CEA TUTORIAL

Introduction

Infinite vs Finite Area Combustor

Examples

Thermo.inp format

Adding new propellants to thermo library

```
% CEA_ROCKET_EXAMPLE: Example file for the MATLAB CEA wrapper. For in-depth
% documentation read the headers of cea_rocket_run.m,
% cea_rocket_run_single.m, and cea_rocket_read.m

% Change this variable to true to rerun CEA instead of using saved values
CEA_RUN = true;
CEA_SAVE_FILE = 'Example1.mat';

% The CEA MATLAB code takes a MATLAB map (called a dictionary in Python or
% hash in C) as input. The dictionary uses MATLAB character arrays as the
% keys, and the value data type varies by which key is used. Details of
% each key are listed in cea_rocket_run.m
% For example: inp('key') = value.
inp = containers.Map;
inp('type') = 'eq';              % Sets the type of CEA calculation
inp('p') = 53.3172;                % Chamber pressure
inp('p_unit') = 'bar';              % Chamber pressure units
inp('o/f') = 5.55157;               % Mixture ratio
inp('sub') = 1.58;               % Subsonic area ratio/contraction ratio
inp('sup') = [1,25,50,75];                % Supersonic area ratios
% inp('pip') = [10,100,1000,1500, 2000, 2500, 3000];  % Pressure ratios
inp('fuel') = 'H2(L)';              % Fuel name from thermo.inp
inp('fuel_t') = 20.27;                 % Fuel inlet temperature
inp('fuel_wt%') = 100;             % Fuel name from thermo.inp
inp('ox') = 'O2(L)';               % Ox name from thermo.inp
inp('ox_t') = 90.17;                   % Ox inlet temperature
inp('ox_wt%') = 100;             % Fuel name from thermo.inp
inp('file_name') = 'Example1_.inp';    % Input/output file name
```

CAGLAR YILMAZ

January 27th, 2021

Introduction

- CEA MATLAB wrapper provided has some limitations when compared with online CEA.
  - Can not use pressure ratio together with subsonic or/and supersonic area ratios
  - Does not parse the throat values by default. A throat subsonic or supersonic area ratio has to be specified explicitly
  - The molar concentrations are not parsed for frozen flow
  - Nfz option is not defined. Used for frozen flow calculations. Default is at combustion chamber (nfz = 1).
  - Some others
- Read header sections of “cea_rocket_read.m” and “cea_rocket_run.m”. You may get answers to many of your questions
- I will go through several examples from user manual to demonstrate how to use CEA MATLAB wrapper.
- Always check your input and output files if you are not sure about things
- I will also talk about how to define a new reactant → Editing thermo.inp or defining manually in the case input file

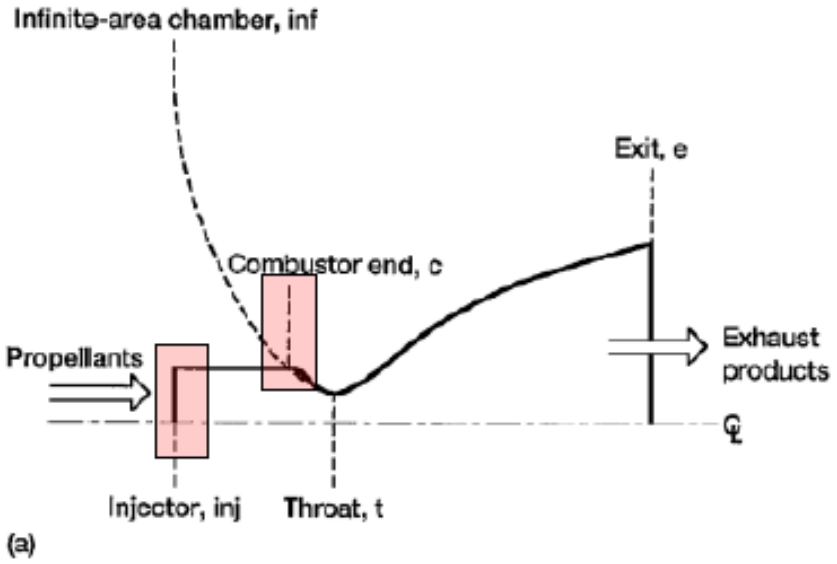
Infinite vs Finite Area Combustor

INFINITE-AREA COMBUSTOR

- All points are isentropic with respect to the combustion point.
- Can assume chemical equilibrium or products frozen after a specified point.
- Freeze points selections are combustion, throat, or exit1.

FINITE-AREA COMBUSTOR

- In this chamber combustion is a non-isentropic, irreversible process.
- During the burning process, part of the energy released is used to raise the entropy, and the pressure drops.
- Expansion in the nozzle is assumed to be isentropic.
- Can only assume chemical equilibrium.



Examples

```
EXAMPLE 8:
    (a)   Rocket problem with infinite-area combustor (rocket iac by default).
    (b)   The fuel is H2(L) at 20.27 K; the oxidant is O2(L) at 90.17 K.
          Both are in thermo.lib so that the enthalpies and "exploded" formulas
          do not need to be given.
    (c)   The oxidant-to-fuel ratio is 5.55157 (o/f=5.55157).
    (d)   The chamber pressure is 53.3172 bars (p,bar=53.3172).
    (e)   Calculations are with equilibrium chemistry only (equilibrium).
    (f)   For exit points there are three pressure ratios (pi/p=10,100,1000),
          one subsonic area ratio (subar=1.58), and three supersonic area
          ratios (supar=25,50,75).
```

CEA Input:

```
          problem  rocket  equilibrium  o/f=5.55157
 case=8   p,bar=53.3172 subar=1.58,pi/p=10,100,1000,supar=25,50,75
          reactants
 fuel = H2(L)   wt% 100.    t(k) 20.27
 oxid = O2(L)   wt% 100.    t(k) 90.17
          output  siunits
          end
```

Example 8 illustrates a typical rocket performance problem based on the model of an infinite-area combustor, `iac`. Note that there are nine output points (columns): the chamber, the throat, three pressure ratios, one subsonic area ratio, and three supersonic area ratios. Since NCOL (number of columns or points) was set to 8 in the program, output for the last supersonic area ratio was printed on the second page along with the chamber and throat, which are repeated for convenience.

Examples

```
EXAMPLE 9:
   (a)  Rocket problem with a finite-area combustor (rocket  fac).
   (b)  Contraction ratio of 1.58 (acat=1.58) is assigned.
   (c)  Fuel, oxidant, and the remaining parameters are the same as in
        example 8.
```

CEA Input:

```
problem  o/f=5.55157 case=9  rocket  fac  p,bar=53.3172 acat=1.58
         pi/p=10,100,1000, supar=25,50,75

reac  fuel = H2(L)  wt%=100.  t,k= 20.27
      oxid = O2(L)  wt%=100.  t,k= 90.17

  output  siunits
  end
```

```
EXAMPLE 10:
   (a)  Rocket problem with a finite-area combustor (rocket fac).
   (b)  A ratio of mass flow rate to chamber area of 1333.9 (ma=1333.9)
        is assigned.  This value was calculated from the results
        of example 9 where a contraction ratio of 1.58 was assigned.
   (c)  Fuel, oxidant, and the remaining parameters are the same as in
        examples 8 and 9.
```

CEA Input:

```
reac  fuel = H2(L)  t,k= 20.27
      oxid = O2(L)  t,k= 90.17
problem  o/f=5.55157 case=10  rocket  fac  p,bar=53.3172 ma=1333.9
         pi/p=10,100,1000, sup-ae/at=25,50,75
output  short
end

WARNING!!  AMOUNT MISSING FOR REACTANT  1.
PROGRAM SETS WEIGHT PERCENT = 100. (REACT)

WARNING!!  AMOUNT MISSING FOR REACTANT  2.
PROGRAM SETS WEIGHT PERCENT = 100. (REACT)
```

For one oxidizer and fuel, specifying weight percentage is optional. CEA assumes 1. (100%)

Frozen flow does not work with a finite area combustor. This is a CEA limitation
If you ask for frozen data, you will get an error.

Examples

```
EXAMPLE 13:
   (a)  Rocket problem with an infinite-area combustor (rocket). This
        problem was selected to show some unusual derivatives.
   (b)  Tripropellant.  Fuels are N2H4(L) and Be(L) and oxidant is H2O2(L),
        all at 298.15 K.
   (c)  Reactant mixture is given as 67% fuel by weight (%fuel=67.).
   (d)  Chamber pressure is 3000 psia (p,psia=3000).
   (e)  Calculations are to be for equilibrium conditions only (equilibrium).
   (f)  Four exit pressure ratios are assigned (pi/p=3,10,30,300).
   (g)  BeO(L) is included as possible combustion product for the first
        point (insert).
   (h)  Mole fractions > 1.e-10 are to be in e-format (trace=1.e-10).
   (i)  Units in final tables to be non-SI (calories).
```

Define O/F instead of this in MATLAB wrapper

O/F = 33/67

CEA Input:

```
reac fuel = N2H4(L)     wt%= 80    t=298.15
     fuel = Be(a)       wt%= 20    t=298.15
     oxid = H2O2(L)     wt%=100    t=298.15
prob  rocket  case=13  p,psia=3000, pi/p=3,10,30,300,equilibrium   %fuel = 67.

outp  trace= 1.e-10 calories
insert  BeO(L)
end
```

For one oxidizer and fuel, specifying weight percentage is optional. CEA assumes 1. (100%)

Another multi fuel/ox example is in the CEA folder “cea_high_isp”

Thermo.inp format

Format for gaseous and condensed species with data extending over T range

```
CL2                Chlorine gas. TPIS 1989, v1, pt2, p88.
 2 tpis89 CL  2.00    0.00     0.00     0.00     0.00 0        70.90540          0.000
   200.000  1000.000 7 -2.0 -1.0  0.0  1.0  2.0  3.0  4.0  0.0          9181.110
 3.46281724D+04 -5.54712949D+02  6.20759103D+00 -2.98963673D-03  3.17303416D-06
-1.79363467D-09  4.26005863D-13  0.00000000D+00  1.53407075D+03 -9.43835303D+00
   1000.000  6000.000 7 -2.0 -1.0  0.0  1.0  2.0  3.0  4.0  0.0          9181.110
 6.09256675D+06 -1.94962688D+04  2.85453491D+01 -1.44996828D-02  4.46388943D-06
-6.35852403D-10  3.32735931D-14  0.00000000D+00  1.21211722D+05 -1.69077832D+02
```

Empirical equations for example A.1:

Heat capacity: $\frac{C_p^o}{R} = a_1 T^{-2} + a_2 T^{-1} + a_3 + a_4 T + a_5 T^2 + a_6 T^3 + a_7 T^4$

Enthalpy: $\frac{H^o(T)}{RT} = -a_1 T^{-2} + a_2 T^{-1} \ln T + a_3 + a_4 \frac{T}{2} + a_5 \frac{T^2}{3} + a_6 \frac{T^3}{4} + a_7 \frac{T^4}{5} + \frac{b_1}{T}$

Entropy: $\frac{S^o(T)}{R} = -a_1 \frac{T^{-2}}{2} - a_2 T^{-1} + a_3 \ln T + a_4 T + a_5 \frac{T^2}{2} + a_6 \frac{T^3}{3} + a_7 \frac{T^4}{4} + b_2$

TABLE A.1.—GENERAL FORMAT FOR NINE-CONSTANT FUNCTIONAL FORM

Record	Constants	Format	Columns
1	Species name or formula	A24	1–24
	Comments (data source)	A56	25–80
2	Number of T intervals	I2	2
	Optional identification code	A6	4–9
	Chemical formulas, symbols, and numbers	5(A2,F6.2)	11–50
	Zero for gas and nonzero for condensed phases	I1	52
	Molecular weight	F13.5	53–65
	Heat of formation at 298.15 K, J mol	F13.5	66–80
3	Temperature range	2F10.3	2–21
	Number of coefficients for $C_p^\circ R$	I1	23
	T exponents in empirical equation for $C_p^\circ R$	8F5.1	24–63
	$\{H^\circ(298.15) - H^\circ(0)\}$, J mol	F15.3	66–80
4	First five coefficients for $C_p^\circ R$	5D16.8	1–80
5	Last three coefficients for $C_p^\circ R$	3D16.8	1–48
	Integration constants b_1 and b_2	2D16.8	49–80
----	Repeat 3, 4, and 5 for each interval	----------	----------

Format for condensed species with data at only one T

```
C2H2(L),acetyle   Acetylene. JANAF Prop.Ser.E,1/67. TRC a-3000,10/86.
 0 1 3/95 C   2.00H   2.00    0.00     0.00     0.00 1     26.03788      207599.000
    192.35
```

You can also define new reactants without editing thermo.lib directly in the case input file. Not possible with the MATLAB wrapper

Examples

- ADN based monopropellant example :
  - Ammonium dinitramide, ADN, a green substitute for both ammonium perchlorate, AP, and for monopropellant hydrazine
  - Propellant is a mixture of ammonia, methanol, water and ADN

```
problem
      rocket  equilibrium  frozen  nfz=1
  p,psi=200,
  sup,ae/at=150

reac

fuel = ADN h,kj/mol=-134.6 H 4 N 4 O 4 wt%=63 t(k)=298.15
fuel = H2O(L) wt%=13.95 t(k)=298.15
fuel = CH3OH(L) wt%=18.4 t(k)=298.15
fuel = NH3(L) wt%=4.65 t(k)=239.72

output siunits
end
```

$NH_4N(NO_2)_2$

Adding new propellants to thermo library

- To add your own component to thermo.lib, do following:
  - Delete existing thermo.out and thermo_lib files
  - Add your reactant to thermo.inp using explained format and save. Unit for enthalpy is kJ/kmole (or J/mole)
  - Run FCEA2.exe with thermo (without inp)

```
 ENTER INPUT FILE NAME WITHOUT .inp EXTENSION.
   THE OUTPUT FILES FOR LISTING AND PLOTTING WILL HAVE
 THE SAME NAME WITH EXTENSIONS .out AND .plt RESPECTIVELY


thermo
```

  - It will create new thermo.out and thermo_lib
  - Thermo_lib is in binary format. You can not open/edit it with Notepad or similar software
  - Check thermo.out for your reactant and its enthalpy (check if changes took place (enthalpy of formation/assigned enthalpy). You can also look at your case output file for that)