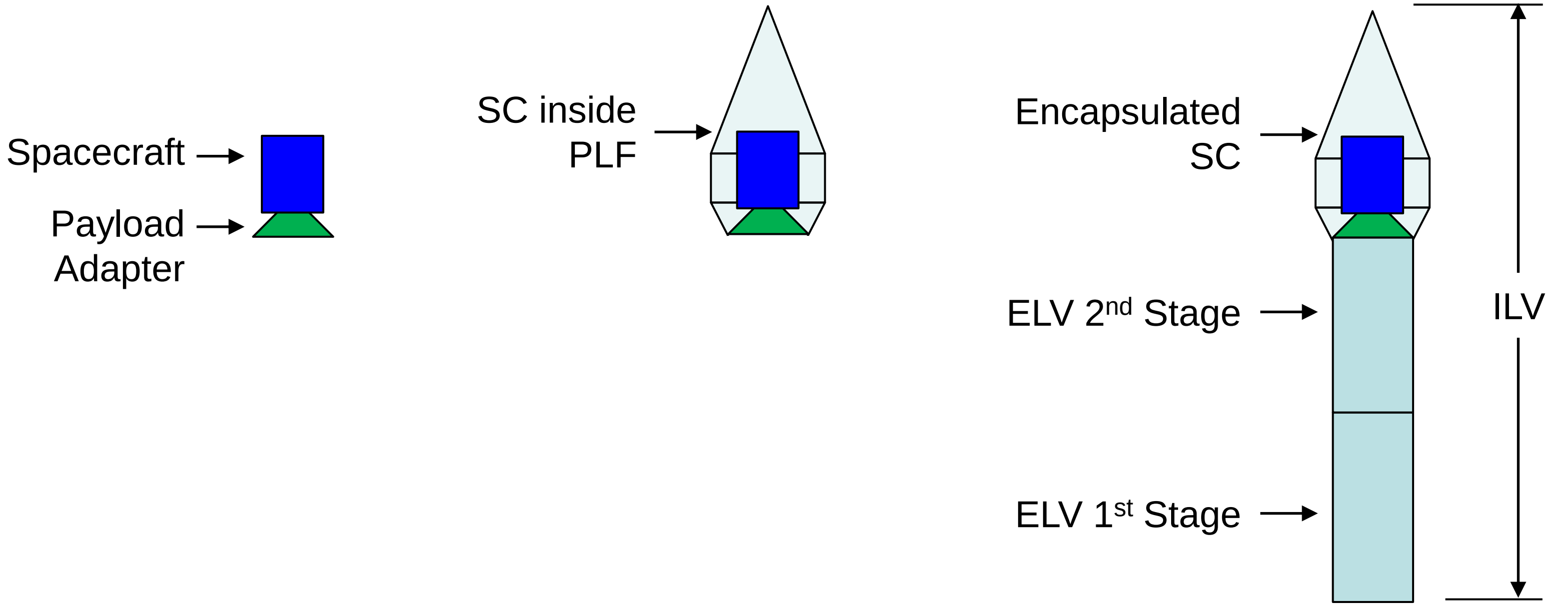
# Simplified Method for Simulating Ascent Configuration Changes of Large Scale Integrated Launch Vehicles in Thermal Desktop

Robert Kwas
NASA Kennedy Space Center
Launch Services Program

Thermal & Fluids Analysis Workshop
TFAWS 2013
July 29-August 2, 2013
Kennedy Space Center
KSC, FL

# Integrated Launch Vehicle Overview

- Integrated Launch Vehicle (ILV) basically consists of a Spacecraft (SC) and an Expendable Launch Vehicle (ELV)
  - The ELV is typically a two stage rocket
  - The SC is typically encapsulated in a protective covering known as a Payload Fairing (PLF)
  - The encapsulated SC is mated to forward end of the ELV by a structural fitting known as a payload adapter
  - In the mated configuration, the ELV and SC are commonly referred to as an ILV

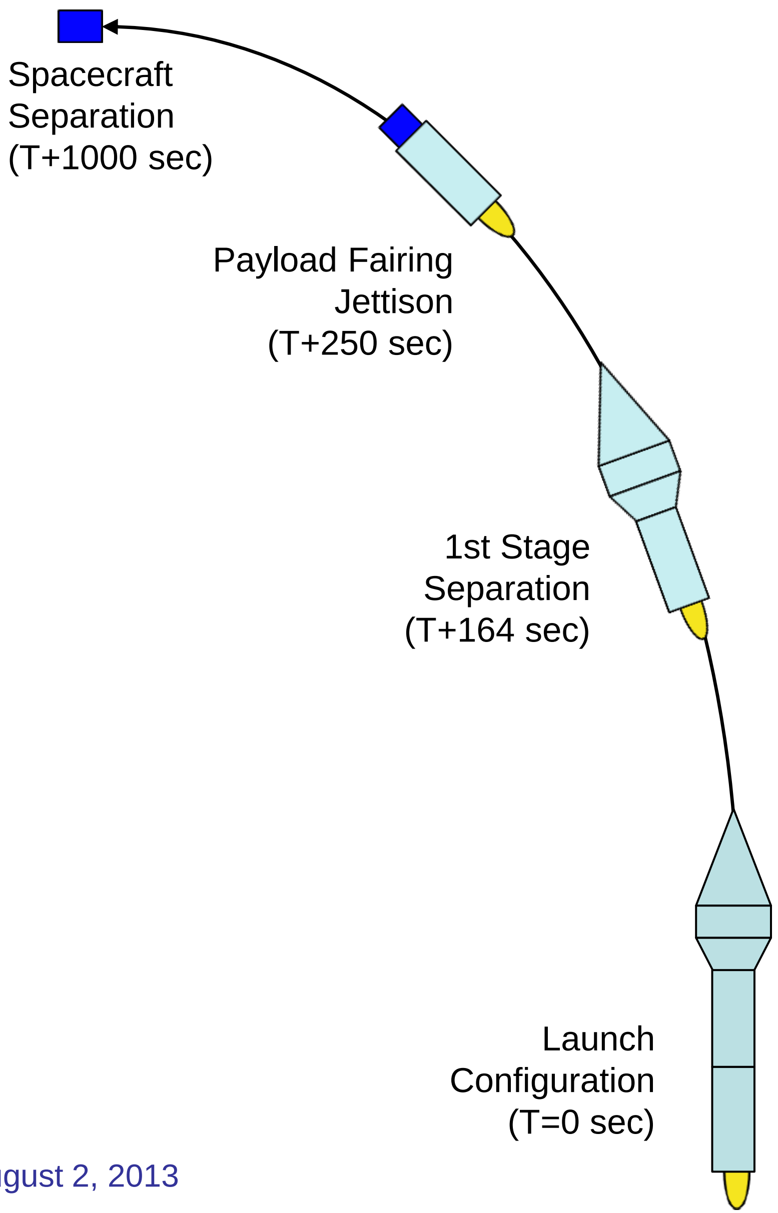


# Integrated Launch Vehicle Overview

- An ILV Thermal Analysis (ITA) would typically cover most portions of the prelaunch and ascent mission phase
  - The prelaunch analysis would typically start with roll – out and extend through Countdown Launch Procedures (CLP)
  - The ascent analysis would typically start with launch and extend through spacecraft separation

# Integrated Launch Vehicle Overview

- One of the more challenging aspects of performing an ITA is simulating the ascent mission phase
- The ILV typically incurs several changes to its configuration during ascent including:
  - Launch
  - Post 1st Stage Separation
  - Post PLF Jettison
  - Spacecraft Separation
- How can you make a single Thermal Desktop (TD) model simulate all four ILV ascent configurations?

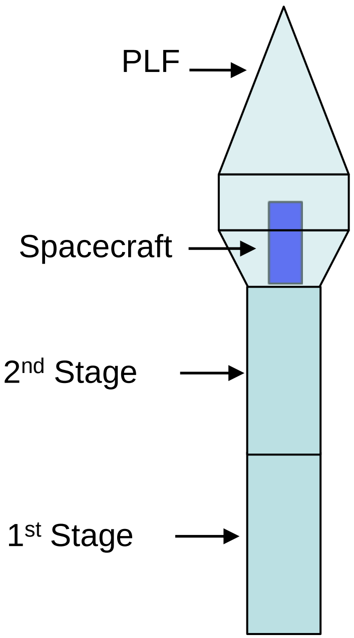


# Configuration Changes: Traditional Method

- Traditionally, the ascent mission phase is simulated by developing a specific case for each configuration
  - Build statements are utilized to build only that portion of the thermal linear network associated with the current case
    - A build statement is a SINDA command that precedes a user specified list of submodels
    - Only those submodels in the list are built by the SINDA preprocessor.
  - Radiation Analysis Groups (RAG) are utilized to build only that portion of the radiation network associated with the current case.
    - RAG is a list of TD surfaces referenced in the radiation task tab
    - Only those surfaces listed in the RAG are consider by RadCAD during the radiation solution
  - All cases are then thermally sequenced by initializing their temperatures to the final predicted temperatures from the previous case

# Configuration Changes: Traditional Method

| | CASE1 | CASE2 | CASE3 | CASE4 |
|---|---|---|---|---|
| **Configuration** | Launch | Post 1st Stage Separation | Post Payload Fairing Jettison | Spacecraft Separation |
| **Build Statement** | All Submodels | PLF, Spacecraft & 2nd Stage Submodels | Spacecraft & 2nd Stage Submodels | Spacecraft Submodels |
| **RAG** | All Surfaces | PLF, Spacecraft & 2nd Stage Surfaces | Spacecraft & 2nd Stage Surfaces | Spacecraft Surfaces |



# Configuration Changes: Traditional Method

- Traditional method can be very difficult to apply to large - scale ILV models
  - The case temperature initializations can be very tedious
  - RAG can contain thousands of surfaces and are difficult to verify
  - Build statements can contain hundreds of submodels and are difficult to verify
  - Example build statement for large ILV model

```
BUILD AV5, TABLES, FEM, AVNICS, LH2TNK, STUBAD, PLA, TDPROPS, BAFFLE, TANK1, FLANGE,
   +    AFTSPL, BNDRY, TAPROPS, ARRAY1, ARRAY2, BATT1, BATT2, BATT3, MLI1, MLI2, MLI3, MLI4,
   +    AFTBLK, EBEAM, ECU, ELCPNL, AIP, H2FEDL, O2FEDL, RDU, ACTRS, WALL, TRACKER2, TVC,
   +    HEBOT1, HEBT12, HEBOT3, HYDBOT, RL10, VNTVLV, T4PNL, REMS, NOZZLE, CONE, TRACKER1,
   +    O2TANK, H2TANK, WRTNL, EGTVC1, EGTVC2, ISA500, FAIRNG, D1666, THRUSTR, PAF, VENT,
   +    SEPSYSTM, BOX, C2A8, C0A7, C2A0, C2A1, C2A9, C20A195, MAIN, SOLAR, BOX2, BOX533,
   +    ESAC, OMNI, C2A10, STWREFL, TRUSS, WSAC, CONTLR, PANEL, STAR1, STAR2, VANE, COVER,
   +    VALVE1, VALVE2, VALVE3, FAIRING_CL1, FAIRING_NC1, FAIRING_NC2, FAIRING_NC3, DOOR,
   +    FAIRING_NC4, FAIRING_CL2, FAIRING_BT1, SPACE, BLOCK1, BLOCK2, BLOCK3, BRACKET1,
   +    FAIRING_NC3, FAIRING_CL1, FAIRING_BT2, BLOCK4, BLOCK5, BLOCK6, BRACKET2, DAMPER,
   +    LOX_TANK, IU1, IU2, IU3, SM1, SM2, SM3, BLANKET1, BLANKET2, BLANKET3, BLANKET4,
   +    RADIATR1, RADIATR2, RADIATR3, THRUSTR1, THRUSTER2, THRUSTER3, THRUSTER4
```

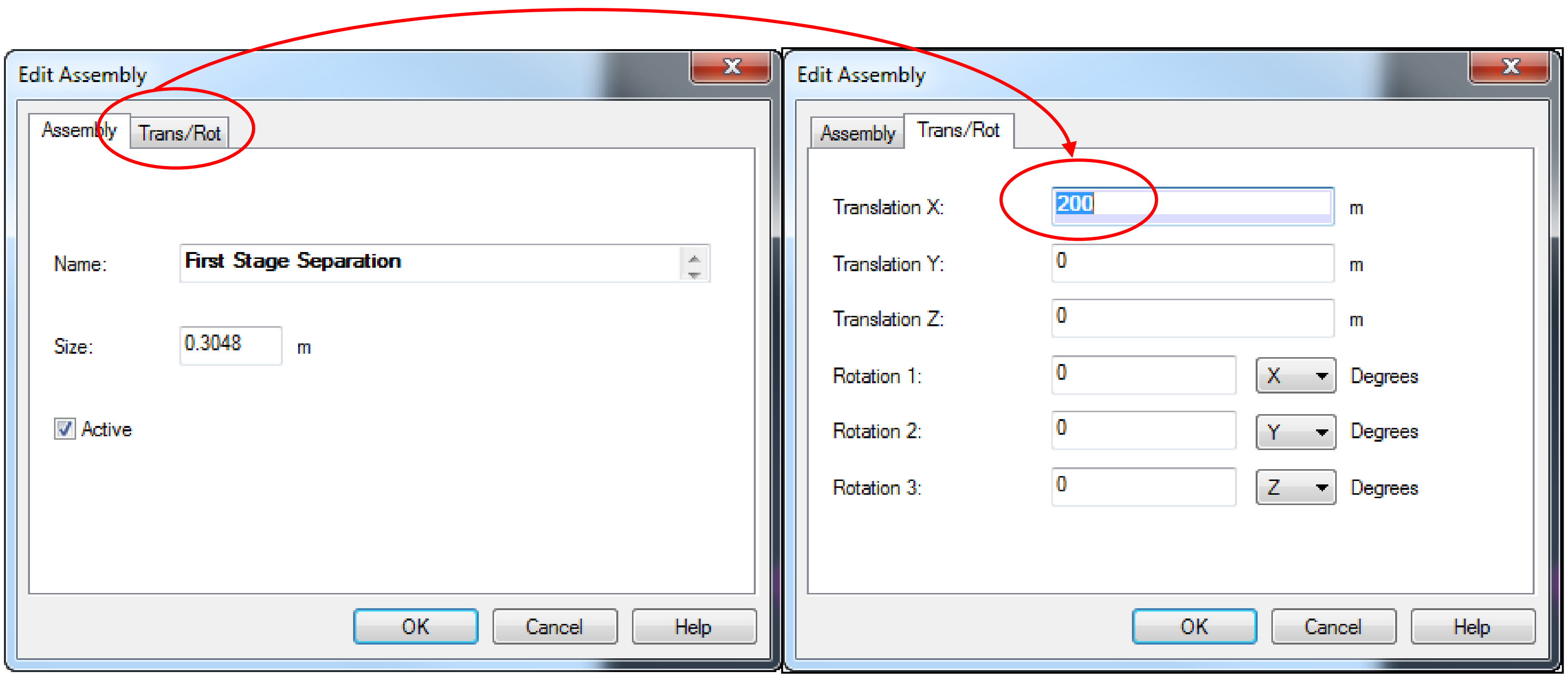
# Configuration Changes: Traditional Method

- Traditional method generates lots of TD cases
  - Assessing ten ascent trajectories would require a run matrix containing forty cases
  - Correct build statements and RAG's must be constructed for each case

| Ascent Trajectory | Ascent Configurations | | | |
|---|---|---|---|---|
| | Launch | Post 1st Stage Separation | Post PLF Jettison | Post 2nd Stage Separation |
| 1 | CASE1 | CASE2 | CASE3 | CASE4 |
| 2 | CASE5 | CASE6 | CASE7 | CASE8 |
| 3 | CASE9 | CASE10 | CASE11 | CASE12 |
| 4 | CASE13 | CASE14 | CASE15 | CASE16 |
| 5 | CASE17 | CASE18 | CASE19 | CASE20 |
| 6 | CASE21 | CASE22 | CASE23 | CASE24 |
| 7 | CASE25 | CASE26 | CASE27 | CASE28 |
| 8 | CASE29 | CASE30 | CASE31 | CASE32 |
| 9 | CASE33 | CASE34 | CASE35 | CASE36 |
| 10 | CASE37 | CASE38 | CASE39 | CASE40 |

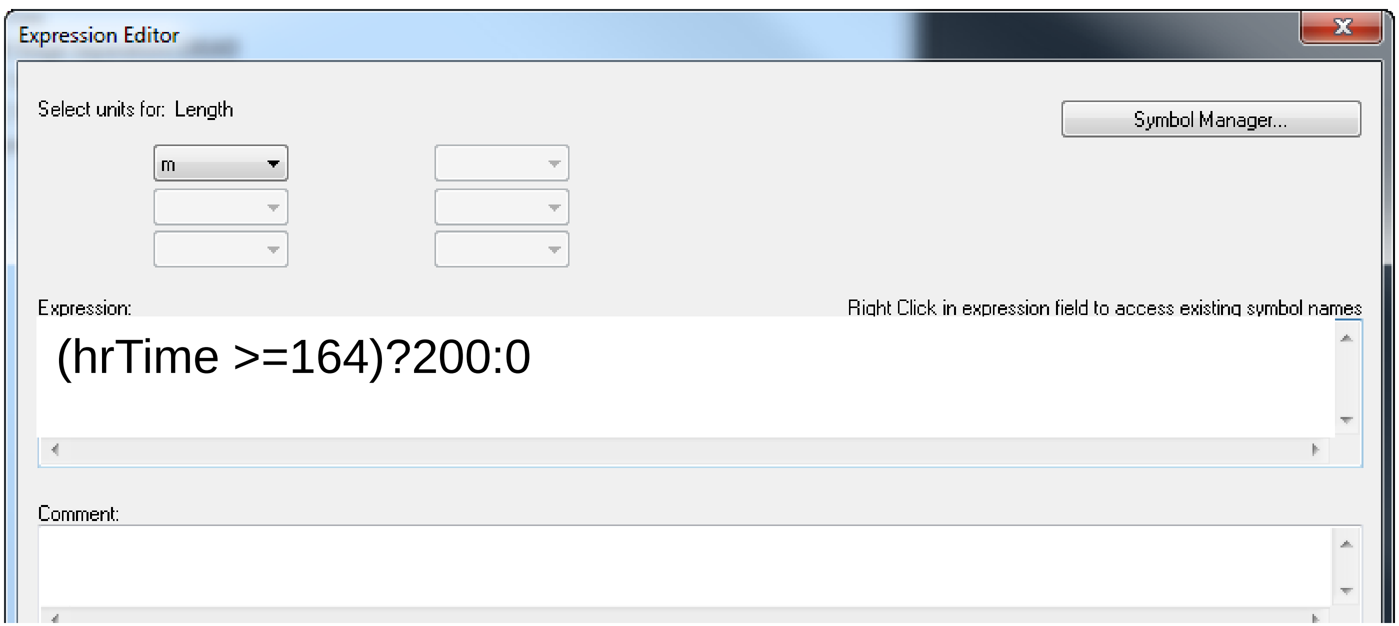
# Configuration Changes: Assembly Method

- An assembly is an articulator that can be used to transform model geometry
  - Assemblies can translate and/or rotate surfaces
  - When specified as a constant value in the input field, the assembly transformation is executed prior to running the model (if assembly is active)



# Configuration Changes: Assembly Method

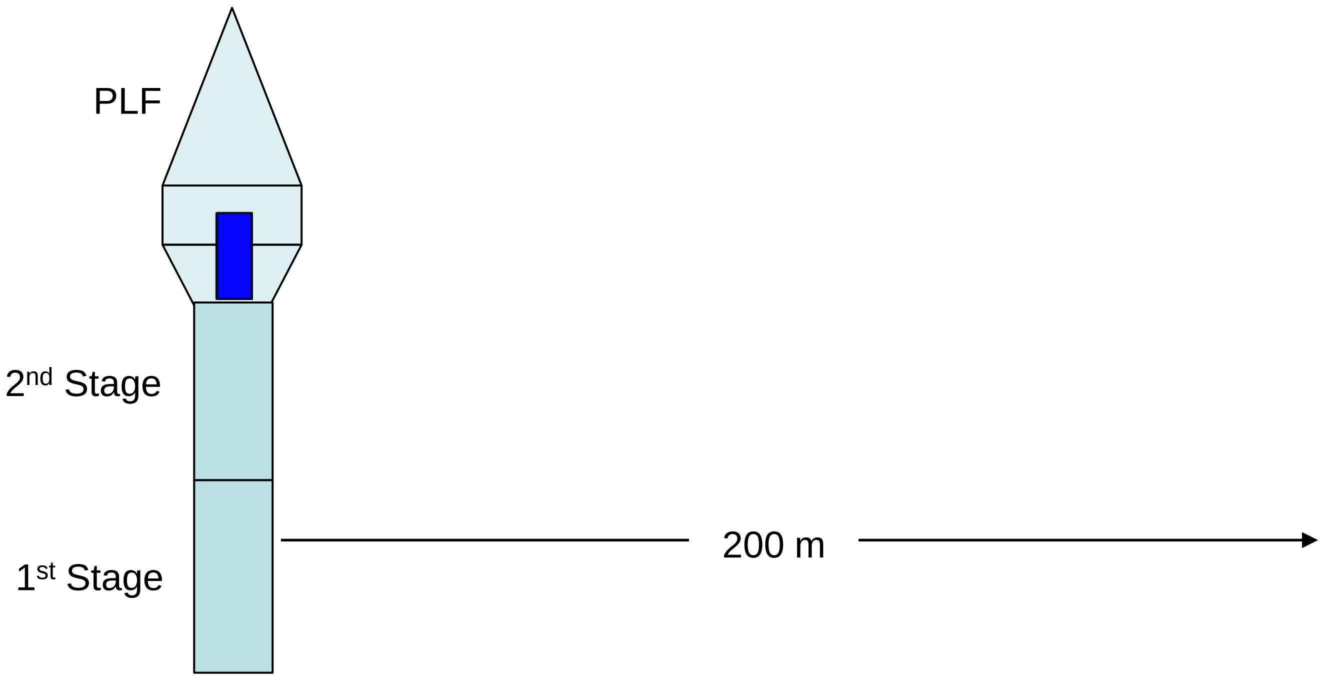
- Used in conjunction with an articulating orbit, assemblies can transform model geometry as a function of orbital time
  - hrTime is an internally generated symbol that automatically stores the time associated with the current orbital position.
  - hrTime can be used in the assembly expression editor in conjunction with TD logic and arithmetic operators
  - This allows the user to express transformations as complex functions of hrTime



# Configuration Changes: Assembly Method

- By attaching the1$^{st}$ Stage surfaces the to aforementioned assembly
  - The model will stay in the launch configuration as long as hrTime <164 seconds
  - When hrTime ≥ 164 seconds, the1$^{st}$ Stage will be translated 200 meters in the +X direction
  - After the translation, the radiant heat transfer between the 1$^{st}$ Stage and the rest of the model is broken
  - Can do the same for the PLF and the 2$^{nd}$ Stage
  - Assembly method is easy to verify

Staging Events.avi



# Configuration Changes: Assembly Method

- Advantages of the assembly method
  - Very easy to verify
  - Can simulate an entire ascent trajectory with one TD case
    - Simplifies post processing
    - Precludes case temperature initializations
  - No need to use build statements
  - No need to use multiple RAG

# Configuration Changes: Assembly Method

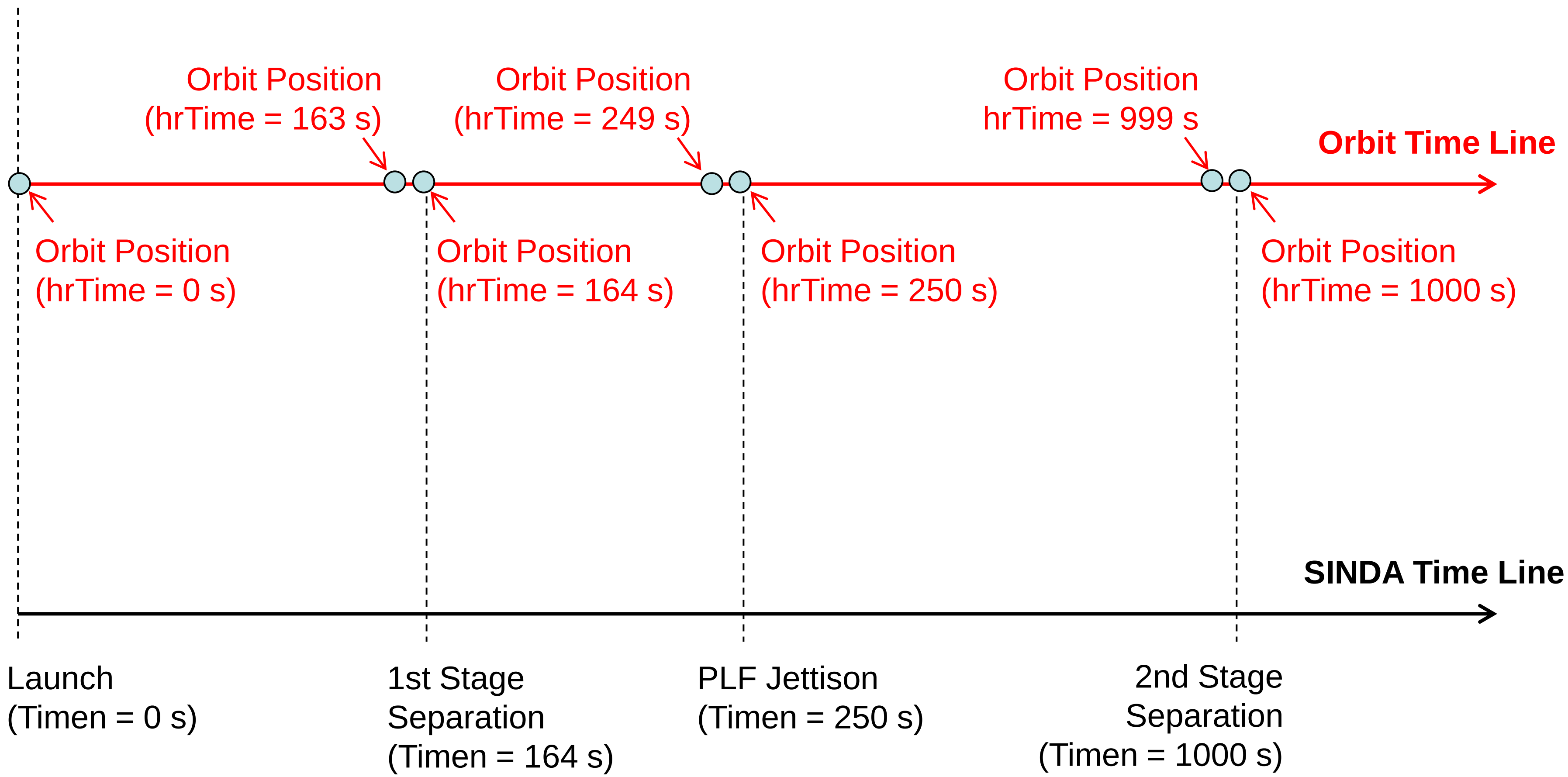
- Note that you would only need ten cases to assess ten ascent trajectories using assembly method (traditional method required 40 cases)

| Ascent Trajectory | Launch – SC Separation |
|---|---|
| **1** | CASE1 |
| **2** | CASE2 |
| **3** | CASE3 |
| **4** | CASE4 |
| **5** | CASE5 |
| **6** | CASE6 |
| **7** | CASE7 |
| **8** | CASE8 |
| **9** | CASE9 |
| **10** | CASE10 |

# Configuration Changes: Assembly Method

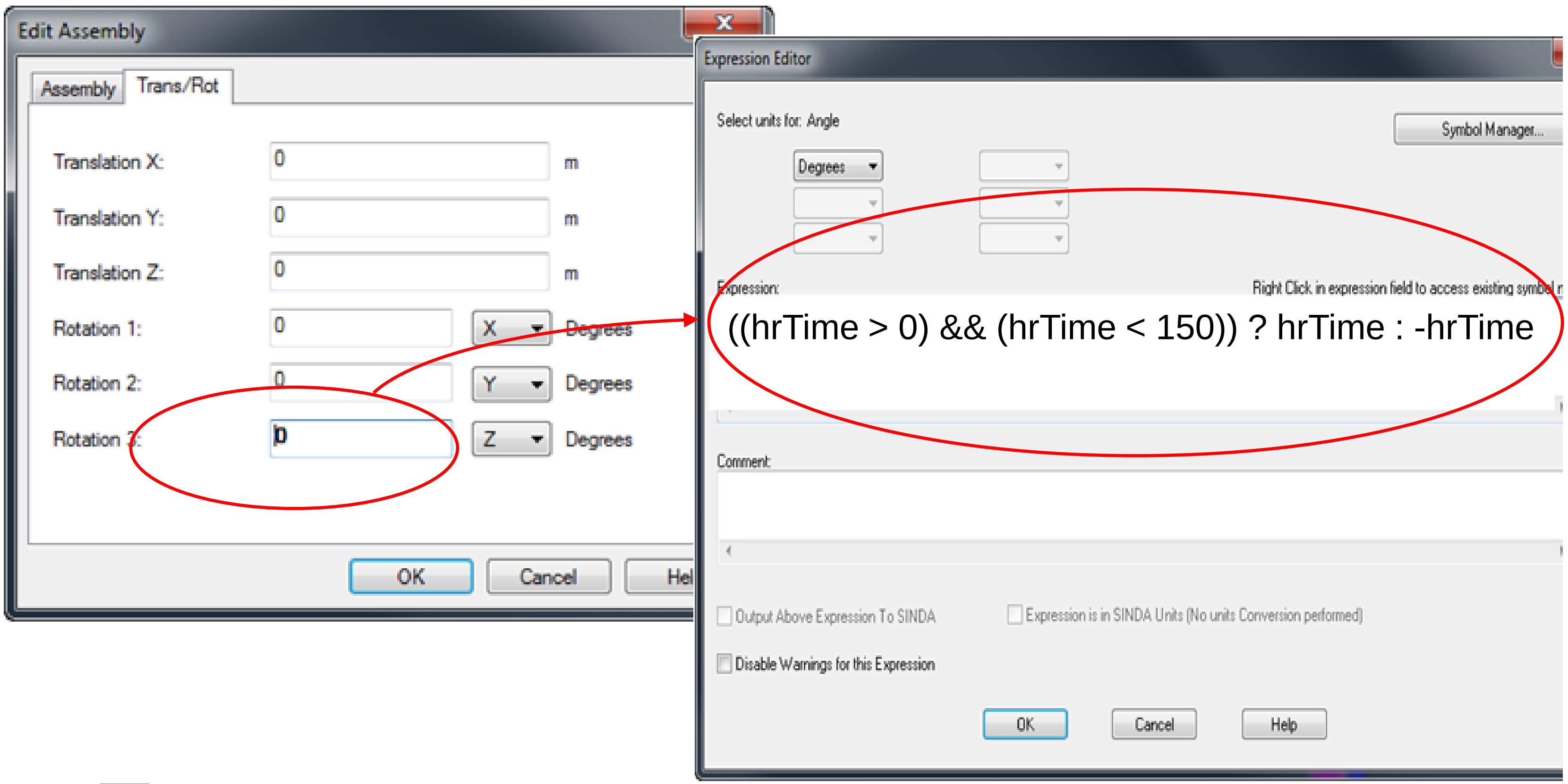
- Disadvantages of using assembly method
  - Translations can significantly increase the radiation model domain
    - RadCAD solution time increases as model domain increases
    - Can mitigate by increasing Oct – Tree parameters (subdivisions & max surfaces per cell).
    - Use as short a translation distance as possible.
  - Need to shoot articulating RadK's
    - RadCAD solution will take longer
      - Requires shooting ~7 orbital positions
      - Generally not significant in comparison to number of positions required for solar, albedo, earth IR, & FMH
    - Orbit time – line and SINDA time – line must be synchronized

# Configuration Changes: Assembly Method



# Using Assemblies to Simulate PTC Rolls

- Assemblies can also be used to rotate model geometry
  - Useful for simulating Passive Thermal Control (PTC) rolls





PTC Roll.avi

# Assembly Method Summary

- Assemblies provide the user with a powerful method for simulating dynamic motion in a TD model.
  - Can simulate translational motion (i.e., staging events)
  - Can simulate rotational motion (i.e., PTC rolls).
- Assemblies can simulate model configuration changes without having to use cumbersome user – written build statements, multiple RAG's and multiple cases.
- Assemblies are relatively easy to verify, but can increase RadCAD solution time.

# Questions

- Questions?

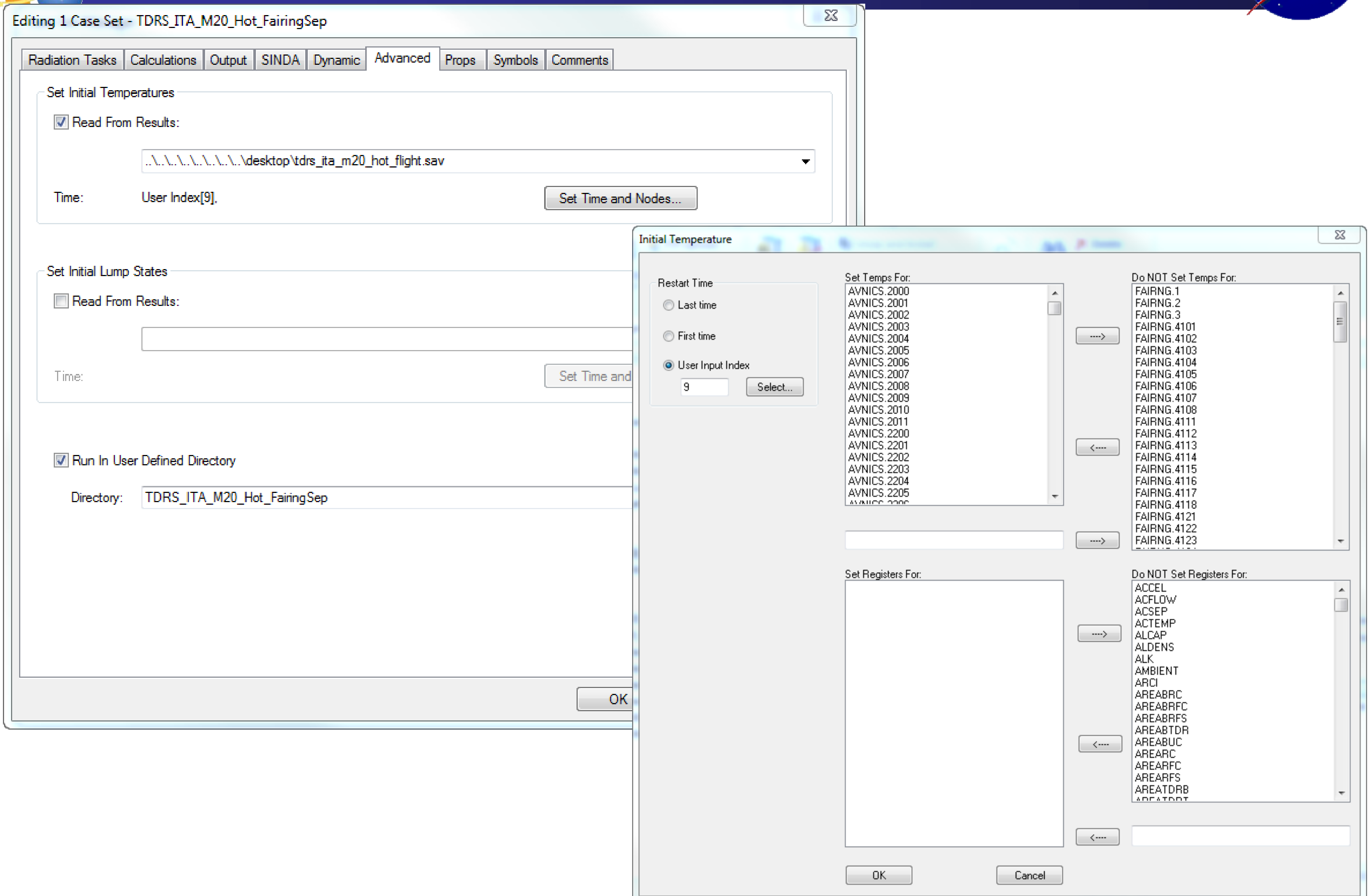
# Back Up

- Create a symbol (hrTime2) and define it as some linear function of hrTime

  hrTime2 = C1 x hrTime + C2

- Use hrTime2 in assembly expression editor

# Back Up



Editing 1 Case Set - TDRS_ITA_M20_Hot_FairingSep
Radiation Tasks
Calculations
Output
SINDA
Dynamic
Advanced
Props
Symbols
Comments
Set Initial Temperatures
Read From Results:
..\..\..\..\..\..\..\..\desktop\tdrs_ita_m20_hot_flight.sav
Time:
User Index[9],
Set Time and Nodes...
Set Initial Lump States
Read From Results:
Time:
Set Time and
Run In User Defined Directory
Directory:
TDRS_ITA_M20_Hot_FairingSep
OK
Initial Temperature
Restart Time
Last time
First time
User Input Index
9
Select...
Set Temps For:
AVNICS.2000
AVNICS.2001
AVNICS.2002
AVNICS.2003
AVNICS.2004
AVNICS.2005
AVNICS.2006
AVNICS.2007
AVNICS.2008
AVNICS.2009
AVNICS.2010
AVNICS.2011
AVNICS.2200
AVNICS.2201
AVNICS.2202
AVNICS.2203
AVNICS.2204
AVNICS.2205
Do NOT Set Temps For:
FAIRNG.1
FAIRNG.2
FAIRNG.3
FAIRNG.4101
FAIRNG.4102
FAIRNG.4103
FAIRNG.4104
FAIRNG.4105
FAIRNG.4106
FAIRNG.4107
FAIRNG.4108
FAIRNG.4111
FAIRNG.4112
FAIRNG.4113
FAIRNG.4114
FAIRNG.4115
FAIRNG.4116
FAIRNG.4117
FAIRNG.4118
FAIRNG.4121
FAIRNG.4122
FAIRNG.4123
Set Registers For:
Do NOT Set Registers For:
ACCEL
ACFLOW
ACSEP
ACTEMP
ALCAP
ALDENS
ALK
AMBIENT
ARCI
AREABRC
AREABRFC
AREABRFS
AREABTDR
AREABUC
AREARC
AREARFC
AREARFS
AREATDRB
OK
Cancel

# Back Up

**10.1.4 Internally Generated Heating Rate Symbols**

When a heating environment has been defined in a model, Thermal Desktop will automatically create several symbols that define part of the orbit, as well as the status of the currently calculating or displayed position. This allows complicated orbital maneuvers, or any other parameterization, to be programmed as a function of orbital position. These symbols are listed in Table 10-2.

Users can reference these symbols to program the behavior of trackers, assemblies, and vehicle rotations of an orbit. These symbols, however, are not user-definable (do not modify them directly) and are not updated in SINDA.

While the symbols are not updated in SINDA, the time-dependent values can be accessed from SINDA logic by selecting the Output HR Symbols to SINDA check box on the Heatrate Output tab of the heatrate task in the Case Set Manager (see Section 9.1.4)