Preliminary Development of a TSS and SINDA/FLUINT to ESARAD/ESATAN Thermal Model Converter

Kan Yang and Hume Peabody
NASA Goddard Space Flight Center

Thermal & Fluids Analysis Workshop
TFAWS 2014
August 4-8, 2014
NASA Glenn Research Center
Cleveland, OH

Overview

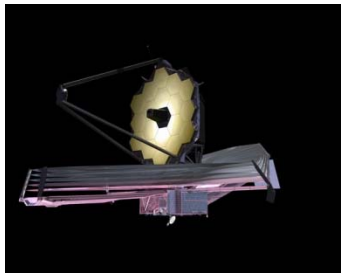
A recent effort has been initiated by NASA GSFC to develop a standardized converter between ESA-based thermal model files and Thermal Desktop/TSS in interagency collaborations

The following topics will be covered:

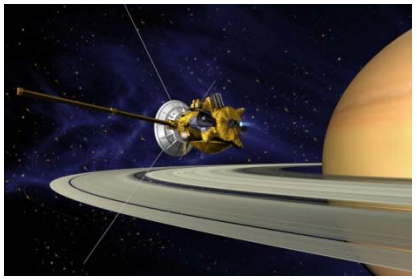
- Introduction to TSS/SINDA and ESARAD/ESATAN
- Geometric Math Model Framework
  - Conversion examples
- Thermal Math Model Framework
  - Comparison between SINDA and ESATAN
  - Conversion from ESATAN to SINDA
  - Conversion from SINDA to ESATAN
- Conclusions and Future Work

Introduction

- Proposal submitted to NESC to develop an agency-supported utility to address conversion between ESA and NASA thermal model formats
- Some joint projects between ESA and NASA include:


JWST


MetOp


Cassini-Huygens


Solar Orbiter

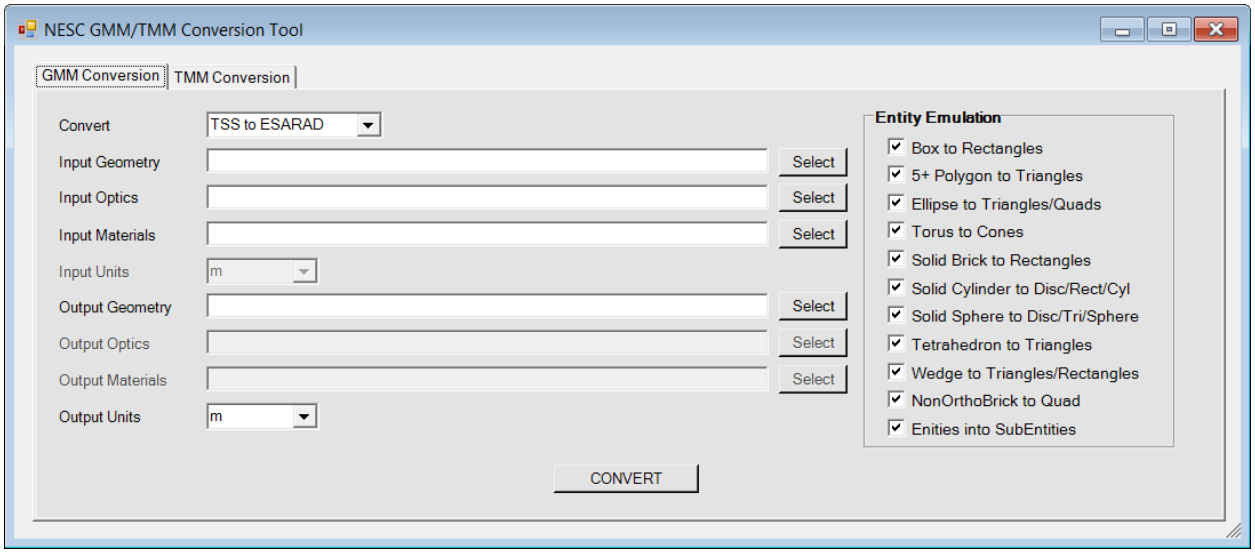
- Conversion between models requires code to handle the following thermal model attributes:
  - Geometric Math Model: spatial location and size definition of surfaces that represent the radiating surfaces of a spacecraft, including thermo-optical properties, active sides, and node numbers
  - Thermal Math Model: all of the nodes, conductors, arrays, variables, and logic required by the solver to achieve an energy balance on the spacecraft and output a thermal solution in terms of temperatures and heat flows

Introduction

Function	ESATAN-TMS Suite	TSS/SINDA
Graphical User Interface (GUI) / GMM Generator	ESATAN-TMS Workbench	SPACE3D / TSS: .tssma, .tssgm, .tssop files
Radiation (Radk) Calculations	ESARAD (.erk): instructions to execute ray trace → Radk files	TSS → Ray trace produces radiation coupling and heat rate database
Thermal Analysis (TMM)	ESATAN	SINDA/FLUINT

GMM Framework

- GMM converter developed in VB.NET and takes advantage of .NET framework
- Includes all necessary properties to specify input/output file names, formats and units (GUI shown below)



- Dedicated reader methods take input ESARAD or TSS code and store it as collection of surfaces, optical properties, thermophysical properties, points, and variables
- Dedicated writer methods to export collections in desired output format
- Use of Boolean/cutting operations and FE/edge node numbering not handled (ESARAD doesn't support user-defined node number of elements)

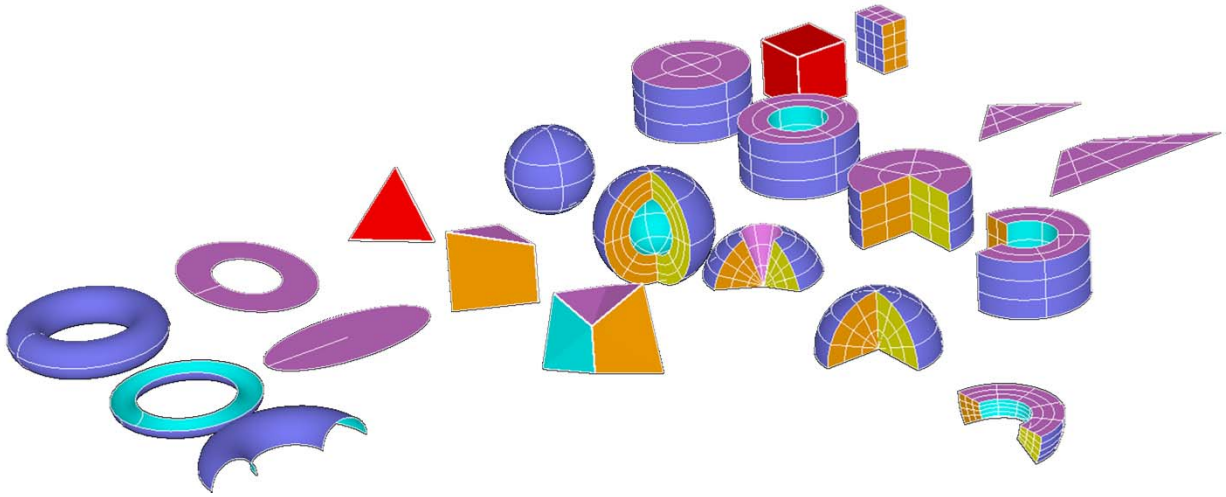
GMM Framework

- Methods provide emulation of surface types that don't exist in either code
- Subdivided surfaces represented by assemblies of single node surfaces to account for differences in node number and surface subdivision capabilities

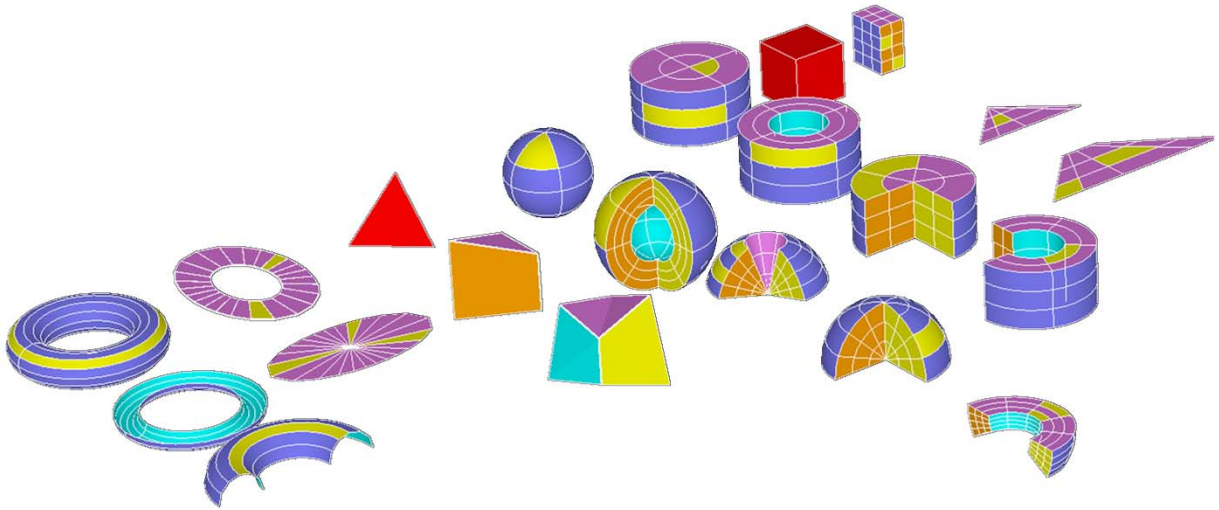
Entity Type	TSS	ESARAD	Subdiv-idable?	Emulated?	Emulation
Assembly	Y	Y	N/A	N/A	
Rectangle	Y	Y	Y	Y	Rectangles
Triangle	Y	Y	N	N/A	
Trapezoid	Y	Y	Y	Y	Triangles+Quadrilaterals
Disc	Y	Y	Y	Y	Discs
Cylinder	Y	Y	Y	Y	Cylinders
Sphere	Y	Y	Y	Y	Spheres
Cone	Y	Y	Y	Y	Cones
Paraboloid	Y	Y	Y	Y	Paraboloids
Polygon	Y	N	N	Y	Triangles
Box5Sides	Y	N	N	Y	Rectangles
Box6Sides*	Y	N	N	Y	Rectangles
Torus	Y	N	Y	Y	Cones
Brick**	Y	N	Y	Y	Rectangles
SolidCylinder**	Y	N	Y	Y	Disc+Rect+Cyl
SolidSphere**	Y	N	Y	Y	Sphere+Disc
Tetrahedron**	Y	N	N	Y	Triangles
SolidWedge**	Y	N	N	Y	Triangles+Rectangles
NonUniformBrick**	Y	N	N	Y	Quadrilateral
Ellipse	Y	N	Y	Y	Triangles or Quadrilaterals
Ellipsoid	Y	N	Y	N	
Ogive	Y	N	Y	N	
Hyperboloid	Y	N	Y	N	
Elliptic_Cone	Y	N	Y	N	
Box*	N	Y	Y	Y	Rectangles
Triangle (subdividable)	N	Y	Y	Y	Triangles+Quadrilaterals
Quadrilateral (subdividable)	N	Y	Y	Y	Quadrilaterals
Triangular Prism	N	Y	Y	N	
Half Space	N	Y	N/A	N	
* Box type entities have different numbering methodologies on faces					
** Only outer shell surfaces are emulated					

GMM Conversion Examples

Base Set of TSS Surfaces

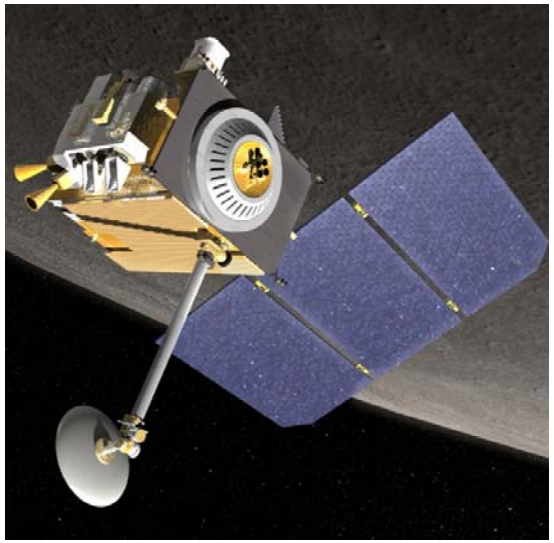


After emulation and nodal subdivision…

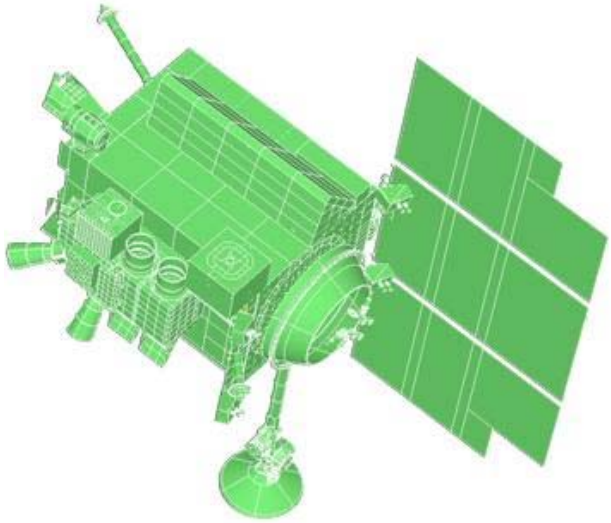


GMM Conversion: TSS ➔ ESARAD

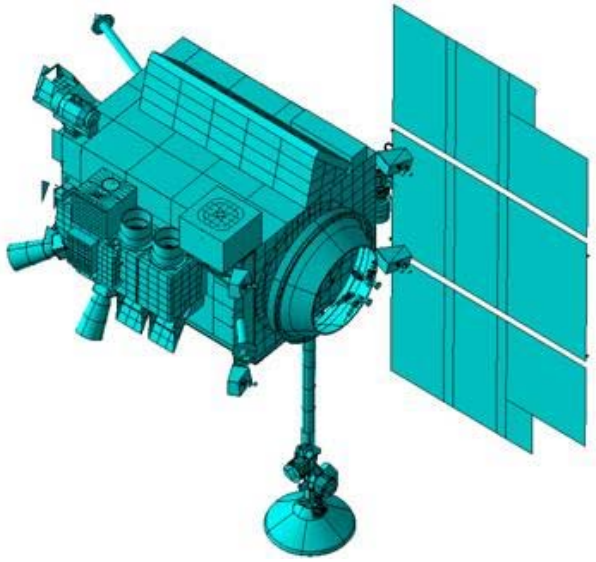
Lunar Reconnaissance Orbiter



Spacecraft

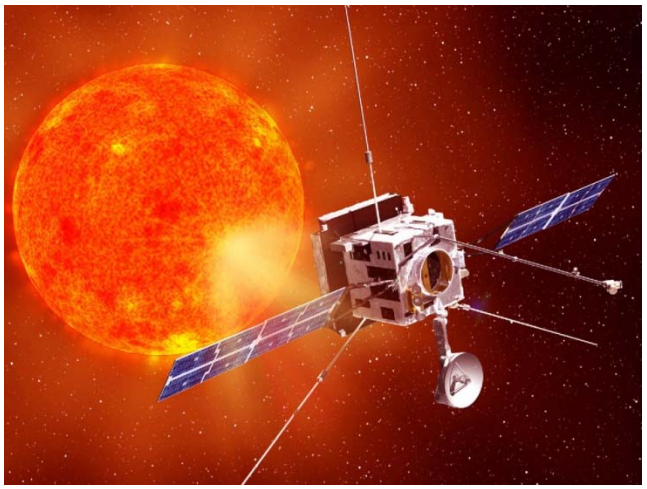


TSS



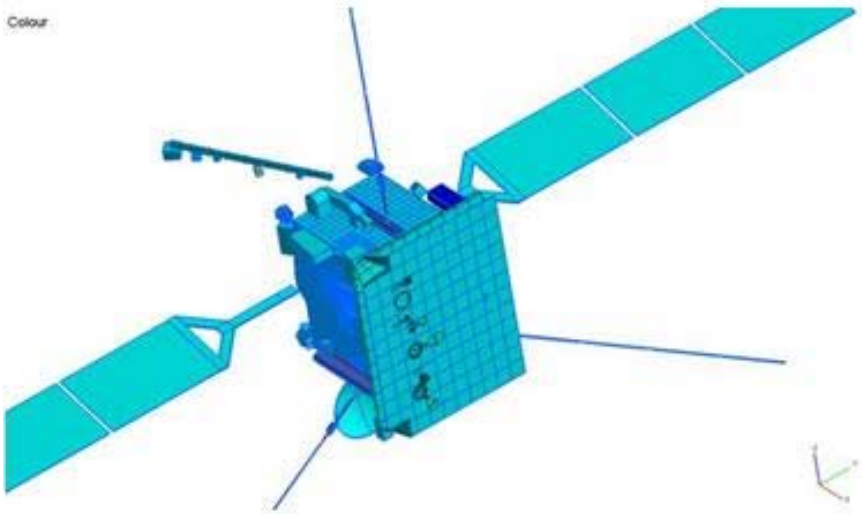
ESARAD

GMM Conversion : ESARAD ➔ TSS

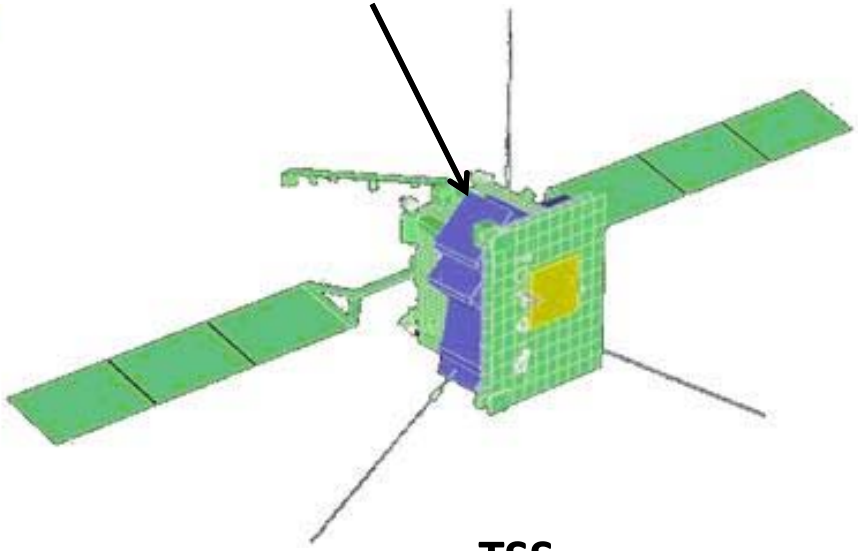


Spacecraft

ESARAD cutting operations not converted



ESARAD



TSS

TMM Framework

- Includes necessary properties to support read/write of ESATAN and SINDA/FLUINT input files
- Dedicated reader methods populate collections of data blocks (nodes, arrays, registers, conductors) and comments, with writer methods to output to desired format
- Other methods used to convert FORTRAN or MORTRAN logic in VARIABLE and execution blocks

Goal of TMM converter development: to develop a program capable of converting a majority of TMM statements encountered in SINDA and ESATAN codes of major flight projects

➔ Too many variations in syntax specific to both formats to develop converter in feasible amount of time that can support every feature, contingency or special case.

➔ Due to unstructured nature of logic, all statements output by converter must be checked to ensure accurate, physical solution can be obtained in converted format

ESATAN vs. SINDA: Nodes and Conductors

	Description	ESATAN	SINDA
Node Block	Diffusion Node	Y	Y
	Arithmetic Node	N (Diffusion with C=0)	Y
	Boundary Node	Y	Y
	Inactive Node	Y	N
	Heater Node	N (Same as Boundary)	Y
	Nested Submodel Hierarchy	Y	N
	Global Submodel References	N	Y
	Node Variables Created	T,QS,QA,QE,QI,QR,A,EPS, ALP,L,C	T,Q,C
	Simultaneous creation of multiple nodes	DO Loop	Supported: GEN, SIM, DIM statements
	Nodes with temp-dependent capacitances	INTERP function call in C value	Supported: SIV, DIV statements
Conductor Block	Linear Conductor	Y	Y
	Radiative Conductor	Y	Y
	Conductor Reference	Node Pair	Number
	Simultaneous creation of multiple conductors	DO Loop	Supported: GEN, SIM, DIM statements
	Conductors with temp-dependent conductances	INTERP function call in conductor value definition	Supported: SIV, DIV statements
	Trans-submodel connections	Only at higher submodel	Global

- SINDA supports simultaneous creation of multiple nodes/conductors in single statement; this is only accomplished with DO loop in ESATAN
- SINDA allows SIV, DIV calls for temp-dependent capacitances or conductances; ESATAN must emulate with INTERP function

ESATAN vs. SINDA: Arrays

	Description	ESATAN	SINDA
ARRAY BLOCK	Singlet and Doublet Arrays	Y	Y
	Bivariate and Trivariate Arrays	Supported in $TABLES	Y
	Array Reference	String	Number
CARRAY BLOCK	Character Array Support	N	Y

- SINDA has array numbers, ESATAN has string names to identify arrays
- SINDA allows for singlet, doublet, bivariate, and trivariate array inputs, but array type not explicitly defined in code ➔ need to use array header format to determine array type
- ESATAN allows for singlets and doublets in $ARRAYS block and provides explicit array or matrix size definitions after the array name, uses $TABLES block for bivariates and trivariates
- MIXARRAY and character array formats not allowed in ESATAN (therefore not converted from SINDA)

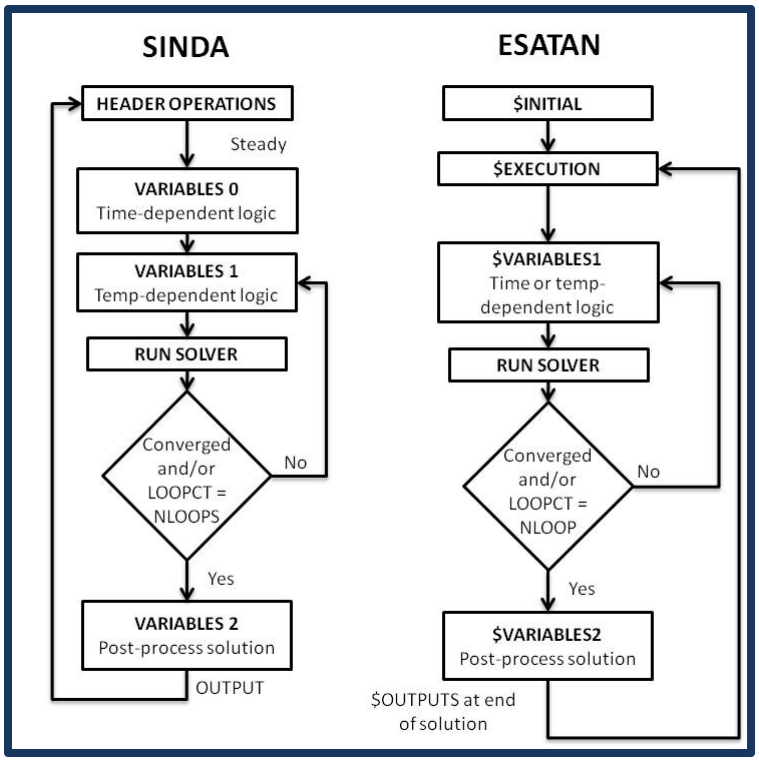
ESATAN vs. SINDA: Logic and Execution

Model FORTRAN Entry Point		Y	Y
Include additional files		$INCLUDE	INCLUDE, INSERT
Initialize Values		Y	N
Node and Conductor Variable scope		Only at higher submodel	Global
Q values reset to SOURCE DATA or zero at start of Timestep		N	Y
Instructions for start of Timestep (VARIABLES 0)		N	Y
Instructions for start of Iterations (VARIABLES 1)		Y	Y
Instructions for Post Convergence (VARIABLES 2)		Y	Y
Instructions at regular Output Intervals		Y	Y
User Subroutines		Y	Y
SOURCE BLOCK	Heat Source Reference	N/A	Node Number
VARIABLE BLOCKS	Global Variables	N	Y
	Submodel Specific Variables	Y	Y
	Submodel Specific Variable Reference	String	Number (K,XK)
	Submodel Specific Variables Globally Accessible	N	Y

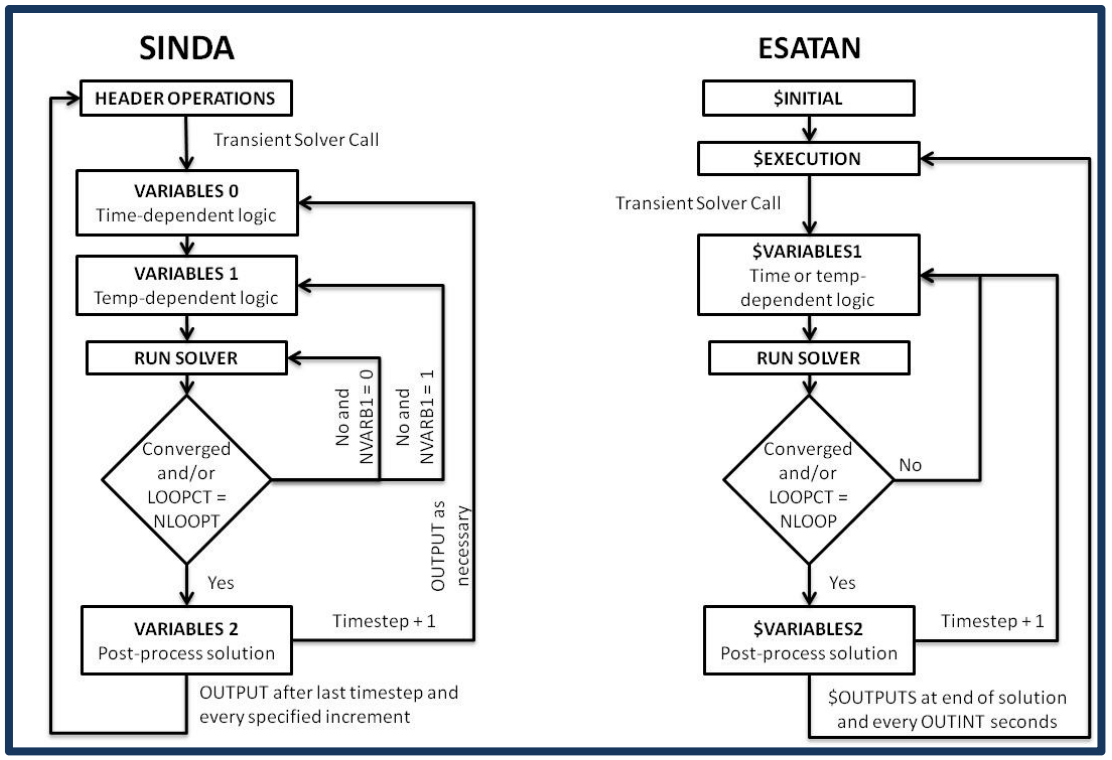
- ESATAN allows for $INITIAL block: initialization prior to execution
- SINDA has Variables 0 for time-dependent logic and Variables 1 for temp-dependent logic; ESATAN just has Variables 1
- Most solution routines and function calls have direct equivalents in ESATAN and SINDA

ESATAN vs. SINDA: Operations Flow

Steady-State Model



Transient Model



ESATAN to SINDA Conversion

- ESATAN submodel hierarchy must be removed: all submodels on same level (like SINDA)
- Arithmetic nodes defined as diffusion nodes with zero capacitance in ESATAN: when converting to SINDA, must define as diffusion nodes with fixed value, then cast as arithmetic in VARIABLES 1 (this prevents nodes with no capacitance from being defined in SINDA)
- "Inactive" nodes in ESATAN handled in SINDA by holding with HTRNOD, then commenting out all conductors to that node ➔ allows reference of nodal properties without impact on solution
  - Can't handle this dynamically within code execution
- Conductors from ESATAN placed in global submodel so that there are no accessibility issues in SINDA
- Arrays need to handle ESATAN "shorthand" array input, e.g. 5@1.0
- Wrapper functions provided in separate subroutine library to replicate ESATAN subroutines with SINDA subroutines

SINDA to ESATAN Conversion

- Specific submodel logic in SINDA moved to global submodel in ESATAN
- SINDA statements to generate multiple nodes or conductors (GEN, SIM) expanded before storing in collections and writing to ESATAN
- Array numbers in SINDA converted to string names in ESATAN
- References to conductor numbers in logic are converted so that in ESATAN, logic references node pair that defines conductor
- SOURCE DATA block transferred to $VARIABLES1, added as impressed heat to destination node
- VARIABLES 0 logic transferred to $VARIABLES1
- USER DATA variables (K, XK) transferred to appropriate submodel $INTEGER and $REAL blocks
- In SINDA, nodal Q variables are reset to zero at the start of VARIABLES 0. However, in ESATAN, the previous value from the last timestep or iteration persists upon entering VARIABLES 1 ➔ All nodal Q variables are reset to zero using the SETNDR function in ESATAN

Summary and Conclusions

- TSS/SINDA to ESARAD/ESATAN conversion tool developed by NASA Goddard Space Flight Center Thermal Engineering Branch
- GMM converter allows for conversion of radiation surfaces between TSS and ESARAD
- GMM converter emulates surface types that do not exist in both codes
- TMM converter allows for direct conversion of data blocks due to their structured nature
- TMM converter attempts to convert logic blocks by converting general functions and references
- Special considerations were made for specific ESATAN-to-SINDA and SINDA-to-ESATAN conversions, due to difference in architecture between both programs

Future Work

- SPV, DPV support
- A more robust bivariate/trivariate array detector
- Greater subroutine library with more wrapper functions
  (this is also modifiable by user)
- ESATAN intrinsic unit conversion support
- Adaptation of converter to fluid nodes
- Identification / conversion of specific control logic (like thermostatic control)

Acknowledgements

The presenters would like to thank

Steven Rickman

and the NASA Engineering & Safety Center (NESC) for their invaluable support

Questions?