# Lumped Mass Heat Transfer Experiment

## Thermal Network Solution with TNSolver

Bob Cochran
Applied Computational Heat Transfer
Seattle, WA
`TNSolver@heattransfer.org`

ME 331 Introduction to Heat Transfer
University of Washington
October 4, 2016

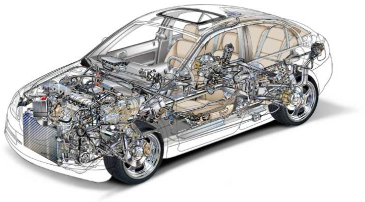
# Outline

- Heat Transfer Analysis
- Math Model
- The Thermal Network Solver: TNSolver
- Lumped Mass Heat Transfer Experiment

# Heat Transfer in Industry

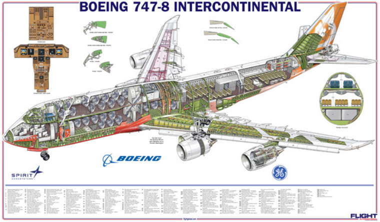
## Math Model

### Automotive

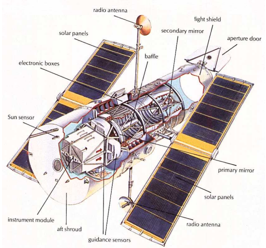


### Electronics Packaging

### Aircraft



### Aerospace



# Heat Transfer Analysis

## Math Model

Answering design questions about thermal energy and temperature

- Hand calculation - back-of-the-envelope
    - On the order of 1-10 equations
- Spreadsheet style
    - LibreOffice Calc, Microsoft Excel, MathCAD
- Thermal network or lumped parameter approach
    - On the order of 10-1,000 equations
- Continuum approach - solid model/mesh generation
    - On the order of 1,000-1,000,000 equations
    - Finite Volume Method (FVM)
    - Finite Element Method (FEM)

# Commercial Thermal Network Solvers

## Math Model

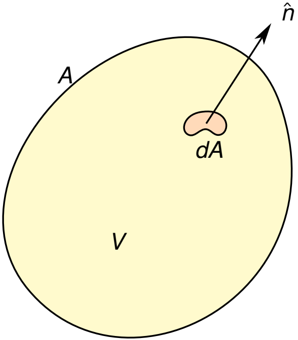
- C&R Technologies
  - SINDA/FLUINT, Thermal Desktop, RadCAD
- MSC Software
  - Sinda, SindaRad, Patran
- ESATAN-TMS
  - Thermal, Radiative, CADbench

# The Control Volume Concept

## Math Model

$$\sum \text{Energy In} - \sum \text{Energy Out} = \text{Energy Stored, Generated and/or Consumed}$$

Heat (transfer) is thermal energy transfer due to a temperature difference



# The Lumped Capacitance Method: Biot Number

## Math Model

The Biot number, $Bi$, is:

$$Bi = \frac{hL_c}{k} < 0.1 \qquad L_c = \frac{\text{volume}}{\text{surface area}} = \frac{V}{A}$$

where the characteristic length, $L_c$, is:

| Brick | Cylinder | Sphere |
|---|---|---|
| $\frac{HWL}{2(HW+LH+WL)}$ | $\frac{\pi r^2 L}{2\pi r^2+2\pi rL} = \frac{DL}{2(D+2L)}$ | $\frac{(4/3)\pi r^3}{4\pi r^2} = \frac{D}{6}$ |

# Convection Correlations

## Math Model

The heat flow rate is:

$$Q = hA(T_s - T_\infty)$$

where $h$ is the heat transfer coefficient, $T_s$ is the surface temperature and $T_\infty$ is the fluid temperature.
Correlations in terms of the Nusselt number are often used to determine $h$:

$$Nu = \frac{hL_c}{k} \qquad h = \frac{kNu}{L_c}$$

where $L_c$ is a characteristic length associated with the fluid flow geometry.

# Surface Radiation

## Math Model

Radiation exchange between a surface and *large* surroundings
The heat flow rate is (Equation (1.32), p. 32 in [LL16]):

$$Q = \sigma \epsilon_s A_s (T_s^4 - T_{sur}^4)$$

where $\sigma$ is the Stefan-Boltzmann constant, $\epsilon_s$ is the surface emissivity and $A_s$ is the area of the surface.
Note that the surface area, $A_s$, must be *much* smaller than the surrounding surface area, $A_{sur}$:

$$A_s \ll A_{sur}$$

Note that the temperatures must be the absolute temperature, $K$ or $^\circ R$

# Radiation Heat Transfer Coefficient

## Math Model

Define the radiation heat transfer coefficient, $h_r$ (see Equation (2.29), page 74 in [LL16]):

$$h_r = \epsilon\sigma(T_s + T_{sur})(T_s^2 + T_{sur}^2)$$

Then,

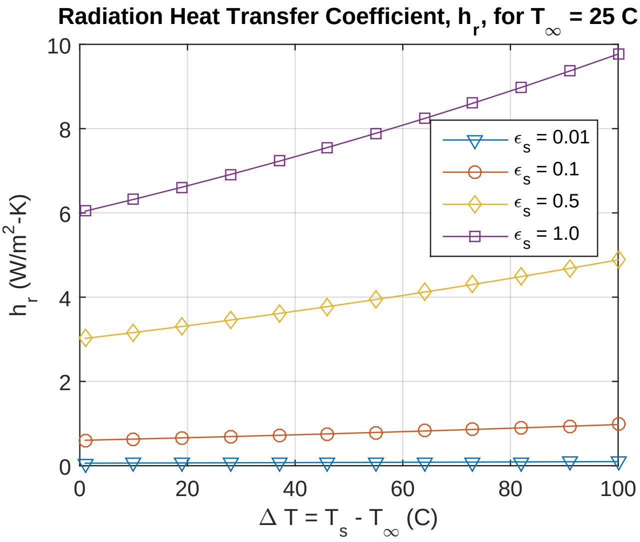
$$Q = h_r A_s (T_s - T_{sur})$$

Note:

- $h_r$ is temperature dependent
- $h_r$ can be used to compare the radiation to the convection heat transfer from a surface, $h$ (if $T_{sur}$ and $T_\infty$ have similar values)

# Range of Radiation Heat Transfer Coefficient

## Math Model



# Introducing TNSolver

- Thermal Network Solver - TNSolver
- MATLAB/Octave program
  - GNU Octave is an open source clone of MATLAB
- Thermal model is described in a text input file
  - Do not use a word processor, use a text editor, such as:
    - Cross-platform: vim/gvim, emacs, Bluefish, among many others
    - Windows: notepad, Notepad++
    - MacOS: TextEdit, Smultron
    - Linux: see cross-platform options
- Simulation results are both returned from the function and written to text output files for post-processing

# Thermal Network Terminology

## TNSolver User Guide

- Time dependency
  - Steady state or transient
  - Initial condition is required for transient
- Geometry
  - Control Volume - volume, $V = \int_V dV$
    - Node: ●, $T_{\text{node}} = \int_V T(x_i)dV$, finite volume
  - Control Volume Surface - area, $A = \int_A dA$
    - Surface Node: ○, $T_{\text{surface node}} = \int_A T(x_i)dA$, zero volume
- Material properties
- Conductors
  - Conduction
  - Convection
  - Radiation
- Boundary conditions
  - Boundary node: ▲
- Sources/sinks

# TNSolver Input Example of Text Input File

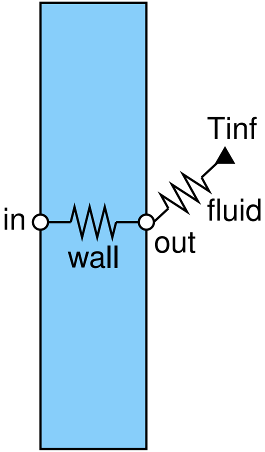
## TNSolver User Guide

```
!  Simple Wall Model

Begin Solution Parameters
  type  = steady
End Solution Parameters

Begin Conductors
 wall  conduction in  out   2.3  1.2   1.0 ! k L A
 fluid convection out Tinf  2.3  1.0       ! h A
End Conductors

Begin Boundary Conditions
  fixed_T  21.0   in    ! Inner wall T
  fixed_T  5.0    Tinf  ! Fluid T
End Boundary Conditions
```



! begins a comment (MATLAB uses %)

# TNSolver Input File

- What do we need in the input file for the lumped mass heat transfer experiment?
- Transient convection problem, with surface radiation
- Lumped capacitance approximation, $Bi < 0.1$, so no conduction in the solid object

# Solution Parameters

## TNSolver Input File

```
Begin Solution Parameters

  title = Lumped Mass Heat Transfer Experiment
  type = transient
  begin time  = (R)
  end time    = (R)
  time step   = (R)
  number of time steps = (I)

End Solution Parameters
```

(R) is a single real number
(I) is a single integer number

# Nodes

## TNSolver Input File

Define nodes which have a volume

```
Begin Nodes

! label    rho*c     V
    (S)       (R)      (R)

End Nodes
```

(S) is a single character string

# Convection Conductor

## TNSolver Input File

$$Q_{ij} = hA(T_s - T_\infty)$$

The heat transfer coefficient *h* is known.

```
Begin Conductors

! label   type    nd_i nd_j  parameters
   (S)  convection (S)  (S)  (R)  (R)  ! h,  A

End Conductors
```

# External Forced Convection (EFC) Conductor

## TNSolver Input File

$$Q_{ij} = hA(T_s - T_\infty)$$

Heat transfer coefficient, *h*, is evaluated using the correlation for external forced convection from a sphere with diameter *D* and fluid velocity of *u*.

```
Begin Conductors
!                 Ts   Tinf
! label   type    nd_i nd_j parameters
  (S)  EFCsphere (S)  (S)  (S) (R) (R) ! material, u, D

End Conductors
```

Note that Re, Nu and h are reported in the output file.

# External Natural Convection (ENC) Conductor

## TNSolver Input File

$$Q_{ij} = hA(T_s - T_\infty)$$

Heat transfer coefficient, h, is evaluated using the correlation for external natural convection from a sphere with diameter *D*.

```
Begin Conductors
!                    Ts   Tinf
! label   type       nd_i nd_j  parameters
   (S)  ENCsphere   (S)  (S)    (S)   (R)   ! material, D

End Conductors
```

Note that Ra, Nu and h are reported in the output file.

# Surface Radiation Conductor

## TNSolver Input File

$$Q_{ij} = \sigma \epsilon A_s (T_s^4 - T_{env}^4)$$

$\sigma$ is the Stefan-Boltzmann constant and $\epsilon$ is the surface emissivity.

```
Begin Conductors

! label type     nd_i nd_j parameters
   (S) surfrad  (S)   (S)   (R)   (R)   ! emissivity, A

End Conductors
```

Note that $h_r$ is reported in the output file.

# Boundary Conditions

## TNSolver Input File

Specify a fixed temperature boundary condition, $T_b$, to one or more nodes in the model.

```
Begin Boundary Conditions

!  type        Tb       Node(s)
  fixed_T      (R)      (S ...)

End Boundary Conditions
```

(S ...) one or more character strings

# Initial Conditions

## TNSolver Input File

Specify the initial temperatures, $T_0$, to the nodes in the model.

```
Begin Initial Conditions

! T0    Node(s)
  (R)   (S ...)

End Initial Conditions
```

# Example Input File

## TNSolver Input File

```
Begin Solution Parameters
  title = Lumped Capacitance Experiment - Object A
  type = transient
  begin time  = 0.0
  end time    = 341.5
  time step   = 0.5
!  number of time steps = 20
End Solution Parameters

Begin Nodes
  1  3925000.0  6.2892e-05  !  rho*c,  V
End Nodes

Begin Conductors
  conv convection 1 Tinf 12.0 0.0076  ! h, A
! conv EFCsphere  1 Tinf air  13.13 0.04934  ! material, u, D
! conv ENCsphere  1 Tinf air  0.04934 ! material, D
  rad  surfrad    1 Tinf 0.95 0.0076  ! emissivity, A
End Conductors
```

# Example Input File (continued)

## TNSolver Input File

```
Begin Boundary Conditions
  fixed_T  25.0  Tinf
End Boundary Conditions

Begin Initial Conditions
  99.0   1  !  Ti, node
End Initial Conditions
```

# Verification using Analytical Solution

## TNSolver Verification

Backward Euler time integration is used in TNSolver.
How does time step affect accuracy?
Utilitize the analytical solution Equation (1.22), p. 22 in [LL16]:

$$\frac{T - T_\infty}{T_i - T_\infty} = \exp\left[ - \left( \frac{hA}{\rho c V} \right) t \right]$$

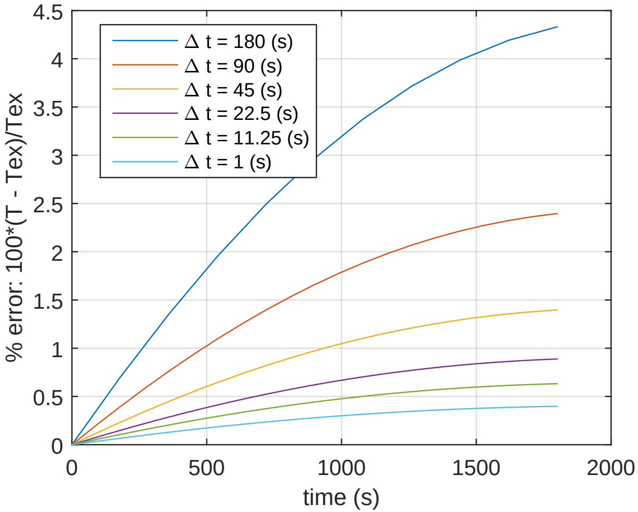
This is provided in the MATLAB function `lumpedmass.m`:

```
[T, Bi] = lumpedmass(time, rho, c, V, h, A, Ti, Tinf, k)
```

Example calculation using:
$D = 0.04931 \ m$, $T_i = 100 \ C$, $T_\infty = 25 \ C$
$\rho = 7850 \ kg/m^3$, $c = 500 \ J/kg \cdot K$
$h = 25.0 \ W/m^2 \cdot K$, $k = 62.0 \ W/m \cdot K$

# Verification using Analytical Solution

## TNSolver Verification



# Experiment Data Analysis with TNSolver

## Data Analysis

Three MATLAB functions are provided for a least-squares analysis using TNSolver.
Recommend placing the experimental data into a MATLAB `.mat` file using `save` in order to `load` the experimental specimen temperature `expT`.

1. Estimate convection heat transfer coefficient, *h*, for the natural convection data using `ls_lumped_h.m`
2. Estimate velocity, *u*, for the forced convection data using `ls_lumped_vel.m`
3. Estimate surface emissivity, $\epsilon$, using `ls_lumped_emiss.m`

# Estimate *h*

## Results

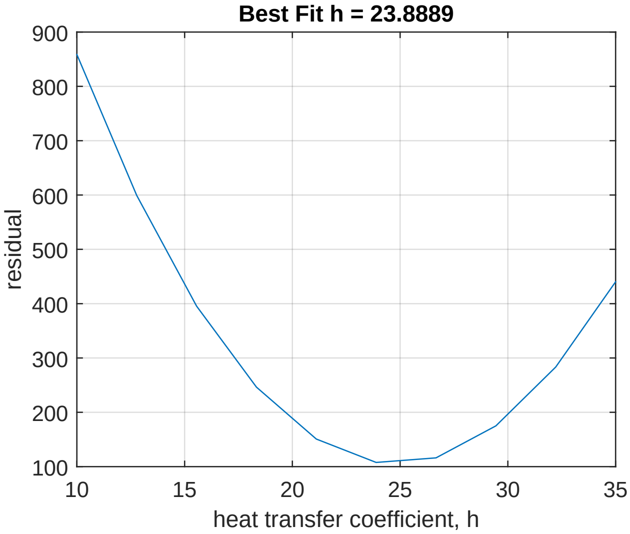
Example for object A, natural convection input file `ANC.inp`

1. Set begin and end time to match experimental data range
2. Set the time step to match experimental data sample rate
3. Set material properties and object geometries in input file
4. Set the boundary and initial conditions to match experiment
5. Use the `convection` conductor

```
>> load NC_A
>> h = linspace(10,35,10);
>> [besth] = ls_lumped_h(’ANC’, expT, h)
```

# Estimate *h* Results and Plot

## Results



# Estimate Velocity

## Results

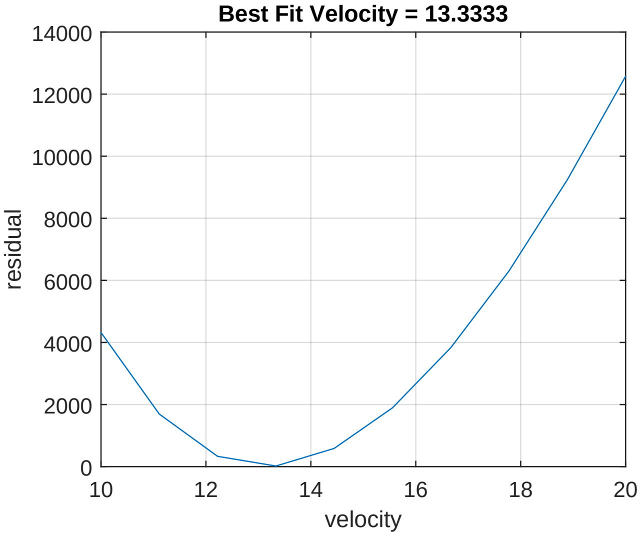
Example for object A, forced convection input file `AFC.inp`

1. Set begin and end time to match experimental data range
2. Set the time step to match experimental data sample rate
3. Set material properties and object geometries in input file
4. Set the boundary and initial conditions to match experiment
5. Use the `EFCsphere` convection conductor

```
>> load FC_A
>> u = linspace(10,20,10);
>> [bestvel] = ls_lumped_vel('AFC', expT, u)
```

# Estimate Velocity Results and Plot

## Results



# Estimate Emissivity

## Results

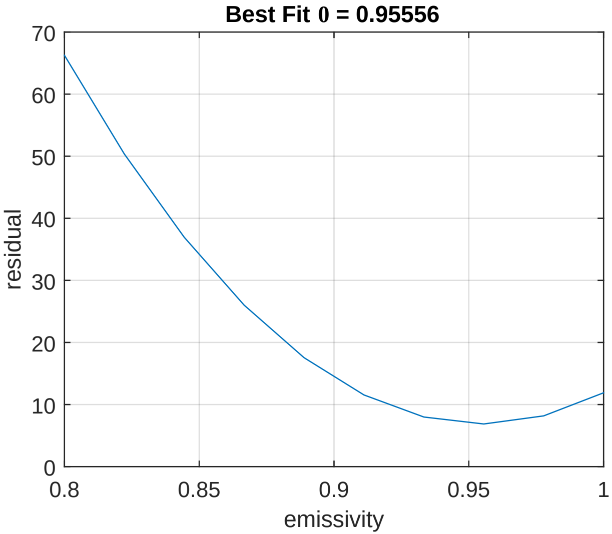
Example for object A, forced convection input file `AFC.inp`

1. Set begin and end time to match experimental data range
2. Set the time step to match experimental data sample rate
3. Set material properties and object geometries in input file
4. Set the boundary and initial conditions to match experiment
5. Use the `EFCsphere` convection conductor and the estimated velocity from the previous analysis

```
>> load FC_A
>> eps = linspace(.8,1,10);
>> [beste] = ls_lumped_emiss('AFC', expT, eps)
```

# Estimate Emissivity Results and Plot

## Results



# Conclusion

- Math model for lumped capacitance method
- TNSolver input file described
- TNSolver thermal network model verification with analytical solution demonstrated
- Lumped Mass Experiment data analysis

Questions?

# Appendix

# Obtaining GNU Octave

## GNU Octave

- GNU Octave
  - `http://www.gnu.org/software/octave/`
- Octave Wiki
  - `http://wiki.octave.org`
- Octave-Forge Packages (similar to MATLAB Toolbox packages)
  - `http://octave.sourceforge.net`
- Windows Installation
  - Binaries are at: `https://ftp.gnu.org/gnu/octave/windows/`
  - As of August 1, 2016, the latest version of Octave is 4.0.3
  - Download the octave-4.0.3.zip file and unzip in a Windows folder

# External Forced Convection over a Sphere

## Math Model

Equation (7.48), p. 444, in [BLID11]

$$\overline{Nu}_D = 2 + \left(0.4 Re_D^{1/2} + 0.06 Re_D^{2/3}\right) Pr^{0.4} \left(\frac{\mu}{\mu_s}\right)^{1/4}$$

where $D$ is the diameter of the sphere and the Reynolds number, $Re_D$, is:.

$$Re_D = \frac{\rho u D}{\mu} = \frac{uD}{\nu}$$

Note that the fluid properties are evaluated at the fluid temperature, $T_\infty$, except the viscosity, $\mu_s$, evaluated at the surface temperature, $T_s$.

# External Natural Convection over a Sphere

## Math Model

Equation (9.35), page 585 in [BLID11]

$$\overline{Nu}_D = 2 + \frac{0.589 Ra_D^{1/4}}{\left[1 + (0.469/Pr)^{9/16}\right]^{4/9}}$$

where $D$ is the diameter of the sphere and the the Rayleigh number, $Ra_D$, is:.

$$Ra_D = GrPr = \frac{g\rho^2 c\beta D^3 \left(T_s - T_\infty\right)}{k\mu} = \frac{g\beta D^3 \left(T_s - T_\infty\right)}{\nu\alpha}$$

Note that the fluid properties are evaluated at the film temperature, $T_f$:

$$T_f = \frac{T_s + T_\infty}{2}$$

# References I

[BLID11] T.L. Bergman, A.S. Lavine, F.P. Incropera, and D.P. DeWitt.
*Introduction to Heat Transfer*.
John Wiley & Sons, New York, sixth edition, 2011.

[LL16] J. H. Lienhard, IV and J. H. Lienhard, V.
*A Heat Transfer Textbook*.
Phlogiston Press, Cambridge, Massachusetts, fourth edition, 2016.
Available at: `http://ahtt.mit.edu`.