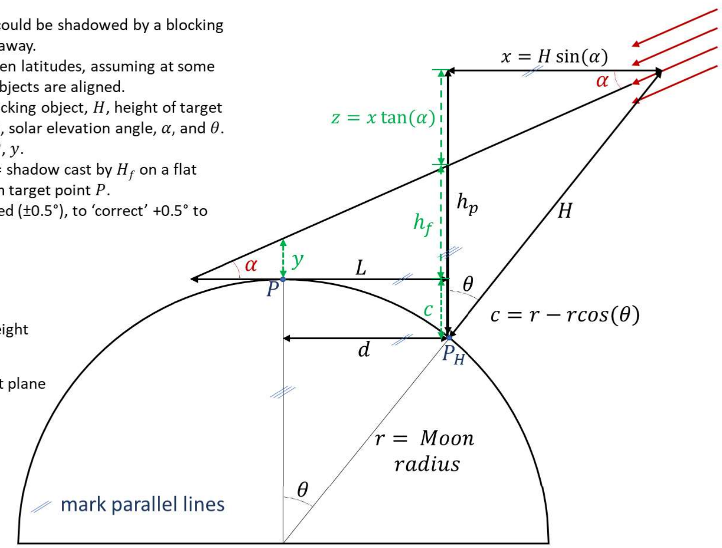
$L = h_f/\tan(\alpha)$ length of shadow on flat plane

By similar triangles:

$\frac{y}{h_f} = \frac{L-d}{L}$

$y = \left(\frac{L-d}{L}\right)h_f$

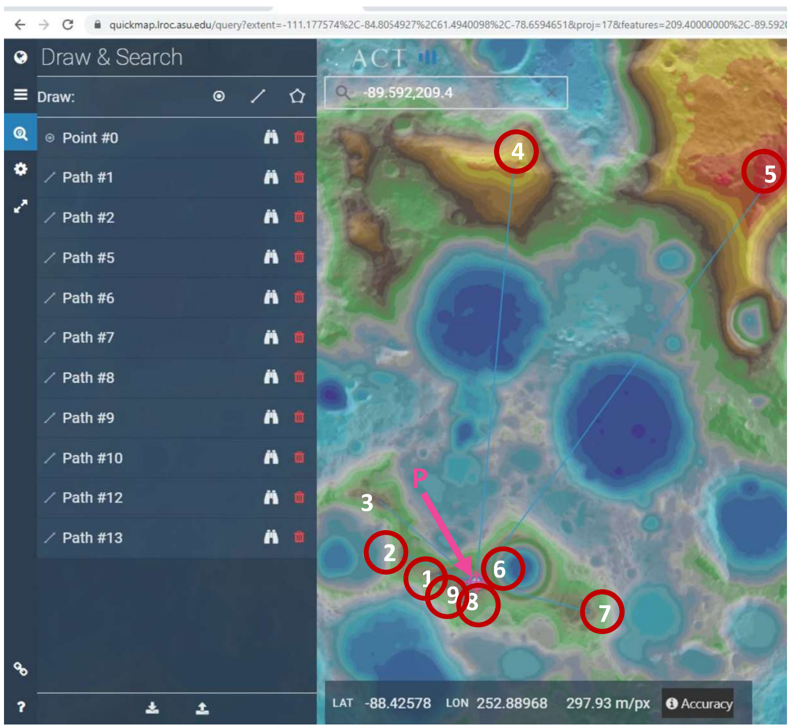
If $y > H_T$ target is shadowed



Note: sunlight is imperfectly collimated (±0.5°), this estimate can be improved by checking for shadowing from a range of elevation angles $0 < \alpha \leq \alpha max + 0.5$.
Time in shadow may be short.
Section 9.3.2 "Human Landing System Thermal Analysis Gradebook" HLS-UG-001

Example: Effect of Far-field Terrain

- Used geometry to check if terrain features could shadow terrain at point P

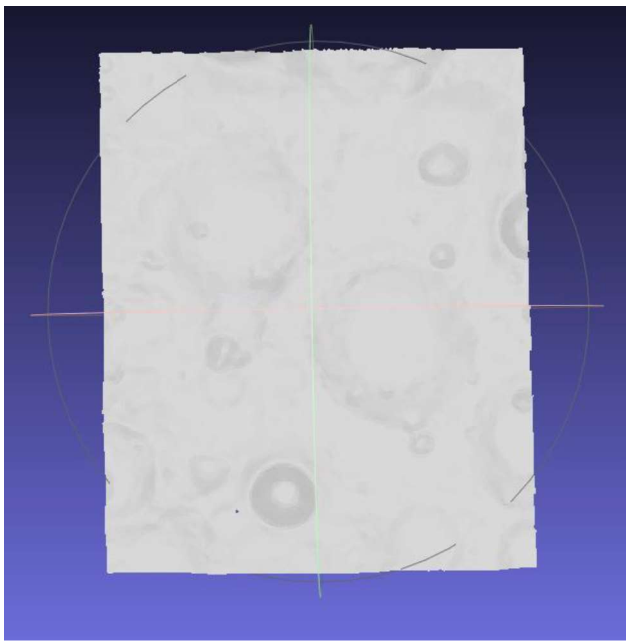


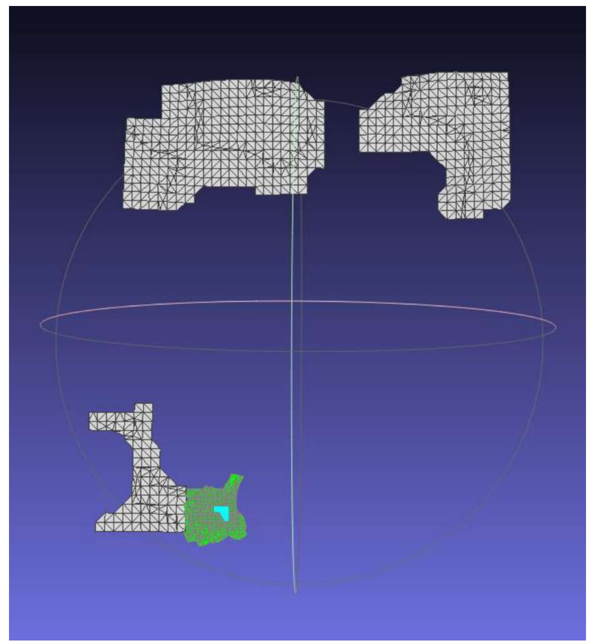
case	Target Lat	Block Lon	Target Lat	Block Lon	H_T [m]	H [m]	Target location shadowed?
1	-89.592	209.4	-89.4591	-137.052	1465	5246	yes
2	-89.592	209.4	-89.3081	-113.824	1465	1691	yes for elevation <1.36°
3	-89.592	209.4	-89.0099	-92.9862	1465	1425	No
4	-89.592	209.4	-88.6715	-67.9055	1465	1798	yes for elevation <0.22°
5	-89.592	209.4	-85.9757	2.72545	1465	5140	yes for elevation <0.11°
6	-89.592	209.4	-85.2456	36.6487	1465	6749	yes for elevation <0.12°
7	-89.592	209.4	-89.7476	-157.942	1465	1996	Yes
8	-89.592	209.4	-88.8017	123.906	1465	1641	yes for elevation <0.03°
9	-89.592	209.4	-89.5403	-150.05	1465	1646	yes

Total days where elevation angle is ≤ 0° = 169
Total days where elevation angle is ≤ 0.22 and > 0° = 190
Closer hills could negate impact of farther hills

Example: Effect of Far-field Terrain

- Meshed large terrain & removed elements at low elevations

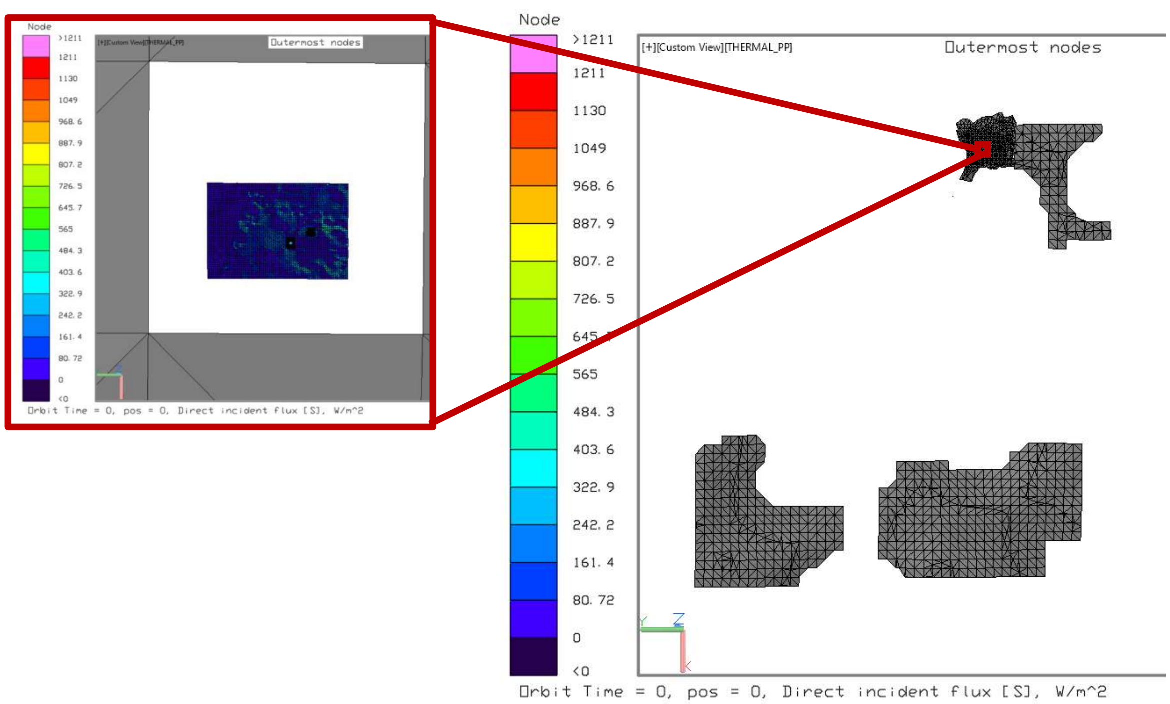




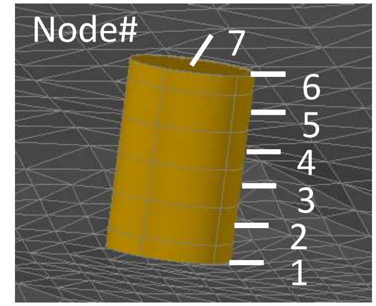
LAT/LON Bounds:
top_right = [-84.76164, 35.20046, 6116.21] top_left = [-85.49455, -19.77071, 2310.69]
bottom_left = [-87.93420, -120.80487,-865.08] bottom_right = [-87.05400, 110.14100,-1390.71]

Example: Effect of Far-field Terrain

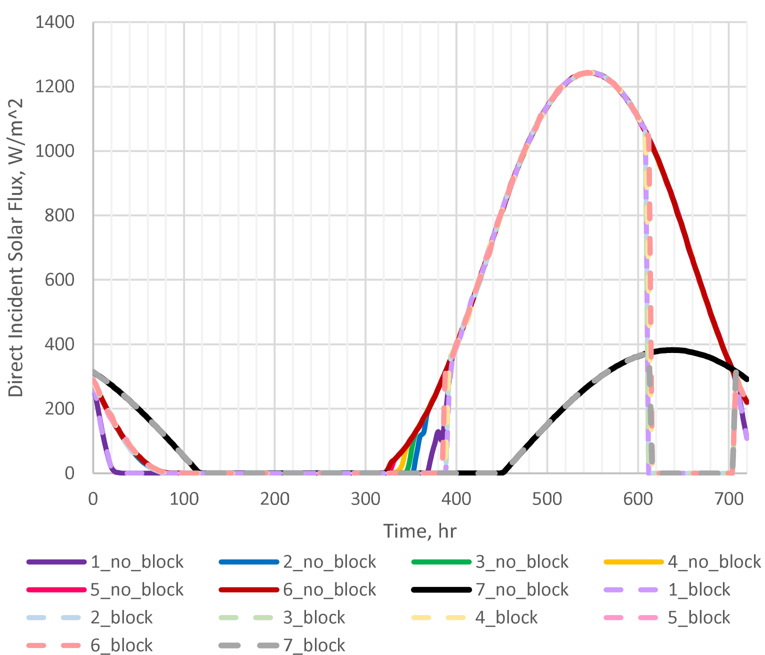
- Added far-field terrain to TD model
  - set active = none to act as a reflector/blocker
- Compared results with and without far-field terrain blockers



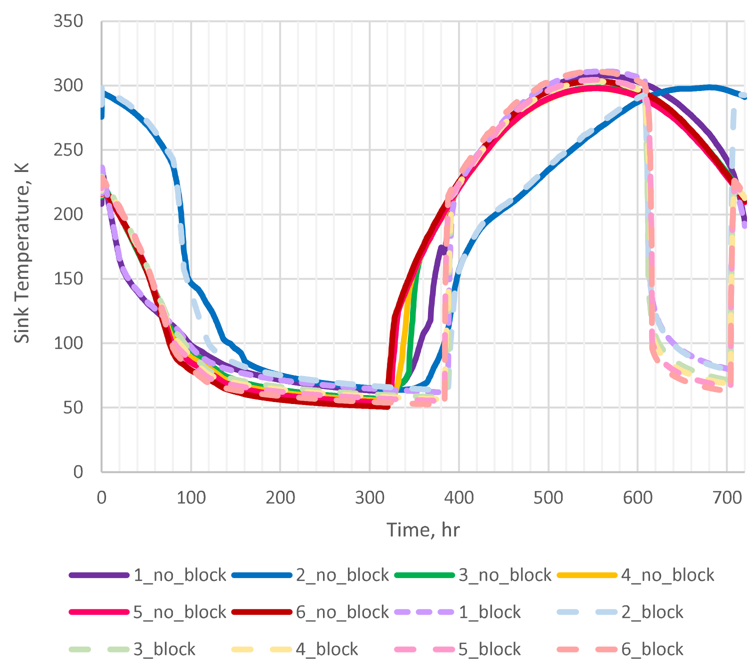
Ground temperatures



- Far-field terrain blocked solar rays from hitting cylinder



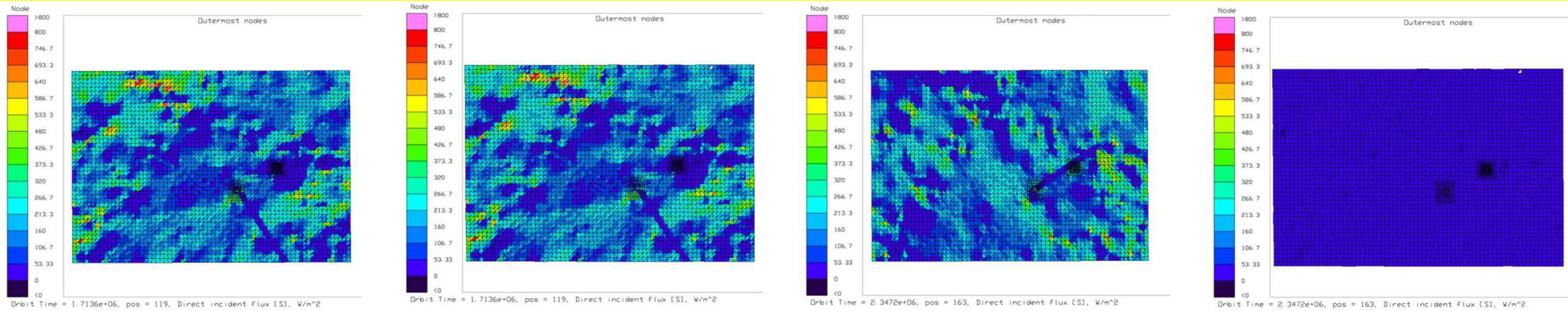
Direct Incident Solar flux



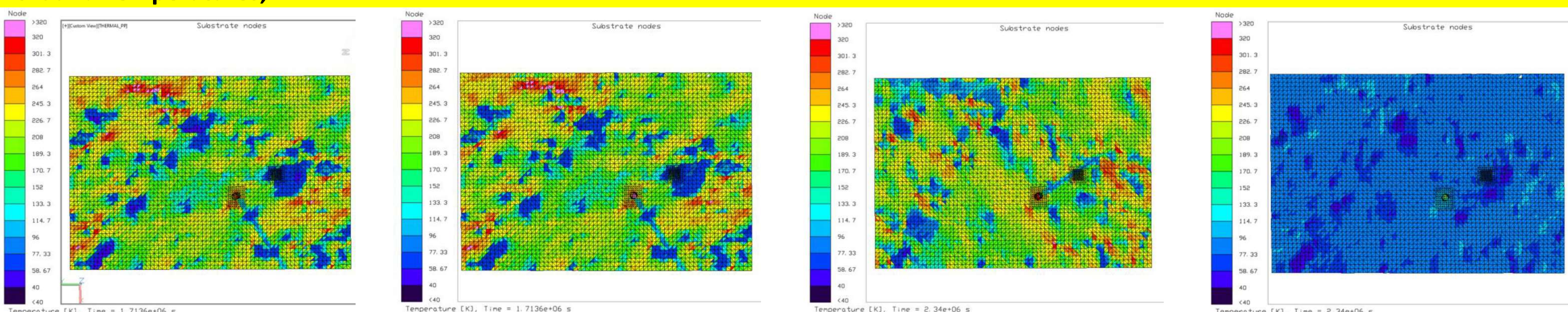
Sink Temperatures

Example: Effect of Far-field Terrain

Direct Incident Solar Flux, W/(m²K)



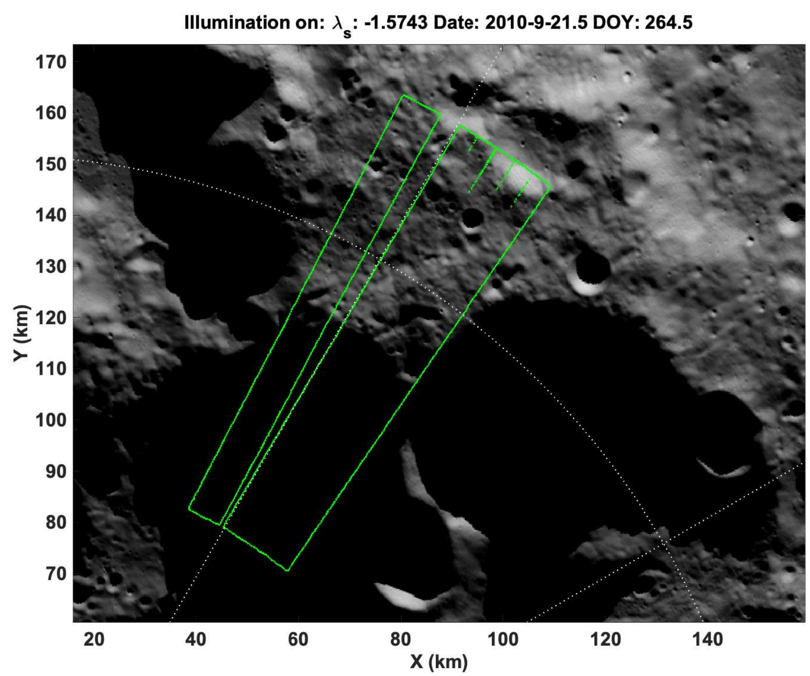
Ground Temperatures, K



[481 hrs] without far-field terrain | **<u>with</u> far-field terrain** | **[650-652 hrs] without far-field terrain** | **<u>with</u> far-field terrain**

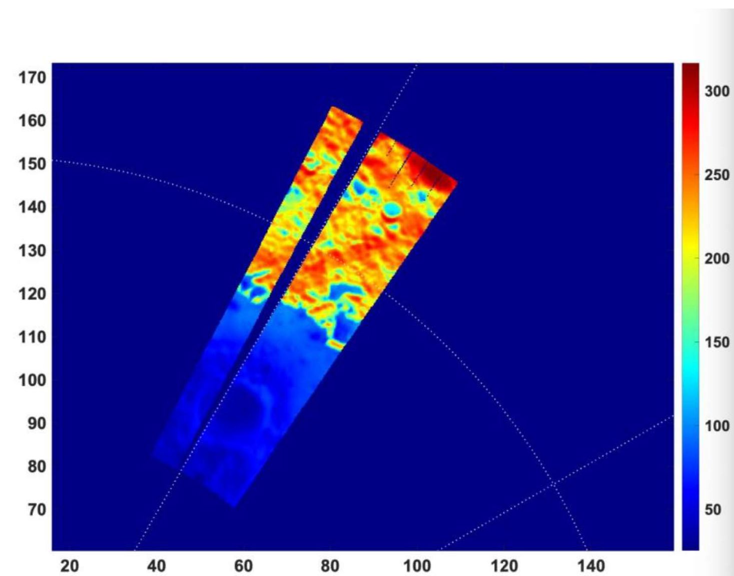
Comparison to Diviner Data

- Goal: Check Thermal Desktop ground temperature predictions with LRO DIVINER data [250m × 250m resolution]
- Comparison Date / Time: 2010-09-21 12.76 hours
- Nobile location:



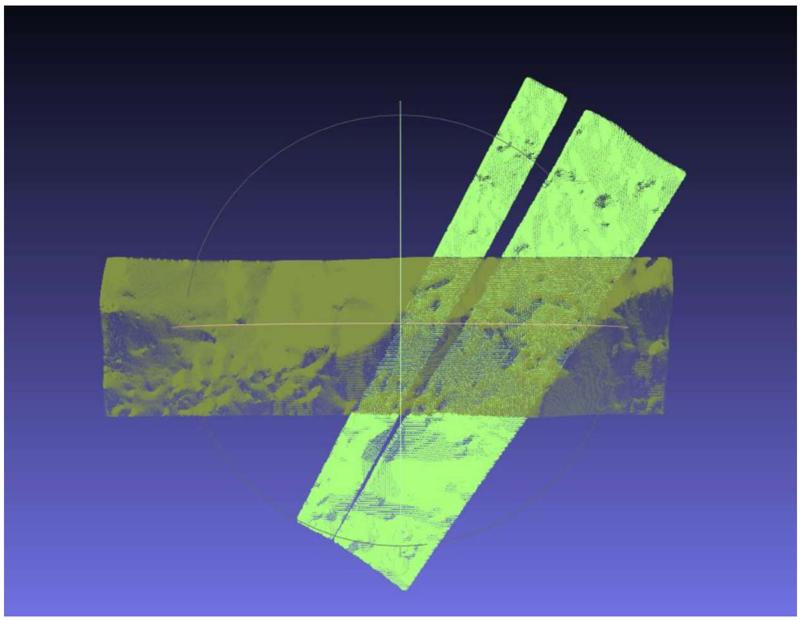
Comparison to Diviner Data

- Comparison:
  1) Received DIVINER Tbols from Pierre Williams for Comparison Date / Time: 2010-09-21 12.76 hours
     Tbol = Bolometric brightness temperature (measures spectrally integrated radiative flux from surface)
     Paige et al. “Diviner Lunar Radiometer Observations of Cold Traps in the Moon’s South Polar Region” & Gläser and Gläser “Modeling near-surface temperatures of airless bodies with application to the Moon” showed polar region good agreement of Tbols to model surface temperatures.
  2) Used DEM to determine x, y, z of DIVINER Tbols
  3) Used TD to interpolate node temperature to Tbol points

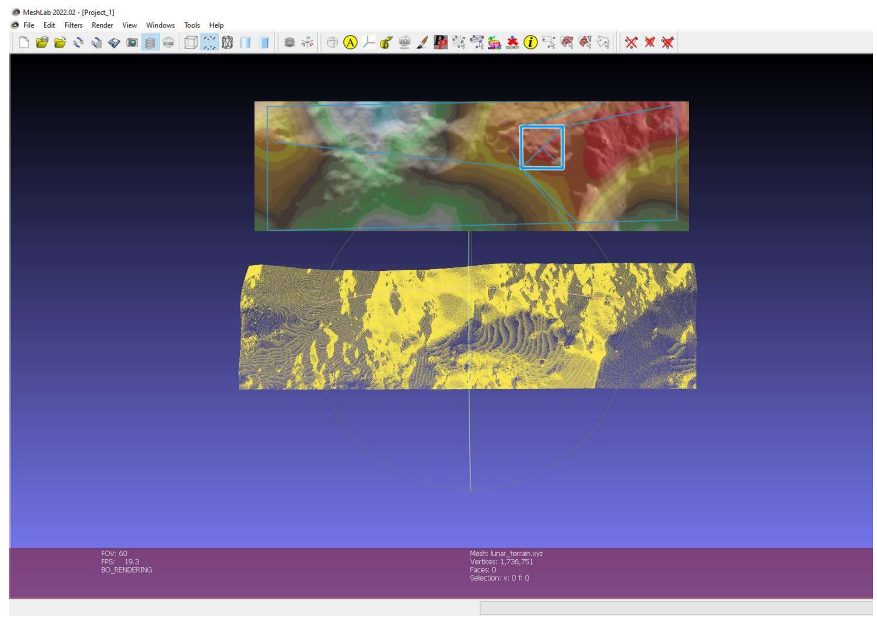


Lunar Surface Temperature on X – Y map (km)

Comparison to Diviner Data



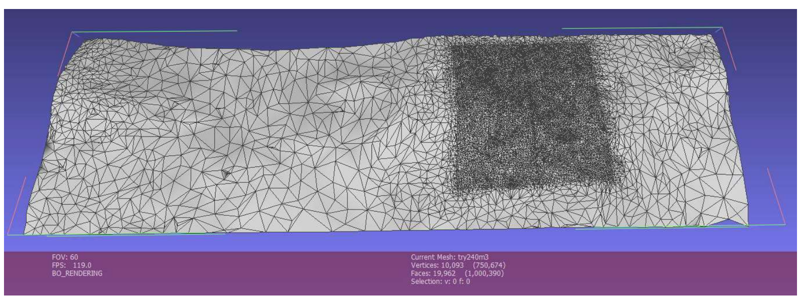
DEM points (tan) & Diviner Points (green)



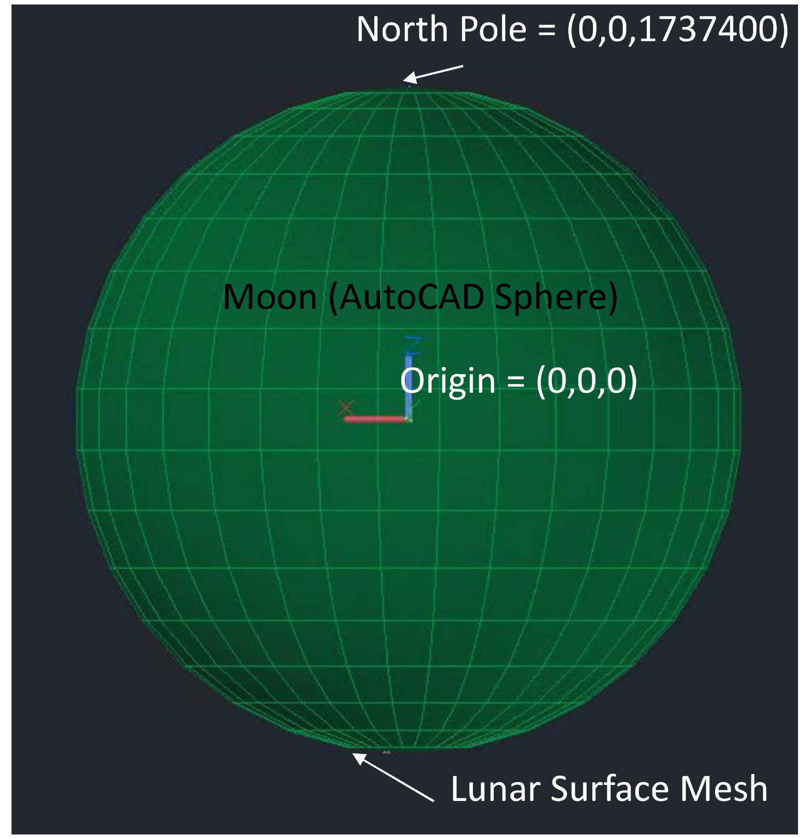
Point cloud (yellow) & Elevation from Lunar QuickMap Website

Comparison to Diviner Data

- Analysis Method:
  1) Create point cloud from 40m × 40m DEM
  2) Mesh in MeshLab use "Screened Poisson Surface" method
  3) Reduce mesh & import it into Thermal Desktop



Dense region element sizes are ~250m



Comparison to Diviner Data

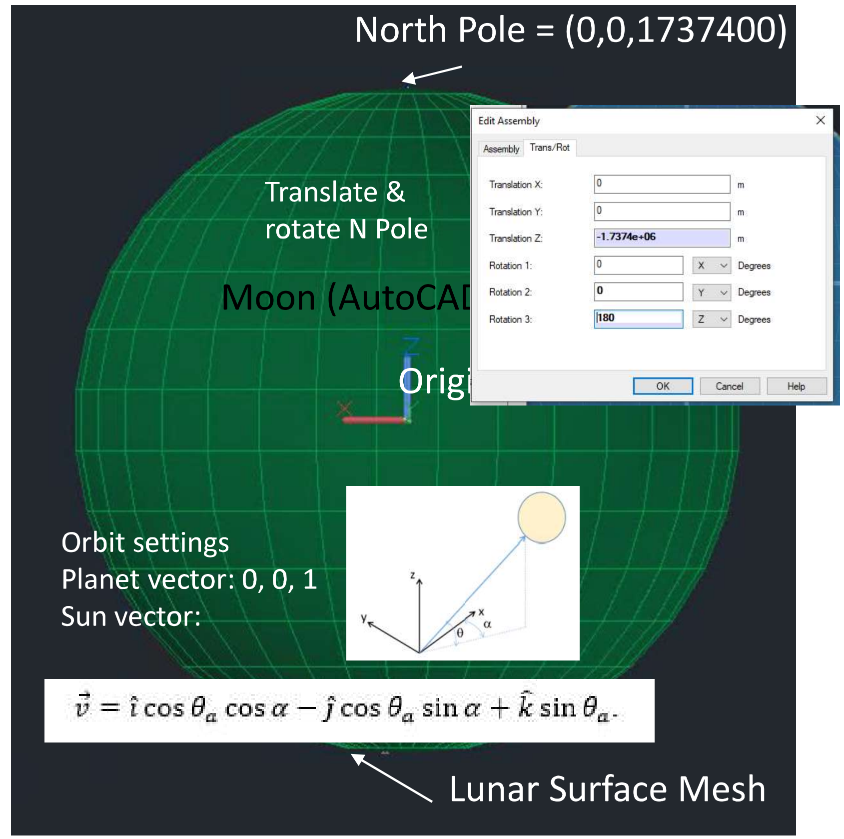
- Analysis Method:
  4) Obtain sun azimuth and elevation angles from JPL horizons database for comparison date to 5 yrs prior
  (angles are with respect to local zenith)
  5) Match origin in model with Horizon's assumed origin

JPL Horizons settings:

Current Settings

Ephemeris Type [change] : OBSERVER
Target Body [change] : **Sun [Sol]** [10]
Observer Location [change] : Topocentric @301 [Moon] (**0°00'00.0"E, 90°00'00.0"N**)
Time Span [change] : Start=**2010-09-21**, Stop=**2010-09-22**, Step=**30 m**
Table Settings [change] : QUANTITIES=**4,13**
Display/Output [change] : *default* (formatted HTML)

```
 Date__(UT)__HR:MN     Azi_(a-appr)_Elev Ang-diam
**************************************************
$$SOE
 2010-Sep-21 00:00     152.6407  -1.5625 1906.840
 2010-Sep-21 00:30     152.8943  -1.5626 1906.842
 2010-Sep-21 01:00     153.1479  -1.5628 1906.844
 2010-Sep-21 01:30     153.4015  -1.5630 1906.847
 2010-Sep-21 02:00     153.6551  -1.5632 1906.849
 2010-Sep-21 02:30     153.9088  -1.5634 1906.852
 2010-Sep-21 03:00     154.1624  -1.5635 1906.854
 2010-Sep-21 03:30     154.4160  -1.5637 1906.857
 2010-Sep-21 04:00     154.6696  -1.5639 1906.860
 2010-Sep-21 04:30     154.9232  -1.5641 1906.862
 2010-Sep-21 05:00     155.1768  -1.5642 1906.865
 2010-Sep-21 05:30     155.4304  -1.5644 1906.868
 2010-Sep-21 06:00     155.6840  -1.5646 1906.871
 2010-Sep-21 06:30     155.9376  -1.5647 1906.874
 2010-Sep-21 07:00     156.1912  -1.5649 1906.878
 2010-Sep-21 07:30     156.4448  -1.5651 1906.881
 2010-Sep-21 08:00     156.6984  -1.5652 1906.884
 2010-Sep-21 08:30     156.9520  -1.5654 1906.888
 2010-Sep-21 09:00     157.2056  -1.5655 1906.891
 2010-Sep-21 09:30     157.4592  -1.5657 1906.895
 2010-Sep-21 10:00     157.7128  -1.5659 1906.898
 2010-Sep-21 10:30     157.9664  -1.5660 1906.902
 2010-Sep-21 11:00     158.2200  -1.5662 1906.906
 2010-Sep-21 11:30     158.4736  -1.5663 1906.910
 2010-Sep-21 12:00     158.7272  -1.5665 1906.914
 2010-Sep-21 12:30     158.9808  -1.5666 1906.918
 2010-Sep-21 13:00     159.2344  -1.5668 1906.922
```



Comparison to Diviner Data

- Analysis Method:
  6) Supply a time varying solar constant
  - Used JPL WebGeocalc (Spice) to determine distance from sun at specified dates
    - output: Orbiting body range (km)
  - Calculated solar constant at these distances

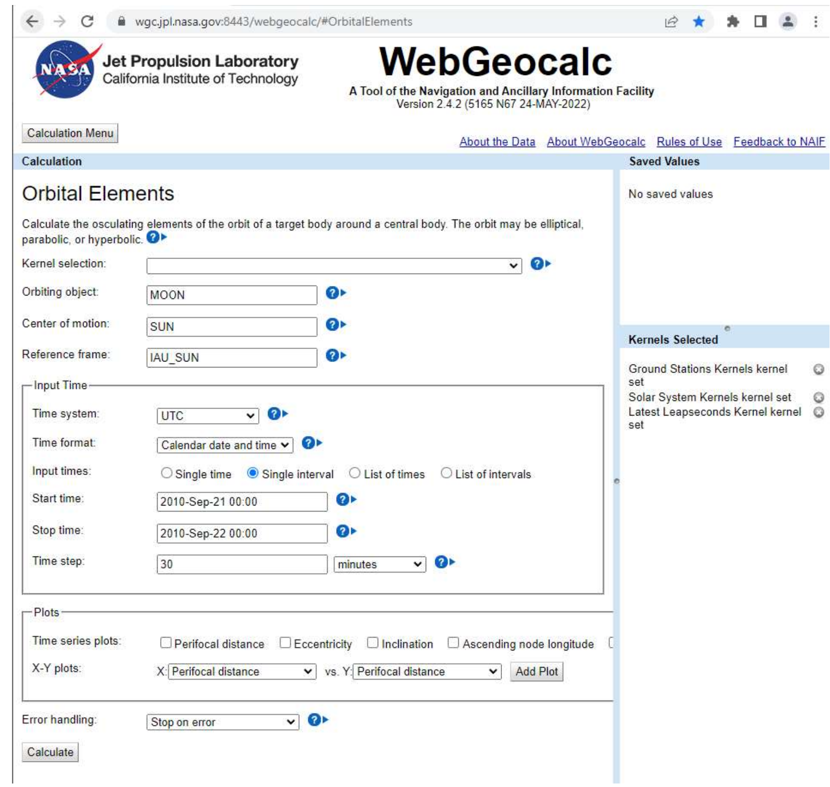
Solar flux is proportional to inverse square of distance to sun

$$q_s = I_b \frac{A_p}{S^2} \rightarrow q_s \propto \frac{1}{S^2}$$

A_p = projected are of sun
I_b = black body intensity
S = distance to sun

Can use $q_s = 1367\ W/m^2$ at 1 AU (149597870.7 km) to find q_s at other distances

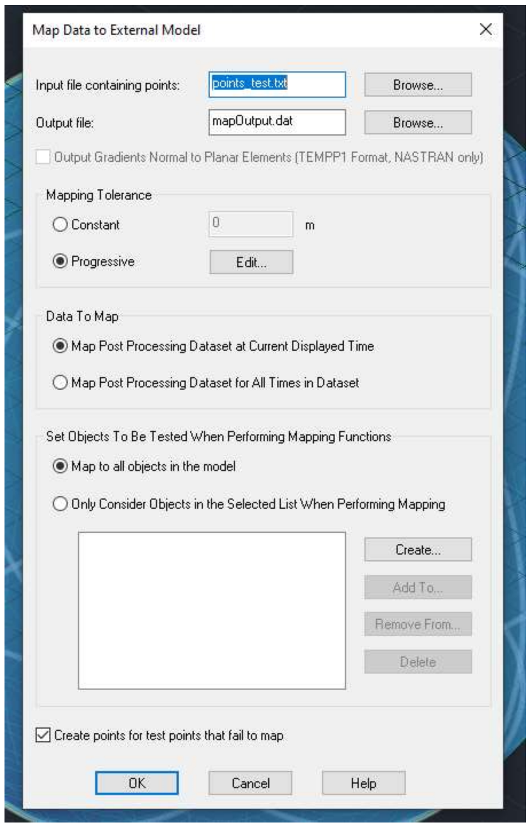
$$q_{s,2} = \frac{q_{s,1}\, S_1^2}{S_2^2}$$



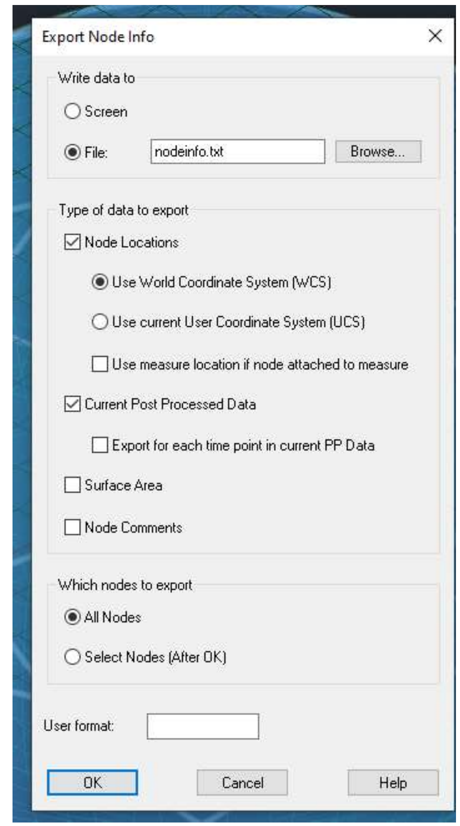
https://wgc.jpl.nasa.gov:8443/webgeocalc/#OrbitalElements

Comparison to Diviner Data

To use TD to interpolate node temperature to Tbol points: Thermal > Export > Map Data to Locations



To output temperatures at nodes in TD mesh: Thermal > Export > Write Note Information



Comparison to Diviner Data

Model 1:

- Solar Flux 1421 W/m^2
- Lunar Surface

  9 layers, 2m deep, increasing density

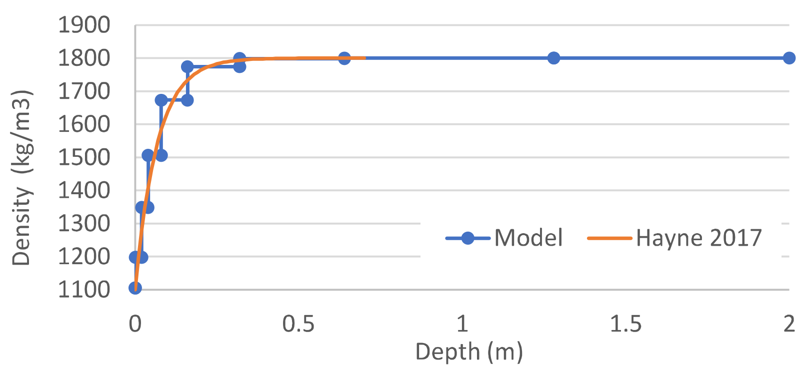
  α = angular dependent, Feng 2019[1]

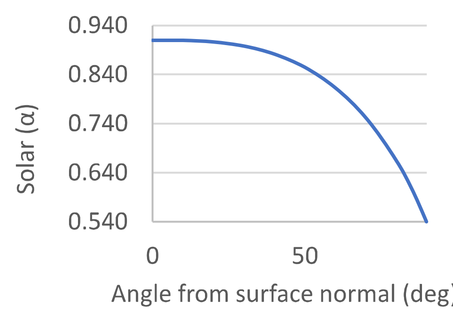
  ε = 0.97

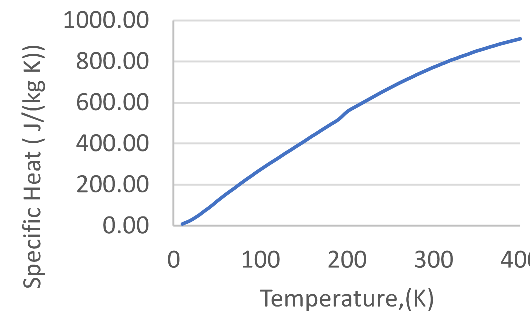
  Interior Heat Flux= 0.018 w/m^2

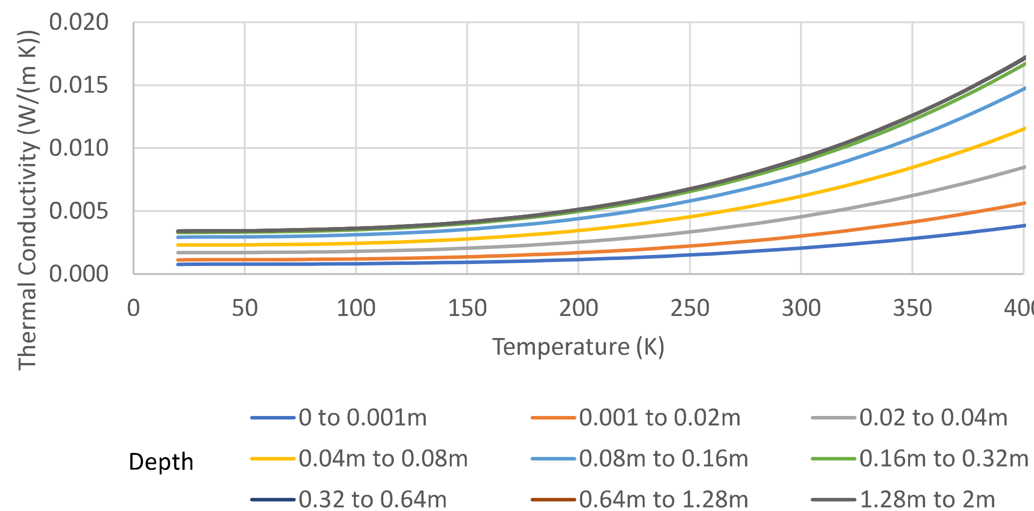
  k & ρ = Hayne 2017 (better when >100K)[2]

  c_p = Woods-Robinson 2019[3]







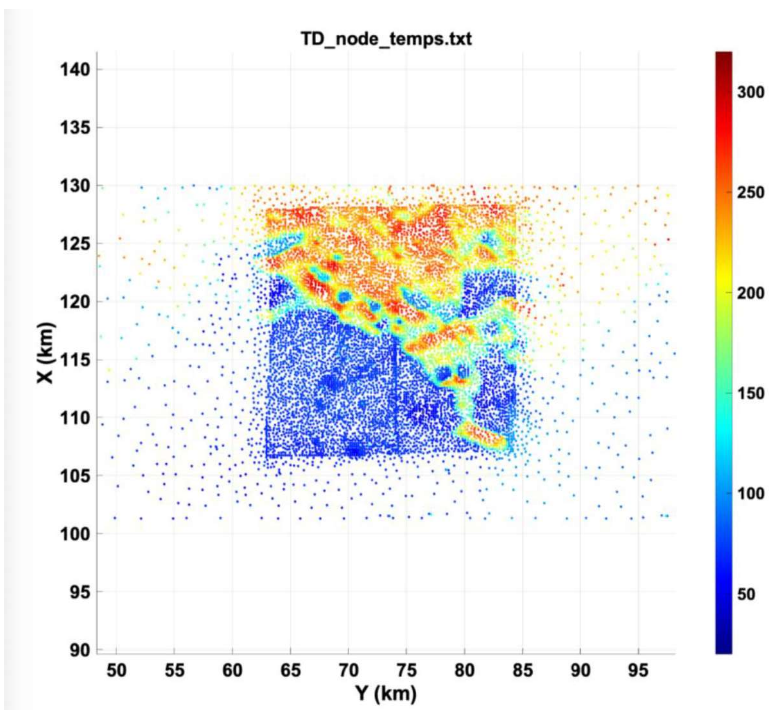


[1] Feng et al., “Lunar Regolith Properties Constrained by LRO Diviner and Chang’e-2 Microwave Radiometer Data,” 2019.
[2] Hayne et al., “Global Regolith Thermophysical Properties of the Moon From the Diviner Lunar Radiometer Experiment: Lunar Regolith Thermophysical Properties,” 2017.
[3]Woods-Robinson et al., " A model for the Thermophysical Properties of Lunar Regolith at Low Temperatures" 2019.

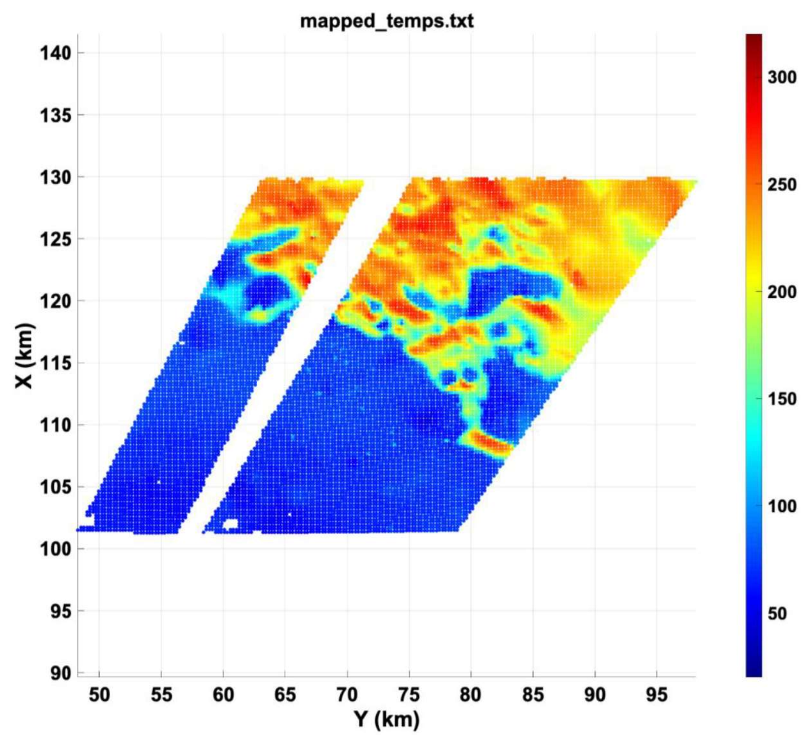
Comparison to Diviner Data

Comparison:

3) Used TD to interpolate node temperature to Diviner temperatures points (shown below)

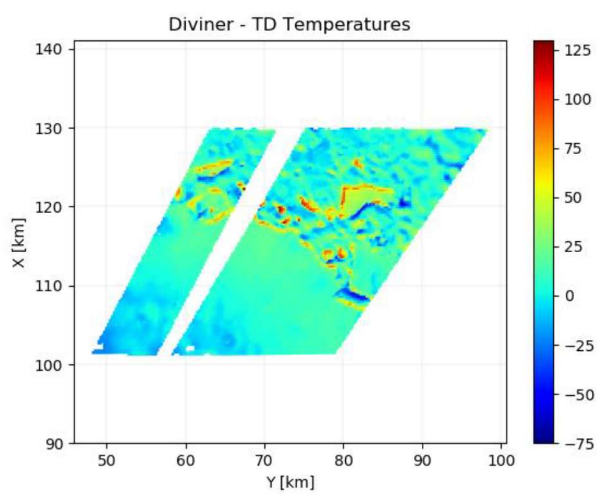


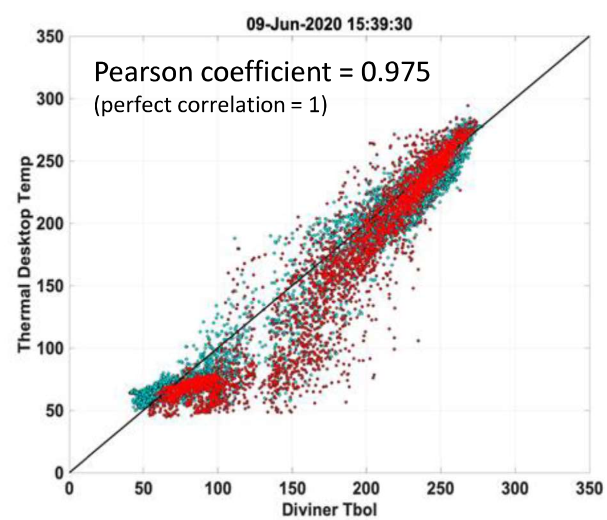
Node Temperatures (K)



Comparison to Diviner Data

- Results on par with comparison in P. Gläser and D. Gläser, "Modeling near-surface temperatures of airless bodies with application to the Moon" 2019.
- Hypothesize that scatter around 120-150K is largely due to mismatch in shadow boundaries from DEM mesh resolution
  - Gläser and Gläser found DEM resolution impacted how well the model's shadow boundaries matched with Diviner temperatures.





Red: High Resolution area of mesh
Green: Low Resolution area of mesh

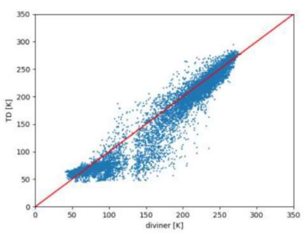
Comparison to Diviner Data – Property Sensitivity

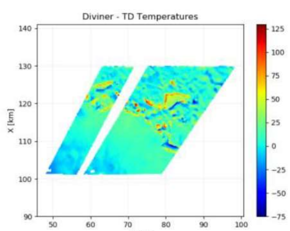
- Repeated comparison with different ground properties

Model 1:

α = angular dependent, Feng 2019
k & ρ = Hayne 2017
c_p = Woods-Robinson 2019

Pearson coefficient = 0.975

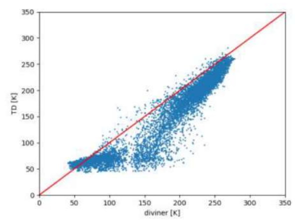


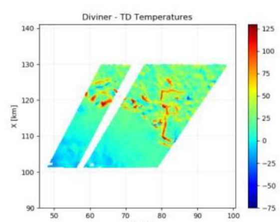


Model 2:

α = angular dependent, Feng 2020
k & ρ = Hayne 2017
c_p = Woods-Robinson 2019

Pearson coefficient = 0.963



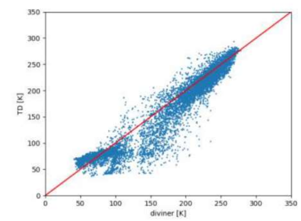


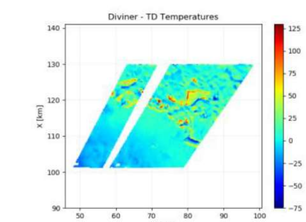
Model 3:

α = angular dependent, Feng 2020
k & ρ = Martinez and Siegler 2021
- thickness of surface needed to increase from 0.001 m to 0.01 m for time step reasons

c_p = Woods-Robinson 2019

Pearson coefficient = 0.971





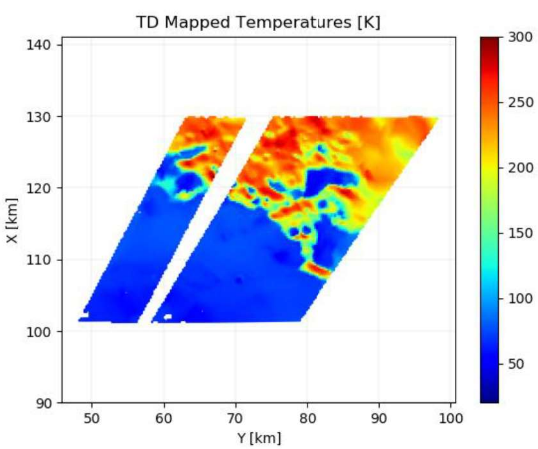
[1] Feng et al., “Lunar Regolith Properties Constrained by LRO Diviner and Chang’e-2 Microwave Radiometer Data,” 2019.
[2] Hayne et al., “Global Regolith Thermophysical Properties of the Moon From the Diviner Lunar Radiometer Experiment: Lunar Regolith Thermophysical Properties,” 2017.
[3] Woods-Robinson et al., " A model for the Thermophysical Properties of Lunar Regolith at Low Temperatures“, 2019.
[4] Martinez and Siegler “A global Thermal Conductivity Model for Regolith at Low Temperatures,” 2021.
[5] Feng, M. A. Siegler, and P. O. Hayne, “New Constraints on Thermal and Dielectric Properties of Lunar Regolith from LRO Diviner and CE-2 Microwave Radiometer,” 2020 – use full (not approximate) equation 8

Comparison to Diviner Data – Property Sensitivity

- Repeated comparison with different ground properties

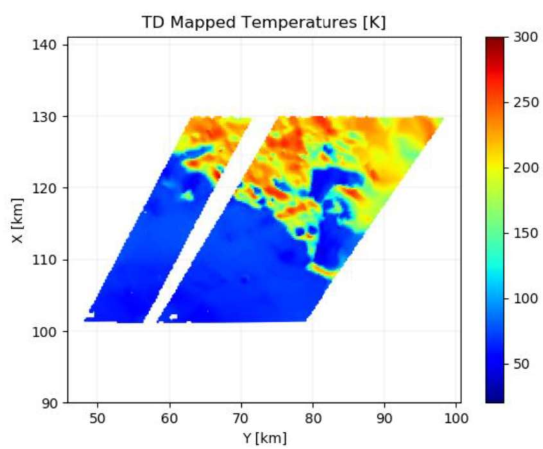
Model 1:

α = angular dependent, Feng 2019
k & ρ = Hayne 2017
c_p = Woods-Robinson 2019



Model 2:

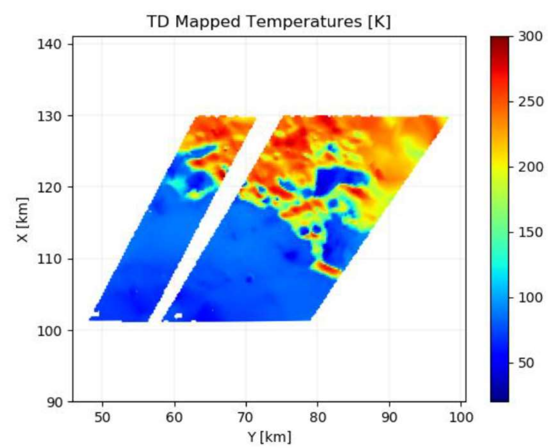
α = angular dependent, Feng 2020
k & ρ = Hayne 2017
c_p = Woods-Robinson 2019



Model 3:

α = angular dependent, Feng 2020
k & ρ = Martinez and Siegler 2021
- thickness of surface needed to increase from 0.001 m to 0.01 m for time step reasons

c_p = Woods-Robinson 2019



[1] Feng et al., "Lunar Regolith Properties Constrained by LRO Diviner and Chang'e-2 Microwave Radiometer Data," 2019.
[2] Hayne et al., "Global Regolith Thermophysical Properties of the Moon From the Diviner Lunar Radiometer Experiment: Lunar Regolith Thermophysical Properties," 2017.
[3] Woods-Robinson et al., " A model for the Thermophysical Properties of Lunar Regolith at Low Temperatures", 2019.
[4] Martinez and Siegler "A global Thermal Conductivity Model for Regolith at Low Temperatures," 2021.
[5] Feng, M. A. Siegler, and P. O. Hayne, "New Constraints on Thermal and Dielectric Properties of Lunar Regolith from LRO Diviner and CE-2 Microwave Radiometer," 2020 – use full (not approximate) equation 8

Comparison to Diviner Data – Property Sensitivity

- Repeated comparison with different ground properties

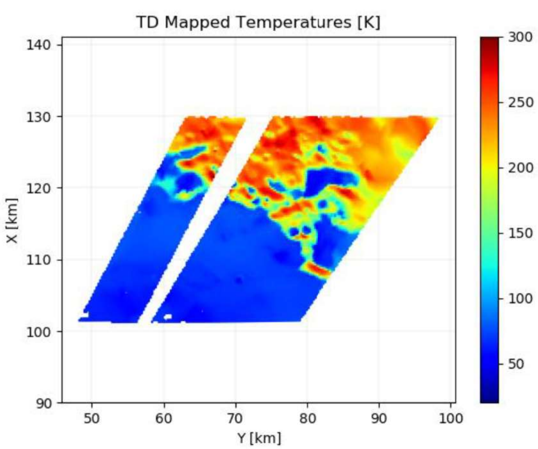
Model 1:

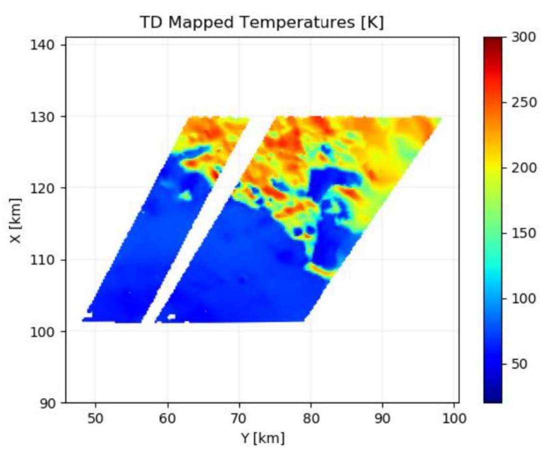
α = angular dependent, Feng 2019
k & ρ = Hayne 2017
c_p = Woods-Robinson 2019

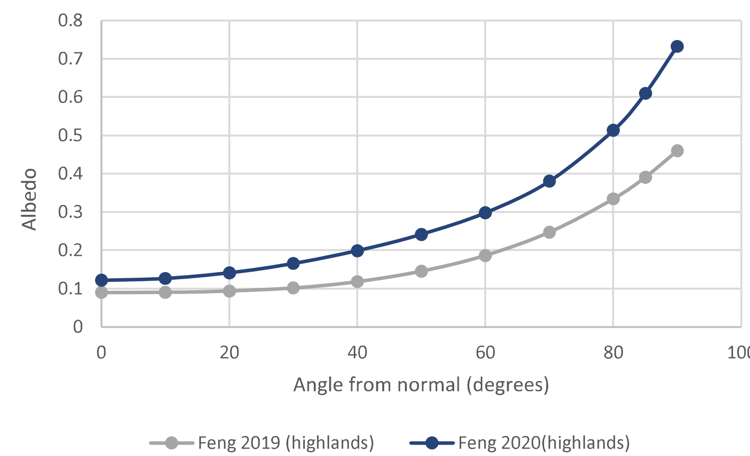
Model 2:

α = angular dependent, Feng 2020
k & ρ = Hayne 2017
c_p = Woods-Robinson 2019

The differences in albedo caused a noticeable change in predicted surface temperatures.





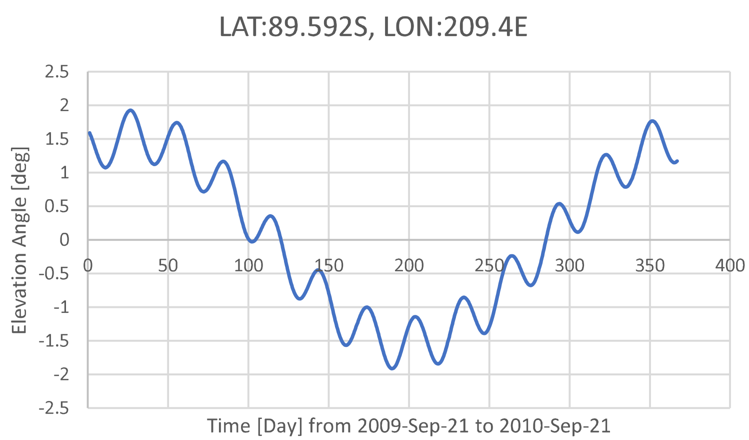


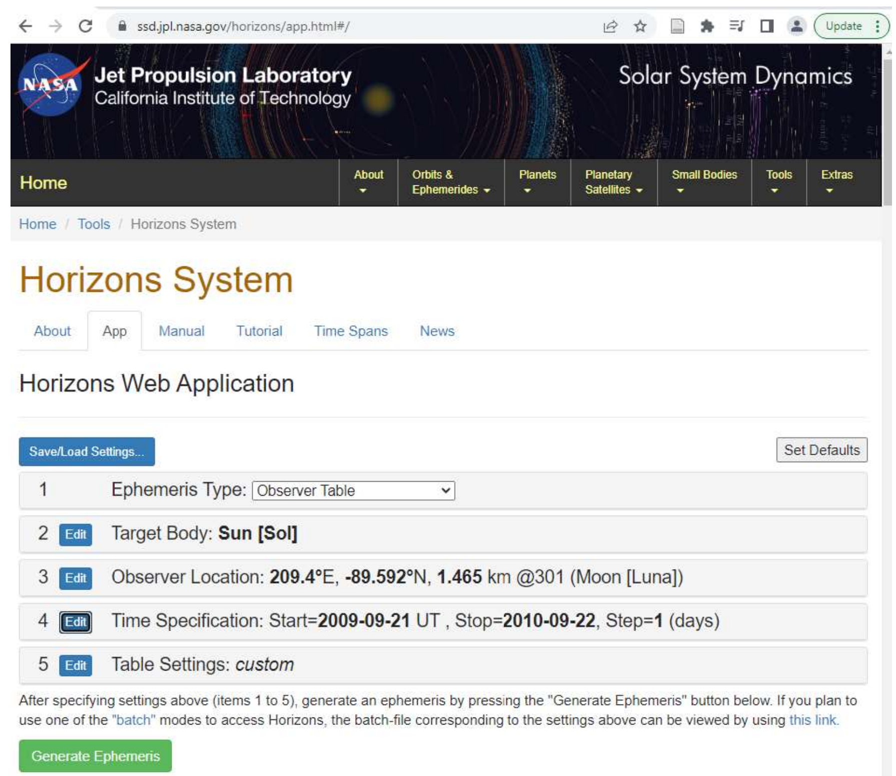
[1] Feng et al., "Lunar Regolith Properties Constrained by LRO Diviner and Chang'e-2 Microwave Radiometer Data," 2019.
[2] Hayne et al., "Global Regolith Thermophysical Properties of the Moon From the Diviner Lunar Radiometer Experiment: Lunar Regolith Thermophysical Properties," 2017.
[3] Woods-Robinson et al., " A model for the Thermophysical Properties of Lunar Regolith at Low Temperatures", 2019.
[4] Martinez and Siegler "A global Thermal Conductivity Model for Regolith at Low Temperatures," 2021.
[5] Feng, M. A. Siegler, and P. O. Hayne, "New Constraints on Thermal and Dielectric Properties of Lunar Regolith from LRO Diviner and CE-2 Microwave Radiometer," 2020 – use full (not approximate) equation 8

Backup

3D Modeling example – Size the Terrain

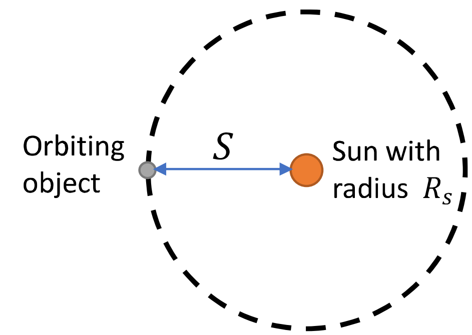
- Obtain local elevation angles
  - Output from JPL Horizons
  - Max: 1.925107°





https://ssd.jpl.nasa.gov/horizons/app.html#/

Heat flux on body orbiting the sun



Solid angle $\Omega_s = \frac{A_p}{S^2} = \frac{\pi R_S^2}{S^2}$

Total black body emissive power in vacuum

$$E_b = \sigma T^4$$

Total emissive power related to intensity to intensity

$$E_b = \int_\Omega I_b \hat{n} \cdot \hat{s} \mathrm{d}\Omega = \int_0^{2\pi} \int_0^{\pi/2} I_b \cos\theta \sin\theta \, d\theta d\psi = \pi I_b \rightarrow I_b = \frac{E_b}{\pi}$$

Solar constant

$$q_s = \int_\Omega I_b \mathrm{d}\Omega = I_b \int_\Omega \mathrm{d}\Omega = I_b \Omega_s = \left(\frac{E_b}{\pi}\right) \Omega_s = \frac{E_b}{\pi} \frac{\pi R_S^2}{S^2}$$

$q_s = I_b \frac{A_p}{S^2} \rightarrow q_s \propto \frac{1}{S^2}$ solar flux is proportional to inverse square of distance to sun

Therefore, if q_s is known at one location we can find q_s at others

$$q_{s,1} S_1^2 = q_{s,2} S_2^2$$

$$q_{s,2} = \frac{q_{s,1} S_1^2}{S_2^2}$$

Reference: Modest, "Chapter 1: Fundamentals of Thermal Radiation" *Radiative Heat Transfer,* 2013