To: Dr. Halil Berberoglu

From: Colin Murphy, Rachel Purvis, Andrew Sheu, Bryan Stockberger, Zach Straw

Date: 2/23/2011

Subject: Ideal Gas Mixture Property Calculator

Problem description and objectives:

The objective of this project is to create a MATLAB program that will calculate thermodynamic properties for an ideal gas mixture that has differing molar compositions of gases. The thermodynamic properties of each constituent gas needed are molecular mass (M), specific gas constant (R), specific heats (cv and cp), ratio of specific heats (k), internal energy (u), enthalpy (h), and entropy (s). These properties are calculated based on the user input for molar composition, pressure, and temperature. The MATLAB program will take these user inputs and calculate properties for each constituent gas. Upon finding properties of the gases, the MATLAB program will validate and apply these properties to an actual ideal gas mixture problem. This property calculator will be able to efficiently calculate properties to achieve accurate power ratings for a turbine.

Procedure:

Firstly, the MATLAB program asks for the temperature[°C], pressure [kPA], and molar compositions for the gases He, Ar, O2, N2, H2O, and CO2.

Molecular mass values were then taken from Table 2 in Schmidt’s *Thermodynamics* for each gas, and then summed together to determine Mmix(Equation 1). The reason for this molar mix calculation is to get the mass fraction (Equation 2) of each gas. The mass fraction was then used to find other mass based thermodynamic properties: specific heat, enthalpy, internal energy, and entropy. These properties were calculated by matrix algebra in the MATLAB program.

Next, specific heats were calculated for each gas. The polynomial expression for cp bar can be used for O2, N2, H2O, and CO2. This is represented in the MATLAB code by a coefficient matrix for the a, b, c, and d values of Equation 3. This coefficient matrix is multiplied by a temperature array producing cp bar. This cp bar value is divided by molecular mass to get cp for each gas. The only gases that did not use this expression were Ar and He because they were not included in Table 3s in *Thermodynamics* by Schmidt, Ezekoye, Howell, and Baker. Since both Ar and He are monatomic gases, Equation 4 was used instead to calculate cp. The R-values were taken from Table 2 in Schmidt’s *Thermodynamics* for each respective gas.

In order to calculate the entropy change in the mixture, primarily caused by pressure and temperature changes in the turbine, the 2nd Law of Thermodynamics can be manipulated to produce Equation 5 with the reference state at a temperature of 298K and a pressure of 1 atm. In this application, the pressures of the mixture remained constant from inlet to outlet. From this knowledge, the pressure term of Equation 5 takes the natural log of 1, yielding a result of zero. This can be interpreted to mean that pressure effects are negligible. Furthermore, the system is adiabatic and reversible, causing there to be no entropy generation. After reducing the equation, the entropy value was readily obtained.

The internal energy and enthalpy values were found by following the same methodology as that used for the entropy calculations. The calculations for the internal energy and the enthalpy were obtained by using Equation 6 and Equation 7, respectively.

Results and Discussion:

Our values for specific heat found using the property calculator were very close to the properties given in Table 5s and follow the same curve. The error between the two curves can be attributed to the error in the tabulated values. When we used the specific heat polynomial expression for air found in Table 3s, the two curves lined up nearly identically. This relationship can be observed from Figure 1 and Figure 2 in the appendix.

All three values, internal energy, enthalpy, and entropy increased with temperature. Internal energy and enthalpy grew at a much faster rate than the entropy. This relationship is due to the fact that entropy is given in units of while both enthalpy and internal energy have units of. Also, entropy is a function of both temperature and pressure, while the other two properties only depend on temperature. The pressure effects can cause a significant decrease in the entropy values. Although pressure does have a major impact on the entropy values, the application problem did not have a pressure change across the turbine. Thus, the difference in the rate of the entropy value is dependent on the reference pressure value. The error between the values in Table 5s and those found using our property calculator is partially due to the fact that the mixture we used for air is not exactly precise. When we used a more accurate composition, as stated in Table 2 in the appendix, for air then just 79% nitrogen and 21% oxygen, our values were closer to those found in the book. The entropy, internal energy, and enthalpy values as well as the error percentages can be seen in Table 1 in the appendix.

After applying the necessary thermodynamic concepts and inputting the given molar composition values into our property calculator, we obtained that the turbine produced a power output of 1623.969 kJ/kg. When assuming pure air, the power output was determined to be 1112.678kJ/kg. Thus, it was obtained that there was a 31.48 percent error in the power output value. This error can be attributed to the inaccuracies of the molar compositions of the air that were given.

Conclusion and recommendations:

From this project we have learned to use MATLAB to create algorithms to compute properties of ideal gas mixtures based on user inputs. We evaluated how certain thermodynamic properties change due to the temperature drop of our working fluid as it flows through a turbine. Our MATLAB code can accept various molar composition and temperature inputs, allowing us to determine not only how the temperature change affects our results, but also how changing the gas mixture does as well.

Appendix

References:

Schmidt, Philip S. *Thermodynamics: an Integrated Learning System.* Hoboken, NJ: Wiley, 2004. Print.

Moran, Michael J. and Howard N. Shapiro. *Fundamentals of engineering thermodynamics.5th ed.* New York: J. Wiley & Sons, 2004. Print.

Relevant equations:

i (kg/kmol)● yi  (Eq. 1)

where yi is the molar composition of each gas, and Mi is the molecular mass of each gas.

x = yi ● M/Mmix (Eq. 2)

 (Eq. 3)

where T is in Kelvin, and cp bar is in kJ/(kmol • K)

 (Eq. 4)

where R is a gas constant with units of kJ/(kg • K).

(Eq.5)

(Eq.6)

where the reference value taken at 0 K is added to the integral of the temperature term

(Eq.7)

where reference values were taken at 0 K, which were assumed to be 0 (kJ/kg) at that temperature.

Figures and Tables:

CpvsTable.tif

Figure 1: Plot of Cp versus Temperature for air composition of 79% N2 and 21% O2

CpvsTable1.tif

Figure 2: Plot of Cp versus Temperature for accurate air composition values

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 1:**  **Comparative table of obtained property values versus tabulated property values** | | | | | | | | | |
|
| **TEMP** | **h** | **h(table)** | **%error** | **u** | **u(table)** | **%error** | **s** | **s(table)** | **%error** |
| **(Kelvin)** | **(kJ/kg)** | **(kJ/kg)** |  | **(kJ/kg)** | **(kJ/kg)** |  | **(kJ/kg\*K)** | **(kJ/kg\*K)** |  |
| 0 | 264.0972 | 273.92 | 3.586024 | 185.124 | 195.46 | 5.287866 | 6.6545102 | 6.6119 | 0.644447 |
| 100 | 364.8599 | 374.6 | 2.600121 | 259.954 | 267.42 | 2.791873 | 6.9687243 | 6.9259 | 0.618321 |
| 200 | 467.2608 | 476.32 | 1.901922 | 336.422 | 340.42 | 1.174535 | 7.2117756 | 7.1673 | 0.620535 |
| 300 | 571.4603 | 579.79 | 1.436682 | 414.688 | 415.17 | 0.116098 | 7.4115054 | 7.3656 | 0.623241 |
| 400 | 677.5783 | 685.48 | 1.152722 | 494.873 | 492.13 | 0.557355 | 7.5821229 | 7.5356 | 0.617375 |
| 500 | 785.6941 | 793.56 | 0.991214 | 577.056 | 571.49 | 0.97387 | 7.7318362 | 7.6852 | 0.606832 |
| 600 | 895.8461 | 903.98 | 0.899788 | 661.274 | 653.19 | 1.237679 | 7.8657936 | 7.8195 | 0.592028 |
| 700 | 1008.032 | 1016.6 | 0.842815 | 747.527 | 737.07 | 1.418738 | 7.9874179 | 7.9415 | 0.578202 |
| 800 | 1122.209 | 1131.1 | 0.786082 | 835.771 | 822.89 | 1.565293 | 8.0990846 | 8.0536 | 0.564773 |
| 900 | 1238.292 | 1274.4 | 2.833301 | 925.921 | 910.45 | 1.6993 | 8.2024962 | 8.1571 | 0.556524 |
| 1000 | 1356.159 | 1365.2 | 0.662264 | 1017.85 | 999.52 | 1.83433 | 8.2989034 | 8.2535 | 0.550111 |
| 1100 | 1475.643 | 1483 | 0.496122 | 1111.41 | 1088.6 | 2.0949 | 8.3892417 | 8.3436 | 0.547026 |
| 1200 | 1596.538 | 1604.6 | 0.502452 | 1206.37 | 1181.5 | 2.104705 | 8.4742203 | 8.4281 | 0.54722 |
| 1300 | 1718.598 | 1726 | 0.428881 | 1302.49 | 1274.1 | 2.228538 | 8.5543819 | 8.5078 | 0.547519 |
| 1400 | 1841.535 | 1848.2 | 0.360638 | 1399.5 | 1367.7 | 2.324912 | 8.6301434 | 8.5831 | 0.548094 |
| 1500 | 1965.021 | 1971.3 | 0.318521 | 1497.05 | 1462 | 2.397468 | 8.7018256 | 8.6546 | 0.545671 |
| 1600 | 2088.688 | 2095.1 | 0.306068 | 1594.78 | 1557.1 | 2.420166 | 8.7696739 | 8.7225 | 0.54083 |
| 1700 | 2212.125 | 2219.6 | 0.336782 | 1692.29 | 1652.9 | 2.382992 | 8.8338743 | 8.7872 | 0.531163 |
| 1800 | 2334.882 | 2344.7 | 0.418719 | 1789.11 | 1749.3 | 2.27593 | 8.894565 | 8.8491 | 0.513781 |
| 1900 | 2456.469 | 2470.4 | 0.563917 | 1884.77 | 1846.2 | 2.088962 | 8.9518453 | 8.9083 | 0.488818 |
| 2000 | 2576.353 | 2596.6 | 0.779746 | 1978.72 | 1943.7 | 1.801584 | 9.0057831 | 8.965 | 0.454915 |

|  |  |
| --- | --- |
| **Table 2:**  **Accurate Composition Values** | |
| yHe= | 5.24E-06 |
| yAr= | 0.00934 |
| yO2= | 0.209476 |
| yN2= | 0.78084 |
| yH2O= | 2.48E-05 |
| yCO2= | 0.000314 |

MATLAB Code:

%property calculator, ME343

clear all;

%Stored reference values

M=[4.003 39.948 31.999 28.013 18.015 44.010]; %In same order as Cp

R=[2.077 .208 .26 .297 .462 .189];

Sref=[31.5375 3.876 6.6999 6.8045 10.423 4.8585];

Uref=[928.419 92.976 194.2 221.44 412.05 156.57];

Href=[1547.365 154.96 271.72 309.99 549.75 212.93];

%Tabulated Cp and Cv values from table 5s

Cp5s=[1.0041 1.0107 1.0249 1.0452 1.0687 1.0927 1.1154 1.1360 1.1544 1.1706 1.1848 1.1973 1.2083 1.2180 1.2267 1.2345 1.2416 1.2480 1.2539 1.2593 1.2644];

Cv5s=[.7169 .7235 .7376 .758 .7815 .8054 .8281 .8488 .8672 .8834 .8976 .9101 .9210 .9308 .9395 .9473 .9544 .9608 .9667 .9721 .9771];

Pref=10^2;%in kPa

Tref=298;%in Kelvin

COEFF=[0 0 0 0;%Coefficients of Cp equations

0 0 0 0;

0.7963 4.7501e-004 -2.2360e-007 4.1001e-011;

1.0317 -5.6081e-005 2.8847e-007 -1.0256e-010;

1.7896 1.0674e-004 5.8562e-007 -1.9956e-010;

0.5058 0.0014 7.9550e-007 1.6971e-010];

CvCOEFF=COEFF;%Coefficients of Cv equations where Cv=Cp-R

CvCOEFF(3,1)=COEFF(3,1)-R(3);

CvCOEFF(4,1)=COEFF(4,1)-R(3);

CvCOEFF(5,1)=COEFF(5,1)-R(3);

CvCOEFF(6,1)=COEFF(6,1)-R(3);

sintCOEFF(:,1)=COEFF(:,1);%Divide each coefficient by its respective power for integration to find entropy

sintCOEFF(:,2)=COEFF(:,2);

sintCOEFF(:,3)=COEFF(:,3)/2;

sintCOEFF(:,4)=COEFF(:,4)/3;

intCOEFF(:,1)=COEFF(:,1);%Divide each coefficient by its respective power for integration to find h

intCOEFF(:,2)=COEFF(:,2)/2;

intCOEFF(:,3)=COEFF(:,3)/3;

intCOEFF(:,4)=COEFF(:,4)/4;

intCvCOEFF(:,1)=CvCOEFF(:,1);%Divide each coefficient by its respective power for integration to find u

intCvCOEFF(:,2)=CvCOEFF(:,2)/2;

intCvCOEFF(:,3)=CvCOEFF(:,3)/3;

intCvCOEFF(:,4)=CvCOEFF(:,4)/4;

Y=[0 0 0.18 0.51 0.18 0.14];%mole fraction values

MMIX=M\*Y';%Calculate other mixture values

X=Y.\*M./MMIX;

RMIX=8.3145/MMIX;

hrefMIX=Href\*X';

urefMIX=Uref\*X';

srefMIX=Sref\*X';

%get values of mixture pressure and temperature

%disp 'Input the mixture Pressure(kPa) and Temperature(C):';

press=100;%input('Pressure(kPa): ');

tempC=1000;%input('Temperature(C): ');

temp1=1500;%tempC+273.15;

temp2=520;

tempA=[temp1 temp2];%put temperatures of interest into an array of size 2 to run for loop

intCpMIXref=303.6432;%Constant integrated value of Cp at reference temperature (298K)

intCvMIXref=226.1387;%Constant integrated value of Cv at reference temperature (298K)

sMIXref=0.0271;%Constant integrated value of Cpdt/T at reference temperature (298K)

%calculate values of Cp and Cv, integrals of Cp and Cv, h, u and s

for i=0:1

TEMP=[1 tempA(i+1) tempA(i+1)^2 tempA(i+1)^3];

sTEMP=[log(tempA(i+1)/298) tempA(i+1) tempA(i+1)^2 tempA(i+1)^3];

Cp=COEFF\*TEMP';

Cp(1)=5/2\*R(1);

Cp(2)=5/2\*R(2);

Cp=COEFF\*TEMP';

s=sintCOEFF\*sTEMP';

s(1)=5/2\*R(1)\*log(tempA(i+1)/298);

s(2)=5/2\*R(2)\*log(tempA(i+1)/298);

intTEMP=tempA(i+1)\*TEMP;%Add a power to each temp

intCp=intCOEFF\*intTEMP';%Calculate integral values for 4 polynomial functions

intCp(1)=5/2\*R(1)\*tempA(i+1);%Assign integrated values of constant 5/2\*R values = 5/2\*R\*temp

intCp(2)=5/2\*R(2)\*tempA(i+1);

intCv=intCvCOEFF\*intTEMP';

intCv(1)=3/2\*R(1)\*tempA(i+1);

intCv(2)=3/2\*R(2)\*tempA(i+1);

CpMIXA(i+1)=X\*Cp;%multiply matrices to calculate values

intCpMIXA(i+1)=X\*intCp;

intCvMIXA(i+1)=X\*intCv;

CvMIXA(i+1)=CpMIXA(i+1)-RMIX;

sMIX(i+1)=X\*s;

h(i+1)=intCpMIXA(i+1)-intCpMIXref+hrefMIX;

u(i+1)=intCvMIXA(i+1)-intCvMIXref+urefMIX;

S(i+1)=sMIX(i+1)-sMIXref+srefMIX;

end

%Calculate other values

CpMIX=X\*Cp;

CvMIX=RMIX-CpMIX;

K=CpMIX/CvMIX;

work=h(1)-h(2)

**PART B.**

clear all

%Stored reference values

M=[4.003 39.948 31.999 28.013 18.015 44.010]; %All arrays in same order as Cp

R=[2.077 .208 .26 .297 .462 .189];

Sref=[31.5375 3.876 6.6999 6.8045 10.423 4.8585];

Uref=[928.419 92.976 194.2 221.44 412.05 156.57];

Href=[1547.365 154.96 271.72 309.99 549.75 212.93];

%Tabulated Cp and Cv values from table 5s

Cp5s=[1.0041 1.0107 1.0249 1.0452 1.0687 1.0927 1.1154 1.1360 1.1544 1.1706 1.1848 1.1973 1.2083 1.2180 1.2267 1.2345 1.2416 1.2480 1.2539 1.2593 1.2644];

Cv5s=[.7169 .7235 .7376 .758 .7815 .8054 .8281 .8488 .8672 .8834 .8976 .9101 .9210 .9308 .9395 .9473 .9544 .9608 .9667 .9721 .9771];

h5s=[273.92 374.6 476.32 579.79 685.48 793.56 903.98 1016.6 1131.1 1274.4 1365.2 1483.0 1604.6 1726 1848.2 1971.3 2095.1 2219.6 2344.7 2470.4 2596.6];

u5s=[195.46 267.42 340.42 415.17 492.13 571.49 653.19 737.07 822.89 910.45 999.52 1088.6 1181.5 1274.1 1367.7 1462 1557.1 1652.9 1749.3 1846.2 1943.7];

s5s=[6.6119 6.9259 7.1673 7.3656 7.5356 7.6852 7.8195 7.9415 8.0536 8.1571 8.2535 8.3436 8.4281 8.5078 8.5831 8.6546 8.7225 8.7872 8.8491 8.9083 8.965];

Y=[0 0 0.18 0.51 0.18 0.14];%mole fraction values

%Calculate other mixture values

Y=[yHe yAr yO2 yN2 yH2O yCO2];

MMIX=M\*Y';%molar mass of mixture

RMIX=8.3145/MMIX;%R value of mixture

X=Y.\*M./MMIX;%mass fraction

hrefMIX=Href\*X';%href of mixture

urefMIX=Uref\*X';%uref of mixture

srefMIX=Sref\*X';%sref of mixture

TMIN=0;

TMAX=2000;

int=100;

tempC=[TMIN:int:TMAX];%temperature array in degrees celsius

tempA=tempC+273.15;%temperature array in degrees kelvin for calculations

tempB=[0:100:2000];%temperature array to use for graphing tabulated values

COEFF=[0 0 0 0;

0 0 0 0;

0.7963 4.7501e-004 -2.2360e-007 4.1001e-011;

1.0317 -5.6081e-005 2.8847e-007 -1.0256e-010;

1.7896 1.0674e-004 5.8562e-007 -1.9956e-010;

0.5058 0.0014 7.9550e-007 1.6971e-010];

CvCOEFF=COEFF;

CvCOEFF(3,1)=COEFF(3,1)-R(3);

CvCOEFF(4,1)=COEFF(4,1)-R(3);

CvCOEFF(5,1)=COEFF(5,1)-R(3);

CvCOEFF(6,1)=COEFF(6,1)-R(3);

sintCOEFF(:,1)=COEFF(:,1);%Divide each coefficient by its respective power for interation

sintCOEFF(:,2)=COEFF(:,2);

sintCOEFF(:,3)=COEFF(:,3)/2;

sintCOEFF(:,4)=COEFF(:,4)/3;

intCOEFF(:,1)=COEFF(:,1);%Divide each coefficient by its respective power for interation

intCOEFF(:,2)=COEFF(:,2)/2;

intCOEFF(:,3)=COEFF(:,3)/3;

intCOEFF(:,4)=COEFF(:,4)/4;

intCvCOEFF(:,1)=CvCOEFF(:,1);

intCvCOEFF(:,2)=CvCOEFF(:,2)/2;

intCvCOEFF(:,3)=CvCOEFF(:,3)/3;

intCvCOEFF(:,4)=CvCOEFF(:,4)/4;

intCpMIXref=303.6432;%Constant integrated value of Cp at reference temperature (298K)

intCvMIXref=226.1387;%Constant integrated value of Cv at reference temperature (298K)

sMIXref=0.0271;%Constant integrated value of Cpdt/T at reference temperature (298K)

for i=0:1:(TMAX-TMIN)/int

TEMP=[1 tempA(i+1) tempA(i+1)^2 tempA(i+1)^3];

sTEMP=[log(tempA(i+1)/298) tempA(i+1) tempA(i+1)^2 tempA(i+1)^3];

Cp=COEFF\*TEMP';

Cp(1)=5/2\*R(1);

Cp(2)=5/2\*R(2);

Cp=COEFF\*TEMP';

s=sintCOEFF\*sTEMP';

s(1)=5/2\*R(1)\*log(tempA(i+1)/298);

s(2)=5/2\*R(2)\*log(tempA(i+1)/298);

intTEMP=tempA(i+1)\*TEMP;%Add a power to each temp

intCp=intCOEFF\*intTEMP';%Calculate integral values for 4 polynomial functions

intCp(1)=5/2\*R(1)\*tempA(i+1);%Assign integrated values of constant 5/2\*R values = 5/2\*R\*temp

intCp(2)=5/2\*R(2)\*tempA(i+1);

intCv=intCvCOEFF\*intTEMP';

intCv(1)=3/2\*R(1)\*tempA(i+1);

intCv(2)=3/2\*R(2)\*tempA(i+1);

CpMIXA(i+1)=X\*Cp;

intCpMIXA(i+1)=X\*intCp;

intCvMIXA(i+1)=X\*intCv;

CvMIXA(i+1)=CpMIXA(i+1)-RMIX;

sMIX(i+1)=X\*s;

h(i+1)=intCpMIXA(i+1)-intCpMIXref+hrefMIX;

u(i+1)=intCvMIXA(i+1)-intCvMIXref+urefMIX;

S(i+1)=sMIX(i+1)-sMIXref+srefMIX;

hErr(i+1)=abs((h5s(i+1)-h(i+1))/h5s(i+1))\*100;

uErr(i+1)=abs((u5s(i+1)-u(i+1))/u5s(i+1))\*100;

sErr(i+1)=abs((s5s(i+1)-S(i+1))/s5s(i+1))\*100;

end

TABLE=[h' h5s' hErr' u' u5s' uErr' S' s5s' sErr'];%variable to combine all calculated values, tabulated values, and percent errors

plot(tempC,CpMIXA,'o');

hold on

plot(tempB,Cp5s,'r');

TITLE('Cp vs. Temperature')

XLABEL('Temperature [C]')

YLABEL('Cp [KJ/(Kg\*K)')

Legend('Calculated values','Tabulated (Table 5s)')