Cantera Integration with the Toolbox for Modeling and Analysis of Thermodynamic Systems (T-MATS)

Thomas M. Lavelle

NASA GRC

Jeffryes W. Chapman

Vantage Partners

Ryan D. May

Vantage Partners

Jonathan S. Litt and Ten-Huei Guo

NASA GRC

Team

- Thomas M. Lavelle

  NASA Glenn Research Center, Cleveland, OH 44135

- Jeffryes W. Chapman

  Vantage Partners, LLC. Cleveland, OH 44135

- Ryan D. May

  Vantage Partners, LLC. Cleveland, OH 44135

- Jonathan S. Litt

  NASA Glenn Research Center, Cleveland, OH 44135

- Ten-Huei Guo

  NASA Glenn Research Center, Cleveland, OH 44135

Goal

- Increase flexibility of T-MATS

- Cantera increases flexibility of thermodynamics
  - Can model any flow

- M-file elements allow users to prototype engineering elements

- Slower than standard T-MATS

T-MATS

- Simulink code
- Library of thermodynamic elements
  - Standard library includes elements typical of aeropropulsion
- Newton Raphson solver
- Default thermodynamic table is air, water, and a hydrocarbon fuel
- Systems can be modeled outside the standard elements/thermo
  - Create new thermo tables
  - Create elements

Cantera

- Object-oriented software tools for problems involving chemical kinetics, thermodynamics, and/or transport properties
- C++ based code with interfaces for python, matlab, C, and fortran 90
- https://code.google.com/p/cantera/

Integration of T-MATS with Cantera

- Allows any fluid combination to be modeled
- Specify the thermodynamics of the possible products
  - Similar to CEA thermo.inp file
- Requires specification of all “reactants” for the simulation
  - Similar to CEA reactant cards
  - Specify the different possible starting flows by composition

Species = { .7547 .232 .0128 0 0 0;
1 0 0 0 0 0;
.922189 .077811 0 0 0 0 0;
0 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 0 0};

Name = { 'N2' 'O2' 'AR' '' '' '';
'H2O' '' '' '' '' ''; ‘CH2' ‘CH' '' '' '' '';
'' '' '' '' '' ; '' '' '' '' '' ; '' '' '' '' '' '' }

- Species and Name arrays need to be defined
- A model with this definition can run with mixtures of Air, Water, and JP-7
- Allows for models of aircraft engines with humidity

T-MATS Cantera Fluid Arrays

Information	**Index**	**Description**
W	1	Weight of the flow
Tt	2	Total temperature
Pt	3	Total pressure
ht	4	Total enthalpy
comp1 (to) comp10	5-14	Percentage of flow composition for reactants 1 to 10
s	15	Entropy
rhot	16	Total density
Ts	17	Static temperature
Ps	18	Static pressure
hs	19	Static enthalpy
rhos	20	Static density
Vflow	21	Flow velocity
MN	22	Flow Mach number
A	23	Flow area
gamt	24	Total gamma
gams	25	Static gamma

- Each fluid location in a thermodynamic model is represented by an array that contains all the fluid properties at a given location

T-MATS Cantera Fluid Functions

Function	**Description**
`add(`*`flow1`*`,`*`flow2`*`)`	Add add *flow1* and *flow2* together, conserving enthalpy and mass
`copyFlow(`*`flow`*`)`	Copy the information from *flow* to another flow
`getMassFraction(`*`flow`*`,c)`	Return the mass fraction of compound `c` in the object *flow*
`set_hP(`*`flow`*`,ht,Pt)`	Set the total conditions based on *flow*, total enthalpy and total pressure
`set_MN1(`*`flow`*`)`	Set the static conditions to sonic based on flow conditions
`set_MNPs(`*`flow`*`,Ps)`	Set the static conditions based on *flow* and input static pressure
`set_SP(`*`flow`*`,S,Pt)`	Set the total conditions based on *flow* and input entropy and total pressure
`set_TP(`*`flow`*`,Tt,Pt)`	Set the total conditions based on flow, total temperature and total pressure
`set_TsPsMN(`*`flow`*`,Ts,Ps,MN)`	Set the conditions based on *flow*, static temperature, static pressure and Mach

- All communication between Cantera and T-MATS is handled by these functions
- Functions return a new Cantera Fluid Array based on inputs (see previous slide)

T-MATS Element Files

- Library of standard elements released in Simulink m-file format
- Allows for development and prototyping
- Elements are interpreted
  - No need to compile
- Engineers can quickly create new elements
- Block sets are released with T-MATS Cantera package

Instance Information

- Needed a way to store instance information from one pass to another
- Created to functions to store and retrieve information from one pass to another
  - Variables are stored in the MATLAB workspace with the object instance name attached to the variable instance name
- setV sets the value or a variable in the workspace
- getV gets the value of a variable from the workspace

```
path = stripchar( gcb() );
setV( 's_C_Nc', path, s_C_Nc );
s_C_Nc = getV( 's_C_Nc', path );
```

Some Examples from Compressor Element

- Setting the exit conditions

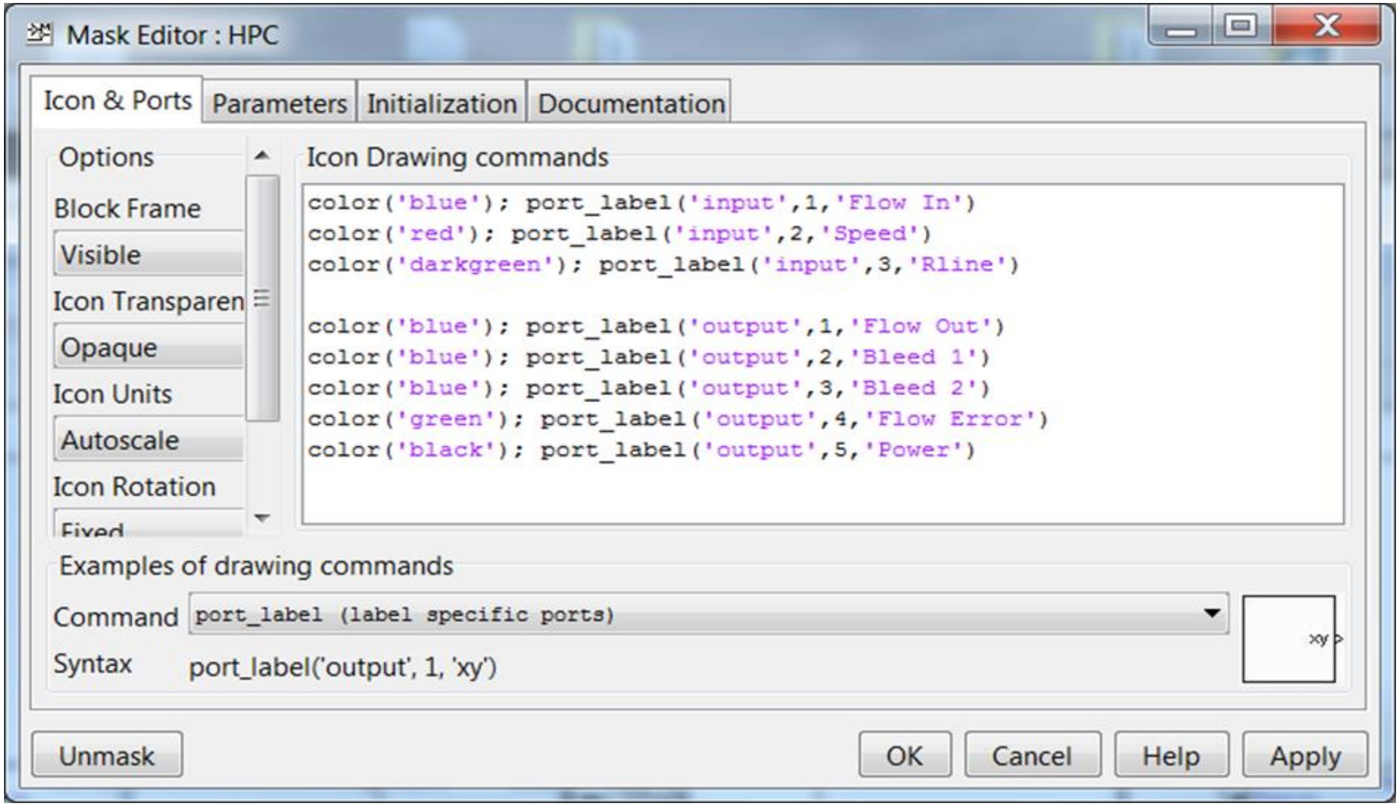
```
FOideal = set_SP( FI, FI(s),PtOut );
htOut = FI(ht) + ( FOideal(ht) - FI(ht) )/eff;
% set the exit conditions to known enthalpy and
%pressure
FO = set_hP( FI, htOut, PtOut );
```

Some Examples from Compressor Element

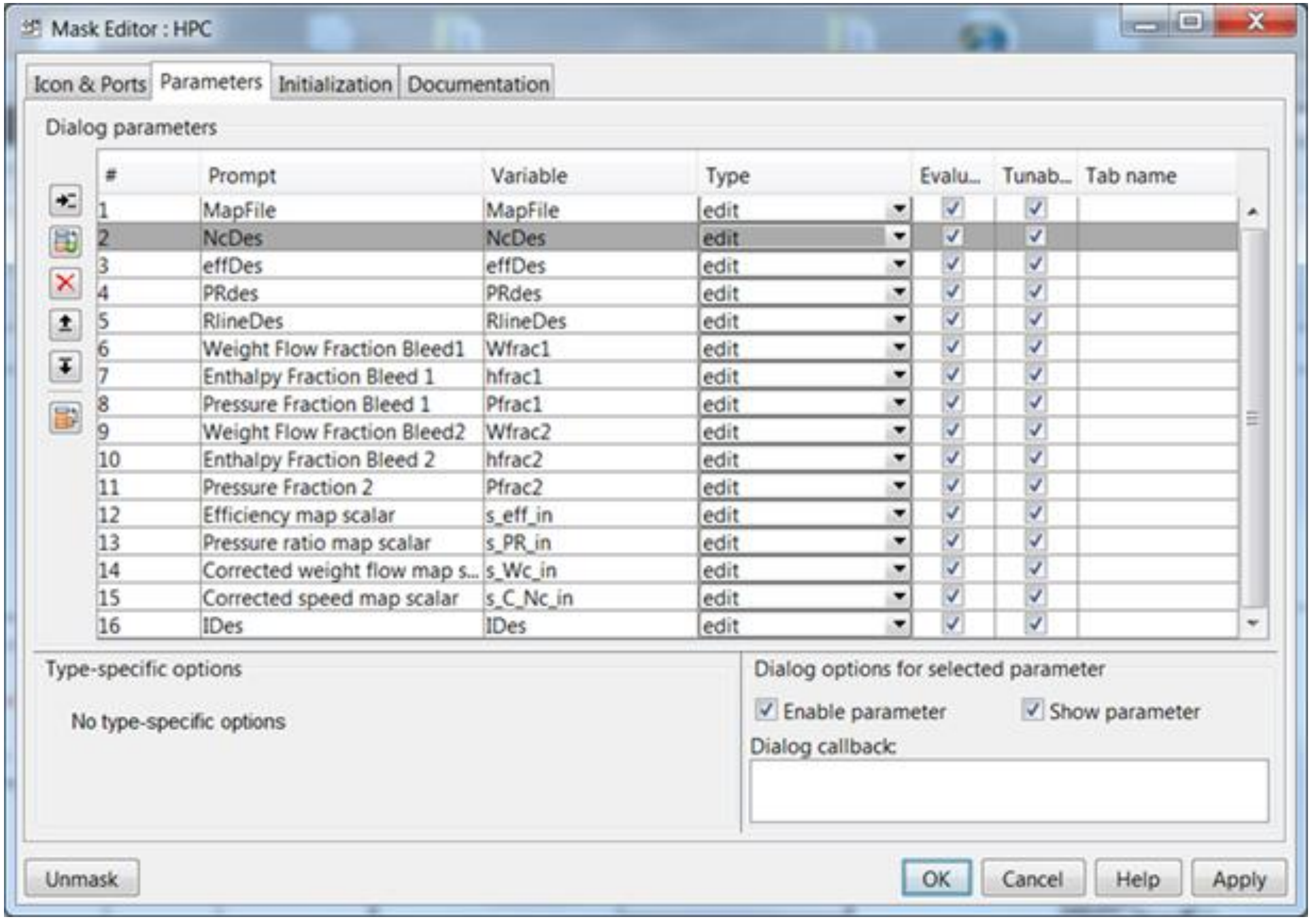
- Design Point Scaling

```
if IDes < .5
   s_eff = effDes / effMap;
   s_PR = ( PRdes - 1 )/( PRmap - 1 );
   s_Wc = WcIn/ WcMap;
   setV( 's_eff', path, s_eff );
   setV( 's_Wc', path, s_Wc );
   setV( 's_PR', path, s_PR );
elseif IDes < 1.5
   % get the maps scalars from the workspace
   s_eff= getV( 's_eff', path );
   s_Wc= getV( 's_Wc', path );
   s_PR= getV( 's_PR', path );
else
   % use the input values
   s_eff = s_eff_in;
   s_Wc = s_Wc_in;
   s_PR = s_PR_in;
end
```

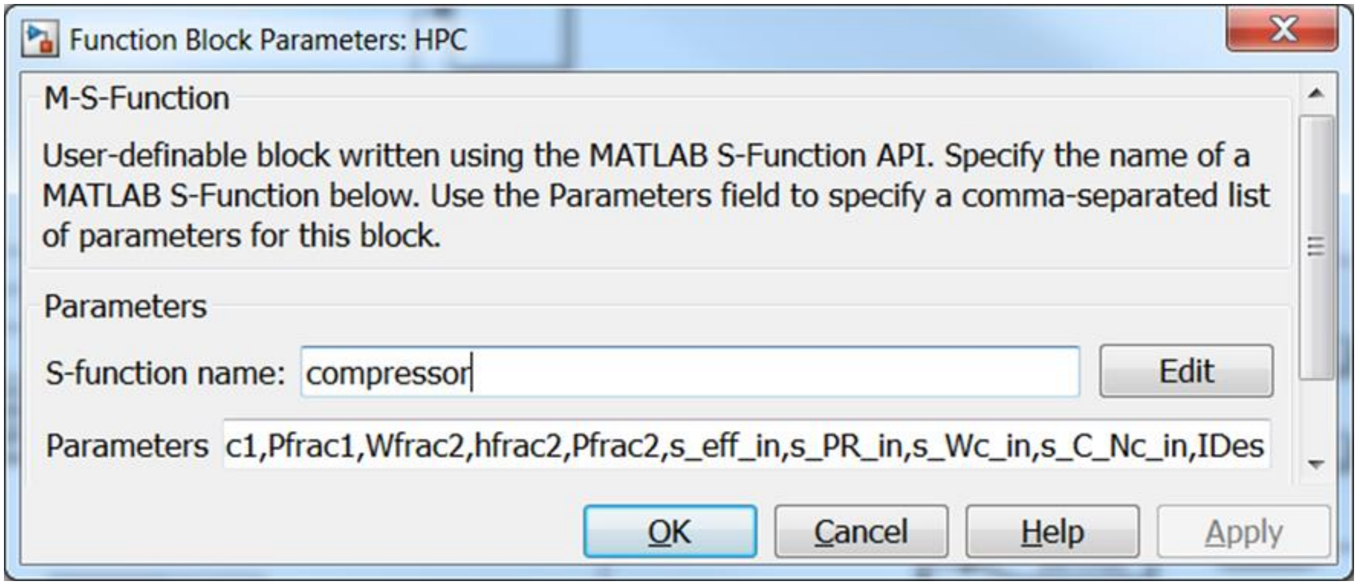
Simulink Objects



- Mask appearance
- Describes port labels and colors
- Label colors are standard based on T-MATS style

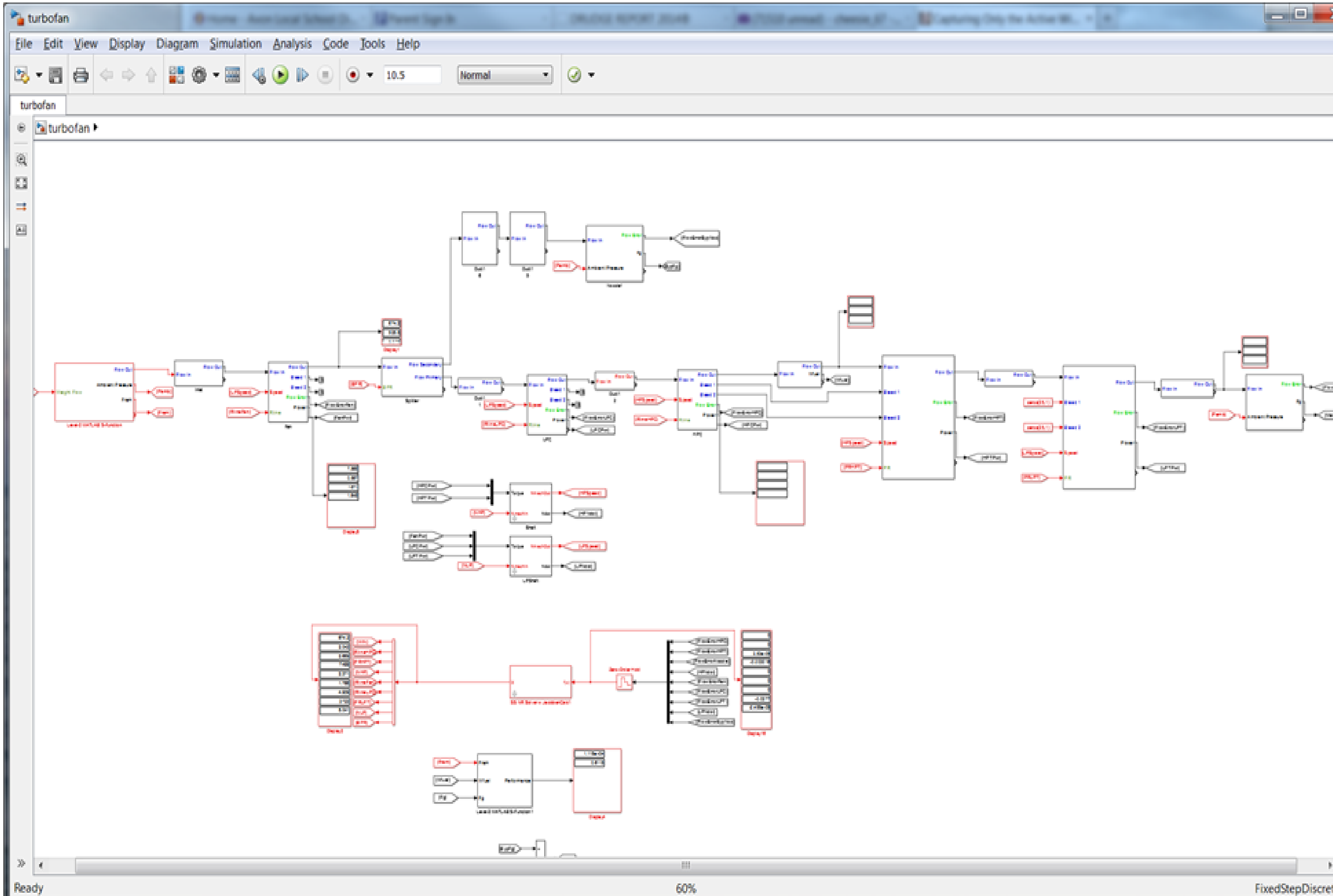


- Parameter list
- Lists the variables that can be input by the user to the dialog box



- S-function block parameters
- Utilizes m-file to create S-function
- Maps parameter dialog box to m-file

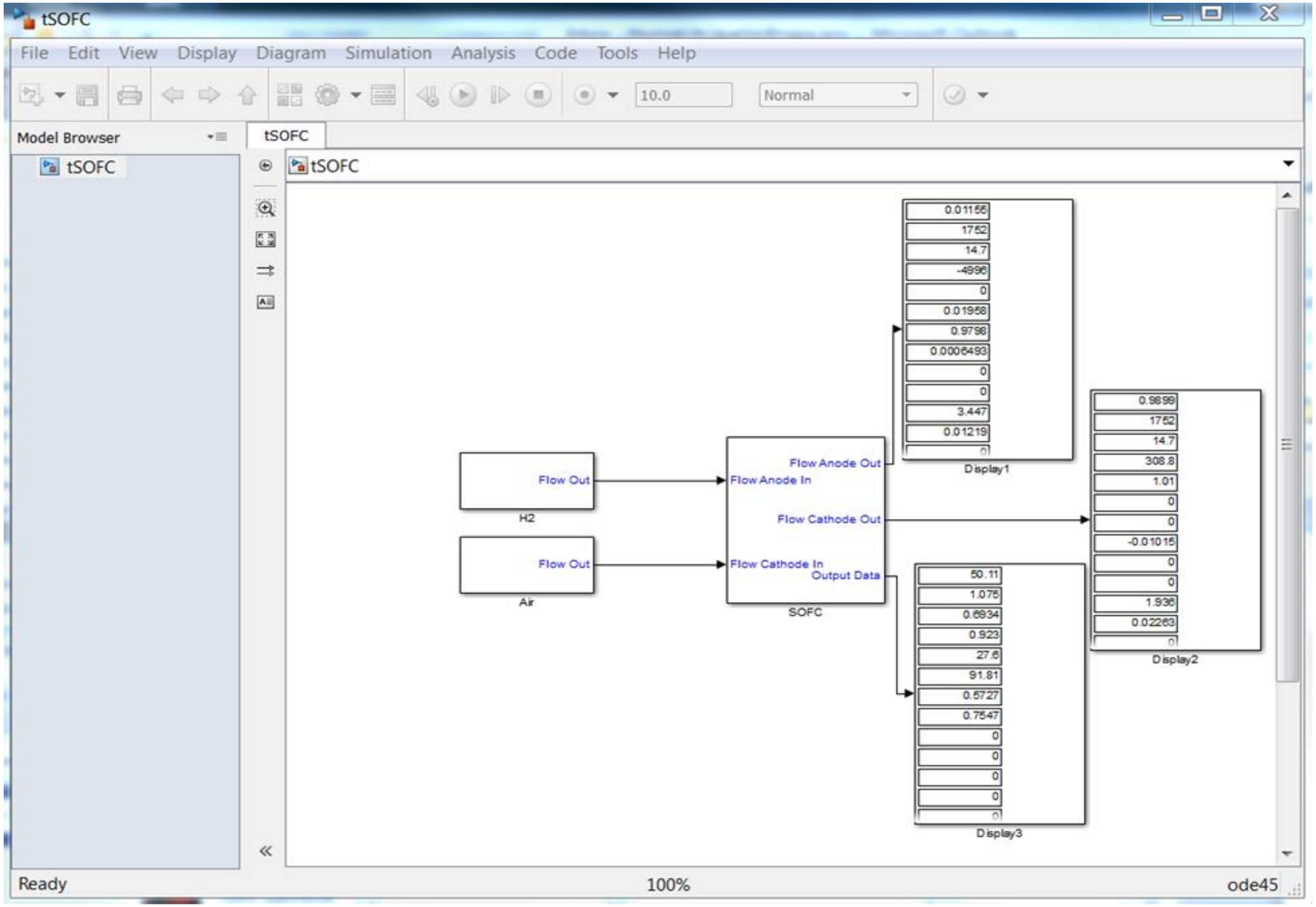
Turbofan Model –JT9D



Turbofan Model –JT9D

	NPSS with JANAF Output	**TMATS Cantera Output**
Altitude	34000 ft	34000 ft
Mach number	.8	.8
Weight flow	674 lbm/sec	674 lbm/sec
Thrust	11194 lbf	11182 lbf
SFC	.6113	.6116

Fuel Cell Model



- Reactants are Air, H2, O2, and H20

Fuel Cell Model

Specifying the reactants:

```
Species = {  .7547  .232 .0128 0 0 0;
              1 0 0 0 0 0;
              1 0 0 0 0 0;
              1 0 0 0 0 0; 0 0 0 0 0 0; 0 0 0 0 0 0};
Name = { 'N2' 'O2' 'AR' '' '' '';
           'H2' '' '' '' '' ''; ‘O2' '' '' '' '' '';
           ‘H2O' '' '' '' '' '‘; '' '' '' '' '' '‘; '' '' '' '' '' '' }
```

Getting the mass fractions of an element:

```
xN2_cOut = getMassFraction( Fl_O2, 'N2' );
xO2_cOut = getMassFraction( Fl_O2, 'O2' );
```

Fuel Cell Model

Removing oxygen from the flow:

```
xO2_Cathode1 = getMassFraction( Fl_Cathode1, 'O2' )

%Composition as mass flow (g/sec)
wO2_Cathode1 = xO2_Cathode1 * w_Cathode1

%Composition as molar flow rate (mol/sec)
M_O2_Cathode1 = wO2_Cathode1 / 32.

%Calculates composition after electrochemistry...
M_O2_Cathode2 = M_O2_Cathode1 - ((M_H2_Anode1 / 2.0) * pctH2util);

M_O2_Cathode2 = M_O2_Cathode1 - ((M_H2_Anode1 / 2.0) * pctH2util);
wO2_lost = (M_O2_Cathode1 - M_O2_Cathode2)*32. * 0.002205 % lb/sec

Fl_tempO2(8) = 1;
Fl_tempO2(W) = -wO2_lost;
Fl_tempO2= set_TP( Fl_tempO2, Fl_Cathode2(Tt), Fl_Cathode2(Pt) );
```

Conclusion

- Cantera has been integrated with T-MATS
  - Capable of modeling any thermodynamic flow
- Simulink block sets and MATLAB m-files
  - Allows for prototyping
- Greatly increases the flexibility of T-MATS
- Slower than standard T-MATS

- Download information may be found at: https://github.com/nasa/T-MATS/releases/