Thermal PID Controller and Thermostat Ansys Mechanical Extensions

THERMAL PID CONTROLLER ACT Extension (Rev 1)

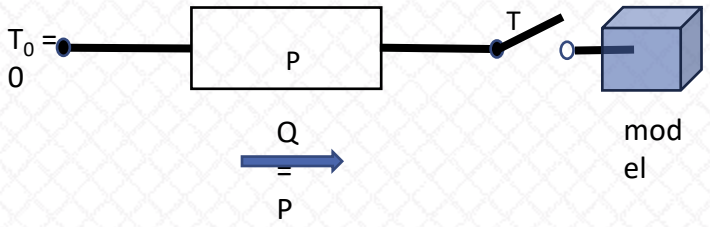
Formerly “PID Thermostat Controller”

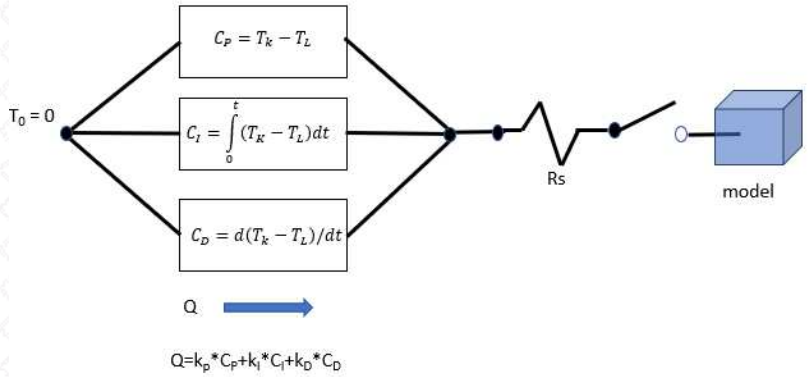
THERMOSTAT ACT Extension (Rev 0)

Alex Grishin, PhD

PADT, Inc

July, 2023





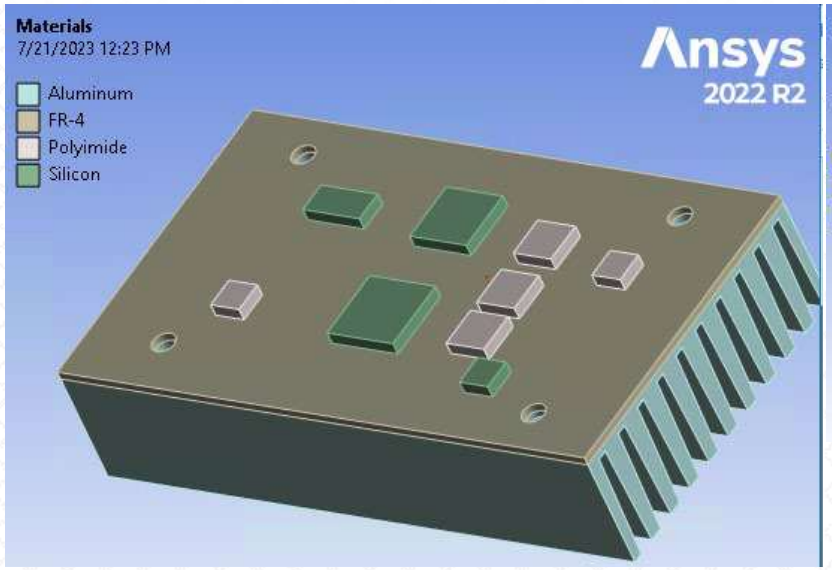
Introducing Two New ANSYS ACT Extensions

- In February, 2023, PADT began hosting an ACT extension which was previously, but no longer available on the ANSYS App Store
- You can read about the details and history of this app in the link. In this post, we're introducing a new version of this extension, as well as an entirely new (but related) extension
- You can read about the specific enhancements made in the accompanying release notes, but we thought it was important to change the name. This app is a thermal PID controller –not a thermostat
- The distinction is important, as a thermostat is typically a constant-power device, operating between two states (sometimes called a big bang controller). This is not what the PID controller does
- Simulating both types of control is important, so we changed the name of our controller appropriately (a variable-power device used to track a target temperature by varying power accordingly), and introduced a new true thermostat device.

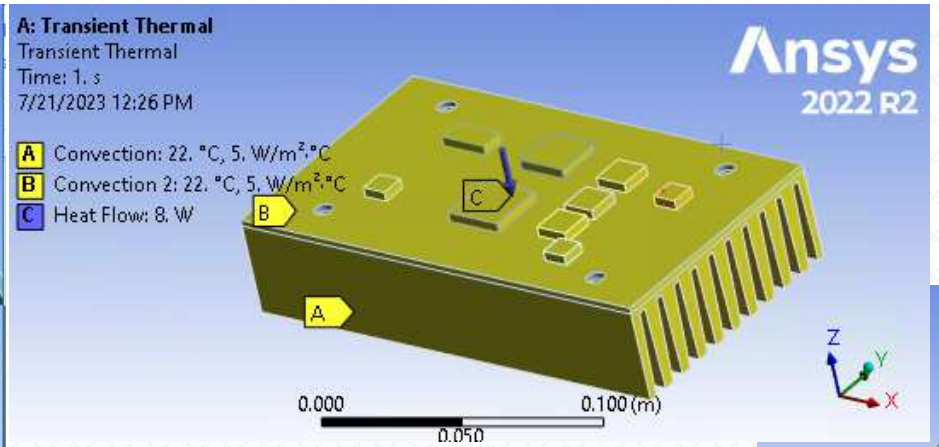
- NOTE: All Workbench examples accompanying this post are in version 2023R1!

The PID Controller: What It Does (Recap)

- Before describing the latest enhancement, we'll demonstrate the main feaures of the PID controller on the PCB board model below (a 2022R2 archive of this model is available with this post)

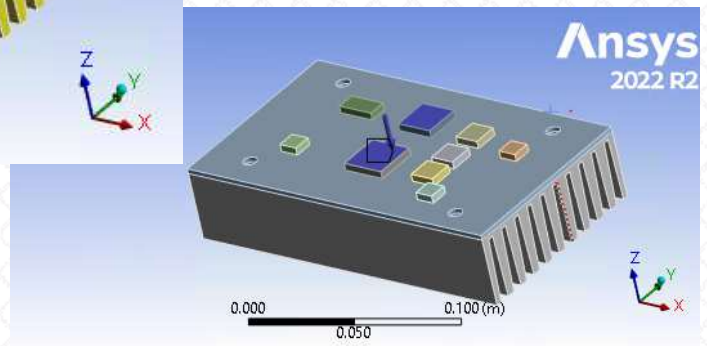


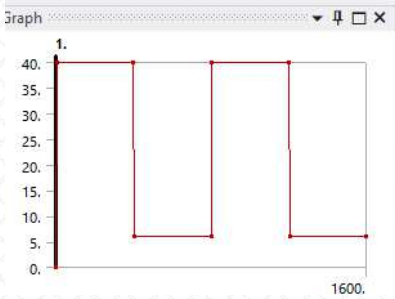
A pcb board (FR-4) with multiple TTL components connected to an Aluminum heatsink

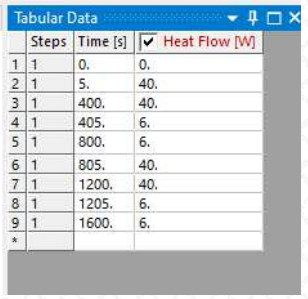


- All surfaces experience a uniform convection of 5 W/(m^2 °C) to 22 °C ambient air

- 2 TTL components generate heat with 40 W pulses over 1600s as shown

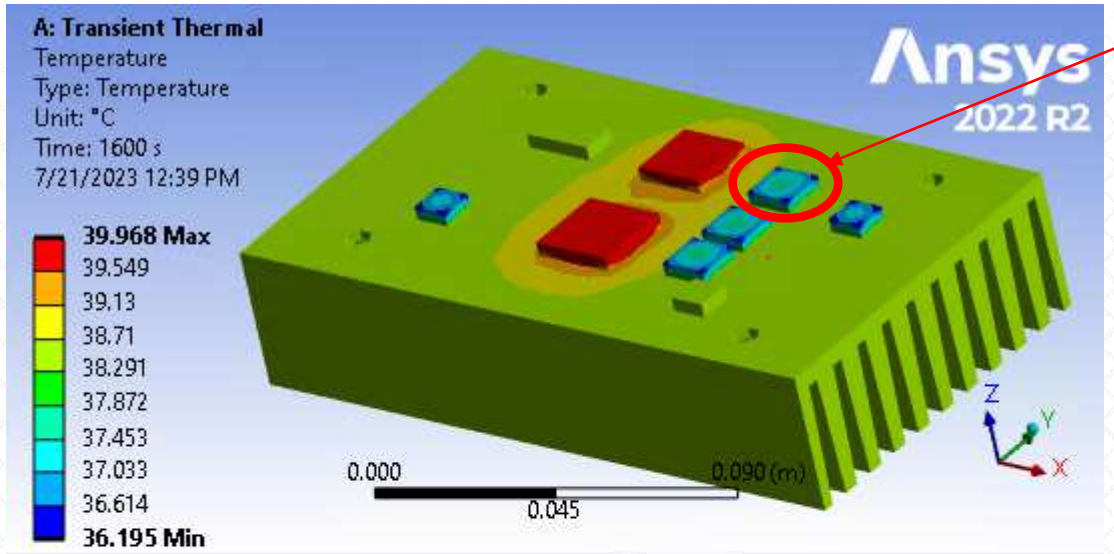


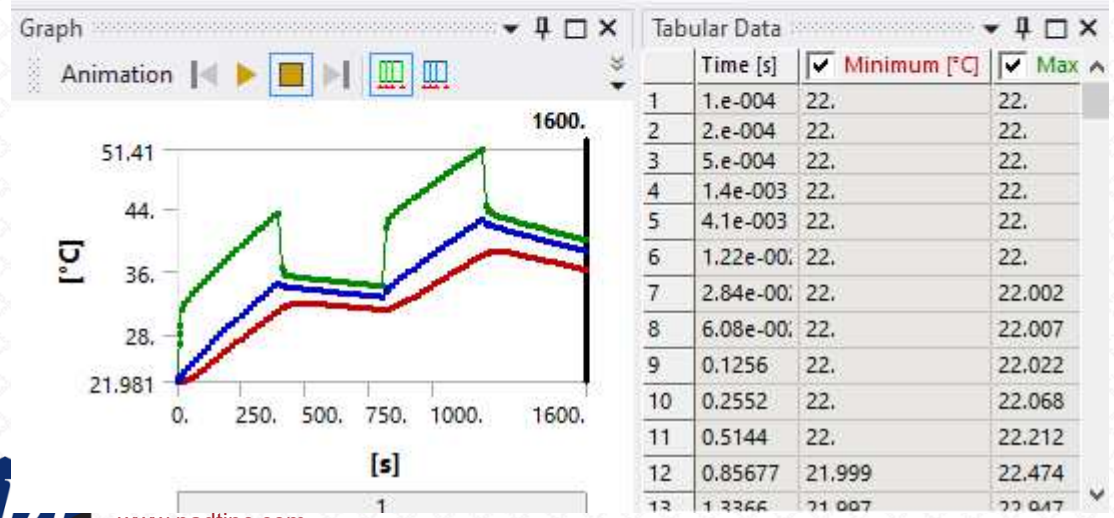




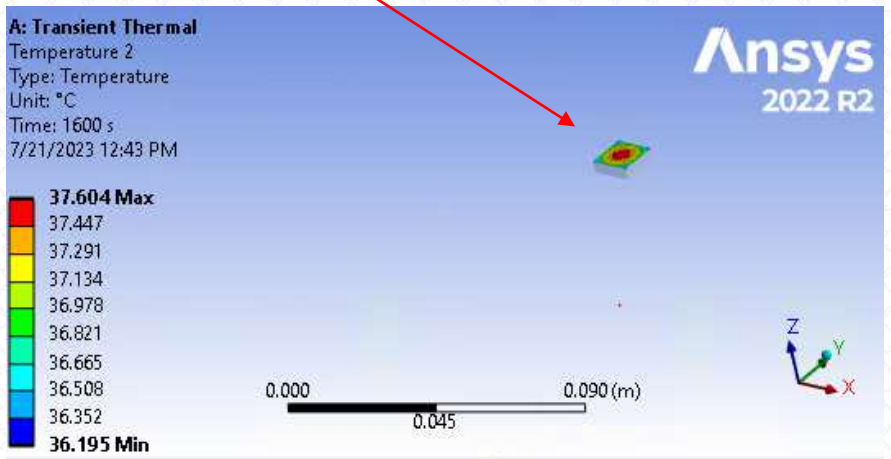
The PID Controller: What It Does (Recap)

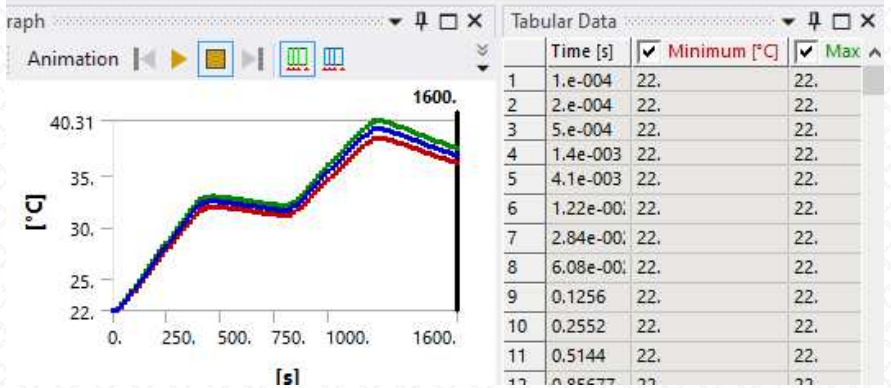
- The resulting temperature after 1600 s is shown below





- Suppose we're interested in the temperature of this TTL component...

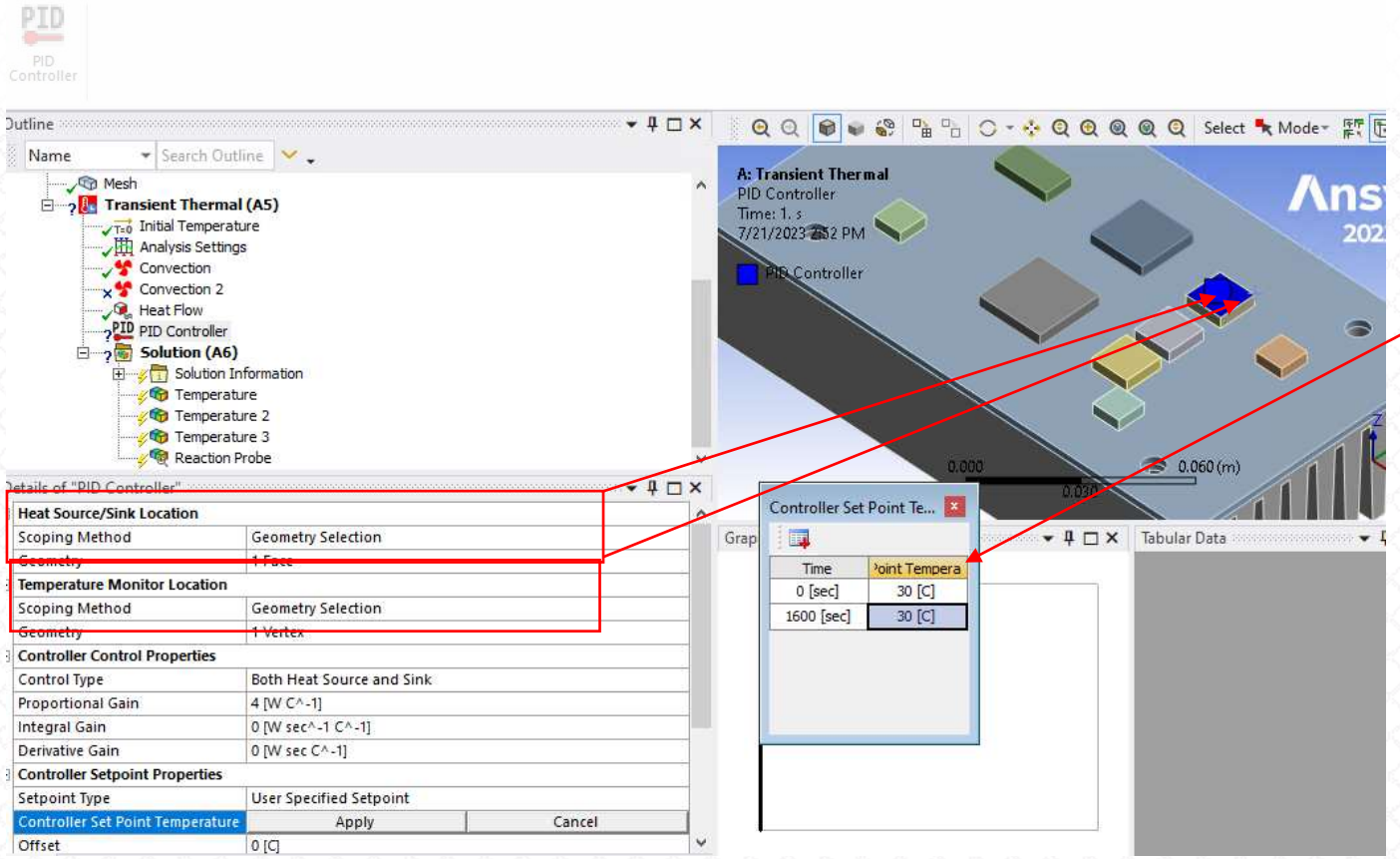




The PID Controller: What It Does (Recap)

- Let's say that a design requirement is that this component must be kept at or below 30°C (perhaps with a Peltier device). Let's further assume that we need to do this with some sort of 'active control' device. In other words, the amount of (variable) power required to achieve this must be determined.
- We can use the PID controller do this

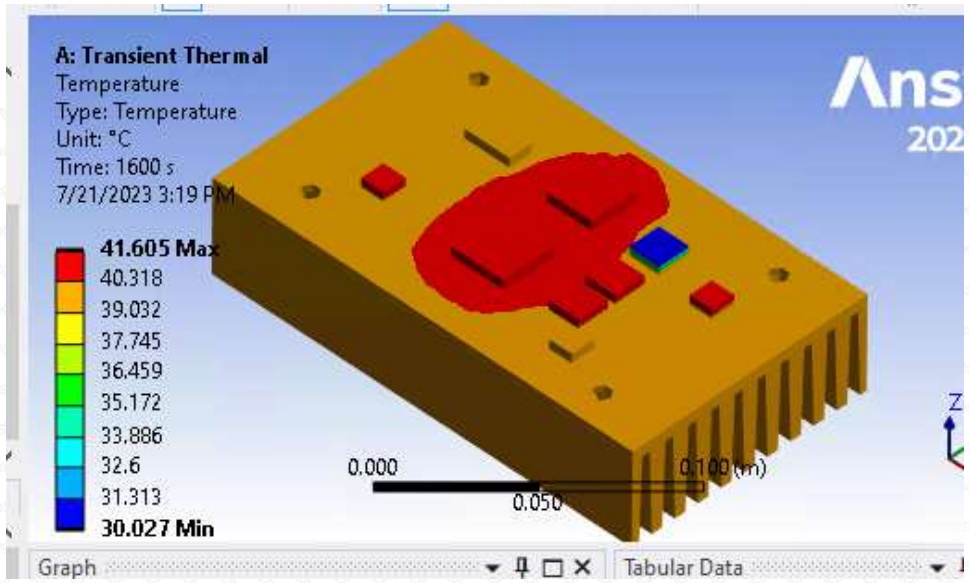
- Specify the top surface of the TTL component to be the "Heat Source/Sink Location"
- Specify a corner (vertex) of the component to be "Temperature Monitor Location"

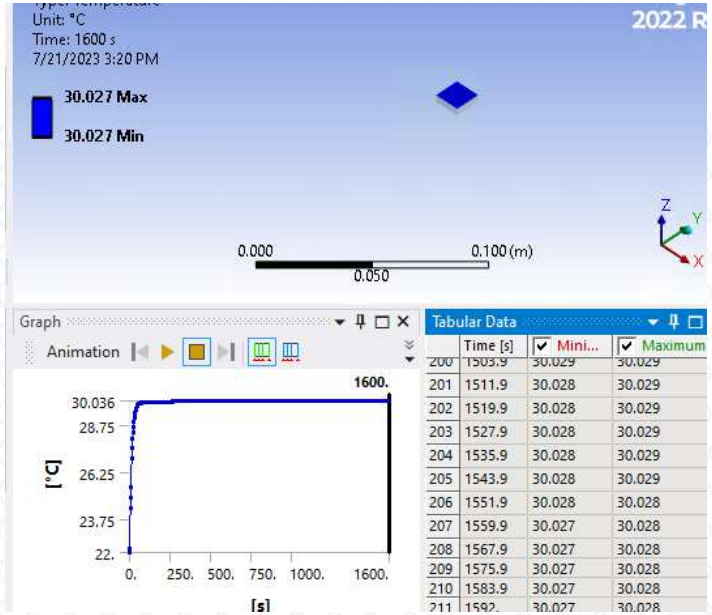


- Define a table for "Thermost at Setpoint Properties

The PID Controller: What It Does (Recap)

- In this example, we'll accept the default setting for "Control Type" ("Both Heat Source and Sink")
- In this mode, we can both add heat and extract it as needed. In general, this will result in the fastest convergence to a setpoint temperature in an environment with fluctuating heat flows.
- Finally, set the "Proptortional Gain" to 4 W/°C and set both the "Integral Gain" and "Derivative Gain" to zero as shown
- Run the model...

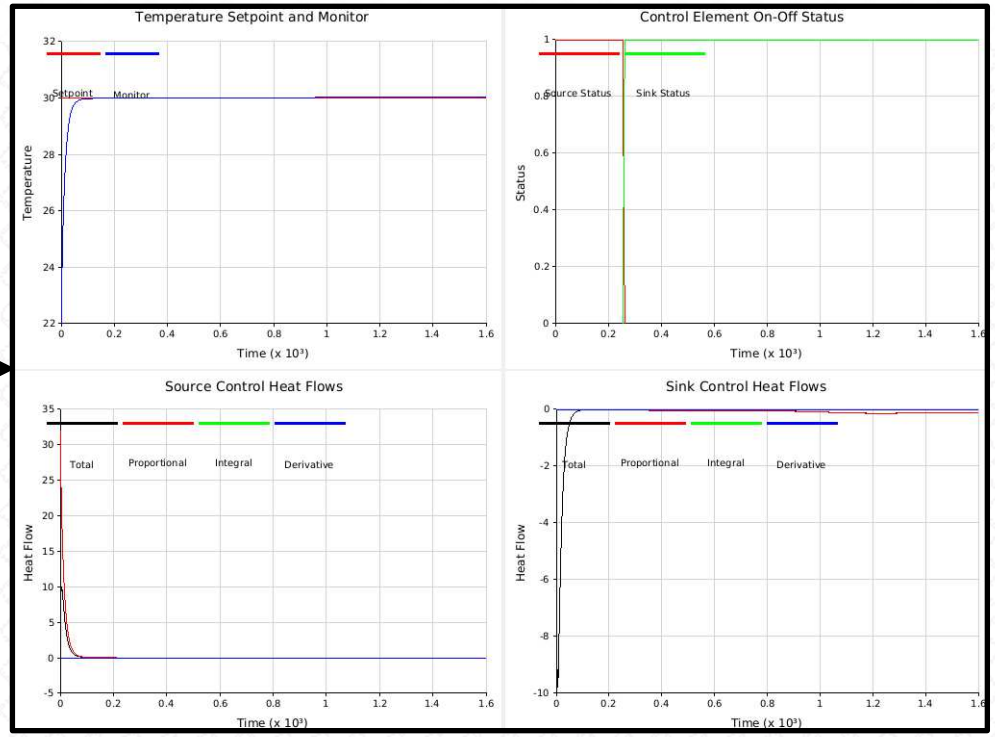


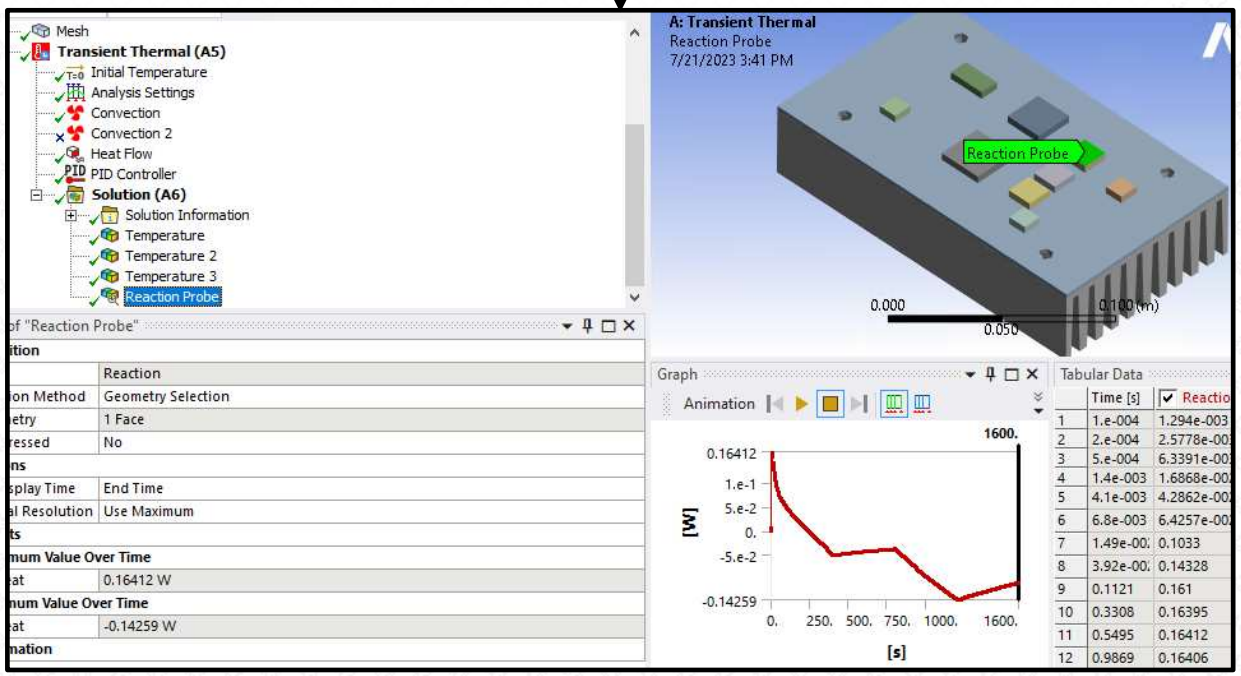


- At the end of 1600 s, you should see something like this...
- The target component should be ~30°C

The PID Controller: What It Does (Recap)

- Now, lets review the heat flow (power) to the TTL surface required to keep it cool at 30°C
- You can do this two ways:
  1. By adding a reaction probe to the TTL surface
  2. By selecting “View Results?” in the controller details view

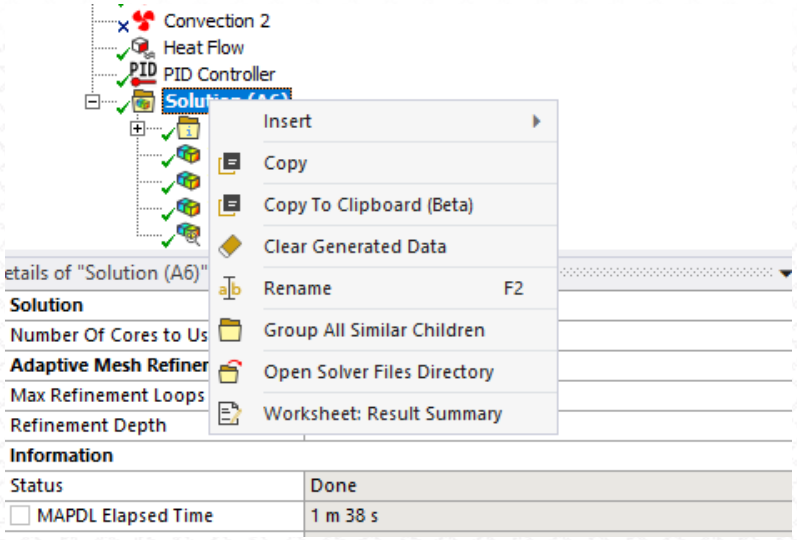


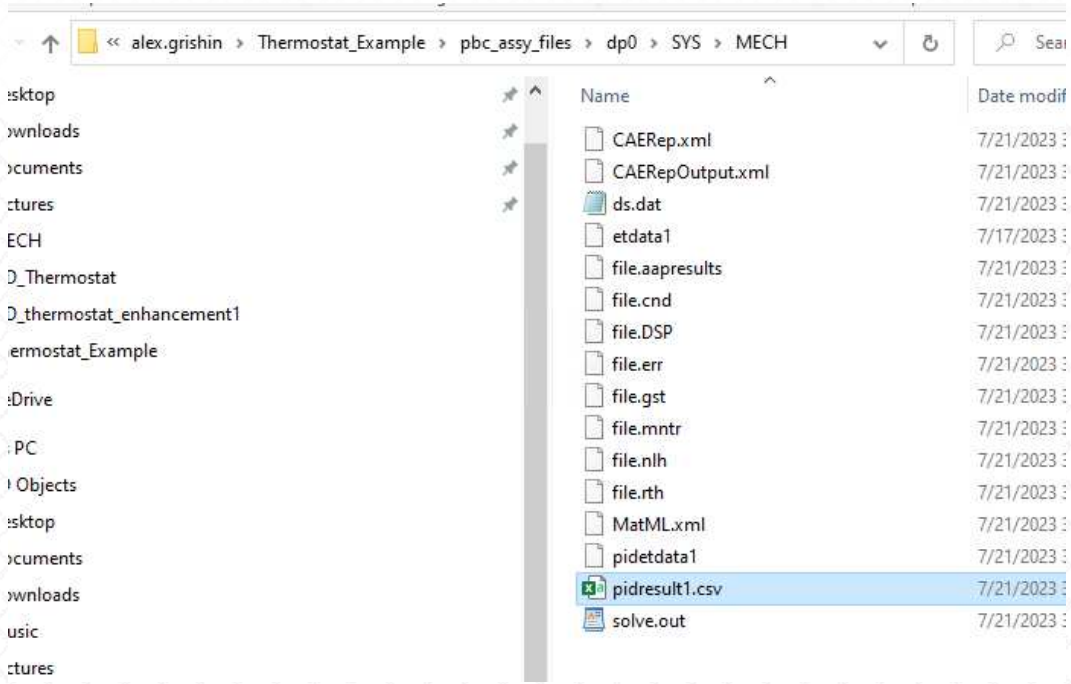


- The reaction probe yields a single result, while the controller result shows the sink and source power individually

The PID Controller: What It Does (Recap)

- To reconcile these two results, we have to look at the actual numerical values used to plot the source and sink values
- You can do this by either selecting “Export New?” and saving the results to a spreadsheet, or you can view the file used to produce these plots directly in the solution folder
- Let’s do the latter. Right-click on the ‘Solution’ object of the tree outline and select “Open Solver Files directory”
- Open the file ‘pidresult1.csv’

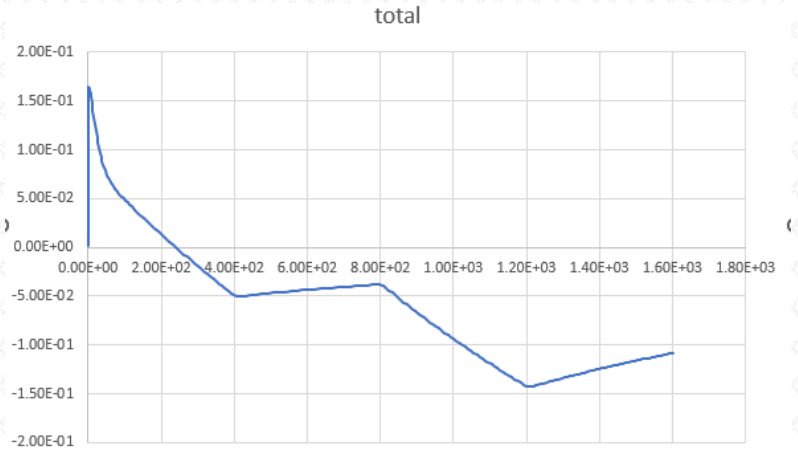


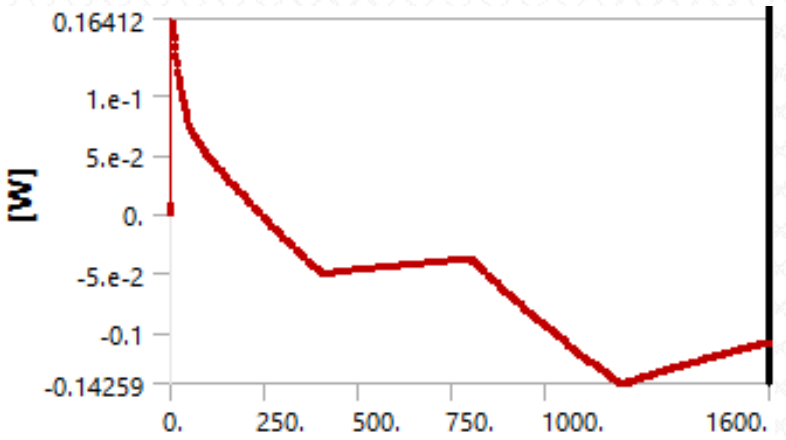


The PID Controller: What It Does (Recap)

- The source and sink power values plotted on slide 7 are highlighted below
- Adding these two columns together does indeed result in the net heat flow result of the reaction probe

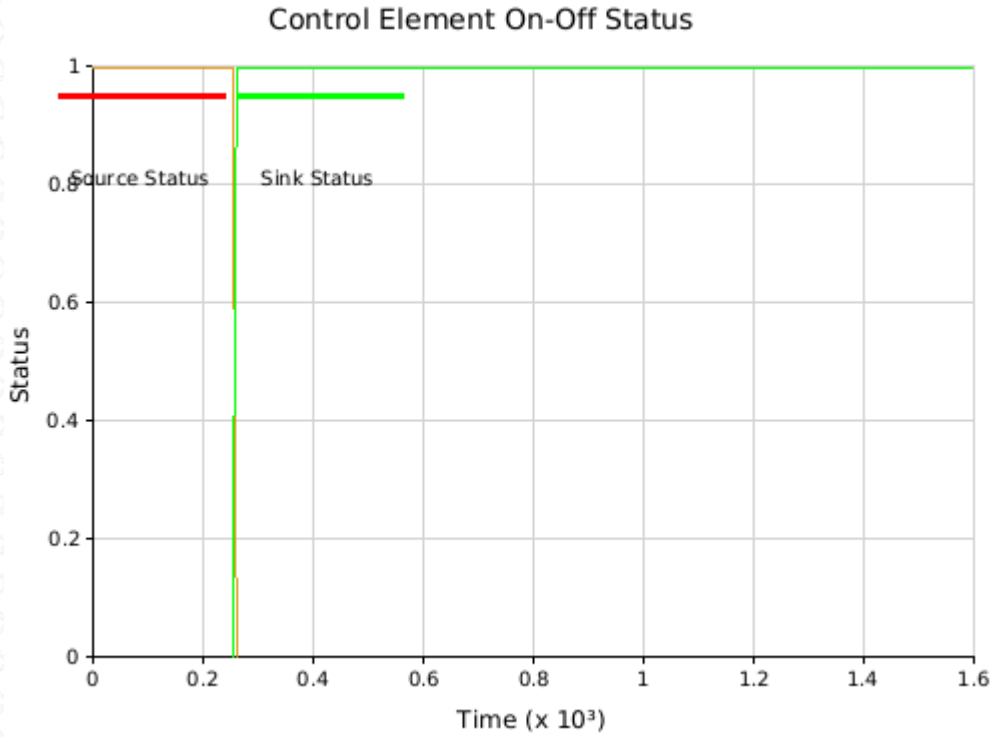
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
	time(s)	Setpoint T	Monitor T	Source Po	Source Pr	Source Int	Source De	source Co	Sink Powe	Sink Prop	Sink Integ	Sink Deriv	Sink Control	Status
2	1.00E-04	30	22	1.30E-03	32	0	0	1	-5.13E-06	0	0	0	0	
3	2.00E-04	30	22	2.59E-03	31.99999	0	0	1	-1.54E-05	0	0	0	0	
4	5.00E-04	30	22.00001	6.43E-03	31.99996	0	0	1	-9.08E-05	0	0	0	0	
5	1.40E-03	30	22.00007	1.76E-02	31.99972	0	0	1	-6.92E-04	0	0	0	0	
6	4.10E-03	30	22.00053	4.81E-02	31.99788	0	0	1	-5.27E-03	0	0	0	0	
7	6.80E-03	30	22.00122	7.64E-02	31.99512	0	0	1	-1.21E-02	0	0	0	0	
8	1.49E-02	30	22.00457	0.148384	31.98173	0	0	1	-4.51E-02	0	0	0	0	
9	3.92E-02	30	22.01855	0.323985	31.9258	0	0	1	-0.1807	0	0	0	0	
10	0.1121	30	22.06556	0.78347	31.73776	0	0	1	-0.62247	0	0	0	0	
11	0.3308	30	22.20669	2.003117	31.17326	0	0	1	-1.83917	0	0	0	0	
12	0.5495	30	22.34543	3.098962	30.61827	0	0	1	-2.93484	0	0	0	0	
13	0.9869	30	22.6132	4.877914	29.54719	0	0	1	-4.71385	0	0	0	0	
14	1.8617	30	23.11234	7.237745	27.55062	0	0	1	-7.07421	0	0	0	0	
15	2.7365	30	23.57778	8.733871	25.68887	0	0	1	-8.57112	0	0	0	0	
16	4.356198	30	24.33664	9.794435	22.65345	0	0	1	-9.63418	0	0	0	0	
17	5.753219	30	24.92311	9.95188	20.30757	0	0	1	-9.79421	0	0	0	0	
18	7.150239	30	25.44853	9.670326	18.20589	0	0	1	-9.51565	0	0	0	0	
19	9.94428	30	26.30198	8.500252	14.79208	0	0	1	-8.3524	0	0	0	0	
20	12.73832	30	26.99451	7.213063	12.02196	0	0	1	-7.07252	0	0	0	0	
21	15.53236	30	27.55651	6.010335	9.773973	0	0	1	-5.87711	0	0	0	0	
22	18.3264	30	28.01262	4.960055	7.949513	0	0	1	-4.83394	0	0	0	0	
23	21.12044	30	28.38284	4.072726	6.468633	0	0	1	-3.95335	0	0	0	0	
24	23.91449	30	28.68338	3.335992	5.2665	0	0	1	-3.22291	0	0	0	0	
25	26.70853	30	28.92737	2.730064	4.290533	0	0	1	-2.6228	0	0	0	0	





The PID Controller: What It Does (Recap)

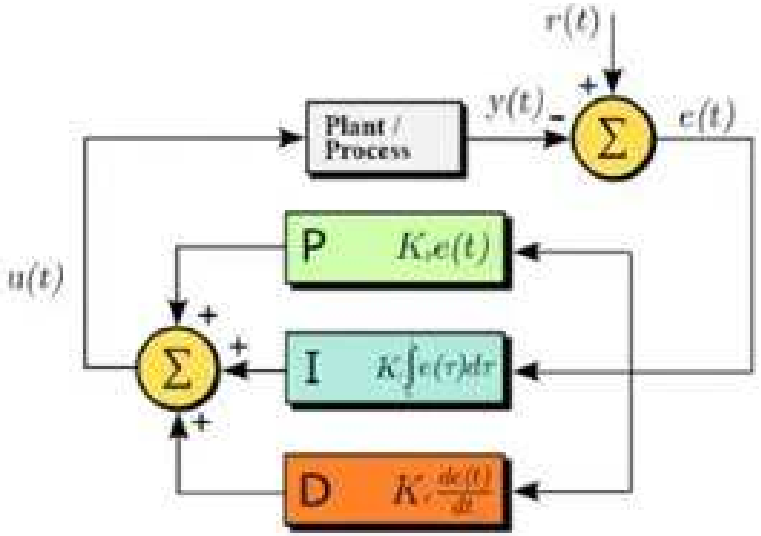
- When the ‘Control Type’ is set to ‘Both Heat Source and Sink’, we can both add and remove heat
- You can see this happening by noting the status of the ‘source’ and ‘sink’ (the graph in the upper right-hand corner –’Control Element On-Off Status’). For the first 255 seconds or so, we have to add heat to the TTL component for it reach the required 30°C. After that, however, we’re removing heat (the ‘sink’ control turns ‘on’, while the ‘source’ turns ‘off’)

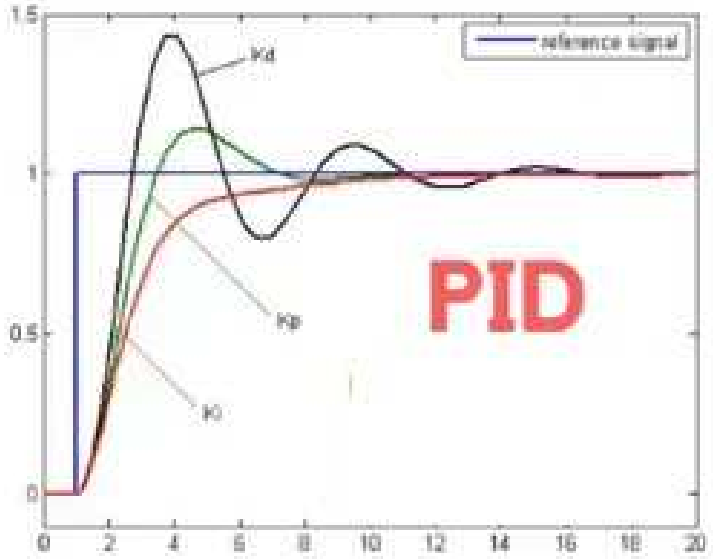


A	B	C	D	E	F	G	H	I	J	K	L	M	N
134.9119	30	29.98914	3.87E-02	4.34E-02	0	0	1	-2.87E-03	0	0	0	0	
142.9119	30	29.98977	3.52E-02	4.09E-02	0	0	1	-2.14E-03	0	0	0	0	
150.9119	30	29.99042	3.22E-02	3.83E-02	0	0	1	-2.00E-03	0	0	0	0	
158.9119	30	29.99108	2.94E-02	3.57E-02	0	0	1	-1.99E-03	0	0	0	0	
166.9119	30	29.99259	2.87E-02	2.96E-02	0	0	1	-3.97E-03	0	0	0	0	
174.9119	30	29.99307	2.40E-02	2.77E-02	0	0	1	-2.05E-03	0	0	0	0	
182.9119	30	29.9936	2.09E-02	2.56E-02	0	0	1	-1.70E-03	0	0	0	0	
190.9119	30	29.99416	1.81E-02	2.34E-02	0	0	1	-1.70E-03	0	0	0	0	
198.9119	30	29.99474	1.55E-02	2.10E-02	0	0	1	-1.76E-03	0	0	0	0	
206.9119	30	29.99535	1.29E-02	1.86E-02	0	0	1	-1.82E-03	0	0	0	0	
214.9119	30	29.99597	1.02E-02	1.61E-02	0	0	1	-1.86E-03	0	0	0	0	
222.9119	30	29.99659	7.61E-03	1.36E-02	0	0	1	-1.90E-03	0	0	0	0	
230.9119	30	29.99723	4.99E-03	1.11E-02	0	0	1	-1.92E-03	0	0	0	0	
238.9119	30	29.99787	2.37E-03	8.51E-03	0	0	1	-1.94E-03	0	0	0	0	
246.9119	30	29.99852	-2.34E-04	5.94E-03	0	0	1	-1.95E-03	0	0	0	0	
254.9119	30	29.99944	-2.18E-03	2.26E-03	0	0	1	-2.59E-03	0	0	0	0	
262.9119	30	30.00115	-2.76E-03	0	0	0	0	-4.56E-03	-4.60E-03	0	0	1	
270.9119	30	30.00215	-2.97E-03	0	0	0	0	-6.93E-03	-8.62E-03	0	0	1	
278.9119	30	30.00292	-2.46E-03	0	0	0	0	-1.00E-02	-1.17E-02	0	0	1	
286.9119	30	30.00359	-2.15E-03	0	0	0	0	-1.29E-02	-1.44E-02	0	0	1	
294.9119	30	30.00424	-2.01E-03	0	0	0	0	-1.55E-02	-1.70E-02	0	0	1	
302.9119	30	30.00487	-1.94E-03	0	0	0	0	-1.81E-02	-1.95E-02	0	0	1	
310.9119	30	30.0055	-1.91E-03	0	0	0	0	-2.07E-02	-2.20E-02	0	0	1	
318.9119	30	30.00612	-1.90E-03	0	0	0	0	-2.32E-02	-2.45E-02	0	0	1	
326.9119	30	30.00674	-1.88E-03	0	0	0	0	-2.56E-02	-2.70E-02	0	0	1	

- columns h and m list the source and sink control status, respectively. While one is ‘on’, the other is ‘off’

New: PID Controller: Release 1.0





PID Controller: New Enhancemets (Release 1.0)

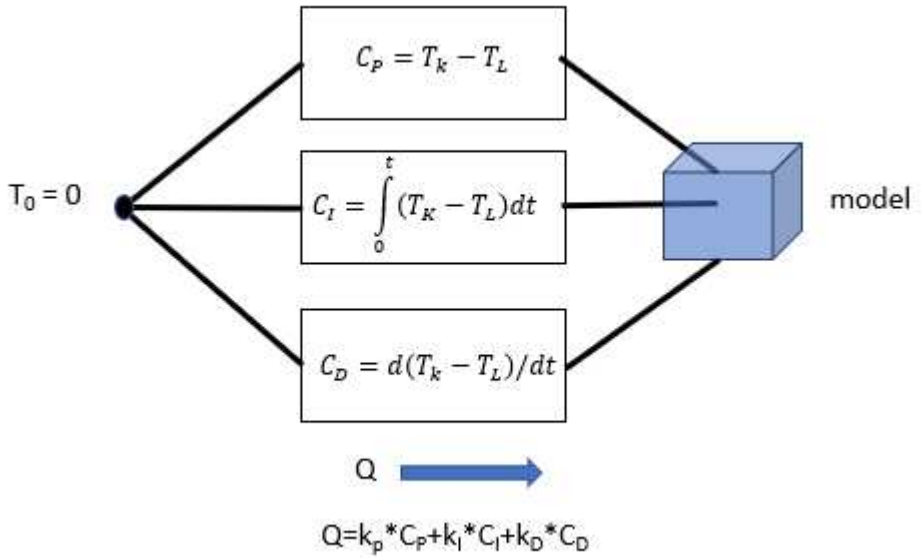
- In this release of the PID controller, we've added a 'Limit Load' –or a power threshhold to more accurately simulate real control mechanisms (which can't support unlimited heat flows).

Details of "PID Controller"

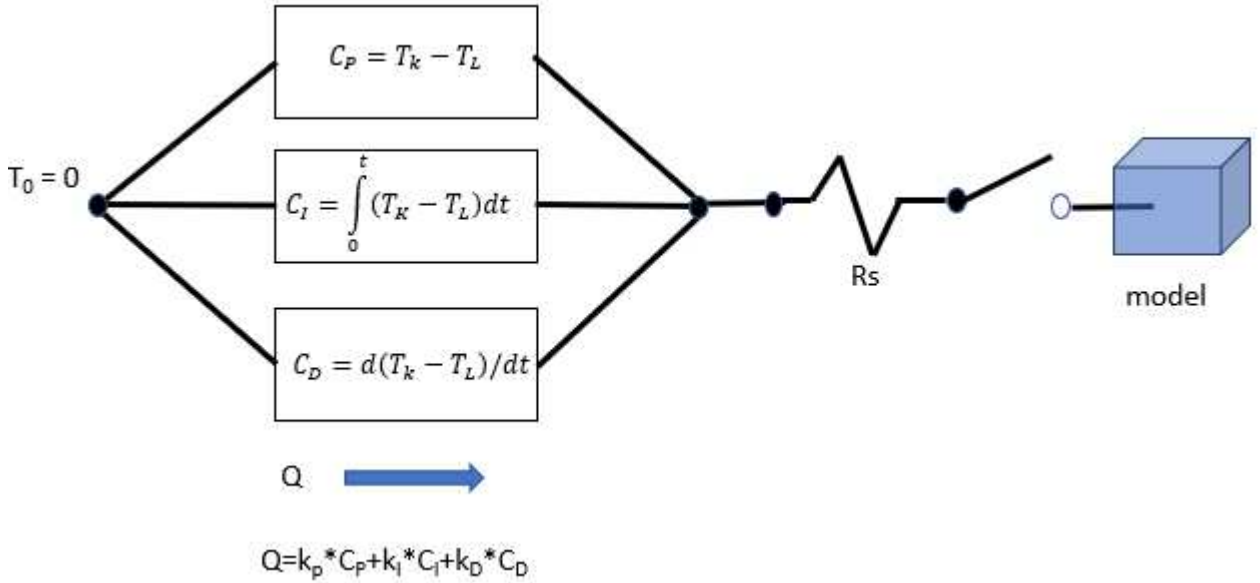
Control Type	Both Heat Source and Sink
Proportional Gain	4 [W C^-1]
Integral Gain	0 [W sec^-1 C^-1]
Derivative Gain	0 [W sec C^-1]
Controller Setpoint Properties	
Setpoint Type	User Specified Setpoint
Controller Set Point Temperature	Tabular Data
Offset	0 [C]
Controller Limit Load	
Limit Load	No Limit Load
No Limit	
Controller Results	
View Results?	No
Export Now?	No

PID Controller: New Enhancemets (Release 1.0)

- This has been implemented in ANSYS using a combin39 spring with a limit load equal to the user-selected power threshhold value
- This is shown graphically below



- Previous versions of the PID controller



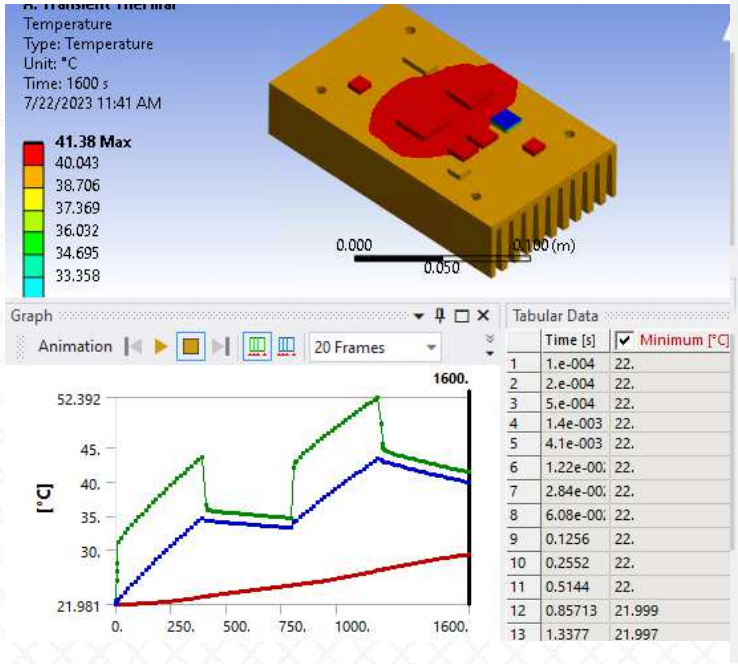
- Release 1.0

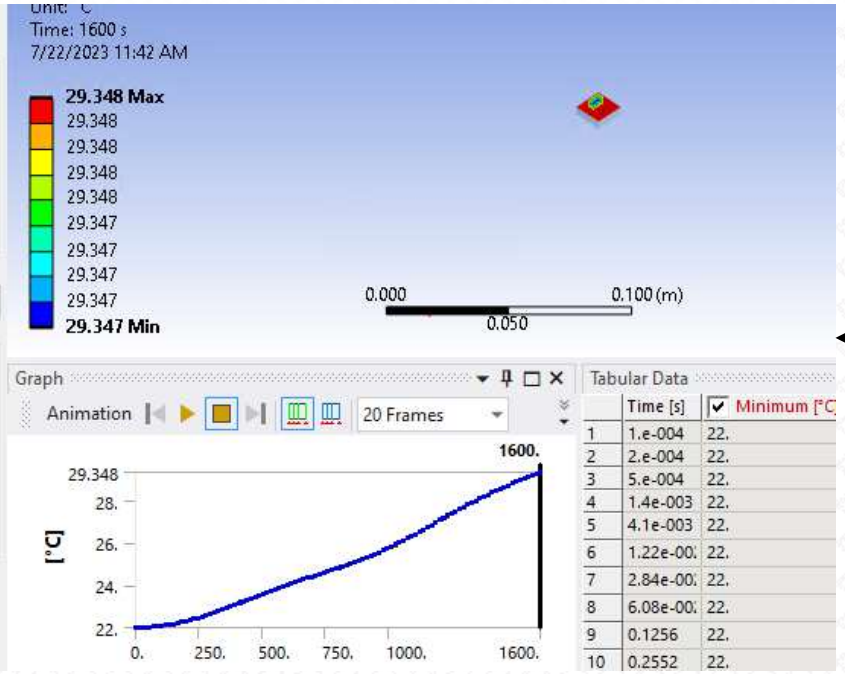
PID Controller: New Enhancemets (Release 1.0)

- To demonstrate the controller behavior with the addition of the combin39 element, lets change the ‘Control Type’ to ‘Heat Sink’ to see the difference that makes after 1600 s
- Before doing so, be sure to close any Excel session in wich the previous result was open (otherwise the controller results won’t update properly)
- Do not specify a ‘limit load’ yet. Even without this feature, we’ll see some peculuarities of the new release
- Run the model with the changed controller setting.
- Note that, with a single proportional gain, the controller in ‘heat sink’ mode takes a lot longer to reach the target temperature. It’s not doing a very good job. In a moment, we’ll see why...

Details of "PID Controller"

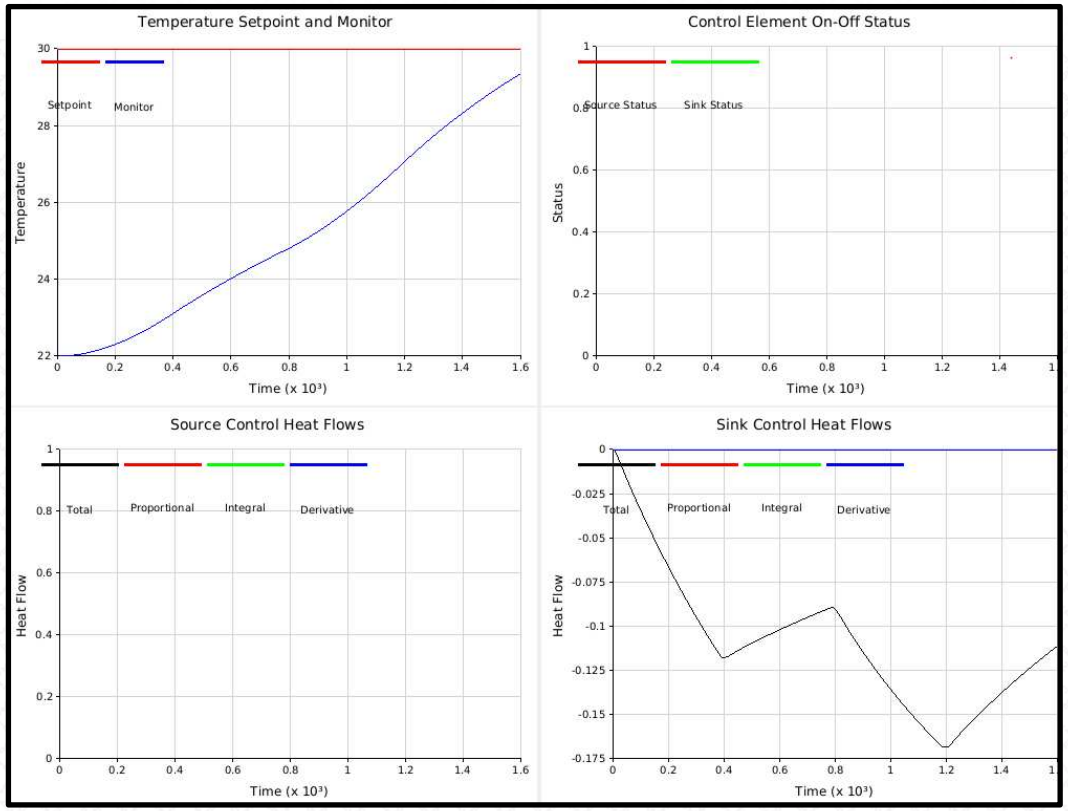
Heat Source/Sink Location	
Scoping Method	Geometry Selection
Geometry	1 Face
Temperature Monitor Location	
Scoping Method	Geometry Selection
Geometry	1 Vertex
Controller Control Properties	
Control Type	Heat Sink
Proportional Gain	4 [W C^-1]
Integral Gain	0 [W sec^-1 C^-1]
Derivative Gain	0 [W sec C^-1]
Controller Setpoint Properties	
Setpoint Type	User Specified Setpoint
Controller Set Point Temperature	Tabular Data
Offset	0 [C]

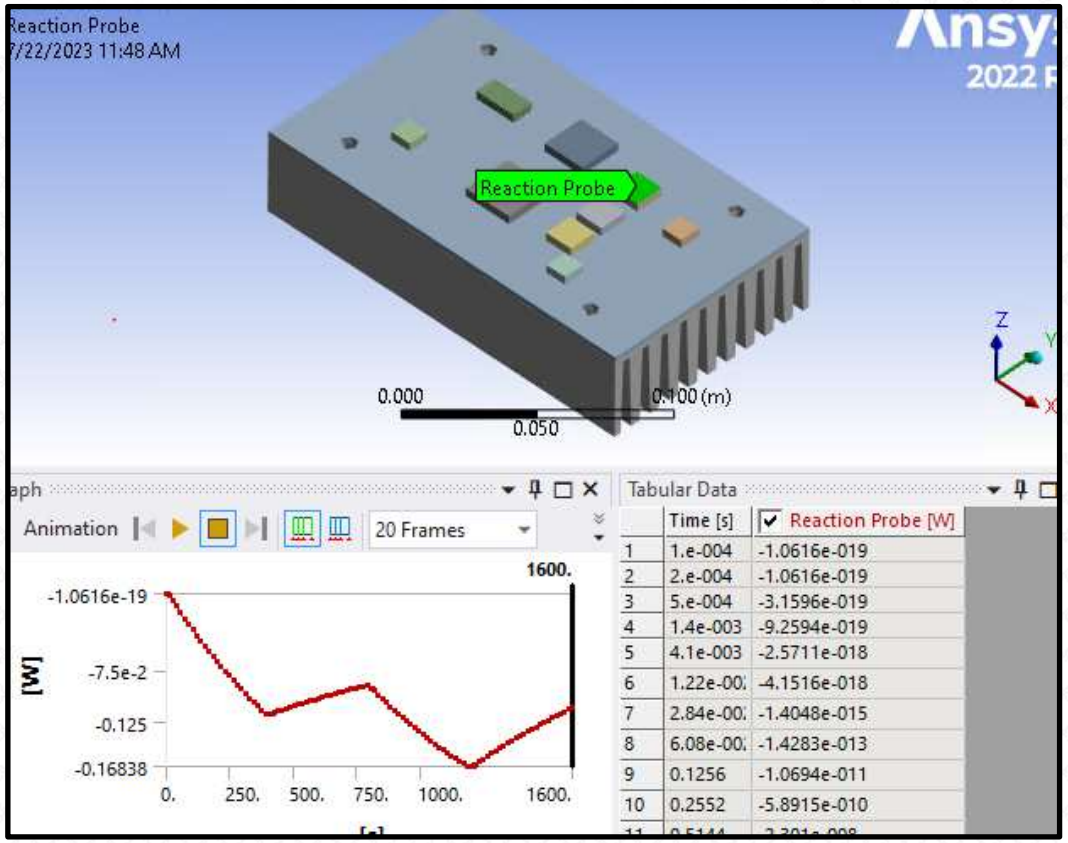




PID Controller: New Enhancemets (Release 1.0)

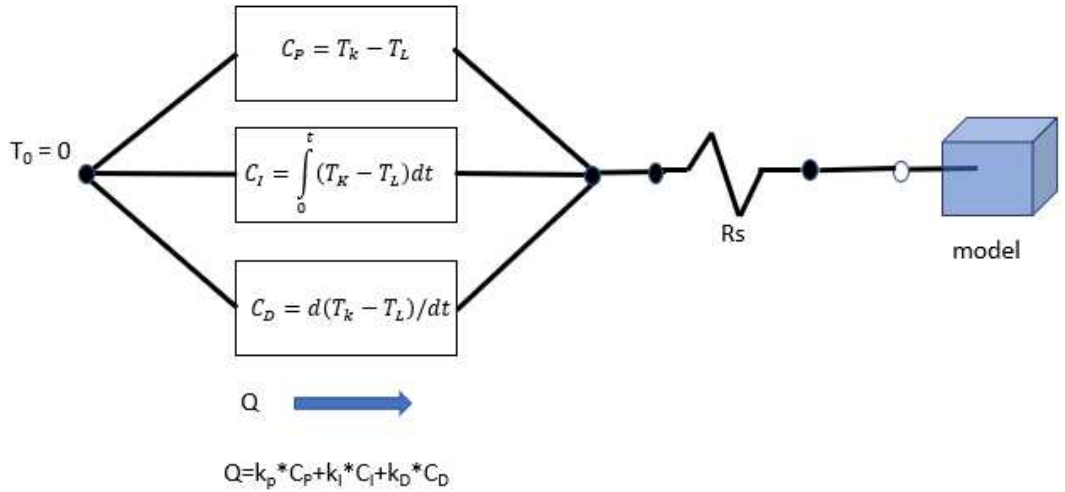
- Because we only have a single controller (in ‘sink’ mode), the net heat flow reported by the controller (“View Results?” set to “Yes”) and the reaction probe agree
- However, take a look at the ‘Control Element On-Off Status (the graph in the upper right-hand corner. The sink never turns ‘On’!
- How can this be? How can we get a cooling effect without turning on the controller?

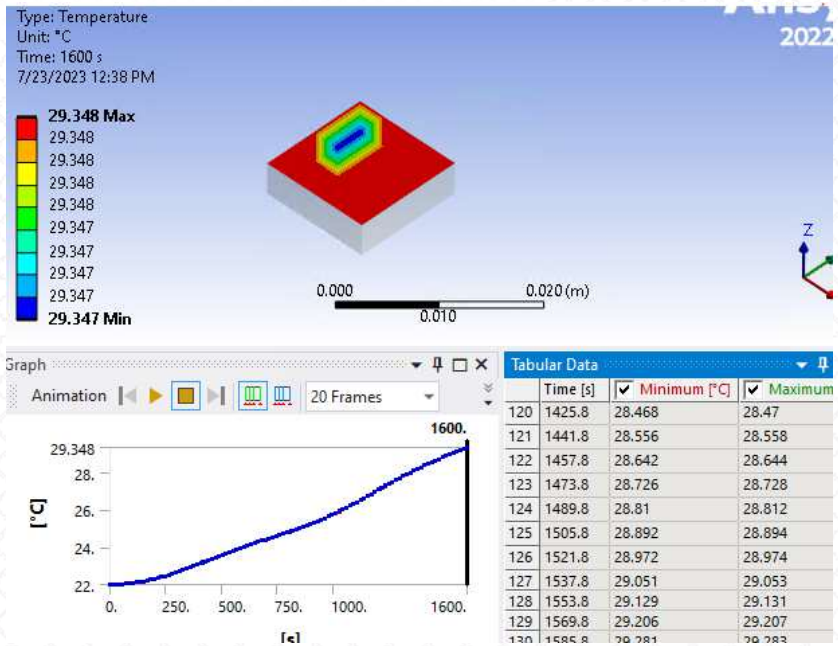




PID Controller: New Enhancemets (Release 1.0)

- This is happening simply due to the addition of the combin39 spring element
- Without specifying a limit load, the effective limit switch between the controller and the model is closed
- But the combin39 spring has a hard-coded thermal conductance of 1000 W/°C (Resistance, Rs = 0.001 °C/W)
- This has the same effect as covering the target TTL component with a conductive material (which causes a thermal short-circuit between the model load and the heat sink)
- And this allows the TTL component to from it's initial temperature to the target temperature at a much slower rate!

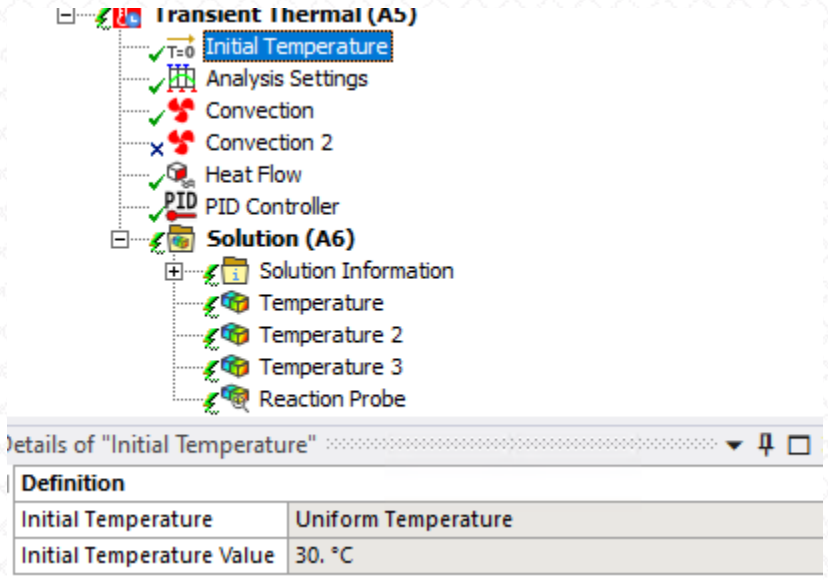


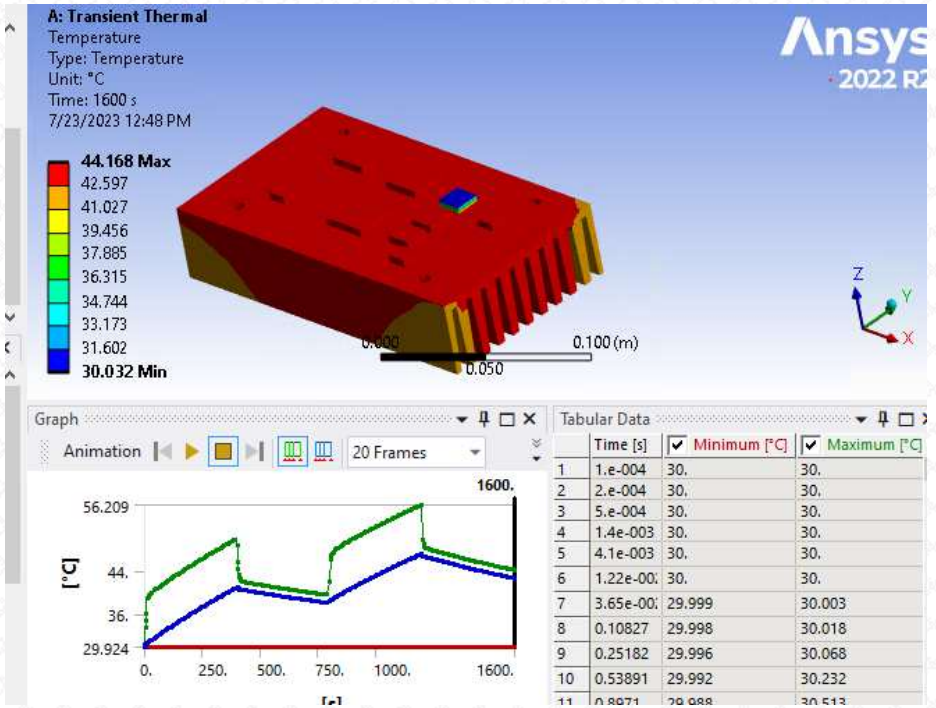


PID Controller: New Enhancemets (Release 1.0)

- Readers may think that, because of this, it might make more sense to use a very low conductance value (high resistance) for the combin39 spring. But this would cause other problems (future versions may allow users to specfiy the resistance value themselves)

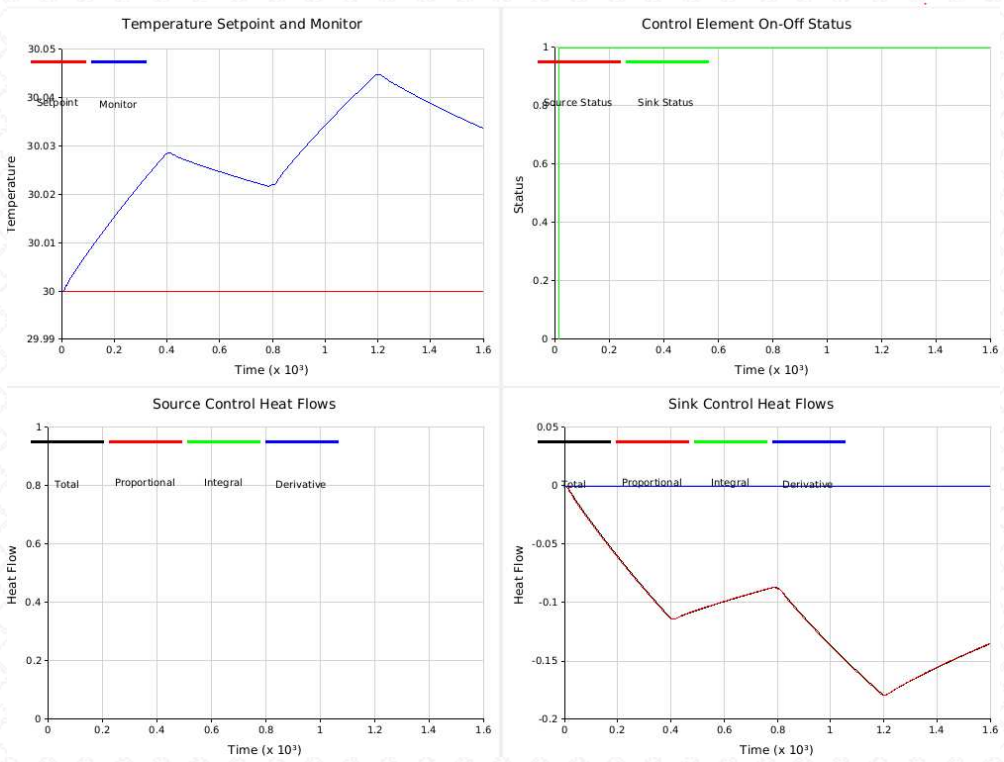
- Go back and set the initial temperature to be equal to the target temperature (30°C) and re-run the model to see the difference this makes

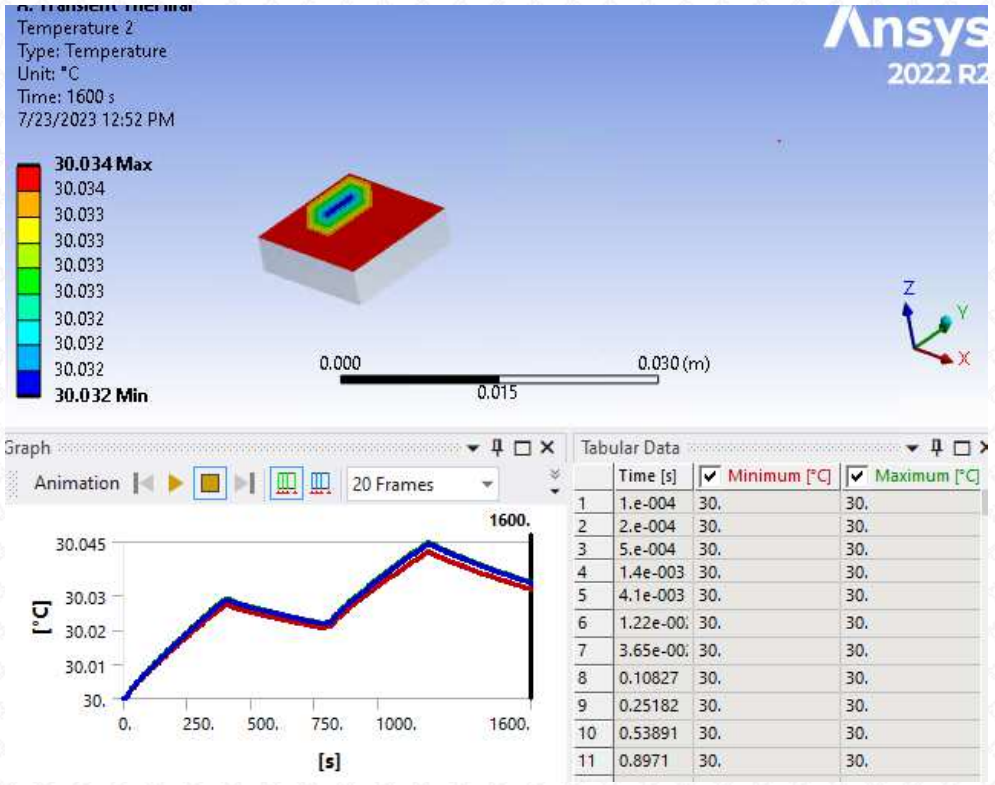




PID Controller: New Enhancemets (Release 1.0)

- This time, the controller turns ‘on’ immediately and stays on
- This is because there was no initial temperature difference which could cause a thermal short circuit –the only way to keep the TTL component cool is to actively extract heat



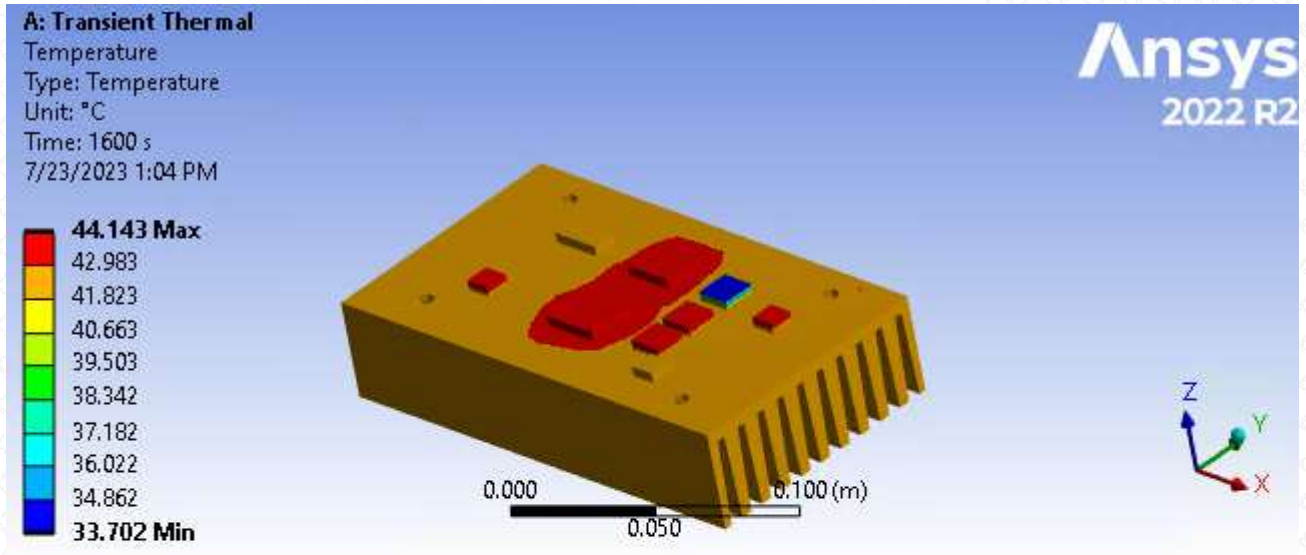


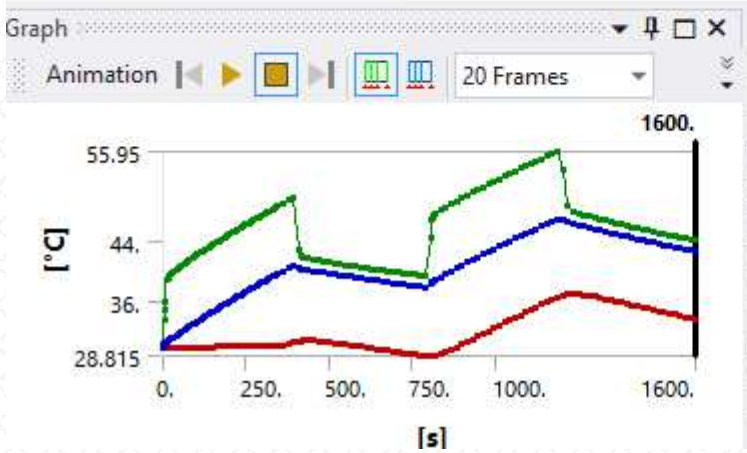
PID Controller: New Enhancemets (Release 1.0)

- Ok. Now, suppose we only have a 0.1 W power available. We can simulate this situation by applying this values as a ‘limit load’ as below
- Re-run and view the results

Details of "PID Controller"

Control Type	Heat Sink
Proportional Gain	4 [W C^-1]
Integral Gain	0 [W sec^-1 C^-1]
Derivative Gain	0 [W sec C^-1]
Controller Setpoint Properties	
Setpoint Type	User Specified Setpoint
Controller Set Point Temperature	Tabular Data
Offset	0 [C]
Controller Limit Load	
Limit Load	Set Limit Load
Limit Load	0.1 [W]
Controller Results	
View Results?	No
Export Now?	No



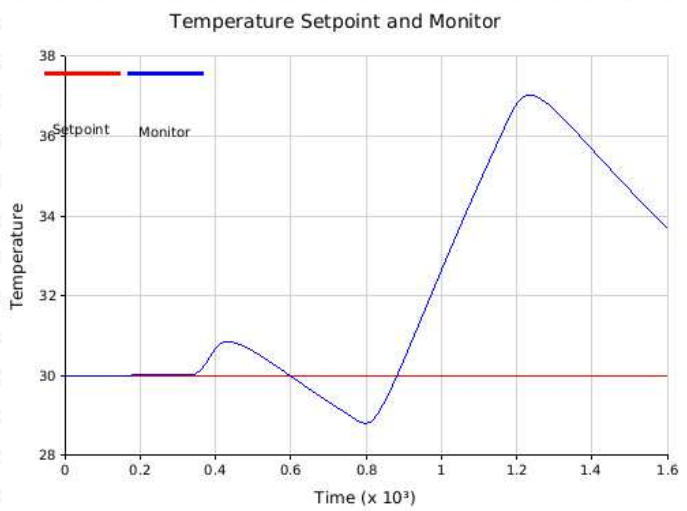


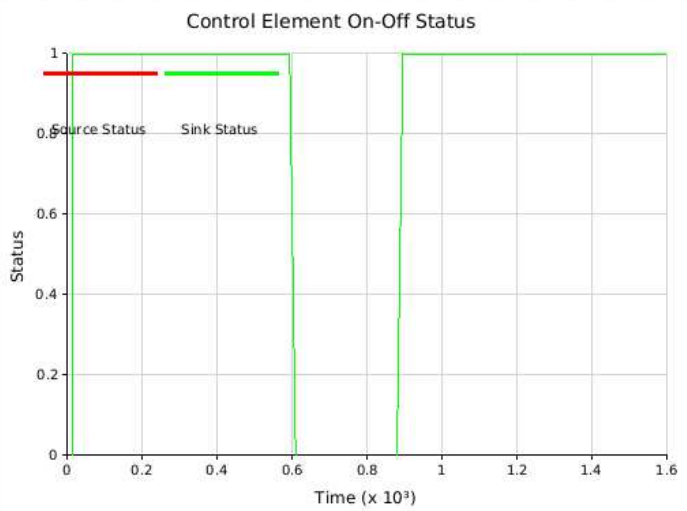
Tabular Data

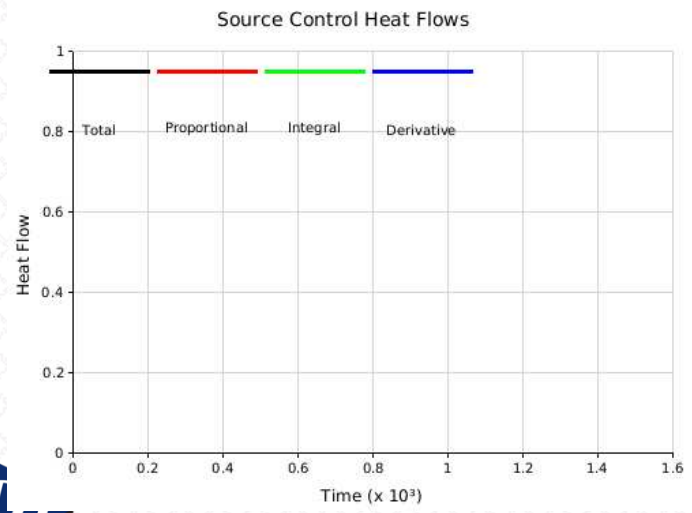
	Time [s]	Minimum [°C]	Maximum [°C]
1	1.e-004	30.	30.
2	2.e-004	30.	30.
3	5.e-004	30.	30.
4	1.4e-003	30.	30.
5	4.1e-003	30.	30.
6	1.22e-00:	30.	30.
7	3.65e-00:	29.999	30.003
8	0.10827	29.998	30.018
9	0.25182	29.996	30.068
10	0.53891	29.992	30.232
11	0.8971	29.988	30.513

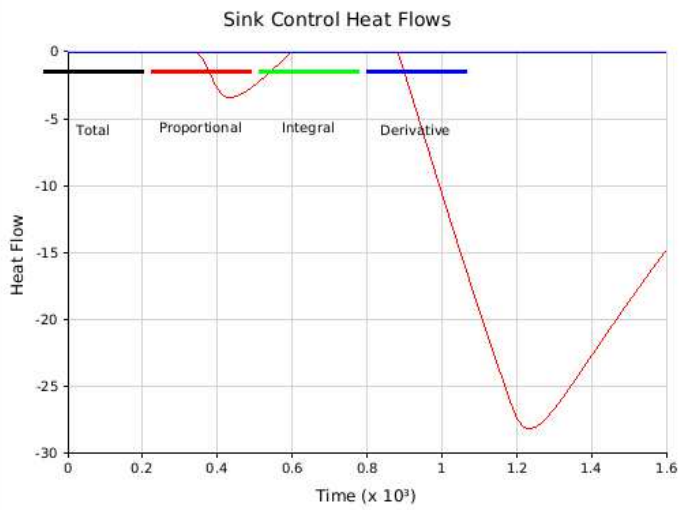
PID Controller: New Enhancement

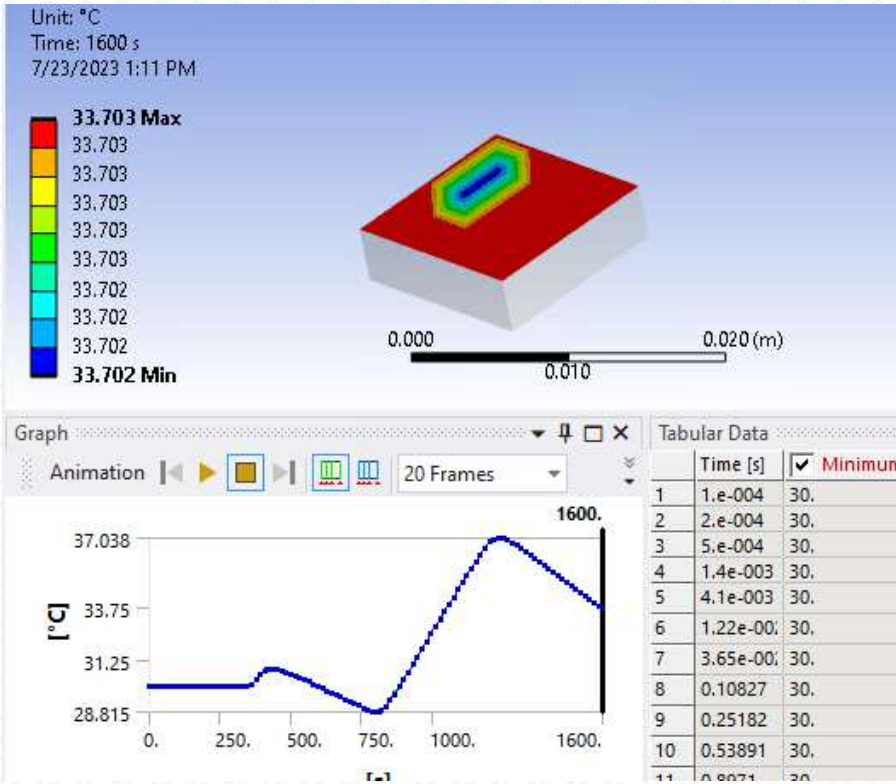
- The limited power results in an inability to maintain the target temperature...







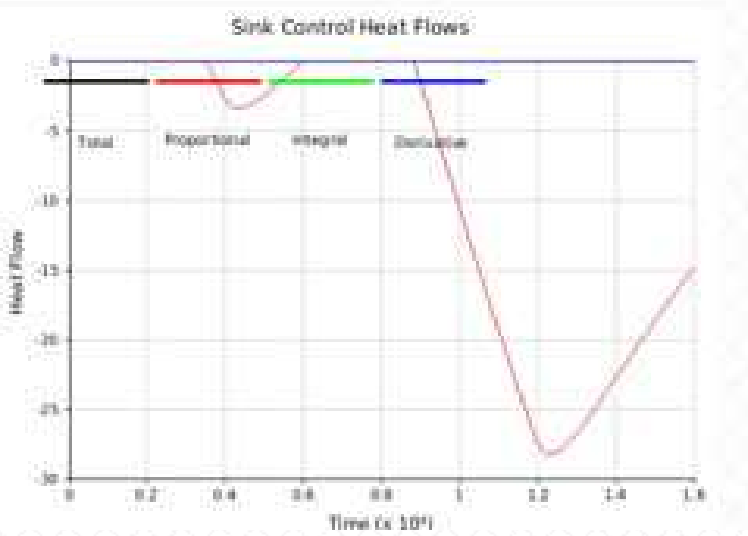


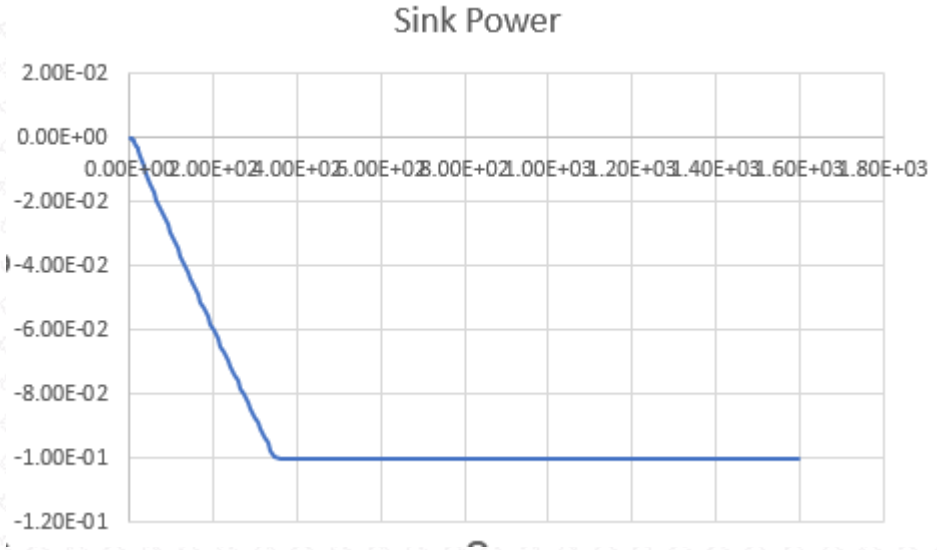


PID Controller: New Enhancement

- Note also the difference between the proportinal gain component and the net power

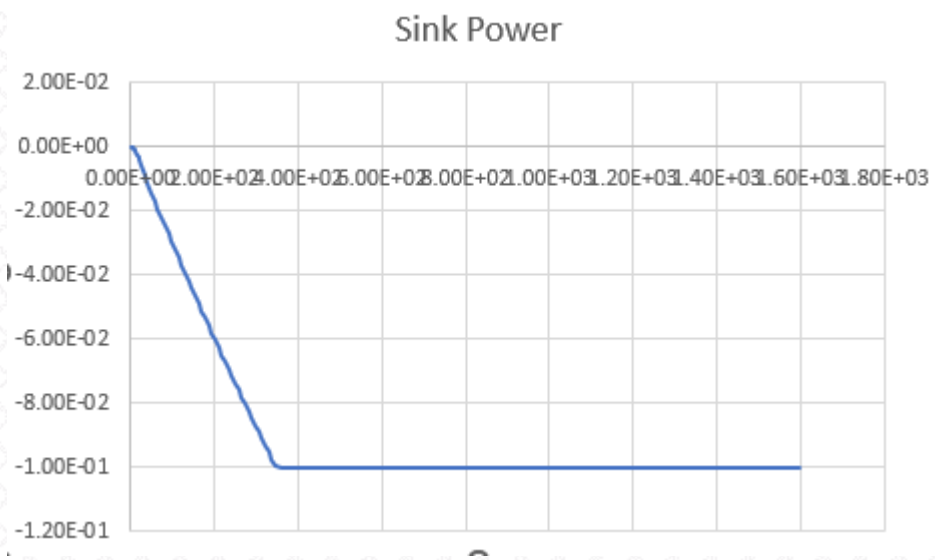
- The proportional gain component dissipates almost -30 W max (!), the ‘total’ net power is much less. To view it, open the file ‘pidresult1.csv’
- The Total sink power is in column i, while the sink proportional powr is in column j
- Plot the total sink power (column i)...

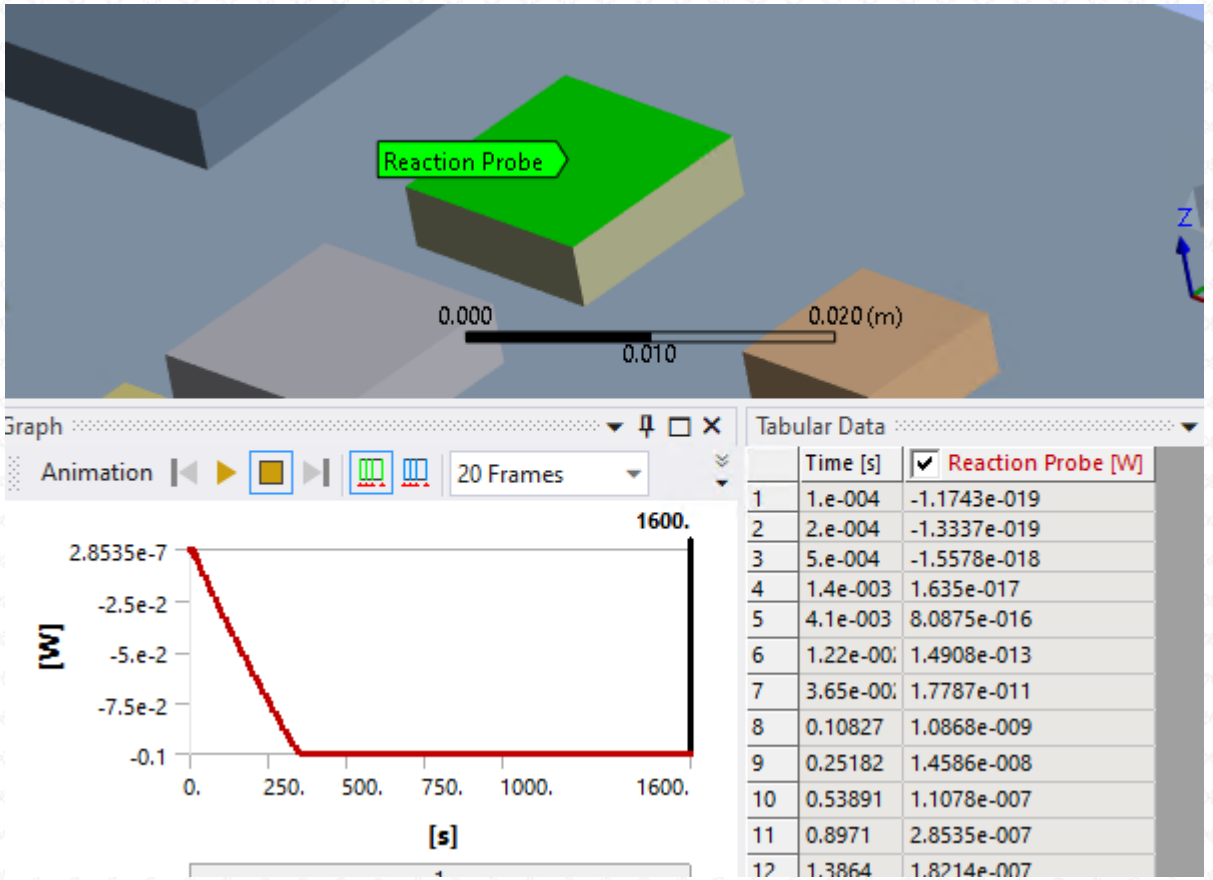




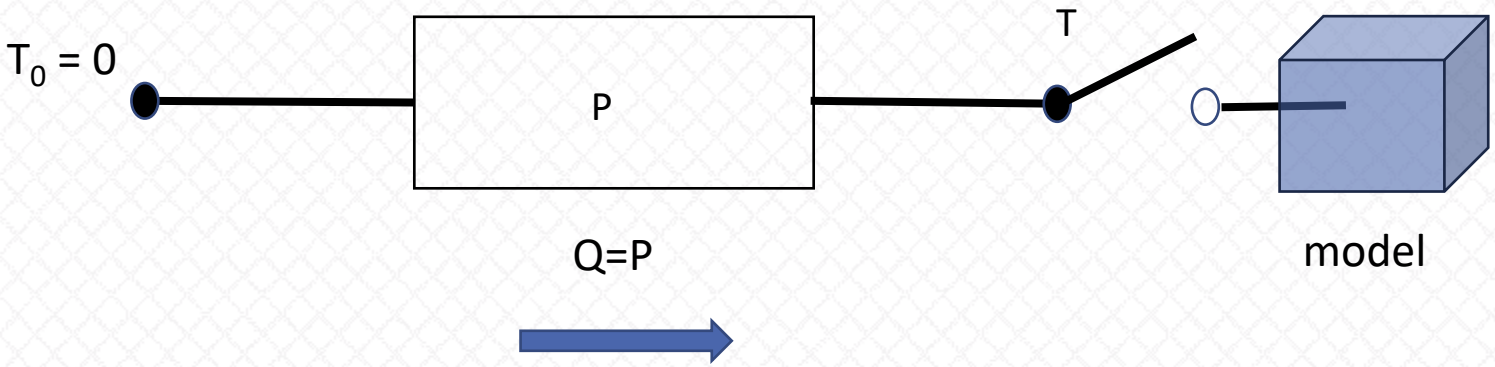
PID Controller: New Enhancement

- Now verify that this equal the power dissipation measured by the reaction probe





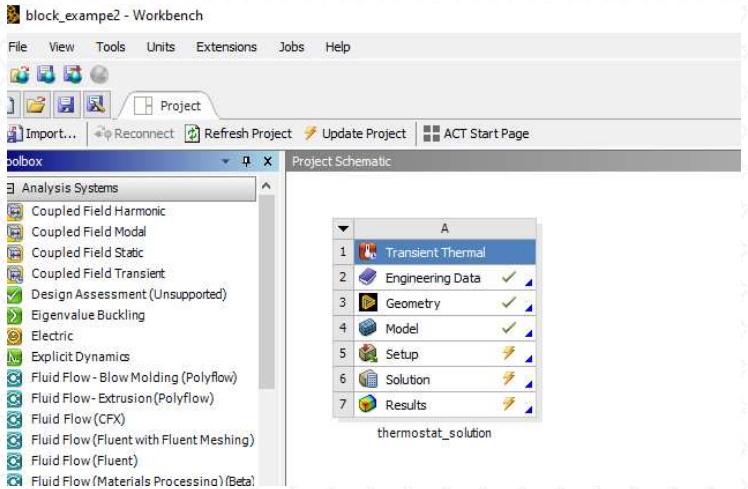
New:Thermostat: Alpha Release



Thermostat: Alpha Release

- As we mentioned in the introduction, thermostats and PID controllers function differently (and are used to solve different problems, in general)
- Thermostats are usually connected to a constant (known) power source
- They have a limit switch, which is typically tripped by a set temperature (instead of power)
- Before introducing the new thermostat ACT extension, which again makes use of the powerful combin37 control element, we'll demostrate the basic features of a thermostat with an APDL solution script ('thermostat.inp' –included with this post)

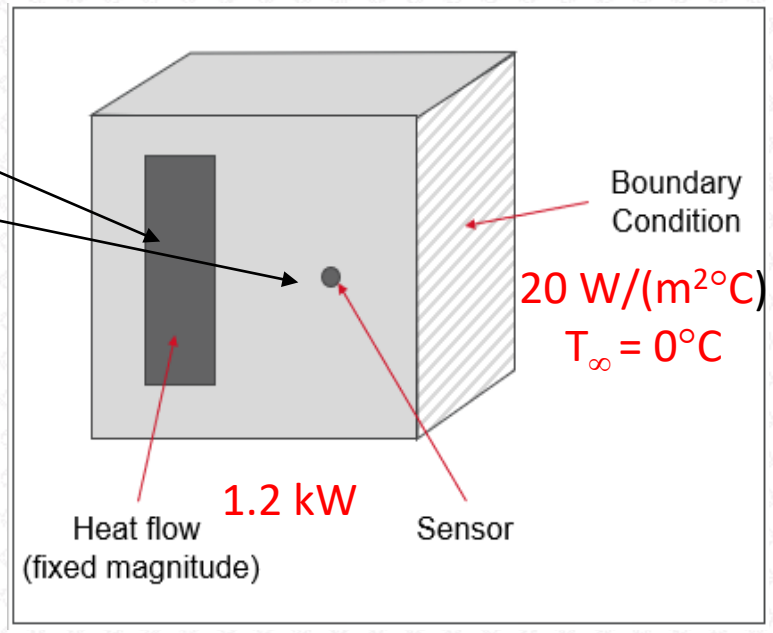
- Open the Workbench archive "block_example.wbpz" (also included with this post).



Thermostat: Alpha Release

- This model has the following properties

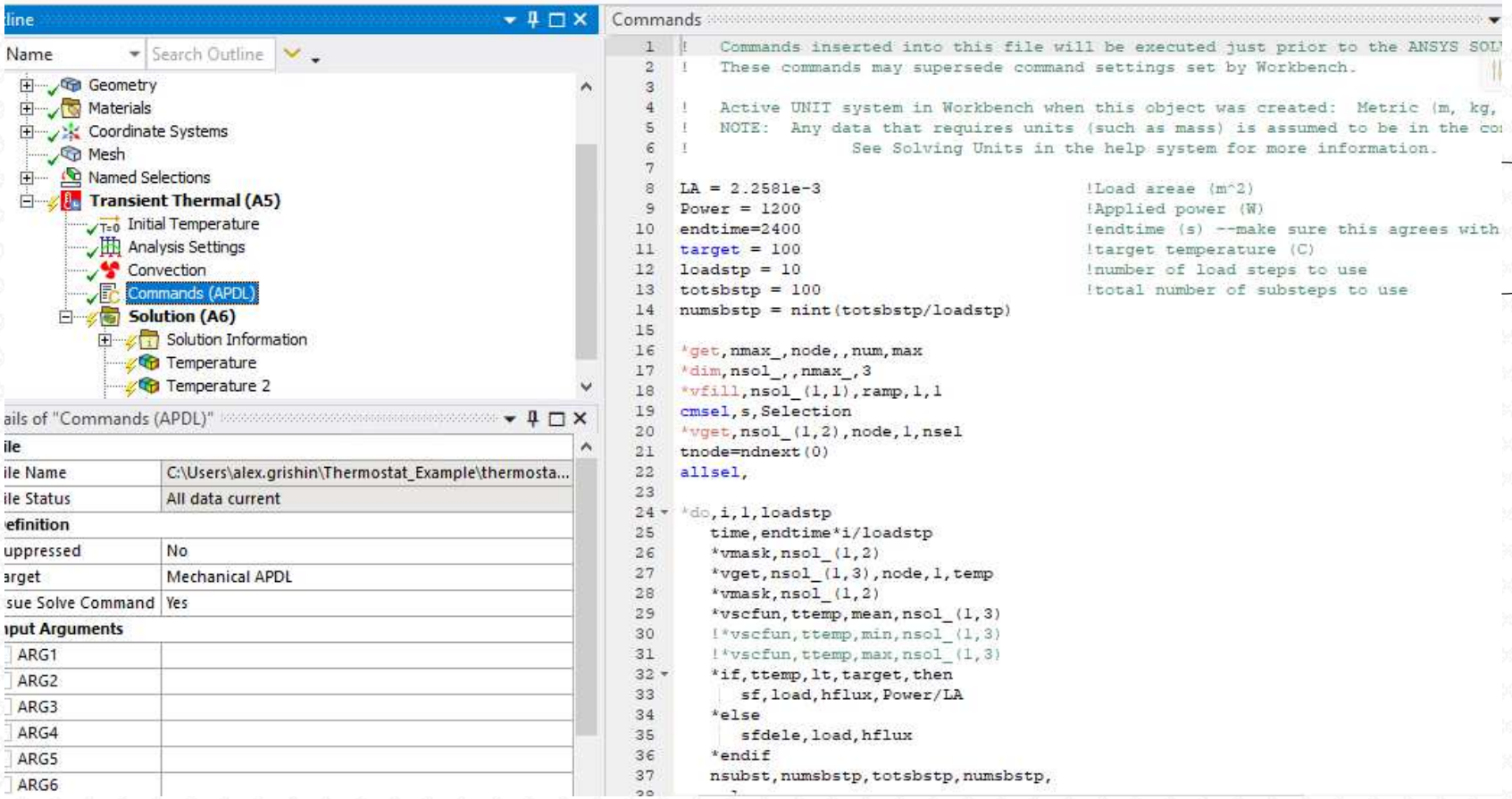
- Scope Geometry for Heat Source/Sink Location to ‘Face’ and select the region shown
- Scope Geometry for Temperature Monitor Location to Vertex and select the point shown
- Specify Off Set Point to be 100°C
- Specify Heat to 1200 W
- Specify Mode to be ‘Heating’ (the default)



- In this example, we want to heat the block by adding heat to the source location (the grey rectangle) and monitoring at the sensor location shown
- The thermostat will turn on whenever the sensor location < 100°C
- Check the monitor temperature after each run by creating a ‘Temperature’ Result scoped to just the monitor point vertex

Thermostat: Alpha Release

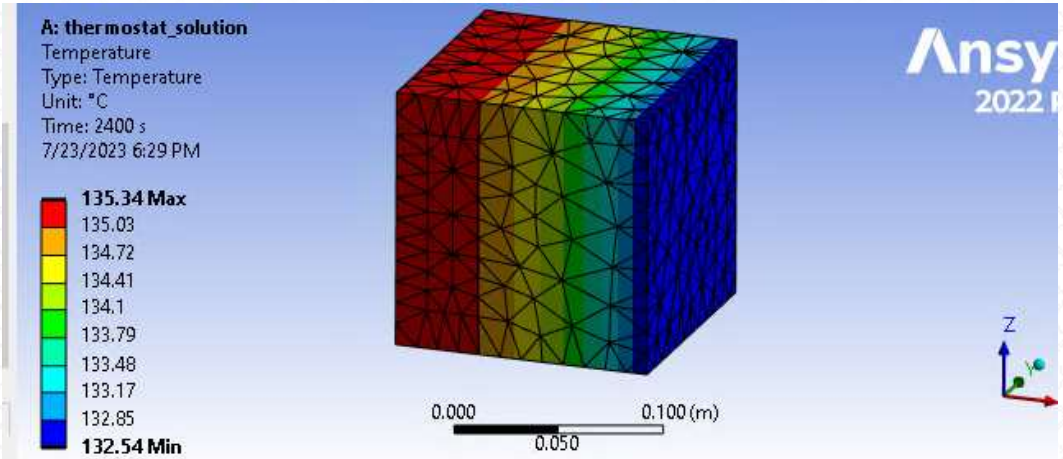
- Right-click on the ‘Transient Thermal’ environment and insert a command object
- Import the file ‘thermostat.inp’

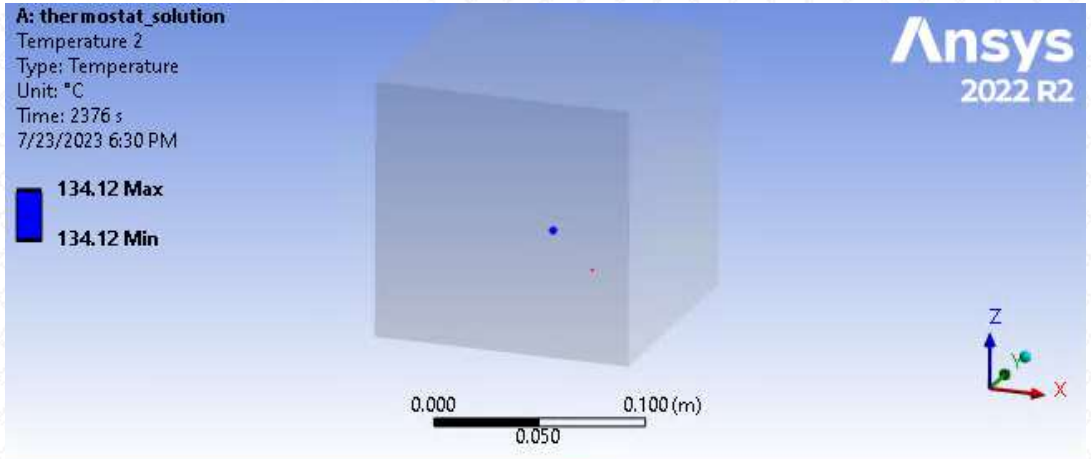


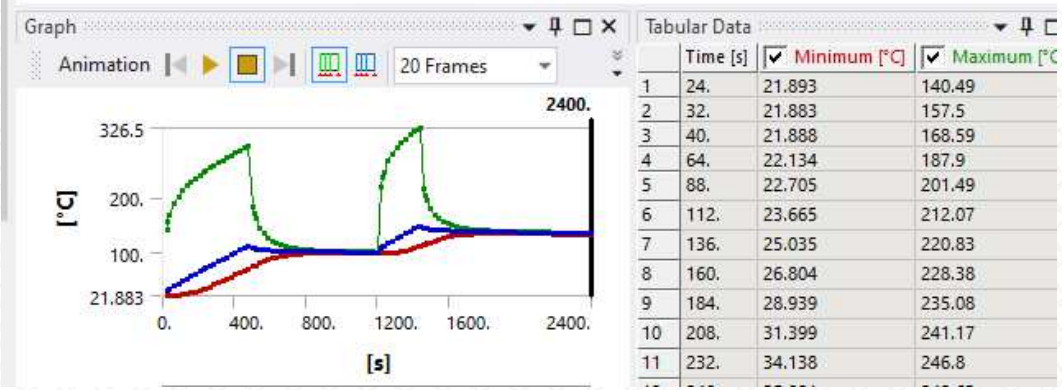
Double-check that the settings agree with those of the previous slide

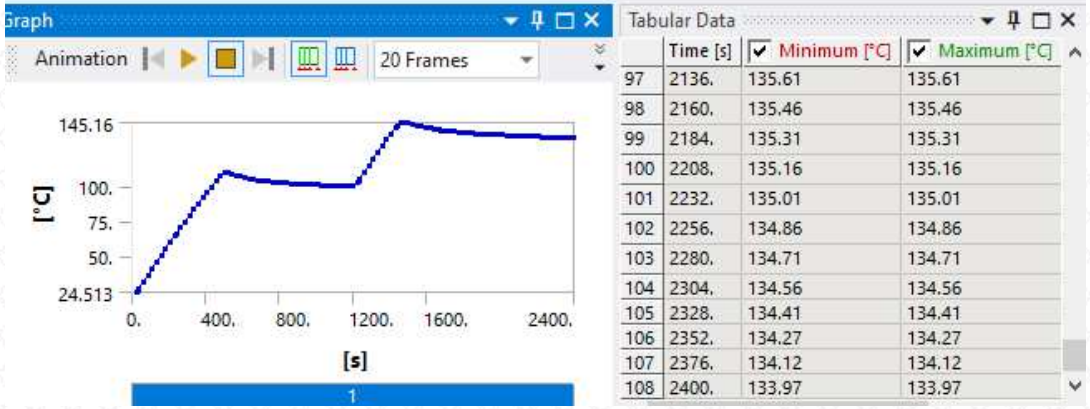
Thermostat: Alpha Release

- run the model
- Check the target temperature. Remember, the target is 100°C. We're not very close. Let's explore why...









Thermostat: Alpha Release

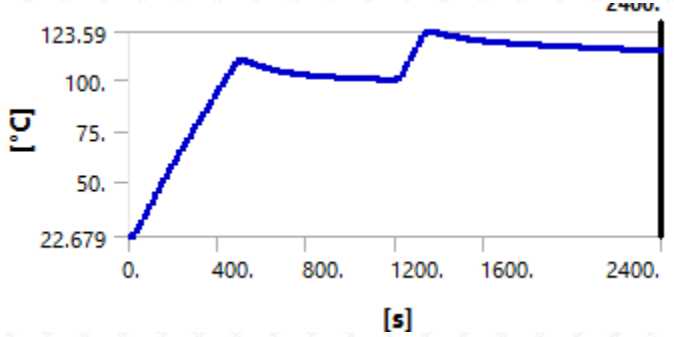
- As we mentioned, this algorithm evaluates the temperature at the monitor location within a loop increment
- It can only do this once per increment. The accuracy of this technique is therefore directly connected to how many loop increments we define
- This is a basic limitation of all APDL solution algorithms: actions can only be taken within user-defined loop increments
- Double the number of load steps (line 12: set 'loadstep = 20') and re-run...

```
 8  LA = 2.2581e-3                          !Load areae (m^2)
 9  Power = 1200                            !Applied power (W)
10  endtime=2400                            !endtime (s) --make sure this agrees with WB
11  target = 100                            !target temperature (C)
12  loadstp = 20                            !number of load steps to use
13  totsbstp = 10*loadstp                   !total number of substeps to use
14  numsbstp = nint(totsbstp/loadstp)
```

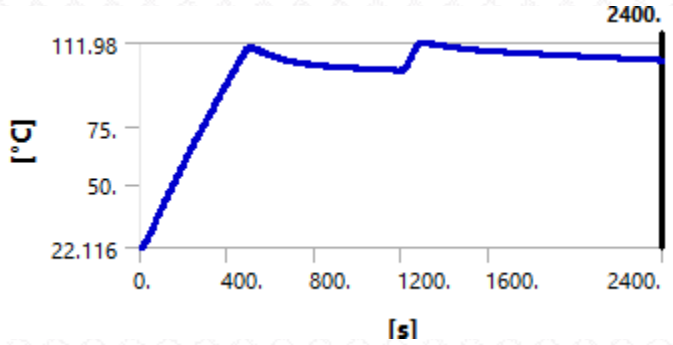
```
24  *do,i,1,loadstp
25    time,endtime*i/loadstp
26    *vmask,nsol_(1,2)
27    *vget,nsol_(1,3),node,1,temp      !get temperature
28    *vmask,nsol_(1,2)
29    !*vscfun,ttemp,mean,nsol_(1,3)
30    !*vscfun,ttemp,min,nsol_(1,3)
31    *vscfun,ttemp,max,nsol_(1,3)      !get maximum value
32    *if,ttemp,lt,target,then          !if temperature < target then apply load
33       sf,load,hflux,Power/LA
34    *else                             !...otherwise, delete load
35       sfdele,load,hflux
36    *endif
37    nsubst,numsbstp,totsbstp,numsbstp,
38    solve
39  *ENDDO
```

Thermostat: Alpha Release

- The result is now closer...



  - If we double the # of load steps again (‘loadstp=40), the solution improves yet again, confirming our intuition.
  - The accuracy of this technique is directly proportional to how frequently we can check the temperature and set turn ‘on’ the load accordingly.

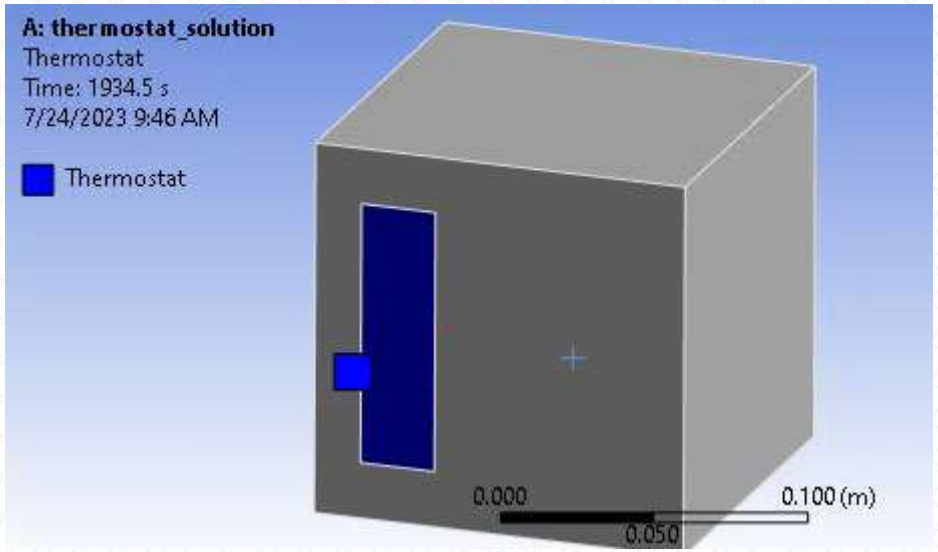


Thermostat: Alpha Release

- A better way to implement a thermostat is to use a special element type built for this purpose (the combin37 element)
- This is what the new Thermostat app does
- After installing it, go to the ‘Thermostat’ tab in the Mechanical top menu and hit the ‘Thermostat Control’ button (make sure to either suppress or delete the previous command object first).
- Then make the selections shown (refer back to slide 25 if necessary. The monitor location is a single vertex on the same face as the load application –not connected to any edges or corners)

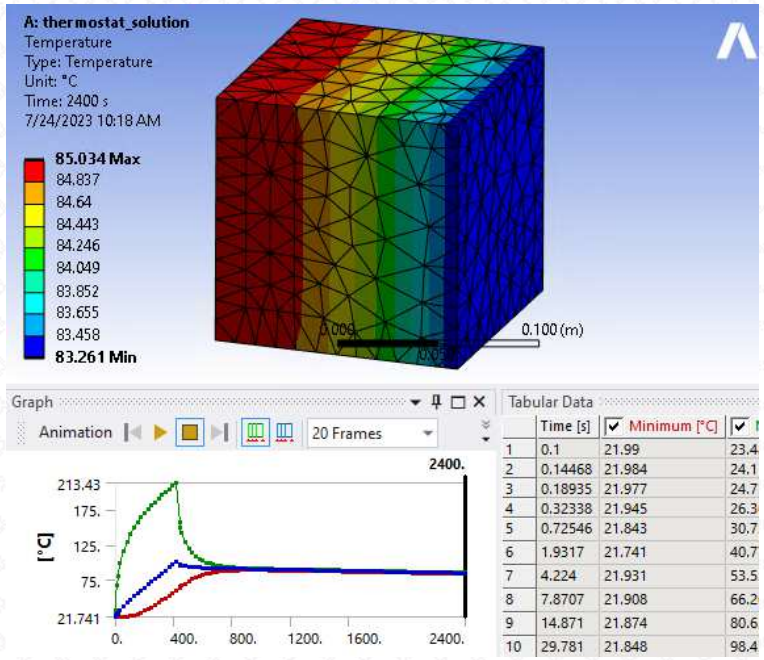
Details of "Thermostat"

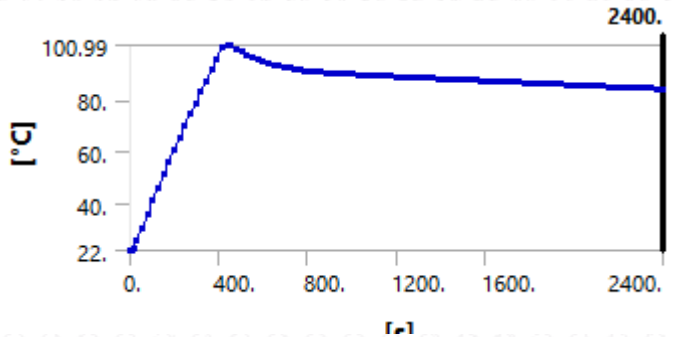
Heat Source/Sink Location	
Scoping Method	Geometry Selection
Geometry	1 Face
Temperature Monitor Location	
Scoping Method	Geometry Selection
Geometry	1 Vertex
Thermostat Control Properties	
Off Set Point	100 °C
On Set Point	80 °C
Heat	1200 W
Mode	Heating
Definition	
View Results?	No



Thermostat: Alpha Release

- Because the the combin37 element is not limited to load steps for temperature checking, it is much more efficient at converging to a target temperature (it turned ‘off’ at nearly the correct time)
- Note that, because the thermostat element checks the monitor temperature continuously, it requires both an ‘on’ temperature check and an ‘off’ check (hence the reason for these fields in the thermostat app)
- In this example, the monitor temperature never reaches the ‘on’ setpoint of 80°C, and so the thermostat only turns on once within 2400 s.





Thermostat: Alpha Release

- We can get increased accuracy by making the ‘On Set Point’ temperature closer to the ‘Off Set Point’ temperature, but this requires smaller timesteps (this is a highler nonlinear type of analysis, and so nonlinear analysis settings become quite important).
- Modify the ‘On Set Point’ to be 90°C
- This change will require smaller timesteps, so make changes to the Analysis Settings as shown
- Re-run the analysis

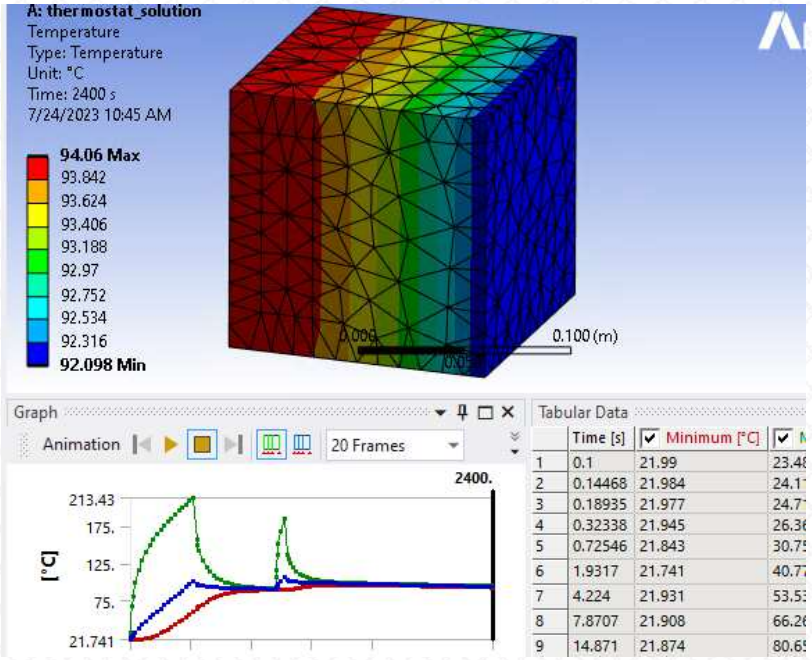
Heat Source/Sink Location	
Scoping Method	Geometry Selection
Geometry	1 Face
Temperature Monitor Location	
Scoping Method	Geometry Selection
Geometry	1 Vertex
Thermostat Control Properties	
Off Set Point	100 °C
On Set Point	90 °C
Heat	1200 W
Mode	Heating
Definition	
View Results?	No

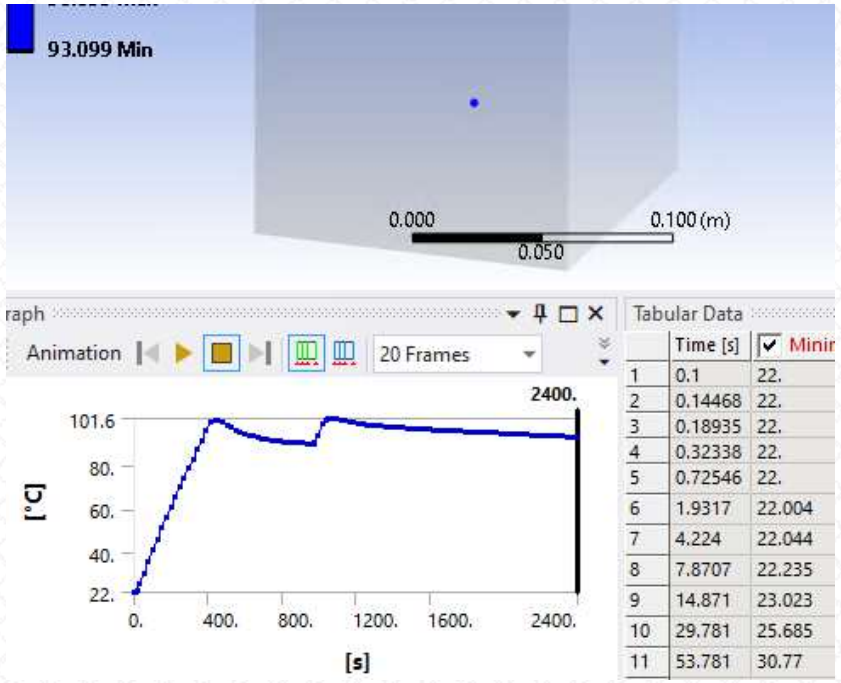
Details of "Analysis Settings"

Step Controls	
Number Of Steps	1.
Current Step Number	1.
Step End Time	2400. s
Auto Time Stepping	On
Define By	Time
Initial Time Step	0.1 s
Minimum Time Step	1.e-006 s
Maximum Time Step	24. s
Time Integration	On
Solver Controls	
Solver Type	Program Controlled
Radiosity Controls	
Nonlinear Controls	
Heat Convergence	Program Controlled

Thermostat: Alpha Release

- As the ‘On Set Point’ approaches the ‘Off Set Point’, accuracy increases, but the analysis becomes more challenging (requires smaller timesteps)





Thermostat: Alpha Release

- We've only explored the thermostat functionality in 'heating' mode
- It can also be used to cool volumes in much the same way. The only real differnce in usage in the 'cooling' case is that the 'On Set Point' temperature should be greater than the 'Off Set Point' temperature –the opposite of our usage in this example.
- See the documenation folder of the thermostat app for a cooling example.

Final Notes

- In addition to the new power threshold capability (the ‘limit load’ field) added to the PID controller, we have changed it’s name to ‘PID Controller’ from the earlier ‘PID Thermostat’ to more accurately reflect its function
- In addition, we’ve added a new ‘Thermostat’ app to our lineup of thermal tools, so that users may not only see the distinction between these two types of devices, but explore each.