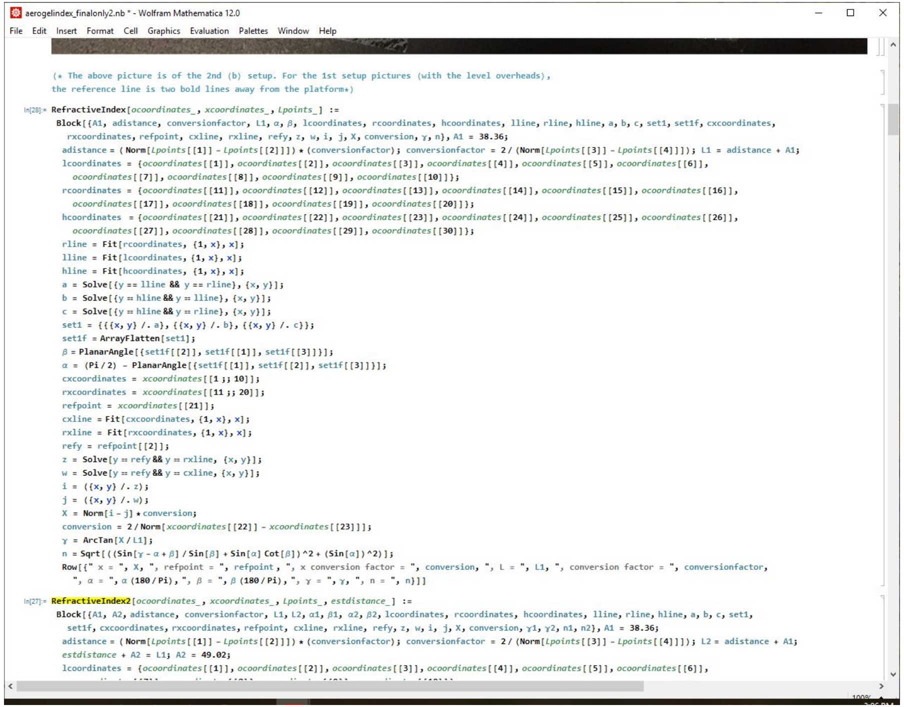
Aerogel Refractive Index

Update 8/19/19

Mathematica Program for measuring the refractive index

- This past week I focused on getting the Mathematica for the aerogel tile pictures up and running
- To the right is the code for the finished first version of the image analysis program




```
(* The above picture is of the 2nd (b) setup. For the 1st setup pictures (with the level overheads),
the reference line is two bold lines away from the platform*)

In[28]:= RefractiveIndex[ocoordinates_, xcoordinates_, Lpoints_] :=
 Block[{A1, adistance, conversionfactor, L1, α, β, lcoordinates, rcoordinates, hcoordinates, lline, rline, hline, a, b, c, set1, set1f, cxcoordinates,
   rxcoordinates, refpoint, cxline, rxline, refy, z, w, i, j, X, conversion, γ, n}, A1 = 38.36;
  adistance = (Norm[Lpoints[[1]] - Lpoints[[2]]]) * (conversionfactor); conversionfactor = 2 / (Norm[Lpoints[[3]] - Lpoints[[4]]]); L1 = adistance + A1;
  lcoordinates = {ocoordinates[[1]], ocoordinates[[2]], ocoordinates[[3]], ocoordinates[[4]], ocoordinates[[5]], ocoordinates[[6]],
    ocoordinates[[7]], ocoordinates[[8]], ocoordinates[[9]], ocoordinates[[10]]};
  rcoordinates = {ocoordinates[[11]], ocoordinates[[12]], ocoordinates[[13]], ocoordinates[[14]], ocoordinates[[15]], ocoordinates[[16]],
    ocoordinates[[17]], ocoordinates[[18]], ocoordinates[[19]], ocoordinates[[20]]};
  hcoordinates = {ocoordinates[[21]], ocoordinates[[22]], ocoordinates[[23]], ocoordinates[[24]], ocoordinates[[25]], ocoordinates[[26]],
    ocoordinates[[27]], ocoordinates[[28]], ocoordinates[[29]], ocoordinates[[30]]};
  rline = Fit[rcoordinates, {1, x}, x];
  lline = Fit[lcoordinates, {1, x}, x];
  hline = Fit[hcoordinates, {1, x}, x];
  a = Solve[{y == lline && y == rline}, {x, y}];
  b = Solve[{y == hline && y == lline}, {x, y}];
  c = Solve[{y == hline && y == rline}, {x, y}];
  set1 = {{{x, y} /. a}, {{x, y} /. b}, {{x, y} /. c}};
  set1f = ArrayFlatten[set1];
  β = PlanarAngle[{set1f[[2]], set1f[[1]], set1f[[3]]}];
  α = (Pi / 2) - PlanarAngle[{set1f[[1]], set1f[[2]], set1f[[3]]}];
  cxcoordinates = xcoordinates[[1 ;; 10]];
  rxcoordinates = xcoordinates[[11 ;; 20]];
  refpoint = xcoordinates[[21]];
  cxline = Fit[cxcoordinates, {1, x}, x];
  rxline = Fit[rxcoordinates, {1, x}, x];
  refy = refpoint[[2]];
  z = Solve[y == refy && y == rxline, {x, y}];
  w = Solve[y == refy && y == cxline, {x, y}];
  i = ({x, y} /. z);
  j = ({x, y} /. w);
  X = Norm[i - j] * conversion;
  conversion = 2 / Norm[xcoordinates[[22]] - xcoordinates[[23]]];
  γ = ArcTan[X / L1];
  n = Sqrt[((Sin[γ - α + β] / Sin[β] + Sin[α] Cot[β])^2 + (Sin[α])^2)];
  Row[{" x = ", X, ", refpoint = ", refpoint, ", x conversion factor = ", conversion, ", L = ", L1, ", conversion factor = ", conversionfactor,
    ", α = ", α (180 / Pi), ", β = ", β (180 / Pi), ", γ = ", γ, ", n = ", n}]]

In[27]:= RefractiveIndex2[ocoordinates_, xcoordinates_, Lpoints_, estdistance_] :=
 Block[{A1, A2, adistance, conversionfactor, L1, L2, α1, β1, α2, β2, lcoordinates, rcoordinates, hcoordinates, lline, rline, hline, a, b, c, set1,
   set1f, cxcoordinates, rxcoordinates, refpoint, cxline, rxline, refy, z, w, i, j, X, conversion, γ1, γ2, n1, n2}, A1 = 38.36;
  adistance = (Norm[Lpoints[[1]] - Lpoints[[2]]]) * (conversionfactor); conversionfactor = 2 / (Norm[Lpoints[[3]] - Lpoints[[4]]]); L2 = adistance + A1;
  estdistance + A2 = L1; A2 = 49.02;
  lcoordinates = {ocoordinates[[1]], ocoordinates[[2]], ocoordinates[[3]], ocoordinates[[4]], ocoordinates[[5]], ocoordinates[[6]],
```

Mathematica Program

- There are 2 versions of the program, with 4 different parts for measuring: alpha and beta, x, L (using picture set A), and L (using picture set B)
- The first program is for use with picture set A of the 3rd tile. The overhead shots in this set were confirmed to be level, so they are more reliable for the calculations
- The 2nd program is for use with set B, which has pictures of x going deeper into the tile
    - The overhead shots in set B are not good enough to get accurate measurements from, so I added in a second set of calculations using α = 45 deg, β = 90 deg, and a estimated distance as an input for L.

Program instructions

Below are instructions for taking the coordinate sets and finding the values to use as inpust for the program. The first 3 are coordinate sets taken manually by the user from pictures, and the *estdistance* is a value estimated from the graph paper in the pictures.

Instructions:

For *ocoordinates* (overhead coordinates): using the coordinates tool on the level overhead picture, take ten coordinates outlining the edge of the aerogel tile that the incident laser beam strikes. Then take ten coordinates outlining the other edge, and finally ten coordinates outlining one of the edges of the part of the incident laser beam that is outside the tile.

For *xcoordinates*: using the coordinates tool on the target picture, take ten coordinates outlining the left edge of the non refracted part of the laser that hits the target, then ten coordinates along the left side of the refracted beam. Pick a point to use as a reference y value, this should approx. divide the segment of the refracted beam in half. Take 2 points giving the horizontal distance across one of the larger black squares on the grid (containing 5 small squares).

For *Lpoints*: using the coordinates tool on either the level overhead or reference overhead pictures, take two points giving the distance from the tip of the aerogel to the reference line (which is either the 1st or 2nd darker line off of the platform depending on whether you're looking at set up 1 or 2)

For *estdistance*: count how many small squares the tip of the aerogel tile is from the center line on the platform (the 4th one from either edge) and multiply by .4. See stars in pics showing reference lines for the 2nd setup. If on the opposite side from the target, estdistance should be positive; if on the same side, negative.

Use the 1st program for the 1st setup and the 2nd for the second setup.

Testing the program

```
In[29]:= RefractiveIndex[ocoordinates1, xcoordinates1, Lpoints1]

Out[29]= x = 3.10951, refpoint = {273., 388.}, x conversion factor = 0.00840306
        , L = 51.4311, conversion factor = 0.00783076, α = 44.7717, β = 88.0564, γ = 0.0603862, n = 1.03076

In[30]:= RefractiveIndex[ocoordinates2, xcoordinates2, Lpoints2]

Out[30]= x = 3.20714, refpoint = {101., 377.}, x conversion factor = 0.00836813
        , L = 51.3017, conversion factor = 0.00782899, α = 43.7238, β = 89.9584, γ = 0.0624339, n = 1.03058

In[32]:= RefractiveIndex2[ocoordinates3, xcoordinates3, Lpoints3, estdistance3]

         Set: Tag Plus in 2.012 + A2 is Protected.

Out[32]= x = 3.17724, refpoint = {554., 205.}, x conversion factor = 0.00829868, L1 = L1, L2 = 57.2427, conversion factor = 0.0046672, α1 = 90 deg
```

$$, \beta 1 = 45\text{ deg}, \alpha 2 = 46.1376, \beta 2 = 89.5281, \gamma 1 = \mathrm{ArcTan}\left[\frac{3.17724}{L1}\right] 0.0554479, \gamma 2 = , n1 = \sqrt{\frac{1}{2} + \mathrm{Sin}\left[\frac{\pi}{4} + \mathrm{ArcTan}\left[\frac{3.17724}{L1}\right]\right]^2}, n2 = 1.02756$$

(Results so far: n = 1.031, n = 1.031, n = 1.030)

Remaining work on the program

- On Friday I Imported 2 all of the aerogel pictures I had for 2 of the angles of the 3rd tile into the Mathematica file and tried to obtain values for n. This resulted in the program crashing and my work for that day being lost
- What's left for this program:
  - Import the pictures in smaller batches and multiple saved files to obtain values for n
  - Estimate error when using this method
  - Look for a better way to input data and present results from this program