Teaching thermo-chemical equilibrium using a MATLAB algorithm

L. Massa and P. Jha

Mechanical and Aerospace Engineering Department
University of Texas at Arlington

Abstract

Computers are an integral part of learning in different fields of education. The ability of scientific computing to solve realistic problems can strengthen engineering education by allowing the students to analyze complex systems. To improve the quality of learning along this path, educators must take a step to make their teaching style flexible and include elements of numerical analysis as an ingredient of upper-class engineering courses. This paper documents our attempt to teach a difficult problem, chemical equilibrium in combustion systems, using a method based on matrix factorization that is very well suited to be implemented in MATLAB. This paper presents our novel numerical algorithm that treats chemical equilibrium beyond the simple balance. By doing so, we emphasize the thermodynamic and detailed nature of chemical equilibrium. These two concepts arise naturally by performing a singular value decomposition of the stoichiometric matrix, avoiding the necessity to specify reaction paths, and providing a lean and easy-to-understand algorithm based on matrix-vector multiplications. A complete MATLAB code is presented, verified and discussed in details. Educational effectiveness is investigated via in-class student surveys. Based on the student evaluations and feedback, it is evident that this module proved beneficial towards developing a sound understanding of the topic. Our results emphasize the benefit of teaching engineering courses from first principle. Educators should refrain from using web applications to teach this fundamental of mechanical engineering and challenge their students to carry out a detailed computational analysis.

Introduction

Computer technology plays a two-fold role in the field of engineering education. On the one hand, using computer software to create multimedia demonstrations in class aids the students in understanding new concepts. Previous research[1,2] has shown that students who learned from teachings supplemented by animations performed better than those who learned through the text-only technique. Using graphics, simulations, animations of concepts and their applications has the potential to explain concepts more clearly and in a shorter time when compared to the conventional lecture-only approach. On the other hand, scientific computing allows the analysis of large, complex engineering problems, involving, for example the solution of coupled non-

Proceedings of the 2013 ASEE Gulf-Southwest Annual Conference,
The University of Texas at Arlington, March 21 – 23, 2013.

linear equations. The advantage of computers in education is, in this second case, to make the lecture less abstract by presenting a realistic analysis.

The present research investigates benefits related to the second issue. The topic under investigation is the definition of thermo-chemical equilibrium, which mechanical and aerospace engineering students learn and apply to determine chemical compositions, flame temperatures, specific impulses of rockets, etc..., in combustion, propulsion and atmospheric modeling classes. In this paper we argue that explaining the concept of thermodynamic equilibrium based on chemical reactions[3] is an uninformative approach for two reasons. First, it does not establish that the equilibrium principle is derived based on purely thermodynamic considerations, where chemical paths play no role. The rationale for its applicability to various stoichiometric balances is that thermo-chemical equilibrium is a detailed balance principle[4], meaning that each subsystem of the mixture is itself in equilibrium. Nonetheless, subsystem balances are neither chemical paths nor reactions and can include, for example, fractional stoichiometric coefficients. Second, it does not identify what information is actually needed for an equilibrium computation, which should include two thermodynamic variables and a number of additional constraints equal to the number of atom types in the mixture. Since these constraints are imposed by the stoichiometry, they are equal in number to the non-zero singular values of the stoichiometric matrix. In this regard, we argue that a proper definition of the number of system constraints is the dimension of the range of the stoichiometric matrix.
The main topic of the present paper is a computational module prepared for teaching chemical equilibrium in a combustion course. The algorithm uses singular value decomposition (SVD) both to define the problem and to solve it through non-linear searches on a (reduced) manifold spanned by the range of the stoichiometric matrix. The numerical operations are cast in a matrix-vector form, leading to a lean presentation and implementation. A similar SVD approach was used by Fox[5] to reduce the finite rate chemistry species into conserved and reactive subspaces. Further, the idea of reducing the search manifold by manipulation of the stoichiometric matrix is similar to the concept of element potentials introduced by Reynolds[6], but more suitable for education because of its definition in terms of matrix-vector products.
In the remainder of this paper a complete MATLAB implementation of the algorithm is presented, verified and discussed. The educational outcomes of this work are analyzed through in class surveys, and, finally, the conclusions are discussed.

Chemical Equilibrium

Combustion of hydrocarbon fuels releases a variety of product species. At high temperature, the products of hydrocarbon combustion are not just represented by the major species (CO_2, O_2, H_2O, N_2). These species dissociate and produce a variety of minor species, which may be important from both the energetic and the environmental stand-points. In this section we briefly discuss the theoretical background and method to calculate the mole fractions of the product species at a given temperature and pressure. This problem is also referred to as *TP* (temperature and pressure), but the outlined solution procedure can be easily extended to *HP*

(enthalpy and pressure) or *SP* (entropy and pressure) problems.

Second Law of Thermodynamics

The second law of thermodynamics identifies the equilibrium condition in composition space as the state of maximum entropy of the system. In order for equilibrium to represent a detailed (rather than global) balance, the entropy must be maximal over all the degrees of freedom of the system. For a system at given pressure and temperature, the maximization of the entropy leads to (see Ref. [7])

$$\sum \mu_i \mathrm{d}n_i = \mu^T \ \mathrm{d}n = 0, \tag{1}$$

where μ_i are the species Gibbs functions and $\mathrm{d}n_i$ the changes in mole numbers. Constraints are of stoichiometric nature, and in the absence of any, the only detailed solution would be $\mu_i = 0 \ \forall i$, which violates conservation of mass, thus it is discarded.
Stoichiometry constraints on equation (1) are typically expressed in terms of the stoichiometric matrix

$$A\,n = b \Rightarrow A\ \mathrm{d}n = 0, \tag{2}$$

where the matrix A is of size $n_{\text{element}} \times n_{\text{species}}$, and expresses the number of each atomic element in each species molecule. Therefore, the product $A\,n$ counts the total number of elements in the systems, which is set to a constant by imposing equation (2). The evaluation of A can be implemented in MATLAB in terms of the array of strings for the species and elements, as demonstrated in the code fragment reported in Fig. 1.

```
elements={'c','h','o'};
species = {'ch4','o2','co2','co','h2o','h','h2','o','oh','ho2'};
Nel = numel(elements);
Nsp = numel(species);
A=zeros(Nel,Nsp);
for i = 1:Nel
   for j = 1:Nsp
      ip=strfind(species{j},elements{i});
      if ~isempty(ip);
         ip1= min(ip+1,numel(species{j}));
         coe=str2num(species{j}(ip1));
         if ~isempty(coe);
            A(i,j) = coe;
         else
            A(i,j)=1;
         end
      end
   end
```

end

Figure 1. Fragment of code to determine the stoichiometric matrix A in equation (2).

The implementation and manipulation of the stoichiometric constraints renders equilibrium didactically challenging: the Lagrange multipliers strategy discussed in Ref. [7] is effective but cumbersome. We propose a singular value decomposition of the stoichiometric matrix as a viable solution strategy, because of its lean and straightforward implementation. The algorithm starts with identifying the effective number of constraints as the codimension of the nullspace of A (*N(A)*), which is, as a consequence, mapped by the right singular vectors corresponding to zero singular values. Hence,

$$A = USV^T \Rightarrow \mathrm{d}n = V\left(:, n_{\text{element}} + 1 : n_{\text{species}}\right) \mathrm{d}c, \tag{3}$$

where dc is the projection of dn on the nullspace *N(A),* and the columns of the matrix

$$S_1^T \equiv V\left(:, n_{\text{element}} + 1 : n_{\text{species}}\right)$$

span *N(A)*. Thus, equation (1) becomes,

$$\mu^T S_1^T \mathrm{d}c = \mathrm{d}c^T S_1 \mu = 0 \Rightarrow S_1 \mu = 0, \tag{4}$$

where the implication is supported by the fact that dc can be any element of *N(A),* which is a consequence of the detailed balancing principle. The MATLAB implementation is simple, i.e.,

```
[U,S,V]=svd(A);
S1 = V(:,Nel+1:end)'.
```

Note that the vector μ contains the unknown mole fractions X, in fact,

$$\mu / \hat{R}T \equiv g^0 / \hat{R}T + \log X + \log p / p_0,$$

where $g^0 \equiv h\text{-}Ts^0$, and p^0 is the pressure at which s^0 is evaluated (typically 1 bar).

We start manipulating eq. (4) by focusing on a *TP* problem. The unknowns are brought to the left-hand side and the following system is obtained,

$$S_1 \log X = -S_1 \frac{g^0}{\hat{R}T} - S_1[1,1,\ldots,1]^T \log p / p_0 \equiv v_1. \tag{5}$$

Equation (5) provides $n_{species} - n_{elements}$ equations that are supplemented by equation (2), which is recast in an homogeneous form dependent on the mole fractions as follows,

$$S_2 X = 0, \text{ where, } S_2 \equiv A - \tilde{b}\hat{M}^T A. \tag{6}$$

Here $\tilde{b}$ is the vector of the mole mass ratios of the elements, and $\hat{M}$ is the vector of the atomic molecular weights; e.g., $\hat{M} = [12,1,16]^T$ for the MATLAB code in Fig. 1. The matrix S_2 in equation (6) is obviously singular because it does not enforce conservation of mass, and thus one of its rows is replaced by the condition

$$\sum_{i=1}^{n_{\text{species}}} X_i = 1 \quad ;$$

whereby the system is non-singular, but also non-homogeneous. Finally, we have

$$S_2 X = [1, 0, \ldots, 0]^T \equiv v_2, \tag{7}$$

as second part of the resolvent system.

Each of the two sub-systems is linear; equation (4) in log X and equation (7) in X. Nonetheless, the combination of the two systems is not linear. Since the dimension (n_r) of the range of S_1 is typically much larger than that of S_2, we find it computationally efficient to reduce equation (4) by parameterizing the variation of its solution with a vector of size equal to the codimension of the range of S_1. Hence, we carry out a singular value decomposition

$$S_1 = L K R_0^T \quad ,$$

and set

$$\log X = N + R c, \tag{8}$$

where,

$$N \equiv R_1 (S_1 R_1)^{-1} v_1, \tag{9}$$

and

$$R_1 \equiv R_0(:, 1 : n_r) \text{ and } R \equiv R_0(:, n_r + 1 : \text{end}). \tag{10}$$

This operation is performed in MATLAB without the need of the matrix inversion as shown in Fig. 2.

```
[L,K,R0]=svd(S1)
N=R0(:,1:nr)*((S1*R0(:,1:nr))\V1);
R=R0(:,nr+1:end);
```

Figure 2. Fragment of code to reduce the dimensions of the solution log X by manipulation of equation (4).

Thus, the search of the solution is restricted to a vector of size equal to the codimension of the range of S_1, which is typically equal to the number of elements. This drastic reduction of the unknown space from (possibly) hundreds to a few elements sharply decreases the computational burden associated with the solution of the non-linear system in equation (7). Another advantage of equation (8) is that it voids the problem with species disappearing from the mixture at high temperature, in which case the Newton update of the original formulation becomes singular because of the problem with $\log(0) \to -\infty$.

The solution of a *TP* problem can be accomplished by a multivariate minimization procedure instead of the Newton method, thus augmenting the solution convergence radius. This outcome becomes important when one uses equilibrium to teach problems with variable parameters, e.g.,

when evaluating Hugoniot curves for detonation waves. Multivariate minimization is implemented in MATLAB using the intrinsic function *fminsearch* as described in Fig. 3.

```
options=optimset('TolFun',1d-9,'TolX',1d-9,'MaxFunEvals',10000);
cv0 = cguess*ones(nv,1);
c = fminsearch(@locfun,cv0,options);disp(c);
X=exp(N+R*c);
   function out=locfun(cv)
      l=N+R*cv;
      resid=b2-Z2*exp(l);
      out = norm(resid);
   end
end
```

Figure 3. Code for the evaluation of chemical equilibrium using a multivariate minimization strategy.

Properties

The evaluation of the mixture properties is an important aspect of the algorithm and comes about in the definition of the potentials g_0. Assuming the mixture composed of calorically perfect gases, the information necessary for the computation of the chemical potential is the temperature dependent heat capacity C_p°, plus the enthalpy and entropy at formation. We find it useful to point students to a website where they can obtain thermodynamic information on a wide variety of gases, so that they can use this knowledge for other problems, beyond the scope of combustion education. On the National Institute of Standards website (http://webbook.nist.gov/chemistry/), the Shomate equation defines the heat capacity in terms of five coefficients A - E, which can be used to evaluate the standard entropy and enthalpy by means of additional integration constants,

$$C_p^\circ = A + Bt + Ct^2 + Dt^3 + E/t^2, \tag{11a}$$

$$h - h_{298K} = \int_{298K}^{T} C_p^\circ \mathrm{d}T, \tag{11b}$$

$$s^\circ - s_{298K}^\circ = \int_{298K}^{T} \frac{C_p^\circ}{T} \mathrm{d}T. \tag{11c}$$

Verification

We have carried out sample computations involving small hydrocarbons (C_5H_{12}, C_3H_8, CH_4 and air, and found that with the initial guess $c = [-20, -20, -20, -20]^T$ the algorithm converges everywhere in the temperature range $T \in [1000 - 3000]K$, and for fuel weight fractions

$W \in [0.05, 0.95]$. The accuracy of the approach is verified against the NASA thermochemical equilibrium code[7] by evaluating the product composition supported by CH_4 + Air burning at 3000 K, 1 bar and with weight fractions $W_{CH_4} = 0.055, W_{O_2} = 0.21, \text{and}, W_{N_2} = 0.735$. Results reported in table (1) show a good agreement between the two algorithms, also considering that the interpolating polynomials used to define the thermodynamic properties (equation (11)) differ.

Species	NASA	Present
CO	5.9803×10^{-2}	5.9832×10^{-2}
CO_2	2.6807×10^{-2}	2.6956×10^{-2}
H	2.8659×10^{-2}	2.8901×10^{-2}
H_2	3.2837×10^{-2}	3.341×10^{-2}
H_2O	1.0895×10^{-1}	1.096×10^{-1}
N	1.1266×10^{-5}	1.126×10^{-5}
NO	1.4111×10^{-2}	1.4716×10^{-2}
NO_2	2.6669×10^{-6}	2.4375×10^{-6}
N_2	6.5577×10^{-1}	6.5538×10^{-1}
O	1.6799×10^{-2}	1.6847×10^{-2}
OH	3.4207×10^{-2}	3.2225×10^{-2}
O_2	2.2040×10^{-2}	2.2107×10^{-2}

Table 1. Verification of the chemical equilibrium algorithm against the NASA thermochemical equilibrium code[7].

HP algorithm

HP problems are rather important in mechanical/aerospace engineering education because they define flame temperatures, rocket chamber pressures and specific impulses, and the detonation speed according to the Chapman-Jouguet theory. The core algorithm for an *HP* problem is only slightly different from what discussed in the previous sections. The conservation of energy in terms of the mass fraction is written as

$$X^T \left(h - A^T \hat{M} \overline{h}_0 \right) = 0, \tag{12}$$

where h is the vector of molar enthalpies of products and h_0 is the enthalpy of the reactants per unit mass. The *HP* algorithm requires two modifications with respect to the *TP* analog. First, the vector v_1 is not fixed at the beginning of the computations, thus we find useful to rewrite equation (8) as

$$\log X = -R_1 (S_1 R_1)^{-1} S_1 \left(\nu + [1,1,\ldots,1]^T \log p / p_0 \right) + Rc = N_0 \nu + Rc + N_p, \tag{13}$$

where N_0 should be evaluated before the non-linear search. Note that if the matrix S_1 was full-rank, equation (13) would become

$$\log X = \left(\nu + [1,1,\ldots,1]^T \log p / p_0\right),$$

which we previously identified as the solution for the non-constrained problem. Second, solving the problem as a multivariate minimization is inefficient, because the computer time necessary in evaluating mixture properties ($\nu \equiv \frac{g^0}{\hat{R}T}$ and h) overwhelms the linear algebra time. The large computational time becomes an issue when the algorithm is run with a large set of initial values, as when evaluating the Hugoniot curve for a given fuel in the context of a combustion wave analysis. Therefore, we use the Newton method with solution vector $[c^T, T^T]$, residual,

$$r = \left[v_2 - S_2 X, X^T\left(h - A^T \hat{M}\bar{h}_0\right)\right]^T, \tag{14}$$

and Jacobian,

$$J = \begin{pmatrix} J_1 & J_2 \\ J_3 & J_4 \end{pmatrix}, \tag{15}$$

with

$$\begin{aligned} J_1 &\equiv S_2 \mathrm{D}\left[X\right] R, & J_2 &\equiv S_2 \mathrm{D}\left[X\right] N_0 \nu_1 \\ J_3 &\equiv \left(h - A^T \hat{M}\bar{h}_0\right)^T \mathrm{D}\left[X\right] R, & J_4 &\equiv X^T h' + \left(h - A^T \hat{M}\bar{h}_0\right)^T \mathrm{D}\left[X\right] N_0 \nu', \end{aligned} \tag{16}$$

where $D[X]$ is a square matrix having X on the main diagonal and the prime indicates differentiation with respect to the temperature. A sample code for the calculation of the adiabatic flame temperature of a generic hydrocarbon $C_m H_n$ in stoichiometric air is included in Fig. 4. The only module to be added to this program is the *perfgas.m* routine, needed to evaluate the thermodynamic properties of the mixture. This code was validated against data provided in Ref [3] (Appendix B) and results shown in table (2) show that the flame temperatures are evaluated with an error lower than 2 K. No convergence problem was detected when analyzing hydrocarbons with *m* up to 10.

Fuel	Glassman[3]	Present
CH_4	2226K	2227 K
C_2H_2	2541K	2539K
C_2H_6	2260K	2261K
C_3H_8(L)	2257K	2258K
C_5H_{12}(L)	2262K	2263K
$C_{10}H_{16}$(L)	2308K	2308K

Table 2**.** Verification of the *HP* algorithm against adiabatic flame calculations in normal stoichiometric conditions reported in Ref. [3].

```
function T= Tflame(m,n,Tcold,p)
p0=1; %bar
Nfrac = 3.76; %molar ratio between nitrogen and oxygen in air

%set the problem data
W_F = (m*12+n)/(m*12+n + (m+n/4)*(32+3.76*28)); W_A = 1-W_F;
fuel = ['c',num2str(m),'h',num2str(n)];
W_O = W_A*32/(32+Nfrac*28); W_N = W_A*Nfrac*28/(32+Nfrac*28);

%Mass matrix:
elements={'c','h','o','n'};
species = {fuel,'o2','n2','co2','co','h2o','h','h2','o',...
    'oh','ho2','no','hno','n','no2'};
Nel = numel(elements);Nsp = numel(species);
A=zeros(Nel,Nsp);
for i = 1:Nel;
   for j = 1:Nsp
      ip=strfind(species{j},elements{i});
      if ~isempty(ip);
         ip1 = ip-1+ regexp(species{j}(ip:min(ip+2,end)),'\d');
         coe=str2num(species{j}(ip1));
         if ~isempty(coe)&& ip1(1)-ip <=1;
            A(i,j) = coe;
         else
            A(i,j)=1;
         end;
      end;
   end;
end

%Molar Mass of elements and species
Mel = [12,1,16,14];Msp = Mel*A;

%system matrices
[U,S,V]=svd(A);S1 = V(:,Nel+1:end)';S2 = A;

%RHS vectors
Sp = ones(size(S1,2),1)*log(p/p0);
V2 = W_F/Msp(1)*A(1:4,1)+[0;0;W_O/Mel(3);W_N/Mel(4)];
```

```
%mole-mass of reactants
eta0=[W_F,W_O,W_N]./Msp(1:3);
%enthalpy of reactants (per unit mass)
H0=0;for k=1:3;H0=H0+ eta0(k)*perfgas('h',Tcold,species{k});end

Tguess=1000;cguess = -10*ones(Nel,1);
[X,cv] = NewtonHP(S1,S2,V2,Msp,species,H0,Sp,[cguess;Tguess]);
T=cv(end);

function [X,cva]=NewtonHP(S1,S2,V2,Msp,species,H0,Sp,cguess)
Runi = 8.31447215;  %KJ/Kmole
[L,K,R0]=svd(S1);
nc=size(S1,2);nr=size(S1,1);nv=nc-nr;Nsp = numel(species);
N0=-R0(:,1:nr)*inv(S1*R0(:,1:nr))*S1;R=R0(:,nr+1:end);Np= N0*Sp;
nu = zeros(Nsp,1);h  = zeros(Nsp,1);nu1= zeros(Nsp,1);h1 = zeros(Nsp,1);

Z2=[ones(1,size(S2,2));S2-V2*Msp];
Z2=Z2(1:nv,:);b2=zeros(nv,1);b2(1)=1;
Hi=(Msp*H0)';

cva = cguess;iter=0;
while iter < 5000
   iter=iter+1;
   cv=cva(1:end-1);T=cva(end);
   for k = 1:Nsp;nu(k) = perfgas('g',T,species{k})/(Runi*T);end
   for k = 1:Nsp;h(k) = perfgas('h',T,species{k});end
   Tp=T+.1;
   for k = 1:Nsp;nu1(k) = perfgas('g',Tp,species{k})/(Runi*Tp);end
   for k = 1:Nsp;h1(k) = perfgas('h',Tp,species{k});end
   h1=(h1-h)*10;nu1=(nu1-nu)*10;
   X=min(max(exp(N0*nu+R*cv+Np),1d-18),1);
   resid = [b2-Z2*X;-X'*(h-Hi)];
   if norm(resid) < 1d-8;break;end;
   J=[Z2*diag(X)*[R,N0*nu1];(X'.*(h-Hi)')*R,X'*h1+(X.*(N0*nu1))'*(h-Hi)];
   cva = cva + min(max(J\resid,-10),10);
end
```

Figure 4. Complete code to evaluate the adiabatic flame temperature of a generic hydrocarbon C_mH_n in stoichiometric air at the given pressure and temperature.

Educational approach and results

We first introduced the topic of chemical equilibrium to the students with a verbal lecture that emphasized the concepts of singular value decomposition and projection over finite dimensional vector fields. Then, we used a computer connected to a projector to explain the same concepts using MATLAB. The code was explained to the students line-by-line. The students were given a demo on how to use the code to obtain results. Homework was assigned where the students were asked to perform two tasks. a) Validate the code by comparing the results of the SVD procedure to the NASA thermo-chemical equilibrium code[7]. b) Solve a series of problems involving thermo-chemical equilibrium.

The analysis of the educational outcomes focused on the following issues:

1. Student keenness in using advanced linear algebra concepts to solve problem.
2. Ease with which students can learn the tool.
3. Ease with which students can apply the tool.
4. Student's learning performance with the aid of the tool.

Student Feedback

Students were asked to give their feedback, by answering a questionnaire, so that we could assess if the use of computer technology to teach this topic was beneficial. The questionnaire contained the following questions.

On a scale of 1 to 5, 1 being the lowest and 5 being the highest, rate the following:

Q_1 How well did you understand the principle of chemical equilibrium in combustion?

Q_2 How helpful was the MATLAB program in understanding the principle of chemical equilibrium?

Q_3 What is your level of experience with MATLAB?

Q_4 How good would your understanding be if you were explained only the theory behind the principle of chemical equilibrium?

Q_5 To what extent do you think that computer aided teaching can replace just lecturing?

Q_6 How helpful do you think it would be to apply this computer aided teaching technique to other topics?

The feedback was anonymous and the students were given a week to answer all the questions at home.

Survey Results

Thirteen of the fifteen students enrolled in the class responded to the survey. The responses to the six questions listed in the previous section are shown in Fig. 5. Overall, the students considered the present computer approach to teaching combustion useful. Only one student strongly disagreed with the teaching method, and he marked both questions Q_2 and Q_6 with a score of one. In his comments such a student remarked that he previously took a combustion course at another university, that he understood chemical equilibrium well (the Q_1 score was 4), and that he would have preferred an approach based on an existing graphic user interface (GUI)

program (the NASA CEA code[7]); in other words, he deemed modifying an existing program provided to the class by the instructor "frustrating" and "futile". Based on these and other comments made in person to the instructor, we conclude that having previously being taught the subject with a different approach made him reject our alternative explanation. We also remark that the student has a strong visual and global approach to learning[2], which explains his preference for a GUI computer program. His learning style might interfere with the analytical computational approach to teaching proposed here.

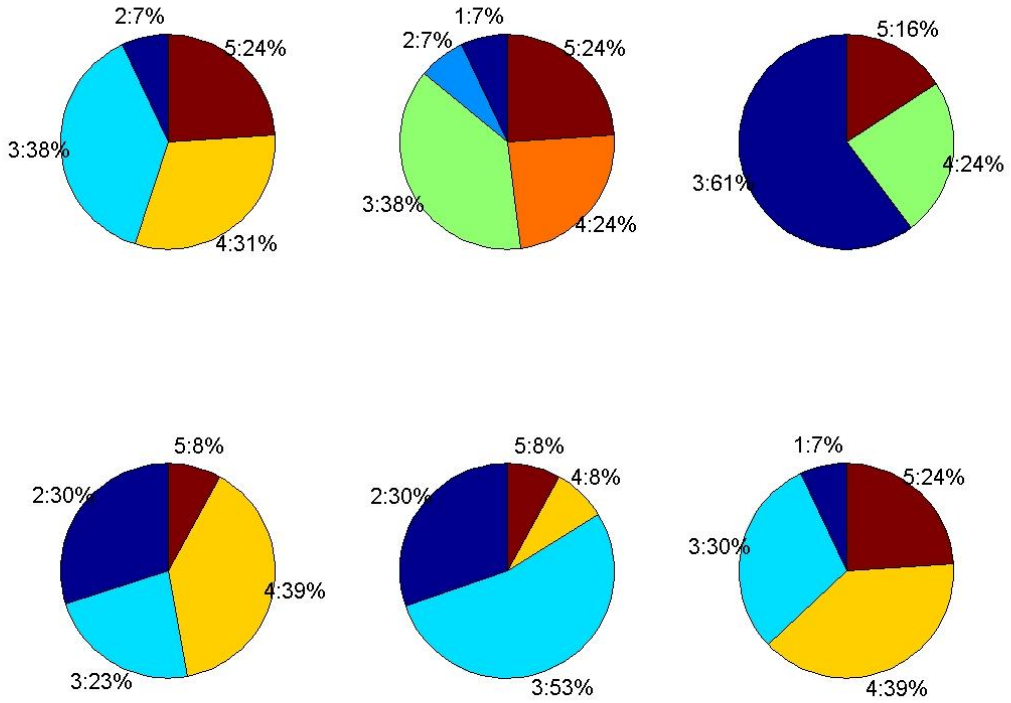

Figure 5. Results of student surveys. The six questions are analyzed independently and displayed in increasing order from left to right, top to bottom. The first number close to each pie segment refers to the score (1 lowest, 5 highest), the second number to the percentage of students agreeing with that score.

The correlation coefficient matrix based on the survey answers is shown in table (3). As expected, questions Q_2 and Q_6 are strongly correlated. Surprisingly, question Q_3 (i.e., MATLAB knowledge) is negatively correlated with both Q_2 and Q_6, which indicates that students with weak prior MATLAB knowledge are willing to learn this language and use it to solve engineering problems. Further, the strong negative correlation between Q_4 and both Q_2 and Q_6

indicates that students thought that the MATLAB approach helped them learning the theory.

Question	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6
Q_1	1	0.5	0.24	0.08	0.17	0.22
Q_2	0.5	1	-0.2	-0.44	0.28	0.81
Q_3	0.24	-0.2	1	0.36	-0.06	-0.37
Q_4	0.08	-0.44	0.36	1	0.12	-0.45
Q_5	0.17	0.28	-0.06	0.12	1	0.32
Q_6	0.22	0.81	-0.37	-0.45	0.32	1

Table 3. Correlation coefficient based on the answers to questions Q_1 - Q_6 reported in this section.

Summary and Conclusions

We propose a MATLAB program based on singular value decomposition (SVD) of the stoichiometric matrix to explain the equilibrium composition of high temperature combustion. The teaching of equilibrium should focus on two aspects 1) that it is a purely thermodynamic rather than kinetic principle 2) that it represents a detailed rather than global balance. The proposed formulation accomplishes these objectives by eliminating reaction paths, and reducing the degrees of freedom of the system to the vector basis spanning the nullspace of the stoichiometric matrix: detailed balance implies maximization of entropy over these degrees of freedom.

The proposed use of SVD to impose the atom conservation constraints and to reduce the size of the unknown (search) manifold leads to a simple implementation that involves only matrix-vector products, suitable for engineering education.

From the evaluation of the student feedback, we found that 48% of the students found the computer aided teaching either helpful or very helpful. A follow-up test proved that the students had understood the concept well. Regardless of their previous experience with MATLAB, a majority of students found it easy to understand a provided code and modify it towards learning the chemical equilibrium principle. Hence, we conclude that the use of computer technology as an aid to traditional teaching methods can help the students understand concepts more easily and remember them for a longer time.

References

1. Y.K. Baek and B.H. Layne. Color, graphics, and animation in a computer-assisted learning tutorial lesson. *Journal of Computer-Based Instruction*, 1988.
2. R.M. Felder and L.K. Silverman. Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681, 1988.
3. I. Glassman and R.A. Yetter. *Combustion*. Academic Press, 2008.
4. W.G. Vincenti and C.H. Kruger. *Introduction to physical gas dynamics*. Wiley, New York, 1965.
5. R.O. Fox. *Computational models for turbulent reacting flows*. Cambridge Univ. Press, 2003.
6. W.C. Reynolds. The element potential method for chemical equilibrium analysis: Implementation in the interactive program Stanjan, version 3. Technical report, Dept. of Mechanical Engineering, Stanford Univ., Stanford, 1986.
7. S. Gordon and B.J. McBride. Computer program for calculation of complex chemical equilibrium compositions, rocket performance, incident and reflected shocks and Chapman-Jouguet detonations. NASA-SP 273, National Aeronautics and Space Administration, 1971.

LUCA MASSA

Dr. Massa currently serves as an Assistant Professor of Mechanical and Aerospace Engineering at the University of Texas at Arlington. His research interests include combustion, detonation, transition modeling, boundary layers and computational fluid dynamics.

PALLAV JHA

Mr. Jha is a PhD student in the Department of Mechanical and Aerospace Engineering at the University of Texas at Arlington. He is currently a teaching assistant at this institution. His research interests are internal combustion engine and innovative concepts in automotive engineering.