HUNGARIAN JOURNAL
OF INDUSTRIAL CHEMSITRY
VESZPRÉM
Vol. 19 pp. 289 - 292 (1991)

GENERATING MINIMAL REACTIONS IN STOICHIOMETRY USING LINEAR ALGEBRA*

ISTVÁN SZALKAI

(Dept. Mathematics, University of Veszprém, Hungary)

Received: August 6, 1991

*Presented at the International Workshop on Chemical Engineering Mathematics, Göttingen, D, July 7-11, 1991

We sketch a computer algorithm, which generates all minimal reactions among the elements of a given set species and presents some new computations.

Introduction

In the study of stoichiometric equations and mechanisms both on an undergraduate and research level an important role is played by keeping the minimal reactions (mechanisms) among a given set of chemical species in a firm hand. E.g. in constructing pH-diagrams (KRISTÓF-MIHÁLYKÓ-SALAMON [2]) or in the general treatment of chemical reaction-systems (HAPPEL-OTAROD-SELLERS [1]). As PETHÓ in [6] pointed out, the linear algebraic modelling of the problem is easily applicable for finding dimensionless groups in dimensional analysis or for finding mechanisms, etc.)

The following linear algebraic approach of the problem was presented by Professor PETHÓ at the 2nd North-American/German Workshop on Chemical Engineering Math. & Comp. (1990,Göttingen, Germany) and is described in [6]. (As similar method can be found also in [1].)

Let the chemical species $A_1, \ldots, A_n$ consist of the elements

$$E_1, \ldots, E_m \text{ as } A_j \sum_{i=1}^{m} a_{i,j} E_1 \quad (a_{i,j} \in N) \text{ for } j = 1, \ldots, n.$$

Let us fix the above sets $\{A_1, \ldots, A_n\}$ and $\{E_1, \ldots, E_n\}$. Writing $\mathbf{A_j}$ for the vector $[a_{1,j}, a_{2,j}, \ldots, a_{m,j}]^T$ we know that a chemical reaction might exist between the species $\{A_j : j \in S\}$ for any $S \subseteq \{1,2,\ldots n\}$ if the homogen linear equation:

$$\sum_{j \in S} x_j A_j = 0 \qquad (1)$$

has a nontrivial solution for $x_j \in R$ $(j \in S)$.

A solution $\{x_j : j \in S\}$ easily and uniquely determines a chemical reaction among the species $\{A_j : j \in S\}$.) This means that the vector set $\{\mathbf{A_j} : j \in S\}$ is *linearly dependent.*

The reaction is called minimal if for *no* $T \subset S$, $T \neq S$ there might be any reaction among the species $\{A_j : j \in T\}$. That is, the set $\{A_j : j \in T\}$ *is linearly independent for any* $T \subset S$, $T \neq S$. This motivates the following definition:

DEFINITION ([6]) *A set of vectors* $B = \{\mathbf{b_j} : j \in S\}$ *is called* **a simplex** *if* B *is linearly dependent but the set* $\{\mathbf{b_j} : j \in T\}$ *is linearly independent for any* $T \subset S$, $T \neq S$.

The main result here is to provide an algorithm for finding all the simplexes in a given set of vectors, without repetition. (This problem is pure linear algebraic.)

There is a computer programme mentioned in [3], but we have no facility to compare that one with this yet. The approach of [1] slightly differs from the one described by above. [5] presents a theoretical approach. Of course, the reactions obtained in the above way are only possibilities, since, for example the reaction:

$$2\,Au + 6\,HCl = 2\,AuCl_3 + 3\,H_2$$

does not yield under normal conditions.

The sizes of the simplexes must not be the same, moreover, they do not have any connection with the bases of the space. However, the number of simplexes in a given set of vectors is an open, possibly hard question.

An Example

Consider the species $A_1 = C$, $A_2 = O$, $A_3 = CO$ and $A_4 = CO_2$, that is we use the elements $E_1 = C$ and $E_2 = O$, so $m = 2$ and $n = 4$. Now the columns of the following table represent the vectors $\mathbf{A}_1$ through $\mathbf{A}_4$

	A_1	A_2	A_3	A_4
C	1	0	1	1
O	0	1	1	2
	C	O	CO	CO_2

Since the only simplexes in this example (we write the sub-indexes of the vectors only) are $S_1 = \{1,2,3\}$, $S_2 = \{1,2,4\}$, $S_3 = \{2,3,4\}$ and $S_4 = \{1,3,4\}$, after solving the corresponding *Eq. (1)*, we have the following (complete) list of minimal reactions:

$X_1 = [\ 1,\ 1, -1,\ 0\ ]$: $C + O = CO$ Rutherford
$X_2 = [\ 1,\ 2,\ 0, -1\ ]$: $O + 2\,O = CO_2$ Curie
$X_3 = [\ 0,\ 1,\ 1, -1\ ]$: $O + CO = CO_2$ Nerst
$X_4 = [\ 1,\ 0, -2,\ 1\ ]$: $C + CO_2 = 2\,CO$ Hahn

We can build up (minimal) mechanisms from the above reactions in the same way as the reactions from the species. If the minimal reactions (the solutions of the appropriate (1)) are $\mathbf{X}_1,\ldots,\mathbf{X}_k$, built up from the species $\mathbf{A}_1,\ldots,\mathbf{A}_n$, that is

$$X_j = \sum_{i=1}^{n} b_{i,j} A_i \qquad (b_{i,j} \in N) \quad \text{for } j = 1,\ldots,k\,,$$

then solutions of the linear equations $\sum_{j \in S} y_j X_j = 0$ are called **reactions** (which may be **minimal** for appropriate $S \subseteq \{1,\ldots,k\}$). So we can speak about the first three stages of hierarchy : 1st stage = species , 2nd stage = reactions, 3rd stage = mechanisms .

In our example the co-ordinates of the vectors $\mathbf{X}_1$,...,$\mathbf{X}_k$ are:

	X_1	X_2	X_3	X_4
A_1	1	1	0	1
A_2	1	2	1	0
A_3	−1	0	1	−2
A_4	0	−1	−1	1

In this example we again have 4 simplexes. Solving the corresponding equations, we get the minimal mechanisms:

$Y_1 = [\ -1,\ 0,\ 1,\ 1\ ]$: Phoenix
$Y_2 = [\ -1,\ 1, -1,\ 0\ ]$: Los Angeles
$Y_3 = [\ 2,\ -1,\ 0, -1\ ]$: Norman
$Y_4 = [\ 0,\ -1,\ 2,\ 1\ ]$: Stillwater

The Programme

We wrote a simple computer programme for searching the minimal reactions among given species in PASCAL language. The only interesting part of it might be how to generate (= book-keeping + modifying) all simplexes of a given set of vectors. (Checking whether a chosen set of vectors is simplex or not is standard.) We chose the lexicographic enumeration of all (possible) simplexes (subsets) and the "back-and-forth" method for modifying. To store the elements of a set of vectors (or equivalently the indexes of the vectors) we chose a string (called **simplex[]** in the programme) with an information-character on the last byte of it. (We put the info character to the last position only for convenience. Also for convenience we label the vectors with the characters A,B,... .) The last character may be one of the following:

' ' (= space) - untested
'i' - the whole vector set is independent
'd' - one of the subsets is dependent
's' - simplex

The procedure, which modifies the vector-subset, always puts a space to the last info character, is outlined later. (The main programme keeps testing whether the new vector set is a simplex or not and fills out the last info char according to this, after this it calls this procedure to modify the vector-subset using this procedure and solves (1) for simplexes.)

Now let us see the procedure in more detail. (See the programme list.) Here S will denote the set of (indices of) the vectors with the last information character, that is $S \subset \{1, 2,\ldots, n,\ ' ',\ 'i',\ 'd',\ 's'\}$ (set theorists would call S rather a sequence than a set ...). c always denotes an arbitrary character, k, $t \le n$ and T is any subset of $\{1,2,\ldots, t\text{-}1\}$.

```
   S:={1} ;
while not end do
   begin
      if  S={k, k+1,..., n; c} and  c ≠ d  then  end ;
      if  S={k,k+1,...,n; 'd'} then S:={k ,k+1,...,n-1, n; ' '};
      if  S={T, t, n; c}       then   S:={T, t+1;  ' '};
      if  S={T ,t; 'i'}        then   S:={T, t, t+1; ' '} ;
      if  S={T, t; 'd'}        then   S:={T, t+1; ' '} ;
      if  S={T, t; 's'}        then   S:={T, t+1; ' '} ;
   end ;
```

Outputs

The [4] contains some small and large examples, which can be used for testing the programme. However, there are more untested systems of species. Some outputs are also enclosed, including Professor PETHŐ's questions. The programme was written in Turbo Pascal and run on an IBM PC/XT at 10 MHz . The 213 minimal reactions for the system {CO, CO_2, O_2, H_2 , CH_2O , CH_3OH , C_2H_5OH , $(CH_3)_2CO$, CH_4, CH_3CHO, H_2O} were obtained in less than 5 minutes.

Acknowledgements

The author is grateful for Professor PETHŐ for presenting the problem and supporting this research.

Appendix

```
PROCEDURE MODIFY ;
        {modifying the vector - list }

var     hossz, elso, utolso: integer;

label   ret;

begin
   hossz:= length (szimplex);
   elso:= ord(szimplex[1]) - 64;
   utolso:=ord (szimplex [ hossz-1]) - 64;
   if szimplex [hossz] = ' ' then go to ret; { not checked }
   if ( elso + hossz - 2 = utolso ) and ( utolso = meretx )
   and ( 'd' <> szimplex [ hossz] ) then
      begin {end sequence }
         vege:= true;{programme ends }
         go to ret;
      end;
   if ( elso + hossz - 2 = utolso ) and ( utolso = meretx )
      and ( 'd' <> szimplex [ hossz] ) then
      {end sequence }
      begin
         utolso:= ord (szimplex) [hossz-2]) + 1;
                                        {new last item}
         sstr:= copy (szimplex, 1, hossz-3 ) +
         char (utolso) +  ' ' ;
         go to ret;
      end;
   if utolso=meretx then
      {end reached }
      begin
         utolso:= ord (szimplex) [hossz-2]) + 1;
                                        {new last item }
         sstr:= copy (szimplex, 1, hossz-3 ) ;
         szimplex:= sstr + chr (utolso ) + ' ' ;
         go to ret;
      end;
   if szimplex [ hossz ]:= 'i' then
      { independent ⇒ enlarging )
      begin
         szimplex [ hossz ] := chr ( utolso + 64 + 1 );
         szimplex:= szimplex + ' ';
         go to ret;
   end;
   if szimplex [ hossz ]:= 'd' then
      { dependent ⇒ new last item )
      begin
         szimplex [ hossz ] := char ( ord (szimplex
          [ hossz - 1 ]) + 1 );
         szimplex [ hossz] :=  ' ';
         go to ret;
      end;
   if szimplex [ hossz ]:= 's' then
      { simplex  ⇒ test another )
      begin
         szimplex [ hossz-1 ] := char(ord (szimplex
          [ hossz - 1 ] )+ 1 ); { last enlarging }
         szimplex [ hossz] :=  ' ';
         go to ret;
      end;
ret: end ; { proc. modify }
```

An Output

The species:
1st speci: H_2
2nd speci: O_2
3st speci: HO
4th speci: HO_2
5th speci: H_2O
6th speci: H_2O_2

The system matrix:
1st row: 2, 0, 1, 1, 2, 2,
2nd row: 0, 2, 1, 2, 1, 2,

The minimal reactions:

1. $+\frac{1}{2}H_2 + \frac{1}{2}O_2 - 1HO = 0$
2. $+\frac{1}{2}H_2 + 1O_2 - 1HO_2 = 0$
3. $+ 1H_2 + \frac{1}{2}O_2 - 1H_2O = 0$
4. $+ 1H_2 + 1O_2 - 1H_2O_2 = 0$
5. $-\frac{1}{2}H_2 + 2HO_2 - 1HO_2 = 0$
6. $+\frac{1}{2}H_2 + 1HO - 1H_2O = 0$
7. $+\frac{3}{4}H_2 + \frac{1}{2}HO_2 - 2H_2O = 0$
8. $+\frac{1}{2}H_2 + 1HO_2 - 1H_2O_2 = 0$
9. $- 1H_2 + 2H_2O - 1H_2O_2 = 0$
10. $+\frac{1}{2}O_2 + 1HO - 1HO_2 = 0$
11. $+\frac{1}{2}O_2 + 2HO - 1H_2O = 0$
12. $+\frac{3}{2}O_2 + 2HO_2 - 1H_2O = 0$
13. $- 1O_2 + 2HO_2 - 1H_2O_2 = 0$
14. $+\frac{1}{2}O_2 + 1H_2O - 1H_2O = 0$
15. $+ 3OH - 1HO_2 - 1H_2O = 0$
16. $+ 2OH - 1H_2O_2 = 0$
17. $+\frac{2}{3}OH_2 + \frac{2}{3}H_2O - 1H_2O_2 = 0$

REFERENCES

1. HAPPEL,J., OTAROD,M. and SELLERS, P. H.: Ind. Eng. Chem. Res. *1990*, 29, 1057
2. KRISTÓF,T., MIHÁLYKÓ, Cs. and SALAMON,T.: Construction of equilibrum potential - pH diagrams in complex aqueous systems, submitted to Corrosion Science
3. KUMAR, S. and PETHÓ.: Á.: Intern. Chem. Eng. *1985*, 5, 767
4. PETHÓ, Á.: "Lectures on linear methods in chemical engineering mathematics (stoichiometry)", Preprint, 1990, Institut für Technische Chemie, Univ. Hannover, Germany
5. PETHÓ, Á.: Acta Math. Hungaricae *1967* , 18, 19
6. PETHÓ, Á.: Chem. Eng. Technol., *1990* , 13, 328.