

APPENDIX A

A.1 Rank and checking linear independence of reactions using MATLAB

In this appendix we will describe the procedures involved in obtaining the rank of the matrix and identifying independent reactions.

Consider the carbon dioxide formation considered earlier. The reaction scheme is



Step 1: type in the stoichiometric matrix as follows: Each row is reaction index and each column is the species index. Thus each row represents one reaction in the system. We take carbon, C, to be species 1, oxygen to be 2, carbon monoxide is 3, and carbon dioxide is 4. Then the matrix of stoichiometric coefficients is:

$$\begin{bmatrix} -2 & -1 & 2 & 0 \\ 0 & -1 & -2 & 2 \\ -1 & -1 & 0 & 1 \end{bmatrix}$$

Step 2: find the rank by typing the following script in MATLAB:

```
rank(v)
```

For the above case the answer will be 2 so that there are only two independent reactions. In other words the third reaction is a linear combination of the first two reactions. To find this combination go to step 2a.Step 2a: do this if the rank is less than the number of reactions. Type the following script in MATALB:

```
rref(v')
```

Note that v' is the transpose of the stoichiometric matrix and the above command reduces the transposed matrix to the Echelon form (in this form the main diagonal of the matrix has ones on it). The result will be:

$$\begin{bmatrix} 1 & 0 & 0.5 \\ 0 & 1 & 0.5 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The last column gives the linear combination for the third reaction. Thus reaction (5c) is equal to 0.5 times the first reaction plus 0.5 times the second reaction. In this simple example it is obvious by inspection as well.

Step 3: To find the system invariants (these will be relations between moles produced or consumed among various species), type the following script in MATABL:

```
rref(v)
```

This reduces the stoichiometric matrix to the echelon form. The result for our example is:

$$\begin{bmatrix} 1 & 0 & -2 & 1 \\ 0 & 1 & 2 & -2 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The last two columns here provide the invariants. Thus

$$\Delta F_3 = -2 * \Delta F_1 + 2 * \Delta F_2$$

and

$$\Delta F_4 = 1 * \Delta F_1 - 2 * \Delta F_2$$

Here ΔF represents the change in molar flow rate between the exit and entrance for species I .

Note that the last row is all zero indicating that the third reaction is redundant and does not contribute to the invariant (recall that the rank was two).

Let us see what the implication of all this is; the above equations tell us that only two values ΔF_1 and ΔF_2 are independent. For example, in an experiment suppose we find that 5 mole/sec of carbon has reacted.

$$\text{Then: } \Delta F_1 = -5.$$

We also find that 4 mole/sec of oxygen is consumed.

$$\Delta F_2 = -4.$$

Then using the equations above we find

$$\Delta F_3 = (-2)(-5) + (2)(-4) = 2 \text{ mole/sec is the amount of CO formed.}$$

Also

$$\Delta F_4 = (1)(-5) - (2)(-4) = 3 \text{ mole/sec of } CO_2 \text{ is formed.}$$

If the measurements show otherwise, then either the measurements are in error or some other C bearing species is being formed (not likely here). Hence in this case the measurements would be in error. Thus, the invariants of a reaction are useful in proper bookkeeping of the various species.

A.2 MATLAB PROGRAM FOR CALCULATION OF EQUILIBRIUM COMPOSITION OF A REACTING GAS MIXTURE

The following program calculates the equilibrium composition of a gas mixture. The program is written for the SO₂ example and can be modified easily for other cases. The best way to learn this is for the students to TYPE out the program as it is and execute it on MATLAB. The program also illustrates the use of the solver FSOLVE to solve a set of non-linear algebraic equations. The program is interspaced with some explanation and these statements need to be omitted in the actual program.

PREAMBLE SECTION

% filename gaseq; created on jan 31-03 by P.A.Ramachandran.

% computes the equilibrium composition of a reacting mixture.

% preamble

global ng ns nr tempin prin ctot xg nu ...

global keq298 keq delhr

global rgas

global n ntot

At this part of the program the number of gas phase species ng and number of reactions nr are to be entered.

%number of species and number of reactions

ng = 3 ;

nr = 1 ;

DIMENSIONING THE VECTORS

This is required but no action by user is needed.

% initialize the vectors.

keq298 = zeros(nr, 1);

keq = zeros(nr, 1);

delhr = zeros(nr, 1);

xg = zeros(ng,1);

```
nu = zeros ( nr, ng);
```

```
zeta = zeros(nr,1) ;
```

USER DATA SECTION

A: REACTIONS

Here the reaction specific variables are entered

These are the stoichiometric matrices and the equilibrium constants for each reaction at 298K and the heat of reaction at 298K. The program assumes that the heat of reaction is independent of temperature.

Note that the gas constant, rgas, must have consistent unit with the heat of reaction.-----

% provide values for stoichimetry and eq constants.

```
nu(1,1) = -2. ;
```

```
nu(1,2) = -1.
```

```
nu(1,3) = 2. ;
```

```
keq298(1) = 6.1593E+24 ;
```

```
delhr(1) = -47000.0 ; %assumed constant (not a function of T)
```

```
rgas = 1.97 ; % gas constant; use consistent units
```

B: PROCESS CONDITIONS

Here the temperature, pressure and the moles of each species present in the initial mixture are specified.

The user must provide a guess value for the extent of reaction (zeta vector). Try to provide realistic values, for example the conversion of a key component can not be greater than one. This will fix the maximum value for zeta

% provide the feed conditions.

```
tempin = 873.0;
```

```
prin = 1.0e+05;
```

% initial moles.

```
xg(1) = 2.0;
```

```
xg(2) = 1.0;
```

```
xg(3) = 0.0;
```

% provide initial values for extent of reactions

```
zeta(1) = 0.4
```

CALCULATION SECTION

At this stage the MATLAB takes care of the rest of the calculations

```

% mtlab function fsolve solver is called to find the roots
% the required equations are programmed in a file fun_eq.m
zeta = fsolve ( 'fun_eq', zeta)
% post process the results.
% molar flow rates at exit = TYPE n
% total moles = TYPE ntot

```

Post processing can be done by typing “n” which gives the moles of species at Equilibrium. The conversion and other required information can be calculated easily on the MATLAB command window.

FUNCTION SUBROUTINE

The program requires a function subroutine which calculates the function to be solved. The subroutine is in a file (fun_eq.m) and the listing is shown below. No changes are needed in this file by the user . Students may want to study how this is written by following this with the text earlier.

```

function fvec = fun_eq( zeta)
% filename fun_eq; created on jan 31-03by P.A.Ramachandran.
% Defines the (nr) functions to be solved.

% preamble
global ng ns nr tempin prin ctot xg nu ...
global keq298 keq delhr
global n ntot
global rgas
fvec = zeros ( nr,1) ;

ptotatm = prin /1.0e+05 ;
% find K at the desired conditions
for i = 1: nr

```

```

keq(i)=keq298(i)*exp( delhr(i) /rgas *(1./298-1./tempin)) ;
end
%number of moles of each species for the given extents
% nj0 = 1 for these calculations
for j = 1: ng
    n(j) = xg(j);
    for i=1:nr
        n(j) = n(j) + nu(i,j) * zeta(i);
    end
end
% find ntot
ntot = 0.0;
for j = 1: ng
    ntot = ntot + n(j);
end
% find the mole fractions for each species
pp = n /ntot * ptotatm;
% set up discrepancy function for each reaction
for i = 1: nr
    prod = 1.0;
    for j = 1: ng
        prod = prod * pp(j)^nu(i,j);
    end
fvec(i) = keq(i) - prod
end

```

The students should run the program for other cases. For example to study the effect of inerts, like nitrogen, pressure or other systems involving multiple reactions.

We will now illustrate use of this program for a combustion application.

Example 1: Chemical equilibrium sets an upper limit on the composition of combustion gases. Given a mixture of CO, CO₂ and O₂ we need to find the CO concentration as a function of exhaust gas

temperature. Once CO_2 is formed, it is unlikely that CO will be formed as the gas cools since the reaction rates are slowed down. Hence the maximum CO in the exhaust gas can be found from the equilibrium calculations.

As an example, consider an exhaust gas mixture with 18% CO_2 , 3.3% O_2 and the rest N_2 . Temperature of this gas is raised to 1600K. Find the equilibrium composition of this gas assuming the following reaction:



Solution:

We run the program with 4 species the fourth being inert (N_2). Sample results are as follows:

***** COMPUTED RESULTS *****

- Equilibrium constant of reaction 1: 1.9643907E-05
- Heat of reaction 1: 279958.6 (J/mol)
- Free energy change of reaction 1: 144168.0 (J/mol)
- The extent of reaction 1: 1.9459629E-03 (mole)

We used 100 moles total as the input conditions. Moles of CO in the exhaust gas are therefore 1.955E-03% of the total mole or 19.5 ppm (by moles or volume).

Problem for discussion

Predict the effect of temperature (range of 1000 to 2000K) on the CO content of exhaust gas and plot the results. If CO reduction was the only goal, would you operate at high temperature or low temperature?

Example 2: Consider a gas to be 3.3% O_2 and rest N_2 . This presumably simulates an exhaust gas from an internal combustion engine. The following reactions take place



Find the concentrations of NO and NO_2 at 1800K and 1 atm.

Solution

We will use the FOTRAN program since all these gases are in the chemkin database. We modified the data file and run the program to find the following results.

***** COMPUTED RESULTS *****

- Equilibrium constant of reaction 1: 1.1928983E-04
- Heat of reaction 1: 181059.9 (J/mole)
- Free energy change of reaction 1: 135194.9
- Equilibrium constant of reaction 2: 5.1639900E-03
- Heat of reaction 2: -57594.48 (J/mole)
- Free energy change of reaction 2: 78807.43 (J/mole)
- The extent of reaction 1: 8.5949510E-02 (J/mole)
- The extent of reaction 2: 1.7736210E-04 (J/mole)

$$\text{NO formed} = 2 x_1 - x_2$$

$$\text{N}_2\text{O Formed} = x_2$$

The above are percentages since the total starting moles are assumed as 100. Hence the gas has 172 ppm of NO and 1.77 ppm of NO₂