Nov 10th

Task: Calculate the approximate thickness of the coated layer on the mirror, with Chromium and Aluminum being evaporated from the source for 20 mins.

Total evaporated mass from the source:

$$\bar{M}_e = \int_0^t \int_{A_e} \Gamma_e dA_e \, dt$$

A_e is the surface area of the souce needed to be measured.

I assumed that the evaporation rate Γ at each point of the source is the same, and that it doesn't change w.r.t. time.

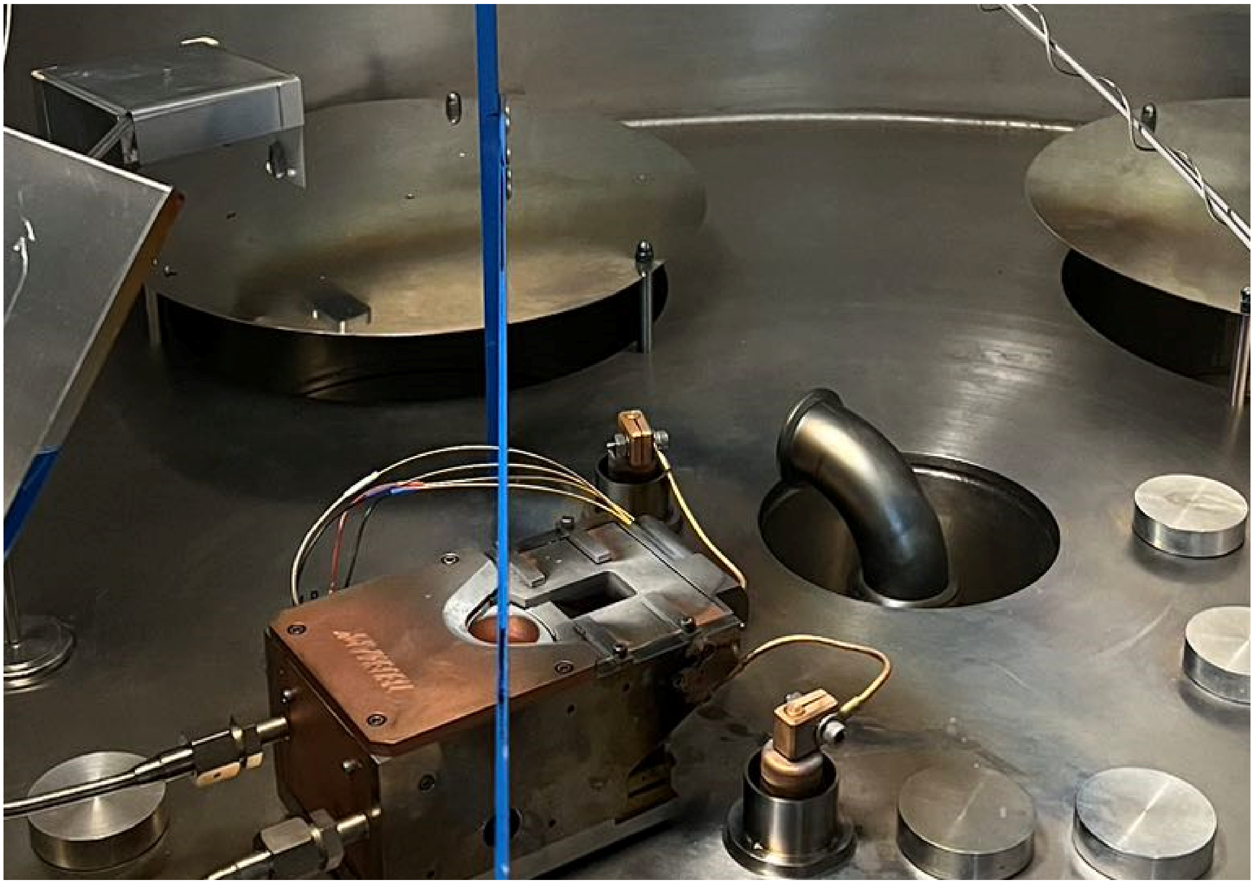
<u>Approximation of the evaporation rate:</u>

According to what Bill had told me, I made an estimate of the evaporation rate according to the document he sent me, although the angles in our cases are less than 45 degrees. Below is a screenshot of that paragraph about the estimation:

0.6. Provided the constant values, eq(1) is applied to find that, at an angle of $45°$ between the source and substrate from the source norm, the mass deposited per unit area is $\approx 0.0101859 \frac{g}{cm} \frac{W}{Hz}$, and where the evaporation rate is given as $\Gamma \approx 10^{-4} \frac{g}{cm}^2 sec^{-1}$ a comparable value is found for the deposition.

i.e. $\Gamma \approx 10^{-3} (\frac{kg}{m^2}) \frac{1}{s}$

<u>Measuring the surface area of the source:</u>



$r_{surface} \approx \frac{9}{16} \pm \frac{1}{32}\ inches$

$A_e = \pi r_{surface}^2 \pm 2\pi r_{surface} \triangle r_{surface}$

$\approx 0.9940 \pm 0.1104\ \ inches^2 \approx (6.413 \pm 0.713) \times 10^{-4}\ m^2$

If assuming the whole evaporation process lasts for 20 mins, then the total evaporated mass is:

$\bar{M_e} = \Gamma \cdot A_e \cdot t \approx (7.6956 \pm 0.8551) \times 10^{-7}\ kg$

Surface mass denstity on the mirrors:

$$\frac{d\bar{M_s}}{dA_s} = \frac{\bar{M_e} cos(\theta) cos^n(\phi)(n+1)}{2\pi r^2}$$

θ is the angle subtended by **the direction of the propagation** and **the normal vector of the substrate**; ϕ is the angle substanded by **the propagation direction and source normal**. In this case, Since the substrate is in parallel with the source, $\theta = \phi$.

r is the **radius of the evaporator envelope.**

According to what was written in the later part of this document,

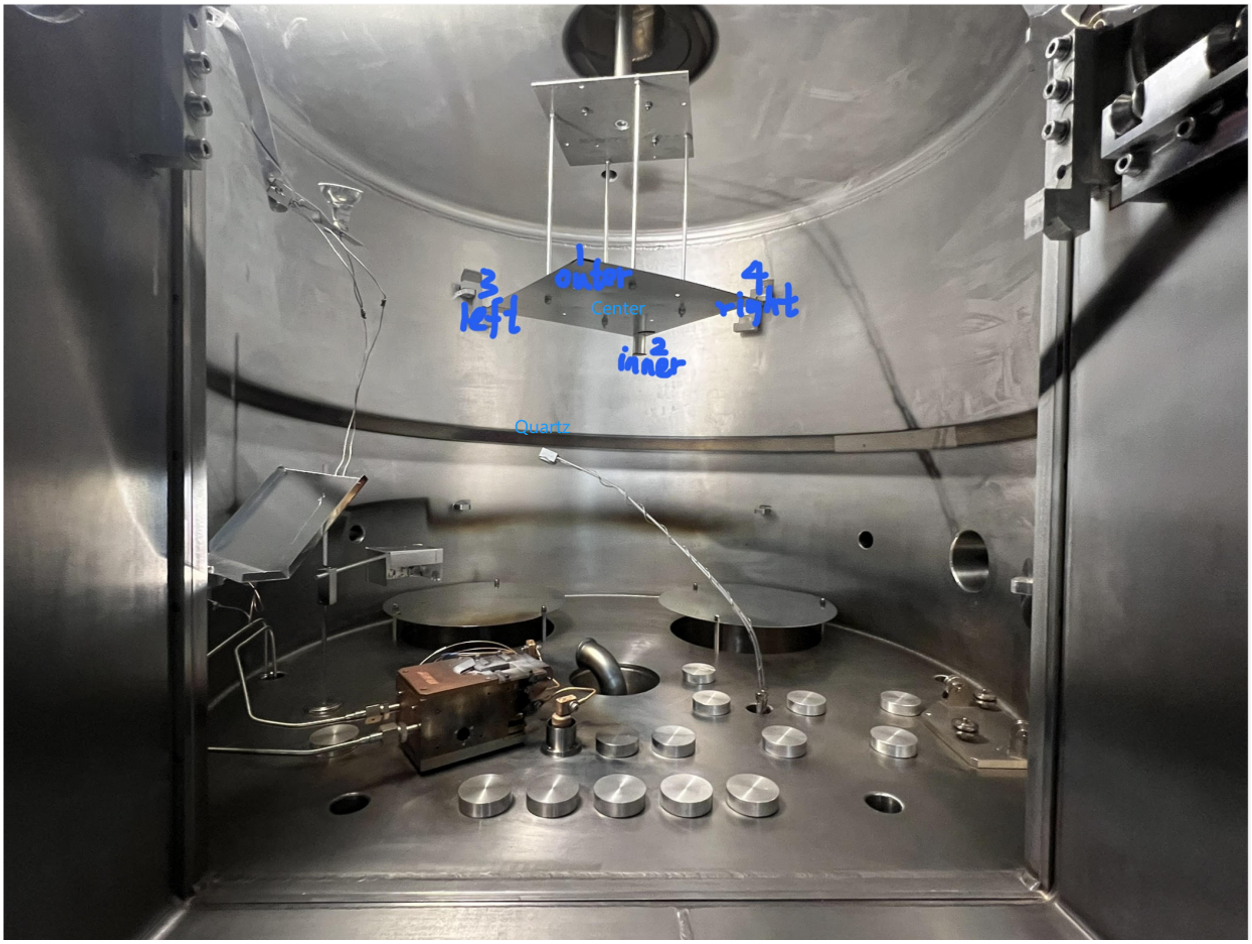
assume the spectral power n = 0.6.

This calculation **neglects the change in spectral power n, time elasped.**

Measuring the radius and the angle:

I'm going to measure such angle of 6 different places, the 4 mirrors, the center of the substrate, and the quartz.

There are 2 mirrors at the inner side, but the variation of θ angle of these 2 mirrors isn't a lot.



I measured the **radius of the evaporator envelope**, denoted as R,

as well as **it's projection**, denoted as d.

The uncertainty maybe big, since it's a crude measurement by eye and ruler. R and d may have uncertainty around $\frac{1}{8}$ inches(for those locations easier to measure the distances to) to $\frac{1}{2}$ inches (hopefully this is the biggest uncertainty that may caused for those locations hard to measure the distance to).

	R (inch)	upmost uncertainty of R (inch)	d (inch)	upmost uncertainty of d (inch)
Outer Corner (nearest)	$20+\frac{7}{8}$	$\frac{1}{4}$	$4+\frac{5}{8}$	$\frac{1}{5}$
Inner Corner (farest)	$26+\frac{1}{2}$	$\frac{1}{2}$	$18+\frac{11}{16}$	$\frac{1}{3}$
Left Corner	$22+\frac{3}{8}$	$\frac{1}{3}$	$11+\frac{5}{8}$	$\frac{1}{4}$
Right Corner	$25+\frac{1}{8}$	$\frac{1}{3}$	$14+\frac{1}{4}$	$\frac{1}{4}$
Center	$22+\frac{3}{8}$	$\frac{1}{3}$	$11+0$	$\frac{1}{4}$
Quartz	$13+0$	$\frac{1}{4}$	$7+\frac{5}{8}$	$\frac{1}{5}$

And I calculated the angle $\theta(=\phi)=arcsin\frac{d}{R}$.

To get the uncertainty of $\theta(=\phi)$, I need to use the formula for error propagation:

$$\triangle\theta=\sqrt{(\frac{\partial\theta}{\partial d}\triangle d)^2+(\frac{\partial\theta}{\partial R}\triangle R)^2}$$
$$=\sqrt{(\frac{1}{R\sqrt{1-\frac{d^2}{R^2}}}\triangle d)^2+(-\frac{d}{R^2\sqrt{1-\frac{d^2}{R^2}}}\triangle R)^2}$$

With all the information above, I can get the surface density $\frac{d\bar{M}_s}{dA_s}$.

$$\frac{d\bar{M}_s}{dA_s}=\frac{\bar{M}_e cos(\theta)cos^n(\phi)(n+1)}{2\pi r^2}$$

To get $\triangle\frac{d\bar{M}_s}{dA_s}$, we need to use the formula for error propogation:

$$\triangle\frac{d\bar{M}_s}{dA_s}=\sqrt{[\frac{\partial}{\partial M_e}(\frac{d\bar{M}_s}{dA_s})\triangle M_e]^2+[\frac{\partial}{\partial\theta}(\frac{d\bar{M}_s}{dA_s})\triangle\theta]^2+[\frac{\partial}{\partial R}(\frac{d\bar{M}_s}{dA_s})\triangle R]^2}$$

$$= \sqrt{[\frac{(n+1)}{2\pi}\frac{(cos\theta)^{n+1}}{R^2}\triangle M_e]^2 + [\frac{M_e(n+1)}{2\pi}\frac{-(n+1)(sin\theta)(cos\theta)^n}{R^2}\triangle\theta]^2 + [\frac{-M_e(n+1)(cos\theta)^{n+1}}{\pi R^3}\triangle R]^2}$$

Calculate the thickness of coated layer:

After getting the surface mass density, if I divide it by the density, I can get the thickness of the coated layer σ.

$$\sigma = \frac{(\frac{d\bar{M}_s}{dA_s})}{\rho}$$

where $\rho_{chromium} = 7140\ kg/m^3, \rho_{aluminum} = 2710\ kg/m^3$.

In order to get the uncertainty of the thickness, I got the uncertainty of the surface density first, and divide it by ρ. And I got it's uncertainty via:

$$\triangle\sigma = \frac{1}{\rho}\cdot\triangle\frac{d\bar{M}_s}{dA_s}$$

Code and Result:

Then I wrote a CERN ROOT code to calculate the thickness of coating with (hopefully) maximum uncertainty:

```cpp
double get_surface_density(double M_e, double theta, double R, double n)
{
        return (M_e * TMath::Cos(theta) * pow(TMath::Cos(theta), n) * (n+1))/ (2* TMath::Pi() * pow(R, 2));
}
//R is the radius of the evaporater envelop, i.e. the distance from the source to the mirror.
//Since the mirror is in parallel with the evaporator, theta = phi.

double get_thickness(double surface_density, double rho)
{
        return surface_density / rho;
}

double get_delta_A_e(double r, double delta_r)
{
        return 2*TMath::Pi()*r*delta_r;
}

double get_delta_theta (double R, double d, double delta_R, double delta_d)
{
        return TMath::Sqrt( pow( ( delta_d/(R*TMath::Sqrt( 1-(pow(d, 2)/pow(R,2)) ) )  ) ,2) + pow(  ((-d*delta_R)/(pow(R,2)*TMath::Sqrt( 1-(pow(d,2)/pow(R,2)) ) ) ) , 2) );
}

double get_delta_surface_density(double M_e, double theta, double R, double n, double delta_M_e, double delta_R, double delta_theta)
{
        return TMath::Sqrt( pow( (((n+1)*pow(TMath::Cos(theta), n+1)*delta_M_e )/(2*TMath::Pi()*pow(R,2)) ) ,2) + pow( ((-M_e*pow(n+1, 2)*TMath::Sin(theta)*(TMath::Cos(theta), n)*delta_theta)/(2*TMath::Pi()*pow(R, 2))), 2) + pow( ((-M_e*(n+1)*pow( TMath::Cos(theta), n+1)*delta_R)/(TMath::Pi()*pow(R, 3))), 2) );
}


void  thickness_of_coating1()
{
        ///type in the coefficients

        double Gamma_e = 1e-3; //evaporation rate, unit: (kg/m^2) s^{-1}
        double n = 0.6;
        double t = 20*60; //unit: s
        double r_inch = (9.0/16.0); //r is the size of the evaporator, unit: inch
        double R_inch[6] = {20+7.0/8.0, 26+1.0/2.0, 22+3.0/8.0, 25+1.0/8.0, 22+3.0/8.0, 13.0+0.0}; //unit: inch; follow the sequence: 1. outer corner, 2. inner corner, 3. left corner, 4. right corner, 5. center, 6. quartz.
        double d_inch[6] = {4+5.0/8.0, 18+11.0/16.0, 11+5.0/8.0, 14+1.0/4.0, 11+0.0/1.0, 7+5.0/8.0}; //unit: inch
        double rho[2] = {7140.0, 2710.0};
        //rho_{Chromium} = 7140 kg/m^3, rho_{Aluminum} = 2710 kg/m^3


        /// type in the uncertainties for r, R, and theta

        double delta_r_inch = 1.0/32; //unit: inch
        double delta_R_inch[6] = {1.0/4, 1.0/2, 1.0/3, 1.0/3, 1.0/3, 1.0/4}; //unit: inch
        double delta_d_inch[6] = {1.0/5, 1.0/3, 1.0/4, 1.0/4, 1.0/4, 1.0/5}; //unit: inch

```

```
        ///convert the unit of r, R, d, delta_R, delta_d into m
        double r = r_inch * 0.0254; //unit: m
        double delta_r = delta_r_inch * 0.0254; //unit: m
        double R[6] = {0, 0, 0, 0, 0, 0};
        double d[6] = {0, 0, 0, 0, 0, 0};
        double delta_R[6] = {0, 0, 0, 0, 0, 0};
        double delta_d[6] = {0, 0, 0, 0, 0, 0};
        for (int k = 0; k < 6 ; ++k) {
          R[k] = R_inch[k] * 0.0254; //unit: m
          d[k] = d_inch[k] * 0.0254; //unit: m
          delta_R[k] = delta_R_inch[k] * 0.0254;
          delta_d[k] = delta_d_inch[k] * 0.0254;
        }


        ///calculate the total evaporated mass and its uncertainty
        double A_e = TMath::Pi() * pow(r, 2); //surface area of the source, unit: m^2
        double delta_A_e = get_delta_A_e(r, delta_r);
        double M_e = Gamma_e * A_e *t; //total evaporated mass, unit: kg
        double delta_M_e = Gamma_e * t * delta_A_e;
        cout<<M_e<<endl;

        //calculate the theta (= phi) and its uncertainty
        double theta[6] = {0, 0, 0, 0, 0, 0};
        double delta_theta[6] = {0, 0, 0, 0, 0, 0};
        for (int l = 0; l < 6; ++l){
          theta[l] = TMath::ASin(d[l]/R[l]);
          double delta_theta = get_delta_theta( R[l], d[l], delta_R[l], delta_d[l] );
        }
        cout<<theta[4]<<endl;

        ///some edits for cout
        const char *name_of_material[] = {"Chromium", "Aluminum"};
        const char *name_of_corner[] = {"outer corner", "inner corner", "left corner", "right corner", "center", "quartz"};


        ///calculate the thicknesses and print them out
    for (int j = 0; j < 2; ++j) {
        for (int i = 0; i < 6; ++i) {
            double surface_density = get_surface_density( M_e, theta[i], R[i], n );
            double thickness = get_thickness( surface_density, rho[j] );

             // Calculate thickness using increased R, M_e, and decreased theta
            double delta_surface_density = get_delta_surface_density( M_e, theta[i], R[i], n, delta_M_e, delta_R[i], delta_theta[i] );
            double delta_thickness = delta_surface_density / rho[j];

            cout << std::setprecision(10);
            cout << "Thickness for " << name_of_material[j] << " at the " << name_of_corner[i]
                 << " is " << thickness * 1e+10 << " angstroms, with an uncertainty of "
                 <<delta_thickness * 1e+10 << " angstroms." << endl;
        }
        cout<<endl;
    }
```

And I got the results in angstroms:

```
Thickness for Chromium at the outer corner is 937.729478 angstroms, with an uncertainty of 106.585576 angstroms.
Thickness for Chromium at the inner corner is 349.4456227 angstroms, with an uncertainty of 41.00543497 angstroms.
Thickness for Chromium at the left corner is 660.6645766 angstroms, with an uncertainty of 76.00063859 angstroms.
Thickness for Chromium at the right corner is 494.0331453 angstroms, with an uncertainty of 56.43608559 angstroms.
Thickness for Chromium at the center is 681.0347308 angstroms, with an uncertainty of 78.34395286 angstroms.
Thickness for Chromium at the quartz is 1796.552642 angstroms, with an uncertainty of 211.2379915 angstroms.

Thickness for Aluminum at the outer corner is 2470.623053 angstroms, with an uncertainty of 280.819562 angstroms.
Thickness for Aluminum at the inner corner is 920.679611 angstroms, with an uncertainty of 108.0364597 angstroms.
Thickness for Aluminum at the left corner is 1740.64394 angstroms, with an uncertainty of 200.2378449 angstroms.
Thickness for Aluminum at the right corner is 1301.622383 angstroms, with an uncertainty of 148.6913842 angstroms.
Thickness for Aluminum at the center is 1794.312907 angstroms, with an uncertainty of 206.411743 angstroms.
Thickness for Aluminum at the quartz is 4733.352716 angstroms, with an uncertainty of 556.5458522 angstroms.
```

Or in a clearer way:

	Chromium	Aluminum
1. thickness at the outer corner (angstrom)	938	2471
uncertainty (angstrom)	107	281
2. thickness at the inner corner (angstrom)	349	921
uncertainty (angstrom)	41	108
3. thickness at the left corner (angstrom)	661	1741
uncertainty (angstrom)	76	200
4. thickness at the right corner (angstrom)	494	1302
uncertainty (angstrom)	56	149
5. thickness at the center (angstrom)	681	1794
uncertainty (angstrom)	78	206
6. thickness at the quartz (angstrom)	1797	4733
uncertainty (angstrom)	211	557

Remark:

1. I would assume that **if considering in the time elasped**, **the thickness of the coated layer on each mirror will be exactly the same**, which will be **a bit larger than the thickness at the imagined center mirror**, since the source is on the outer side. (more precise value needs more complex calculations)
2. If considering in the change of spectral power n, the problem will be way more complex.