Transient Thermal Lensing and Thermoelastic Abberations of Mirrors

The formulation is limited to right circular cylindrical mirrors with homogeneous surface or in-depth absorption:
Patrice Hello and Jean-Yves Vinet, "Analytical models of thermal aberrations in massive mirrors heated by high power laser beams", J. Phys. France, 51, (1990), 1267-1282
"Analytical models of transient thermoelastic deformations of mirrors heated by high power cw laser beams", J. Phys. France, 51, (1990), 2243-2261.

Dennis Coyne

Version History

Version 1.0 Initial

Version 2.0 restructured to have all inputs at beginning of the notebook and made transient calculations inactive for now.

Version 2.1 (a) changed the Zernike expansion to have the optical zone radius, b, be an input variable so that the Zernikes can be calculated for a nominal optical zone radius b and for the optic radius, (b) calculated the errors associated with the Zernike expansion

Version 2.2 Corrected "t" references to "tau" in the transcendental solutions for ur and vr. Implemented Zernike fit comparison to calculated surface/wavefront more uniformly in the notebooks (was only used in some and in only parts of the notebook.) Discovered and corrected error in the transcendental solutions ur and vr as used in the "analytic_ITM" notebook. Moved the calculation of sp and cp to after calculation of ur and vr. Corrected the calculation of the OPD at the center from the Zernike decomposition (changed from a sum of absolute amplitudes to an alternating signed summation).

Version 2.3 Applied to LIGO Optical Contamination Facility (OCF). See T970212

Preliminaries & Setup

Initialization

In[1]:=
```
Off[General::"spell"]
Off[General::"spell1"]
```

Input Parameters

Optical Contamination Cavity Optic

a = optic radius, m
h = optic thickness, m
k = optic thermal conductivity, W/m/K
w = Gaussian beam waist, m
P = incident power, W
Tx = ambient temperature, K

eps = coating absorptivity
alpha = in-depth absorption, 1/m
Note: If the coating absorptivity == 1 or the in-depth absorption == 1/h , then the incident power == absorbed power.

ro = mass density, kg/m^3
c = specific heat, J/kg/K
dndT = change in refraction index with temperature, 1/K
b = radius(?) for Zernike decomposition, m
lambda = first Lame coefficient, J/m^3
mu = second Lame coefficient, J/m^3

In[3]:=
```
a = 0.0254 / 2;
h = 0.0064;
k = 1.38;
w = 0.0004;
P = 1;
Tx = 293;
```

required for surface heating:

In[9]:=
```
eps = 1;
```

required for in-depth heating:

In[10]:=
```
alpha = 1/h;
```

required for transient analysis:

In[11]:=
```
ro = 2202;
c = 745;
```

required for thermal lensing:

In[13]:=

$$\text{dndT} = \frac{11.8}{10^6};$$

required for Zernike decomposition:

In[14]:=
```
b = 0.05;
```

required for thermoelastic analysis:

In[15]:=

$$\text{lambda} = 15.6\times10^9;$$
$$\text{mu} = 31.3\times10^9;$$

In[17]:=

$$\text{ni} = 5.91\times10^4;$$

Constants

In[18]:=
```
s = UnitConvert[Quantity["StefanBoltzmannConstant"],
   "Watts" / "Kelvins"^4 / "Meters"^2][[1]]
```

Out[18]=

$$5.6704\times10^{-8}$$

Derived Parameters

tau = reduced radiation constant
tc= thermal time constant (for the entire optic), sec

In[19]:=

$$\text{tau} = \frac{4\ s\ \text{Tx}^3\ a}{k};$$

In[20]:=

$$tc = \frac{ro\,c\,a^2}{k}$$

Out[20]= 191.735

In[21]:=

$$\frac{tc}{3600}$$

Out[21]= 0.0532598

Axisymmetric Zernikes

In[22]:=

$$Z2q[q\_, r\_] := \sum_{i=0}^{q} \frac{(-1)^i\,(2\,q-i)!\,r^{2\,q-2\,i}}{i!\,(q-i)!\,(q-i)!};$$

Surface Absorption

Steady-State Thermal Field Solution

1st Transcendental Equation Solution

Equation Plot

Finds roots of the equation u J1(u) = c J0(u). The user defines the constant, c, the root search interval, du, and the upper limit of u, upmax.

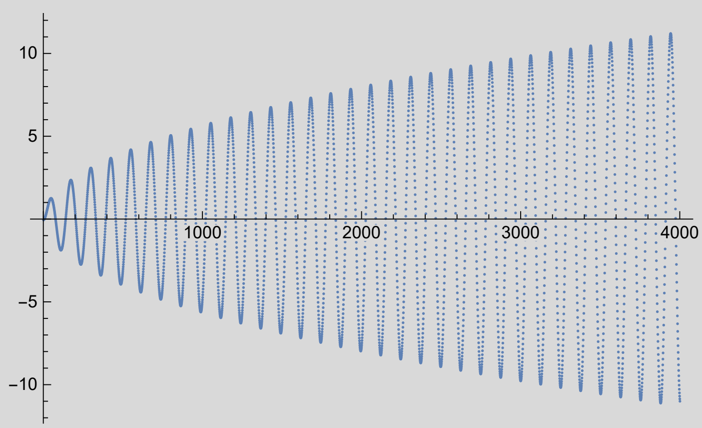
In[23]:=
```
du = 0.05;
upmax = 200;
c = tau;
```

In[26]:=
```
n = upmax/du + 1
up = Table[i du, {i, n}];
```

Out[26]=
```
4001.
```

In[28]:=
```
e = N[up BesselJ[1, up] - c BesselJ[0, up]];
ListPlot[e]
```

Out[29]=



Solution Method:

If the parameters are unchanged, then one can skip the calculation of the transcendental equation solution and go directly to the list of roots below.

In[30]:=
```
es = Rest[e RotateRight[e]];
esw = Select[es, #1 < 0 &];
npos = Dimensions[esw][[1]];
pos = Table[0, {npos}];
Do[pos[[i]] = Position[es, esw[[i]]], {i, 1, npos}];
pos = Flatten[pos];
nu = Dimensions[pos][[1]];
```

In[37]:=
```
ur = Table[0, {nu}];
Do[umin = up[[pos[[i]]]];
  umax = up[[pos[[i]] + 1]];
  ustart = (umax + umin)/2;
  ur[[i]] = u /. FindRoot[u BesselJ[1, u] == c BesselJ[0, u],
     {u, ustart, umin, umax}, AccuracyGoal → 7, PrecisionGoal → 7], {i, 1, nu}];
```

In[39]:=
```
ur
```

Out[39]=
```
{0.321937, 3.84538, 7.02307, 10.1786, 13.3276, 16.4738, 19.6185, 22.7624,
 25.9057, 29.0486, 32.1913, 35.3338, 38.4761, 41.6184, 44.7605, 47.9026,
 51.0446, 54.1865, 57.3284, 60.4703, 63.6122, 66.754, 69.8958, 73.0376,
 76.1794, 79.3211, 82.4629, 85.6046, 88.7464, 91.8881, 95.0298, 98.1715,
 101.313, 104.455, 107.597, 110.738, 113.88, 117.022, 120.163, 123.305,
 126.447, 129.588, 132.73, 135.872, 139.013, 142.155, 145.296, 148.438,
 151.58, 154.721, 157.863, 161.005, 164.146, 167.288, 170.43, 173.571,
 176.713, 179.854, 182.996, 186.138, 189.279, 192.421, 195.562, 198.704}
```

In[40]:=
```
N[(ur - RotateRight[ur]) / Pi]
```

Out[40]=
```
{-63.147, 1.12155, 1.01149, 1.00445, 1.00236, 1.00146, 1.00099, 1.00072,
 1.00055, 1.00043, 1.00034, 1.00028, 1.00024, 1.0002, 1.00017, 1.00015,
 1.00013, 1.00012, 1.0001, 1.00009, 1.00008, 1.00008, 1.00007, 1.00006,
 1.00006, 1.00005, 1.00005, 1.00005, 1.00004, 1.00004, 1.00004, 1.00003,
 1.00003, 1.00003, 1.00003, 1.00003, 1.00003, 1.00002, 1.00002, 1.00002,
 1.00002, 1.00002, 1.00002, 1.00002, 1.00002, 1.00002, 1.00002, 1.00001,
 1.00001, 1.00001, 1.00001, 1.00001, 1.00001, 1.00001, 1.00001, 1.00001,
 1.00001, 1.00001, 1.00001, 1.00001, 1.00001, 1.00001, 1.00001, 1.00001}
```

In[41]:=
```
mmax = Dimensions[ur][[1]];
zetam = ur;
```

Roots:

If the parameters are unchanged, then skip calculation above and evaluate this list of roots.

```
mmax =;
zetam = {};
```

Coefficients of the Dini Series Representation of the Incident Radiation

In[43]:=

```
zetam2 = zetam^2;
gamma = (zetam h)/(2 a);
```

In[45]:=

```
pm = (P zetam2 Exp[-(w^2 zetam2)/(8 a^2)]) / (π a^2 ((zetam2 + tau^2) BesselJ[0, zetam]^2));
```

Steady-State Temperature Distribution for Radiant Surface Heating

In[46]:=

```
T[r_, z_] := Sum[((pm[[m]] a) Exp[-(h zetam[[m]])/(2 a)] ((zetam[[m]] - tau) Exp[-(zetam[[m]] (h - z))/a] +
      (zetam[[m]] + tau) Exp[-(zetam[[m]] z)/a]) BesselJ[0, (zetam[[m]] r)/a]) /
    (k ((zetam[[m]] + tau)^2 - (zetam[[m]] - tau)^2 Exp[-(2 zetam[[m]] h)/a])), {m, 1, mmax}];
```

In[47]:=

```
temp = Table[T[i, j], {i, 0, a, a/50}, {j, -h/2, h/2, h/25}];
```

In[48]:=

```
Max[temp]
```

Out[48]= 821.837

In[49]:=

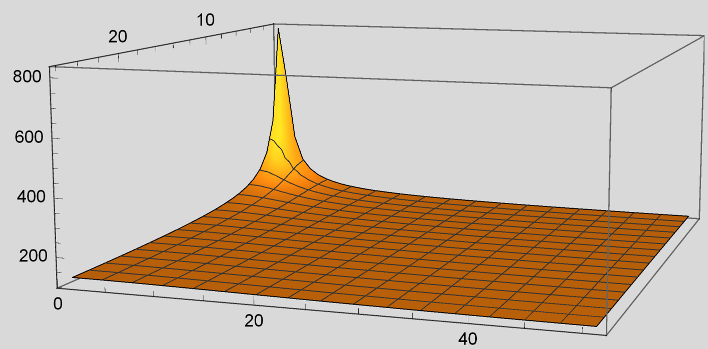
```
Min[temp]
```

Out[49]= 110.39

In[50]:=

```
ListPlot3D[temp, PlotRange → All,
 MeshRange → {{-h/2, h/2}, {0, a}}, ViewPoint → {3, 1, .5}]
```

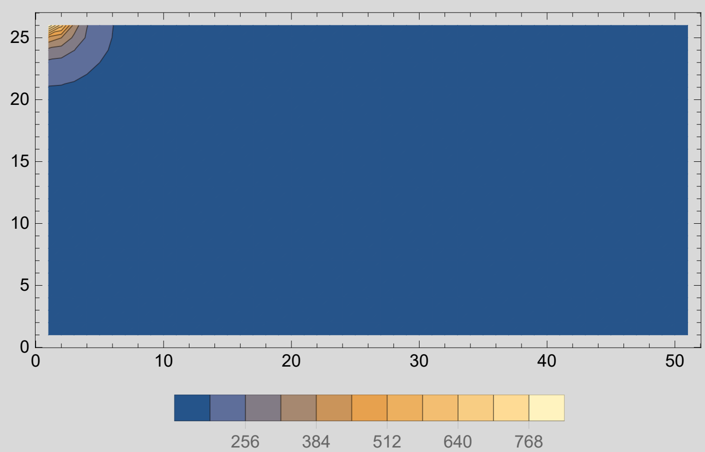
Out[50]=



In[51]:=

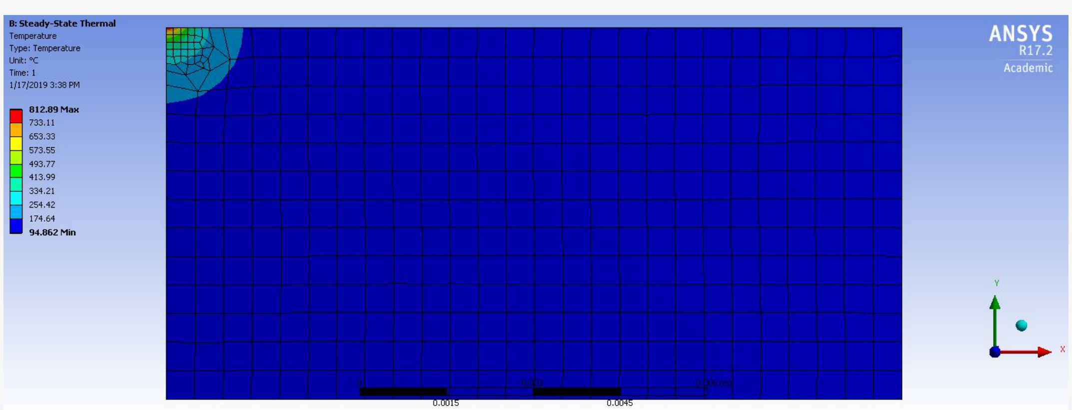
```
ListContourPlot[Reverse[Transpose[temp]], PlotRange → All,
 Contours → 10, AspectRatio → h / a, PlotLegends → Automatic]
```

Out[51]=



Compare to ANSYS nonlinear, steady - state thermal reponse for the OCF mirror with 1 Watt absorbed power :

max temperature = 813 C



Transient Thermal Field Solution

2nd Transcendental Equation Solution

Solution Method:

Finds roots of the equation u = t Cot[u h/(2 a)]. The user defines the constant, t, the root search interval, du, and the upper limit of u, upmax.

In[52]:=
```
dui = 0.0001;
duj = 2 a π / h;
upmax = 120;
offset = 0;
```

In[56]:=
```
nj = Floor[upmax / duj + 1];
ni = Floor[duj / (10 dui) + 1];
up = Flatten[Table[i dui + j duj + offset, {j, 0, nj}, {i, -ni, ni}]];
```

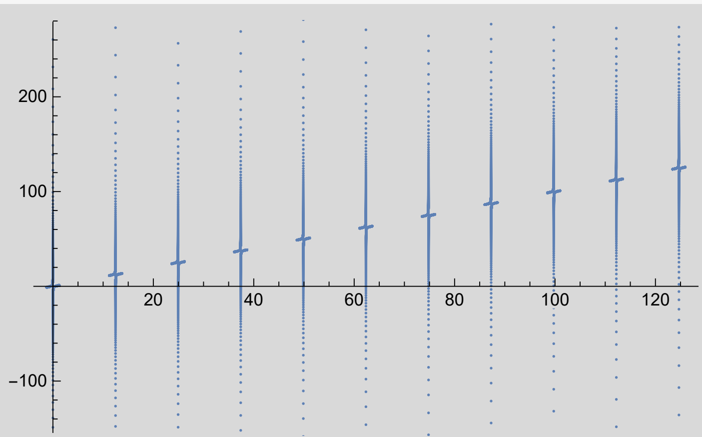
In[59]:=
```
Dimensions[up]
```

Out[59]=
```
{274 329}
```

In[60]:=
```
e = N[up - tau Cot[up h / (2 a)]];
n = Dimensions[e][[1]];
```

In[62]:=
```
ListPlot[Transpose[Partition[Join[up, e], n]]]
```

Out[62]=



In[63]:=
```
es = Rest[e RotateRight[e]];
esw = Select[es, #1 < 0 &];
esw = Transpose[Partition[esw, 2]][[1]];
npos = Dimensions[esw][[1]];
pos = Table[0, {npos}];
Do[pos[[i]] = Position[es, esw[[i]]], {i, 1, npos}];
pos = Flatten[pos];
nu = Dimensions[pos][[1]];
```

Less::nord : Invalid comparison with ComplexInfinity attempted. >>

Less::nord : Invalid comparison with ComplexInfinity attempted. >>

In[71]:=
```
pos
```

Out[71]=
```
{7915, 17 024, 37 409, 62 348, 87 287, 112 226,
 137 164, 162 104, 187 043, 211 982, 236 921, 261 859}
```

In[72]:=
```
npos
```

Out[72]=
```
11
```

In[73]:=
```
ur = Table[0, {nu}];
Do[umin = up[[pos[[i]]]];
  umax = up[[pos[[i]] + 1]];
  ustart = (umax + umin)/2;
  ur[[i]] = u /. FindRoot[u == tau Cot[(u h)/(2 a)], {u, ustart, umin, umax},
     MaxIterations → 100, DampingFactor → 0.8], {i, 1, nu}];
```

FindRoot::reged : The point {12.4683} is at the edge of the search region
{12.4682, 12.4683} in coordinate 1 and the computed search direction points outside the region. >>

FindRoot::reged : The point {24.9365} is at the edge of the search region
{24.9364, 24.9365} in coordinate 1 and the computed search direction points outside the region. >>

FindRoot::reged : The point {37.4047} is at the edge of the search region
{37.4046, 37.4047} in coordinate 1 and the computed search direction points outside the region. >>

General::stop : Further output of FindRoot::reged will be suppressed during this calculation. >>

In[75]:=
```
ur
```

Out[75]=
```
{-0.455481, 0.455481, 12.4683, 24.9365, 37.4047,
 49.8729, 62.3409, 74.8093, 87.2775, 99.7457, 112.214, 124.682}
```

In[76]:=
```
N[(ur - RotateRight[ur])/duj]
```

Out[76]=
```
{-10.0365, 0.0730628, 0.963477, 1.,
 1., 1., 0.999984, 1.00002, 1., 1., 1., 0.999984}
```

In[77]:=
```
ur2 = ur^2;
```

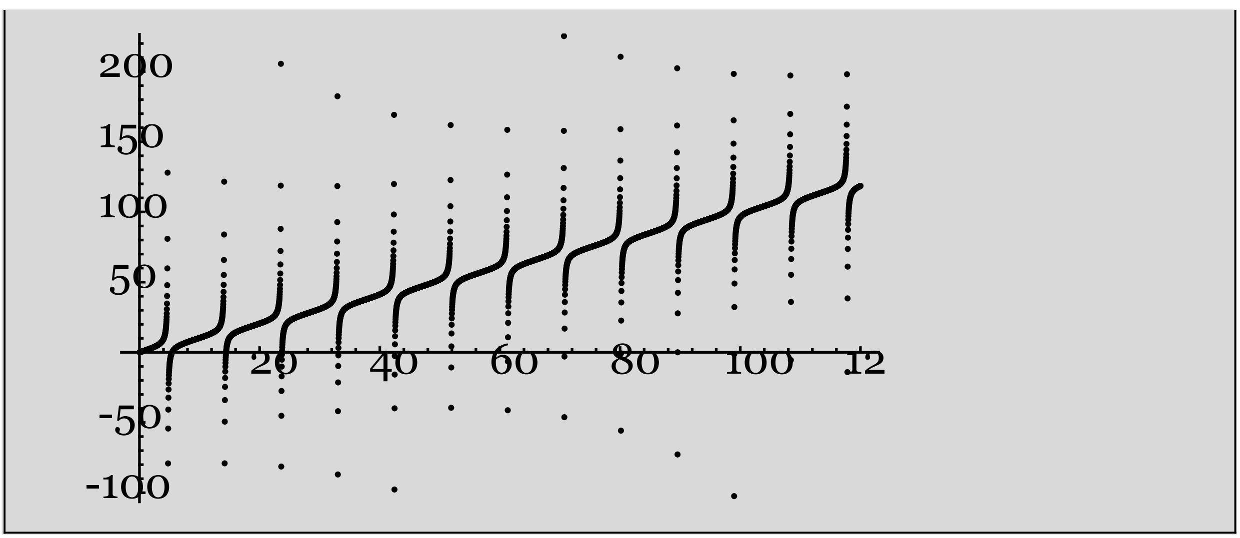
Roots:

```
imax =;
ur = {};
ur2 = ur^2;
```

3rd Transcendental Equation Solution

Finds roots of the equation u = -t Tan[v h/(2 a)]. The user defines the constant, t, the root search interval, du, and the upper limit of v, vpmax.

Equation Plot:



Solution Method:

In[78]:=
```
dvi = 0.002;
dvj = 2 a π / h;
vpmax = 120;
offset = dvj / 2;
```

In[82]:=
```
nj = Floor[vpmax / dvj + 1];
ni = Floor[dvj / (10 dvi) + 1];
vp = Flatten[Table[i dvi + j dvj + offset, {j, 0, nj}, {i, -ni, ni}]];
```

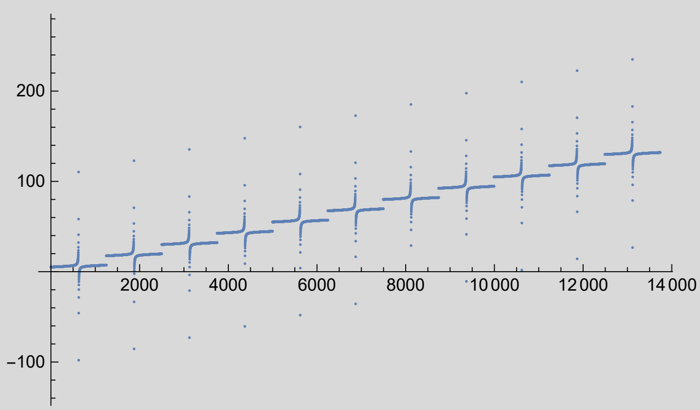
In[85]:=
```
Dimensions[vp]
```

Out[85]=
```
{13 739}
```

In[86]:=
```
e = N[vp + tau Tan[vp h / (2 a)]];
n = Dimensions[e][[1]];
```

In[88]:=
```
ListPlot[e]
```

Out[88]=



In[89]:=
```
es = Rest[e RotateRight[e]];
esw = Select[es, #1 < 0 &];
esw = Transpose[Partition[Drop[esw, 1], 2]][[1]];
npos = Dimensions[esw][[1]];
pos = Table[0, {npos}];
Do[pos[[i]] = Position[es, esw[[i]]], {i, 1, npos}];
pos = Flatten[pos];
nu = Dimensions[pos][[1]];
```

In[97]:=
```
npos
```

Out[97]=
```
10
```

In[98]:=
```
vr = Table[0, {nu}];
Do[vmin = vp[[pos[[i]]]];
  vmax = vp[[pos[[i]] + 1]];
  vstart = (vmax + vmin)/2;
  vr[[i]] = v /. FindRoot[v == -tau Tan[(v h)/(2 a)], {v, vstart, vmin, vmax},
     DampingFactor → 0.8, MaxIterations → 100], {i, 1, nu}];
```

In[100]:=
```
vr
```

Out[100]=
```
{6.26735, 18.7134, 31.1772, 43.6435,
 56.1106, 68.5781, 81.0458, 93.5137, 105.982, 118.45}
```

In[101]:=
```
N[(vr - RotateRight[vr])/dvj]
```

Out[101]=
```
{-8.99747, 0.998227, 0.999643, 0.999847,
 0.999915, 0.999946, 0.999963, 0.999973, 0.999979, 0.999983}
```

In[109]:=
```
imax = Dimensions[vr][[1]];
vr2 = vr^2;
```

Roots:

```
imax =;
vr = {};
vr2 = vr^2;
```

Coefficients

In[103]:=
```
sp = 1 - (a Sin[vr h/a])/(vr h);
cp = 1 + (a Sin[ur h/a])/(ur h);
```

Transient Temperature Distribution for Radiant Surface Heating

In[112]:=
```
T[time_, r_, z_] :=
  1/(k h) (2 a^2 Sum[Sum[(pm[[m]] (1 - Exp[-(time k (ur[[i]]^2 + zetam2[[m]])) / (ro c a^2)])
        Cos[ur[[i]] h/(2 a)] Cos[ur[[i]] z/a] BesselJ[0, zetam[[m]] r/a]) /
       ((ur[[i]]^2 + zetam2[[m]]) cp[[i]]), {m, 1, mmax}], {i, 1, imax}]) - 1/(k h) (2 a^2
    Sum[Sum[(pm[[m]] (1 - Exp[-(time k (vr[[i]]^2 + zetam2[[m]])) / (ro c a^2)]) Sin[vr[[i]] h/(2 a)]
        Sin[vr[[i]] z/a] BesselJ[0, zetam[[m]] r/a]) / ((vr[[i]]^2 + zetam2[[m]]) sp[[i]]), {m, 1, mmax}], {i, 1, imax}])
```

In[113]:=
```
N[T[tc, 0, -h/2]]
```

Out[113]= 823.501

Evolution of the temperature at the center on the surface:

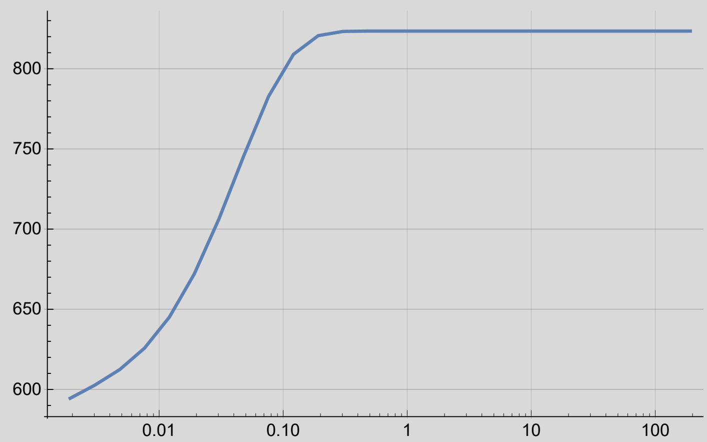
In[114]:=
```
ns = 5;
nd = 5;
nt = nd ns + 1;
to = 1/10^5;
fact = 10^(1/ns);
N[ts = Table[to fact^(i-1), {i, nt}];]
N[temper = Table[T[tc ts[[j]], 0, -h/2], {j, nt}];]
N[Max[temper]]
```

Out[121]= 823.501

In[183]:=

```
ListLogLinearPlot[Transpose[Partition[Join[tc ts, temper], nt]],
 Joined → True, GridLines → Automatic, PlotStyle → Thickness[0.005]]
```

Out[183]=



Diametrical cuts of the surface coating temperature field at various times:

In[170]:=

```
radDiv = 43;
rad = Table[i a / radDiv, {i, 0, radDiv}];
N[tempern = Table[T[tc / 100, rad[[i]], -h/2], {i, 1, radDiv + 1}];]
N[temper2 = Table[T[tc / 1000, rad[[i]], -h/2], {i, 1, radDiv + 1}];]
N[temper3 = Table[T[tc / 10 000, rad[[i]], -h/2], {i, 1, radDiv + 1}];]
N[temper4 = Table[T[tc / 100 000, rad[[i]], -h/2], {i, 1, radDiv + 1}];]
```

In[182]:=

```
tc / 100
```

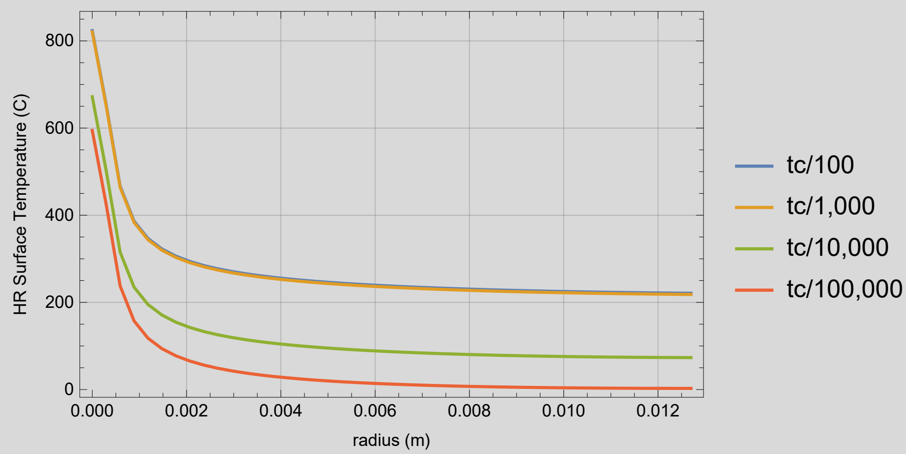
Out[182]=

```
1.91735
```

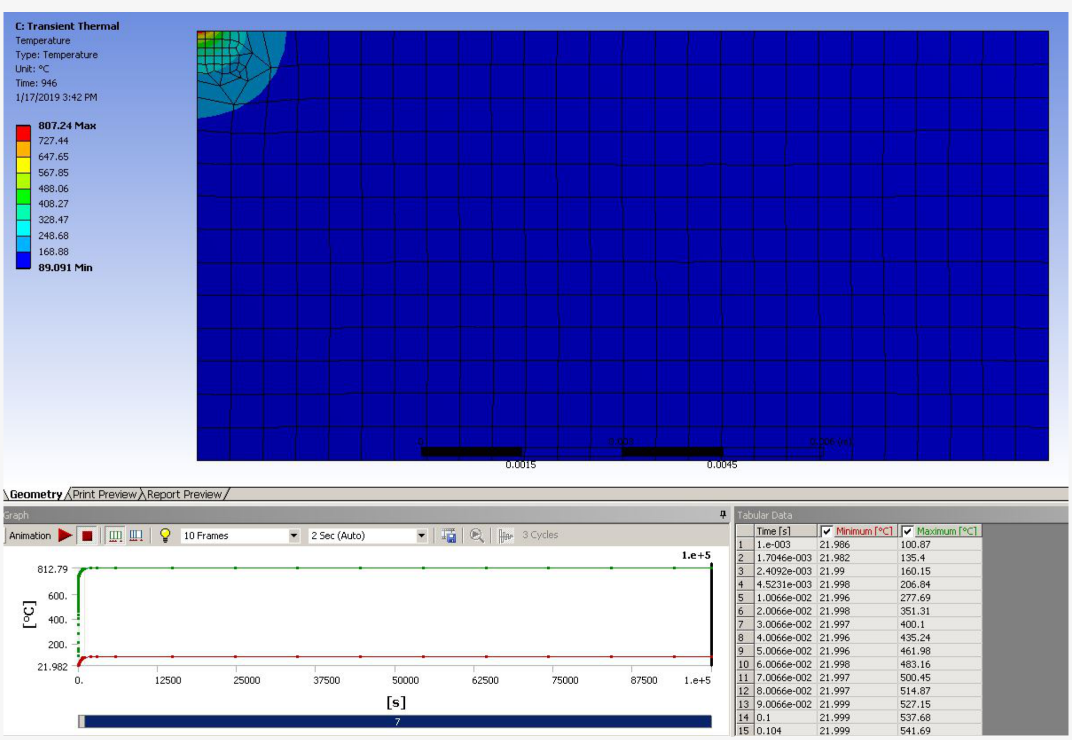
In[180]:=

```
ListPlot[{Transpose[Partition[Join[rad, tempern], radDiv + 1]],
   Transpose[Partition[Join[rad, temper2], radDiv + 1]],
   Transpose[Partition[Join[rad, temper3], radDiv + 1]],
   Transpose[Partition[Join[rad, temper4], radDiv + 1]]},
  Joined → True, GridLines → Automatic,
  PlotRange → All, PlotStyle → Thickness[0.005],
  PlotLegends → {"tc/100", "tc/1,000", "tc/10,000", "tc/100,000"},
  Frame → True, FrameLabel → {"radius (m)", "HR Surface Temperature (C)"}]
```

Out[180]=



Compares reasonably well with an ANSYS nonlinear, transient thermal reponse for the OCF mirror with 1 Watt absorbed power :



	Time [s]	Minimum [°C]	Maximum [°C]
1	1.e-003	21.986	100.87
2	1.7046e-003	21.982	135.4
3	2.4092e-003	21.99	160.15
4	4.5231e-003	21.998	206.84
5	1.0066e-002	21.996	277.69
6	2.0066e-002	21.998	351.31
7	3.0066e-002	21.997	400.1
8	4.0066e-002	21.996	435.24
9	5.0066e-002	21.996	461.98
10	6.0066e-002	21.998	483.16
11	7.0066e-002	21.997	500.45
12	8.0066e-002	21.997	514.87
13	9.0066e-002	21.999	527.15
14	0.1	21.999	537.68
15	0.104	21.999	541.69

Warning : The balance of this note-book has not been debugged for the OCF mirror reponse.

Wavefront Aberrations due to Index Gradients

Formulation:

Steady-State Aberrations:

Transient Aberrations:

Thermoelastic Analysis

Steady-State Wavefront Aberrations due to Surface Distortion:

Transient Wavefront Aberrations due to Surface Distortion

Bulk Absorption

Comparisons