AFCRL-67-0221
APRIL 1967
ENVIRONMENTAL RESEARCH PAPERS, NO. 263

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES
L. G. HANSCOM FIELD, BEDFORD, MASSACHUSETTS

A Technique for Solving the General Reaction-Rate Equations in the Atmosphere

THOMAS J. KENESHEA

OFFICE OF AEROSPACE RESEARCH
United States Air Force

AFCRL-67-0221
APRIL 1967
ENVIRONMENTAL RESEARCH PAPERS, NO. 263

UPPER ATMOSPHERE PHYSICS LABORATORY PROJECT 8605

AIR FORCE CAMBRIDGE RESEARCH LABORATORIES

L. G. HANSCOM FIELD, BEDFORD, MASSACHUSETTS

A Technique for Solving the General Reaction-Rate Equations in the Atmosphere

THOMAS J. KENESHEA

This research was performed under Program Element 6 16 46 01 D Project 5710, subtask 07 010 and was funded in part by the Defense Atomic Support Agency

OFFICE OF AEROSPACE RESEARCH
United States Air Force

Abstract

With the availability of numerical techniques for solving an extensive set of nonlinear differential equations and high-speed computers for performing the calculations, interest in solving the unrestricted reaction-rate equations is growing among ionospheric researchers. In view of this, the author has continued to refine the techniques that he previously developed.

The computer code, as discussed here, is written to solve the photochemical behavior of 15 atmospheric species; these species are electrons, O^-, O_2^-, O_3^-, NO_2^-, O^+, O_2^+, N_2^+, NO^+, NO, N, NO_2, O_3, N_2O, and O. Built into the code are 168 reactions that can conceivably take place among these constituents. Several examples of the results obtained using the code are presented, including the buildup of ionization from zero concentrations at altitudes in the D and E regions and the deionization of an atmosphere with high initial electron densities. The diurnal variation of the atmospheric constituents is also presented along with profiles for the above-mentioned species from 60 km to 120 km.

The computer codes are included in their entirety with complete explanations on their usage.

Contents

Tables

Tables

A Technique for Solving the General Reaction-Rate Equations in the Atmosphere

1. INTRODUCTION

The capability of solving the set of differential equations that describe the time dependence of individual atmospheric constituents is becoming more and more important to ionospheric research. Modification and extension of the computer techniques that were developed by the author (1962, 1963) have continued because of this. Since in certain applications these techniques could not generate solutions and since these applications are important to ionospheric studies, the techniques were reevaluated in order to make their usefulness more general.

The most significant change made in the code is the handling of the solution for a species after it has gone into equilibrium or into quasi-equilibrium with one or more other species. The techniques involved in developing a solution under these conditions have always presented problems. The algebraic equations that were used in previous work for the computation of the concentrations of the species in equilibrium or quasi-equilibrium proved to be inadequate for this purpose. A new approach that lead to a set of exponential equations which replaced the set of algebraic equations was developed involving less stringent restrictions. With this new technique, no code failures were experienced.

(Received for publication 27 February 1967)

Another important change is in the method of numerically integrating the differential equations. Although it develops stable solutions, the Runge Kutta technique used previously is uneconomical in terms of computer time. In order to reduce the amount of computer time required for each solution, the Runge Kutta technique was replaced by the Kutta Merson technique. This later method requires fewer computations of the derivatives over each increment thereby conserving computer time.

Other changes include removal of the sum equations used by the author (1963) since the reason for their being employed was eliminated with the new technique for solving the concentrations of the species in equilibrium, the computation of the largest negative species from charge balance rather than the largest positive species, the use of a separate code to write the subprogram SLOP which computes the values of the derivatives, and the inclusion in the code of two additional species (nitrogen peroxide and atomic oxygen) along with 25 new reactions.

The updated codes are presented here with several practical applications. These examples include the simple buildup of ionization from zero by an external source, the deionization from high initial conditions and the diurnal variation of the atmospheric species in the D and E regions.

2. REVISIONS IN THE TECHNIQUES OF SOLUTION

Since the last report by the author (1963) was written, attempts were made to use the code under diversified conditions. In several of these instances, the code either failed to generate realistic solutions or was unable to generate any solution at all; the source of this trouble was always traced back to the solution of the algebraic set of equations. This set of simultaneous algebraic equations was used to compute the concentrations of the species that are in quasi-equilibrium. Using a criterion of 10^{-2} on the iterations of the algebraic set allowed a certain amount of charge imbalance to be accepted. Because the major positive ion was always being computed from the requirement of balance of charge, any charge imbalance resulting from the solution of the algebraic equations was attributed to this positive ion. As long as this positive ion remained the most abundant throughout the solution, this allowed imbalance did not cause any noticeable problems. Whenever another positive ion became the most abundant, however, the program generally was incapable of advancing the solution. Once a positive ion was superceded as the dominant ion, it was no longer computed from the requirement of charge balance but rather from its differential equation. Since the concentration of this ion had taken up the excess charge that was allowed to creep into the solution, the value of its density was not consistent with that obtained from the solution of its differential equation. The program tried to correct

this situation but the damage was irreparable and the program was forced to maintain a constant mesh of the order of a microsecond.

The simple act of tightening the criterion on the iterations of the algebraic set to 10^{-4} and thus reducing the allowed charge imbalance removed this problem from practical concern. The resulting greater accuracy in the solution of the algebraic equations also permitted the removal of the two sum equations used in the previous code. This change did not solve all of the problems, however.

Occasions arose where it appeared that at some point in the solution neither the differential set nor the algegraic set of equations could generate a solution. The code advanced the solution to a time at which it determined that a species was in quasi-equilibrium. It removed its differential equation from the set and expected a solution from the algebraic set of equations. However, the simultaneous solution of the algebraic equations did not converge in the fixed number of iterations. The code then demanded that the differential equation for this species be used for the solution. In order to obtain a solution from the differential equation at this point, however, it was necessary to reduce the integrating increment. The overall effect of this behavior was to allow the solution to advance but only in very small increments.

This difficulty was attributed to the fact that the species was in quasi-equilibrium and not in true equilibrium. In other words, its derivative was not zero as assumed in deriving the algebraic equations. If the differential equations are written as

$$\frac{dN_i}{dt} = \sum F_j - N_i \sum R_j \qquad (1)$$

and if $\frac{dN_i}{dt} = 0$, then

$$N_i = \frac{\sum F_j}{\sum R_j}. \qquad (2)$$

This is the formula previously used to construct the set of algebraic equations. Because the derivative was not near zero, it could not be expected that the concentration could be computed from Eq.(2). In order to overcome this problem, the assumption that the derivative is zero was replaced by the more realistic assumption that the formation term $\sum F_j$ and the removal term $\sum R_j$ in Eq.(1) are constants. Equation (1) under these circumstances is a first order linear differential equation with constant coefficients and its exact solution is

$$N_i = \left(N_o - \frac{\sum F_j}{\sum R_j} \right) e^{-\sum R_j \cdot \Delta t} + \frac{\sum F_j}{\sum R_j} \qquad (3)$$

where Δt is the length of the increment currently being used by the integrator and N_o is the concentration of the species at the beginning of this increment. Equation (3) is now used to construct the set of simultaneous exponential equations for the solution of the species that are in quasi-equilibrium.

The technique used by the author (1962, 1963) for numerically integrating the differential equations is the classical Runge Kutta fourth-order process. One of the disadvantages of this technique is that it does not contain within itself any measure of the accuracy of the solution at each integration. In order to impose some control on the accumulation of errors, three separate integrations are performed for a given increment. The first integration is made over the increment Δt starting with the concentrations $N_i(t_o)$; the second integration is made over the increment Δt starting with the concentrations $N_i(t_o + \Delta t)$; and, the third integration is made over the increment $2\Delta t$ starting with the concentrations $N_i(t_o)$. The concentrations obtained after the second and third integration are compared; if they differ by more than some preset amount, the solution is considered to be invalid. This procedure requires 12 separate computations of the derivatives. The differential equations solved here generally contain a very large number of terms which means that a great deal of computer time is spent in computing the derivatives. This is very costly of machine time since in many cases the solution is unacceptable and the procedure must be repeated with a smaller increment.

Merson (1957) developed a technique based upon that of Kutta in which only five values of the derivative are required for any one solution. Since it appeared that this could be a great device for reducing the amount of computer time required for each solution, the Runge Kutta technique was replaced by the Kutta Merson technique. A detailed description of this method is given in Section 7.1.2.

3. CHEMICAL REACTIONS AND REACTION-RATE CONSTANTS

The computer code, as presented here, was written to evaluate the time histories of 15 variable atmospheric species: electrons, O^-, O_2^-, O_3^-, NO_2^-, O^+, O_2^+, N_2^+, NO^+, NO, N, NO_2, O_3, N_2O, and atomic oxygen. The model contains reservoirs of molecular oxygen and molecular nitrogen that are allowed to vary during

the solution in order to insure conservation of the total number of oxygen and nitrogen atoms initially present in the system.

The code contains 168 chemical reactions that can possibly take place among the variable species. With a few exceptions, the chemical processes and their rate constants incorporated into the code are those suggested by Bortner (1965). These rates are a combination of laboratory, insitu, theoretical, and estimated values. Although individual rate constants may certainly be subject to argument, they appear to be a reasonable collection and about the only suitable source to use in a program of this magnitude. The values used for the rate constants are generally the middle values within the range of uncertainty (Bortner, 1965). Deviations are made from this general rule in certain instances as in the case of the positive ion-charge transfer and charged rearrangement processes (Ferguson, Fehsenfeld, Goldan, and Schmeltokopf, 1965). Certain other rate constants were modified within the estimated error bounds (Bortner, 1965). These modifications represent minor adjustments that were found to produce results appearing to be more reasonable when compared with ionospheric measurements.

Some of the reactions in the following list produced products for which there are no differential equations in the set. In order to insure that these reactions do not remove from the system any of the charge for which account cannot be made, their rate constants are set to zero. This is equivalent to assuming that the reactions are relatively unimportant. There are also several reactions in the list that are probably unimportant in most ionospheric studies; but, since it is generally impossible to predetermine the importance of a given reaction and since it is the author's purpose to keep the code as general as possible, all reactions are retained in the code.

The following is the basic list of reactions and rate constants built into the code. The dimensions are sec^{-1}, $cm^3 sec^{-1}$, and $cm^6 sec^{-1}$ respectively for one, two, and three body reactions. The third body M is assumed to be $N_2 + O_2$. The temperatures used in computing the rate constants are taken from the U.S. Standard Atmosphere (1962). Those reactions marked with an asterisk have assumed rate constants other than those shown in some of the applications presented here.

A. PHOTODETACHMENT

1.	$O_2^- + h\nu \longrightarrow O_2 + e$	0.44
2.	$O^- + h\nu \longrightarrow O + e$	1.4
3.	$NO_2^- + h\nu \longrightarrow NO_2 + e$	0.04
*4.	$O_3^- + h\nu \longrightarrow O_3 + e$	0.04

B. COLLISIONAL DETACHMENT

5.	$O_2^- + O_2 \longrightarrow O_2 + O_2 + e$	$9 \times 10^{-15} T^2 e^{-(5100/T)}$
6.	$O_2^- + N_2 \longrightarrow O_2 + N_2 + e$	$3.6 \times 10^{-16} T^2 e^{-(5100/T)}$
7.	$O_2^- + O \longrightarrow O_2 + O + e$	$3.6 \times 10^{-16} T^2 e^{-(5100/T)}$

C. ASSOCIATIVE DETACHMENT

8.	$O^- + O \longrightarrow O_2 + e$	1×10^{-13}
9.	$O^- + O_2 \longrightarrow O_3 + e$	$1 \times 10^{-13} e^{-(4700/T)}$
10.	$O^- + N \longrightarrow NO + e$	1×10^{-13}
11.	$O^- + N_2 \longrightarrow N_2O + e$	1×10^{-13}
12.	$O^- + NO \longrightarrow NO_2 + e$	1×10^{-13}
13.	$O^- + O_3 \longrightarrow O_2 + O_2 + e$	1×10^{-13}
14.	$O_2^- + N \longrightarrow NO_2 + e$	1×10^{-13}
15.	$O_2^- + O \longrightarrow O_3 + e$	1×10^{-13}
*167.	$O_3^- + O \longrightarrow O_2 + O_2 + e$	1×10^{-13}
168.	$NO_2^- + O \longrightarrow O_2 + NO + e$	1×10^{-13}

D. RADIATIVE ATTACHMENT

16.	$O + e \longrightarrow O^- + h\nu$	1.31×10^{-15}
17.	$O_2 + e \longrightarrow O_2^- + h\nu$	1×10^{-19}
18.	$NO_2 + e \longrightarrow NO_2^- + h\nu$	1×10^{-17}
19.	$O_3 + e \longrightarrow O_3^- + h\nu$	1×10^{-17}

E. DISSOCIATIVE ATTACHMENT

20.	$O_3 + e \longrightarrow O^- + O_2$	1×10^{-11}
21.	$O_3 + e \longrightarrow O_2^- + O$	$1 \times 10^{-11} e^{-(7200/T)}$

F. THREE-BODY ATTACHMENT

22.	$O + e + O_2 \longrightarrow O^- + O_2$		1×10^{-31}
23.	$O + e + N_2 \longrightarrow O^- + N_2$		1×10^{-31}
24.	$O_2 + e + O_2 \longrightarrow O_2^- + O_2$		$1.4 \times 10^{-31} T^{1/2}$
25.	$O_2 + e + N_2 \longrightarrow O_2^- + N_2$		$5.8 \times 10^{-33} T^{1/2}$
26.	$O_2 + e + O \longrightarrow O_2^- + O$		$1.9 \times 10^{-33} T^{1/2}$
27.	$NO_2 + e + O_2 \longrightarrow NO_2^- + O_2$		6×10^{-28}
28.	$NO_2 + e + N_2 \longrightarrow NO_2^- + N_2$		4×10^{-29}
29.	$NO + e + M \longrightarrow NO^- + M$		0

G. DISSOCIATIVE RECOMBINATION

30.	$O_2^+ + e \longrightarrow O + O$		$6 \times 10^{-5} T^{-1}$
31.	$N_2^+ + e \longrightarrow N + N$		$9 \times 10^{-5} T^{-1}$
*32.	$NO^+ + e \longrightarrow N + O$		$1.5 \times 10^{-4} T^{-1}$

H. THREE-BODY RECOMBINATION

33.	$O^+ + e + M \longrightarrow O + M$		$1 \times 10^{-24} T^{-3/2}$
34.	$O_2^+ + e + M \longrightarrow O_2 + M$		$1 \times 10^{-22} T^{-3/2}$
35.	$N_2^+ + e + M \longrightarrow N_2 + M$		$1 \times 10^{-22} T^{-3/2}$
36.	$NO^+ + e + M \longrightarrow NO + M$		$1 \times 10^{-22} T^{-3/2}$
37.	$NO^+ + e + M \longrightarrow N + O + M$		$1 \times 10^{-23} T^{-3/2}$

I. RADIATIVE RECOMBINATION

38.	$O^+ + e \longrightarrow O + h\nu$		$2.2 \times 10^{-10} T^{-0.7}$
39.	$O_2^+ + e \longrightarrow O_2 + h\nu$		1×10^{-12}
40.	$N_2^+ + e \longrightarrow N_2 + h\nu$		1×10^{-12}
41.	$NO^+ + e \longrightarrow NO + h\nu$		1×10^{-12}

J. ION-ION MUTUAL NEUTRALIZATION

42.	$O^- + O^+ \longrightarrow O + O$		$5 \times 10^{-7} T^{-1/2}$
43.	$O_2^- + O^+ \longrightarrow O_2 + O$		$5 \times 10^{-7} T^{-1/2}$
44.	$NO_2^- + O^+ \longrightarrow NO_2 + O$		$5 \times 10^{-7} T^{-1/2}$

45.	O_3^-	+ O^+	⟶	O_3 + O		$5\times10^{-7}T^{-1/2}$
46.	O^-	+ O_2^+	⟶	O + O_2		$5\times10^{-7}T^{-1/2}$
47.	O_2^-	+ O_2^+	⟶	O_2 + O_2		$5\times10^{-7}T^{-1/2}$
48.	NO_2^-	+ O_2^+	⟶	NO_2 + O_2		$5\times10^{-7}T^{-1/2}$
49.	O_3^-	+ O_2^+	⟶	O_3 + O_2		$5\times10^{-7}T^{-1/2}$
50.	O^-	+ N_2^+	⟶	O + N_2		$5\times10^{-7}T^{-1/2}$
51.	O_2^-	+ N_2^+	⟶	O_2 + N_2		$5\times10^{-7}T^{-1/2}$
52.	NO_2^-	+ N_2^+	⟶	NO_2 + N_2		$5\times10^{-7}T^{-1/2}$
53.	O_3^-	+ N_2^+	⟶	N_2 + O_3		$5\times10^{-7}T^{-1/2}$
54.	O^-	+ NO^+	⟶	O + NO		$5\times10^{-7}T^{-1/2}$
55.	O_2^-	+ NO^+	⟶	O_2 + NO		$5\times10^{-7}T^{-1/2}$
56.	NO_2^-	+ NO^+	⟶	NO_2 + NO		$5\times10^{-7}T^{-1/2}$
57.	O_3^-	+ NO^+	⟶	O_3 + NO		$5\times10^{-7}T^{-1/2}$

K. THREE-BODY ION-ION RECOMBINATION

58.	O^-	+ O^+ + N	⟶	O_2 + N		$1\times10^{-23}T^{-3/2}$
59.	O^-	+ O^+ + O_2	⟶	O_2 + O_2		$1\times10^{-23}T^{-3/2}$
60.	O^-	+ O^+ + N_2	⟶	O_2 + N_2		$1\times10^{-23}T^{-3/2}$
61.	O^-	+ O^+ + O	⟶	O_2 + O		$1\times10^{-23}T^{-3/2}$
62.	O_2^-	+ O^+ + M	⟶	O_3 + M		$1\times10^{-23}T^{-3/2}$
63.	O^-	+ O_2^+ + M	⟶	O_3 + M		$1\times10^{-23}T^{-3/2}$
64.	O_2^-	+ O_2^+ + M	⟶	O_2 + O_2 + M		$1\times10^{-23}T^{-3/2}$
65.	O^-	+ N_2^+ + M	⟶	N_2O + M		$1\times10^{-23}T^{-3/2}$
66.	O_2^-	+ N_2^+ + M	⟶	O_2 + N_2 + M		$1\times10^{-23}T^{-3/2}$
67.	O^-	+ NO^+ + M	⟶	NO_2 + M		$1\times10^{-23}T^{-3/2}$
68.	O_2^-	+ NO^+ + M	⟶	O_2 + NO + M		$1\times10^{-23}T^{-3/2}$

L. ION-ION NEUTRALIZATION WITH REARRANGEMENT

69.	O_2^-	+ N_2^+	⟶	NO	+ NO	1×10^{-11}
70.	O_2^-	+ N_2^+	⟶	N_2O	+ O	1×10^{-13}
71.	O_2^-	+ N_2^+	⟶	NO_2	+ N	1×10^{-13}
72.	O_2^-	+ N_2^+	⟶	2N	+ 2O	0
73.	NO_2^-	+ N_2^+	⟶	N_2O	+ NO	1×10^{-13}
74.	O_3^-	+ N_2^+	⟶	N_2O	+ O_2	1×10^{-13}
75.	O_2^-	+ NO^+	⟶	NO_2	+ O	1×10^{-11}
76.	O_2^-	+ NO^+	⟶	N	+ O_3	1×10^{-11}
77.	O_2^-	+ NO^+	⟶	N	+ O_2 + O	1×10^{-13}
78.	NO_2^-	+ NO^+	⟶	NO_3	+ N	0
79.	NO_2^-	+ NO^+	⟶	NO	+ O_2 + N	1×10^{-13}
80.	NO_2^-	+ NO^+	⟶	O_3	+ N_2	$1 \times 10^{-13} e^{-(5000/T)}$
81.	O_3^-	+ NO^+	⟶	O_2	+ O_2 + N	1×10^{-13}
82.	O_3^-	+ NO^+	⟶	O_2	+ NO_2	1×10^{-13}

M. POSITIVE CHARGE TRANSFER

83.	O^+	+ O_2	⟶	O	+ O_2^+	4×10^{-11}
84.	O^+	+ NO	⟶	O	+ NO^+	2.4×10^{-11}
85.	O^+	+ NO_2	⟶	O	+ NO_2^+	0
86.	O^+	+ N_2O	⟶	O	+ N_2O^+	0
87.	O_2^+	+ NO	⟶	O_2	+ NO^+	8×10^{-10}
88.	N_2^+	+ O	⟶	N_2	+ O^+	1×10^{-12}
89.	N_2^+	+ O_2	⟶	N_2	+ O_2^+	1×10^{-10}
90.	N_2^+	+ N	⟶	N_2	+ N^+	0
*91.	N_2^+	+ NO	⟶	N_2	+ NO^+	5×10^{-12}

N. NEGATIVE CHARGE TRANSFER

92.	$O^- + NO_2 \longrightarrow O + NO_2^-$		1×10^{-9}
93.	$O^- + O_3 \longrightarrow O + O_3^-$		1×10^{-9}
94.	$O_2^- + O \longrightarrow O_2 + O^-$		1×10^{-12}
95.	$O_2^- + NO_2 \longrightarrow O_2 + NO_2^-$		1×10^{-9}
96.	$O_2^- + O_3 \longrightarrow O_2 + O_3^-$		1×10^{-9}
97.	$NO_2^- + O_3 \longrightarrow NO_2 + O_3^-$		1×10^{-9}
98.	$O_3^- + NO_2 \longrightarrow O_3 + NO_2^-$		1×10^{-9}

O. ION-NEUTRAL ASSOCIATION-TWO BODY

99.	$O^+ + O \longrightarrow O_2^+ + h\nu$		1×10^{-16}
100.	$O^+ + N \longrightarrow NO^+ + h\nu$		1×10^{-18}
101.	$O^+ + N_2 \longrightarrow N_2O^+ + h\nu$		0
102.	$O^+ + NO \longrightarrow NO_2^+ + h\nu$		0
103.	$O_2^+ + O \longrightarrow O_3^+ + h\nu$		0
104.	$N_2^+ + N_2 \longrightarrow N_4^+ + h\nu$		0
105.	$NO^+ + O \longrightarrow NO_2^+ + h\nu$		0
106.	$O^- + O_2 \longrightarrow O_3^- + h\nu$		1×10^{-15}

P. ION-NEUTRAL ASSOCIATION-THREE BODY

107.	$O^+ + O + M \longrightarrow O_2^+ + M$		$1 \times 10\text{-}29$
108.	$O^+ + N + M \longrightarrow NO^+ + M$		1×10^{-29}
109.	$O^+ + N_2 + M \longrightarrow N_2O^+ + M$		0
110.	$O^+ + NO + M \longrightarrow NO_2^+ + M$		0
111.	$O_2^+ + O + M \longrightarrow O_3^+ + M$		0
112.	$NO^+ + O + M \longrightarrow NO_2^+ + M$		0
113.	$NO^+ + N + M \longrightarrow N_2O^+ + M$		0
114.	$O^- + NO + M \longrightarrow NO_2^- + M$		1×10^{-30}
115.	$O_2^- + N + M \longrightarrow NO_2^- + M$		1×10^{-30}

Q. CHARGED REARRANGEMENT-POSITIVE ION

116.	O^+	+	N_2	⟶	NO^+	+ N	3×10^{-12}
117.	O^+	+	NO	⟶	O_2^+	+ N	3×10^{-12}
118.	O_2^+	+	N	⟶	O^+	+ NO	$3 \times 10^{-12} e^{-(2000/T)}$
119.	O_2^+	+	N_2	⟶	NO^+	+ NO	$1 \times 10^{-11} e^{-(3000/T)}$
120.	O_2^+	+	N	⟶	NO^+	+ O	1.8×10^{-10}
121.	O_2^+	+	NO_2	⟶	NO^+	+ O_3	1×10^{-11}
122.	N_2^+	+	O	⟶	NO^+	+ N	2.5×10^{-10}
123.	N_2^+	+	O_2	⟶	NO^+	+ NO	$1 \times 10^{-11} e^{-(3500/T)}$

R. CHARGED REARRANGEMENT-NEGATIVE ION

124.	O^-	+	O_2 + O_2	⟶	O_3^-	+ O_2	$1 \times 10^{-28} T^{-1}$
125.	O_2^-	+	O_2 + N_2	⟶	NO_2^-	+ NO_2	$1 \times 10^{-34} e^{-(5000/T)}$
126.	O_3^-	+	N_2	⟶	NO_2^-	+ NO	$1 \times 10^{-17} e^{-(4000/T)}$

S. TWO-BODY ATOM RECOMBINATION

127.	O	+	O	⟶	O_2	+ $h\nu$	1×10^{-21}
128.	O	+	O_2	⟶	O_3	+ $h\nu$	1×10^{-21}
129.	O	+	N	⟶	NO	+ $h\nu$	2×10^{-17}
130.	O	+	N_2	⟶	N_2O	+ $h\nu$	1×10^{-24}
131.	O	+	NO	⟶	NO_2	+ $h\nu$	6.4×10^{-17}
132.	N	+	N	⟶	N_2	+ $h\nu$	1×10^{-24}
133.	N	+	O_2	⟶	NO_2	+ $h\nu$	1×10^{-22}
134.	N	+	NO	⟶	N_2O	+ $h\nu$	1×10^{-22}
*135.	O	+	O + O_2	⟶	O_2	+ O_2	$5 \times 10^{-32} T^{-1/2}$
*136.	O	+	O + O	⟶	O_2	+ O	$2 \times 10^{-31} T^{-1/2}$
*137.	O	+	O + N_2	⟶	O_2	+ N_2	$5 \times 10^{-32} T^{-1/2}$
*138.	O	+	O_2 + O_2	⟶	O_3	+ O_2	$3.2 \times 10^{-35} e^{+(900/T)}$
*139.	O	+	O_2 + N_2	⟶	O_3	+ N_2	$2.6 \times 10^{-35} e^{+(900/T)}$
140.	O	+	O_2 + O	⟶	O_3	+ O	6.5×10^{-34}

141.	N	+ O + M	⟶	NO	+ M	$2\times10^{-31}T^{-1/2}$
142.	O	+ N_2 + M	⟶	N_2O	+ M	$2\times10^{-33}e^{-(20000/T)}$
143.	O	+ NO + O_2	⟶	NO_2	+ O_2	$3\times10^{-33}e^{+(900/T)}$
144.	O	+ NO + N_2	⟶	NO_2	+ N_2	$4.5\times10^{-33}e^{+(900/T)}$
145.	N	+ N + M	⟶	N_2	+ M	$3\times10^{-30}T^{-1}$
146.	N	+ O_2 + M	⟶	NO_2	+ M	$1\times10^{-33}e^{-(3000/T)}$
147.	N	+ NO + M	⟶	N_2O	+ M	$1\times10^{-33}e^{-(10000/T)}$

U. NEUTRAL REARRANGEMENT

148.	NO	+ O_2 + NO	⟶	NO_2	+ NO_2	0
149.	O	+ N_2	⟶	NO	+ N	$1.1\times10^{-10}e^{-(37500/T)}$
150.	O	+ NO	⟶	O_2	+ N	$7.1\times10^{-17}T^{1.5}e^{-(19000/T)}$
151.	O	+ NO_2	⟶	NO	+ O_2	$3\times10^{-11}e^{-(530/T)}$
152.	O	+ N_2O	⟶	NO	+ NO	$2\times10^{-10}e^{-(14000/T)}$
153.	O	+ N_2O	⟶	O_2	+ N_2	$5\times10^{-11}e^{-(13500/T)}$
154.	O	+ O_3	⟶	O_2	+ O_2	$5\times10^{-10}e^{-(2800/T)}$
155.	N	+ O_2	⟶	NO	+ O	$3\times10^{-16}T^{1.5}e^{-(6600/RT)}$
156.	N	+ NO	⟶	N_2	+ O	2.5×10^{-11}
157.	N	+ NO_2	⟶	N_2	+ O_2	$2\times10^{-13}e^{-(7000/T)}$
158.	N	+ NO_2	⟶	NO	+ NO	4×10^{-12}
159.	N	+ NO_2	⟶	N_2O	+ O	2×10^{-11}
160.	NO	+ O_3	⟶	NO_2	+ O_2	$8\times10^{-13}e^{-(1200/T)}$

V. PHOTODISSOCIATION

161.	O_2	+ hν	⟶	O	+ O	5×10^{-6}
162.	NO	+ hν	⟶	N	+ O	6×10^{-8}
163.	NO_2	+ hν	⟶	NO	+ O	3×10^{-3}
164.	N_2O	+ hν	⟶	N_2	+ O	4.08×10^{-7}
165.	N_2O	+ hν	⟶	N	+ NO	5.58×10^{-8}
166.	O_3	+ hν	⟶	O	+ O_2	5.3×10^{-3}

4. APPLICATION OF THE CODE TO THE DEIONIZATION PROBLEM

The code, as described in Section 7.1, was used to generate solutions to the differential equations under many different conditions. Four examples of a simple application were selected and the results obtained are presented here. Two of these describe the buildup of ionization from zero concentrations of the charged species with a continuing constant source at altitudes in the D and E regions. The remaining two describe the decay of ionization at these same altitudes from high initial values of electrons and positive ions with a small continuing source. The altitudes chosen are 70 km and 110 km.

Some changes were made in the basic list of rate constants given in Section 3. Furguson, et al. (1965) have determined that the rate of the charge transfer reaction $N_2^+ + NO \longrightarrow NO^+ + N_2$ is much faster than previously thought. Their value of 5×10^{-10} is used for this process instead of 5×10^{-12}. For reasons that are discussed in Section 5.3, the NO^+ dissociative recombination rate constant was changed to $6 \times 10^{-5} T^{-1}$ to make it equal to the O_2^+ dissociative recombination rate constant. The rate constants for the three-body recombination reactions Nos. 135, 136, and 137 were changed to 2.7×10^{-33} and the rate constants for reactions Nos. 138 and 139 were changed to $3.7 \times 10^{-34} e^{-151}$. Since these five reactions are important processes for the removal of atomic oxygen, adjustment of their rate constants was necessary in order to obtain a reasonable behavior of the atomic-oxygen concentration in the D region. In the examples presented here, the associative detachment reaction $O_3^- + O \longrightarrow O_2 + O_2 + e$ is important for the establishment of the negative ion to electron ratio in the D region. With all other important rate constants remaining unchanged, the rate constant for this process was increased to 5×10^{-11} in order to obtain a ratio of unity at 70 km. All other rate constants used in runs to be discussed in this section are as given in the above list.

Figure 1 shows the solution at 70 km with zero initial concentrations of all the changed species and with estimated initial concentrations of the minor neutral species. The continuing source creates 1.122 ion-pairs/cm^3/sec. The production subroutine was set up to ionize the neutral species as if the source were UV radiation. Therefore, O_2^+ is the most abundantly produced positive ion. For this demonstration of the functioning of the computer code, it is immaterial that the assumed production was unrealistic because in the D region L_α and cosmic rays are actually the ionizing agents.

The atomic oxygen concentration remains constant until 10^{-2} secs when the photodissociation of O_2 and O_3 becomes effective in producing this atom. The time constant for its removal by the three-body recombination reaction

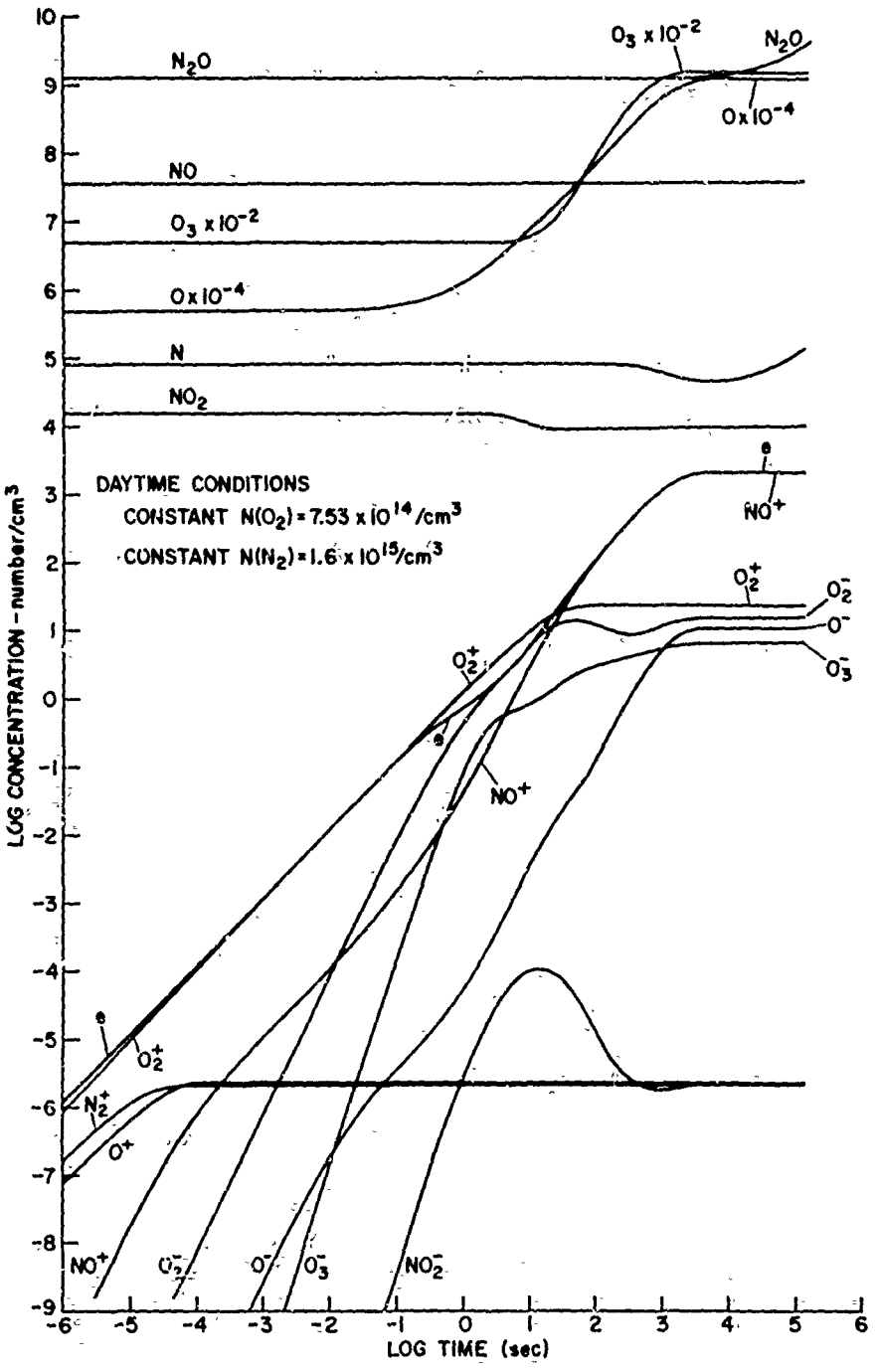

Figure 1. Solution at 70 km with Zero Initial Conditions on the Charged Species and a Constant Source of 1.122 ion-pairs/cm^3/sec

$O + O_2 + M \longrightarrow O_3 + M$ is about 3×10^3 secs. After this time, the formation rate by photodissociation and the removal rate by recombination become equal, causing the atomic oxygen concentration to go into equilibrium.

The ozone concentration cannot change until enough of these molecules are formed by the three-body recombination reaction $O + O_2 + M \longrightarrow O_3 + M$ to start increasing the number present. They are removed by photodissociation with a time constant of about 100 secs. After this time, the recombination rate equals the photodissociation rate causing the ozone concentration to level off at a value established by the equilibrium concentration of atomic oxygen.

The time constants for the removal of N_2O are all greater than 10^5 secs. However, by 10^4 secs, the two-body atom recombination reaction $O + N_2 \longrightarrow N_2O + h\nu$ has formed a sufficient number of these molecules to start increasing its concentration. Its density increases by this process and continues to do so until the removal processes become effective after 10^5 secs.

The time constant for the removal of nitric oxide by the three-body recombination reaction $O + NO + M \longrightarrow NO_2 + M$ is about 10^4 secs. Before any NO molecules can be removed, however, their production by the neutral rearrangement reaction $O + NO_2 \longrightarrow NO_2 + O_2$ becomes effective. Around 10^4 secs the rates of these two processes become equal which is the reason for there being no change in the NO concentration. The atomic nitrogen concentration remains constant until about 100 secs, the time constant for the removal process $N + O \longrightarrow NO + h\nu$, when its concentration starts to decrease. When the rate of this removal reaction becomes equal to the rate of formation of atomic nitrogen by the photodissociation of N_2O, the N concentration goes into quasi-equilibrium with N_2O.

There is no effective removal process for NO_2 molecules at early times. The time constant for their removal by the neutral rearrangement reaction $O + NO_2 \longrightarrow NO + O_2$ is about 2 secs. By 10 secs, however, their rate of production by the three-body recombination $O + NO + M \longrightarrow NO_2 + M$ is sufficient to balance their removal rate thus preventing any further change in their concentration.

The electron density increases with unit slope at early times as electrons are formed by the constant source. By 0.1 sec, 0.1122 electrons are produced. During this time, the O_2^- concentration increases with slope 2 as this ion is formed by the three-body attachment reaction $O_2 + e + O_2 \longrightarrow O_2^- + O_2$. The time constant for the removal of electrons by this three-body attachment is about 0.8 sec. It was expected that O_2^- photodetachment become effective around this time, causing a balance between the attachment and detachment and putting the O_2^- density into quasi-equilibrium with the electron density. This quasi-equilibrium effect is clearly seen in Figure 7 of Keneshea (1963). The time constant for photodetachment is about 2 secs, however, so that the production of O_2^- by attachment continues, causing the electron density to decrease. Another important removal process for

O_2^- ions is the charge transfer reaction $O_2^- + O_3 \longrightarrow O_3^- + O_2$. Because of the increasing O_3 density after 10 secs, it is not possible to fix a definite time constant to this reaction. The final equilibrium between the electrons and the O_2^- ions is determined by the ozone equilibrium concentration. This behavior can be understood if the O_2^- to electron ratio is determined from equilibrium considerations. The O_2^- differential equation at equilibrium is essentially

$$\frac{dN(O_2^-)}{dt} = -N(O_2^-)\left[N(O_3) \cdot k_{96} + k_1\right] + N(O_2)^2 \cdot N(e) \cdot k_{24} = 0 \quad . \tag{4}$$

From Eq. (4), the ratio of O_2^- to electrons is

$$\frac{N(O_2^-)}{N(e)} = \frac{N(O_2)^2 \cdot k_{24}}{N(O_3) \cdot k_{96} + k_1} \quad . \tag{5}$$

It is obvious that this ratio depends only on the ozone concentration. After 100 secs, the electron concentration returns to a linear increase resulting from the production by the source and goes into equilibrium when the rate of production by the source equals the rate of removal by dissociative recombination with NO^+.

The O^- concentration increases at early times with a slope of 2 as this ion is formed by the dissociative attachment reaction $O_3 + e \longrightarrow O^- + O_2$. The time constant for the associative detachment reaction $O^- + N_2 \longrightarrow N_2O + e$ is about 6×10^{-3} secs around which time removal of this ion becomes effective. As the concentration of O_2^- increases, the charge transfer reaction $O_2^- + O \longrightarrow O^- + O_2$ eventually becomes effective in forming O^- ions. Around 1 sec, the density of this ion goes into quasi-equilibrium with the electrons and O_2^- as the rate of formation by the dissociative attachment and the charge transfer equals the removal rate of the associative detachment. Because of its quasi-equilibrium status, the O^- concentration continues to increase after 1 sec. Around 100 secs, the concentrations of ozone and O^- have reached values that make the removal of this ion by the charge transfer reaction $O^- + O_3 \longrightarrow O_3^- + O$ become important. The final equilibrium level of O^- and its ratio to the electron density are determined by the equilibrium value of ozone.

At early times, the O_3^- concentration increases with a slope of 3 being formed by the charge transfer reaction $O_2^- + O_3 \longrightarrow O_3^- + O_2$. Before the O density changes, the time constant for the removal of this ion by the associative detachment reaction $O_3^- + O \longrightarrow 2O_2 + e$ is about 2 secs. As the concentration of atomic oxygen increases, this time constant becomes smaller. When both atomic oxygen and ozone reach equilibrium values, the O_3^- concentration goes into equilibrium at a ratio to the O_2^- density that is established by the atomic oxygen and the ozone concentrations.

The NO_2^- concentration increases at early times with a slope of 3 since it is formed by the charge transfer reaction $O_2^- + NO_2 \longrightarrow NO_2^- + O_2$. As long as the ozone concentration is constant, the time constant for the removal of this ion by the charge transfer reaction $NO_2^- + O_3 \longrightarrow NO_2 + O_3^-$ is about 2 secs. This is the same time that the ozone concentration starts to increase, causing the rate of the charge transfer to O_3^- to become much larger than the rate of the charge transfer to NO_2^-. This causes the NO_2^- concentration to decay. This decay continues until the ozone concentration reaches a constant value, causing the NO_2^- density to go into equilibrium at a fixed ratio to the O_2^- density.

The concentrations of both O^+ and N_2^+ increase linearly at early times as these ions are formed by the constant source. The time constant for the removal of O^+ ions by the charge transfer reaction $O^+ + O_2 \longrightarrow O_2^+ + O$ is about 3×10^{-5} secs while the time constant for the removal of N_2^+ ions by the charge transfer reaction $N_2^+ + O_2 \longrightarrow O_2^+ + N_2$ is about 1×10^{-5} secs. Around these times, therefore, the charge transfer rates become equal to the rates of production of these ions by the source and their concentrations go into equilibrium. All O^+ and N_2^+ ions produced by the source after this time immediately transfer their charge to form O_2^+ ions.

The O_2^+ concentration increases at early times because of the production by the constant source and the transfer of charge from the N_2^+ ions as the latter are formed by the constant source. The charged rearrangement reaction $O_2^+ + N_2 \longrightarrow NO^+ + NO$ has a time constant of about 50 secs. After this time, the O_2^+ goes into equilibrium at a value that is determined by the balance between the source function and the charged rearrangement reaction.

The NO^+ concentration builds up with a slope of 2 at very early times through its formation by the charged rearrangement reaction $O^+ + N_2 \longrightarrow NO^+ + N$. When O^+ goes into equilibrium, the slope of the NO^+ profile changes to 1. By 0.1 sec, enough O_2^+ ions are formed to make the charge transfer process $O_2^+ + NO \longrightarrow NO^+ + O_2$ become more important in the formation of NO^+ ions, causing the slope to change back to 2. When O_2^+ goes into equilibrium, the NO^+ concentration reverts to unit slope increase. The time constant for dissociative recombination is about 10^3 secs after which time the production rate of NO^+ by charge transfer from O_2^+ equals the removal rate, causing the NO^+ concentration to go into equilibrium.

Figure 2 shows the solution at 110 km with zero initial concentrations of the ionized species and estimated initial concentrations of the minor neutral species. There is a continuing constant source producing ion-pairs at the rate of $2 \times 10^3/cm^3/sec$.

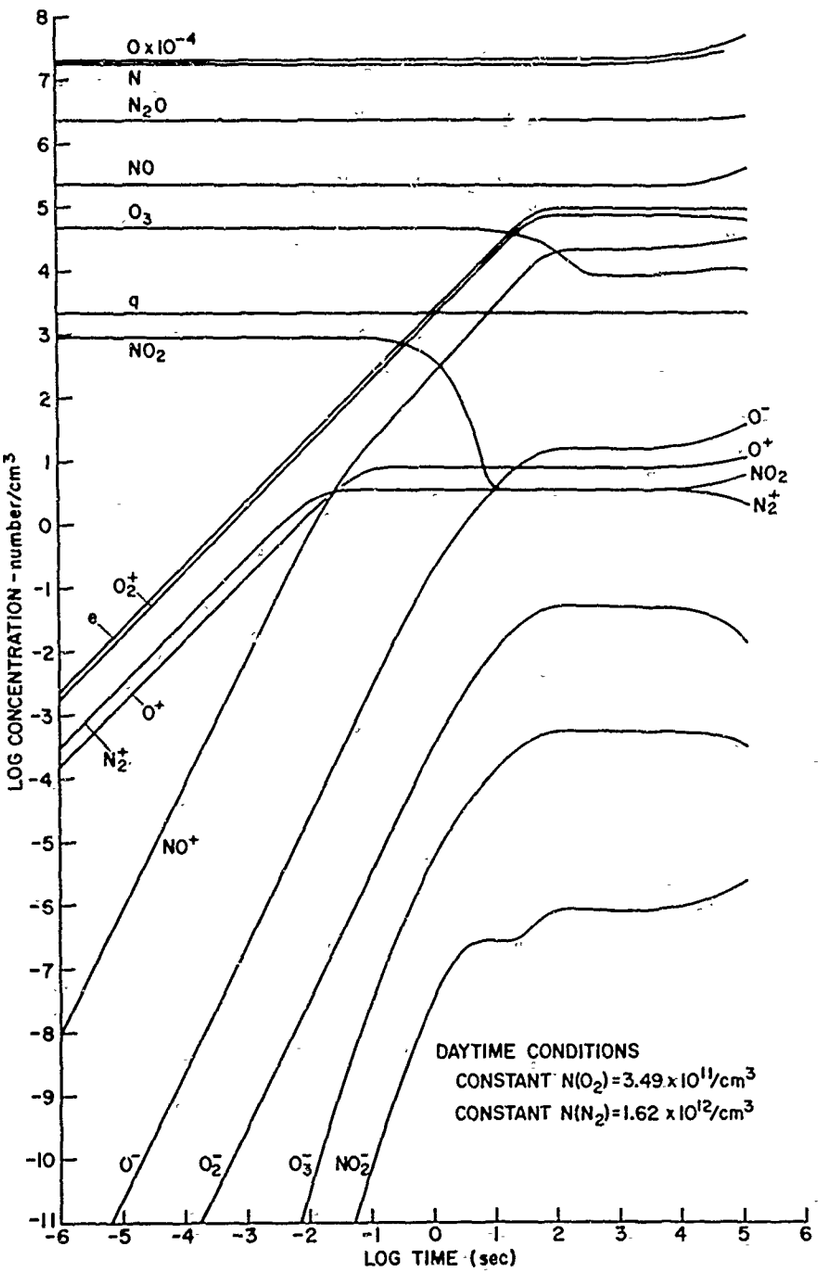

Figure 2. Solution at 110 km with Zero Initial Conditions on the Charged Species and a Constant Source of 2×10^3 ion-pairs/cm^3/sec

The time constants of the processes that remove atomic oxygen are greater than 10^5 secs. The increase in the density of this atom at late times results from the photodissociation of O_2. The time constants of the processes that remove N, NO, and N_2O are all greater than 10^5 secs. The atomic nitrogen concentration increases at late times because the dissociative recombination of NO^+ has started to form a sufficient number of these atoms to start increasing their concentration. The associative detachment reaction $O^- + N_2 \longrightarrow N_2O + e$ has formed about 10^6 N_2O molecules in 10^5 secs which accounts for the slight increase in their concentration at very late times. The increase in the NO density at late times results from the increasing N and O concentrations in the two body atom recombination reaction $N + O \longrightarrow NO + h\nu$.

The ozone concentration remains constant until about 100 secs when O_3 starts to be removed by photodissociation. The time constant for the two-body atom recombination reaction $O + O_2 \longrightarrow O_3 + h\nu$ is about 600 secs. After this time, the ozone concentration goes into equilibrium which is established by the atomic-oxygen concentration. The increase in O_3 at late times is caused by the increasing oxygen supply for the two-body atom recombination reaction. The time constant for the neutral rearrangement reaction $O + NO_2 \longrightarrow NO + O_2$ is about 1 sec. After this time, the NO_2 concentration decays until the rate of removal is reduced to the rate of formation by the two-body atom recombination reaction $O + NO \longrightarrow NO_2 + h\nu$ when the NO_2 concentration remains at the equilibrium value established by these reactions.

The electron concentration increases at early times with unit slope as they are formed by the constant source. By 1 sec, 2×10^3 electrons are formed. The time constant for the O_2^+ dissociative recombination is about 50 secs around which time the rate of production by the source and the rate of removal by dissociative recombination become equal and the electron concentration goes into equilibrium. The most important formation process for O^- ions is the radiative attachment reaction $O + e \longrightarrow O^- + h\nu$. The O^- concentration increases with a slope of 2 as this ion is formed by this process. The time constant for O^- photodetachment is about 0.75 sec when their removal starts to become effective and they assume a constant ratio to the electrons. The increase in the O^- concentration at late times reflects the dependence of the equilibrium value of this ion on the atomic-oxygen concentration.

At early times, the O_2^- concentration increases with a slope of 2 as it is formed by the three-body attachment reaction $O_2 + e + M \longrightarrow O_2^- + M$. There are two important removal processes for this ion both with time constants of about 2 secs. These are the photodetachment and the charge transfer reaction $O_2^- + O \longrightarrow O^- + O_2$. Around this time O_2^- goes into quasi-equilibrium with the

electrons and assumes a fixed ratio to them. At very late times, the increasing atomic-oxygen concentration increases the rate of removal of O_2^- by the charge transfer, causing the O_2^- concentration to decrease and thereby change the ratio established between O_2^- and the electrons.

The O_3^- ions are formed primarily by the two-body ion-neutral association reaction $O^- + O_2 \rightarrow O_3^- + h\nu$. The O_3^- density increases with a slope of 3 at early times as they are formed by this process. The time constant for the associative detachment reaction $O_3^- + O \rightarrow O_2 + e + O_2$ is about 0.1 sec after which time it becomes effective in removing O_3^- ions and causes their concentration to remain at a constant ratio to the electrons. This ratio is no longer constant after 10^4 secs because the rate of the associative detachment reaction increases with the increasing atomic-oxygen concentration.

The NO_2^- concentration increases with a slope of 3 at early times as it is formed by the charge transfer reaction $O^- + NO_2 \rightarrow NO_2^- + O$. Changes in slope are dictated by the O^- and NO_2 curves. The time constant for NO_2^- photodetachment is about 25 secs after which time quasi-equilibrium with O^- is established.

The O^+ and N_2^+ concentrations increase with unit slope at early times through their formation by the constant source. The time constant for the charge transfer reaction $O^+ + O_2 \rightarrow O_2^+ + O$ is about 0.07 sec. After this time, the removal of O^+ ions becomes effective and quickly equalizes the production by the source forcing the O^+ concentration into equilibrium. Removal of N_2^+ ions starts around 0.02 sec which is the time constant for their removal by the charged rearrangement reaction $N_2^+ + O \rightarrow NO^+ + N$. The equilibrium value of N_2^+ is the result of the balance achieved between the source and the charged rearrangement. The decay of N_2^+ after 10^4 secs is simply the result of the increasing rate of the charged rearrangement reaction resulting from the increasing atomic-oxygen concentration. Any further production of O^+ and N_2^+ ions by the source after they have gone into equilibrium is immediately transferred into O_2^+ ions.

The O_2^+ concentration increases with unit slope until about 50 secs when dissociative recombination starts to remove them. By 100 secs, the O_2^+ concentration reaches an equilibrium determined by the source and the dissociative recombination. At late times, the charged rearrangement reaction $O_2^+ + N \rightarrow NO^+ + O$ becomes important to the equilibrium of O_2^+. The decrease in O_2^+ after 10^4 secs is caused by the increased rate of this reaction as the density of atomic nitrogen increases. The major source of NO^+ ions at early times is the charged rearrangement reaction $N_2^+ + O \rightarrow NO^+ + N$. The NO^+ concentration increases with a slope of 2 until N_2^+ goes into equilibrium at which time the slope of the NO^+ profile changes to unity. Around 50 secs, the charged rearrangement reaction

$O_2^+ + N \longrightarrow NO^+ + O$ becomes the important formation process for NO^+ ions. This is also the time that the dissociative recombination becomes effective in removing these ions. The equilibrium value of NO^+ is determined by the balance between the charged rearrangement reaction and the dissociative recombination. The increase in the NO^+ concentration at late times results from the increased production by the charged rearrangement reaction as the atomic-nitrogen density increases.

Figure 3 shows the solution at 70 km assuming high initial concentrations, such as during a nuclear blackout, for the electrons, O^+, O_2^+, and N_2^+, zero concentrations of the other charged species and estimated initial concentrations for the minor neutral species. A small continuing source of 0.1 ion-pairs/cm^3/sec is used. The time constant for dissociative recombination is very short (about 0.2 sec) because of the high initial concentrations of electrons and O_2^+ ions. The electrons remain constant until this time and they start to decay by dissociative recombination with O_2^+ and three-body attachment to O_2.

The O_2^- concentration increases linearly from early times by the three-body attachment reaction $O_2 + e + O_2 \longrightarrow O_2^- + O_2$. This increase continues until about 1 sec which is the time constant for O_2^- photodetachment. The charge-transfer reactions $O_2^- + O \longrightarrow O^- + O_2$ and $O_2^- + O_3 \longrightarrow O_3^- + O_2$ also become effective in removing O_2^- after 1 sec. By 50 secs, the O_2^- concentration has gone into quasi-equilibrium with the electrons, O, and O_3. The behavior of the negative ions after they have gone into quasi-equilibrium can be understood by looking at their equilibrium equations. The differential equation for O_2^- in equilibrium can be written as

$$\frac{dN(O_2^-)}{dt} = -N(O_2^-) \cdot \left[k_1 + N(O) \cdot k_{94} + N(O_3) \cdot k_{96} \right] + N^2(O_2) \cdot N(e) \cdot k_{24} = O \quad . \tag{6}$$

Solving Eq. (6) for the ratio $N(O_2^-)/N(e)$ gives

$$\frac{N(O_2^-)}{N(e)} = \frac{N^2(O_2) \cdot k_{24}}{(k_1 + N(O) \cdot k_{94} + N(O_3) \cdot k_{96})} \quad . \tag{7}$$

As long as the O and O_3 concentrations are increasing, this ratio becomes smaller and smaller assuming a constant value only when O and O_3 go into equilibrium.

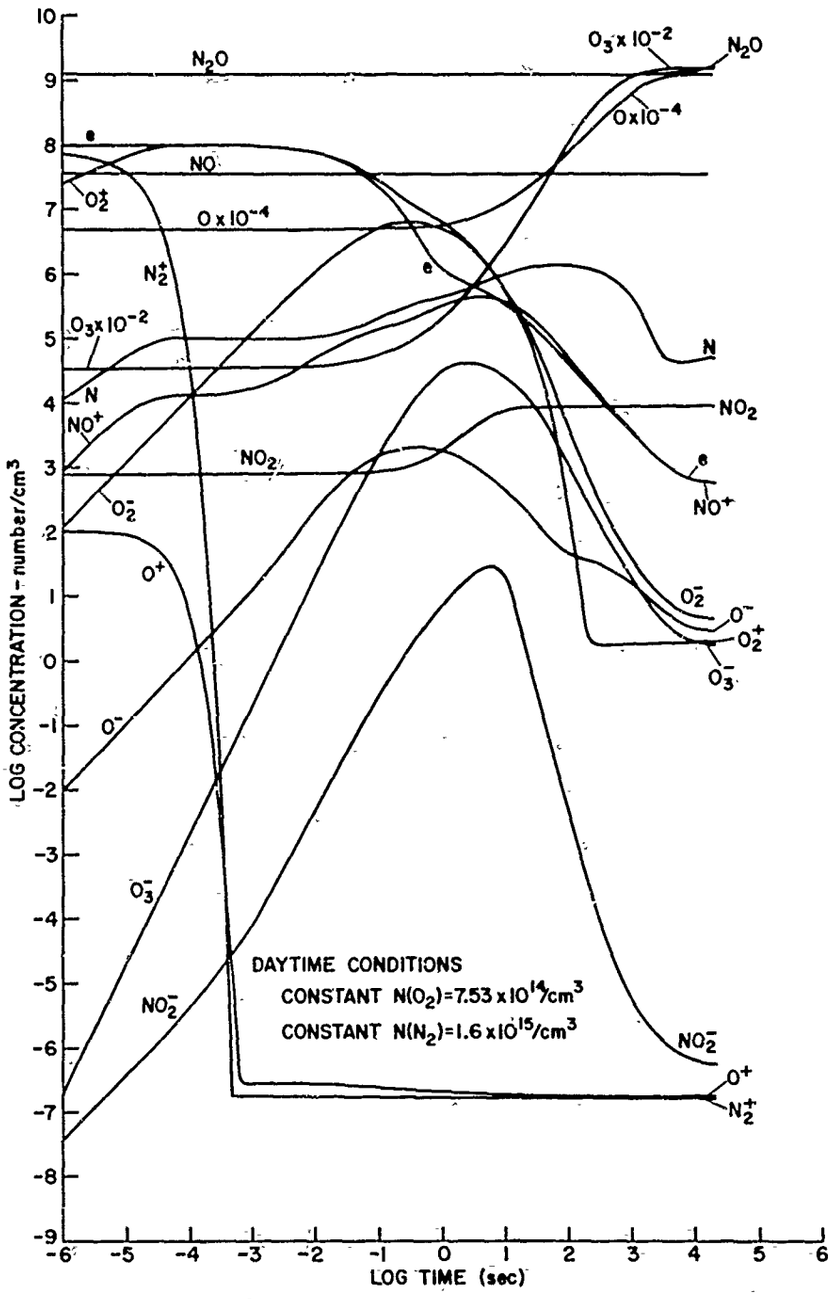

Figure 3. Solution at 70 km with High Initial Electron Concentration

The O^- concentration increases with unit slope at early times as this ion is formed by the associative detachment reaction $O_3 + e \rightarrow O^- + O_2$. The formation of O^- ions by the charge transfer reaction $O_2^- + O \rightarrow O^- + O_2$ becomes effective around 10^{-3} secs. The time constant for removal of O^- ions by the associative detachment reaction $O^- + N_2 \rightarrow N_2O + e$ is about 10^{-2} secs. This reaction starts to consume this ion after this time and by 1 sec the O^- concentration goes into quasi-equilibrium with the electrons, O_2^-, O, and O_3. The differential equation for O^- at equilibrium is

$$\frac{dN(O^-)}{dt} = -N(O^-) \cdot \left[N(N_2) \cdot k_{11} + N(O_3) \cdot k_{93} \right] + N(O_3) \cdot N(e) \cdot k_{20} + N(O_2^-) \cdot N(O) \cdot k_{24} = O \quad . \tag{8}$$

Solving Eq. (8) for the ratio $N(O^-)/N(e)$ gives

$$\frac{N(O^-)}{N(e)} = \frac{N(O_3) \cdot k_{20} + \frac{N(O_2^-)}{N(e)} \cdot N(O) \cdot k_{94}}{N(N_2) \cdot k_{11} + N(O_3) \cdot k_{93}} \quad . \tag{9}$$

The ratio $N(O^-)/N(e)$ is not a simple one but it is obvious that it depends only upon the O and O_3 concentrations and becomes constant only when O and O_3 go into equilibrium.

The O_3^- concentration increases with a slope of 2 at the beginning as this ion is formed by the charge transfer reaction $O_2^- + O_3 \rightarrow O_3^- + O_2$. The time constant for the associative detachment reaction $O_3^- + O \rightarrow 2O_2 + e$ is about 1 sec. Shortly after this time, the detachment rate becomes equal to the charge transfer rate, causing the O_3^- concentration to go into quasi-equilibrium with the electrons, O, and O_3. The differential equation for O_3^- at equilibrium can be written as

$$\frac{dN(O_3^-)}{dt} = -N(O_3^-) \cdot N(O) \cdot k_{167} + N(O_2^-) \cdot N(O_3) \cdot k_{96} = O \, . \tag{10}$$

Solving Eq. (10) for the ratio $N(O_3^-)/N(e)$ gives

$$\frac{N(O_3^-)}{N(e)} = \frac{N(O_3) \cdot k_{96} \cdot N^2(O_2) \cdot k_{24}}{N(O) \cdot k_{167} \left[K_1 + N(O) \cdot k_{94} + N(O_3) \cdot k_{96} \right]} \quad . \tag{11}$$

This ratio depends only upon the concentration of O and O_3 and becomes constant when O and O_3 go into equilibrium.

The NO_2^- ions are formed at early times by the three-body attachment reaction $NO_2 + e + O_2 \longrightarrow NO_2^- + O_2$. By 10^{-3} secs, there are enough O_2^- ions around to make the charge transfer reaction $O_2^- + NO_2 \longrightarrow NO_2^- + O_2$ more important in forming NO_2^- ions accounting for the change of slope to 2. The time constant for the charge transfer reaction $NO_2^- + O_3 \longrightarrow NO_2 + O_3^-$ is about 3 secs. Shortly after this time the NO_2^- concentration goes into quasi-equilibrium with O_3^-, O_2^-, and O_3. The differential equation for NO_2^- at equilibrium is

$$\frac{dN(NO_2^-)}{dt} = -N(NO_2^-) \cdot N(O_3) \cdot k_{97} + N(O_3^-) \cdot N(NO_2) \cdot k_{98} + N(O_2^-) \cdot N(NO_2) \cdot k_{95} = O \quad . \tag{12}$$

Solving Eq. (12) for the ratio $N(NO_2^-)/N(O_2^-)$ gives

$$\frac{N(NO_2^-)}{N(O_2^-)} = \frac{\frac{N(O_3^-)}{N(O_2^-)} \cdot N(NO_2) \cdot k_{98} + N(NO_2) \cdot k_{95}}{N(O_3) \cdot k_{97}} \quad . \tag{13}$$

From Eq. (13) it is obvious that the ratio $N(NO_2^-)/N(O_2^-)$ is solely a function of NO_2, O, and O_3 and that this ratio becomes constant only when NO_2, O, and O_3 go into equilibrium.

The O^+ and N_2^+ concentrations decay at early times by charge transfer to O_2. The concentrations of both of these ions continues to decay until the rate of removal is reduced to the small rate of production by the source function and they go into equilibrium. The O_2^+ concentration increases at very early times as these ions are formed by charge transfer from N_2^+. This increase stops when the N_2^+ is effectively consumed. The time constant for O_2^+ dissociative recombination is about 0.2 sec around which time these ions start to be removed. The mutual neutralization reaction $O_2^- + O_2^+ \longrightarrow O_2 + O_2$ and the charge transfer reaction $O_2^+ + NO \longrightarrow NO^+ + O_2$ both have time constants of about 2 secs when they become effective in removing the O_2^+ ions. These processes continue to remove O_2^+ until its concentration goes into equilibrium when the rates of removal equal the small rate of production by the constant source.

NO^+ ions are formed at early times by the charge transfer reaction $N_2^+ + NO \longrightarrow NO^+ + N_2$. When the supply of N_2^+ is used up, NO^+ is formed at about equal rates by the charge transfer reaction $O_2^+ + NO \longrightarrow NO^+ + O_2$ and the charged rearrangement reaction $O_2^+ + N_2 \longrightarrow NO^+ + NO$. The time constant for NO^+ dissociative recombination is about 0.1 sec around which time removal of NO^+ becomes effective. At about 3 secs removal and production of NO^+ become equal. The production of NO^+ becomes constant around 200 secs when O_2^+ goes into equilibrium.

After this time the NO^+ concentration decays by dissociative recombination along with the electron concentration until the removal rate by recombination equals the production rate from O_2^+ and the NO^+ concentration goes into equilibrium.

The atomic nitrogen concentration increases at early times as a product of the charged rearrangement reaction $N_2^+ + O \rightarrow NO^+ + N$. When N_2^+ has decayed away this production of N atoms stops. By 10^{-2} secs, enough N atoms are formed as a product of the NO^+ dissociative recombination to start increasing the N concentration. The neutral rearrangement reaction $N + NO \rightarrow N_2 + O$ has a time constant of about 10^3 secs. Around this time, the production of N atoms by dissociative recombination has slowed down because of the decaying NO^+ concentration allowing these atoms to be consumed by the neutral rearrangement. At very late times, NO^+ dissociative recombination ceases to be the important formation mechanism for N atoms because of the relatively low NO^+ concentration. The important formation process for N atoms becomes the photodissociation of N_2O which has formed about 5×10^4 atoms in 10^4 secs. The atomic nitrogen concentration after 10^4 secs, therefore, is determined by the neutral rearrangement and the photodissociation.

The time constant for the removal of NO_2 by the neutral rearrangement reaction $O + NO_2 \rightarrow NO + O_2$ is about 5 secs. The NO_2 concentration remains constant up to about 0.1 sec by which time the three-body atom recombination reaction $O + NO + M \rightarrow NO_2 + M$ has formed about 10^2 of these molecules. This is sufficient for the NO_2 concentration to show an increase. By 10 secs, the neutral rearrangement balances the three-body atom recombination forcing the NO_2 concentration into equilibrium. The fact that O is involved both in the production and removal processes makes NO_2 independent of O.

The behavior of N_2O, O_3, and atomic oxygen are essentially the same as shown in Figure 1.

Figure 4 shows the behavior at 110 km of initial high ionization and a very small continuing source. Initially there are 10^8 electrons/cm^3 with equal positive ionization partitioned among O^+, O_2^+, and N_2^+ according to the relative amounts of the corresponding neutral species present. The negative ions and NO^+ have zero initial concentrations while the minor neutral species start out at estimated initial concentrations. For convenience, a small constant source of 0.1 ion-pairs/cm^3/sec is used.

The time constant for O_2^+ and NO^+ dissociative recombination is about 0.25 sec because of the high ion and electron concentrations. The electron density remains constant until about 0.25 sec, therefore, and decays by the simple recombination law after this time. Around 10^4 secs, the rate of removal of electrons by recombination equals the rate of formation by the small continuing source, causing the electron concentration to go into equilibrium. The O_2^+ concentration remains

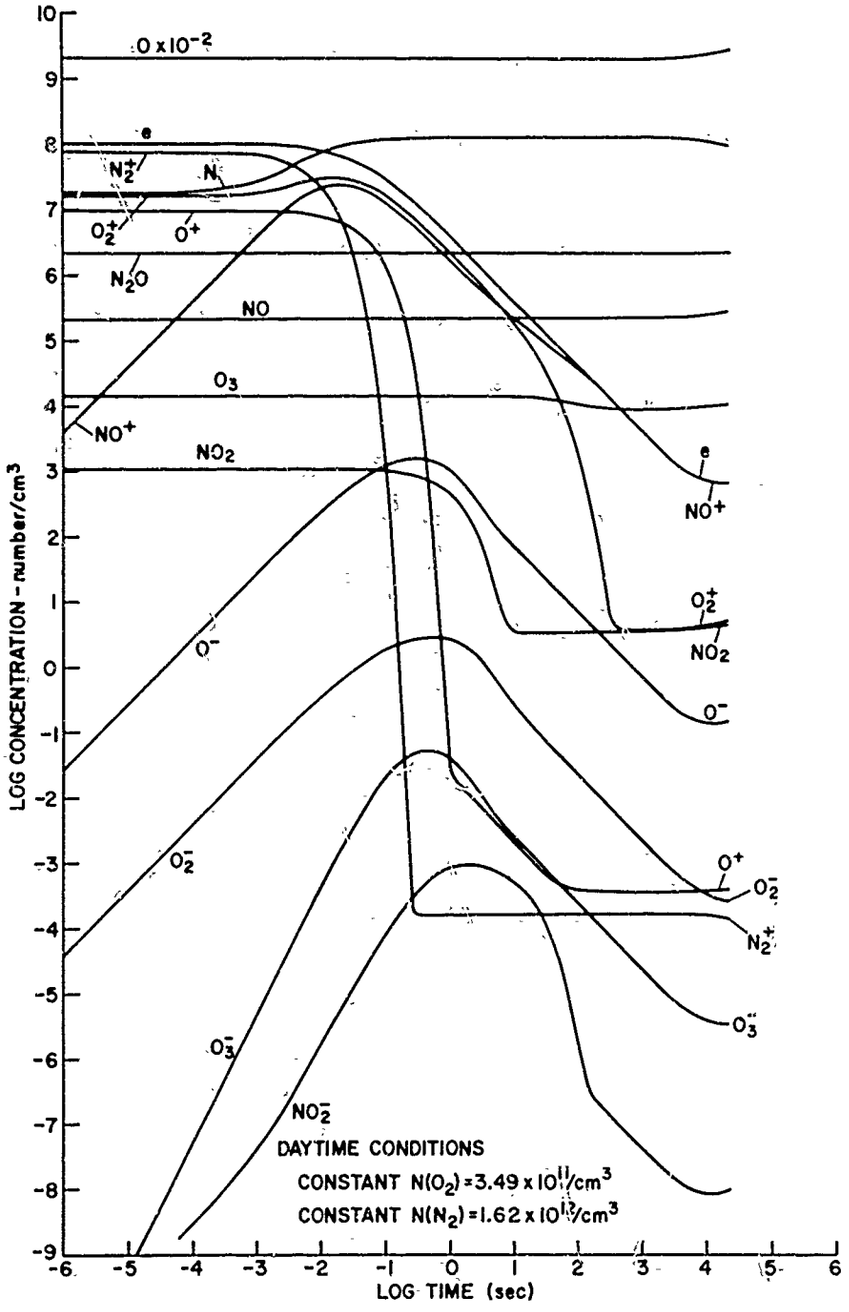

Figure 4. Solution at 110 km with High Initial Electron Concentrations

constant at early times with a slight increase aroung 10^{-2} secs because of the increased production rate of the charge transfer reaction $N_2^+ + O_2 \longrightarrow O_2^+ + N_2$. Around 0.25 sec, dissociative recombination becomes effective in removing this ion, causing it to go into a recombination-type decay. By 50 secs, however, the time constant for O_2^+ dissociative recombination has become rather long because of the much lower electron concentration at this time. The time constant for the charged rearrangement reaction $O_2^+ + N \longrightarrow NO^+ + O$ is about 50 secs. Around this time, therefore, the removal of O_2^+ is accomplished by the charged rearrangement which has become faster than the dissociative recombination. The O_2^+ concentration goes into equilibrium when the rate of removal by charged rearrangement equals the rate of formation by the small constant source. The NO^+ concentration increases with unit slope at the start of the solution as this ion is formed by the charged rearrangement reaction $N_2^+ + O \longrightarrow NO^+ + N$. This increase continues until about 0.25 sec which is the time constant for NO^+ dissociative recombination. After this time, the NO^+ concentration decays by the simple recombination law. Around 10 secs, the production of NO^+ by the charged rearrangement reaction $O_2^+ + N \longrightarrow NO^+ + O$ becomes effective. When the O_2^+ density is sufficiently reduced, the charged rearrangement reaction becomes ineffectual and the NO^+ decay returns to the recombination law.

The time constant for the removal of O^+ ions by the charge transfer reaction $O^+ + O_2 \longrightarrow O_2^+ + O$ is about 0.07 sec around which time the O^+ concentration starts to decay. This removal continues until about 1 sec when the charged rearrangement reaction $O_2^+ + N \longrightarrow O^+ + NO$ becomes effective in forming this ion and around 7 secs the O^+ concentration goes into quasi-equilibrium. The differential equation for O^+ at equilibrium can be written as

$$\frac{dN(O^+)}{dt} = -N(O^+) \cdot \left[N(O_2) \cdot k_{83} + N(N_2) \cdot k_{116} \right] + N(O_2^+) \cdot N(N) \cdot k_{118} + q = O . \tag{14}$$

Solving Eq. (14) for $N(O^+)$ gives

$$N(O^+) = \frac{q + N(O_2^+) \cdot N(N) \cdot k_{118}}{N(O_2) \cdot k_{83} + N(N_2) \cdot k_{116}} . \tag{15}$$

From this result it is obvious that the variation in the O^+ concentration must follow the variation in the O_2^+ concentration. This is the reason for the linear decrease around 10 secs. When the rate of the charged rearrangement $O_2^+ + N \longrightarrow O^+ + NO$ becomes smaller than the production rate of the continuing source, the O^+ density becomes constant.

The time constant for the removal of N_2^+ by the charged rearrangement reaction $N_2^+ + O \rightarrow NO^+ + N$ and the charge transfer reaction $N_2^+ + O_2 \rightarrow O_2^+ + N_2$ is about 0.02 sec. The N_2^+ concentration decays after this time until the rate of removal equals the rate of formation by the continuing source and the N_2^+ concentration goes into equilibrium.

The O^- concentration increases with unit slope at the beginning of the solution as this ion is formed by the radiative attachment reaction $O + e \rightarrow O^- + h\nu$. The time constant for photodetachment from O^- is about 0.75 sec around which time the detachment rate exceeds the attachment rate, causing the O^- concentration to decay. The density of O^- does not go into quasi-equilibrium with the electrons until about 70 secs. After this time, the O^- density assumes a constant ratio to the electron density. The O_2^- concentration also increases with unit slope at early times as this ion is formed by the three-body attachment reaction $O_2 + e + M \rightarrow O_2^- + M$. Around 1 sec, the time constant for photodetachment from O_2^- and the charge transfer reaction $O_2^- + O \rightarrow O^- + O_2$, the removal rate exceeds the formation rate, causing O_2^- to decay. Around 100 secs the O_2^- concentration goes into quasi-equilibrium with the electrons and after this time it varies at a fixed ratio to the electron concentration.

The O_3^- concentration increases with a slope of 2 at the start because of the formation of this ion by the ion-neutral association reaction $O^- + O_2 \rightarrow O_3^- + h\nu$. The associative detachment reaction $O_3^- + O \rightarrow 2O_2 + e$ has a time constant of about 0.1 sec after which it exceeds the rate of attachment, causing the O_3^- to decay. The O_3^- goes into quasi-equilibrium around 5 secs and assumes a fixed ratio to the electron concentration. The NO_2^- ions are formed at early times by the three-body attachment reaction $NO_2 + e + M \rightarrow NO_2^- + M$. The NO_2^- concentration increases with unit slope, therefore, before 10^{-4} secs. As the O^- concentration increases, the NO_2^- production process changes to the charge transfer reaction $O^- + NO_2 \rightarrow NO_2^- + O$ and the slope of the NO_2^- profile changes to 2. Around 1 sec, the rate of photodetachment exceeds the rate of the charge transfer, causing the NO_2^- to decay. The associative detachment reaction $NO_2^- + O \rightarrow O_2 + NO + e$ also becomes effective in removing this ion around 50 secs, the time constant for this reaction. After 10 secs, the production falls off linearly as O^- decays while the photodetachment and associative detachment are removing this ion very rapidly, causing the sharp decrease between 10 and 100 secs. This removal continues until the rate of the charge transfer reaction again becomes equal to the rate of photodetachment and associative detachment and the NO_2^- concentration remains fixed at a constant ratio to the O^- concentration. The slight increase after 10^4 secs reflects the quasi-equilibrium dependence of this ion on the atomic oxygen concentration.

The atomic nitrogen starts to increase around 10^{-4} secs because of the production of this atom by NO^+ dissociative recombination. When the ionization starts to decay, this production stops, causing the atomic-nitrogen concentration to become constant again. The decay after 10^4 secs is caused by the increasing oxygen concentration in the recombination reaction $N + O \rightarrow NO + h\nu$. The behavior of the other minor neutral species is the same as given in the description of Figure 2.

5. THE DIURNAL VARIATION OF THE ATMOSPHERIC CONSTITUENTS

The basic code, as presented in Section 6.1, is ideally suited to such phenomena as the short-time effects on the atmosphere resulting from the detonation of nuclear weapons. The major source of the data on ionospheric parameters, however, is to be found in experimental studies of the natural ionosphere. These include ground-based probing techniques, measurements made in aircraft, and insitu measurements made with rockets. These experimental measurements cover most of the basic ionospheric characteristics of importance. In order to utilize these data for reaction-rate studies, for which this code is basically written, additions to the code are necessary so that it can solve the diurnal variation problem.

For this application it is required that the production of ionization by solar radiations be known as a function of the solar zenith angle. The code must also be capable of computing the local solar time for the location at which the computations are being made. The changes required in the deionization code for the computation of the diurnal variation of the atmospheric species are discussed in Section 7.2.

5.1 The Photoionization Source Function

In order to compute the diurnal variation of the atmospheric constituents, the production of ionization by direct and scattered solar radiation as well as that produced by cosmic rays is required.

Let $\Phi_\infty(\Delta\lambda)$ be the photon flux incident on the top of the atmosphere and $\Phi_z(\Delta\lambda)$ be the photon flux arriving at some height z in the wavelength region $\Delta\lambda$. Then

$$\Phi_z(\Delta\lambda) = \Phi_\infty(\Delta\lambda)e^{-\tau(\Delta\lambda)} \tag{16}$$

where $\tau(\Delta\lambda)$ is the optical depth for energies in the wavelength region $\Delta\lambda$. If only atomic and molecular oxygen and molecular nitrogen are considered to be absorbers of this radiation, then the optical depth can be written as

$$\tau(\Delta\lambda) = \tau_O(\Delta\lambda) + \tau_{O_2}(\Delta\lambda) + \tau_{N_2}(\Delta\lambda) \quad . \tag{17}$$

The optical depth for each of the constituents is defined as

$$\tau_O(\Delta\lambda) = \sigma_O(\Delta\lambda) \cdot \int_z^\infty N(O) \cdot dl \quad , \tag{18}$$

$$\tau_{O_2}(\Delta\lambda) = \sigma_{O_2}(\Delta\lambda) \cdot \int_z^\infty N(O_2) \cdot dl \quad , \tag{19}$$

and

$$\tau_{N_2}(\Delta\lambda) = \sigma_{N_2}(\Delta\lambda) \cdot \int_z^\infty N(N_2) \cdot dl \tag{20}$$

where $\sigma_O(\Delta\lambda)$, $\sigma_{O_2}(\Delta\lambda)$, and $\sigma_{N_2}(\Delta\lambda)$ are the absorption cross sections for atomic oxygen, molecular oxygen, and molecular nitrogen respectively and the integrals represent the total number of the atoms or molecules in a cm^2 column extending from z to infinity along the path l.

For computing the photoionization rates only those wavelengths capable of producing ionization have to be considered. The first ionization potential of O_2 is at 1026.5A, while the first ionization potential of O and N_2 are at shorter wavelengths. In considering photoionization, therefore, it is necessary to take into account only energy fluxes at wavelength below 1026.5A.

The primary electron production rate at altitude z resulting from the ionization of O, O_2, and N_2 in the wavelength region 1-1027A can be written as

$$q = q(O^+) + q(O_2^+) + q(N_2^+) \tag{21}$$

$$q_{UV}(e) = \sum_{i=1}^{3} q_i = \sum_{i=1}^{3} \sum_{\Delta\lambda} N_i \eta \sigma_i^!(\Delta\lambda)\, \Phi_\infty(\Delta\lambda) e^{-\left(\sum_{i=1}^{3} \sigma_i(\Delta\lambda) \int_z^\infty N_i\, dl\right)} \tag{22}$$

where the $\sigma_i^!(\Delta\lambda)$ are the ionization cross sections for the various constituents.

In addition to this, there are the electrons produced by the ionization of nitric oxide by L_α at 1215.7A. The electron production rate resulting from this mechanism can be written as

$$q_{L_\alpha} = N(NO) \cdot \eta\sigma'(NO) \cdot \phi_\infty, L_\alpha) \cdot e^{-\left(\sigma(O_2) \cdot \int_z^\infty N(O_2) \cdot dl\right)} \tag{23}$$

since L_α is absorbed by O_2. For L_α the ionization cross section, $\sigma'(NO)$, according to Watanabe (1954), is $2 \times 10^{-18} cm^2$ and the absorption cross section for molecular oxygen, $\sigma(O_2)$, according to Lee (1955) and Ditchburn, Bradley, Cannon, and Munday (1954), is $8.5 \times 10^{-21} cm^2$. A daytime flux of 3 ergs/cm^2/sec for L_α is assumed for the numerical computations in this paper.

The production rate of electrons by energies in the 1-10A region is given by

$$q_X(e) = \sum_{i=1}^{3} \sum_{\lambda} N_i K(\lambda) e^{-\left(\sum_{i=1}^{3} \sigma_i(\lambda) \int_z^\infty N_i \, dl\right)} \tag{24}$$

where $K(\lambda)$ is the ionization coefficent per molecule per sec at 2, 4, and 6A.

The production rate of electrons by cosmic rays used in this work is simply

$$q_{CR}(e) = 1.0 \times 10^{-16} \cdot N_z \tag{25}$$

where N_z is the total particle concentration at altitude .

The total daytime production rate of electrons is

$$q(e) = q_{UV}(e) + q_{L_\alpha}(e) + q_{CR}(e) + q_X(e) \quad . \tag{26}$$

During the night, in addition to the cosmic-ray flux, there is also some L_α and L_β flux scattered in from the sunlit atmosphere. For the purpose of this work it was assumed that one percent of the noontime L_α flux and 0.1 percent of the noontime L_β flux are available for the production of ionization during the night.

The photon fluxes in the 10-1027A region are those reported by Hinteregger (1960). The 1-10A X-ray ionization coefficients are taken from Nicolet and Aikin (1960) assuming slightly disturbed solar conditions. Table 1 shows the photon flux in each wavelength region and ionization and absorption cross sections for O , O_2 , and N_2 , as given by Hinteregger (1960). For broad spectral ranges, mean values of the ionization cross section are used. Because of the variation of the cross sections within a spectral range, especially for N_2 , several cross sections were used within the range with equal distribution of the total flux within the range over each of the subdivisions. The ionization efficiency factor is η. Its value is one for all wavelengths whose energies are less than 35 ev. For wavelengths with energies greater than 35 ev, η is given by $\eta = \frac{E}{35}$ where E is the energy of the photon in ev.

Table 2 lists the concentrations of O , O_2 , and N_2 used to compute the column integrals. The concentrations above 80 km are taken from the COSPAR International Reference Atmosphere (1965). Concentrations below 80 km are taken from Keneshea (1963). A description of the photoionization code is given in Appendix A.

The electron production rate at noontime by the external production function is shown in Figure 5. Above 90 km, the ionization is by ultraviolet and soft X-rays. Around 90 km, the ionization is produced almost exclusively by X-rays in the 35A region while around 80 km the production is primarily the result of the ionization of NO by L_α . By 60 km, the production results solely from cosmic-ray ionization.

Figure 6 shows the electron-ion production rates at noontime by the external source function. Above 90 km, the O_2^+ ion is the most abundant ion produced. At 70 and 80 km, the production of NO^+ ions by L_α predominates.

5.2 The Simplified Sunrise-Sunset Function

Because of the rapid changes in the concentrations of some of the atmospheric constituents during sunrise and sunset, more realistic results are obtained with the code if the sun is considered to be an extended source rather than a point source. It is necessary, therefore, to consider the geometry of a moving disc obscured by a plane. Because the viewing angle of the earth is much larger than the viewing angle of the sun to an observer in the ionosphere, the earth's horizon is considered to be flat.

In order to handle this problem correctly, it is necessary to take into account the absorption of the specific wavelengths responsible for the photodetachment of each negative ion and the photodissociation of each neutral species. Unfortunately, the spectral dependence of these processes for all the negative ions and neutral constituents is not known. For the purposes of this paper, therefore, the full daylight coefficients for photodetachment and photo dissociation are simply modified by the percentage of the total visible light transmitted as the sun rises or sets.

Table 1. Solar Flux Data, Ionization Cross Sections, and Absorption Cross Sections
(Data of H. E. Hinteregger GRD Technical Note 39, AFCRL-TN-60-485, 1960)

Wavelength A	Photon Flux, $10^8/cm^2/sec$	$\eta\,\sigma'(O_2)$ $10^{-18}cm^2$	$\sigma\ (O_2)$ $10^{-18}cm^2$	$\eta\,\sigma'(O)$ $10^{-18}cm^2$	$\sigma\ (O)$ $10^{-18}cm^2$	$\eta\,\sigma'(N_2)$ $10^{-18}cm^2$	$\sigma\ (N_2)$ $10^{-18}cm^2$
1025.7	26	1	1.7	0	0	0	0
1000-1027	15	1	1.5	0	0	0	0
989.8	5	1.8	2.2	0	0	0	0
977.0	30	3	3.7	0	0	0	0.8
972.5	10	25	30	0	0	0	280
949.7	5	5	5.6	0	0	0	1.9
911-1000	7.4	6	7.4	0	0	0	40
	7.4	6	7.4	0	0	0	10
	7.4	6	7.4	0	0	0	5
	7.4	6	7.4	0	0	0	1
	7.4	6	7.4	0	0	0	0.4
850-911	19	9	11	3	3	0	40
	19	9	11	3	3	0	10
	19	9	11	3	3	0	5
	19	9	11	3	3	0	1
	19	9	11	3	3	0	0.4
796-850	5	9	11	3.5	3.5	0	40
	5	9	11	3.5	3.5	0	10
	5	9	11	3.5	3.5	0	5
	5	9	11	3.5	3.5	0	1
	5	9	11	3.5	3.5	0	0.4
700-796	16.6	15	18	5	5	9	11
	16.6	15	18	5	5	15	19
	16.6	15	18	5	5	30	37
600-700	15.6	15	18	10	10	9	11
	15.6	15	18	10	10	15	19
	15.6	15	18	10	10	30	37
584.3	29	15	18	13	13	15	19

Table 1 (Continued)

Wavelength A	Photon Flux, $10^8/cm^2/sec$	$\eta\,\sigma'(O_2)$ $10^{-18}cm^2$	$\sigma\ (O_2)$ $10^{-18}cm^2$	$\eta\,\sigma'(O)$ $10^{-18}cm^2$	$\sigma\ (O)$ $10^{-18}cm^2$	$\eta\,\sigma'(N_2)$ $10^{-18}cm^2$	$\sigma\ (N_2)$ $10^{-18}cm^2$
500-600	10	15	13	15	15	9	11
	10	15	18	15	15	15	19
	10	15	18	15	15	30	37
400-500	8	15	18	11	11	9	11
	8	15	18	11	11	15	19
	8	15	18	11	11	30	37
303.8	43	19.9	17	11.7	10	11.7	10
300-400	15	16.2	16	10.1	10	11.1	11
	15	16.2	16	10.1	10	19.2	19
230-300	31	20.1	15	12.1	9	12.1	9
170-230	33	19.5	11	12.4	7	10.6	6
110-170	3.5	5.33	2.1	2.54	1	12.7	5
80-110	4.5	6.62	1.77	3.33	0.89	3.74	1
60-80	2.8	4.92	0.97	2.48	0.49	2.86	0.565
45-60	1.8	2.7	0.4	1.35	0.2	1.55	0.23
35-45	1.1	0.887	0.19	0.842	0.095	0.976	0.11
30-35	0.5	0.983	0.09	0.491	0.045	0.568	0.052
20-30	0.12	6.39	0.45	3.27	0.23	13.3	0.94
10-20	0.02	11	0.467	5.51	0.233	7.97	0.337

Table 2. Concentrations of Neutral Species

Altitude (km)	N(0)/CC	$N(0_2)$/CC	$N(N_2)$/CC
0	0	5.34×10^{18}	1.99×10^{19}
10	1.00×10^{5}	1.80×10^{13}	6.71×10^{18}
20	1.50×10^{8}	3.00×10^{17}	1.40×10^{18}
30	1.20×10^{10}	7.00×10^{16}	3.15×10^{17}
40	2.40×10^{10}	1.70×10^{16}	7.60×10^{16}
50	3.50×10^{10}	5.00×10^{15}	1.80×10^{16}
60	4.80×10^{10}	1.50×10^{15}	5.70×10^{15}
70	6.40×10^{10}	7.53×10^{14}	1.60×10^{15}
80	8.50×10^{10}	7.95×10^{13}	2.96×10^{14}
90	1.25×10^{11}	1.33×10^{13}	4.97×10^{13}
100	5.00×10^{11}	1.99×10^{12}	8.18×10^{12}
110	2.00×10^{11}	3.49×10^{11}	1.62×10^{12}
120	7.60×10^{10}	7.46×10^{10}	4.01×10^{11}
130	3.67×10^{10}	2.34×10^{10}	1.40×10^{11}
140	2.16×10^{10}	9.79×10^{9}	6.36×10^{10}
150	1.42×10^{10}	4.78×10^{9}	3.34×10^{10}
160	9.84×10^{9}	2.56×10^{9}	1.91×10^{10}
170	7.15×10^{9}	1.47×10^{9}	1.16×10^{10}
180	5.36×10^{9}	8.83×10^{8}	7.37×10^{9}
190	4.12×10^{9}	5.52×10^{8}	4.85×10^{9}
200	3.23×10^{9}	3.57×10^{8}	3.29×10^{9}
210	2.58×10^{9}	2.37×10^{8}	2.29×10^{9}
220	2.09×10^{9}	1.61×10^{8}	1.62×10^{9}
230	1.71×10^{9}	1.11×10^{8}	1.17×10^{9}
240	1.41×10^{9}	7.75×10^{7}	8.51×10^{8}
250	1.18×10^{9}	5.49×10^{7}	6.28×10^{8}
260	9.80×10^{8}	3.92×10^{7}	4.67×10^{8}
270	8.33×10^{8}	2.82×10^{7}	3.50×10^{8}
280	7.06×10^{8}	2.05×10^{7}	2.64×10^{8}
290	6.00×10^{8}	1.50×10^{7}	2.00×10^{8}
300	5.12×10^{8}	1.10×10^{7}	1.52×10^{8}
310	4.38×10^{8}	8.23×10^{6}	1.17×10^{8}
320	3.70×10^{8}	5.80×10^{6}	8.67×10^{7}
330	3.22×10^{8}	4.50×10^{6}	6.90×10^{7}
340	2.74×10^{8}	3.20×10^{6}	5.14×10^{7}

Table 2 (Continued)

Altitude (km)	N(0)/CC	$N(0_2/CC$	$N(N_2/CC$
350	$2.39X10^8$	$2.49X10^6$	$4.11X10^7$
360	$2.04X10^8$	$1.78X10^6$	$3.08X10^7$
370	$1.78X10^8$	$1.39X10^6$	$2.46X10^7$
380	$1.52X10^8$	$1.00X10^6$	$1.85X10^7$
390	$1.33X10^8$	$7.82X10^5$	$1.49X10^7$
400	$1.14X10^8$	$5.65X10^5$	$1.13X10^7$
410	$1.00X10^8$	$4.43X10^5$	$9.04X10^6$
420	$8.59X10^7$	$3.20X10^5$	$6.83X10^6$
430	$7.53X10^7$	$2.52X10^5$	$5.51X10^6$
440	$6.49X10^7$	$1.83X10^5$	$4.19X10^6$
450	$5.70X10^7$	$1.44X10^5$	$3.38X10^6$
460	$4.91X10^7$	$1.05X10^5$	$2.58X10^6$
470	$4.32X10^7$	$8.30X10^4$	$2.09X10^6$
480	$3.73X10^7$	$6.06X10^4$	$1.59X10^6$
490	$3.28X10^7$	$4.74X10^4$	$1.29X10^6$
500	$2.83X10^7$	$3.51X10^4$	$9.86X10^5$
510	$2.50X10^7$	$2.72X10^4$	$8.00X10^5$

The area of a sector of the circle in Figure 7 subtended by the angle θ is $\frac{r^2\theta}{2}$. From Figure 7

$$\cos\left(\frac{\theta}{2}\right) = \frac{r-x}{r} \quad . \tag{27}$$

Since

$$\cos 2u = 2\cos^2 u - 1 \quad , \tag{28}$$

then

$$\cos\theta = 2\left(\frac{r-x}{r}\right)^2 - 1 \tag{29}$$

or

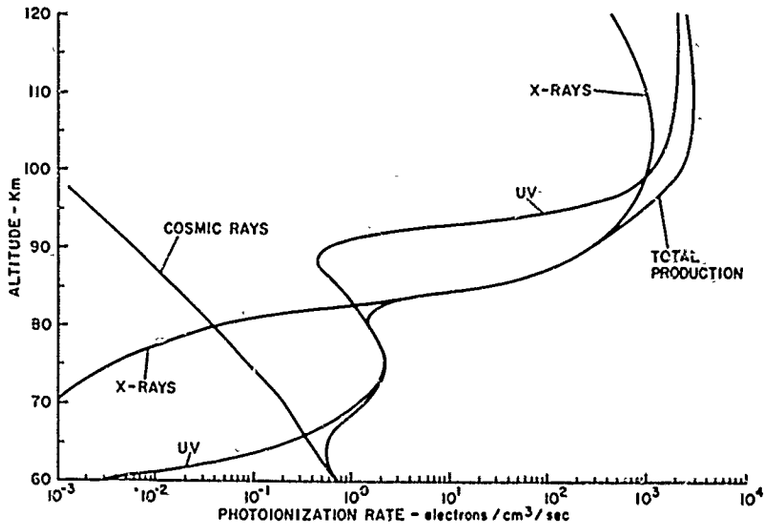

Figure 5. Electron Production Rates by Extraterrestrial Radiations (Brazil noontime)

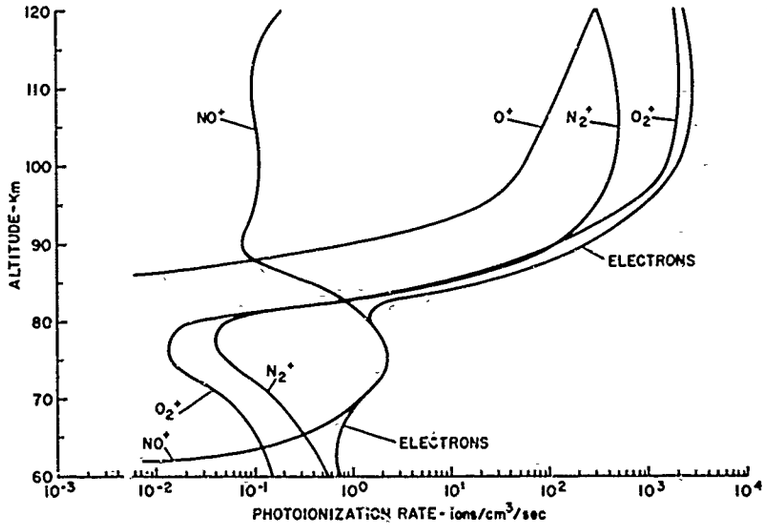

Figure 6. Electron-Ion Production Rates by Extraterrestrial Radiation (Brazil noontime)

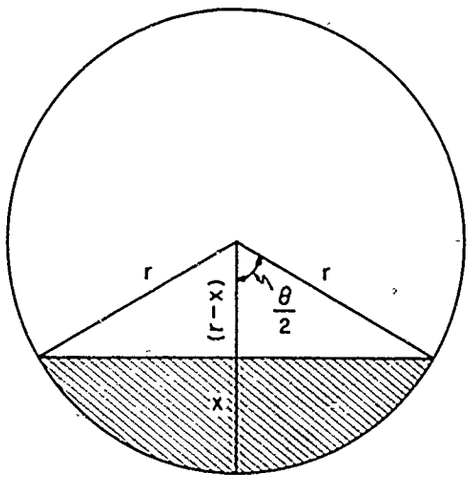

Figure 7. Sunrise-Sunset Geometry

$$\theta = \cos^{-1}\left[2\left(\frac{r-x}{r}\right)^2 - 1\right] \quad . \tag{30}$$

The area of the two triangles in Figure 7 is

$$(r-x)\, r \sin\left(\frac{\theta}{2}\right) \tag{31}$$

which can be written as

$$(r-x)\, r\left[\frac{1-\cos\theta}{2}\right]^{1/2} \quad . \tag{32}$$

The shaded area of the sector is

$$A = \frac{r^2\theta}{2} - (r-x)\, r\left[\frac{1-\cos\theta}{2}\right]^{1/2} \quad . \tag{33}$$

Let r equal one solar radius, then

$$A = \frac{\theta}{2} - (1-x)\left[\frac{1-\cos\theta}{2}\right]^{1/2} \tag{34}$$

where

$$\cos\theta = 2(1-x)^2 - 1 \quad . \tag{35}$$

The transmissivity factor, η, can be written as

$$\eta = 1 - \frac{\Lambda}{\pi r^2} \tag{36}$$

which for $r=1$ becomes

$$\eta = 1 - \frac{\Lambda}{\pi} \tag{37}$$

or

$$\eta = 1 - \left\{ \frac{\left[\frac{\theta}{2} - (1-x) \left| \frac{1-\cos\theta}{2} \right|^{1/2} \right]}{\pi} \right\} \tag{38}$$

5.3 Diurnal Variation Results

In computing the diurnal variation some changes in the basic list of reaction rates are necessary in order to obtain solutions that are generally consistent with available experimental evidence. An attempt was made to achieve the NO^+/O_2^+ daytime ratios observed by Narcisi (1966). There is, of course, no unique way to obtain these ratios in the present system. In the E region they can be achieved by varying the rate constant for the charged rearrangement reaction $O_2^+ + N \longrightarrow NO^+ + O$. They can also be arrived at by using a different concentration of atomic nitrogen for this reaction. This ratio is also sensitive to the ratio of the rate constants for NO^+ and O_2^+ dissociative recombination. Because of the recent measurements by Furguson et al. (1965) of the rate constant for the charged rearrangement reaction and because of the lack of experimental measurements of the concentration of atomic nitrogen, the first two choices were rejected. In order to establish ratios consistent with those of Narcisi (1966), the rate constant for NO^+ dissociative recombination is made equal to that for O_2^+ dissociative recombination.

In addition to the other rate constant adjustments discussed in Section 4, the rate constants for four more reactions are changed for the diurnal variation runs. The associative detachment reaction $O_3^- + O \longrightarrow 2O_2 + e$ is an important reaction in the electron kinetics in the D region. Allowing this detachment to take place produces profiles at night that are contrary to experimental observations. An example of this behavior is presented in Section 6. The rate constant for this reaction and the similar reaction for $NO_2^- + O \longrightarrow O_2 + NO + e$ were therefore set to zero. Problems then arose in trying to obtain a negative ion-to-electron ratio of unity at 70 km. To accomplish this the rate coefficient for O_3^- photodetachment is changed to 1.4 and the rate constant for the charge transfer reaction $O_2^- + O_3 \longrightarrow O_3^- + O_2$ is changed to 7.77×10^{-9}. With these rate constant changes, reasonable negative ion-to-electron ratios are obtained at noontime in the D region.

Because the energy available in the Herzberg continum was not considered in this work and because all of the energy in the Schumann continum is absorbed above 100 km, no photodissociation of O_2 is allowed below 110 km. The incorporation O_2 photodissociation into the code is discussed in Section 7.2.1.1. When viewing the atomic oxygen profiles, this fact should be kept in mind.

Before discussing the actual diurnal variations of the various atmospheric species in the D and E ionospheric regions, it might be worthwhile to study the kinetics of the reactions responsible for the noontime profiles of the more important ionic and neutral constituents.

7.3.1 THE NOONTIME KINETICS

The diurnal variations were computed in height intervals of 10 km and at noontime a snapshot was taken of the rates at which the individual reactions were proceeding. From this output the reactions contributing predominantly to the kinetics of each species were extracted. The height profiles of the rates at which these reactions were proceeding are shown in Figures 8 through 16 for the major species.

The electron kinetics are shown in Figure 8. Above 90 km, the electron concentration results from the equilibrium established between the electrons produced by the external source and those lost by dissociative recombination with O_2^+ and NO^+. Below 80 km, the picture is entirely different. Here the electron concentration is determined by the equilibrium established between the production of electrons by photodetachment from negative ions and the three-body attachment of the electrons to O_2. The altitudes between 80 and 90 km are transition altitudes where the electron concentration results equally from the external source and the photodetachment productions and the dissociative recombination and three-body attachment removals.

Figure 9 shows the kinetics of O^-. Above 90 km, this ion is produced solely by the radiative attachment of electrons to atomic oxygen. Above 110 km the only removal process for these ions is the photodetachment, while below 100 km the associative detachment $O^- + N_2 \longrightarrow N_2O + e$ becomes the only removal mechanism. There is some controversy as to whether this reaction is endothermic or exothermic. Bortner (1965) discussed this problem and accepted the thermochemical data that makes this process exothermic. Below 90-km the O^- ions are produced by the charge transfer between O_2^- and atomic oxygen and by the dissociative attachment reaction $O_3 + e \longrightarrow O^- + O_2$.

From Figure 10 it is obvious that the formation of O_2^- ions over the entire altitude range is by the three-body attachment of electrons to O_2. Above 80 km they are removed by photodetachment and by the charge transfer between O_2^- and atomic oxygen. Below 80 km the predominant removal process becomes the charge transfer between O_2^- and ozone.

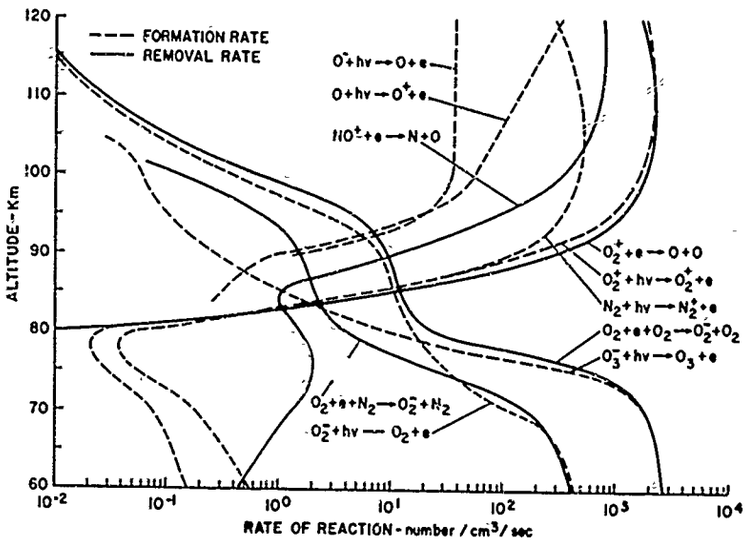

Figure 8. Important Processes for Electrons (Brazil noontime)

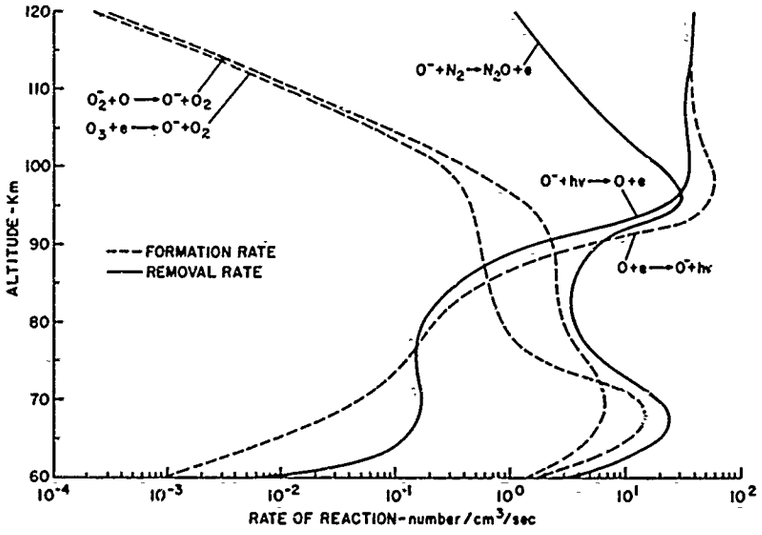

Figure 9. Important Processes for O^- (Brazil noontime)

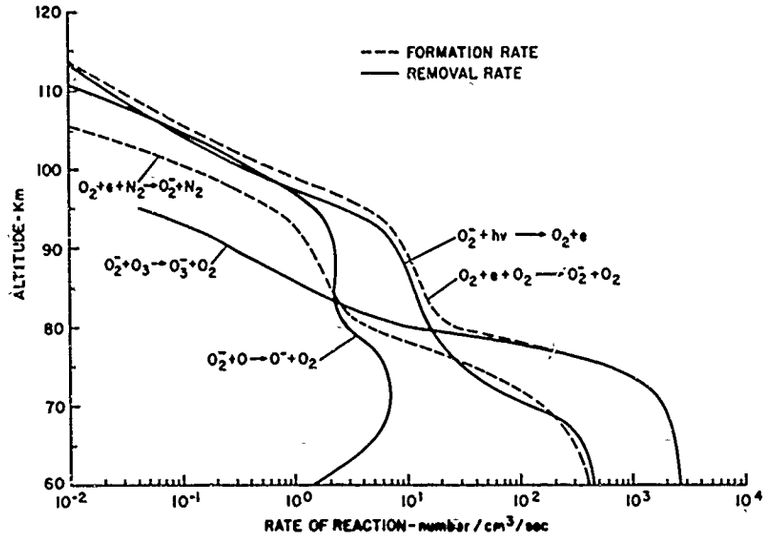

Figure 10. Important Processes for O_2^- (Brazil noontime)

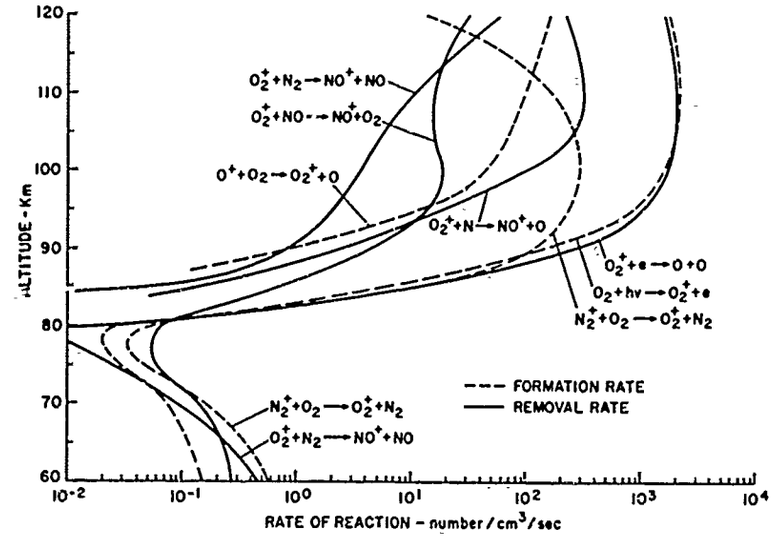

Figure 11. Important Processes for O_2^+ (Brazil noontime)

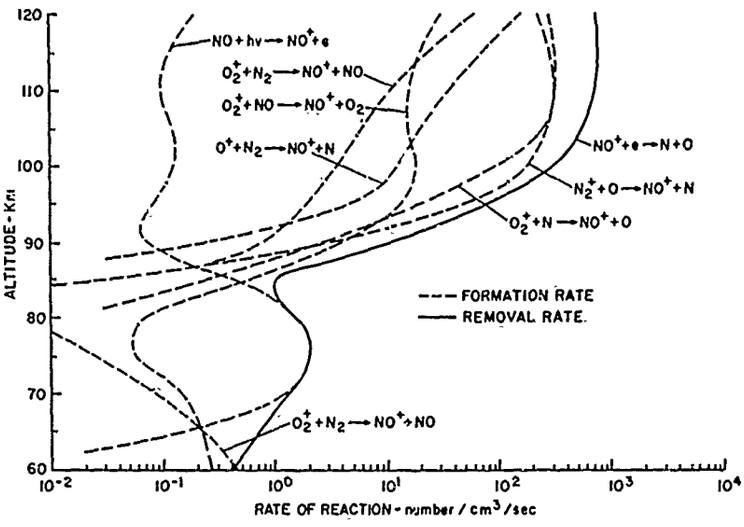

Figure 12. Important Processes for NO^+ (Brazil noontime)

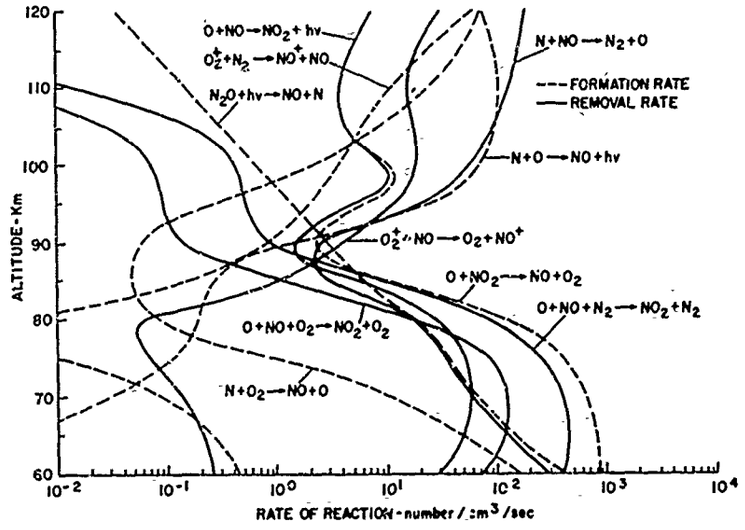

Figure 13. Important Processes for Nitric Oxide (Brazil noontime)

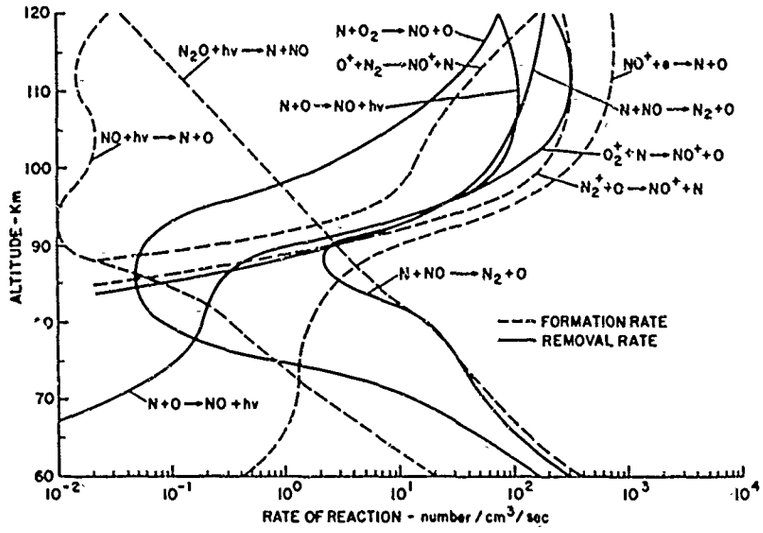

Figure 14. Important Processes for Atomic Nitrogen (Brazil noontime)

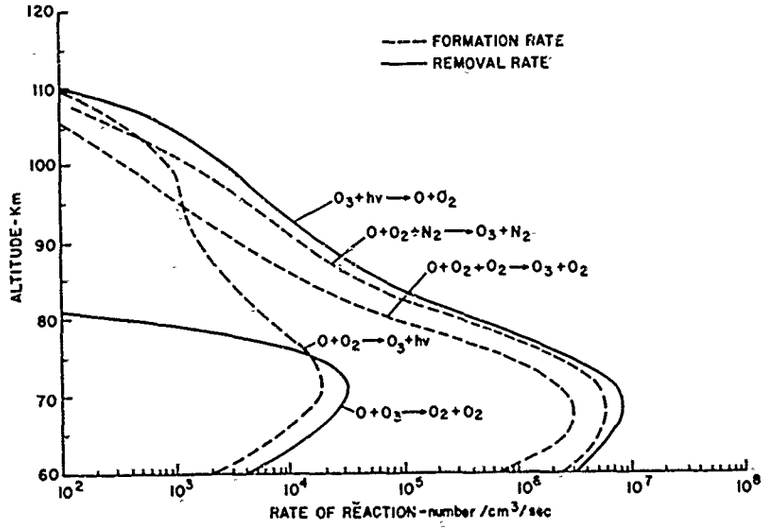

Figure 15. Important Processes for Ozone (Brazil noontime)

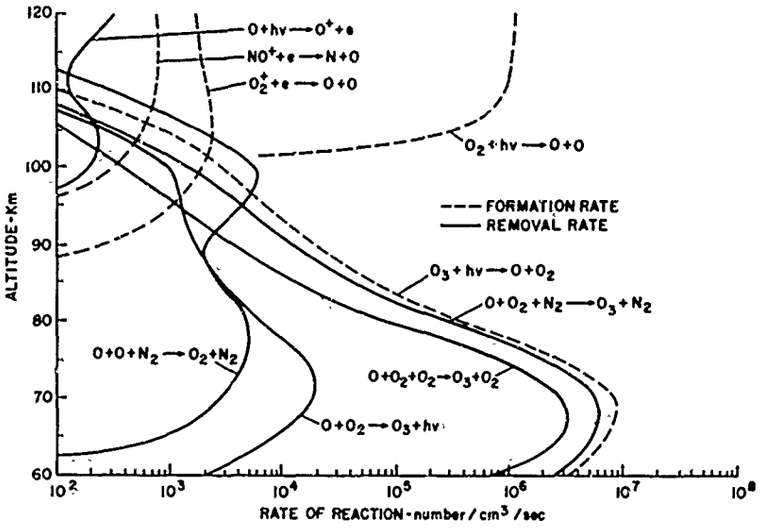

Figure 16. Important Processes for Atomic Oxygen (Brazil noontime)

The kinetics of the O_2^+ ions are shown in Figure 11. Above 80 km these ions are removed almost entirely by dissociative recombination. In the E region they are formed primarily by photoionization. Below 90 km the charge transfer process $N_2^+ + O_2 \rightarrow O_2^+ + N_2$ and the external source contribute about equally to the production of this ion. Below 80 km O_2^+ becomes a minor ion since it is produced by cosmic-ray ionization and from N_2^+ by charge transfer while the latter ions are being formed in small numbers by cosmic-ray ionization. The O_2^+ ions react to form NO^+ ions below 80 km by charged rearrangement with N_2 and charge transfer with NO .

Figure 12 shows the processes responsible for the noontime kinetics of NO^+ . The only removal process that is important at all altitudes is the dissociative recombination. Above 85 km NO^+ is formed primarily from O_2^+ and N_2^+ by charged rearrangement with atomic oxygen and atomic nitrogen respectively. Of lesser importance are the O_2^+ charged rearrangement with N_2 , the charge transfer between O_2^+ and NO and the charged rearrangement between O^+ and N_2 . Between 70 and 80 km this ion is formed almost exclusively by the ionization of NO by L_α .

The kinetics of nitric oxide are shown in Figure 13. The neutral rearrangement reaction $N + NO \rightarrow N_2 + O$ is an important removal process for this molecule at all altitudes. The two-body atom recombination $N + O \rightarrow NO + h\nu$ is important in forming this molecule above 90 km. Below this altitude it becomes unimportant

because of the relatively small amounts of atomic nitrogen available. The abundance of O_2^+ ions in the E region makes the charge transfer to NO also important above 90 km. Below 80 km these molecules are formed by the neutral rearragnement process $O + NO_2 \rightarrow NO + O_2$ and by the photcdissociation of N_2O. They are removed by the three-body atom recombination $O + NO + M \rightarrow NO_2 + M$ and by the atom recombination $O + NO \rightarrow NO_2 + h\nu$. The minimum in the rate of the process $N + O_2 \rightarrow NO + O$ between 80 and 90 km is produced by the temperature dependence of the rate constant for this process which is a minimum in this region.

The kinetics of atomic nitrogen, as shown in Figure 14, are influenced mainly by charged species reactions above 90 km and by neutral reactions below this altitude. In the E layer it is produced primarily by the dissociative recombination of NO^+ and by the charged rearrangement between N_2^+ and atomic oxygen. Its major removal process is the charged rearrangement with O_2^+. Also contributing to its removal are its neutral rearrangement with NO and its recombination with atomic oxygen. In the D region this atom is formed by the photodissociation of N_2O and of NO. It is removed by the netural rearrangement reactions $N + NO \rightarrow N_2 + O$ and $N + O_2 \rightarrow NO + O$.

Figure 15 shows the kinetics for ozone at noontime. The primary removal mechanism for this molecule is photodissociation. Also contributing to its removal is the neutral rearrangement reaction $O + O_3 \rightarrow 2O_2$. The major formation of ozone is by the three-body recombination $O + O_2 + M \rightarrow O_3 + M$. Of minor importance in forming ozone is the two-body atom recombination $O + O_2 \rightarrow O_3 + h\nu$. The rates for the three-body recombination processes fall off rapidly with increasing altitude because of the decreasing molecular concentrations.

The important reactions for atomic oxygen are shown in Figure 16. In the D region the atomic oxygen kinetics are controlled by the same reactions as the ozone concentration. In the E region the photodissociation of molecular oxygen is the major source of oxygen atoms. There is no effective chemical removal of these atoms during the daytime in the E region.

5.3.2 THE DIURNAL VARIATIONS

The code, as discussed in Section 6.2, was used to compute the time histories of the 15 atmospheric species incorporated into the code. Many concentration profiles were computed for various geographic locations. From these computations those made for the coordinates 32.197° S and 52.169° W were chosen for presentation here. This location is in Brazil where sounding rockets were launched during the eclipse of 12 November 1966.

The diurnal variation profiles are generated in the following manner. The code is started at local noon with the concentrations of the charged species initially set to zero and the concentrations of the neutral species initially set to best estimate values.

The code is then run through one diurnal cycle (24 hours). Taking the concentrations of all the species at the end of this diurnal cycle as initial conditions, a solution is generated over a second diurnal cycle. The profiles computed over this second diurnal cycle are presented here.

Technically, this procedure should be repeated over as many diurnal cycles as are required to achieve equilibrium profiles. Equilibrium, as defined here, is achieved when the variation in the concentration of each species reproduces itself over succeeding diurnal cycles. In the profiles presented, it is apparent that for some of the species this equilibrium was not achieved in the second cycle and more diurnal cycles should have been computed. Nevertheless, this does not diminish the validity of the results as they are discussed.

Figure 17 shows the diurnal variation of the major charged species at 120 km. Although the variation of all nine charged species in the code were computed, only those species that attain concentrations above 10 particles/cm^3 are plotted. The noontime value of the electron concentration results from the equilibrium established between the production by photoionization and the dissociative recombination of O_2^+ and NO^+. As the solar zenith angle increases, the rate of production decreases because of the greater attenuation of the solar radiations through the denser columns in the atmosphere. The production rate is reduced to its nighttime value about a half hour before the sun starts to set. As fewer electron are produced, those already made are removed by recombination, causing the concentrations of the electrons and the positive ions to decrease. This decay continues until the production rate is reduced to its constant nighttime value at which time a new equilibrium is established for the electrons and the NO^+ ions. The O_2^+ concentration decays below the NO^+ concentration because of the conversion of O_2^+ ions into NO^+ ions by the charged rearrangement reaction $O_2^+ + N \longrightarrow NO^+ + O$. The O_2^+ concentration increases during the night because the rate of removal by the charged rearrangement reaction $O_2^+ + N \longrightarrow NO^+ + O$ decreases with the decreasing atomic nitrogen concentration. After the sun rises and the ionizing radiation starts to penetrate to this altitude, the production of ionization increases, causing the concentration to increase until the production rate levels off around noontime and the daytime equilibrium is established between the production of ionization and the dissociative recombination.

Figure 18 shows the diurnal profiles for the neutral species at 120 km. Atomic oxygen was not plotted here because its concentration varies only slightly around 2×10^{11} atoms/cm^3. During the daytime the concentration of atomic nitrogen builds up as it is formed by the dissociative recombination $NO^+ + e \longrightarrow N + O$ and by the charged rearrangement $N_2^+ + O \longrightarrow NO^+ + N$ at a faster rate than it is removed by the charged rearrangement $O_2^+ + N \longrightarrow NO^+ + O$ or the neutral rearrangement $N + NO \longrightarrow N_2 + O$. During the night, however, the formation by the

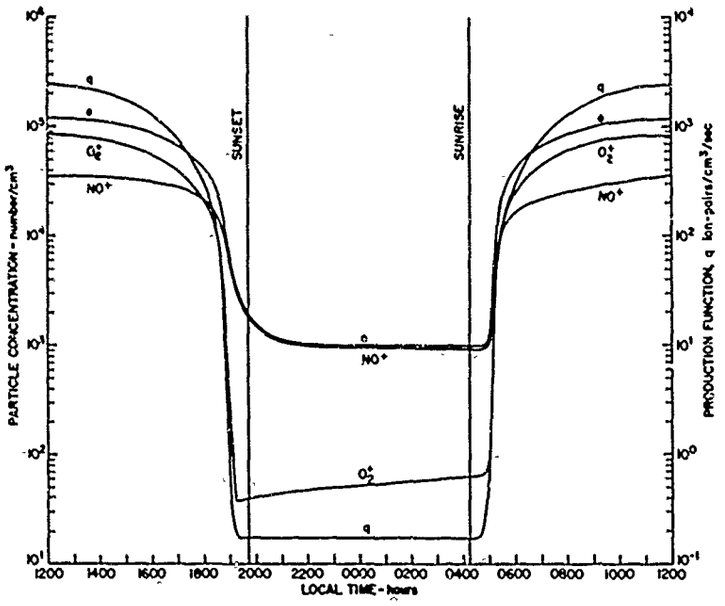

Figure 17. Diurnal Variation of Charged Species at 120 km (Brazil noontime)

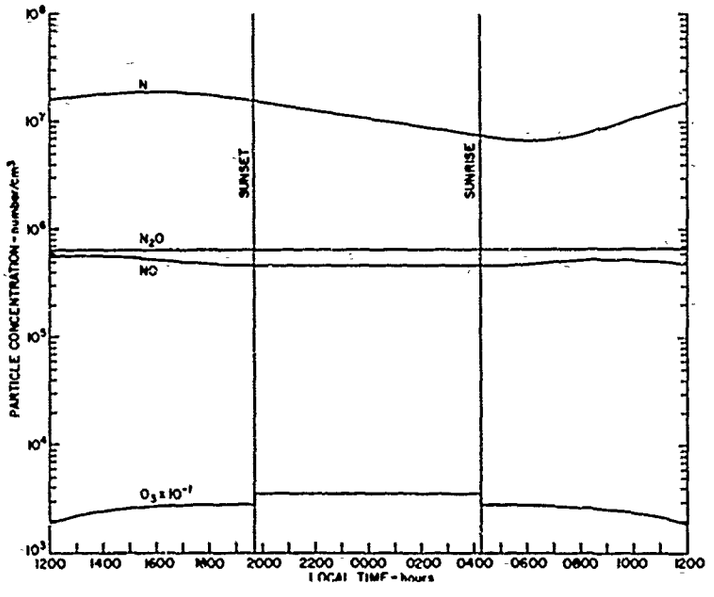

Figure 18. Diurnal Variation of Neutral Species at 120 km (Brazil noontime)

dissociative recombination and the charged rearrangement practically stops. The only process really affecting this atom then is the neutral rearrangement which causes its concentration to decay steadily throughout the night. No change is possible in the N_2O concentration because the time constant for its removal by photodissociation is about 10^5 secs which is longer than one day. There is no effective formation or removal process for this molecule at night. The slight diurnal variation in nitric oxide is caused by the diurnal variation of O_2^+ in the process $O_2^+ + N_2 \longrightarrow NO^+ + NO$. Ozone is only a trace species at 120 km. The slight variation in its concentration is caused by the variation of its rate of photodissociation with the solar zenith angle. When the photodissociation stops at sunset its concentration increases by the two-body recombination reaction $O + O_2 \longrightarrow O_3 + h\nu$ until a new equilibrium is established between this recombination and the netural rearrangement reaction $O + O_3 \longrightarrow O_2 + O_2$.

Figure 19 shows the diurnal variation of the major charged species at 110 km. The behavior of the charged species at this altitude is similar to that at 120 km. The noontime equilibrium value of the electrons is slightly lower here than at 120 km because of the slightly larger recombination rate constants. Since these rate constants are inversely proportional to the temperature, their values are higher at 110 km than at 120 km because of the decreased temperature. The nighttime concentrations are a little higher than they are at 120 km because the nighttime production function is larger at this altitude.

The only real difference in the neutral species profiles at 110 km, as shown in Figure 20, from those at 120 km is the ozone profile. Since the rate of the three-body recombination process $O + O_2 + M \longrightarrow O_3 + M$ is increasing with decreasing altitude, more ozone molecules are formed at this altitude at night when their photodissociation has stopped.

The charged species diurnal variations at 100 km, as shown in Figure 21, display much of the same behavior that is seen at the two higher altitudes. The slightly lower concentrations reflect the slightly lower photoionization rate. At night, however, at this altitude O_3^- starts to become an important negative ion. Because of the increase in the concentration of ozone during the night, the rate of the dissociative attachment reaction $O_3 + e \longrightarrow O^- + O_2$ increases. However, the charge transfer reaction $O^- + O_3 \longrightarrow O_3^- + O$ proceeds so rapidly because of the increased O_3 concentration that the O^- ions created by attachment immediately transfer their charge to O_3, causing the electrons to decay slightly as their charge goes into the formation of O_3^- ions. As soon as the sun starts to rise, however, the electrons are very quickly detached from the O_3^- ions, causing the concentration of this ion to decay rapidly with a corresponding increase in the electron concentration.

Figure 22 shows the diurnal variation of the minor neutral species at 100 km. Their behavior is again similar to that at the higher altitudes. There is, however,

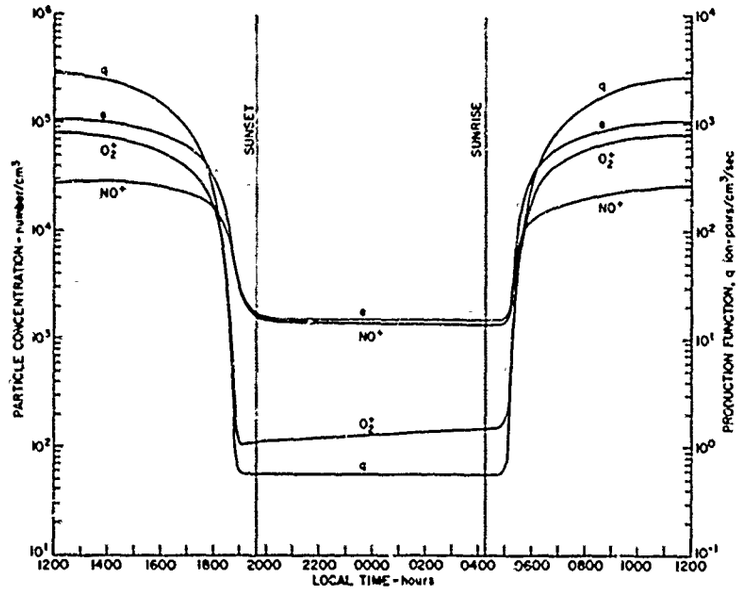

Figure 19. Diurnal Variation of Charged Species at 110 km (Brazil noontime)

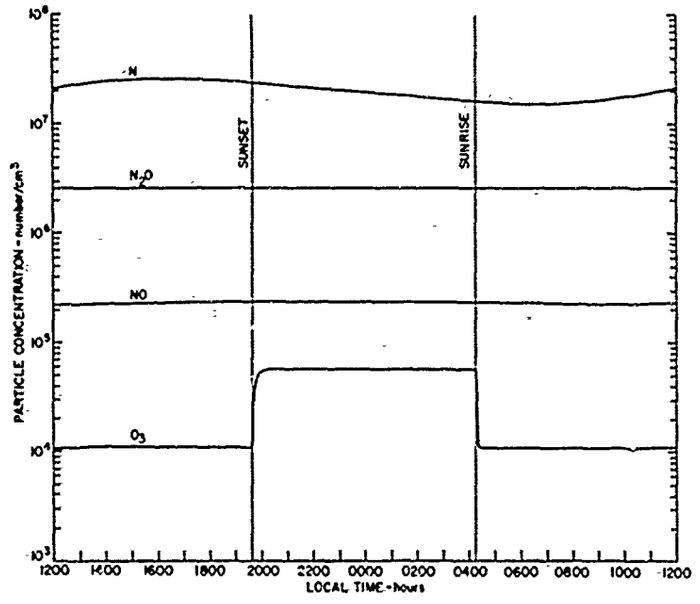

Figure 20. Diurnal Variation of Neutral Species at 110 km (Brazil noontime)

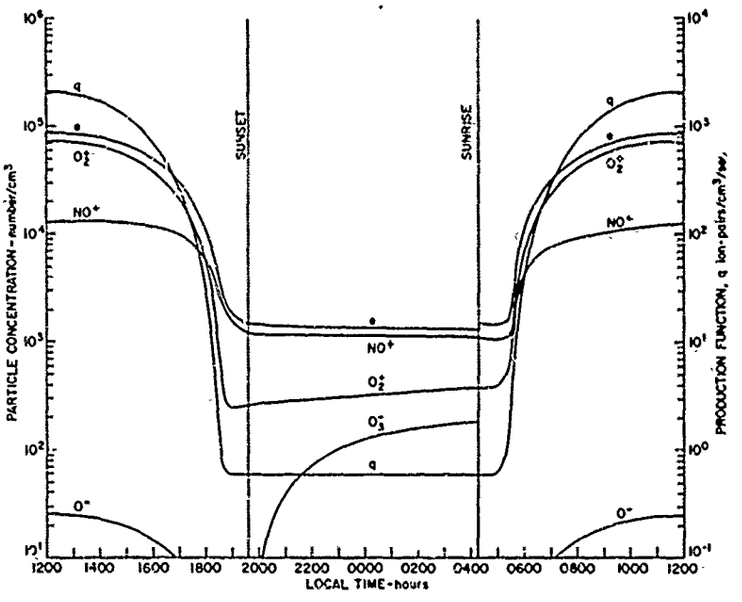

Figure 21. Diurnal Variation of Charged Species at 100 km (Brazil noontime)

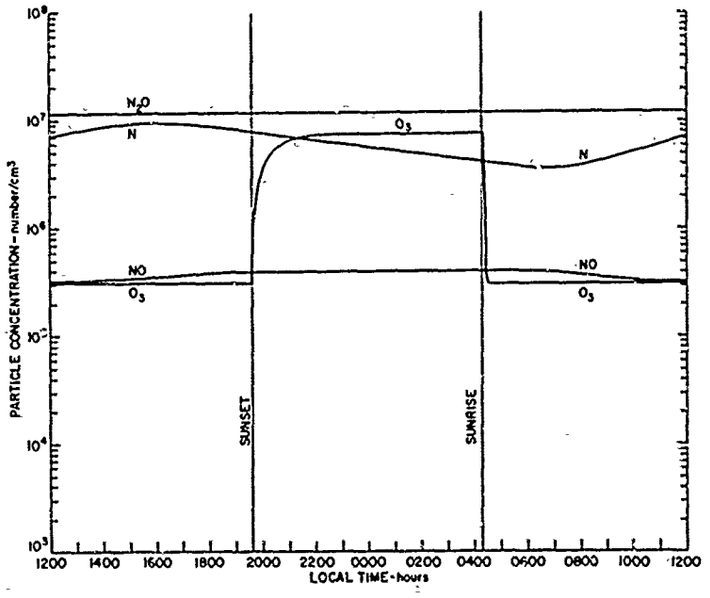

Figure 22. Diurnal Variation of Neutral Species at 100 km (Brazil noontime)

a greater diurnal variation in ozone. Below 100 km the major source of ozone is the three-body recombination reaction $O + O_2 + M \rightarrow O_3 + M$. When photodissociation stops at sunset the ozone concentration increases because of this recombination. The time constant for removal of ozone by the neutral rearrangement reaction $O + O_3 \rightarrow 2O_3$ is about one hour. After this time, therefore, the ozone goes into equilibrium which is established by these two processes.

At 90 km the diurnal variation of the charged species starts to show a behavior different from that at higher altitudes (see Figure 23). The production function no longer shows the smooth decay representative of the absorption of ultraviolet radiation. Around 90 km during the day, the primary source of ionization is the soft X-rays with the UV radiation almost completely absorbed above this altitude. As the solar zenith angle increases, more and more of these X-rays are absorbed because of the increasing air mass in the column. The L_α radiation, however, suffers less absorption in the column as the solar zenith angle increases. A point is reached, therefore, where the incident X-ray flux becomes less than the incident L_α flux. This change over to L_α production is evidenced by the bumps in the production curve in Figure 18 around 1800 and 0600 hours. At this altitude during the day O_2^+ is still the major positive ion and the electrons are still lost by recombination. During the night, however, the rate of electron removal by recombination becomes relatively small because of the low electron density. The process that starts to become important at this altitude is the three-body attachment $O_2 + e + O_2 \rightarrow O_2^- + O_2$. During the night, therefore, the electrons are lost by attachment to O_2. The O_2^- ions thus formed do not stay around very long because the rate of the charge transfer reaction $O_2^- + O_3 \rightarrow O_3^- + O_2$ is increasing rapidly with the increasing ozone concentration. As soon as the O_2^- ions are formed they immediately transfer their charge to ozone forming O_3^- ions. The O_3^- ions increase until an equilibrium is established between the charge transfer and the mutual neutralization reaction $O_3^- + NO^+ \rightarrow O_3 + NO$. As soon as the sun starts to rise, the electrons are detached from the O_3^- ions and in a few seconds all of the electrons that were attached during the night are detached. Because the solar radiation must penetrate very dense air columns just after sunrise, the production function does not become effective in producing ionization for about an hour after the sun has come up. During this period of time, with no source of ionization present, the electrons recombine with the positive ions, causing the ionization to start to decay. When the production function becomes effective in producing ionization, the electron and positive ion concentrations start to increase and proceed to their noontime equilibrium values.

The diurnal variation of the minor neutral species at 90 km is shown in Figure 24. The behavior at this altitude is similar to that at 100 km. The ozone concentration displays a greater diurnal variation at this altitude because of the increase in the rate of the three-body recombination.

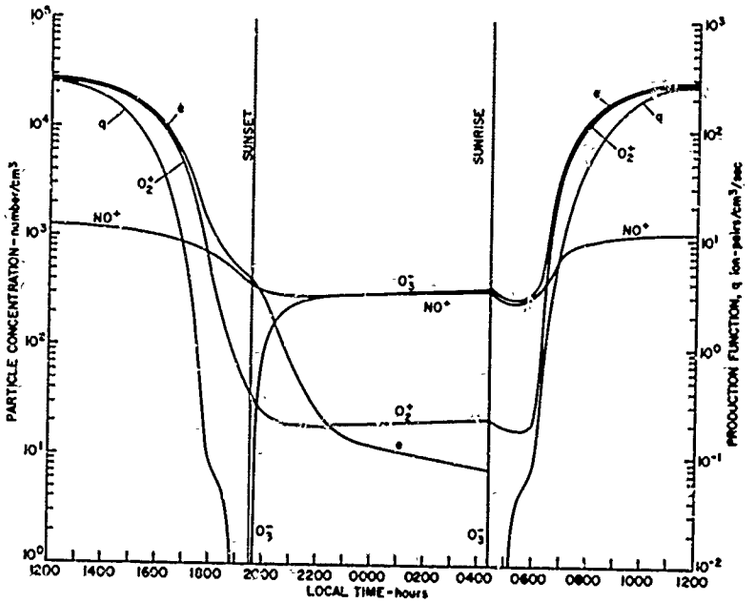

Figure 23. Diurnal Variation of Charged Species at 90 km (Brazil noontime)

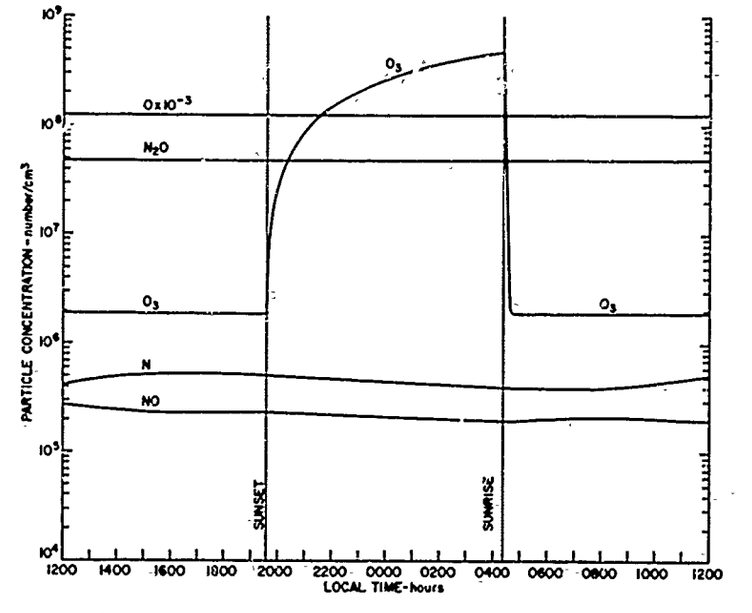

Figure 24. Diurnal Variation of Neutral Species at 90 km (Brazil noontime)

The diurnal variation of the major charged species at 80 km is shown in Figure 25. The daytime ionization picture is now changing considerably. The major positive ion during the entire day is NO^+. The O_2^+ concentration is about two orders of magnitude below the NO^+ concentration. The electrons and positive ions decay during the afternoon by recombination. The O_2^- density follows the electron concentration because of quasi-equilibrium established between the three-body attachment and the photodetachment. The O_3^- concentration follows the other negative ions because of the equilibrium between the charge transfer $O_2^- + O_3 \longrightarrow O_3^- + O_2$ and the photodetachment. As soon as the sun starts to set, the electrons become attached to O_2. This attachment continues until the rate of removal is balanced by the rate of formation by the source function. At this altitude also the O_2^- ions formed by electron attachment are short lived, transferring their charge to ozone to form the O_3^- ions. When the dissociative recombination of NO^+ stops at sunset because of the decreased electron density, the NO^+ concentration increases by the nighttime production function until a quasi-equilibrium is established between O_3^- and NO^+ as determined by the nighttime source and the time constant of the mutual neutralization process $O_3^- + NO^+ \longrightarrow O_3 + NO$. The electron concentration decays slowly throughout the night because of attachment to O_2. The resulting O_2^- ions thus formed immediately transfer their charge to ozone to form O_3^- ions. The behavior of the ionization after sunrise is the same as at 90 km except that three-body attachment and negative charge transfer start to bring negative ions into the picture.

Figure 26 shows the diurnal variation of the minor neutral species at 80 km where again there is the large diurnal variation in ozone. Atomic oxygen is now starting to show a diurnal variation also. The atomic-nitrogen behavior is quite different from what it was at higher altitudes. Below 80 km its kinetics are determined solely by neutral reactions. When the sun sets the photodissociation of N_2O stops thus halting the production of N atoms. With no other source of production present for these atoms, they decay during the night by the netural rearrangement process $N + NO \longrightarrow N_2 + O$. After sunrise the concentration of this atom increases until an equilibrium is established between the photodissociation of N_2O and the neutral rearrangement.

The diurnal variation of the charged species at 70 km is shown in Figure 27. The behavior of the species at this altitude is controlled by the same process as at 90 km. With the values of the rate constants for the photodetachment from O_3^- (1.4) and for the charge transfer reaction $O_2^- + O_3 \longrightarrow O_3^- + O_2$ (7.77×10^{-9}) used in this work a negative ion-to-electron ratio of about unity is obtained at 70 km. The negative ion concentrations are higher here than they are at 80 km because of the higher rate of the three-body attachment reaction which, in turn, results from the higher neutral concentration at 70 km.

Figure 28 shows the diurnal variation of the minor neutral species at 70 km. At this altitude, atomic nitrogen becomes a trace species and is therefore not plotted.

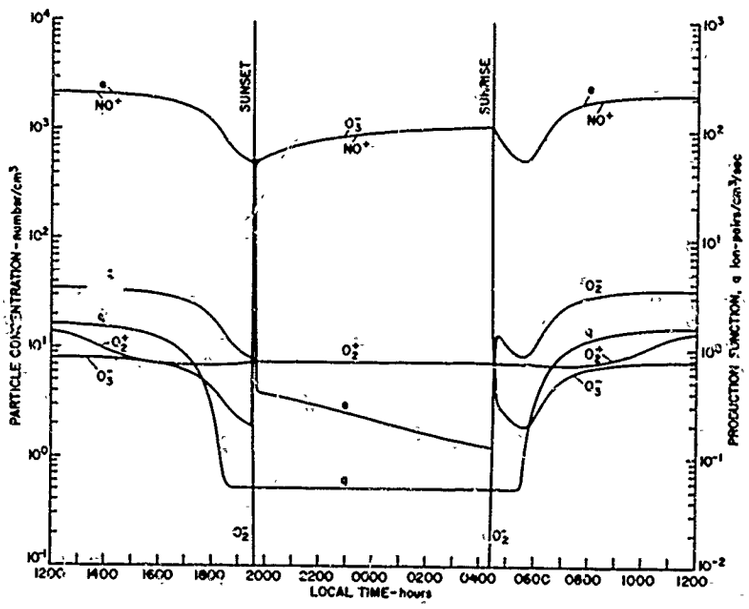

Figure 25. Diurnal Variation of Charged Species at 80 km (Brazil noontime)

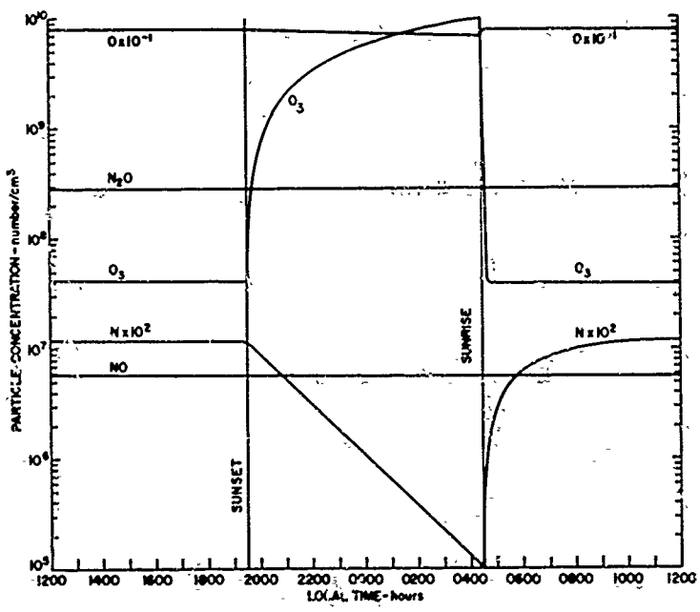

Figure 26. Diurnal Variation of Neutral Species at 80 km (Brazil noontime)

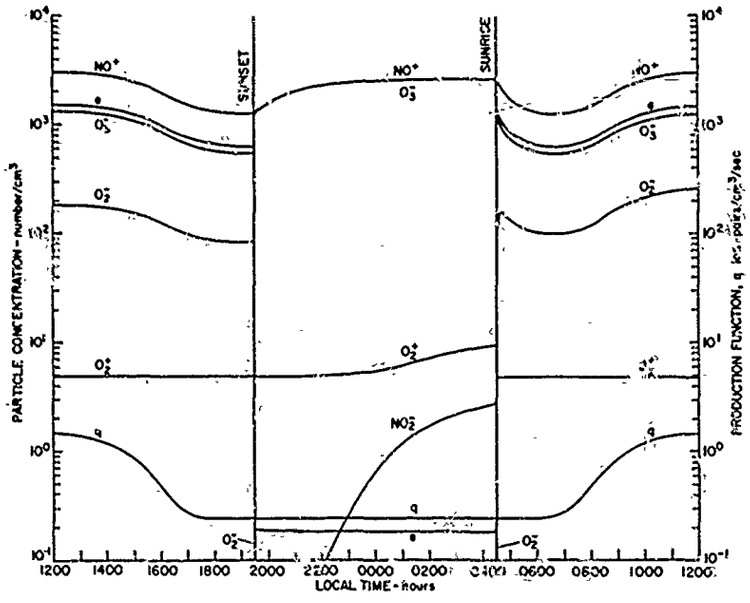

Figure 27. Diurnal Variation of Charged Species at 70 km (Brazil noontime)

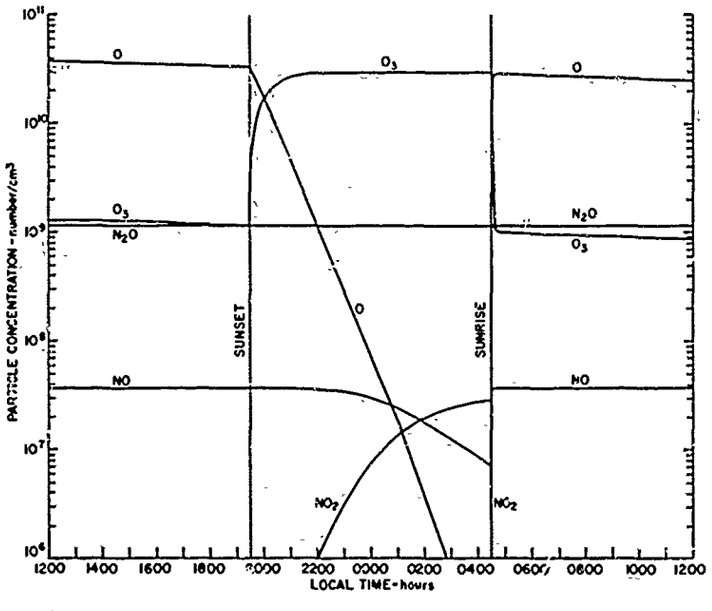

Figure 28. Diurnal Variation of Neutral Species at 70 km (Brazil noontime)

The striking feature of Figure 28 is the considerable diurnal change that takes place in the atomic oxygen concentration. The molecular oxygen and total concentration have become sufficiently high to make the three-body recombination process $O_2 + O + M \rightarrow O_3 + M$ proceed very rapidly. During the night when there is no photodissociation of ozone to replenish the atomic oxygen supply, the three-body recombination converts practically all of the available oxygen atoms to ozone. When the sun comes up, however, the photodissociation of ozone is so rapid that the oxygen atoms that recombined during the night are restored in a few minutes. The nitric oxide concentration also exhibits a substantial diurnal change at 70 km. The neutral rearrangement reaction $O + NO_2 \rightarrow NO + O_2$, the major formation process for this molecule during the day, decreases in effectiveness as the atomic oxygen disappears during the night. Since there is no other effective formation process for this molecule, its concentration decreases through the neutral rearrangement reaction $N + NO \rightarrow N_2 + O$

The nitric oxide concentration returns quickly to its daytime level when the sun rises and the atomic oxygen is restored. Nitrogen dioxide is an extremely minor species down to 70 km. At this altitude it exhibits a sufficient diurnal variation so that by the end of the night it is almost equal in concentration to the nitric oxide. The concentration of this molecule is controlled by its photodissociation and the three-body recombination reaction $O + NO + M \rightarrow NO_2 + M$. During the night when there is no photodissociation, the concentration of this molecule increases by the three body atom recombination. When the sun comes up, the nitrogen dioxide is dissociated very rapidly.

The diurnal variation of the major charged species at 60 km is shown in Figure 29. Because there is no change in the production function throughout the day and because the time constant for recombination is long, there is little change in the electron and NO^+ concentration during the afternoon. The diurnal variation solution at 60 km should have been run over another diurnal cycle. It is apparent that the concentrations at the beginning and at the end of the solution shown in Figure 29 do not coincide sufficiently well. Nevertheless, O_3^- is the most abundant negative species throughout the entire day. During the night the NO_2^- ion, which was a trace ion at all altitudes above 60 km, becomes the second most abundant negative species. This ion follows the nitrogen dioxide profile shown in Figure 30 since it is formed by the charge transfer $O_3^- + NO_2 \rightarrow NO_2^- + O_3$. When the sun comes up and the concentration of NO_2 decreases again, the NO_2^- ions transfer their charge back to ozone, $NO_2^- + O_3 \rightarrow O_3^- + NO_2$.

Figure 30 shows the diurnal variation of the minor neutral species at 60 km. The ozone still undergoes a diurnal variation but the magnitude of the variation is less at 60 km because there is less atomic oxygen available to produce it. Within about an hour after sunset practically all of the atomic oxygen has recombined.

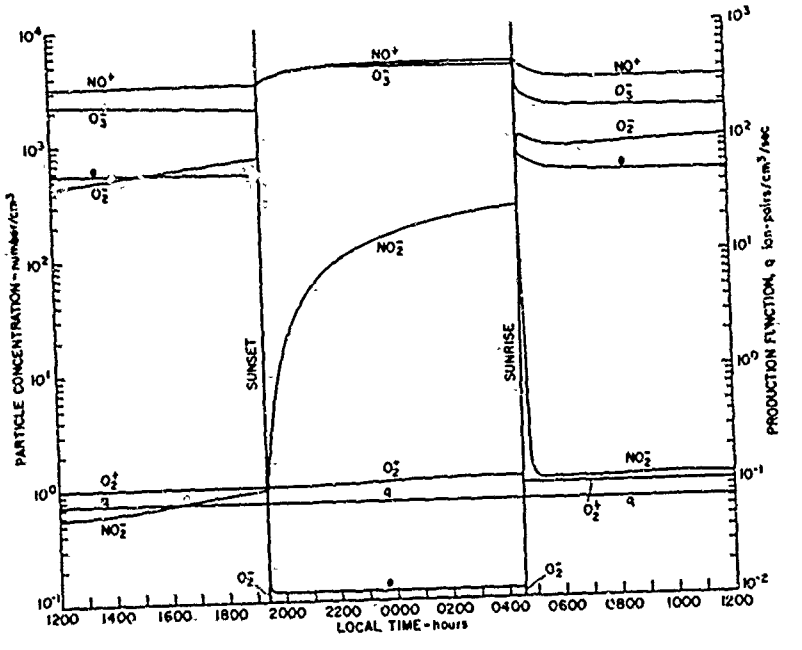

Figure 29. Diurnal Variation of Charged Species at 60 km (Brazil noontime)

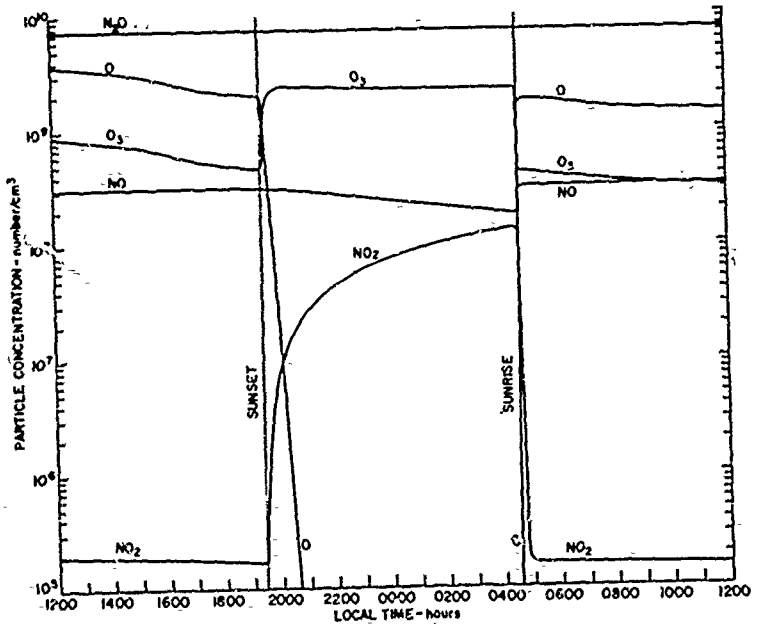

Figure 30. Diurnal Variation of Neutral Species at 60 km (Brazil noontime)

The nitrogen dioxide concentration again exhibits a large diurnal variation arriving at a concentration almost equal to that of nitric oxide at sunrise. The nitric oxide undergoes a lesser variation at 60 km than it did at 70 km because the formation of this molecule stops when the atomic oxygen becomes depleted. Its removal by the neutral rearrangement reaction $N + NO \longrightarrow N_2 + O$ is then much slower because of the greatly reduced atomic nitrogen concentration.

5.3.3 VERTICAL PROFILES OF THE SPECIES

In order to obtain a composite picture of the curves discussed in Section 5.3.2, the concentrations of the species were plotted as a function of altitude for noontime and midnight. Figure 31 shows the vertical profile of the charged species at noontime. The electrons remain the most abundant negative species down to 70 km. Below this altitude O_3^- becomes the most abundant negative species. Above 90 km O_2^+ is the most abundant positive ion. Between 80 and 90 km NO^+ replaces O_2^+ as the most abundant positive ion and remains so at all altitudes below 80 km. Nowhere are O^+, N_2^+, O^-, and O_2^- important ions. In the lower D region O_2^- becomes important.

The vertical distribution of the neutral species at noontime is shown in Figure 32. The NO profile remains fairly constant down to 90 km while below this altitude it increases steadily with decreasing altitude. The ozone profile appears to have a maximum around 70 km. Above this altitude it falls off rapidly to a very minor constituent. The N_2O concentration increases with decreasing altitude throughout the entire range. Atomic nitrogen appears to have a peak in the E region around 110 km and becomes a very minor species in the D region. The atomic oxygen profile is that obtained by the chemistry after the second diurnal variation cycle. In this time atomic oxygen has not gone into chemical equilbrium in the E region. Several more diurnal cycles would have had to be made to obtain this equilibrium. The dashed curve is the initial profile assumed for atomic oxygen.

Figure 33 shows the vertical profile of the charged species at midnight. The distribution is considerably different from that shown in Figure 31 for noontime. The electrons remain the most abundant negative species in the E region. Below 95 km, however, O_3^- is the most abundant negative species. At all altitudes NO^+ is the predominent positive ion. No other ion is important except for O_2^+ in the E region and NO_2^- at 60 km.

The height profiles for the neutral species at midnight are shown in Figure 34. There is little change in the profile of atomic oxygen down to 80 km but below this altitude it suffers a severe diurnal variation. The ozone profile has shifted to higher concentrations by a little more than an order of magnitude. This reflects the increased production of this molecule during the night by the three-body atom recombination reaction. Atomic nitrogen varies little above 90 km but below this

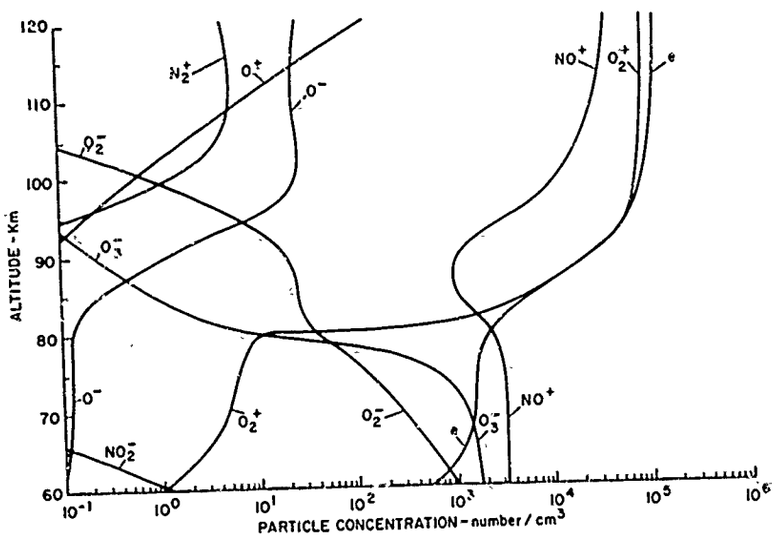

Figure 31. Vertical Distribution of Charged Species (Brazil noontime)

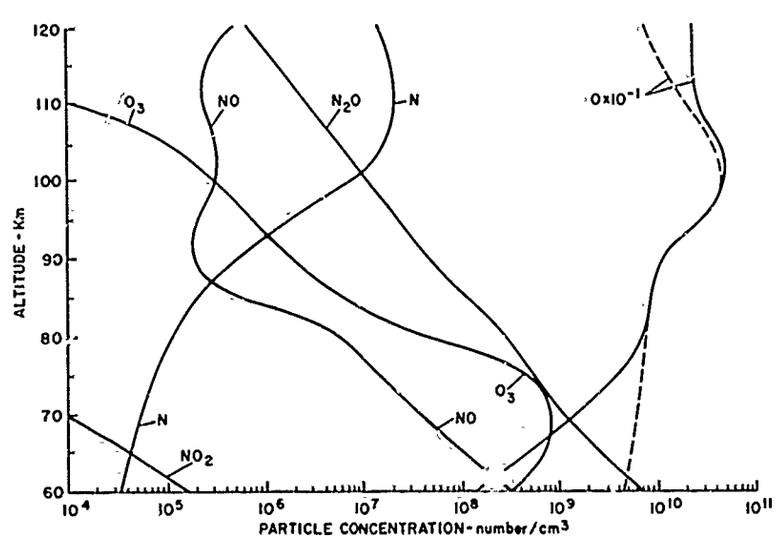

Figure 32. Vertical Distribution of Neutral Species (Brazil noontime)

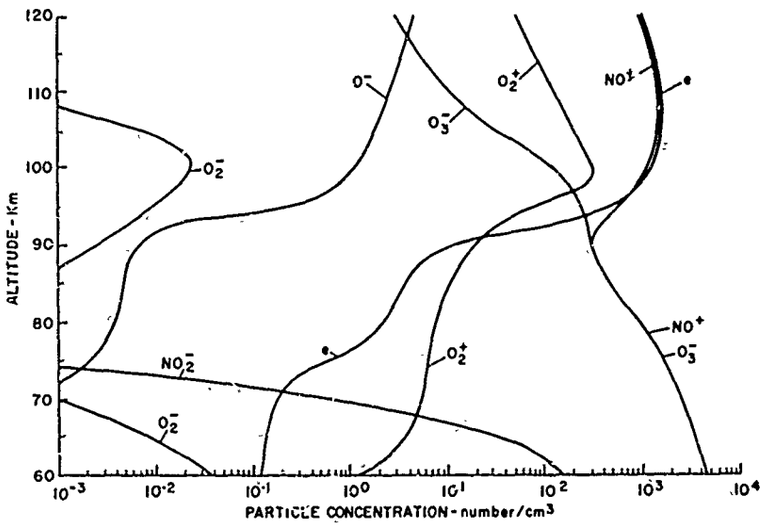

Figure 33. Vertical Distribution of Charged Species (Brazil midnight)

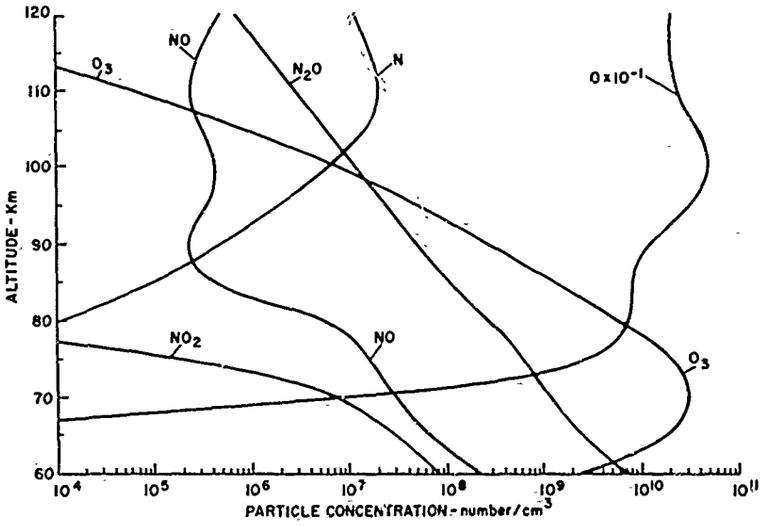

Figure 34. Vertical Distribution of Neutral Species (Brazil midnight)

altitude it undergoes considerable diurnal change. The nitric oxide and the nitrogen peroxide profiles show very little change from noontime to midnight. The nitrogen dioxide concentration is much higher at midnight than at noontime below 80 km, again showing diurnal variation.

6. DISCUSSION OF THE CODE

6.1 Example of How Experimentel Evidence is Used to Adjust Rate Constants

It would be interesting at this point to present one example of a solution using rate constants for some reactions different from those already discussed. This will demonstrate how the code can be used to detect discrepancies in the values of rate constants or difficiencies in the chemistry. Figure 35 shows a diurnal variation solution at 70 km using the same code that produced Figure 27. The only differences between the two runs are the values of two rate constants and the inclusion of two additional reactions. The two reactions whose rate constants are changed are

$$O_3^- + h\nu \rightarrow O_3 + e \qquad k_4 = 0.04$$

$$O_2^- + O_3 \rightarrow O_3^- + O_2 \qquad k_{96} = 1\times10^{-9}$$

These are the rate constants listed in the basic list in Section 3 as compared to 1.4 and 7.77×10^{-9} respectively, used for the diurnal variation calculations. The two additional reactions are

$$O_3^- + O \rightarrow O_2 + e + O_2 \qquad k_{167} = 5\times10^{-11}$$

$$NO_2^- + O \rightarrow O_2 + NO + e \qquad k_{168} = 1\times10^{-13}$$

The two rate constant changes have the effect of slowing down both of the reactions. The rate constants for these four reactions are the same as those used to compute the deionization profiles in Section 4.

Since fewer O_2^- ions are losing their charge to ozone, the second largest negative species in Figure 35 is O_2^- whereas in Figure 27 it is O_3^-. Otherwise the behavior of the charged species during the day is much the same in both Figures 27 and 35. During the night, however, the behavior is quite different. The rapid decay of the electrons by attachment to O_2 and of the O_2^- ions by charge transfer to ozone as seen in Figure 27 does not take place in Figure 35, although the electrons still attach to O_2 to form O_2^-. In this computation the

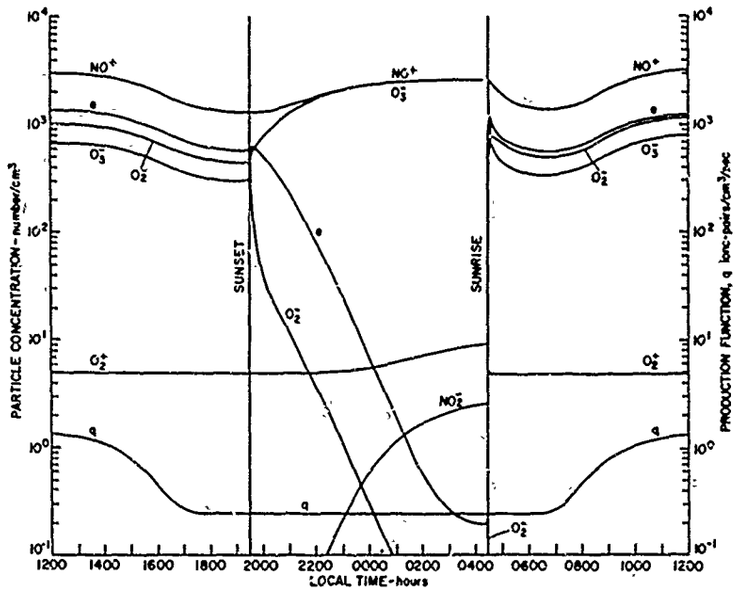

Figure 35. Diurnal Variation of Charged Species at 70 km Using Different Reaction Rates

associative detachment reaction $O_3^- + O \rightarrow 2O_2 + e$ is the fastest process in the kinetics, so that the electrons consumed by attachment are immediately restored. Since the associative detachment reaction is the controlling process for the electrons, the electron density is determined solely by the behavior of O_3^- and atomic oxygen. Accordingly, during the night the electron concentration decays slowly along with the atomic oxygen concentration. In this run the atomic-oxygen profile is the same as that shown in Figure 28. Since the O_2^- density goes into quasi-equilibrium with the electrons shortly after sunset, it decays during the night at a constant ratio to the electron density.

It is obvious that this nighttime behavior is contrary to all experimental evidence where the electrons do decay rapidly after sunset at this altitude. Some of the conclusions that can be drawn are the following:

a) The associative detachment reaction $O_3^- + O \rightarrow 2O_2 + e$ either does not take place or it has an extremely small rate constant.

b) There is some chemical reaction missing that is faster than the associative detachment for the removal of O_3^- and which does not produce electrons.

c) Atomic oxygen recombines much more rapidly that it does in this solution so that the associative detachment reaction becomes ineffective much sooner after sunset.

Bortner (1966) stated that the products of the associative detachment reaction are most probably not O_2 and an electron but O_2 and O_2^- and that the reaction has a rate constant of about 1.4×10^{-10}. If this is the case the negative ion picture will certainly be different and the strange behavior of the electrons, as shown in Figure 35, will be eliminated.

6.2 Recommendations

It can be assumed that the complete chemical behavior of the atmospheric constituents is not contained in the long list of reactions built into the code. Not only could significant chemical processes be missing but also important constituents may be omitted. For example, it is possible that the NO_2^+ ion is an important charged species. In work recently presented by Fehsenfeld et al. (1966), they have indicated that the ions CO_3^- and NO_3^- might be important. Narcisi (1965) detected the presence in the ionosphere of metallic ions as well as water vapor and its derivative molecular ions. Therefore, it is evident that the chemical make-up of the ionosphere has not been completely defined. As new information becomes available on the presence of any particular species in the ionosphere, the code will have to up-dated to include this species as soon as reasonable chemistry for its kinetics can be determined. The code on the other hand is written so that it can accommodate such additions with almost no effort.

It should be pointed out that more information is needed on some of the reaction rates. For application to the sunrise problem, for instance, it is essential that work be done to determine the spectral dependence of the photodetachment of the important negative ions as well as the spectral dependence of the photodissociation of the important minor neutral molecules. Since the code produces results that are only as good as our knowledge of the rate constants of the reactions, laboratory measurements of the important processes not yet studied must be undertaken. Another serious deficiency is the almost complete lack of insitu data on the identification of the most abundant negative ion in the D region. Until experimentalists have determined which ion this is it will be difficult to improve the chemistry of the D region.

As more data become available the code should be able to produce profiles that are more in conformity with experimental measurements and thereby increase its potential as a prediction technique. Meanwhile, the code remains a powerful tool for investigating complicated ionospheric phenomena that cannot be adequately explained by simpler means.

All of the solutions discussed in this paper were obtained using the code as described in Section 7. This code was written to compute only the photochemical behavior of the atmospheric constituents. It does not pretend to be able to solve the complete dynamical problem. To make the code completely general such

physical phenomena as diffusion, turbulent mixing, or eddy diffusion and wind motions and shears must necessarily be included. It is conceivable that none of these dynamic forces are of great importance to the behavior of the charged species incorporated into the present code because of the short time constants of the important chemical reactions. For the neutral species, however, these transport mechanisms can be significant especially if any of these motions have time constants shorter than the time constants of the important chemical processes. From the work that Colgrove, Johnson, and Hanson (1966) have done on the eddy diffusion of atomic oxygen, it is obvious that the problem of mixing cannot be ignored.

For most applications the transport problems are of minor importance. Therefore, it is not intended at the present time to extend the code in this direction. For certain minor neutral species it might become necessary to establish initial conditions by taking vertical transport into account. However, it is anticipated that this might be accomplished satisfactorily by a semi-empirical approach.

7. DESCRIPTION OF THE CODE

The code consists of a main program and several subroutines all written in FORTRAN-IV language for an IBM-7044 computer. Application of the code to any FORTRAN-compatible computer is easily made. The basic program package consists of the following programs:

a) Main program
b) Subroutine INTEG
c) Subroutine ALGA
d) Subroutine SLOP
e) Subroutine DAUX
f) Subroutine INITIAL
g) Subroutine PRODUC
h) Subroutine BALAN
i) Subroutine DAUXT

7.1 Deionization Codes

Two of the simpler problems to which the code can be applied are the buildup of ionization under the influence of a constant or variable source of ionization and the decay of ionization from some initial values with or without an external source of ionization. The results presented in Section 4 were obtained using the code as described here.

7.1.1 MAIN PROGRAM RATEQ

The functions of the main program are to fix a standard set of rate constants in DATA statements to control the input of necessary parameters and the output of results, to initialize the entire code before starting a solution, to monitor the flow of the computations, and to determine the equilibrium status of the species.

7.1.1.1 The Rate Constants

In order to avoid reading the rate constants into the computer for every run, a standard set of rate constants was built into the code. These rate constants are listed in Section 3. Every rate constant is considered to be of the form

$$k = A \cdot T^{B} \cdot e^{-C/T} \tag{39}$$

where A, B, and C are constants and T is the temperature in degrees Kelvin. The constants A are set into regions A1(69), A2(69), and A3(30). This split up is necessary because of the restriction on the number of CONTINUATION cards allowed by the version of FORTRAN that is used. The B's and C's are set into regions B(168) and C(168) respectively. After the temperature and any required changes in the rate constants are read into the computer, the actual rate constants are computed from Eq. (39) and put into the CON region.

7.1.1.2 Determination of Equilibrium

The decision as to whether a species has reached its equilibrium or quasi-equilibrium value is made by the main program. For this purpose the four regions LOCK (15), KEY(15), CRTNO(15), and CRITN(15) are used. Upon completion of an integration, the values

$$\text{CRTNO(J)} = N_j \sum R_i \tag{40}$$

and

$$\text{CRITN(J)} = \frac{\Delta N_j}{\Delta t} \tag{41}$$

are computed. If the slopes $\frac{\Delta N_j}{\Delta t}$ are less than 10^{-3}, the criterion for quasi-equilibrium is

$$\left| 1 - \frac{\sum F_i}{N_j \sum R_i} \right| < \text{DEL} \quad . \tag{42}$$

If the slopes $\frac{\Delta N_j}{\Delta t}$ are greater than 10^{-3}, the criterion for quasi-equilibrium is

$$\frac{\left|\frac{\Delta N_j}{\Delta t}\right|}{N_j \sum R_i} < DEL \quad . \tag{43}$$

The equilibrium status of each species is determined by using that criterion which pertains.

If the criterion fails for any species, the respective LOCK(J) is set to zero. If the test is satisfied, a one is added to the respective LOCK(J). After the criterion is tested for all the species, the LOCK(15) are tested to determine if any have reached three. If a particular LOCK(J) is equal to or greater than three, the respective KEY(J) is set to two. Following this another test is made to determine if any species has decayed below the value CHI. If any one has, the respective KEY(J) is set equal to three. Once a KEY(J) is set to three it will remain at that value for the remainder of the solution. The values in the KEY region are used by several of the subroutines for branching, depending on the status of the concentrations of the species.

The status of any particular species can be determined at any time by simply investigating its KEY value. The species j will not be in equilibrium if its KEY is on one; it will be in equilibrium if its KEY is on two; its concentration will be zero or a constant; its equations are removed from the sets if its KEY is on three; and, its concentration will be computed from the charge balance equation

$$\sum N_- = \sum N_+ \tag{44}$$

if its KEY is on four.

7.1.1.3 Input Parameters

Those parameters that the code requires to solve a specific problem are put on cards that immediately follow the binary decks for the program. The binary decks and the BCD data cards are read into the computer from FORTRAN-tape 5. The purpose of each data card and the FORMAT in which it is prepared are as follows:

CARD 1 FORMAT(12A6)
This card contains up to 70 BCD characters that can be used for identification at the discretion of the user.

CARD 2 FORMAT(1P4E12.5)
This card contains four variables:

(1) EUBAR - the criterion for the solution in the integrating routine. This must be some number less than 10^{-2} which is determined only by the degree of accuracy desired in the integrated values.

(2) ELBAR - the criterion for the solution of the exponential equations. This must be some number less than 10^{-2} which is determined by the amount of charge imbalance that can be tolerated in the results.

(3) DEL - the criterion for equilibrium. This should be some number around 10^{-2}. If it is much larger than this, a species will be put into quasi-equilibrium too soon and the solution of the exponential equations can fail. If it is much smaller than 10^{-2} and a species is in equilibrium, its differential equation may not be removed from the set. The set of differential equations cannot be solved with the current increment if the set contains an equation for a species whose concentration is in equilibrium or quasi-equilibrium. In order to solve such a set of equations, the increment will have to be reduced considerably at the expense of much computer time.

(4) ENDT - the time in seconds to which the solution is to run. This is the time associated with dN/dt and is not to be confused with the running time of the code on the computer.

CARD 3 FORMAT(I4)

This card contains the integer NOC indicating the number of rate constants that are to be changed for the run. If no changes are necessary, this card must contain a zero. Following this card are the changes themselves. If there are no changes, CARD 4 is read next. The changes are punched one per card in FORMAT(I4,1PE10.2,OPF5.1, 1PE10.2,7A6). Field 1 is the reaction number as listed in Section 3, fields 2, 3, and 4 are the A, B, and C in Eq. (39), field 6 represents any 42-BCD characters that the user wishes to insert as comments.

CARD 4 FORMAT(1P5E12.5)

This card contains five input parameters:

(1) ALT - the altitude in centimeters at which the solution is to be made.

(2) D - the total number density in cm^{-3}.

(3) DO2 - the concentration of molecular oxygen in cm^{-3} at the given altitude.

(4) DN2 - the concentration of molecular nitrogen in cm^{-3} at the given altitude.

(5) T - the temperature in degrees Kelvin at the given altitude.

CARDS 5-7 FORMAT(1P6E12.5)

These cards contain the initial concentrations of the species in the following order: electrons, O^-, O_2^-, O_3^-, NO_2^-, O^+, O_2^+, N_2^+, NO^+, NO, N, NO_2, O_3, N_2O, and atomic oxygen. The last value in this set is the constant production rate in number of ion-pairs/cm^3/sec.

CARD 8 FORMAT(4I2)

This card contains the option switches.

KB1=0 for logarithmic output.

KB1=1 for decimal output.

KB2=0 calls DAUXT to print the history of the reactions.

KB2=1 suppresses the call of DAUXT.

KB3=0 prints this history after every integration if DAUXT is called.

KB3=1 prints this history only once for each decade of time if DAUXT is called.

KB4=0 returns control to the system on an error.

KB4=1 program returns to the input area to read another set of data cards when an error occurs.

It is possible to stack as many sets of these data cards as desired because the program always returns to the input area when it has completed the computations for a given set of data.

7.1.1.4 Programmed Variables

There are other parameters that can be changed by reassembly of the main program:

a) CHI - the lowest value to which a species may decay. This test is incorporated to prevent computer underflow in the event that any species concentration decays below 10^{-38}. The limiting value used in the programs given here is 10^{-10}. If a species should try to decay below this value, its concentration is set to zero and its equations are removed from the sets by setting its KEY on three.

b) NOCOM - the number of reactions built into the system.

c) ENDE - a preset lower limit on the electron density. This test is included in order to terminate the solution when the electron density falls below a certain value. This permits the computation of densities only over the range of interest. When the electron concentration decays below this value, control is transferred to the input area of the code. If no limit is to be placed on the electron density, this value must be set to zero.

d) ITER - the maximum number of iterations that will be made in attempting to solve the simultaneous exponential equations. If a solution cannot be found within this number of iterations, iteration of the exponential equations will stop and the unsuccessful return from subroutine ALGA will be taken.

e) NUMB - the number of differential equations in the set to be solved.

7.1.1.5 Output

FORTRAN-tape 6 is the normal system output tape. When the computations are completed, this tape will contain all of the output information with the exception of the history of the reactions. The first thing written on this tape is the comment read into the computer on the first data card. This card contains any 72 hollorith characters that the user wishes to insert as identification. Following this the program writes the list of changes that are made in the build-in standard list of reaction-rate constants. The program then writes the altitude at which the calculations are being made, the total number density, and the temperature. After this is written, the rate constants for the reactions with the coefficient A first and underneath this the total reaction rate with the proper temperature dependence. Following this is the time and the computed concentrations for this time of all the the negative and the positive species.

During the execution of the program the time, the concentrations of the neutral species computed for this time, the value of the production function, and the setting of the KEY switches are written on Tape 1 in binary. Upon completion of the solution

for a particular set of input data, this information is automatically transferred to Tape 6. Tape 1 is left in a rewound condition.

FORTRAN-tape 4 is used by subroutine DAUXT for writing the history of the reactions if such a history is requested. Upon completion of the computations, this tape must be removed for off-line printing.

The following statements, beginning on page 72, are a listing of the main program.

7.1.2 THE INTEGRATING SUBROUTING INTEG

This subprogram is called by the main program and performs the numerical integration of the differential equations. For solving a set of differential equations

$$\frac{dN_1}{dt} = F_1(t, N_1, N_2, N_3 \ldots N_m) ,$$

$$\frac{dN_2}{dt} = F_2(t, N_1, N_2, N_3 \ldots N_m) ,$$

$$\frac{dN_3}{dt} = F_3(t, N_1, N_2, N_3 \ldots N_m) ,$$

$$\vdots$$

$$\frac{dN_m}{dt} = F_m(t, N_1, N_2, N_3 \ldots N_m) .$$

The Kutta Merson solution uses the equations

$$N_{j1} = N_{j0} + \tfrac{1}{3} \Delta t \, F_j(t_0, N_{10}, N_{20}, N_{30} \ldots N_{m0})$$

$$N_{j2} = N_{j0} + \tfrac{1}{6} \Delta t \, F_j(t_0, N_{10}, N_{20}, N_{30} \ldots N_{m0}) + \tfrac{1}{6} \Delta t \, F_j(t_0 + \tfrac{1}{3} \Delta t, N_{11}, N_{21}, N_{31} \ldots N_{m1})$$

$$N_{j3} = N_{j0} + \tfrac{1}{8} \Delta t \, F_j(t_0, N_{10}, N_{20}, N_{30} \ldots N_{m0}) + \tfrac{3}{8} \Delta t \, F_j(t_0 + \tfrac{1}{3} \Delta t, N_{12}, N_{22}, N_{32} \ldots N_{m2})$$

```
$IBFTC RATEQS  LIST                                                    RATE0000
C    SOLUTION OF THE REACTION RATE EQUATIONS IN THE IONOSPHERE FOR 15 SPERATE0005
C    IES AND WITH 168 REACTIONS.                                       RATE0010
C      KB1=0 FOR LOG OUTPUT.                                           RATE0015
C      KB1=1 FOR DECIMAL OUTPUT                                        RATE0020
C      KB2=0 PRINTS THE HISTORY OF THE REACTIONS.                      RATE0025
C      KB2=1 REACTION HISTORY NOT COMPUTED.                            RATE0030
C      KB3=0 PRINTS HISTORY AFTER EACH INTEGRATION.                    RATE0035
C      KB3=1 PRINTS HISTORY ONCE EACH DECADE OF TIME.                  RATE0040
C      KB4=0 HALTS ON ERROR.                                           RATE0045
C      KB4=1 READS ANOTHER POINT CARD AFTER AN ERROR.                  RATE0050
C      IPLOT=0 NO PLOTTER TAPE MADE.                                   RATE0055
C      IPLOT=1 MAKES A LOG PLOTTER TAPE.                               RATE0060
C      UNITS OF INPUT PARAMETERS ARE CGS.                              RATE0065
C                                                                      RATE0070
       COMMON TREG(150),KEY(15),FORM(15),REMV(15),CON(168),LKEY(15),BEGINRATE0075
      1(15)                                                            RATE0080
       COMMON NUMB,EUBAR,ELBAR,D,DO2,DN2,T,PNE,PNO,PO2,PO,PN2,TOTAL,JACK,RATE0085
      1JAKE,JAM,ITER,LINT,I2NT,K2NT,J2NT,N2NT,FIRST,PROD,LAM,TIME,KB3,TOTRATE0090
      20,TOTN                                                          RATE0095
       COMMON VQ(30),IQ(30),SIGN(30),ITEM                              RATE0100
       DIMENSION DONT(20),TITLE(12),TTREG(20),CRITN(15),CRTNO(15),A(168),RATE0105
      18(168),C(168),LOCK(15),ALF(7),G(168),A1(69),A2(69),A3(30)       RATE0110
       DATA A1/ 0.44E+00,1.40E+00,0.04E+00,0.04E+00,9.00E-15,3.60E-16,  RATE0115
      13.60E-16,1.00E-13,1.00E-13,1.00E-13,1.00E-13,1.00E-13,1.00E-13,  RATE0120
      21.00E-13,1.00E-13,1.31E-15,1.00E-19,1.00E-17,1.00E-17,1.00E-11,  RATE0125
      31.00E-11,1.00E-31,1.00E-31,1.40E-31,5.80E-33,1.90E-33,6.00E-28,  RATE0130
      44.00E-29,0.00E-00,6.00E-05,9.00E-05,1.50E-04,1.00E-24,1.00E-22,  RATE0135
      51.00E-22,1.00E-22,1.00E-23,2.20E-10,1.00E-12,1.00E-12,1.00E-12,  RATE0140
      65.00E-07,5.00E-07,5.00E-07,5.00E-07,5.00E-07,5.00E-07,5.00E-07,  RATE0145
      75.00E-07,5.00E-07,5.00E-07,5.00E-07,5.00E-07,5.00E-07,5.00E-07,  RATE0150
      85.00E-07,5.00E-07,1.00E-23,1.00E-23,1.00E-23,1.00E-23,1.00E-23,  RATE0155
      91.00E-23,1.00E-23,1.00E-23,1.00E-23,1.00E-23,1.00E-23,1.00E 11/  RATE0160
       DATA A2/ 1.00E-13,1.00E-13,0.00E-00,1.00E-13,1.00E-13,1.00E-11,  RATE0165
      11.00E-11,1.00E-13,0.00E-00,1.00E-13,1.00E-13,1.00E-13,1.00E-13,  RATE0170
      24.00E-11,2.40E-11,0.00E-00,0.00E-00,8.00E-10,1.00E-12,1.00E-10,  RATE0175
      30.00E-00,5.00E-12,1.00E-09,1.00E-09,1.00E-12,1.00E-09,1.00E-09,  RATE0180
      41.00E-09,1.00E-09,1.00E-16,1.00E-18,0.00E-00,0.00E-00,0.00E-00,  RATE0185
      50.00E-00,0.00E-00,1.00E-15,1.00E-29,1.00E-29,0.00E-00,0.00E-00,  RATE0190
      60.00E-00,0.00E-00,0.00E-00,1.00E-30,1.00E-30,3.00E-12,3.00E-12,  RATE0195
      73.00E-12,1.00E-11,1.80E-10,1.00E-11,2.50E-10,1.00E-11,1.00E-28,  RATE0200
      81.00E-34,1.00E-17,1.00E-21,1.00E-21,2.00E-17,1.00E-24,6.40E-17,  RATE0205
      91.00E-24,1.00E-22,1.00E-22,5.00E-32,2.00E-31,5.00E-32,3.20E-35/  RATE0210
       DATA A3/ 2.60E-35,6.50E-34,2.00E-31,2.00E-33,3.00E-33,4.50E-33,  RATE0215
      13.00E-30,1.00E-33,1.00E-33,0.00E-00,1.10E-10,7.10E-17,3.00E-11,  RATE0220
      22.00E-10,5.00E-11,5.00E-10,3.00E-16,2.50E-11,2.00E-13,4.00E-12,  RATE0225
      32.00E-11,8.00E-13,5.00E-06,6.00E-08,3.00E-03,4.08E-07,5.58E-08,  RATE0230
      45.34E-03,1.00E-13,1.00E-13/                                     RATE0235
       DATA B/4*0.0,3*2.0,16*0.0,3*0.5,3*0.0,3*-1.0,5*-1.5,-0.7,3*0.0,16*RATE0240
      1-0.5,11*-1.5,55*0.0,-1.0,10*0.0,3*-0.5,3*0.0,-0.5,3*0.0,-1.0,4*0.0RATE0245
      2,1.5,4*0.0,1.5,13*0.0/                                          RATE0250
       DATA C/4*0.0,3*5.1E3,0.0,4.7E3,11*0.0,7.2E3,47*0.0,3E3,2*0.0,1E4,7RATE0255
      1*0.0,5E3,37*0.0,2E3,3E3,3*0.0,3.5E3,0.0,5E3,4E3,11*0.0,2*-9E2,2*0.RATE0260
      20,2E4,2*-9E2,0.0,3E3,1E4,0.0,3.75E4,1.9E4,5.3E2,1.4E4,1.35E4,2.8E3RATE0265
      3,6.6E3,0.0,7E3,2*0.0,1.2E3,8*0.0/                               RATE0270
C                                                                      RATE0275
C      INITIALIZATION OF SYSTEM AND INPUT.                             RATE0280
C                                                                      RATE0285
       REWIND 0                                                        RATE0290
```

```
      REWIND 1                                                          RATE0295
      REWIND 3                                                          RATE0300
      REWIND 8                                                          RATE0305
      CHI=1.0E-10                                                       RATE0310
      NOCOM=168                                                         RATE0315
      NUMB=15                                                           RATE0320
      ENDE=0.0                                                          RATE0325
      ITER=30                                                           RATE0330
      KB6=1                                                             RATE0335
    5 WRITE(6,400)                                                      RATE0340
      READ(5,445) (TITLE(N),N=1,12)                                     RATE0345
      WRITE(6,450) (TITLE(N),N=1,12)                                    RATE0350
      READ(5,395) EUBAR,ELBAR,DEL,ENDT                                  RATE0355
      READ(5,420) NOC                                                   RATE0360
      IF(NOC .EQ. 0) GO TO 30                                           RATE0365
      WRITE(6,490)                                                      RATE0370
      DO 25 J=1,NOC                                                     RATE0375
      READ(5,500) I,G(I),B(I),C(I),(ALF(N),N=1,7)                       RATE0380
      IF(I .GT. 69) GO TO 10                                            RATE0385
      A1(I)=G(I)                                                        RATE0390
      GO TO 20                                                          RATE0395
   10 IF(I .GT. 138) GO TO 15                                           RATE0400
      A2(I-69)=G(I)                                                     RATE0405
      GO TO 20                                                          RATE0410
   15 A3(I-138)=G(I)                                                    RATE0415
   20 WRITE(6,500) I,G(I),B(I),C(I),(ALF(N),N=1,7)                      RATE0420
   25 CONTINUE                                                          RATE0425
   30 READ(5,395) ALT,O,O2,ON2,T                                        RATE0430
      READ(5,395) (BEGIN(J),J=1,15),PROD                                RATE0435
      READ(5,415) KB1,KB2,KB3,KB4,IPLOT                                 RATE0440
      TIME=1.0E-6                                                       RATE0445
      MOUNT=0                                                           RATE0450
      KNT=0                                                             RATE0455
      JAKE=1                                                            RATE0460
      JACK=1                                                            RATE0465
      LAM=6                                                             RATE0470
      KLOT=0                                                            RATE0475
      DO 35 J=1,150                                                     RATE0480
   35 TREG(J)=0.0                                                       RATE0485
      LINT=((2*NUMB)+4)                                                 RATE0490
      I2NT=((3*NUMB)+4)                                                 RATE0495
      K2NT=((4*NUMB)+4)                                                 RATE0500
      J2NT=((5*NUMB)+4)                                                 RATE0505
      N2NT=((6*NUMB)+4)                                                 RATE0510
      DO 40 J=1,NUMB                                                    RATE0515
      TTREG(J+3)=0.0                                                    RATE0520
      LOCK(J)=0                                                         RATE0525
   40 KEY(J)=1                                                          RATE0530
      CALL INITAL                                                       RATE0535
C                                                                       RATE0540
C     COMPUTE RATE CONSTANTS IN THE FORM K=A*(T**B)*EXP(-C/T)           RATE0545
C                                                                       RATE0550
      K=1                                                               RATE0555
      DO 45 J=1,69                                                      RATE0560
      A(K)=A1(J)                                                        RATE0565
   45 K=K+1                                                             RATE0570
      DO 50 J=1,69                                                      RATE0575
      A(K)=A2(J)                                                        RATE0580
   50 K=K+1                                                             RATE0585
```

```
      DO 55 J=1,30                                                    RATE0590
      A(K)=A3(J)                                                      RATE0595
   55 K=K+1                                                           RATE0600
      DO 60 J=1,NOCOM                                                 RATE0605
   60 CON(J)=A(J)*(T**B(J))*EXP(-C(J)/T)                              RATE0610
                                                                      RATE0615
      PREPARE OUTPUT TAPES AND WRITE RATE CONSTANTS AND INITIAL       RATE0620
      CONDITIONS.                                                     RATE0625
                                                                      RATE0630
      WRITE(3,450) (TITLE(M),M=1,12)                                  RATE0635
      WRITE(6,405) ALT,D,T                                            RATE0640
      WRITE(6,480)                                                    RATE0645
      IS=1                                                            RATE0650
      DO 65 J=1,15                                                    RATE0655
      IT=IS+5                                                         RATE0660
      WRITE(6,455) (I,A(I),I=IS,IT)                                   RATE0665
      WRITE(6,485) (I,CON(I),I=IS,IT)                                 RATE0670
      IS=IS+6                                                         RATE0675
   65 CONTINUE                                                        RATE0680
      WRITE(6,515)                                                    RATE0685
      DO 70 J=1,13                                                    RATE0690
      IT=IS+5                                                         RATE0695
      WRITE(6,455) (I,A(I),I=IS,IT)                                   RATE0700
      WRITE(6,485) (I,CON(I),I=IS,IT)                                 RATE0705
      IS=IS+6                                                         RATE0710
   70 CONTINUE                                                        RATE0715
      WRITE(6,410)                                                    RATE0720
      IF(K81 .EQ. 0) GO TO 80                                         RATE0725
   75 WRITE(6,505) TREG(2),(TREG(J),J=4,12),PROD                      RATE0730
      WRITE(1) TREG(2),(TREG(J),J=13,18),DO2,DN2,(KEY(J),J=1,15)      RATE0735
      MOUNT=MOUNT+1                                                   RATE0740
      GO TO 135                                                       RATE0745
   80 M=NUMB+3                                                        RATE0750
      DO 95 J=2,M                                                     RATE0755
      DEC=TREG(J)                                                     RATE0760
      IF(DEC) 360,85,90                                               RATE0765
   85 DONT(J)=0.0                                                     RATE0770
      GO TO 95                                                        RATE0775
   90 DONT(J)=ALOG10(DEC)                                             RATE0780
   95 CONTINUE                                                        RATE0785
      IF(DO2 .EQ. 0.0) GO TO 100                                      RATE0790
      DDO2=ALOG10(DO2)                                                RATE0795
      GO TO 105                                                       RATE0800
  100 DDO2=0.0                                                        RATE0805
  105 IF(DN2 .EQ. 0.0) GO TO 110                                      RATE0810
      DDN2=ALOG10(DN2)                                                RATE0815
      GO TO 115                                                       RATE0820
  110 DDN2=0.0                                                        RATE0825
  115 IF(PROD) 360,120,125                                            RATE0830
  120 DPROD=0.0                                                       RATE0835
      GO TO 130                                                       RATE0840
  125 DPROD=ALOG10(PROD)                                              RATE0845
  130 WRITE(6,505) DONT(2),(DONT(J),J=4,12),DPROD                     RATE0850
      WRITE(1) DONT(2),(DONT(J),J=13,18),DDO2,DDN2,(KEY(J),J=1,15)    RATE0855
      MOUNT=MOUNT+1                                                   RATE0860
  135 DO 140 K=1,NUMB                                                 RATE0865
      IF(KEY(K)-2) 145,140,140                                        RATE0870
  140 CONTINUE                                                        RATE0875
      CALL ALGA                                                       RATE0880
```

```
      TREG(3)=2.0*TREG(3)                                              RATE0885
      TREG(2)=TREG(2)+TREG(3)                                          RATE0890
      GO TO 150                                                        RATE0895
C                                                                      RATE0900
C     INTEGRATION  OF EQUATIONS STARTS HERE.                           RATE0905
C                                                                      RATE0910
  145 CALL INTEG                                                       RATE0915
  150 CALL BALAN                                                       RATE0920
      IF(TREG(3) .NE. 2.0E-6) GO TO 155                                RATE0925
      KLOT=KLOT+1                                                      RATE0930
      IF(KLOT .GT. 10) GO TO 160                                       RATE0935
  155 KLOT=0                                                           RATE0940
      GO TO 165                                                        RATE0945
  160 PRINT 470                                                        RATE0950
      GO TO 365                                                        RATE0955
C                                                                      RATE0960
C     OUTPUT OF RESULTS STARTS HERE.                                   RATE0965
C                                                                      RATE0970
  165 IF(KNT .NE. 50) GO TO 170                                        RATE0975
      WRITE(6,410)                                                     RATE0980
      KNT=0                                                            RATE0985
  170 KNT=KNT+1                                                        RATE0990
      IF(KB1 .EQ. 0) GO TO 175                                         RATE0995
      WRITE(6,505) TREG(2),(TREG(J),J=4,12),TOTAL                      RATE1000
      WRITE(1) TREG(2),(TREG(J),J=13,18),DO2,DN2,(KEY(J),J=1,15)       RATE1005
      GO TO 210                                                        RATE1010
  175 K=NUMB+3                                                         RATE1015
      DO 190 J=2,K                                                     RATE1020
      DEC=TREG(J)                                                      RATE1025
      IF(DEC) 360,180,185                                              RATE1030
  180 DONT(J)=0.0                                                      RATE1035
      GO TO 190                                                        RATE1040
  185 DONT(J)=ALOG10(DEC)                                              RATE1045
  190 CONTINUE                                                         RATE1050
      DO2=ALOG10(DO2)                                                  RATE1055
      DN2=ALOG10(DN2)                                                  RATE1060
      IF(TOTAL) 360,195,200                                            RATE1065
  195 TOTL=0.0                                                         RATE1070
      GO TO 205                                                        RATE1075
  200 TOTL=ALOG10(TOTAL)                                               RATE1080
  205 WRITE(6,505) DONT(2),(DONT(J),J=4,12),TOTL                       RATE1085
      WRITE(1) DONT(2),(DONT(J),J=13,18),DO2,DN2,(KEY(J),J=1,15)       RATE1090
  210 MOUNT=MOUNT+1                                                    RATE1095
      CALL PLOT(IPLOT)                                                 RATE1100
      IF(JAKE-2) 215,350,355                                           RATE1105
  215 JACK=2                                                           RATE1110
      KIND=1                                                           RATE1115
      CALL SLOP(KIND)                                                  RATE1120
      KIND=2                                                           RATE1125
      CALL SLOP(KIND)                                                  RATE1130
      N2=LINT                                                          RATE1135
      DO 220 I=1,NUMB                                                  RATE1140
      CRITN(I)=2.0*ABS((TREG(N2)-TREG(I+3))/TREG(3))                   RATE1145
      CRTNO(I)=REMV(I)*TREG(I+3)                                       RATE1150
      N2=N2+1                                                          RATE1155
  220 CONTINUE                                                         RATE1160
      CRTNO(10)=CRTNO(10)+PNO                                          RATE1165
      CRTNO(15)=CRTNO(15)+PO                                           RATE1170
      DO 250 J=1,NUMB                                                  RATE1175
```

```
      IF(CRITN(J)-1.0E-2) 225,225,230                                    RATE1180
  225 IF(ABS(1.0-(FORM(J)/CRTNO(J)))-DEL) 245,245,240                    RATE1185
  230 IF(CRTNO(J)) 235,250,235                                           RATE1190
  235 IF((CRITN(J)/CRTNO(J))-DEL) 245,240,240                            RATE1195
  240 LOCK(J)=0                                                          RATE1200
      GO TO 250                                                          RATE1205
  245 LOCK(J)=LOCK(J)+1                                                  RATE1210
  250 CONTINUE                                                           RATE1215
  255 DO 275 J=1,NUMB                                                    RATE1220
      IF(KEY(J)-3) 260,275,275                                           RATE1225
  260 IF(LOCK(J)-3) 265,270,270                                          RATE1230
  265 KEY(J)=1                                                           RATE1235
      GO TO 275                                                          RATE1240
  270 KEY(J)=2                                                           RATE1245
  275 CONTINUE                                                           RATE1250
      DO 280 J=1,NUMB                                                    RATE1255
      IF(KEY(J) .GT. 2) GO TO 280                                        RATE1260
      IF(TREG(J+3) .GT. TTREG(J+3)) GO TO 280                            RATE1265
      IF(TREG(J+3) .GT. CHI) GO TO 280                                   RATE1270
      KEY(J)=3                                                           RATE1275
      TREG(J+3)=0.0                                                      RATE1280
  280 CONTINUE                                                           RATE1285
      IF(PNE) 300,285,300                                                RATE1290
  285 IF((TREG(4)/BEGIN(1))-1.0E-3) 290,300,300                          RATE1295
  290 DO 295 J=1,3                                                       RATE1300
      TREG(J+3)=0.0                                                      RATE1305
  295 KEY(J)=3                                                           RATE1310
  300 DO 305 J=1,NUMB                                                    RATE1315
  305 TTREG(J+3)=TREG(J+3)                                               RATE1320
      IF(KB2 .EQ. 1) GO TO 315                                           RATE1325
  310 CALL DAUXT                                                         RATE1330
C                                                                        RATE1335
C     DECISION TO CONTINUE INTEGRATION OR STOP IS MADE HERE.             RATE1340
C                                                                        RATE1345
  315 IF(TREG(4) .LT. ENDE) GO TO 370                                    RATE1350
      IF(TREG(2) .LT. ENDT) GO TO 320                                    RATE1355
      GO TO 370                                                          RATE1360
C                                                                        RATE1365
C  TEST SYSTEM CLOCK FOR IMMINENT TIMER OVERFLOW.                        RATE1370
C                                                                        RATE1375
  320 CALL     ICK                                                       RATE1380
      CAL     ET(4,KODOFX)                                               RATE1385
      GO T    5,325),KODOFX                                              RATE1390
  325 BIG= MA     TREG(4),TREG(5),TREG(6),TREG(7),TREG(8))               RATE1395
      DO 330 J=1,5                                                       RATE1400
      IF(BIG .EQ. TREG(J+3)) GO TO 335                                   RATE1405
  330 CONTINUE                                                           RATE1410
  335 IF(J-LAM) 340,345,340                                              RATE1415
  340 KEY(LAM)=1                                                         RATE1420
      JACK=1                                                             RATE1425
  345 LAM=J                                                              RATE1430
      KEY(LAM)=4                                                         RATE1435
      GO TO 135                                                          RATE1440
C                                                                        RATE1445
C     ERROR COMMENT OUTPUTS.                                             RATE1450
C                                                                        RATE1455
  350 WRITE(6,430) TREG(2)                                               RATE1460
      GO TO 365                                                          RATE1465
  355 WRITE(6,435) TREG(2)                                               RATE1470
```

```
      GO TO 365                                                          RATE1475
  360 WRITE(6,425)                                                       RATE1480
      K=NUMB+3                                                           RATE1485
      WRITE(6,440) (TREG(J),J=2,K),TOTAL                                 RATE1490
  365 IF(KB4 .EQ. 1) GO TO 370                                           RATE1495
      KB6=2                                                              RATE1500
C                                                                        RATE1505
C     TRANSFER ALL RESULTS TO OUTPUT TAPE HERE.                          RATE1510
C                                                                        RATE1515
  370 REWIND 1                                                           RATE1520
      END FILE 3                                                         RATE1525
      IF(IPLOT .NE. 1) GO TO 380                                         RATE1530
      END FILE 0                                                         RATE1535
      REWIND 8                                                           RATE1540
      DO 375 K=1,ITEM                                                    RATE1545
      READ(8)SIGM(1),IQ(1),(SIGM(J),IQ(J),J=10,17)                       RATE1550
      WRITE(0,520)SIGM(1),IQ(1),(SIGM(J),IQ(J),J=10,17)                  RATE1555
  375 CONTINUE                                                           RATE1560
      END FILE 0                                                         RATE1565
      REWIND 8                                                           RATE1570
      ITEM=0                                                             RATE1575
  380 KNT=0                                                              RATE1580
      WRITE(6,475)                                                       RATE1585
      DO 385 K=1,MOUNT                                                   RATE1590
      READ(1) TREG(2),(TREG(J),J=13,18),DO2,DN2,(KEY(J),J=1,15)          RATE1595
      WRITE(6,510) TREG(2),(TREG(J),J=13,18),DO2,DN2,(KEY(J),J=1,15)     RATE1600
      KNT=KNT+1                                                          RATE1605
      IF(KNT .LT. 50) GO TO 385                                          RATE1610
      WRITE(6,475)                                                       RATE1615
      KNT=0                                                              RATE1620
  385 CONTINUE                                                           RATE1625
      REWIND 1                                                           RATE1630
      GO TO (5,390),KB6                                                  RATE1635
  390 REWIND 3                                                           RATE1640
      CALL EXIT                                                          RATE1645
C                                                                        RATE1650
  395 FORMAT (1P6E12.5)                                                  RATE1655
  400 FORMAT(76H1SOLUTION OF THE REACTION RATE EQUATIONS WITH 15 SPECIESRATE1660
     1 AND 168 REACTIONS. )                                              RATE1665
  405 FORMAT(11H1ALTITUDE =1PE11.4,4H CM.,16H TOTAL DENSITY =1PE12.5,14HRATE1670
     1 TEMPERATURE =0PF7.2)                                              RATE1675
  410 FORMAT(122H1 TIME(SEC)   N(E)/CC    N(O-)/CC  N(O2-)/CC  N(O3-)/CCRATE1680
     1 N(NO2-)/CC   N(O+)/CC  N(O2+)/CC  N(N2+)/CC  N(NO+)/CC  PRODUCTIORATE1685
     2N)                                                                 RATE1690
  415 FORMAT(7I2)                                                        RATE1695
  420 FORMAT(I4)                                                         RATE1700
  425 FORMAT(65HOTHE PROGRAM IN TRYING TO GENERATE THE LOG OF A NEGATIVERATE1705
     1 NUMBER. )                                                         RATE1710
  430 FORMAT(45HOTHE INTERGRATING MESH IS VANISHING IN INT AT 1PE11.5,6HRATE1715
     1 SEC. )                                                            RATE1720
  435 FORMAT(47HOTHE INTERGRAYING MESH IS VANISHING IN ALGA AT 1PE11.5,6RATE1725
     1H SEC. )                                                           RATE1730
  440 FORMAT(1P10E10.2)                                                  RATE1735
  445 FORMAT(12A6)                                                       RATE1740
  450 FORMAT(1HO,12A6)                                                   RATE1745
  455 FORMAT(1HO,(6(2X4H  A(,I3,2H)=1PE10.3)))                           RATE1750
  460 FORMAT(1HO,(4(2X4H  A(,I3,2H)=1PE10.3)))                           RATE1755
  465 FORMAT(1H ,(4(2X4HCON(,I3,2H)=1PE10.3)))                           RATE1760
  470 FORMAT(43H THE INCREMENT IS CONSTANT AT 1.0E-06 SEC. )             RATE1765
```

```
475 FORMAT(121H1 TIME(SEC)   N(NO)/CC   N(N)/CC   N(NO2)/CC   N(O3)/CCRATE1770
   1  N(N2O)/CC   N(O)/CC   N(O2)/CC   N(N2)/CC        KEYS 1 THRU 15RATE1775
   2)                                                                RATE1780
480 FORMAT(22HOREACTION COEFFICIENTS)                                RATE1785
485 FORMAT(1H ,(6(2X4HCON(,I3,2H)=1PE10.3)))                         RATE1790
490 FORMAT(1HO,20X,86HTHE FOLLOWING IS A LIST OF RATE CONSTANT CHANGESRATE1795
   1 FROM THE STND LIST USED IN THIS RUN. )                          RATE1800
495 FORMAT(102H1  TIME (SEC)  O2 DENSITY   N2 DENSITY     E  O- O2- ORATE1805
   12- NO2- O+ O2+ N2+ NO  N  NO2  O3  NO+ N2O  O  )                 RATE1810
500 FORMAT(I4,1PE10.2,0PF5.1,1PE10.2,7A6)                            RATE1815
505 FORMAT(1P11E11.3)                                                RATE1820
510 FORMAT(1P9E11.3,5I2,1X,4I2,1X,6I2)                               RATE1825
515 FORMAT(1H1)                                                      RATE1830
520 FORMAT(6X11(A1,I5))                                              RATE1835
    END                                                              RATE1840
```

$$N_{j4} = N_{jO} + \frac{1}{2}\,\Delta t\, F_j(t_O, N_{10}, N_{20}, N_{30} \cdots N_{mO})$$
$$- \frac{3}{2}\,\Delta t\, F_j(t_O + \frac{1}{3}\,\Delta t, N_{12}, N_{22}, N_{32} \cdots N_{m2})$$
$$+ 2\Delta t\, F_j(t_O + \frac{1}{2}\,\Delta t, N_{13}, N_{23}, N_{33} \cdots N_{m3})$$

$$N_{j5} = N_{jO} + \frac{1}{6}\,\Delta t\, F_j(t_O, N_{10}, N_{20}, N_{30} \cdots N_{mO})$$
$$+ \frac{2}{3}\,\Delta t\, F_j(t_O + \frac{1}{2}\,\Delta t, N_{13}, N_{23}, N_{33} \cdots N_{m3})$$
$$+ \frac{1}{6}\,\Delta t\, F_j(t + \Delta t, N_{14}, N_{24}, N_{34} \cdots N_{m4})\ .$$

Merson (1957) has shown that the error in N_{j4} is $-\frac{1}{120}\,\Delta t^5 N_j^{(V)}$ and that the error in N_{j5} is $-\frac{1}{720}\,\Delta t^5 N_j^{(V)}$. Therefore, a good estimate of the error in N_{j5} which are the accepted values of N_j at the end of the increment is

$$\frac{1}{5}(N_{j4} - N_{j5})\ .$$

The criterion for the acceptance of a solution is that one-fifth of the relative difference between N_{j4} and N_{j5} must be less than ϵ for every N_j. That is

$$\frac{\left|\frac{N_{j4}}{N_{j5}} - 1\right|}{5} < \epsilon\ .$$

Should this criterion fail for any N_j, the program cuts the increment in half, re-initializes itself, and computes a solution over a smaller increment. It uses this

factor of two cutback for the first three failures. If the solution is still unable to satisfy the criterion after these three tries, the increment is cut by a factor of ten and a new solution is computed. The program then tries to obtain an acceptable solution three times using decade cutbacks. Should the solution still be unacceptable after these three attempts, all the differential equations are returned to the set, the increment is cut another decade, and a new solution is computed. Should the solution still be unacceptable, decade cutbacks are continued until an acceptable solution is obtained or until the increment is reduced below 10^{-8}. If no solution is acceptable by the time the increment is reduced to 10^{-8}, the integration is stopped, the JAKE switch is set to two, and control is transferred to the main program.

This program sets two switches that are used externally to determine conditions under which the solution was generated.

1) IFAIL - INTEG will return with this switch on zero if the solution was generated over the first increment used. If the subprogram had to reduce the increment, this switch is set to one. This information is required if the main program is looking for solutions at specific times.

2) JAKE - this switch is set to its normal setting of one by the main program. As long as a valid solution is generated, its setting remains on one. If the integrator is unable for any reason to develop a solution with any increment greater than 10^{-8}, this switch is set to two and control is transferred to the main program where the comment THE INTEGRATING MESH IS VANISHING IN INT AT X.XX SEC is written.

This subprogram calls ALGA for a solution to the exponential equations after the computation of each of the five sets of N_j. In this way, the program always uses a consistent set of N_j's every time a derivative computation is called for.

If ALGA returns to INTEG with the JAM switch on two indicating that ALGA could not solve the exponential set of equations, INTEG makes one of two decisions to remedy the difficulty.

1) If the concentration of any species is being computed from its exponential equation over the current increment and was computed from its differential over the proceeding increment, its differential equation is returned to the differential set and its exponential equation is removed from the exponential set. The system is reinitialized and the integration is started again over the current increment.

2) If condition i does not prevail, all of the differential equations are returned to the differential set, the exponential set is emptied, the increment is reduced by a factor of ten, the system is reinitialized, and the integration is performed over the smaller increment.

A COMMON subscripted variable TREG is required by the subroutine. Its dimension is (6*NUMB+4) where NUMB is the number of differential equations in the full set. After each integration, the locations of TREG contain the following values:

TREG(2) = Independent variable t(sec)
TREG(3) = Integrating increment
TREG(4) - TREG(NUMB+3) = Dependent variables N_{j5} .

The increment for the next integration is automatically set to twice the value used for the integration just completed. If the new increment is too large it will be cut back by the program during the next integration.

The following statements, beginning on page 81, are a listing of this subroutine.

7.1.3 THE ITERATION SUBROUTINE ALGA

This routine solves the simultaneous set of exponential equations. It is called by subroutine INTEG except when the set of differential equations is empty when it is called by the main program. The set of exponential equations (see Eq. 3) is solved by the method of successive substitutions using N_o as the initial guess and computing successive guesses from

$$N_{j+1} = \frac{N_j + N_{j-1}}{2} \qquad j = 0,1,2,3,\ldots ITER \ .$$

The solution is said to converge if

$$(1-\delta) \leq \frac{N_{j-1}}{N_j} \leq (1+\delta) \tag{45}$$

where δ is some number less than one. Choice of the value of δ is determined by the amount of charge imbalance that can be tolerated in the solution. Generally, the problem of charge imbalance can be neglected if δ is less than 0.001.

If the criterion (Eq. 45) cannot be satisfied within ITER iterations, ALGA makes one of two decisions.

1) If any differential equation is still in the differential set the JAM switch is set to two and control is returned to INTEG (see INTEG for remedy taken).

```
$IBFTC INTEG   LIST                                                    INT00000
      SUBROUTINE INTEG                                                 INT00005
C     INTEGRATOR USING THE KUTTA MERSON TECHNIQUE.                     INT00010
      COMMON TREG(150),KEY(15),FORM(15),REMV(15),CON(168),LKEY(15),BEGININT00015
     1(15)                                                             INT00020
      COMMON NUMB,EUBAR,ELBAR,D,DO2,DN2,T,PNE,PNO,PO2,PO,PN2,TOTAL,JACK,INT00025
     1JAKE,JAM,ITER,LINT,I2NT,K2NT,J2NT,N2NT,FIRST,PROD,LAM,TIME,KB3,TOTINT00030
     20,TOTN                                                           INT00035
      DIMENSION C(7)                                                   INT00040
      DATA C/0.33333333,0.16666666,0.125,0.375,0.5,1.5,0.66666666/     INT00045
      N1=NUMB+4                                                        INT00050
      N2=LINT                                                          INT00055
      N3=I2NT                                                          INT00060
      N4=K2NT                                                          INT00065
      N5=J2NT                                                          INT00070
      N6=N2NT                                                          INT00075
      KOUNT=0                                                          INT00080
      DO 5 J=1,NUMB                                                    INT00085
      TREG(N2)=TREG(J+3)                                               INT00090
    5 N2=N2+1                                                          INT00095
      N2=LINT                                                          INT00100
      TREG(1)=TREG(2)                                                  INT00105
   10 DO 90 I=1,5                                                      INT00110
      CALL DAUX                                                        INT00115
      DO 50 J=1,NUMB                                                   INT00120
      IF(KEY(J)-1) 45,15,45                                            INT00125
   15 GO TO (20,25,30,35,40),I                                         INT00130
   20 TREG(N3)=TREG(N1)*TREG(3)                                        INT00135
      TREG(J+3)=TREG(N2)+C(1)*TREG(N3)                                 INT00140
      GO TO 45                                                         INT00145
   25 TREG(J+3)=TREG(N2)+C(2)*TREG(N3)+C(2)*TREG(3)*TREG(N1)           INT00150
      GO TO 45                                                         INT00155
   30 TREG(N4)=TREG(N1)*TREG(3)                                        INT00160
      TREG(J+3)=TREG(N2)+C(3)*TREG(N3)+C(4)*TREG(N4)                   INT00165
      GO TO 45                                                         INT00170
   35 TREG(N5)=TREG(N1)*TREG(3)                                        INT00175
      TREG(J+3)=TREG(N2)+C(5)*TREG(N3)-C(6)*TREG(N4)+2.0*TREG(N5)      INT00180
      TREG(N6)=TREG(J+3)                                               INT00185
      GO TO 45                                                         INT00190
   40 TREG(J+3)=TREG(N2)+C(2)*TREG(N3)+C(7)*TREG(N5)+C(2)*TREG(3)*TREG(NINT00195
     11)                                                               INT00200
   45 N1=N1+1                                                          INT00205
      N2=N2+1                                                          INT00210
      N3=N3+1                                                          INT00215
      N4=N4+1                                                          INT00220
      N5=N5+1                                                          INT00225
      N6=N6+1                                                          INT00230
   50 CONTINUE                                                         INT00235
      N1=NUMB+4                                                        INT00240
      N2=LINT                                                          INT00245
      N3=I2NT                                                          INT00250
      N4=K2NT                                                          INT00255
      N5=J2NT                                                          INT00260
      N6=N2NT                                                          INT00265
      DO 55 K=1,NUMB                                                   INT00270
      IF(KEY(K)-2) 55,60,55                                            INT00275
   55 CONTINUE                                                         INT00280
      GO TO 70                                                         INT00285
   60 CALL ALGA                                                        INT00290
```

```
      L=JAKE                                                            INT00295
      GO TO (65,65,195),L                                               INT00300
   65 L=JAM                                                             INT00305
      GO TO (70,110),L                                                  INT00310
   70 GO TO (75,90,80,85,90),I                                          INT00315
   75 TREG(2)=TREG(1)+C(1)*TREG(3)                                      INT00320
      GO TO 90                                                          INT00325
   80 TREG(2)=TREG(1)+C(5)*TREG(3)                                      INT00330
      GO TO 90                                                          INT00335
   85 TREG(2)=TREG(1)+TREG(3)                                           INT00340
   90 CONTINUE                                                          INT00345
      DO 105 J=1,NUMB                                                   INT00350
      IF(KEY(J)-1) 100,95,100                                           INT00355
   95 IF((ABS(1.0-(TREG(N6)/TREG(J+3)))/5.0) .GT. EUBAR) GO TO 135      INT00360
  100 N6=N6+1                                                           INT00365
  105 CONTINUE                                                          INT00370
      N6=N2NT                                                           INT00375
      GO TO 185                                                         INT00380
  110 DO 120 J=1,NUMB                                                   INT00385
      IF(KEY(J)-2) 120,115,120                                          INT00390
  115 IF(LKEY(J)-1) 120,125,120                                         INT00395
  120 CONTINUE                                                          INT00400
      GO TO 165                                                         INT00405
  125 KEY(J)=1                                                          INT00410
      DO 130 J=1,NUMB                                                   INT00415
      TREG(J+3)=TREG(N2)                                                INT00420
  130 N2=N2+1                                                           INT00425
      GO TO 155                                                         INT00430
  135 KOUNT=KOUNT+1                                                     INT00435
      IF(KOUNT .GT. 3) GO TO 160                                        INT00440
      TREG(3)=TREG(3)/2.0                                               INT00445
  140 DO 145 J=1,NUMB                                                   INT00450
      TREG(J+3)=TREG(N2)                                                INT00455
  145 N2=N2+1                                                           INT00460
      GO TO (155,150),JACK                                              INT00465
  150 IF(TREG(3) .LT. 1.0E-8) GO TO 190                                 INT00470
  155 TREG(2)=TREG(1)                                                   INT00475
      N2=LINT                                                           INT00480
      N6=N2NT                                                           INT00485
      GO TO 10                                                          INT00490
  160 IF(KOUNT .LT. 6) GO TO 180                                        INT00495
  165 DO 175 J=1,NUMB                                                   INT00500
      IF(KEY(J)-2) 175,170,175                                          INT00505
  170 KEY(J)=1                                                          INT00510
  175 CONTINUE                                                          INT00515
  180 TREG(3)=TREG(3)/10.0                                              INT00520
      GO TO 140                                                         INT00525
  185 TREG(3)=2.0*TREG(3)                                               INT00530
      GO TO 195                                                         INT00535
  190 JAKE=2                                                            INT00540
  195 RETURN                                                            INT00545
      END                                                               INT00550
```

2) If the differential equation set is empty the integrating mesh is halved, the initial conditions are restored, and the iteration procedure is repeated.

Should ALGA be unable to compute a solution to the full set of exponential equations before the increment is reduced to the starting increment (usually 10^{-6} secs), the JAKE switch is set to three and control is returned to the main program where the comment THE INTEGRATING MESH IS VANISHING IN ALGA AT X.XX SEC is written.

The following statements, beginning on page 84, are a listing of subroutine ALGA.

7.1.4 SUBROUTINE TO COMPUTE THE FORM AND REMV TERMS, SLOP

This subroutine computes the $\sum F_i$ and $\sum R_i$ terms required in computing the derivatives. This has always been a tedious program to write because it is here that the sum of all the rates at which the species are formed and removed is computed. For a large number of reactions and species the effort to hand code this program is tremendous.

A program was written by David McIntyre (1965) for an IBM-6000 computer which, upon being fed in coded form the reactions and the species, writes a program for computing the total and the partial derivatives of each species. This code was rewritten by the author for use on an IBM-7044-7094 computer to compute only the formation and the removal equations required for each species. This program in its latest version is described in Appendix B.

Since the $\sum F_i$ and $\sum R_i$ terms are required by both INTEG and ALGA, SLOP is written in such a way that it can be called by either of these subroutines. In order to conserve computer time, SLOP is so written that only the FORM(J) and the REMV(J) terms for the species that are not in equilibrium are computed when SLOP is called by INTEG(KIND=1) and only the FORM(J) and the REMV(J) terms for the species that are in equilibrium are computed when SLOP is called by ALGA (KIND=2). This eliminates the waste of time in computing derivatives that will not be used. In addition, SLOP calls PRODUC for the rate of production of ionization by the external source only when SLOP is called by INTEG.

Subroutine SLOP is now written by the computer. The following statements, beginning on page 85, are written for 15 species and 168 reactions.

```
$IBFTC ALGA    LIST                                                          ALGA0000
       SUBROUTINE ALGA                                                       ALGA0005
       COMMON TREG(150),KEY(15),FORM(15),REMV(15),CON(168),LKEY(15),BEGINALGA0010
      1(15)                                                                  ALGA0015
       COMMON NUMB,EUBAR,ELBAR,D,DO2,DN2,T,PNE,PNO,PO2,PO,PN2,TOTAL,JACK,ALGA0020
      1JAKE,JAM,ITER,LINT,I2NT,K2NT,J2NT,N2NT,FIRST,PROD,LAM,TIME,KB3,TOTALGA0025
      20,TOTN                                                                ALGA0030
       DIMENSION Y(17),RAT(17),ZINCH(17)                                     ALGA0035
       DO 5 J=1,NUMB                                                         ALGA0040
       Y(J)=TREG(J+3)                                                        ALGA0045
       ZINCH(J)=TREG(J+3)                                                    ALGA0050
       RAT(J)=0.0                                                            ALGA0055
     5 CONTINUE                                                              ALGA0060
    10 KOUNT=0                                                               ALGA0065
    15 KIND=2                                                                ALGA0070
       CALL SLOP(KIND)                                                       ALGA0075
       FORM(10)=FORM(10)-PNO                                                 ALGA0080
       FORM(15)=FORM(15)-PO                                                  ALGA0085
       DO 30 J=1,NUMB                                                        ALGA0090
       IF(KEY(J)-2) 30,20,30                                                 ALGA0095
    20 YEJK=FORM(J)/REMV(J)                                                  ALGA0100
       YIPES=ABS(REMV(J)*TREG(3))                                            ALGA0105
       IF(YIPES .GT. 30.0) GO TO 25                                          ALGA0110
       Y(J)=(ZINCH(J)-YEJK)*EXP(-YIPES)+YEJK                                 ALGA0115
       GO TO 30                                                              ALGA0120
    25 Y(J)=YEJK                                                             ALGA0125
    30 CONTINUE                                                              ALGA0130
    35 DO 45 J=1,NUMB                                                        ALGA0135
       IF(KEY(J)-2) 45,40,45                                                 ALGA0140
    40 RAT(J)=(TREG(J+3)/Y(J))                                               ALGA0145
    45 CONTINUE                                                              ALGA0150
       DO 60 J=1,NUMB                                                        ALGA0155
       IF(KEY(J)-2) 60,50,60                                                 ALGA0160
    50 IF(RAT(J)-(1.0+ELBAR)) 55,65,65                                       ALGA0165
    55 IF(RAT(J)-(1.0-ELBAR)) 65,60,60                                       ALGA0170
    60 CONTINUE                                                              ALGA0175
       GO TO 110                                                             ALGA0180
    65 DO 75 J=1,NUMB                                                        ALGA0185
       IF(KEY(J)-2) 75,70,75                                                 ALGA0190
    70 TREG(J+3)=(TREG(J+3)+Y(J))/2.0                                        ALGA0195
    75 CONTINUE                                                              ALGA0200
       KOUNT=KOUNT+1                                                         ALGA0205
       IF(KOUNT-ITER) 15,80,80                                               ALGA0210
    80 DO 85 J=1,NUMB                                                        ALGA0215
       IF(KEY(J)-1) 85,105,85                                                ALGA0220
    85 CONTINUE                                                              ALGA0225
       TREG(3)=TREG(3)/2.0                                                   ALGA0230
       TREG(2)=TREG(2)-TREG(3)                                               ALGA0235
       IF(FIRST-TREG(3)) 95,90,90                                            ALGA0240
    90 JAKE=3                                                                ALGA0245
       GO TO 125                                                             ALGA0250
    95 L=LINT                                                                ALGA0255
       DO 100 J=1,NUMB                                                       ALGA0260
       TREG(J+3)=TREG(L)                                                     ALGA0265
       L=L+1                                                                 ALGA0270
   100 CONTINUE                                                              ALGA0275
       GO TO 10                                                              ALGA0280
   105 JAM=2                                                                 ALGA0285
       GO TO 125                                                             ALGA0290
   110 DO 120 J=1,NUMB                                                       ALGA0295
       IF(KEY(J)-2) 120,115,120                                              ALGA0300
   115 TREG(J+3)=(TREG(J+3)+Y(J))/2.0                                        ALGA0305
   120 CONTINUE                                                              ALGA0310
       JAM=1                                                                 ALGA0315
   125 RETURN                                                                ALGA0320
       END                                                                   ALGA0325
```

```
$IBFTC SLOP    LIST                                                    SLOP0000
      SUBROUTINE SLOP(KIND)                                            SLOP0005
      COMMON TREG(150),KEY(15),FORM(15),REMV(15),CON(168),LKEY(15),BEGINSLOP0010
     1(15)                                                             SLOP0015
      COMMON NUMB,EUBAR,ELBAR,D,DO2,DN2,T,PNE,PNO,PO2,PO,PN2,TOTAL,JACK,SLOP0020
     1JAKE,JAM,ITER,LINT,I2NT,K2NT,J2NT,N2NT,FIRST,PROD,LAM,TIME,KB3,TOTSLOP0025
     20,TOTN                                                           SLOP0030
      DIMENSION Y(150),C(168)                                          SLOP0035
      EQUIVALENCE (Y(1),TREG(4)),(C,CON)                               SLOP0040
      CALL BALAN                                                       SLOP0045
      Y(16)=DO2                                                        SLOP0050
      Y(17)=DN2                                                        SLOP0055
      Y(18)=D                                                          SLOP0060
      IF(KIND .EQ. 2) GO TO 5                                          SLOP0065
      CALL PRODUC                                                      SLOP0070
    5 IF((KIND .EQ. 1 .AND. KEY( 1) .NE. 1) .OR. (KIND .EQ. 2 .AND. KEY(SLOP0075
     1 1) .NE. 2)) GO TO   10                                          SLOP0080
      REMV( 1)     =(Y(9)*(+C(32)+C(36)*Y(18)+C(37)*Y(18)+C(41))+Y(15)*(+SLOP0085
     1C(16)+C(22)*Y(16)+C(23)*Y(17)+C(26)*Y(16))+Y(7)*(+C(30)+C(34)*Y(18SLOP0090
     1)+C(39))+Y(8)*(+C(31)+C(35)*Y(18)+C(40))+Y(12)*(+C(18)+C(27)*Y(16)SLOP0095
     1+C(28)*Y(17))+Y(13)*(+C(19)+C(20)+C(21))+Y(6)*(+C(33)*Y(18)+C(38)SLOP0100
     1+C(17)*Y(16)+C(24)*Y(16)*Y(16)+C(25)*Y(16)*Y(17)+C(29)*Y(10)*Y(18)SLOP0105
     1)                                                                SLOP0110
      FORM( 1)     =(Y(2)*(+C(2)+C(8)*Y(15)+C(9)*Y(16)+C(10)*Y(11)+C(11)*SLOP0115
     1Y(17)+C(12)*Y(10)+C(13)*Y(13))+Y(3)*(Y(15)*(+C(7)+C(15))+C(1)+C(6)SLOP0120
     1*Y(17)+C(14)*Y(11)+C(5)*Y(16))+Y(4)*(+C(4)+C(167)*Y(15))+Y(5)*(+C(SLOP0125
     13)+C(168)*Y(15))/)+PNE                                           SLOP0130
   10 IF((KIND .EQ. 1 .AND. KEY( 2) .NE. 1) .OR. (KIND .EQ. 2 .AND. KEY(SLOP0135
     1 2) .NE. 2)) GO TO   15                                          SLOP0140
      REMV( 2)     =(Y(6)*(+C(42)+C(58)*Y(11)+C(59)*Y(16)+C(60)*Y(17)+C(6SLOP0145
     11)*Y(15))+Y(7)*(+C(46)+C(63)*Y(18))+Y(8)*(+C(50)+C(55)*Y(18))+Y(9)SLOP0150
     1*(+C(54)+C(67)*Y(18))+Y(10)*(+C(12)+C(114)*Y(18))+Y(13)*(+C(13)+C(SLOP0155
     193))+C(2)+C(8)*Y(15)+C(9)*Y(16)+C(10)*Y(11)+C(11)*Y(17)+C(92)*Y(12SLOP0160
     1)+C(106)*Y(16)+C(124)*Y(16)*Y(16))                               SLOP0165
      FORM( 2)     =(Y(1)*(Y(15)*(+C(16)+C(22)*Y(16)+C(23)*Y(17))+C(20)*YSLOP0170
     1(13))+C(94)*Y(3)*Y(15))                                          SLOP0175
   15 IF((KIND .EQ. 1 .AND. KEY( 3) .NE. 1) .OR. (KIND .EQ. 2 .AND. KEY(SLOP0180
     1 3) .NE. 2)) GO TO   20                                          SLOP0185
      REMV( 3)     =(Y(8)*(+C(51)+C(66)*Y(18)+C(69)+C(70)+C(71)+C(72))+Y(SLOP0190
     19)*(+C(55)+C(68)*Y(18)+C(75)+C(76)+C(77))+Y(15)*(+C(7)+C(15)+C(94)SLOP0195
     1)+Y(6)*(+C(43)+C(62)*Y(18))+Y(7)*(+C(47)+C(64)*Y(18))+Y(11)*(+C(14SLOP0200
     1)+C(115)*Y(18))+C(1)+C(5)*Y(16)+C(6)*Y(17)+C(95)*Y(12)+C(96)*Y(13)SLOP0205
     1+C(125)*Y(16)*Y(17))                                             SLOP0210
      FORM( 3)     =(Y(1)*(+C(17)*Y(16)+C(21)*Y(13)+C(24)*Y(16)*Y(16)+C(2SLOP0215
     15)*Y(16)*Y(17)+C(26)*Y(16)*Y(15)))                               SLOP0220
   20 IF((KIND .EQ. 1 .AND. KEY( 4) .NE. 1) .OR. (KIND .EQ. 2 .AND. KEY(SLOP0225
     1 4) .NE. 2)) GO TO   25                                          SLOP0230
      REMV( 4)     =(Y(9)*(+C(57)+C(81)+C(82))+Y(8)*(+C(53)+C(74))+C(4)+CSLOP0235
     1(45)*Y(6)+C(49)*Y(7)+C(98)*Y(12)+C(126)*Y(17)+C(167)*Y(15))      SLOP0240
      FORM( 4)     =(Y(13)*(+C(19)*Y(1)+C(93)*Y(2)+C(96)*Y(3)+C(97)*Y(5))SLOP0245
     1+Y(2)*(+C(106)*Y(16)+C(124)*Y(16)*Y(16)))                        SLOP0250
   25 IF((KIND .EQ. 1 .AND. KEY( 5) .NE. 1) .OR. (KIND .EQ. 2 .AND. KEY(SLOP0255
     1 5) .NE. 2)) GO TO   30                                          SLOP0260
      REMV( 5)     =(Y(9)*(+C(56)+C(78)+C(79)+C(80))+Y(8)*(+C(52)+C(73))+SLOP0265
     1C(3)+C(44)*Y(6)+C(48)*Y(7)+C(97)*Y(13)+C(168)*Y(15))             SLOP0270
      FORM( 5)     =(Y(12)*(Y(1)*(+C(18)+C(27)*Y(16)+C(28)*Y(17))+C(92)*YSLOP0275
     1(2)+C(95)*Y(3)+C(98)*Y(4))+Y(3)*(+C(115)*Y(11)*Y(18)+C(125)*Y(16)*SLOP0280
     1Y(17))+C(114)*Y(2)*Y(10)*Y(18)+C(126)*Y(4)*Y(17))                SLOP0285
   30 IF((KIND .EQ. 1 .AND. KEY( 6) .NE. 1) .OR. (KIND .EQ. 2 .AND. KEY(SLOP0290
```

```
  1 6) .NE. 2)) GO TO   35                                                SLOP0295
   REMV( 6)     =(Y(2)*(+C(42)+C(58)*Y(11)+C(59)*Y(16)+C(60)*Y(17)+C(6SLOP0300
  11)*Y(15))+Y(10)*(+C(84)+C(102)+C(110)*Y(18)+C(117))+Y(1)*(+C(33)*YSLOP0305
  1(18)+C(38))+Y(3)*(+C(43)+C(62)*Y(18))+Y(11)*(+C(100)+C(108)*Y(18))SLOP0310
  1+Y(15)*(+C(99)+C(107)*Y(18))+C(83)*Y(16)+C(85)*Y(12)+C(86)*Y(14)+CSLOP0315
  1(101)*Y(17)+C(109)*Y(17)*Y(18)+C(116)*Y(17)+C(44)*Y(5)+C(45)*Y(4))SLOP0320
   FORM( 6)     =(+C(118)*Y(7)*Y(11)+C(88)*Y(8)*Y(15))+PO                 SLOP0325
35 IF((KIND .EQ. 1 .AND. KEY( 7) .NE. 1) .OR. (KIND .EQ. 2 .AND. KEY(SLOP0330
  1 7) .NE. 2)) GO TO   40                                                SLOP0335
   REMV( 7)     =(Y(1)*(+C(30)+C(34)*Y(18)+C(39))+Y(2)*(+C(46)+C(63)*YSLOP0340
  1(18))+Y(3)*(+C(47)+C(64)*Y(18))+Y(11)*(+C(118)+C(120))+Y(15)*(+C(1SLOP0345
  103)+C(111)*Y(18))+C(87)*Y(10)+C(119)*Y(17)+C(121)*Y(12)+C(48)*Y(5)SLOP0350
  1+C(49)*Y(4))                                                           SLOP0355
   FORM( 7)     =(Y(6)*(Y(15)*(+C(99)+C(107)*Y(18))+C(117)*Y(10)+C(83)SLOP0360
  1*Y(16))+C(89)*Y(8)*Y(16))+PO2                                          SLOP0365
40 IF((KIND .EQ. 1 .AND. KEY( 8) .NE. 1) .OR. (KIND .EQ. 2 .AND. KEY(SLOP0370
  1 8) .NE. 2)) GO TO   45                                                SLOP0375
   REMV( 8)     =(Y(3)*(+C(51)+C(66)*Y(18)+C(69)+C(70)+C(71)+C(72))+Y(SLOP0380
  11)*(+C(31)+C(35)*Y(18)+C(40))+Y(2)*(+C(50)+C(65)*Y(18))+Y(4)*(+C(5SLOP0385
  13)+C(74))+Y(5)*(+C(52)+C(73))+Y(15)*(+C(88)+C(122))+C(89)*Y(16)+C(SLOP0390
  190)*Y(11)+C(91)*Y(10)+C(104)*Y(17)+C(123)*Y(16))                       SLOP0395
   FORM( 8)     =+PN2                                                     SLOP0400
45 IF((KIND .EQ. 1 .AND. KEY( 9) .NE. 1) .OR. (KIND .EQ. 2 .AND. KEY(SLOP0405
  1 9) .NE. 2)) GO TO   50                                                SLOP0410
   REMV( 9)     =(Y(3)*(+C(55)+C(68)*Y(18)+C(75)+C(76)+C(77))+Y(1)*(+CSLOP0415
  1(32)+C(36)*Y(18)+C(37)*Y(18)+C(41))+Y(5)*(+C(56)+C(78)+C(79)+C(80)SLOP0420
  11)+Y(4)*(+C(57)+C(81)+C(82))+Y(2)*(+C(54)+C(67)*Y(18))+Y(15)*(+C(10SLOP0425
  15)+C(112)*Y(18))+C(113)*Y(11)*Y(18))                                   SLOP0430
   FORM( 9)     =(Y(6)*(Y(11)*(+C(100)+C(108)*Y(18))+C(116)*Y(17)+C(84SLOP0435
  1)+Y(10))+Y(7)*(+C(119)*Y(17)+C(120)*Y(11)+C(121)*Y(12)+C(87)*Y(10)SLOP0440
  1)+Y(8)*(+C(122)*Y(15)+C(123)*Y(16)+C(91)*Y(10)))+PNO                   SLOP0445
50 IF((KIND .EQ. 1 .AND. KEY(10) .NE. 1) .OR. (KIND .EQ. 2 .AND. KEY(SLOP0450
  110) .NE. 2)) GO TO   55                                                SLOP0455
   REMV(10)     =(Y(6)*(+C(84)+C(102)+C(110)*Y(18)+C(117))+Y(15)*(+C(1SLOP0460
  131)+C(143)*Y(16)+C(144)*Y(17)+C(150))+Y(11)*(+C(134)+C(147)*Y(18)+SLOP0465
  1C(156))+Y(2)*(+C(12)+C(114)*Y(18))+Y(10)*(+C(148)*Y(16)+C(148)*Y(1SLOP0470
  16))+C(20)*Y(1)*Y(18)+C(160)*Y(13)+C(162)+C(87)*Y(7)+C(91)*Y(8))        SLOP0475
   FORM(10)     =(Y(9)*(Y(1)*(+C(36)*Y(18)+C(41))+Y(3)*(+C(55)+C(68)*YSLOP0480
  1(18))+Y(5)*(+C(79)+C(56))+C(54)*Y(2)+C(57)*Y(4))+Y(11)*(Y(12)*(+C(SLOP0485
  1158)+C(158))+Y(15)*(+C(129)+C(141)*Y(18))+C(10)*Y(2)+C(155)*Y(16)+SLOP0490
  1C(118)*Y(7))+Y(15)*(Y(14)*(+C(152)+C(152))+C(149)*Y(17)+C(151)*Y(1SLOP0495
  12)+C(168)*Y(5))+Y(8)*(Y(3)*(+C(69)+C(69))+C(73)*Y(5)+C(123)*Y(16))SLOP0500
  1+C(163)*Y(12)+C(119)*Y(7)*Y(17)+C(126)*Y(4)*Y(17)+C(165)*Y(14))        SLOP0505
55 IF((KIND .EQ. 1 .AND. KEY(11) .NE. 1) .OR. (KIND .EQ. 2 .AND. KEY(SLOP0510
  111) .NE. 2)) GO TO   60                                                SLOP0515
   REMV(11)     =(Y(11)*(+C(132)+C(145)*Y(18)+C(132)+C(145)*Y(18))+Y(6SLOP0520
  1)*(+C(100)+C(108)*Y(18)+C(58)*Y(2))+Y(10)*(+C(134)+C(147)*Y(18)+C(SLOP0525
  11156))+Y(12)*(+C(157)+C(158)+C(159))+Y(3)*(+C(14)+C(115)*Y(18))+Y(7SLOP0530
  11)*(+C(118)+C(120))+Y(15)*(+C(129)+C(141)*Y(18))+C(133)*Y(16)+C(146SLOP0535
  1)*Y(16)*Y(18)+C(155)*Y(16)+C(10)*Y(2)+C(90)*Y(8)+C(113)*Y(9)*Y(18)SLOP0540
  1)                                                                      SLOP0545
   FORM(11)     =(Y(9)*(Y(1)*(+C(32)+C(37)*Y(18))+Y(3)*(+C(76)+C(77))+SLOP0550
  1Y(5)*(+C(78)+C(79))+C(81)*Y(4))+Y(8)*(Y(3)*(+C(72)+C(71)+C(72))+Y(SLOP0555
  11)*(+C(31)+C(31))+C(122)*Y(15))+Y(6)*(+C(58)*Y(2)*Y(11)+C(116)*Y(1SLOP0560
  17)+C(117)*Y(10))+Y(10)*(+C(162)+C(150)*Y(15))+C(165)*Y(14)+C(149)*SLOP0565
  1Y(15)*Y(17))                                                           SLOP0570
60 IF((KIND .EQ. 1 .AND. KEY(12) .NE. 1) .OR. (KIND .EQ. 2 .AND. KEY(SLOP0575
  112) .NE. 2)) GO TO   65                                                SLOP0580
   REMV(12)     =(Y(1)*(+C(18)+C(27)*Y(16)+C(28)*Y(17))+Y(11)*(+C(157)SLOP0585
```

```
     1+C(158)+C(159))+C(163)+C(85)*Y(6)+C(92)*Y(2)+C(95)*Y(3)+C(98)*Y(4)SLOP0590
     1+C(121)*Y(7)+C(151)*Y(15))                                        SLOP0595
      FORM(12)     =(Y(10)*(Y(15)*(+C(131)+C(143)*Y(16)+C(144)*Y(17))+Y(1SLOP0600
     10)*(+C(148)*Y(16)+C(148)*Y(16))+C(12)*Y(2)+C(160)*Y(13))+Y(5)*(+C(SLOP0605
     13)+C(44)*Y(6)+C(48)*Y(7)+C(56)*Y(9)+C(97)*Y(13)+C(52)*Y(8))+Y(3)*(SLOP0610
     1+C(14)*Y(11)+C(71)*Y(8)+C(75)*Y(9)+C(125)*Y(16)*Y(17))+Y(9)*(+C(67SLOP0615
     1)*Y(2)*Y(18)+C(82)*Y(4))+Y(11)*(+C(133)*Y(16)+C(146)*Y(16)*Y(18)))SLOP0620
65 IF((KIND .EQ. 1 .AND. KEY(13) .NE. 1) .OR. (KIND .EQ. 2 .AND. KEY(SLOP0625
     113) .NE. 2)) GO TO  70                                            SLOP0630
      REMV(13)     =(Y(1)*(+C(19)+C(20)+C(21))+Y(2)*(+C(13)+C(93))+C(166)SLOP0635
     1+C(96)*Y(3)+C(97)*Y(5)+C(154)*Y(15)+C(160)*Y(10))                 SLOP0640
      FORM(13)     =(Y(4)*(+C(4)+C(45)*Y(6)+C(49)*Y(7)+C(57)*Y(9)+C(98)*YSLOP0645
     1(12)+C(53)*Y(8))+Y(15)*(+C(15)*Y(3)+C(128)*Y(16)+C(138)*Y(16)*Y(16SLOP0650
     1)+C(139)*Y(16)*Y(17)+C(140)*Y(16)*Y(15))+Y(2)*(+C(9)*Y(16)+C(63)*YSLOP0655
     1(7)*Y(18))+Y(3)*(+C(62)*Y(6)*Y(18)+C(76)*Y(9))+C(80)*Y(5)*Y(  )+C(1SLOP0660
     121)*Y(7)*Y(12))                                                   SLOP0665
70 IF((KIND .EQ. 1 .AND. KEY(14) .NE. 1) .OR. (KIND .EQ. 2 .AND. KEY(SLOP0670
     114) .NE. 2)) GO TO  75                                            SLOP0675
      REMV(14)     =(Y(15)*(+C(152)+C(153))+C(164)+C(165)+C(86)*Y(6))    SLOP0680
      FORM(14)     =(Y(8)*(+C(65)*Y(2)*Y(16)+C(70)*Y(3)+C(73)*Y(5)+C(74)*SLOP0685
     1Y(4))+Y(11)*(Y(10)*(+C(134)+C(147)*Y(18))+C(159)*Y(12))+Y(15)*(+C(SLOP0690
     1130)*Y(17)+C(142)*Y(17)*Y(18))+C(11)*Y(2)*Y(17))                  SLOP0695
75 IF((KIND .EQ. 1 .AND. KEY(15) .NE. 1) .OR. (KIND .EQ. 2 .AND. KEY(SLOP0700
     115) .NE. 2)) GO TO  80                                            SLOP0705
      REMV(15)     =(Y(15)*(Y(15)*(+C(136)+C(136)+C(136))+C(127)+C(135)*YSLOP0710
     1(16)+C(137)*Y(17)+C(140)*Y(16)+C(127)+C(135)*Y(16)+C(137)*Y(17)+C(SLOP0715
     1140)*Y(16))+Y(1)*(+C(16)+C(22)*Y(16)+C(23)*Y(17)+C(26)*Y(16))+Y(10SLOP0720
     1)*(+C(131)+C(143)*Y(16)+C(144)*Y(17)+C(150))+Y(3)*(+C(7)+C(15)+C(9SLOP0725
     14))+Y(6)*(+C(99)+C(107)*Y(18)+C(61)*Y(2))+Y(7)*(+C(103)+C(111)*Y(1SLOP0730
     18))+Y(8)*(+C(88)+C(122))+Y(9)*(+C(105)+C(112)*Y(18))+Y(11)*(+C(129SLOP0735
     1)+C(141)*Y(18))+Y(14)*(+C(152)+C(153))+C(128)*Y(16)+C(130)*Y(17)+CSLOP0740
     1(138)*Y(16)*Y(16)+C(139)*Y(16)*Y(17)+C(142)*Y(17)*Y(18)+C(149)*Y(1SLOP0745
     17)+C(151)*Y(12)+C(154)*Y(13)+C(8)*Y(2)+C(167)*Y(4)+C(168)*Y(5))   SLOP0750
      FORM(15)     =(Y(6)*(Y(2)*(+C(42)+C(42)+C(61)*Y(15))+Y(1)*(+C(33)*YSLOP0755
     1(18)+C(38))+C(83)*Y(16)+C(84)*Y(10)+C(85)*Y(12)+C(86)*Y(14)+C(43)*SLOP0760
     1Y(3)+C(44)*Y(5)+C(45)*Y(4))+Y(1)*(Y(7)*(+C(30)+C(30))+Y(9)*(+C(32)SLOP0765
     1+C(37)*Y(18))+C(21)*Y(13)+C(26)*Y(16)*Y(15))+Y(2)*(+C(2)+C(46)*Y(7SLOP0770
     1)+C(50)*Y(8)+C(54)*Y(9)+C(93)*Y(13)+C(92)*Y(12))+Y(3)*(Y(8)*(+C(70SLOP0775
     1)+C(72))+Y(9)*(+C(75)+C(77))+C(7)*Y(15))+Y(11)*(+C(120)*Y(7)+C(155SLOP0780
     1)*Y(16)+C(156)*Y(10)+C(159)*Y(12))+Y(15)*(Y(15)*(+C(136)*Y(15)+C(1SLOP0785
     140)*Y(16)))+C(161)*Y(16)+C(166)*Y(13)+C(161)*Y(16)+C(162)*Y(10)+C(SLOP0790
     1163)*Y(12)+C(164)*Y(14))                                          SLOP0795
80 RETURN                                                               SLOP0800
     END                                                                SLOP0805
```

7.1.5 SUBROUTINE TO COMPUTE THE DERIVATIVES, DAUX

This subroutine is called by INTEG and computes the derivatives

$$\frac{dN_j}{dt} = \sum F_i - N_j \sum R_i .$$

This is a relatively short subroutine since the bulk of the computations are performed in SLOP which computes $\sum F_i$ and $\sum R_i$. If any species is in equilibrium this subroutine sets its derivative to zero. The following statements, beginning on page 88, are a listing for this subroutine.

```
IBFTC DAUX    LIST                                                          DAUX0000
      SUBROUTINE DAUX                                                       DAUX0005
      COMMON TREG(150),KEY(15),FORM(15),REMV(15),CON(168),LKEY(15),BEGINDAUX0010
     1(15)                                                                  DAUX0015
     -COMMON NUMB,EUBAR,ELBAR,D,DO2,DN2,T,PNE,PNO,PO2,PO,PN2,TOTAL,JACK,DAUX0020
     1JAKE,JAM,ITER,LINT,I2NT,K2NT,J2NT,N2NT,FIRST,PROD,LAM,TIME,K83,TOTDAUX0025
     20,TOTN                                                                DAUX0030
      KIND=1                                                                DAUX0035
      CALL SLOP(KIND)                                                       DAUX0040
    5 L=NUMB+4                                                              DAUX0045
      DO 20 J=1,NUMB                                                        DAUX0050
      IF(KEY(J) .NE. 1) GO TO 10                                            DAUX0055
      TREG(L)=FORM(J)-(REMV(J)*TREG(J+3))                                   DAUX0060
      GO TO 15                                                              DAUX0065
   10 TREG(L)=0.0                                                           DAUX0070
   15 L=L+1                                                                 DAUX0075
   20 CONTINUE                                                              DAUX0080
      IF(KEY(10) .NE. 1) GO TO 25                                           DAUX0085
      TREG(NUMB+13)=TREG(NUMB+13)-PNO                                       DAUX0090
   25 IF(KEY(15) .NE. 1) GO TO 30                                           DAUX0095
      TREG(NUMB+18)=TREG(NUMB+18)-PO                                        DAUX0100
   30 RETURN                                                                DAUX0105
      END                                                                   DAUX0110
```

7.1.6 SUBROUTINE TO SET INITIAL CONDITIONS, INITAL

This subroutine is incorporated into the package for the purpose of establishing the initial conditions on the solution of the differential equations. In most applications, this is a very simple subroutine. It simply sets the starting time to zero, the initial increment to 10^{-6}, and transfers the initial concentrations read into the BEGIN region to the proper TREG locations. It also computes the total numbers of oxygen (TOTO) and nitrogen(TOTN) atoms initially put into the system.

Since the situation could arise where it would be necessary to use a complicated function to compute the initial concentrations, as in the case of nuclear weapons effects, a separate subprogram is used to permit flexibility in setting up the initial conditions for the solutions without disrupting the entire code. The following locations of TREG must be set by this subroutine.

TREG(2) = 0.0, the starting time in seconds .

TREG(3) = 10^{-6}, the initial increment in seconds .

TREG(4) to TREG(NUMB+3) = the initial concentrations of the dependent variables N_{oi}.

The initial increment is always taken as 10^{-6} secs. There could be cases where a species has decayed or build up from its initial value to its equilibrium value in less than 10^{-6} secs. In the event that this should happen the initial integrating increment must be some value smaller than the equilibrium time. If a mesh of 10^{-6} secs is too large, the criterion set on the integration cannot be satisfied and the program will automatically cut it back until it arrives at an increment that can be accepted. If it is known that a species will reach its

equilibrium value before 10^{-6} secs, the increment can be initially set to some smaller value, thus saving computer time.

The following statements are a listing for this code.

```
$IBFTC INITAL  LIST                                                    INIT0000
       SUBROUTINE INITAL                                               INIT0005
       COMMON TREG(150),KEY(15),FORM(15),REMV(15),CON(168),LKEY(15),BEGININIT0010
      1(15)                                                            INIT0015
       COMMON NUMB,EUBAR,ELBAR,D,DO2,DN2,T,PNE,PNO,PO2,PO,PN2,TOTAL,JACK,INIT0020
      1JAKE,JAM,ITER,LINT,I2NT,K2NT,J2NT,N2NT,FIRST,PROD,LAM,TIME,KB3,TOTINIT0025
      20,TOTN                                                          INIT0030
       TREG(2)=0.0                                                     INIT0035
       TREG(3)=1.0E-6                                                  INIT0040
       FIRST=TREG(3)                                                   INIT0045
       DO 5 J=1,15                                                     INIT0050
     5 TREG(J+3)=BEGIN(J)                                              INIT0055
       TOTAL=PROD                                                      INIT0060
       TOTO=TREG(5)+2.0*TREG(6)+3.0*TREG(7)+2.0*TREG(8)+TREG(9)+2.0*TREG(INIT0065
      110)+TREG(12)+2.0*TREG(15)+3.0*TREG(16)+TREG(17)+TREG(13)+TREG(18)+INIT0070
      22.0*DO2                                                         INIT0075
       TOTN=TREG(8)+2.0*TREG(11)+TREG(12)+TREG(13)+TREG(14)+TREG(15)+2.0*INIT0080
      1TREG(17)+2.0*DN2                                                INIT0085
       RETURN                                                          INIT0090
       END                                                             INIT0095
```

7.1.7 SUBROUTINE TO COMPUTE THE EXTERNAL IONIZATION SOURCE FUNCTION, PRODUC

The insertion of any particular ionization source function into the system is facilitated by incorporating a separate subroutine for this purpose. The routine used with the deionization code simply injects a constant rate of production of ion-pairs/cm^3/sec into the differential equations. For simplicity this constant is read into the computer on one of the input data cards. This production represents the number of electrons produced by some external source other than the chemical equations. For simple deionization problems this constant production rate is partitioned among the positive ions as if it were an ultraviolet source. That is,

$$q(N_2^+) = 0.13926*PROD$$

$$q(O_2^+) = 0.79272*PROD$$

$$q(O^+) = 0.06802*PROD \quad .$$

Although the code will handle any time-dependent function it will not operate properly if a sharp discontinuity is encountered in the production function inside of the increment Δt. If discontinuities are present in the source function they must be accommodated for by the main program before the integration is attempted over the time increment in which they exist.

The following statements are a listing of this subroutine.

```
$IBFTC PRODUC  LIST                                                    PROD0000
       SUBROUTINE PRODUC                                               PROD0005
       COMMON TREG(150),KEY(15),FORM(15),REMV(15),CON(168),LKEY(15),BEGINPROD0010
      1(15)                                                            PROD0015
       COMMON NUMB,EUBAR,ELBAR,D,DO2,DN2,T,PNE,PNO,PO2,PO,PN2,TOTAL,JACK,PROD0020
      1JAKE,JAM,ITER,LINT,I2NT,K2NT,J2NT,N2NT,FIRST,PROD,LAM,TIME,KB3,TOTPROD0025
      20,TOTN                                                          PROD0030
     5 TOT=DO2+DN2+TREG(18)                                            PROD0035
       PO2=0.79272*PROD                                                PROD0040
       PN2=0.13926*PROD                                                PROD0045
       PO=0.06802*PROD                                                 PROD0050
       PNE=PROD                                                        PROD0055
       TOTAL=PNE                                                       PROD0060
       RETURN                                                          PROD0065
       END                                                             PROD0070
```

7.1.8 SUBROUTINE TO ESTABLISH CHARGE BALANCE, BALAN

This subroutine is called by SLOP whenever it is necessary to compute the concentration of a negative species from charge balance. After every successful integration the main program finds the largest negative species and sets the variable LAM to an appropriate integer. Using LAM in a computed GO TO, BALAN computes the concentration of this species merely by subtracting from the total number of positive ions the sum of the remaining negative ions. BALAN then adjusts the molecular-oxygen and the molecular-nitrogen reservoirs to insure that the total numbers of oxygen and nitrogen atoms initially in the system remains constant. The following statements are a listing of BALAN.

```
$IBFTC BALAN   LIST                                                    BALN0000
       SUBROUTINE BALAN                                                BALN0005
       COMMON TREG(150),KEY(15),FORM(15),REMV(15),CON(168),LKEY(15),BEGINBALN0010
      1(15)                                                            BALN0015
       COMMON NUMB,EUBAR,ELBAR,D,DO2,DN2,T,PNE,PNO,PO2,PO,PN2,TOTAL,JACK,BALN0020
      1JAKE,JAM,ITER,LINT,I2NT,K2NT,J2NT,N2NT,FIRST,PROD,LAM,TIME,KB3,TOTBALN0025
      20,TOTN                                                          BALN0030
     5 SUM=TREG(9)+TREG(10)+TREG(11)+TREG(12)                          BALN0035
       GO TO (10,15,20,25,30,35),LAM                                   BALN0040
    10 TREG(4)=SUM-TREG(5)-TREG(6)-TREG(7)-TREG(8)                     BALN0045
       GO TO 35                                                        BALN0050
    15 TREG(5)=SUM-TREG(4)-TREG(6)-TREG(7)-TREG(8)                     BALN0055
       GO TO 35                                                        BALN0060
    20 TREG(6)=SUM-TREG(4)-TREG(5)-TREG(7)-TREG(8)                     BALN0065
       GO TO 35                                                        BALN0070
    25 TREG(7)=SUM-TREG(4)-TREG(5)-TREG(6)-TREG(8)                     BALN0075
       GO TO 35                                                        BALN0080
    30 TREG(8)=SUM-TREG(4)-TREG(5)-TREG(6)-TREG(7)                     BALN0085
    35 DO2=((TOTO-TREG(5)-3.0*TREG(7)-TREG(9)-TREG(12)-3.0*TREG(16)-TREG(BALN0090
      113)-TREG(17)-TREG(18))/2.0)-TREG(6)-TREG(8)-TREG(10)-TREG(15)   BALN0095
       DN2=((TOTN-TREG(8)-TREG(12)-TREG(13)-TREG(14)-TREG(15))/2.0)-TREG(BALN0100
      111)-TREG(17)                                                    BALN0105
    40 RETURN                                                          BALN0110
       END                                                             BALN0115
```

7.1.9 SUBROUTINE TO COMPUTE THE HISTORY OF THE REACTIONS, DAUXT

Since it is not possible to determine the actual kinetics of the atmospheric constituents simply by studying the solutions to the differential equations, this program is incorporated to give a snapshot of the chemical reactions whenever it is called.

Each time DAUXT is called, each individual reaction is computed and the entire array of reactions along with the corresponding rate constants and the reaction rates are written on FORTRAN-tape 3. The reactions are written in a pseudochemical notation that is easy to read. By examination of this output it is a simple matter to determine which reactions are responsible for the behavior of any species at any particular time.

Computation of this information is optional. If switch KB2 is on zero the snapshot is taken. If switch KB2 is on one computation of this information is bypassed. There is also an option to print this history after every integration or only once at the beginning of each decade of tir e. If switch KB3 is on one the history is printed once for each decade of time. I switch KB3 is on zero the history is printed after every integration. These switches are set by one of the input data cards.

The following statements, beginning on page 92, are a listing of this code.

7.2 Diurnal Variation Code

The code, as described in Section 7.1, is readily applicable to the problem of the diurnal variation of the atmosp eric constituents. The only programs that require any modification are the main program and subroutines PRODUC and BALAN. One additional subprogram is necessary for the computation of the percent obscuration of the visible light from the sun by the solid earth during periods of sunrise and sunset.

7.2.1 MAIN PROGRAM DIURN

In addition to the functions described in Section 7.1.1, the main program must also compute the times of sunrise and sunset for the particular altitude and geographic location under consideration. Besides accumulating the time in seconds, the main program must also compute the local solar time in hours. Although the code must start with time equal to zero seconds, this zero time may refer to any local solar time. To accommodate these additional features more input parameters are required. The following are the changes in the input data as described in Section 7.1.1.3.

```
$IBFTC DAUXT   LIST                                                    DXT00000
      SUBROUTINE DAUXT                                                 DXT00005
C                                                                      DXT00010
C     REACTIONS FOR 15 SPECIES.                                        DXT00015
      COMMON TREG(150),KEY(15),FORM(15),RENV(15),CON(168),LKEY(15),BEGINDXT00020
     1(15)                                                             DXT00025
      COMMON NUMB,EUBAR,ELBAR,D,DO2,DN2,T,PNE,PNO,PO2,PO,PN2,TOTAL,JACK,DXT00030
     1JAKE,JAM,ITER,LINT,I2NT,K2NT,J2NT,N2NT,FIRST,PROD,LAM,TIME,KB3,TOTDXT00035
     20,TOTN                                                           DXT00040
      DIMENSION R(168)                                                 DXT00045
C                                                                      DXT00050
      IF(KB3 .NE. 1) GO TO 5                                           DXT00055
      IF(TIME .GT. TREG(2)) GO TO 10                                   DXT00060
    5 DO=TREG(18)                                                      DXT00065
      R(1)=TREG(6)*CON(1)                                              DXT00070
      R(  2)=TREG(5)*CON(2)                                            DXT00075
      R(  3)=TREG(8)*CON(3)                                            DXT00080
      R(  4)=TREG(7)*CON(4)                                            DXT00085
      R(  5)=TREG(6)*DO2*CON(5)                                        DXT00090
      R(  6)=TREG(6)*DN2*CON(6)                                        DXT00095
      R(  7)=TREG(6)*DO*CON(7)                                         DXT00100
      R(  8)=TREG(5)*DO*CON(8)                                         DXT00105
      R(  9)=TREG(5)*DO2*CON(9)                                        DXT00110
      R( 10)=TREG(5)*TREG(14)*CON(10)                                  DXT00115
      R( 11)=TREG(5)*DN2*CON(11)                                       DXT00120
      R( 12)=TREG(5)*TREG(13)*CON(12)                                  DXT00125
      R( 13)=TREG(5)*TREG(16)*CON(13)                                  DXT00130
      R( 14)=TREG(6)*TREG(14)*CON(14)                                  DXT00135
      R( 15)=TREG(6)*DO*CON(15)                                        DXT00140
      R( 16)=TREG(4)*DO*CON(16)                                        DXT00145
      R( 17)=TREG(4)*DO2*CON(17)                                       DXT00150
      R( 18)=TREG(4)*TREG(15)*CON(18)                                  DXT00155
      R( 19)=TREG(4)*TREG(16)*CON(19)                                  DXT00160
      R( 20)=TREG(4)*TREG(16)*CON(20)                                  DXT00165
      R( 21)=TREG(4)*TREG(15)*CON(21)                                  DXT00170
      R( 22)=TREG(4)*DO*(DO2*CON(22))                                  DXT00175
      R( 23)=TREG(4)*DO*(DN2*CON(23))                                  DXT00180
      R( 24)=TREG(4)*DO2*(DO2*CON(24))                                 DXT00185
      R( 25)=TREG(4)*DO2*(DN2*CON(25))                                 DXT00190
      R( 26)=TREG(4)*DO*(DO2*CON(26))                                  DXT00195
      R( 27)=TREG(4)*TREG(15)*(DO2*CON(27))                            DXT00200
      R( 28)=TREG(4)*TREG(15)*(DN2*CON(28))                            DXT00205
      R( 29)=TREG(4)*TREG(13)*(D*CON(29))                              DXT00210
      R( 30)=TREG(4)*TREG(10)*CON(30)                                  DXT00215
      R( 31)=TREG(4)*TREG(11)*CON(31)                                  DXT00220
      R( 32)=TREG(4)*TREG(12)*CON(32)                                  DXT00225
      R( 33)=TREG(4)*TREG(9)*(D*CON(33))                               DXT00230
      R( 34)=TREG(4)*TREG(10)*(D*CON(34))                              DXT00235
      R( 35)=TREG(4)*TREG(11)*(D*CON(35))                              DXT00240
      R( 36)=TREG(4)*TREG(12)*(D*CON(36))                              DXT00245
      R( 37)=TREG(4)*TREG(12)*(D*CON(37))                              DXT00250
      R( 38)=TREG(4)*TREG(9)*CON(38)                                   DXT00255
      R( 39)=TREG(4)*TREG(10)*CON(39)                                  DXT00260
      R( 40)=TREG(4)*TREG(11)*CON(40)                                  DXT00265
      R( 41)=TREG(4)*TREG(12)*CON(41)                                  DXT00270
      R( 42)=TREG(5)*TREG(9)*CON(42)                                   DXT00275
      R( 43)=TREG(6)*TREG(9)*CON(43)                                   DXT00280
      R( 44)=TREG(8)*TREG(9)*CON(44)                                   DXT00285
      R( 45)=TREG(7)*TREG(9)*CON(45)                                   DXT00290
```

```
R( 46)=TREG(5)*TREG(10)*CON(46)                                    DXT00295
R( 47)=TREG(6)*TREG(10)*CON(47)                                    DXT00300
R( 48)=TREG(8)*TREG(10)*CON(48)                                    DXT00305
R( 49)=TREG(7)*TREG(10)*CON(49)                                    DXT00310
R( 50)=TREG(5)*TREG(11)*CON(50)                                    DXT00315
R( 51)=TREG(6)*TREG(11)*CON(51)                                    DXT00320
R( 52)=TREG(8)*TREG(11)*CON(52)                                    DXT00325
R( 53)=TREG(7)*TREG(11)*CON(53)                                    DXT00330
R( 54)=TREG(5)*TREG(12)*CON(54)                                    DXT00335
R( 55)=TREG(6)*TREG(12)*CON(55)                                    DXT00340
R( 56)=TREG(8)*TREG(12)*CON(56)                                    DXT00345
R( 57)=TREG(7)*TREG(12)*CON(57)                                    DXT00350
R( 58)=TREG(5)*TREG(9)*(TREG(14)*CON(58))                          DXT00355
R( 59)=TREG(5)*TREG(9)*(DO2*CON(59))                               DXT00360
R( 60)=TREG(5)*TREG(9)*(DN2*CON(60))                               DXT00365
R( 61)=TREG(5)*TREG(9)*(DO*CON(61))                                DXT00370
R( 62)=TREG(6)*TREG(9)*(D*CON(62))                                 DXT00375
R( 63)=TREG(5)*TREG(10)*(D*CON(63))                                DXT00380
R( 64)=TREG(6)*TREG(10)*(D*CON(64))                                DXT00385
R( 65)=TREG(5)*TREG(11)*(D*CON(65))                                DXT00390
R( 66)=TREG(6)*TREG(11)*(D*CON(66))                                DXT00395
R( 67)=TREG(5)*TREG(12)*(D*CON(67))                                DXT00400
R( 68)=TREG(6)*TREG(12)*(D*CON(68))                                DXT00405
R( 69)=TREG(6)*TREG(11)*CON(69)                                    DXT00410
R( 70)=TREG(6)*TREG(11)*CON(70)                                    DXT00415
R( 71)=TREG(6)*TREG(11)*CON(71)                                    DXT00420
R( 72)=TREG(6)*TREG(11)*CON(72)                                    DXT00425
R( 73)=TREG(8)*TREG(11)*CON(73)                                    DXT00430
R( 74)=TREG(7)*TREG(11)*CON(74)                                    DXT00435
R( 75)=TREG(6)*TREG(12)*CON(75)                                    DXT00440
R( 76)=TREG(6)*TREG(12)*CON(76)                                    DXT00445
R( 77)=TREG(6)*TREG(12)*CON(77)                                    DXT00450
R( 78)=TREG(8)*TREG(12)*CON(78)                                    DXT00455
R( 79)=TREG(8)*TREG(12)*CON(79)                                    DXT00460
R( 80)=TREG(8)*TREG(12)*CON(80)                                    DXT00465
R( 81)=TREG(7)*TREG(12)*CON(81)                                    DXT00470
R( 82)=TREG(7)*TREG(12)*CON(82)                                    DXT00475
R( 83)=TREG(9)*DO2*CON(83)                                         DXT00480
R( 84)=TREG(9)*TREG(13)*CON(84)                                    DXT00485
R( 85)=TREG(9)*TREG(15)*CON(85)                                    DXT00490
R( 86)=TREG(9)*TREG(17)*CON(86)                                    DXT00495
R( 87)=TREG(10)*TREG(13)*CON(87)                                   DXT00500
R( 88)=TREG(11)*DO*CON(88)                                         DXT00505
R( 89)=TREG(11)*DO2*CON(89)                                        DXT00510
R( 90)=TREG(11)*TREG(14)*CON(90)                                   DXT00515
R( 91)=TREG(11)*TREG(13)*CON(91)                                   DXT00520
R( 92)=TREG(5)*TREG(15)*CON(92)                                    DXT00525
R( 93)=TREG(5)*TREG(16)*CON(93)                                    DXT00530
R( 94)=TREG(6)*DO*CON(94)                                          DXT00535
R( 95)=TREG(6)*TREG(15)*CON(95)                                    DXT00540
R( 96)=TREG(6)*TREG(16)*CON(96)                                    DXT00545
R( 97)=TREG(8)*TREG(16)*CON(97)                                    DXT00550
R( 98)=TREG(7)*TREG(15)*CON(98)                                    DXT00555
R( 99)=TREG(9)*DO*CON(99)                                          DXT00560
R(100)=TREG(9)*TREG(14)*CON(100)                                   DXT00565
R(101)=TREG(9)*DN2*CON(101)                                        DXT00570
R(102)=TREG(9)*TREG(13)*CON(102)                                   DXT00575
R(103)=TREG(10)*DO*CON(103)                                        DXT00580
R(104)=TREG(11)*DN2*CON(104)                                       DXT00585
```

```
      R(105)=TREG(12)*DO*CON(105)                                    DXT00590
      R(106)=TREG(5)*DO2*CON(106)                                    DXT00595
      R(107)=TREG(9)*DO*(D*CON(107))                                 DXT00600
      R(108)=TREG(9)*TREG(14)*(D*CON(108))                           DXT00605
      R(109)=TREG(9)*DN2*(D*CON(109))                                DXT00610
      R(110)=TREG(9)*TREG(13)*(D*CON(110))                           DXT00615
      R(111)=TREG(10)*DO*(D*CON(111))                                DXT00620
      R(112)=TREG(12)*DO*(D*CON(112))                                DXT00625
      R(113)=TREG(12)*TREG(14)*(D*CON(113))                          DXT00630
      R(114)=TREG(5)*TREG(13)*(D*CON(114))                           DXT00635
      R(115)=TREG(6)*TREG(14)*(D*CON(115))                           DXT00640
      R(116)=TREG(9)*DN2*CON(116)                                    DXT00645
      R(117)=TREG(9)*TREG(13)*CON(117)                               DXT00650
      R(118)=TREG(10)*TREG(14)*CON(118)                              DXT00655
      R(119)=TREG(10)*DN2*CON(119)                                   DXT00660
      R(120)=TREG(10)*TREG(14)*CON(120)                              DXT00665
      R(121)=TREG(10)*TREG(15)*CON(121)                              DXT00670
      R(122)=TREG(11)*DO*CON(122)                                    DXT00675
      R(123)=TREG(11)*DO2*CON(123)                                   DXT00680
      R(124)=TREG(5)*DO2*(DO2*CON(124))                              DXT00685
      R(125)=TREG(6)*DO2*(DN2*CON(125))                              DXT00690
      R(126)=TREG(7)*DN2*CON(126)                                    DXT00695
      R(127)=DO*(DO*CON(127))                                        DXT00700
      R(128)=DO*(DO2*CON(128))                                       DXT00705
      R(129)=TREG(14)*DO*CON(129)                                    DXT00710
      R(130)=DO*DN2*CON(130)                                         DXT00715
      R(131)=TREG(13)*DO*CON(131)                                    DXT00720
      R(132)=TREG(14)*TREG(14)*CON(132)                              DXT00725
      R(133)=TREG(14)*DO2*CON(133)                                   DXT00730
      R(134)=TREG(14)*TREG(13)*CON(134)                              DXT00735
      R(135)=DO*DO*(DO2*CON(135))                                    DXT00740
      R(136)=DO*DO*(DO*CON(136))                                     DXT00745
      R(137)=DO*DO*(DN2*CON(137))                                    DXT00750
      R(138)=DO*DO2*(DO2*CON(138))                                   DXT00755
      R(139)=DO*DO2*(DN2*CON(139))                                   DXT00760
      R(140)=DO*DO2*(DO*CON(140))                                    DXT00765
      R(141)=TREG(14)*DO*(D*CON(141))                                DXT00770
      R(142)=DO*(DN2*CON(142))*D                                     DXT00775
      R(143)=DO*TREG(13)*(DO2*CON(143))                              DXT00780
      R(144)=DO*TREG(13)*(DN2*CON(144))                              DXT00785
      R(145)=TREG(14)*TREG(14)*(D*CON(145))                          DXT00790
      R(146)=TREG(14)*(DO2*CON(146))*D                               DXT00795
      R(147)=TREG(14)*TREG(13)*(D*CON(147))                          DXT00800
      R(148)=TREG(13)*(DO2*CON(148))*TREG(13)                        DXT00805
      R(149)=(DO*CON(149))*DN2                                       DXT00810
      R(150)=(DO*CON(150))*TREG(13)                                  DXT00815
      R(151)=(DO*CON(151))*TREG(15)                                  DXT00820
      R(152)=(DO*CON(152))*TREG(17)                                  DXT00825
      R(153)=(DO*CON(153))*TREG(17)                                  DXT00830
      R(154)=(DO*CON(154))*TREG(16)                                  DXT00835
      R(155)=TREG(14)*(DO2*CON(155))                                 DXT00840
      R(156)=TREG(14)*TREG(13)*CON(156)                              DXT00845
      R(157)=TREG(14)*TREG(15)*CON(157)                              DXT00850
      R(158)=TREG(14)*TREG(15)*CON(158)                              DXT00855
      R(159)=TREG(14)*TREG(15)*CON(159)                              DXT00860
      R(160)=TREG(14)*TREG(16)*CON(160)                              DXT00865
      R(161)=DO2*CON(161)                                            DXT00870
      R(162)=TREG(13)*CON(162)                                       DXT00875
      R(163)=TREG(15)*CON(163)                                       DXT00880
```

```
R(164)=TREG(17)*CON(164)                                                DXT00885
R(165)=TREG(17)*CON(165)                                                DXT00890
R(166)=TREG(16)*CON(166)                                                DXT00895
R(167)=TREG(7)*TREG(18)*CON(167)                                        DXT00900
R(168)=TREG(8)*TREG(18)*CON(168)                                        DXT00905
WRITE(3,15)                                                             DXT00910
WRITE (3,20)     TREG(2)                                                DXT00915
WRITE(3,25)                                                             DXT00920
WRITE(3,30)                                                             DXT00925
WRITE (3,35)     (CON(J),R(J),J=1,4)                                    DXT00930
WRITE(3,40)                                                             DXT00935
WRITE(3,30)                                                             DXT00940
WRITE (3,45)     (CON(J),R(J),J=5,7)                                    DXT00945
WRITE(3,50)                                                             DXT00950
WRITE(3,30)                                                             DXT00955
WRITE(3,55) (CON(J),R(J),J=8,15),(CON(J),R(J),J=167,168)                DXT00960
WRITE(3,60)                                                             DXT00965
WRITE(3,30)                                                             DXT00970
WRITE (3,65)     (CON(J),R(J),J=16,19)                                  DXT00975
WRITE(3,70)                                                             DXT00980
WRITE(3,30)                                                             DXT00985
WRITE (3,75)     (CON(J),R(J),J=20,21)                                  DXT00990
WRITE(3,80)                                                             DXT00995
WRITE(3,30)                                                             DXT01000
WRITE (3,85)     (CON(J),R(J),J=22,29)                                  DXT01005
WRITE(3,90)                                                             DXT01010
WRITE(3,30)                                                             DXT01015
WRITE (3,95)     (CON(J),R(J),J=30,32)                                  DXT01020
WRITE(3,100)                                                            DXT01025
WRITE(3,30)                                                             DXT01030
WRITE (3,105)     (CON(J),R(J),J=33,37)                                 DXT01035
WRITE(3,110)                                                            DXT01040
WRITE(3,30)                                                             DXT01045
WRITE (3,115)     (CON(J),R(J),J=38,41)                                 DXT01050
WRITE(3,120)                                                            DXT01055
WRITE(3,30)                                                             DXT01060
WRITE (3,125)     (CON(J),R(J),J=42,53)                                 DXT01065
WRITE (3,130)     (CON(J),R(J),J=54,57)                                 DXT01070
WRITE(3,135)                                                            DXT01075
WRITE(3,30)                                                             DXT01080
WRITE (3,140)     (CON(J),R(J),J=58,68)                                 DXT01085
WRITE(3,145)                                                            DXT01090
WRITE(3,30)                                                             DXT01095
WRITE (3,150)     (CON(J),R(J),J=69,80)                                 DXT01100
WRITE (3,155)     (CON(J),R(J),J=81,82)                                 DXT01105
WRITE(3,160)                                                            DXT01110
WRITE(3,30)                                                             DXT01115
WRITE (3,165)     (CON(J),R(J),J=83,91)                                 DXT01120
WRITE(3,170)                                                            DXT01125
WRITE(3,30)                                                             DXT01130
WRITE (3,175)     (CON(J),R(J),J=92,98)                                 DXT01135
WRITE(3,180)                                                            DXT01140
WRITE(3,30)                                                             DXT01145
WRITE (3,185)     (CON(J),R(J),J=99,106)                                DXT01150
WRITE(3,190)                                                            DXT01155
WRITE(3,30)                                                             DXT01160
WRITE (3,195)     (CON(J),R(J),J=107,115)                               DXT01165
WRITE(3,200)                                                            DXT01170
WRITE(3,30)                                                             DXT01175
```

```
      WRITE (3,205)    (CON(J),R(J),J=116,123)                                DXT01180
      WRITE(3,210)                                                            DXT01185
      WRITE(3,30)                                                             DXT01190
      WRITE (3,215)    (CON(J),R(J),J=124,126)                                DXT01195
      WRITE(3,220)                                                            DXT01200
      WRITE(3,30)                                                             DXT01205
      WRITE (3,225)    (CON(J),R(J),J=127,134)                                DXT01210
      WRITE(3,230)                                                            DXT01215
      WRITE(3,30)                                                             DXT01220
      WRITE (3,235)    (CON(J),R(J),J=135,146)                                DXT01225
      WRITE (3,260)     (CON(147),R(147))                                     DXT01230
      WRITE(3,240)                                                            DXT01235
      WRITE(3,30)                                                             DXT01240
      WRITE (3,285)     (CON(148),R(148))                                     DXT01245
      WRITE (3,245)    (CON(J),R(J),J=149,160)                                DXT01250
      WRITE(3,250)                                                            DXT01255
      WRITE(3,30)                                                             DXT01260
      WRITE (3,255)     (CON(J),R(J),J=161,166)                               DXT01265
      WRITE (3,265)     TOTAL                                                 DXT01270
      WRITE(3,290)                                                            DXT01275
      WRITE (3,270)     PTNO,PNO,PTO2,PO2,PTO,PO,PTN2,PN2                     DXT01280
      WRITE (3,295)                                                           DXT01285
      WRITE(3,290)                                                            DXT01290
      WRITE (3,275)     (J,R(J),J=1,165)                                      DXT01295
      WRITE(3,280) (J,R(J),J=166,168)                                         DXT01300
      TIME=TIME*10.0                                                          DXT01305
   10 RETURN                                                                  DXT01310
   15 FORMAT(1H1,47X28H KINETICS OF THE REACTIONS. )                          DXT01315
   20 FORMAT(1HO,47X7H TIME =1PE12.5,6H SEC. )                                DXT01320
   25 FORMAT(19HOPHOTO DETACHMENT. )                                          DXT01325
   30 FORMAT(6HO  NO.,15X8HREACTION,17X16HCONSTANT     RATE,6X3HNO.,15X8HDXT01330
     1REACTION,17X17HCONSTANT     RATE )                                      DXT01335
   35 FORMAT(4X12H1. O2-  + HV,9X10H= O2    + E,10X1P2E10.2,4X12H2. O-       DXT01340
     1+ HV,9X10H= O     + E,10X1P2E10.2/4X12H3. NO2- + HV,9X10H= NO2  + EDXT01345
     2,10X1P2E10.2,4X12H4. O3-  + HV,9X10H= O3    + E,10X1P2E10.2)           DXT01350
   40 FORMAT(25HOCOLLISIONAL DETACHMENT. )                                    DXT01355
   45 FORMAT(4X12H5. O2-  + O2,9X18H= O2    + E    + O2,2X1P2E10.2,4X12H6DXT01360
     1. O2-  + N2,9X18H= O2    + E    + N2,2X1P2E10.2/4X11H7. O2-  + O,10DXT01365
     2X17H= O2    + E    + O,3X1P2E10.2)                                     DXT01370
   50 FORMAT(25HOASSOCIATIVE DETACHMENT. )                                    DXT01375
   55 FORMAT(4X11H8. O-    + O,10X10H= O2    + E,10X1P2E10.2,4X12H9. O-      DXT01380
     1+ O2,9X10H= O3    + E,10X1P2E10.2/3X12H10. O-    + N,10X10H= NO    + DXT01385
     2E,10X1P2E10.2,3X13H11. O-    + N2,9X10H= N2O   + E,10X1P2E10.2/3X13HDXT01390
     312. O-    + NO,9X10H= NO2   + E,10X1P2E10.2,3X13H13. O-    + O3,9X20HDXT01395
     4= O2    + E    + O2  1P2E10.2/3X12H14. O2-  + N,10X10H= NO2  + E,10DXT01400
     5X1P2E10.2,3X12H15. O2-  + O,10X10H= O3    + E,10X1P2E10.2 /3X13H167DXT01405
     6. O3-  + O,10X17H= O2    + O2    + E,3X1P2E10.2,2X13H168. NO2- + O,1DXT01410
     70X17H= O2    + NO    + E,3X1P2E10.2)                                   DXT01415
   60 FORMAT(23HORADIATIVE ATTACHMENT. )                                      DXT01420
   65 FORMAT(3X12H16. O    + E,10X11H= O-    + HV,9X1P2E10.2,3X12H17. O2 DXT01425
     1  + E,10X11H= O2-  + HV,9X1P2E10.2/3X12H18. NO2  + E,10X11H= NO2- DXT01430
     2+ HV,9X1P2E10.2,3X12H19. O3    + E,10X11H= O3-  + HV,9X1P2E10.2)       DXT01435
   70 FORMAT(26HODISSOCIATIVE ATTACHMENT. )                                   DXT01440
   75 FORMAT(3X12H20. O3    + E,10X11H= O-    + O2,9X1P2E10.2,3X12H21. O3 DXT01445
     1  + O,10X10H= O2-  + O,10X1P2E10.2)                                     DXT01450
   80 FORMAT(24HOTHREE BODY ATTACHMENT. )                                     DXT01455
   85 FORMAT(3X33H22. O    + E    + O2  = O-    + O2,9X1P2E10.2,3X33H23. DXT01460
     10    + E    + N2  = O-    + N2,9X1P2E10.2/3X33H24. O2    + E    + O2DXT01465
     2  = O2-  + O2,9X1P2E10.2,3X33H25. O2    + E    + N2  = O2-  + N2,9XDXT01470
```

```
   31P2E10.2/3X32H26. O2   + E    + O   = O2-  + O,10X1P2E10.2,3X33H27DXT01475
   4. NO2  + E    + O2  = NO2- + O2,9X1P2E10.2/3X33H28. NO2  + E    + DXT01480
   5N2  = NO2- + N2,9X1P2E10.2,3X32H29. NO   + E    + M   = NO-  + M,1DXT01485
   60X1P2E10.2)                                                      DXT01490
 90 FORMAT(29HODISSOCIATIVE RECOMBINATION. )                         DXT01495
 95 FORMAT(3X12H30. O2+  + E,10X10H= O    + O,10X1P2E10.2,3X12H31. N2+DXT01500
   1  + E,10X10H= N    + N,10X1P2E10.2/3X12H32. NO+  + E,10X10H= N    DXT01505
   .2+ O,10X1P2E10.2)                                                DXT01510
100 FORMAT(27HOTHREE BODY RECOMBINATION. )                           DXT01515
105 FORMAT(3X32H33. O+   + E    + M   = O    + M,10X1P2E10.2,3X32H34. DXT01520
   102+  + E    + M   = O2   + M,10X1P2E10.2/3X32H35. N2+  + E    + M DXT01525
   2  = N2   + M,10X1P2E10.2,3X32H36. NO+  + E    + M   = NO   + M,10XDXT01530
   31P2E10.2/3X32H37. NO+  + E    + M   = N    + O    + M,3X1P2E10.2) DXT01535
110 FORMAT(26HORADIATIVE RECOMBINATION. )                             DXT01540
115 FORMAT(3X12H38. O+   + E,10X11H= O    + HV,9X1P2E10.2,3X12H39. O2+DXT01545
   1  + E,10X11H= O2   + HV,9X1P2E10.2/3X12H40. N2+  + E,10X11H= N2   DXT01550
   2+ HV,9X1P2E10.2,3X12H41. NO+  + E,10X11H= NO   + HV,9X1P2E10.2)   DXT01555
120 FORMAT(32H1ION-ION MUTUAL NEUTRALIZATION. )                       DXT01560
125 FORMAT(3X13H42. O-   + O+,9X10H= O    + O,10X1P2E10.2,3X13H43. O2-DXT01565
   1  + O+,9X10H= O2   + O,10X1P2E10.2/3X13H44. NO2- + O+,9X10H= NO2  DXT01570
   2+ O,10X1P2E10.2,3X13H45. O3-  + O+,9X10H= O3   + O,10X1P2E10.2/3X1DXT01575
   34H46. O-   + O2+,8X11H= O    + O2,9X1P2E10.2,3X14H47. O2-  + O2+,8DXT01580
   4X11H= O2   + O2,9X1P2E10.2/3X14H48. NO2- + O2+,8X11H= NO2  + O2,9XDXT01585
   51P2E10.2,3X14H49. O3-  + O2+,8X11H= O3   + O2,9X1P2E10.2/3X14H50. DXT01590
   6O-   + N2+,8X11H= O    + N2,9X1P2E10.2,3X14H51. O2-  + N2+,8X11H= DXT01595
   702   + N2,9X1P2E10.2/3X14H52. NO2- + N2+,8X11H= NO2  + N2,9X1P2E10DXT01600
   8.2,3X14H53. O3-  + N2+,8X11H= O3   + N2,9X1P2E10.2)                DXT01605
130 FORMAT(3X14H54. O-   + NO+,8X11H= O    + NO,9X1P2E10.2,3X14H55. O2DXT01610
   1-  + NO+,8X11H= O2   + NO,9X1P2E10.2/3X14H56. NO2- + NO+,8X11H= NODXT01615
   22  + NO,9X1P2E10.2,3X14H57. O3-  + NO+,8X11H= O3   + NO,9X1P2E10.2DXT01620
   3)                                                                DXT01625
135 FORMAT(35HOTHREE BODY ION-ION RECOMBINATION. )                    DXT01630
140 FORMAT(3X32H58. O-   + O+   + N   = O2   + N,10X1P2E10.2,3X33H59. DXT01635
   10-   + O+   + O2  = O2   + O2,9X1P2E10.2/3X33H60. O-   + O2   + N2DXT01640
   2  = O2   + N2,9X1P2E10.2,3X32H61. O-   + O+   + O   = O2   + O,10XDXT01645
   31P2E10.2/3X32H62. O2-  + O+   + M   = O3   + M,10X1P2E10.2,3X32H63DXT01650
   4. O-   + O2+  + M   = O3   + M,10X1P2E10.2/3X39H64. O2-  + O2+  + DXT01655
   5M   = O2   + O2   + M,3X1P2E10.2,3X32H65. O-   + N2+  + M   = N2O DXT01660
   6 + M,10X1P2E10.2/3X39H66. O2-  + N2+  + M   = O2   + N2   + M,3X1PDXT01665
   72E10.2,3X32H67. O-   + NO+  + M   = NO2  + M,10X1P2E10.2/3X39H68. DXT01670
   8O2-  + NO+  + M   = O2   + NO   + M,3X1P2E10.2)                  DXT01675
145 FORMAT(44HOION-ION NEUTRALIZATION WITH REARRANGEMENT. )           DXT01680
150 FORMAT(3X14H69. O2-  + N2+,8X11H= NO   + NO,9X1P2E10.2,3X14H70. O2DXT01685
   1-  + N2+,8X10H= N2O  + O,10X1P2E10.2/3X14H71. O2-  + N2+,8X10H= NODXT01690
   22  + N,10X1P2E10.2,3X14H72. O2-  + N2+,8X11H= 2N   + 2O,9X1P2E10.2DXT01695
   3/3X14H73. NO2- + N2+,8X11H= N2O  + NO,9X1P2E10.2,3X14H74. O3-  + NDXT01700
   42+,8X11H= N2O  + O2,9X1P2E10.2/3X14H75. O2-  + NO+,8X10H= NO2  + ODXT01705
   5,10X1P2E10.2,3X14H76. O2-  + NO+,8X11H= N    + O3,9X1P2E10.2/3X14HDXT01710
   677. O2-  + NO+,8X17H= N    + O2   + O,3X1P2E10.2,3X14H78. NO2- + NDXT01715
   70+,8X10H= NO3  + N,10X1P2E10.2/3X14H79. NO2- + NO+,8X17H= NO   + ODXT01720
   82   + N,3X1P2E10.2,3X14H80. NO2- + NO+,8X11H= O3   + N2,9X1P2E10.2DXT01725
   9)                                                                DXT01730
155 FORMAT(3X14H81. O3-  + NO+,8X17H= O2   + O2   + N,3X1P2E10.2,3X14HDXT01735
   182. O3-  + NO+,8X12H= O2   + NO2,8X1P2E10.2)                    DXT01740
160 FORMAT(27HOPOSITIVE CHARGE TRANSFER. )                           DXT01745
165 FORMAT(3X13H83. O+   + O2,9X12H= O    + O2+,8X1P2E10.2,3X13H84. O+DXT01750
   1  + NO,9X12H= O    + NO+,8X1P2E10.2/3X14H85. O+   + NO2,8X13H= O DXT01755
   2   + NO2+,7X1P2E10.2,3X14H86. O+   + N2O,8X13H= O    + N2O+,7X1P2EDXT01760
   310.2/3X13H87. O2+  + NO,9X12H= O2   + NO+,8X1P2E10.2,3X12H88. N2+ DXT01765
```

```
   4 + O,10X11H= N2    + O+,9X1P2E10.2/3X13H89. N2+   + O2,9X12H= N2    +DXT01770
   5 O2+,8X1P2E10.2,3X12H90. N2+   + N,10X11H= N2    + N+,9X1P2E10.2/3X1DXT01775
   63H91. N2+   + NO,9X12H= N2    + NO+,8X1P2E10.2)                        DXT01780
170 FORMAT(27HONEGATIVE CHARGE TRANSFER. )                                DXT01785
175 FORMAT(3X14H92. O-    + NO2,8X13H= O     + NO2-,7X1P2E10.2,3X13H93. DXT01790
   10-    + O3,9X12H= O     + O3-,8X1P2E10.2/3X12H94. O2-   + O,10X11H= ODXT01795
   22    + O-,9X1P2E10.2,3X14H95. O2-   + NO2,8X13H= O2    + NO2-,7X1P2E1DXT01800
   30.2/3X13H96. O2-   + O3,9X12H= O2    + O3-,8X1P2E10.2,3X13H97. NO2-  DXT01805
   4+ O3,9X12H= NO2   + O3-,8X1P2E10.2/3X14H98. O3-   + NO2,8X13H= O3    DXT01810
   5+ NO2-,7X1P2E10.2)                                                    DXT01815
180 FORMAT(37HOION-NEUTRAL ASSOCIATION (TWO-BODY). )                      DXT01820
185 FORMAT(3X12H99. O+    + O,10X11H= O2+   + HV,9X1P2E10.2,2X13H100. O+DXT01825
   1    + N,10X11H= NO+   + HV,9X1P2E10.2/2X14H101. O+    + N2,9X11H= N2ODXT01830
   2+ + HV,9X1P2E10.2,2X14H102. O+    + NO,9X11H= NO2+  + HV,9X1P2E10.2/DXT01835
   32X13H103. O2+   + O,10X11H= O3+   + HV,9X1P2E10.2,2X14H104. N2+   + NDXT01840
   42,9X11H= N4+   + HV,9X1P2E10.2/2X13H105. NO+   + O,10X11H= NO2+  + HVDXT01845
   5,9X1P2E10.2,2X14H106. O-    + O2,9X11H= O3-   + HV,9X1P2E10.2)       DXT01850
190 FORMAT(39H1ION-NEUTRAL ASSOCIATION (THREE-BODY). )                     DXT01855
195 FORMAT(2X33H107. O+    + O     + M    = O2+   + M,10X1P2E10.2,2X33H108DXT01860
   1. O+    + N     + M    = NO+   + M,10X1P2E10.2/2X33H109. O+    + N2   +DXT01865
   2 M    = N2O+  + M,10X1P2E10.2,2X33H110. O+    + NO    + M    = NO2+  + MDXT01870
   3,10X1P2E10.2/2X33H111. O2+   + O     + M    = O3+   + M,10X1P2E10.2,2XDXT01875
   433H112. NO+   + O     + M    = NO2+  + M,10X1P2E10.2/2X33H113. NO+   + DXT01880
   5N     + M    = N2O+  + M,10X1P2E10.2,2X33H114. O-    + NO    + M    = NODXT01885
   62- + M,10X1P2E10.2/2X33H115. O2-   + N     + M    = NO2- + M,10X1P2E1DXT01890
   70.2)                                                                  DXT01895
200 FORMAT(37HOCHARGED REARRANGEMENT POSITIVE-ION. )                      DXT01900
205 FORMAT(2X14H116. O+    + N2,9X10H= NO+   + N,10X1P2E10.2,2X14H117. ODXT01905
   1+    + NO,9X10H= O2+   + N,10X1P2E10.2/2X13H118. O2+   + N,10X11H= O+DXT01910
   2    + NO,9X1P2E10.2,2X14H119. O2+   + N2,9X11H= NO+   + NO,9X1P2E10.2DXT01915
   3/2X13H120. O2+   + N,10X10H= NO+   + O,10X1P2E10.2,2X15H121. O2+   + DXT01920
   4NO2,8X11H= NO+   + O3,9X1P2E10.2/2X13H122. N2+   + O,10X10H= NO+   + DXT01925
   5N,10X1P2E10.2,2X14H123. N2+   + O2,9X11H= NO+   + NO,9X1P2E10.2)     DXT01930
210 FORMAT(37HOCHARGED REARRANGEMENT NEGATIVE-ION. )                       DXT01935
215 FORMAT(2X34H124. O-    + O2    + O2   = O3-   + O2,9X1P2E10.2,2X35H125DXT01940
   1. O2-   + O2    + N2   = NO2-  + NO2,8X1P2E10.2/2X14H126. O3-   + N2,9XDXT01945
   211H= NO2-  + NO,9X1P2E10.2)                                           DXT01950
220 FORMAT(30HOTWO-BODY ATOM RECOMBINATION. )                             DXT01955
225 FORMAT(2X13H127. O     + O,10X11H= O2    + HV,9X1P2E10.2,2X14H128. ODXT01960
   1    + O2,9X11H= O3    + HV,9X1P2E10.2/2X13H129. N     + O,10X11H= NODXT01965
   2    + HV,9X1P2E10.2,2X14H130. O     + N2,9X11H= N2O   + HV,9X1P2E10.2DXT01970
   3/2X14H131. O     + NO,9X11H= NO2   + HV,9X1P2E10.2,2X13H132. N     + DXT01975
   4N,10X11H= N2    + HV,9X1P2E10.2/2X14H133. N     + O2,9X11H= NO2   + HDXT01980
   5V,9X1P2E10.2,2X14H134. N     + NO,9X11H= N2O   + HV,9X1P2E10.2)       DXT01985
230 FORMAT(32HOTHREE-BODY ATOM RECOMBINATION. )                            DXT01990
235 FORMAT(2X34H135. O     + O     + O2   = O2    + O2,9X1P2E10.2,2X33H136DXT01995
   1. O     + O     + O    = O2    + O,10X1P2E10.2/2X34H137. O     + O    +DXT02000
   2 N2   = O2    + N2,9X1P2E10.2,2X34H138. O     + O2    + O2   = O3    + ODXT02005
   32,9X1P2E10.2/2X34H139. O     + O2    + N2   = O3    + N2,9X1P2E10.2,2XDXT02010
   433H140. O     + O2    + O    = O3    + O,10X1P2E10.2/2X33H141. N     + DXT02015
   50    + M    = NO    + M,10X1P2E10.2,2X33H142. O     + N2    + M    = N2OXT02020
   60  + M,10X1P2E10.2/2X34H143. O     + NO    + O2   = NO2   + O2,9X1P2E10DXT02025
   70.2,2X34H144. O     + NO    + N2   = NO2   + N2,9X1P2E10.2/2X33H145. NDXT02030
   8     + N     + M    = N2    + M,10X1P2E10.2,2X33H146. N     + O2    + M DXT02035
   9   = NO2   + M,10X1P2E10.2)                                           DXT02040
240 FORMAT(24HONEUTRAL REARRANGEMENT. )                                    DXT02045
245 FORMAT(2X14H149. O     + N2,9X10H= NO    + N,10X1P2E10.2,2X14H150. ODXT02050
   1    + NO,9X10H= O2    + N,10X1P2E10.2/2X15H151. O     + NO2,8X11H= NDXT02055
   20    + O2,9X1P2E10.2,2X15H152. O     + N2O,8X11H= NO    + NO,9X1P2E10DXT02060
```

```
   3.2/2X15H153. O     + N2O,8X11H= O2    + N2,9X1P2E10.2,2X14H154. O    DXT02065
   4 + O3,9X11H= O2    + O2,9X1P2E10.2/2X14H155. N     + O2,9X10H= NO    DXT02070
   5+ O,10X1P2E10.2,2X14H156. N     + NO,9X10H= N2    + O,10X1P2E10.2/2XDXT02075
   615H157. N     + NO2,8X11H= N2    + O2,9X1P2E10.2,2X15H158. N     + NODXT02080
   72,8X11H= NO    + NO,9X1P2E10.2/2X15H159. N     + NO2,8X10H= N2O   + ODXT02085
   8,10X1P2E10.2,2X14H160. NO    + O3,9X11H= NO2   + O2,9X1P2E10.2)      DXT02090
250 FORMAT(21H1PHOTO DISSOCIATION. )                                     DXT02095
255 FORMAT(2X14H161. O2    + HV,9X10H= O     + O,10X1P2E10.2,2X14H162. NOXT02100
   10    + HV,9X10H= N     + O,10X1P2E10.2/2X14H163. NO2   + HV,9X10H= NODXT02105
   2    + O,10X1P2E10.2,2X14H164. N2O   + HV,9X10H= N2    + O,10X1P2E10.2DXT02110
   3/2X14H165. N2O   + HV,9X11H= N     + NO,9X1P2E10.2,2X14H166. O3    + DXT02115
   4HV,9X11H= O     + O2,9X1P2E10.2)                                     DXT02120
260 FORMAT(2X33H147. N     + NO    + M    = N2O   + M,10X1P2E10.2)       DXT02125
265 FORMAT(18HOSOURCE FUNCTION =1P1E12.5,20H  ION PAIRS/CC/SEC. )        DXT02130
270 FORMAT(6X9HNO    + HV,7X10H= NO+   + E,7X1P2E10.2,8X9HO2    + HV,7X10DXT02135
   1H= O2+   + E,7X1P2E10.2,2/6X9HO     + HV,7X10H= O+    + E,7X1P2E10.2,8XDXT02140
   29HN2    + HV,7X10H= N2+   + E,7X1P2E10.2 )                           DXT02145
275 FORMAT(5(2X2HR(,I3,2H)=1PE14.7))                                     DXT02150
280 FORMAT(3(2X2HR(,I3,2H)=1PE14.7))                                     DXT02155
285 FORMAT(2X35H148. NO    + O2    + NO   = NO2   + NO2,8X1P2E10.2)       DXT02160
290 FORMAT(1HO)                                                          DXT02165
295 FORMAT(1HO,41HRATES AT WHICH REACTIONS ARE PROCEEDING. )              DXT02170
    END                                                                  DXT02175
```

7.2.1.1 Input Parameters

CARD 1 No change.

CARD 2 In addition to EUBAR, ELBAR, and DEL there is a parameter B4 which is the maximum allowable percentage change in the production rate per step. Should the change in production over the next integration step attempt to exceed this amount, the main program will reduce the integrating increment until this requirement is satisfied. This control is necessary to prevent any difficulties that the code might experience in performing the integration, especially around sunrise and sunset when the rate of production of ionization is undergoing its most rapid change. The variable ENDT is replaced by the variable ENDHR. This parameter is set to the total number of local hours over which a solution is to be generated. These hours are counted continuously from the midnight preceeding the local hour at which the computations are started.

CARD 3 No change.

CARD 4 In addition to ALT, D, DO2, DN2, and T this card contains the following parameters:

(1) EAR = the earth's radius in centimeters.

(2) DECL = the solar declination in degrees.

(3) CLAT = the latitude in degrees.

(4) B6 = the noontime ionization rate of O_2 by Lyman beta.

(5) B7 = the noontime ionization rate of NO by Lyman alpha.

(6) B8 and B9 = the constants for the linear approximation to the photodissociation rate coefficient for O_2. This approximation is written in the form K = B8 + B9* (solar zenith angle). The linear approximation is determined by solving the integral

$$k(\chi) = N(O_2) \cdot \int_{\lambda_1}^{\lambda_2} \sigma_d(\lambda) \cdot \Phi_o(\lambda) \cdot e^{-\left(\sigma_a \cdot \int_z^{\infty} N(O_2)\, ds\right)_{d\lambda}}$$

for various values of χ. The constants are

120 km B8 = 4.2×10^{-5}
B9 = 2.88×10^{-5}
110 km B8 = 3.6×10^{-6}
B9 = 4.19×10^{-6} .

CARDS 5-7 The last value read on these cards (ATIME) is the local hour in seconds at which the solution is to start. ATIME = 0 is midnight.

CARD 8 No change.

The photoionization rate tables, as computed and punched externally by the photoionization program, are read following CARD 8.

7.2.1.2 Output

In the diurnal variation code two additional parameters appear in the output. These are the local solar time in hours and the value of the solar zenith angle at this time. Because of the inclusion of these parameters in the output, another auxilliary output tape is required. On tape 6 are written the time in seconds, the concentrations of electrons, O^-, O_2^-, O_3^-, NO_2^-, O^+, O_2^+, N_2^+, and the local time in hours. Tape 1 contains the time in seconds, the concentrations of NO^+, NO, N,

NO_2 , O_3 , N_2O , O, the value of the production function and the local time in hours. Tape 4 is the added auxilliary tape and contains the times in seconds, the concentrations of O_2 and N_2 , the settings of the KEY switches, the local time in hours, and the solar zenith angle in degrees.

7.2.1.3 The Solar Zenith Angle

The main program computes the angle of grazing incidence CIMAX which is the angle at the altitude in question between the zenith and the tangent to the earth.

$$\chi_{max} = \pi - \sin^{-1}\left(\frac{R}{R+A}\right)$$

where R is the radius of the earth and A is the altitude. The sun is assumed to subtend an angle of 0.5°. Therefore, sunset first contact occurs when

$$\chi = \chi_{max} - 0.25°$$

and last contact occurs when

$$\chi = \chi_{max} + 0.25° \quad .$$

The solar zenith angle χ is defined by

$$\cos\chi = \cos\delta\cos\phi\cos\theta + \sin\delta\sin\phi \quad .$$

From this the time in seconds corresponding to any zenith angle χ is

$$t = \cos^{-1}\left\{\frac{(\cos\chi - \sin\delta\sin\phi)}{\cos\delta\cos\phi}\right\} \cdot 1.3751\times10^4 \quad .$$

During the daytime portion of the solution the code keeps the rate coefficients for photodetachment and photodissociation at fixed constants except for the O_2 photodissociation which is computed according to the linear approximation previously discussed. During the nighttime portion of the solution, these rate constants are set to zero. During sunrise and sunset the rate coefficients for photdetachment and photodissociation are both increased or decreased according to the transmissivity percentage computed by the subprogram SUN.

The following statements, beginning on page 102, are a listing of the main program DIURN.

```
$IBFTC DIURN   LIST                                                    DIUR0000
C   DIURNAL VARIATION OF ATMOSPHERIC SPECIES AT ANY LATITUDE.          DIUR0005
C   SOLUTION OF THE REACTION RATE EQUATIONS IN THE IONOSPHERE FOR 15 SPEDIUR0010
C   IES AND WITH 168 REACTIONS.                                        DIUR0015
C     KB1=0 FOR LOG OUTPUT.                                            DIUR0020
C     KB1=1 FOR DECIMAL OUTPUT                                         DIUR0025
C     KB2=0 PRINTS HISTORY OF THE REACTIONS.                           DIUR0030
C     KB2=1 DOES NOT PRINT HISTORY OF THE REACTIONS.                   DIUR0035
C     KB3=0 PRINTS HISTORY AFTER EVERY SUCCESSFUL INTEGRATION.         DIUR0040
C     KB3=1 PRINTS HISTORY ONCE FOR EACH DECADE OF TIME.               DIUR0045
C     KB4=0 HALTS ON ERROR.                                            DIUR0050
C     KB4=1 READS ANOTHER POINT CARD AFTER AN ERROR.                   DIUR0055
C     IPLOT=0 DOES NOT MAKE A PLOTTER TAPE.                            DIUR0060
C     IPLOT=1 MAKES A LOG PLOTTER TAPE.                                DIUR0065
C     B4=ALLOWABLE PERCENTAGE CHANGE IN PRODUCTION PER STEP.           DIUR0070
C     B6=NOONTIME LYMAN BETA O2 IONIZATION)                            DIUR0075
C     B7=NOONTIME LYMAN ALPHA NO IONIZATION.                           DIUR0080
C     B8 AND B9 ARE THE CONSTANTS FOR O2 PHOTODISSOCIATION APPROXIMATIONDIUR0085
C     RATE COEFFICIENT=B8-B9*CXI                                       DIUR0090
C     UNITS OF INPUT PARAMETERS ARE CGS.                               DIUR0095
C                                                                      DIUR0100
      COMMON TREG(150),UN2(73), KEY(15),FORM(15),REMV(15),R(200),LKEY(15DIUR0105
     11),UO(73),  CON(200),BEGIN(17),LOCK(15),XO2(73),XO(73),XN2(73),XNO(DIUR0110
     273),XNE(73),ANGL(73),UO2(73),COB(10)                             DIUR0115
      COMMON NUMB,EUBAR,ELBAR,D,DO2,CIMAX,DN2,T, PNE,PNO,PO2,PO,PN2,TOTADIUR0120
     1L,JAKE,JAM,ITER,ALT,LAM,TIME,KB3,ATIME,EAR,DTIME,CXI,SIND,COSD,JUGUDIUR0125
     2,TIMEX,ITEM,PXO2,PXO,PXN2,TOTO,TOTN,B6,B7,DEL,LINT,JACK,I2NT,K2NT,DIUR0130
     3J2NT,N2NT,FIRST,IFAIL,JIP,B8,B9                                  DIUR0135
      DIMENSION DONT(20),TTREG(20),TITLE(12),HED(12),SIGM(30),IQ(30)   DIUR0140
      DIMENSION CRTNO(15),CRITN(15),A(168),B(168),C(168),G(168),ALF(7),ADIUR0145
     11(69),A2(69),A3(30)                                              DIUR0150
C                                                                      DIUR0155
C   THE FOLLOWING DATA ARE THE A,B,AND C'S FOR THE RATE CONSTANTS.     DIUR0160
C                                                                      DIUR0165
      DATA A1/ 0.44E+00,1.40E+00,0.04E+00,0.04E+00,9.00E-15,3.60E-16,  DIUR0170
     13.60E-16,1.00E-13,1.00E-13,1.00E-13,1.00E-13,1.00E-13,1.00E-13,  DIUR0175
     21.00E-13,1.00E-13,1.31E-15,1.00E-19,1.00E-17,1.00E-17,1.00E-11,  DIUR0180
     31.00E-11,1.00E-31,1.00E-31,1.40E-31,5.80E-33,1.90E-33,6.00E-28,  DIUR0185
     44.00E-29,0.00E-00,6.00E-05,9.00E-05,1.50E-04,1.00E-24,1.00E-22,  DIUR0190
     51.00E-22,1.00E-22,1.00E-23,2.20E-10,1.00E-12,1.00E-12,1.00E-12,  DIUR0195
     65.00E-07,5.00E-07,5.00E-07,5.00E-07,5.00E-07,5.00E-07,5.00E-07,  DIUR0200
     75.00E-07,5.00E-07,5.00E-07,5.00E-07,5.00E-07,5.00E-07,5.00E-07,  DIUR0205
     85.00E-07,5.00E-07,1.00E-23,1.00E-23,1.00E-23,1.00E-23,1.00E-23,  DIUR0210
     91.00E-23,1.00E-23,1.00E-23,1.00E-23,1.00E-23,1.00E-23,1.00E-11/  DIUR0215
      DATA A2/ 1.00E-13,1.00E-13,0.00E-00,1.00E-13,1.00E-13,1.00E-11,  DIUR0220
     11.00E-11,1.00E-13,0.00E-00,1.00E-13,1.00E-13,1.00E-13,1.00E-13,  DIUR0225
     24.00E-11,2.40E-11,0.00E-00,0.00E-00,8.00E-10,1.00E-12,1.00E-10,  DIUR0230
     30.00E-00,5.00E-12,1.00E-09,1.00E-09,1.00E-12,1.00E-09,1.00E-09,  DIUR0235
     41.00E-09,1.00E-09,1.00E-16,1.00E-18,0.00E-00,0.00E-00,0.00E-00,  DIUR0240
     50.00E-00,0.00E-00,1.00E-15,1.00E-29,1.00E-29,0.00E-00,0.00E-00,  DIUR0245
     60.00E-00,0.00E-00,0.00E-00,1.00E-30,1.00E-30,3.00E-12,3.00E-12,  DIUR0250
     73.00E-12,1.00E-11,1.80E-10,1.00E-11,2.50E-10,1.00E-11,1.00E-28,  DIUR0255
     81.00E-34,1.00E-17,1.00E-21,1.00E-21,2.00E-17,1.00E-24,6.40E-17,  DIUR0260
     91.00E-24,1.00E-22,1.00E-22,5.00E-32,2.00E-31,5.00E-32,3.20E-35/  DIUR0265
      DATA A3/ 2.60E-35,6.50E-34,2.00E-31,2.00E-33,3.00E-33,4.50E-33,  DIUR0270
     13.00E-30,1.00E-33,1.00E-33,0.00E-00,1.10E-10,7.10E-17,3.00E-11,  DIUR0275
     22.00E-10,5.00E-11,5.00E-10,3.00E-16,2.50E-11,2.00E-13,4.00E-12,  DIUR0280
     32.00E-11,8.00E-13,5.00E-06,6.00E-08,3.00E-03,4.08E-07,5.58E-08,  DIUR0285
```

```
     45.34E-03,1.00E-13,1.00E-13/                                          DIUR0290
      DATA B/4*0.0,3*2.0,16*0.0,3*0.5,3*0.0,3*-1.0,5*-1.5,-0.7,3*0.0,16*DIUR0295
     1-0.5,11*-1.5,55*0.0,-1.0,10*0.0,3*-0.5,3*0.0,-0.5,3*0.0,-1.0,4*0.0DIUR0300
     2,1.5,4*0.0,1.5,13*0.0/                                               DIUR0305
      DATA C/4*0.0,3*5.1E3,0.0,4.7E3,11*0.0,7.2E3,47*0.0,3E3,2*0.0,1E4,7DIUR0310
     1*0.0,5E3,37*0.0,2E3,3E3,3*0.0,3.5E3,0.0,5E3,4E3,11*0.0,2*-9E2,2*0.DIUR0315
     20,2E4,2*-9E2,0.0,3E3,1E4,0.0,3.75E4,1.9E4,5.3E2,1.4E4,1.35E4,2.8E3DIUR0320
     3,6.6E3,0.0,7E3,2*0.0,1.2E3,8*0.0/                                    DIUR0325
C                                                                          DIUR0330
C      INITIALIZATION OF SYSTEM AND INPUT.                                 DIUR0335
C                                                                          DIUR0340
      REWIND 0                                                             DIUR0345
      REWIND 1                                                             DIUR0350
      REWIND 3                                                             DIUR0355
      REWIND 4                                                             DIUR0360
      REWIND 8                                                             DIUR0365
      CHI=1.0E-10                                                          DIUR0370
      NOCOM=168                                                            DIUR0375
      ENDE=0.0                                                             DIUR0380
      ITER=30                                                              DIUR0385
      KB6=1                                                                DIUR0390
    5 WRITE(6,630)                                                         DIUR0395
      READ(5,675) (TITLE(N),N=1,12)                                        DIUR0400
      WRITE(6,680) (TITLE(N),N=1,12)                                       DIUR0405
      WRITE(3,760)(TITLE(N),N=1,12)                                        DIUR0410
      READ(5,625) EUBAR,ELBAR,DEL,B4,ENDHR                                 DIUR0415
      READ(5,640) NOC                                                      DIUR0420
      IF(NOC .EQ. 0) GO TO 30                                              DIUR0425
      WRITE(6,755)                                                         DIUR0430
      DO 25 J=1,NOC                                                        DIUR0435
      READ(5,725) I,G(I),B(I),C(I),(ALF(N),N=1,7)                          DIUR0440
      IF(I .GT. 69) GO TO 10                                               DIUR0445
      A1(I)=G(I)                                                           DIUR0450
      GO TO 20                                                             DIUR0455
   10 IF(I .GT. 138) GO TO 15                                              DIUR0460
      A2(I-69)=G(I)                                                        DIUR0465
      GO TO 20                                                             DIUR0470
   15 A3(I-138)=G(I)                                                       DIUR0475
   20 WRITE(6,725) I,G(I),B(I),C(I),(ALF(N),N=1,7)                         DIUR0480
   25 CONTINUE                                                             DIUR0485
   30 READ(5,625) ALT,D,DO2,DN2,T,EAR,DECL,CLAT,B6,B7,B8,B9                DIUR0490
C   ATIME IS TIME OF DAY IN SECONDS AT WHICH THE PROGRAM WILL START.       DIUR0495
C ATIME=0 IS MIDNIGHT.                                                     DIUR0500
      READ(5,625) (BEGIN(J),J=1,15),ATIME                                  DIUR0505
      READ(5,745) KB1,KB2,KB3,KB4,IPLOT                                    DIUR0510
      READ(5,740) (HED(J),J=1,12)                                          DIUR0515
      READ(5,620)(UO2(J),UO(J),UN2(J),XNO(J),XNE(J),ANGL(J),XO2(J),XO(J)DIUR0520
     1,XN2(J),SCRAY,J=1,73)                                                DIUR0525
      TIME=1.0E-6                                                          DIUR0530
      KOUNT=0                                                              DIUR0535
      MOUNT=0                                                              DIUR0540
      XNT=0                                                                DIUR0545
      JAKE=1                                                               DIUR0550
      JACK=1                                                               DIUR0555
      LAM=6                                                                DIUR0560
      KLOT=0                                                               DIUR0565
      ITEM=0                                                               DIUR0570
      JIP=1                                                                DIUR0575
      JUG=1                                                                DIUR0580
```

```
      DAY=1.0                                                          DIUR0585
      NDAY=0                                                           DIUR0590
      DTIME=ATIME                                                      DIUR0595
      IGO=1                                                            DIUR0600
      TIMT=4.32E4                                                      DIUR0605
      DO 35 J=1,150                                                    DIUR0610
   35 TREG(J)=0.0                                                      DIUR0615
      NUMB=15                                                          DIUR0620
      L1NT=((2*NUMB)+4)                                                DIUR0625
      I2NT=((3*NUMB)+4)                                                DIUR0630
      K2NT=((4*NUMB)+4)                                                DIUR0635
      J2NT=((5*NUMB)+4)                                                DIUR0640
      N2NT=((6*NUMB)+4)                                                DIUR0645
      DO 40 J=1,NUMB                                                   DIUR0650
      TTREG(J+3)=0.0                                                   DIUR0655
      LOCK(J)=0                                                        DIUR0660
   40 KEY(J)=1                                                         DIUR0665
      DECO=DECL*0.01745329                                             DIUR0670
      CLAO=CLAT*0.01745329                                             DIUR0675
      COSD=COS(DECO)*COS(CLAO)                                         DIUR0680
      SIND=SIN(DECO)*SIN(CLAO)                                         DIUR0685
      CIMAX=3.1415926-ARSIN(EAR/(EAR+ALT))                             DIUR0690
      CIMAD=57.295779*CIMAX                                            DIUR0695
      CIMID=ARCOS(SIND-COSD)                                           DIUR0700
      CIMAP=CIMAX-4.3633231E-3                                         DIUR0705
      CIMAT=CIMAX+4.3633231E-03                                        DIUR0710
      IF(CIMAP .GT. CIMID) GO TO 45                                    DIUR0715
      IF(CIMAT .LT. CIMID) GO TO 50                                    DIUR0720
      KSKP=3                                                           DIUR0725
      GO TO 55                                                         DIUR0730
   45 KSKP=1                                                           DIUR0735
      GO TO 65                                                         DIUR0740
   50 KSKP=2                                                           DIUR0745
C TIM IS SECONDS FROM ZENITH TO CIMAP (NOON TO SUNSET FIRST CONTACT).  DIUR0750
   55 TIM=ARCOS((COS(CIMAP)-SIND)/COSD)*1.3750987E4                    DIUR0755
      IF(KSKP .EQ. 3) GO TO 60                                         DIUR0760
      TOM=ARCOS((COS(CIMAT)-SIND)/COSD)*1.3750987E4                    DIUR0765
      DARK=8.64E4-2.0*TOM                                              DIUR0770
   60 SILLY=TIM+4.32E4                                                 DIUR0775
   65 CALL INITAL                                                      DIUR0780
C     COMPUTE RATE CONSTANTS IN THE FORM K=A*(T**B)*EXP(-C/T)          DIUR0785
C                                                                      DIUR0790
      K=1                                                              DIUR0795
      DO 70 J=1,69                                                     DIUR0800
      A(K)=A1(J)                                                       DIUR0805
   70 K=K+1                                                            DIUR0810
      DO 75 J=1,69                                                     DIUR0815
      A(K)=A2(J)                                                       DIUR0820
   75 K=K+1                                                            DIUR0825
      DO 80 J=1,30                                                     DIUR0830
      A(K)=A3(J)                                                       DIUR0835
   80 K=K+1                                                            DIUR0840
      DO 85 J=1,NOCON                                                  DIUR0845
   85 CON(J)=A(J)*(T**B(J))*EXP(-C(J)/T)                               DIUR0850
C                                                                      DIUR0855
      DO 90 J=1,4                                                      DIUR0860
   90 COB(J)=CON(J)                                                    DIUR0865
      M=161                                                            DIUR0870
      DO 95 J=5,10                                                     DIUR0875
```

```
      COB(J)=CON(M)                                                    DIUR0880
   95 M=M+1                                                            DIUR0885
C     PREPARE OUTPUT TAPES AND WRITE RATE CONSTANTS AND INITIAL        DIUR0890
C     CONDITIONS.                                                      DIUR0895
C                                                                      DIUR0900
      WRITE(6,635) ALT,D,T                                             DIUR0905
      WRITE(6,650) DECL,CLAT,CIMAD                                     DIUR0910
      WRITE(6,700)                                                     DIUR0915
      IS=1                                                             DIUR0920
      DO 100 J=1,15                                                    DIUR0925
      IT=IS+5                                                          DIUR0930
      WRITE(6,685) (I,A(I),I=IS,IT)                                    DIUR0935
      WRITE(6,705) (I,CON(I),I=IS,IT)                                  DIUR0940
      IS=IS+6                                                          DIUR0945
  100 CONTINUE                                                         DIUR0950
      WRITE(6,735)                                                     DIUR0955
      DO 105 J=1,13                                                    DIUR0960
      IT=IS+5                                                          DIUR0965
      WRITE(6,685) (I,A(I),I=IS,IT)                                    DIUR0970
      WRITE(6,705) (I,CON(I),I=IS,IT)                                  DIUR0975
      IS=IS+6                                                          DIUR0980
  105 CONTINUE                                                         DIUR0985
      WRITE(6,645)                                                     DIUR0990
      ATIM=ATIME/3600.0                                                DIUR0995
      IF(KB1 .EQ. 0) GO TO 110                                         DIUR1000
      WRITE(6,730) TREG(2),(TREG(J),J=4,11),ATIM                       DIUR1005
      WRITE(1) TREG(2),(TREG(J),J=12,18),TOTAL,ATIM                    DIUR1010
      MOUNT=MOUNT+1                                                    DIUR1015
      GO TO 145                                                        DIUR1020
  110 M=NUMB+3                                                         DIUR1025
      DO 125 J=2,M                                                     DIUR1030
      DEC=TREG(J)                                                      DIUR1035
      IF(DEC) 580,115,120                                              DIUR1040
  115 DONT(J)=0.0                                                      DIUR1045
      GO TO 125                                                        DIUR1050
  120 DONT(J)=ALOG10(DEC)                                              DIUR1055
  125 CONTINUE                                                         DIUR1060
      IF(TOTAL) 580,130,135                                            DIUR1065
  130 DPROD=0.0                                                        DIUR1070
      GO TO 140                                                        DIUR1075
  135 DPROD=ALOG10(TOTAL)                                              DIUR1080
  140 WRITE(6,730) DONT(2),(DONT(J),J=4,11),ATIM                       DIUR1085
      WRITE(1) DONT(2),(DONT(J),J=12,18),DPROD,ATIM                    DIUR1090
      MOUNT=MOUNT+1                                                    DIUR1095
  145 IF(TREG(3) .LT. 1.8E3) GO TO 150                                 DIUR1100
      TREG(3)=9.0E2                                                    DIUR1105
  150 GO TO (155,160,235),KSKP                                         DIUR1110
  155 TESS=8.46E4*DAY                                                  DIUR1115
      IF(ATIME .LT. TESS) GO TO 265                                    DIUR1120
      DAY=DAY+1.0                                                      DIUR1125
      NDAY=NDAY+1                                                      DIUR1130
      GO TO 265                                                        DIUR1135
  160 TESS=ATIME+TREG(3)                                               DIUR1140
      IF(TESS-SILLY) 265,180,165                                       DIUR1145
  165 GO TO (170,185,185,205,215,215),JIP                              DIUR1150
  170 TREG(3)=(SILLY-ATIME)                                            DIUR1155
  175 JIP=2                                                            DIUR1160
      GO TO 265                                                        DIUR1165
  180 GO TO (175,175,210,210),JIP                                      DIUR1170
```

```
  185 CXI=ARCOS(COSD*COS(7.2722052E-5*(ATIME+4.32E4))+SIND)                DIUR1175
      IF(CXI .GT. CIMAT) GO TO 190                                        DIUR1180
      IF(TREG(3) .LT. 10.0) GO TO 265                                     DIUR1185
      TREG(3)=10.0                                                        DIUR1190
      JIP=3                                                               DIUR1195
      GO TO 265                                                           DIUR1200
  190 SILLY=SILLY+DARK                                                    DIUR1205
      DO 195 J=1,4                                                        DIUR1210
  195 CON(J)=0.0                                                          DIUR1215
      DO 200 J=161,166                                                    DIUR1220
  200 CON(J)=0.0                                                          DIUR1225
      JIP=4                                                               DIUR1230
      NDAY=NDAY+1                                                         DIUR1235
      GO TO 265                                                           DIUR1240
  205 TREG(3)=(SILLY-ATIME)                                               DIUR1245
  210 JIP=5                                                               DIUR1250
      GO TO 265                                                           DIUR1255
  215 CXI=ARCOS(COSD*COS(7.2722052E-5*(ATIME+4.32E4))+SIND)               DIUR1260
      IF(CXI .LT. CIMAP) GO TO 220                                        DIUR1265
      IF(TREG(3) .LT. 10.0) GO TO 265                                     DIUR1270
      TREG(3)=10.0                                                        DIUR1275
      JIP=6                                                               DIUR1280
      GO TO 265                                                           DIUR1285
  220 SILLY=SILLY+(8.46E4-DARK)                                           DIUR1290
      JIP=1                                                               DIUR1295
      DO 225 J=1,4                                                        DIUR1300
  225 CON(J)=A(J)*(T**B(J))*EXP(-C(J)/T)                                  DIUR1305
      DO 230 J=161,166                                                    DIUR1310
  230 CON(J)=A(J)*(T**B(J))*EXP(-C(J)/T)                                  DIUR1315
      GO TO 265                                                           DIUR1320
  235 TESS=ATIME+TREG(3)                                                  DIUR1325
      IF(TESS-SILLY) 265,245,240                                          DIUR1330
  240 GO TO (245,255,250),JIP                                             DIUR1335
  245 TREG(3)=(SILLY-ATIME)                                               DIUR1340
      JIP=2                                                               DIUR1345
      GO TO 265                                                           DIUR1350
  250 CXI=ARCOS(COSD*COS(7.2722052E-5*(ATIME+4.32E4))+SIND)               DIUR1355
      IF(CXI .LT. CIMAP) GO TO 260                                        DIUR1360
      IF(TREG(3) .LT. 10.0) GO TO 265                                     DIUR1365
  255 TREG(3)=10.0                                                        DIUR1370
      JIP=3                                                               DIUR1375
      GO TO 265                                                           DIUR1380
  260 SILLY=SILLY+8.46E4                                                  DIUR1385
      JIP=1                                                               DIUR1390
  265 DO 270 K=1,NUMB                                                     DIUR1395
      IF(KEY(K)-2) 275,270,270                                            DIUR1400
  270 CONTINUE                                                            DIUR1405
      CALL ALGA                                                           DIUR1410
      TREG(3)=2.0*TREG(3)                                                 DIUR1415
      TREG(2)=TREG(2)+TREG(3)                                             DIUR1420
      GO TO 300                                                           DIUR1425
C                                                                         DIUR1430
C     INTEGRATION  OF EQUATIONS STARTS HERE.                              DIUR1435
C                                                                         DIUR1440
  275 CALL INTEG                                                          DIUR1445
      GO TO(300,280,300,300,290,300),JIP                                  DIUR1450
  280 IF(IFAIL) 285,300,285                                               DIUR1455
  285 JIP=1                                                               DIUR1460
      GO TO 300                                                           DIUR1465
```

```
290 IF(IFAIL) 295,300,295                                          DIUR1470
295 JIP=4                                                          DIUR1475
300 CALL BALAN                                                     DIUR1480
    ATIME=ATIME+(TREG(3)/2.0)                                      DIUR1485
    CHID=57.295779*CXI                                             DIUR1490
    IF(TREG(3) .NE. 2.0E-6) GO TO 305                              DIUR1495
    KLOT=KLOT+1                                                    DIUR1500
    IF(KLOT .GT. 10) GO TO 310                                     DIUR1505
305 KLOT=0                                                         DIUR1510
    ATIM=ATIME/3600.0                                              DIUR1515
    GO TO 315                                                      DIUR1520
310 WRITE(6,690)                                                   DIUR1525
    GO TO 585                                                      DIUR1530
315 IF(KBI .NE. 0) GO TO 320                                       DIUR1535
    DDO2=ALOG10(DO2)                                               DIUR1540
    DDN2=ALOG10(DN2)                                               DIUR1545
    WRITE(4) DONT(2),DDO2,DDN2,(KEY(J),J=1,NUMB),ATIM,CHID         DIUR1550
    GO TO 325                                                      DIUR1555
320 WRITE(4) TREG(2),DO2,DN2,(KEY(J),J=1,NUMB),ATIM,CHID           DIUR1560
325 KOUNT=KOUNT+1                                                  DIUR1565
    DO 330 J=1,NUMB                                                DIUR1570
330 LKEY(J)=KEY(J)                                                 DIUR1575
    K=JAKE                                                         DIUR1580
    GO TO (335,570,575),K                                          DIUR1585
335 JACK=2                                                         DIUR1590
    KIND=1                                                         DIUR1595
    CALL SLOP(KIND)                                                DIUR1600
    KIND=2                                                         DIUR1605
    CALL SLOP(KIND)                                                DIUR1610
    N2=LINT                                                        DIUR1615
    DO 340 I=1,NUMB                                                DIUR1620
    CRITN(I)=2.0*ABS((TREG(N2)-TREG(I+3))/TREG(3))                 DIUR1625
    CRTNO(I)=REMV(I)*TREG(I+3)                                     DIUR1630
    N2=N2+1                                                        DIUR1635
340 CONTINUE                                                       DIUR1640
    CRTNO(10)=CRTNO(10)+PNO                                        DIUR1645
    CRTNO(15)=CRTNO(15)+PO                                         DIUR1650
    DO 360 J=1,NUMB                                                DIUR1655
    IF(CRITN(J) .GT. 1.0E-3) GO TO 345                             DIUR1660
    IF(ABS(1.0-(FORM(J)/CRTNO(J))) .GT. DEL) GO TO 350             DIUR1665
345 IF(CRTNO(J) .EQ. 0.0) GO TO 360                                DIUR1670
    IF((CRITN(J)/CRTNO(J)) .LT. DEL) GO TO 355                     DIUR1675
350 LOCK(J)=0                                                      DIUR1680
    GO TO 360                                                      DIUR1685
355 LOCK(J)=LOCK(J)+1                                              DIUR1690
360 CONTINUE                                                       DIUR1695
    DO 380 J=1,NUMB                                                DIUR1700
    IF(KEY(J)-3) 365,380,380                                       DIUR1705
365 IF(LOCK(J)-3) 370,375,375                                      DIUR1710
370 KEY(J)=1                                                       DIUR1715
    GO TO 380                                                      DIUR1720
375 KEY(J)=2                                                       DIUR1725
380 CONTINUE                                                       DIUR1730
    JACK=2                                                         DIUR1735
    DO 385 J=1,NUMB                                                DIUR1740
    IF(KEY(J) .GT. 2) GO TO 385                                    DIUR1745
    IF(TREG(J+3) .GT. TTREG(J+3)) GO TO 385                        DIUR1750
    IF(TREG(J+3) .GT. CHI) GO TO 385                               DIUR1755
    KEY(J)=3                                                       DIUR1760
```

```
      TREG(J+3)=0.0                                                    DIUR1765
  385 CONTINUE                                                         DIUR1770
      IF(PNE) 405,390,405                                              DIUR1775
  390 IF((TREG(4)/BEGIN(1))-1.0E-3) 395,405,405                        DIUR1780
  395 DO 400 J=1,3                                                     DIUR1785
      TREG(J+3)=0.0                                                    DIUR1790
  400 KEY(J)=3                                                         DIUR1795
  405 DO 410 J=1,NUMB                                                  DIUR1800
  410 TTREG(J+3)=TREG(J+3)                                             DIUR1805
                                                                       DIUR1810
      OUTPUT OF RESULTS STARTS HERE.                                   DIUR1815
                                                                       DIUR1820
      IF(KNT-50) 420,415,415                                           DIUR1825
  415 WRITE(6,645)                                                     DIUR1830
      KNT=0                                                            DIUR1835
  420 KNT=KNT+1                                                        DIUR1840
      ATIM=ATIME/3600.0                                                DIUR1845
      IF(KB1 .EQ. 0) GO TO 425                                         DIUR1850
      WRITE(6,730) TREG(2),(TREG(J),J=4,11),ATIM                       DIUR1855
      WRITE(1) TREG(2),(TREG(J),J=12,18),TOTAL,ATIM                    DIUR1860
      GO TO 460                                                        DIUR1865
  425 K=NUMB+3                                                         DIUR1870
      DO 440 J=2,K                                                     DIUR1875
      DEC=TREG(J)                                                      DIUR1880
      IF(DEC) 580,430,4                                                DIUR1885
  430 DONT(J)=0.0                                                      DIUR1890
      GO TO 440                                                        DIUR1895
  435 DONT(J)=ALOG10(DEC)                                              DIUR1900
  440 CONTINUE                                                         DIUR1905
      IF(TOTAL) 580,445,450                                            DIUR1910
  445 TOTL=0.0                                                         DIUR1915
      GO TO 455                                                        DIUR1920
  450 TOTL=ALOG10(TOTAL)                                               DIUR1925
  455 WRITE(6,730)     DONT(2),(DONT(J),J=4,11),ATIM                   DIUR1930
      WRITE(1)     DONT(2),(DONT(J),J=12,18),TOTL,ATIM                 DIUR1935
  460 CALL PLOT(IPLOT)                                                 DIUR1940
      MOUNT=MOUNT+1                                                    DIUR1945
  465 IF(ATIME-TIMT) 475,470,470                                       DIUR1950
  470 TIMT=TIMT+4.32E4                                                 DIUR1955
      CALL DAUXT                                                       DIUR1960
                                                                       DIUR1965
      DECISION TO CONTINUE INTEGRATION OR STOP IS MADE HERE.           DIUR1970
                                                                       DIUR1975
  475 IF(TREG(2) .LT. 1.0) GO TO 500                                   DIUR1980
      GO TO (480,485),IGO                                              DIUR1985
  480 IGO=2                                                            DIUR1990
      SAVE=TREG(2)                                                     DIUR1995
      ICNT=1                                                           DIUR2000
      GO TO 500                                                        DIUR2005
  485 IF(ICNT .GT. 50) GO TO 490                                       DIUR2010
      ICNT=ICNT+1                                                      DIUR2015
      GO TO 500                                                        DIUR2020
  490 IF((TREG(2)-SAVE) .GT. 1.0) GO TO 495                            DIUR2025
      WRITE(6,710)                                                     DIUR2030
      GO TO 585                                                        DIUR2035
  495 IGO=1                                                            DIUR2040
  500 IF(TREG(4) .LT. ENDE) GO TO 585                                  DIUR2045
  505 IF(ATIM .GT. ENDHR) GO TO 585                                    DIUR2050
                                                                       DIUR2055
```

```
C  TEST SYSTEM CLOCK FOR IMMINENT TIMER OVERFLOW.                    DIUR2060
C                                                                   DIUR2065
  510 CALL CLOCK                                                    DIUR2070
      CALL SLITET(4,K000FX)                                         DIUR2075
      GO TO (515,520),K000FX                                        DIUR2080
  515 ILK=2                                                         DIUR2085
      GO TO 585                                                     DIUR2090
  520 BIG=AMAX1(TREG(4),TREG(5),TREG(6),TREG(7),TREG(8))            DIUR2095
      DO 525 J=1,5                                                  DIUR2100
      IF(BIG .EQ. TREG(J+3)) GO TO 530                              DIUR2105
  525 CONTINUE                                                      DIUR2110
  530 IF(J-LAM) 535,540,535                                         DIUR2115
  535 KEY(LAM)=1                                                    DIUR2120
      JACK=1                                                        DIUR2125
  540 LAM=J                                                         DIUR2130
      KEY(LAM)=4                                                    DIUR2135
      STOT=TOTAL                                                    DIUR2140
  545 TREG(2)=TREG(2)+TREG(3)                                       DIUR2145
      CALL PRODUC                                                   DIUR2150
      TREG(2)=TREG(2)-TREG(3)                                       DIUR2155
      IF(TOTAL-STOT) 550,145,555                                    DIUR2160
  550 RATIO=STOT/TOTAL                                              DIUR2165
      GO TO 560                                                     DIUR2170
  555 RATIO=TOTAL/STOT                                              DIUR2175
  560 IF(RATIO-B4) 145,145,565                                      DIUR2180
  565 TREG(3)=TREG(3)/2.0                                           DIUR2185
      GO TO 545                                                     DIUR2190
C                                                                   DIUR2195
C     ERROR COMMENT OUTPUTS.                                        DIUR2200
C                                                                   DIUR2205
  570 WRITE(6,660) TREG(2)                                          DIUR2210
      GO TO 585                                                     DIUR2215
  575 WRITE(6,665) TREG(2)                                          DIUR2220
      GO TO 585                                                     DIUR2225
  580 WRITE(6,655)                                                  DIUR2230
      K=NUMB+3                                                      DIUR2235
      WRITE(6,670) (TREG(J),J=2,K),TOTAL                            DIUR2240
  585 IF(KB4 .EQ. 1) GO TO 590                                      DIUR2245
      KB6=2                                                         DIUR2250
C                                                                   DIUR2255
C     TRANSFER ALL RESULTS TO OUTPUT TAPE HERE.                     DIUR2260
C                                                                   DIUR2265
  590 REWIND 1                                                      DIUR2270
      END FILE 3                                                    DIUR2275
      IF(IPLOT .NE. 1) GO TO 600                                    DIUR2280
      END FILE 0                                                    DIUR2285
      REWIND 8                                                      DIUR2290
      DO 595 K=1,ITEM                                               DIUR2295
      READ(8)SIGM(1),IQ(1),(SIGM(J),IQ(J),J=10,18)                  DIUR2300
      WRITE(0,750)SIGM(1),IQ(1),(SIGM(J),IQ(J),J=10,18)             DIUR2305
  595 CONTINUE                                                      DIUR2310
      END FILE 0                                                    DIUR2315
      REWIND 8                                                      DIUR2320
  600 KNT=0                                                         DIUR2325
      WRITE(6,695)                                                  DIUR2330
      DO 605 K=1,MOUNT                                              DIUR2335
      READ(1)  TREG(2),(TREG(J),J=12,18),TOTAL,ATIM                 DIUR2340
      WRITE(6,730) TREG(2),(TREG(J),J=12,18),TOTAL,ATIM             DIUR2345
      KNT=KNT+1                                                     DIUR2350
```

```
      IF(KNT .LT. 50) GO TO 605                                          DIUR2355
      WRITE(6,695)                                                       DIUR2360
      KNT=0                                                              DIUR2365
  605 CONTINUE                                                           DIUR2370
      REWIND 1                                                           DIUR2375
      REWIND 4                                                           DIUR2380
      KNT=0                                                              DIUR2385
      WRITE(6,720)                                                       DIUR2390
      DO 610 K=1,KOUNT                                                   DIUR2395
      READ(4) (REG(J),DO2,DN2,(KEY(J),J=1,NUMB),ATIM,CHID               DIUR2400
      WRITE(6,715) (REG(J),DO2,DN2,(KEY(J),J=1,NUMB),ATIM,CHID          DIUR2405
      KNT=KNT+1                                                          DIUR2410
      IF(KNT .LT. 50) GO TO 610                                          DIUR2415
      WRITE(6,720)                                                       DIUR2420
      KNT=0                                                              DIUR2425
  610 CONTINUE                                                           DIUR2430
      REWIND 4                                                           DIUR2435
      GO TO (5,615),KB6                                                  DIUR2440
  615 REWIND 3                                                           DIUR2445
      REWIND 0                                                           DIUR2450
      CALL EXIT                                                          DIUR2455
C                                                                        DIUR2460
  620 FORMAT(1P5E14.7)                                                   DIUR2465
  625 FORMAT (1P6E12.5)                                                  DIUR2470
  630 FORMAT(94H1SOLUTION OF THE REACTION RATE EQUATIONS WITH 15 SPECIESDIUR2475
     1 AND 168 REACTIONS DIURNAL VARIATION. )                            DIUR2480
  635 FORMAT(11H1ALTITUDE =1PE11.4,4H CM.,17H  TOTAL DENSITY =1PE12.5,15DIUR2485
     1H  TEMPERATURE =0PF7.2)                                            DIUR2490
  640 FORMAT(I4)                                                         DIUR2495
  645 FORMAT(131H1 TIME (SEC)     N(E) /CC    N(O-) /CC    N(O2-) /CC  NDIUR2500
     1(O3-) /CC    N(NO2-) /CC   N(O+) /CC    N(O2+) /CC   N(N2+) /CC  TIDIUR2505
     2ME(HOUR)  )                                                        DIUR2510
  650 FORMAT(20H0SOLAR DECLINATION =1PE13.5,10H DEGREES. ,6X10HLATITUDE DIUR2515
     1=0PF6.2,10H DEGREES. ,6X20HANGLE OF GRAZING INCIDENCE =1PE13.5,10HDIUR2520
     2 DEGREES. )                                                        DIUR2525
  655 FORMAT(65H0THE PROGRAM IN TRYING TO GENERATE THE LOG OF A NEGATIVEDIUR2530
     1 NUMBER. )                                                         DIUR2535
  660 FORMAT(45H0THE INTERGRATING MESH IS VANISHING IN INT AT 1PE11.5,6HDIUR2540
     1 SEC. )                                                            DIUR2545
  665 FORMAT(47H0THE INTERGRATING MESH IS VANISHING IN ALGA AT 1PE11.5,6DIUR2550
     1H SEC. )                                                           DIUR2555
  670 FORMAT(1P10E10.2)                                                  DIUR2560
  675 FORMAT(12A6)                                                       DIUR2565
  680 FORMAT(1H0,12A6)                                                   DIUR2570
  685 FORMAT(1H0,(6(2X4H  A(,I3,2H)=1PE10.3)))                           DIUR2575
  690 FORMAT(43H THE INCREMENT IS CONSTANT AT 1.0E-06 SEC. )             DIUR2580
  695 FORMAT(130H1 TIME (SEC)    N(NO+)/CC    N(NO)/CC       N(N) /CC   DIUR2585
     1N(NO2)/CC     N(O3) /CC    N(N2O) /CC    N(O) /CC   PRODUCTION  TIMDIUR2590
     2E(HOUR)  )                                                         DIUR2595
  700 FORMAT(22H0REACTION COEFFICIENTS)                                  DIUR2600
  705 FORMAT(1H ,(6(2X4HCON(,I3,2H)=1PE10.3)))                           DIUR2605
  710 FORMAT(60H TIME IS NOT INCREASING RAPIDLY ENOUGH TO ADVANCE SOLUTIDIUR2610
     1ON. )                                                              DIUR2615
  715 FORMAT(1P3E13.5,15I4,1PE13.5,0PF11.5)                              DIUR2620
  720 FORMAT(121H1 TIME (SEC)   O2 DENSITY   N2 DENSITY    E   O- O2- O3DIUR2625
     1- NO2- O+ O2+ N2+ NO+  NO  N  NO2  O3 N2O  O  TIME(HOUR)      CHI DIUR2630
     2)                                                                  DIUR2635
  725 FORMAT(I4,1PE10.2,0PF5.1,1PE10.2,7A6)                              DIUR2640
  730 FORMAT(1P10E13.5)                                                  DIUR2645
  735 FORMAT(1H1)                                                        DIUR2650
  740 FORMAT(12A6)                                                       DIUR2655
  745 FORMAT(5I2)                                                        DIUR2660
  750 FORMAT(6X11(A1,I5))                                                DIUR2665
  755 FORMAT(1H0,20X,86HTHE FOLLOWING IS A LIST OF RATE CONSTANT CHANGESDIUR2670
     1 FROM THE STND LIST USED IN THIS RUN. )                            DIUR2675
  760 FORMAT(1H1,12A6)                                                   DIUR2680
      END                                                                DIUR2685
```

7.2.2 THE PHOTOIONIZATION PRODUCTION SUBROUTINE PRODUC

Upon being called by SLOP or the main program, PRODUC first computes the solar zenith angle for the time at which it was called. Using this zenith angle, the subroutine performs a table look-up in the photoionization rate tables read into the computer by the main program. If the computed zenith angle is not a tabular value, linear interpolations are performed in the tables to obtain the proper production rates for the positive ions.

One-tenth of one percent of the noontime production rate of O_2^+ by L_β is automatically added to the computed production rate of O_2^+. One percent of the noontime production rate of NO^+ by L_α is always added to the production rate of NO^+. These production rates are added in order to allow a certain amount of these radiations to scatter into the nighttime atmosphere.

After the production rates for all the positive ions are computed, they are summed to obtain the production rate of the electrons.

The following statements are a listing of this subroutine.

```
$IBFTC PRODUC  LIST                                                    PROD0000
      SUBROUTINE PRODUC                                                PROD0005
C THIS SUBROUTINE COMPUTES THE PRODUCTION FROM THE Q TABLES.           PROD0010
C                                                                      PROD0015
      COMMON TREG(150),UN2(73), KEY(15),FORM(15),REMV(15),R(200),LKEY(15PROD0020
     1),UO(73),  CON(200),BEGIN(17),LOCK(15),UO2(73),UO(73),UN2(73),UNO(PROD0025
     273),UNE(73),ANGL(73),U02(73),COB(10)                             PROD0030
      COMMON NUMB,EUBAR,ELBAR,D,DO2,CIMAX,DN2,T, PNE,PNO,PO2,PO,PN2,TOTAPROD0035
     1L,JAKE,JAM,ITER,ALT,LAM,TIME,KB3,ATIME,EAR,DTIME,CXI,SIND,COSD,JUGPROD0040
     2,TIMEX,ITEM,PXO2,PXO,PXN2,TOTO,TOTN,B6,B7,DEL,LINT,JACK,I2NT,K2NT,PROD0045
     3J2NT,N2NT,FIRST,IFAIL,JIP,B8,B9                                  PROD0050
      TIMEX=DTIME +TREG(2)                                             PROD0055
      CXI=ARCOS(COSD*COS(7.2722052E-5*(TIMEX+4.32E4))+SIND)            PROD0060
      DO 5 J=1,73                                                      PROD0065
      IF(CXI-ANGL(J)) 15,10,5                                          PROD0070
    5 CONTINUE                                                         PROD0075
   10 PO2=UO2(J)                                                       PROD0080
      PO=UO(J)                                                         PROD0085
      PN2=UN2(J)                                                       PROD0090
      PXO2=XO2(J)                                                      PROD0095
      PXO=XO(J)                                                        PROD0100
      PXN2=XN2(J)                                                      PROD0105
      PNO=XNO(J)*TREG(13)                                              PROD0110
      GO TO 20                                                         PROD0115
   15 PART=(ANGL(J-1)-CXI)/(ANGL(J-1)-ANGL(J))                         PROD0120
      PO2=UO2(J-1)-PART*(UO2(J-1)-UO2(J))                              PROD0125
      PO=UO(J-1)-PART*(UO(J-1)-UO(J))                                  PROD0130
      PN2=UN2(J-1)-PART*(UN2(J-1)-UN2(J))                              PROD0135
      PXO2=XO2(J-1)-PART*(XO2(J-1)-XO2(J))                             PROD0140
      PXO=XO(J-1)-PART*(XO(J-1)-XO(J))                                 PROD0145
      PXN2=XN2(J-1)-PART*(XN2(J-1)-XN2(J))                             PROD0150
      PNO=XNO(J-1)-PART*(XNO(J-1)-XNO(J))                              PROD0155
      PNO=PNO*TREG(13)                                                 PROD0160
   20 QT=(1.0E-16*D)/(DO2+DN2)                                         PROD0165
      PNO=PNO+B7*1.0E-02                                               PROD0170
      PO2=PO2+QT*DO2+PXO2+1.0E-03*B6                                   PROD0175
      PN2=PN2+QT*DN2+PXN2                                              PROD0180
      PO=PO+PXO                                                        PROD0185
      PNE=PO2+PO+PN2+PNO                                               PROD0190
      TOTAL=PNE                                                        PROD0195
      RETURN                                                           PROD0200
      END                                                              PROD0205
```

7.2.3 THE CHARGE BALANCE SUBROUTINE BALAN

This subroutine computes the largest ... e species from charge balance and adjusts the O_2 and N_2 concentrations to ... ure conservation of O and N atoms. In addition to this it also calls subrouti... SUN if the solution has advanced into a sunrise or sunset period. Subroutine SUN returns with a transmissivity percentage by which subroutine BALAN multiplies the photodetachment and photodissociation rate coefficients. The solution has arrived at a period of sunrise or sunset if switch JIP is on three or six respectively. Switch JIP is set by the main program.

The following statements are a listing of this subprogram.

```
$IBFTC BALAN   LIST                                                    BALN0000
       SUBROUTINE BALAN                                                BALN0005
       COMMON TREG(150),UN2(73), KEY(15),FORM(15),REMV(15),R(200),LKEY(15BALN0010
      1),UO(73),  CON(200),BEGIN(17),LOCK(15),XO2(73),XO(73),XN2(73),XNO(BALN0015
      273),XNE(73),ANGL(73),UO2(73),COB(10)                            BALN0020
       COMMON NUMB,EUBAR,ELBAR,D,DO2,CIMAX,DN2,T, PNE,PNO,PO2,PO,PN2,TOTABALN0025
      1L,JAKE,JAM,ITER,ALT,LAM,TIME,KB3,ATIME,EAR,DTIME,CXI,SIND,COSD,JUGBALN0030
      2,TIMEX,ITEM,PXO2,PXO,PXN2,TOTO,TOTN,B6,B7,DEL,LINT,JACK,I2NT,K2NT,BALN0035
      3J2NT,N2NT,FIRST,IFAIL,JIP,B8,B9                                 BALN0040
     5 SUM=TREG(9)+TREG(10)+TREG(11)+TREG(12)                          BALN0045
       GO TO (10,15,20,25,30,35),LAM                                   BALN0050
    10 TREG(4)=SUM-TREG(5)-TREG(6)-TREG(7)-TREG(8)                     BALN0055
       GO TO 35                                                        BALN0060
    15 TREG(5)=SUM-TREG(4)-TREG(6)-TREG(7)-TREG(8)                     BALN0065
       GO TO 35                                                        BALN0070
    20 TREG(6)=SUM-TREG(4)-TREG(5)-TREG(7)-TREG(8)                     BALN0075
       GO TO 35                                                        BALN0080
    25 TREG(7)=SUM-TREG(4)-TREG(5)-TREG(6)-TREG(8)                     BALN0085
       GO TO 35                                                        BALN0090
    30 TREG(8)=SUM-TREG(4)-TREG(5)-TREG(6)-TREG(7)                     BALN0095
    35 DO2=((TOTO-TREG(5)-3.0*TREG(7)-TREG(9)-TREG(12)-3.0*TREG(15)-TREG(BALN0100
      116)-TREG(17)-TREG(18))/2.0)-TREG(6)-TREG(8)-TREG(10)-TREG(14)   BALN0105
       DN2=((TOTN-TREG(8)-TREG(12)-TREG(13)-TREG(14)-TREG(16))/2.0)-TREG(BALN0110
      111)-TREG(17)                                                    BALN0115
       TIMEX=DTIME +TREG(2)                                            BALN0120
       CXI=ARCOS(COSD*COS(7.2722052E-5*(TIMEX+4.32E4))+SIND)           BALN0125
       COB(5)=B8-B9*CXI                                                BALN0130
       IF(COB(5)) 40,45,45                                             BALN0135
    40 COB(5)=0.0                                                      BALN0140
    45 GO TO (65,65,50,65,65,50),JIP                                   BALN0145
    50 FRACT=SUN(CXI,CIMAX)                                            BALN0150
       AT=FRACT                                                        BALN0155
       DO 55 J=1,4                                                     BALN0160
    55 CON(J)=COB(J)*FRACT                                             BALN0165
       N=161                                                           BALN0170
       DO 60 J=5,10                                                    BALN0175
       CON(N)=COB(J)*FRACT                                             BALN0180
    60 N=N+1                                                           BALN0185
       GO TO 70                                                        BALN0190
    65 CON(161)=COB(5)                                                 BALN0195
    70 RETURN                                                          BALN0200
       END                                                             BALN0205
```

7.2.4 SUBROUTINE SUN

Being given the value of the solar zenith angle and the value of CIMAX, this subroutine when called by BALAN computes the transmission factor for use in changing the rate coefficients for photodetachment and photodissociation.

The following statements are a listing of the FORTRAN statements for this subroutine.

```
$IBFTC SUN      LIST                                                    SUN00000
       FUNCTION SUN(CHI,ALPHA)                                          SUN00005
C                                                                       SUN00010
C TO COMPUTE ETA AS A FUNCTION OF CHI.                                  SUN00015
C                                                                       SUN00020
       DATA GAMMA/4.3633231E-3/                                         SUN00025
       X=(CHI-(ALPHA-GAMMA))/GAMMA                                      SUN00030
       IF(X) 5,10,10                                                    SUN00035
     5 X=0.0                                                            SUN00040
       GO TO 20                                                         SUN00045
    10 IF(X-2.0) 20,20,15                                               SUN00050
    15 X=2.0                                                            SUN00055
    20 THETA=ARCOS((2.0*((1.0-X)**2))-1.0)                              SUN00060
       IF(X-1.0) 30,25,25                                               SUN00065
    25 THETA=(6.28318531-THETA)                                         SUN00070
    30 A=(THETA/2.0)-((1.0-X)*SQRT((1.0-COS(THETA))/2.0))               SUN00075
       SUN=1.0-(A/3.1415926)                                            SUN00080
       IF(SUN .LT. 0.0) GO TO 40                                        SUN00085
       IF(SUN .GT. 1.0) GO TO 35                                        SUN00090
       GO TO 45                                                         SUN00095
    35 SUN=1.0                                                          SUN00100
       GO TO 45                                                         SUN00105
    40 SUN=0.0                                                          SUN00110
    45 RETURN                                                           SUN00115
       END                                                              SUN00120
```

Acknowledgments

The author wishes to express his gratitude to Dr. Wolfgang Pfister for his continued keen interest in this problem and for his many helpful discussions and recommendations. Thanks are also extended to Ronald J. Fowler for his ideas and help in modifying the code especially as it applies to the diurnal variation and other applications not presented here. The assistance of Miss Margaret Gardner of Boston College in executing the code and in preparing the results for presentation here is also greatly appreciated.

References

Bortner, M. (1965) Research Directed Toward an Investigation of the Chemical Kinetics of Atmosphere Deionization, AFCRL-65-392, Final Report, Contract AF19(628)-4153, General Electric Company.

Bortner, M. (1965) The Atmospheric Chemical Kinetics of N_2O, AFCRL-65-367, Scientific Report No. 2, Contract No. AF19(628)-4153, General Electric Company.

Bortner, M. (1966) Private communication.

Colegrove, F. D., Johnson, F. S., and Hanson, W. B. (1965) Eddy diffusion and oxygen transport in the lower thermosphere, J. Geophys. Res. 71:2227-2236.

References

Cospar International Reference Atmosphere (1965) North Holland Publishing Company, Amsterdam.

Ditchburn, R.W., Bradley, J.E.S., Cannon, C.G., and Munday (1954) Absorption cross sections for Lyman alpha and neighboring lines, Rocket Exploration of the Upper Atmosphere, Pergamon Press, London, p. 327.

Fehsenfeld, F.C., Schmeltokopf, A.L., and Furguson, E.E. (1966) Paper presented at 19th Gaseous Electronics Conference, Atlanta, Georgia.

Furguson, E.E., Fehsenfeld, F.C., Goldan, P.D., and Schmeltokopf, A.L. (1965) Positive ion-neutral reactions in the ionosphere, J. Geophys. Res. 70:4323-4329.

Hinteregger, H.E. (1960) GRD Tech. Note 39, AFCRL-TN-60-485.

Keneshea, T.J. (1962) A Computer Program for Solving the Reaction-Rate Equations in the E Ionospheric Region, AFCRL-62-828.

Keneshea, T.J. (1963) A Solution to the Reaction-Rate Equations in the Atmosphere Below 150 km, AFCRL-63-711.

Lee, P. (1955) Photodissociation of oxygen as inferred from measured absorption coefficients, J. Opt. Soc. Am. 45:703-709.

McIntyre, D. (1965) Private communication.

Merson, R.H. (1957) An operational method for the study of integration processes, Proceedings of a symposium in data processing, Weapons Research Establishment, Salisbury, South Australia.

Narcisi, R.S., and Bailey, A.D. (1965) Mass spectrometer measurements of positive ions at altitudes from 64 to 112 kilometers, J. Geophys. Res. 70:3687-3700.

Narcisi, R.S. (1966) Paper presented at Cospar meeting, Vienna, Austria.

Nicolet, M., and Aikin, A.C. (1960) The formation of the D region of the ionosphere, J. Geophys. Res. 65:1469-1483.

U.S. Standard Atmosphere (1962) U.S. Government Printing Office, Washington, D.C.

Watanabe, K. (1954) Photoionization and total absorption cross sections of gases, I ionization potentials of several molecules, cross sections of NH_3 and NO, J. Chem. Phys. 22:1564-1573.

Appendix A

The Photoionization Production Function Program

As discussed in Section 5.1, there is a separate code for the computation of the photoionization production function tables. This code consists of three programs; the main program, subroutine PRODUC, and subroutine COLUM. The output from this code is a listing and a deck of 145 cards containing the number of ions produced as a function of the zenith angle for a given altitude and geographic location. This is the deck of cards that is required as part of the input to the diurnal variation code.

The main program simply controls the input and output of this code. A considerable amount of input is required in order to compute the photoionization functions for each of the species ionized. This information is read into the computer by Cards Nos. 40 through 100 of the main program. The input consists of the photoionization and absorption cross sections for O, O_2, and N_2; the flux in photons/cm^2/sec of the incident solar radiation at the top of the atmosphere; the hard X-ray flux and absorption cross sections for air for these X-rays; and the neutral atmosphere profiles from 0 to 520 km of O, O_2, and N_2. These parameters, as taken from Tables 1 and 2, are read in the following order into the designated regions.

SPO2 = 45 values of the photoionization cross section of O_2 .

SO2 = 45 values of the total absorption cross section of O_2 .

SPO = 34 values of the photoionization cross section of O .

SO = 34 values of the total absorption cross section of O .

SPN2 = 24 values of the photoionization cross section of N_2 .

SN2 = 24 values of the total absorption cross section of N_2 .

PIII = values of the solar flux in photons/cm^2/sec broken down into 24 lines or bands.

RAD = the X-ray flux at 2, 4, and 8A.

ABC = absorption cross section for air at 2,4, and 8A.

These parameters are punched on cards in FORMAT (1P6E12.5):

CONO = the height profile of O in number/cm^3 at every 10 km.

CONO2 = the height profile of O_2 in number/cm^3 at every 10 km.

CONN2 = the height profile of N_2 in number/cm^3 at every 10 km.

The last three parameters are punched on cards in FORMAT (1P9E8.2).

The main program sets up the BK region with values of altitude in 10 km increments from 0 to 520 km for ease of table look-up of required concentrations. The concentrations as read into the computer are converted into the corresponding common logarithms since it is more realistic to linearly interpolate the logs of the densities rather than the concentrations themselves whenever a nontabular value is required.

After all of these basic parameters are read into the computer, the parameters of the altitude and the geographic location are read in. Each altitude card is proceeded by a title card on which 72 columns of hollorith information may be punched. The information on this title card is punched out on a heading card preceeding the cards containing the photoionization functions. The second card of this set of two contains the following parameters punched in FORMAT (1P6E12.5).

HITE = altitude in km at which the functions are required.

EPSI = the L_α flux at the top of the atmosphere in ergs/cm^2/sec.

EAR = radius of the earth in km.

DECL = solar declination in degrees.

CLAT = latitude in degrees.

The printed output consists of the following:

a) the altitude, solar declination, and latitude;

b) the solar zenith angle and the number of seconds after noon, corresponding to this zenith angle;

c) $\int_z^{520} M\, dl$, $\int_z^{520} N(O_2)dl$, $\int_z^{520} N(N_2)dl$, and $\int_z^{520} N(O)dl$;

d) the production rate of O_2 , O , and N_2 by X-rays in the wavelength region <170A and the rate coefficient for the ionization of NO by L_α ;

e) the production rate of O_2 , O , and N_2 by UV at wavelengths >170A;

f) the production rate of O_2 and N_2 by cosmic rays; and

g) the total production rates by X-rays, UV, and cosmic rays.

The punched output consists of the following:

a) a title card which is a duplicate of the title card read into the computer; and,

b) a deck of 145 cards containing the production functions.

There are two cards for one set of values of these functions. A set is computed for every ten minutes from noon to midnight. It is assumed that the values from midnight to noon are the same as these.

The parameters which are punched in FORMAT (1P5E14.7,2X,I3,2X,I3) are as follows:

a) SQ2 = total UV production rate of O_2^+ ;

b) SQO = total UV production rate of O^+ ;

c) SQN2 = total UV production rate of N_2^+ ;

d) QNO = rate coefficient for production of NO^+ by ionization of NO by L_α ;

e) WHOLE = total production rate of electrons except for L_α ionization;

f) HHTE = altitude of computations;

g) KARDS = card sequence number;

h) CHI = solar zenith angle;

i) SXO2 = production rate of O_2^+ by X-rays;

j) SXO = production rate of O^+ by X-rays;

k) SNX2 = production rate of N_2^+ by X-rays; and,

l) SCRAY = total cosmic ray production.

The following statements, beginning on page A5, are a listing of the main program.

Subroutine PRODUC computes the actual production rate of each of the species using the input data and the equations derived in Section 5.1. The following statements, beginning on page A7 , are a listing of this code.

Subroutine COLUM computes the number of particles in a cm^2 column along a ray path at a given solar zenith angle and extending from the height z to 520 km. An upper limit of the 520 km was chosen because for paths that extend down to the D and E regions particle concentrations above 520 km do not contribute significantly to the total integral. The subroutine exits with the values of the following integrals in the corresponding locations.

$$\text{DEPTH} = \int M \, dl$$

$$\text{DEPO}_2 = \int N(O_2) \, dl$$

$$\text{DEPO} = \int N(O) \, dl$$

$$\text{DEPN2} = \int N(N_2) \, dl \, .$$

The following statements, beginning on page A9, are a listing of the FORTRAN statements for this subroutine.

```
$18FTC MAIN     LIST                                                    MAIN0000
C PHOTOIONIZATION SOURCE FUNCTION                                       MAIN0005
      COMMON SPO2(45),SO2(45),SPO(45),SO(45),SPN2(45),SN2(45),PHI(24),RAMAIN0010
     10(3),QO2(27),QO(27),QN2(27),ABC(3),BK(150),CONT(150),CONO2(150),COMAIN0015
     2NN2(150),CONO(150),CONTL(150),CONO2L(150),CONN2L(150),CONOL(150)  MAIN0020
      COMMON DEPO2,DEPN2,DEPTH,SQO2,SQO,SQN2,QNO,CO2,CN2,EPSI,R,HITE,CHIMAIN0025
     1,CIMIN,DO2,DN2,DO,D,DNO,EAR,SXO2,SXO,SXN2,KLAW,DEPO              MAIN0030
      DIMENSION TITLE(12)                                               MAIN0035
      READ(5,85)(SPO2(J),J=1,45)                                        MAIN0040
      READ(5,85)(SO2(J),J=1,45)                                         MAIN0045
      READ(5,85)(SPO(J),J=12,45)                                        MAIN0050
      READ(5,85)(SO(J),J=12,45)                                         MAIN0055
      READ(5,85)(SPN2(J),J=22,45)                                       MAIN0060
      READ(5,85)(SN2(J),J=1,45)                                         MAIN0065
      READ(5,85)(PHI(J),J=1,24)                                         MAIN0070
      READ(5,85)(RAD(J),J=1,3)                                          MAIN0075
      READ(5,85)(ABC(J),J=1,3)                                          MAIN0080
      READ(5,65)(CONO(J),J=1,52)                                        MAIN0085
      READ(5,65)(CONO2(J),J=1,52)                                       MAIN0090
      READ(5,65)(CONN2(J),J=1,52)                                       MAIN0095
      BK(1)=0.0                                                         MAIN0100
      DO 5 J=2,52                                                       MAIN0105
    5 BK(J)=BK(J-1)+10.0                                                MAIN0110
      DO 10 J=1,52                                                      MAIN0115
   10 CONT(J)=CONO(J)+CONO2(J)+CONN2(J)                                 MAIN0120
      DO 15 J=1,52                                                      MAIN0125
      CONTL(J)=ALOG10(CONT(J))                                          MAIN0130
      CONOL(J)=ALOG10(CONO(J))                                          MAIN0135
      CONO2L(J)=ALOG10(CONO2(J))                                        MAIN0140
   15 CONN2L(J)=ALOG10(CONN2(J))                                        MAIN0145
   20 READ(5,80)(TITLE(J),J=1,12)                                       MAIN0150
      READ(5,85)HITE,EPSI,B,EAR,DECL,CLAT                               MAIN0155
      IF(HITE) 60,60,25                                                 MAIN0160
   25 WRITE(6,70)HITE,DECL,CLAT                                         MAIN0165
      KOUNT=0                                                           MAIN0170
      KARDS=0                                                           MAIN0175
      IHITE=HITE                                                        MAIN0180
      WRITE(7,80)(TITLE(J),J=1,12)                                      MAIN0185
      DO 30 J=1,51                                                      MAIN0190
      IF(HITE-BK(J)) 35,35,30                                           MAIN0195
   30 CONTINUE                                                          MAIN0200
   35 RATY=(HITE-BK(J-1))/10.0                                          MAIN0205
      DO2=10.0**(CONO2L(J-1)+RATY*(CONO2L(J)-CONO2L(J-1)))              MAIN0210
      DN2=10.0**(CONN2L(J-1)+RATY*(CONN2L(J)-CONN2L(J-1)))              MAIN0215
      DO=10.0**(CONOL(J-1)+RATY*(CONOL(J)-CONOL(J-1)))                  MAIN0220
      D=10.0**(CONTL(J-1)+RATY*(CONTL(J)-CONTL(J-1)))                   MAIN0225
      DECR=DECL*0.01745329                                              MAIN0230
      CLAR=CLAT*0.01745329                                              MAIN0235
      COSD=COS (DECR)*COS (CLAR)                                        MAIN0240
      SIND=SIN (DECR)*SIN (CLAR)                                        MAIN0245
      TYM=0.0                                                           MAIN0250
      CIMIN=3.1415926-ARSIN(EAR/(EAR+HITE))                             MAIN0255
   40 CHI=ARCOS(COSD*COS (7.2722052E-5*TYM)+SIND)                       MAIN0260
      CHID=57.295779*CHI                                                MAIN0265
      WRITE(6,75)CHID,TYM                                               MAIN0270
      CALL PRODUC                                                       MAIN0275
      WRITE(6,110)DEPTH,DEPO2,DEPN2,DEPO                                MAIN0280
      SXRAY=SXO2+SXN2+SXO                                               MAIN0285
      SUV=SQO2+SQO+SQN2                                                 MAIN0290
```

```
      SCRAY=CO2+CN2                                                    MAIN0295
      WHOLE=SXRAY+SUV+SCRAY                                            MAIN0300
      WRITE(6,90)SXO2,SXO,SXN2,QNO                                     MAIN0305
      WRITE(6,95)SQO2,SCO,SQN2                                         MAIN0310
      WRITE(6,105)CO2,CN2                                              MAIN0315
      WRITE(6,100)SXRAY,SUV,SCRAY,WHOLE                                MAIN0320
      WRITE(7,115) SQO2,SQO,SQN2,QNO,WHOLE,IHITE,KARDS                 MAIN0325
      KARDS=KARDS+1                                                    MAIN0330
      WRITE(7,115) CHI,SXO2,SXO,SXN2,SCRAY,IHITE,KARDS                 MAIN0335
      KARDS=KARDS+1                                                    MAIN0340
      KOUNT=KOUNT+8                                                    MAIN0345
      IF(KOUNT-56) 50,45,45                                            MAIN0350
   45 KOUNT=0                                                          MAIN0355
      WRITE(6,70)HITE,DECL,CLAT                                        MAIN0360
   50 IF(B) 60,55,20                                                   MAIN0365
   55 TYM=TYM+6.0E2                                                    MAIN0370
      IF(TYM-4.32E4) 40,40,20                                          MAIN0375
   60 CALL EXIT                                                        MAIN0380
   65 FORMAT(1P9E8.2)                                                  MAIN0385
   70 FORMAT(13H1  ALTITUDE =1PE13.5,4H CM.,6X19HSOLAR DECLINATION =1PE1MAIN0390
     13.5,10H DEGREES. ,6X10HLATITUDE =1PE13.5,10H DEGREES  )          MAIN0395
   75 FORMAT(15HOZENITH ANGLE =F7.2,10H DEGREES. ,18H TIME AFTER NOON =,MAIN0400
     11PE11.3,6H SEC. )                                               MAIN0405
   80 FORMAT(12A6)                                                     MAIN0410
   85 FORMAT(1P6E12.5)                                                 MAIN0415
   90 FORMAT(1H ,15HO2+ FROM XRAYS=1PE12.5,3X,14HO+ FROM XRAYS=1PE12.5,3MAIN0420
     1X,15HN2+ FROM XRAYS=1PE12.5,3X,21HNO+ COEFFICIENT     =1PE12.5)  MAIN0425
   95 FORMAT(1H ,15HO2+ FROM UV   =1PE12.5,3X,14HO+ FROM UV   =1PE12.5,3MAIN0430
     1X,15HN2+ FROM UV   =1PE12.5)                                    MAIN0435
  100 FORMAT(17H XRAY ELECTRONS =1PE12.5,17H   UV ELECTRONS =1PE12.5,25HMAIN0440
     1   COSMIC RAY ELECTRONS =1PE12.5,20H   TOTAL ELECTRONS =1PE12.5)  MAIN0445
  105 FORMAT(16H O2+ COSMIC RAY=1PE12.5,32X15HN2+ COSMIC RAY=1PE12.5)   MAIN0450
  110 FORMAT(18HO TOTAL INTEGRAL =1PE13.5,14H O2 INTEGRAL =E13.5,14H N2 MAIN0455
     1INTEGRAL =E13.5,13H O INTEGRAL =E13.5)                           MAIN0460
  115 FORMAT(1P5E14.7,2X,I3,2X,I3)                                     MAIN0465
      END                                                              MAIN0470
```

```
$IBFTC PRODUC  LIST                                                    PROD0000
       SUBROUTINE PRODUC                                               PROD0005
C                                                                      PROD0010
C      PHOTOIONIZATION PRODUCTION FUNCTION                             PROD0015
C                                                                      PROD0020
       COMMON SPO2(45),SO2(45),SPO(45),SO(45),SPN2(45),SN2(45),PHI(24),RAPROD0025
      1D(3),QO2(27),QO(27),QN2(27),ABC(3),BK(150),CONT(150),CONO2(150),COPROD0030
      2NN2(150),CONO(150),CONTL(150),CONO2L(150),CONN2L(150),CONOL(150) PROD0035
       COMMON DEPO2,DEPN2,DEPTH,SQO2,SQO,SQN2,QNO,CO2,CN2,EPSI,R,HITE,CHIPROD0040
      1,CIMIN,DO2,DN2,DO,D,DNO,EAR,SXO2,SXO,SXN2,KLAW,DEPO             PROD0045
       DIMENSION POWER(45)                                             PROD0050
       DO 5 J=1,27                                                     PROD0055
       QO2(J)=0.0                                                      PROD0060
       QO(J)=0.0                                                       PROD0065
     5 QN2(J)=0.0                                                      PROD0070
       SQO2=0.0                                                        PROD0075
       SQO=0.0                                                         PROD0080
       SQN2=0.0                                                        PROD0085
       QNO=0.0                                                         PROD0090
       SXO2=0.0                                                        PROD0095
       SXN2=0.0                                                        PROD0100
       SXO=0.0                                                         PROD0105
       IF(CHI-CIMIN) 10,10,130                                         PROD0110
    10 CALL COLUM                                                      PROD0115
       DO 15 J=1,45                                                    PROD0120
    15 POWER(J)=EXP(-DEPO2*SO2(J)-DEPO*SO(J)-DEPN2*SN2(J))             PROD0125
       DO 20 J=1,6                                                     PROD0130
    20 QO2(J)=SPO2(J)*DO2*PHI(J)*POWER(J)                              PROD0135
       K=7                                                             PROD0140
       K1=11                                                           PROD0145
       K2=1                                                            PROD0150
       DO 50 J=7,9                                                     PROD0155
    25 PART1=SPO2(K)*DO2*PHI(J)                                        PROD0160
       PART2=SPO(K)*DO*PHI(J)                                          PROD0165
       QO2(J)=QO2(J)+(PART1*POWER(K))                                  PROD0170
       QO(J)=QO(J)+(PART2*POWER(K))                                    PROD0175
       IF(K-K1) 30,35,35                                               PROD0180
    30 K=K+1                                                           PROD0185
       GO TO 25                                                        PROD0190
    35 K=K+1                                                           PROD0195
       GO TO (40,45,55),K2                                             PROD0200
    40 K1=16                                                           PROD0205
       K2=2                                                            PROD0210
       GO TO 50                                                        PROD0215
    45 K1=21                                                           PROD0220
       K2=3                                                            PROD0225
    50 CONTINUE                                                        PROD0230
    55 K1=24                                                           PROD0235
       K2=1                                                            PROD0240
       DO 80 J=10,11                                                   PROD0245
    60 PART1=SPO2(K)*DO2*PHI(J)                                        PROD0250
       PART2=SPO(K)*DO*PHI(J)                                          PROD0255
       PART3=SPN2(K)*DN2*PHI(J)                                        PROD0260
       QO2(J)=QO2(J)+(PART1*POWER(K))                                  PROD0265
       QO(J)=QO(J)+(PART2*POWER(K))                                    PROD0270
       QN2(J)=QN2(J)+(PART3*POWER(K))                                  PROD0275
       IF(K-K1) 65,70,70                                               PROD0280
    65 K=K+1                                                           PROD0285
       GO TO 60                                                        PROD0290
```

```
 70 K=K+1                                                                  PROD0295
    GO TO (75,80),K2                                                       PROD0300
 75 K1=27                                                                  PROD0305
    K2=2                                                                   PROD0310
 80 CONTINUE                                                               PROD0315
    QO2(12)=SPO2(28)*DO2*PHI(12)*POWER(K)                                  PROD0320
    QO(12)=SPO(28)*DO*PHI(12)*POWER(K)                                     PROD0325
    QN2(12)=SPN2(28)*DN2*PHI(12)*POWER(K)                                  PROD0330
    K=29                                                                   PROD0335
    K1=31                                                                  PROD0340
    K2=1                                                                   PROD0345
    DO 105 J=13,14                                                         PROD0350
 85 PART1=SPO2(K)*DO2*PHI(J)                                               PROD0355
    PART2=SPO(K)*DO*PHI(J)                                                 PROD0360
    PART3=SPN2(K)*DO*PHI(J)                                                PROD0365
    QO2(J)=QO2(J)+(PART1*POWER(K))                                         PROD0370
    QO(J)=QO(J)+(PART2*POWER(K))                                           PROD0375
    QN2(J)=QN2(J)+(PART3*POWER(K))                                         PROD0380
    IF(K-K1) 90,95,95                                                      PROD0385
 90 K=K+1                                                                  PROD0390
    GO TO 85                                                               PROD0395
 95 K=K+1                                                                  PROD0400
    GO TO (100,105),K2                                                     PROD0405
100 K1=34                                                                  PROD0410
    K2=2                                                                   PROD0415
105 CONTINUE                                                               PROD0420
    QO2(15)=SPO2(35)*DO2*PHI(15)*POWER(K)                                  PROD0425
    QO(15)=SPO(35)*DO*PHI(15)*POWER(K)                                     PROD0430
    QN2(15)=SPN2(35)*DO*PHI(15)*POWER(K)                                   PROD0435
    QO2(16)=(DO2*PHI(16))*(SPO2(36)*POWER(36)+SPO2(37)*POWER(37))          PROD0440
    QO(16)=(DO*PHI(16))*(SPO(36)*POWER(36)+SPO(37)*POWER(37))              PROD0445
    QN2(16)=DN2*PHI(16)*(SPN2(36)*POWER(36)+SPN2(37)*POWER(37))            PROD0450
    K=38                                                                   PROD0455
    DO 110 J=17,24                                                         PROD0460
    QO2(J)=SPO2(K)*DO2*PHI(J)*POWER(K)                                     PROD0465
    QO(J)=SPO(K)*DO*PHI(J)*POWER(K)                                        PROD0470
    QN2(J)=SPN2(K)*DN2*PHI(J)*POWER(K)                                     PROD0475
110 K=K+1                                                                  PROD0480
    L=1                                                                    PROD0485
    DO 115 J=25,27                                                         PROD0490
    QO2(J)=DO2*RAD(L)*EXP (-(DEPTH*ABC(L)))                                PROD0495
    QO(J)=DO*RAD(L)*0.75*EXP (-(DEPTH*ABC(L)))                             PROD0500
    QN2(J)=DN2*RAD(L)*EXP (-(DEPTH*ABC(L)))                                PROD0505
    L=L+1                                                                  PROD0510
115 CONTINUE                                                               PROD0515
    DO 120 J=1,18                                                          PROD0520
    SQO2=SQO2+QO2(J)                                                       PROD0525
    SQN2=SQN2+QN2(J)                                                       PROD0530
120 SQO=SQO+QO(J)                                                          PROD0535
    DO 125 J=19,27                                                         PROD0540
    SXO2=SXO2+QO2(J)                                                       PROD0545
    SXN2=SXN2+QN2(J)                                                       PROD0550
125 SXO=SXO+QO(J)                                                          PROD0555
    QNO=EPSI*1.34E-7*EXP (-8.5E-21*DEPO2)                                  PROD0560
130 QT=(1.0E-16*D)/(DO2+DN2)                                               PROD0565
    CO2=QT*QO2                                                             PROD0570
    CN2=QT*DN2                                                             PROD0575
135 RETURN                                                                 PROD0580
    END                                                                    PROD0585
```

```
$IBFTC COLUM   LIST                                                    COLM0000
      SUBROUTINE COLUM                                                 COLM0005
      COMMON SPO2(45),SO2(45),SPO(45),SO(45),SPN2(45),SN2(45),PHI(24),RACOLM0010
     1D(3),QO2(27),QO(27),QN2(27),ABC(3),BK(150),CONT(150),CONO2(150),COCOLM0015
     2NN2(150),CONO(150),CONTL(150),CONO2L(150),CONN2L(150),CONOL(150)  COLM0020
      COMMON DEPO2,DEPN2,DEPTH,SQO2,SQO,SQN2,QNO,CO2,CN2,EPSI,R,HITE,CHICOLM0025
     1,CIMIN,DO2,DN2,DO,D,DNO,EAR,SXO2,SXO,SXN2,KLA,DEPO              COLM0030
      DIMENSION ANS(4),A(4),A2(4),ANSL(4),A3(4),ANSQ(4),POT(4),ANSW(4) COLM0035
C                                                                      COLM0040
      DO 5 I=1,4                                                       COLM0045
      A2(I)=0.0                                                        COLM0050
    5 A3(I)=0.0                                                        COLM0055
      TEN=10.0                                                         COLM0060
      DEGRA=0.5236                                                     COLM0065
      P=0.0                                                            COLM0070
      Q=0.0                                                            COLM0075
      KUTY=0                                                           COLM0080
      KUTD=4                                                           COLM0085
   10 TEST=TEN                                                         COLM0090
   15 UTOR=(EAR+HITE)                                                  COLM0095
      BETAR=3.1415926-CHI                                              COLM0100
      IF(BETAR-ARSIN(EAR/UTOR)) 20,30,30                               COLM0105
   20 WRITE (6,25)                                                     COLM0110
   25 FORMAT(34HOTHIS PATH GOES BELOW THE HORIZON )                    COLM0115
      GO TO 230                                                        COLM0120
   30 COSB=COS(BETAR)                                                  COLM0125
      DISTR=0.0                                                        COLM0130
      BUTR2= UTOR**2                                                   COLM0135
      YUR=2.0* UTOR*COSB                                               COLM0140
      IF(HITE-95.0) 40,35,35                                           COLM0145
   35 IF(CHI-1.5707963) 45,45,40                                       COLM0150
   40 KUTY=1                                                           COLM0155
      KUTD=3                                                           COLM0160
   45 HTOR=(SQRT(BUTR2+(DISTR**2)-(DISTR*YUR)))- EAR                   COLM0165
   50 DO 65 J=1,101                                                    COLM0170
      IF(BK(J)-HTOR)65,55,90                                           COLM0175
   55 ANS(1)=CONT(J)                                                   COLM0180
      ANS(2)=CONO2(J)                                                  COLM0185
      ANS(3)=CONN2(J)                                                  COLM0190
      IF(KUTY) 100,60,100                                              COLM0195
   60 ANS(4)=CONO(J)                                                   COLM0200
      GO TO 100                                                        COLM0205
   65 CONTINUE                                                         COLM0210
      IF(Q) 70,80,70                                                   COLM0215
   70 DO 75 I=1,KUTD                                                   COLM0220
      A2(I)=A2(I)-ANSL(I)                                              COLM0225
      A3(I)=A3(I)-ANSQ(I)                                              COLM0230
   75 ANSW(I)=ANSQ(I)                                                  COLM0235
      GO TO 145                                                        COLM0240
   80 DO 85 I=1,KUTD                                                   COLM0245
      A3(I)=A3(I)-ANSQ(I)                                              COLM0250
   85 ANSW(I)=ANSQ(I)                                                  COLM0255
      GO TO 145                                                        COLM0260
   90 TLAZY=(HTOR-BK(J-1))/10.0                                        COLM0265
      ANS(1)=10.0**(CONTL(J-1)+(TLAZY*(CONTL(J)-CONTL(J-1))))          COLM0270
      ANS(2)=10.0**(CONO2L(J-1)+(TLAZY*(CONO2L(J)-CONO2L(J-1))))       COLM0275
      ANS(3)=10.0**(CONN2L(J-1)+(TLAZY*(CONN2L(J)-CONN2L(J-1))))       COLM0280
      IF(KUTY) 100,95,100                                              COLM0285
   95 ANS(4)=10.0**(CONOL(J-1)+(TLAZY*(CONOL(J)-CONOL(J-1))))          COLM0290
```

```
100 IF(P) 115,105,115                                                  COLM0295
105 DO 110 I=1,KUTD                                                    COLM0300
110 A(I)=ANS(I)                                                        COLM0305
    P=1.0                                                              COLM0310
    GO TO 140                                                          COLM0315
115 IF(Q) 130,120,130                                                  COLM0320
120 DO 125 I=1,KUTD                                                    COLM0325
    A2(I)=ANS(I)+A2(I)                                                 COLM0330
125 ANSL(I)=ANS(I)                                                     COLM0335
    Q=1.0                                                              COLM0340
    GO TO 140                                                          COLM0345
130 DO 135 I=1,KUTD                                                    COLM0350
    A3(I)=ANS(I)+A3(I)                                                 COLM0355
135 ANSQ(I)=ANS(I)                                                     COLM0360
    Q=0.0                                                              COLM0365
140 DISYR=DISTR+TEST                                                   COLM0370
    GO TO 45                                                           COLM0375
145 DO 150 I=1,KUTD                                                    COLM0380
150 POT(I)=(((TEST*1.0E5)/3.0)*((A(I)+ANSW(I))+(4.0*A2(I))+(2.0*A3(I)))COLM0385
    DEPTH=POT(1)                                                       COLM0390
    DEPO2=POT(2)                                                       COLM0395
    DEPN2=POT(3)                                                       COLM0400
    IF(KUYY) 160,155,160                                               COLM0405
155 DEPO=POT(4)                                                        COLM0410
    GO TO 230                                                          COLM0415
160 P=0.0                                                              COLM0420
    Q=0.0                                                              COLM0425
    DISYR=0.0                                                          COLM0430
165 HTOR=(SQRT(BUTR2+(DISTR**2)-(DISTR*YUR)))- EAR                     COLM0435
    DO 175 J=1,51                                                      COLM0440
    IF(BK(J)-HTOR) 175,170,165                                         COLM0445
170 ANS(4)=CONO(J)                                                     COLM0450
    GO TO 200                                                          COLM0455
175 CONTINUE                                                           COLM0460
    IF(Q) 180,190,180                                                  COLM0465
180 A2(4)=A2(4)-ANSL(4)                                                COLM0470
    A3(4)=A3(4)-ANSQ(4)                                                COLM0475
    ANSW(4)=ANSQ(4)                                                    COLM0480
185 DEPO=1.666667E5*((A(4)+ANSW(4))+(4.0*A2(4))+(2.0*A3(4)))           COLM0485
    GO TO 230                                                          COLM0490
190 A3(4)=A3(4)-ANSQ(4)                                                COLM0495
    ANSW(4)=ANSQ(4)                                                    COLM0500
    GO TO 185                                                          COLM0505
195 TLAZY=(HTOR-BK(J-1))/10.0                                          COLM0510
    ANS(4)=10.0**(CONOL(J-1)+TLAZY*(CONOL(J)-CONOL(J-1)))              COLM0515
200 IF(P) 210,205,210                                                  COLM0520
205 A(4)=ANS(4)                                                        COLM0525
    P=1.0                                                              COLM0530
    GO TO 225                                                          COLM0535
210 IF(Q) 220,215,220                                                  COLM0540
215 A2(4)=ANS(4)+A2(4)                                                 COLM0545
    ANSL(4)=ANS(4)                                                     COLM0550
    Q=1.0                                                              COLM0555
    GO TO 225                                                          COLM0560
220 A3(4)=ANS(4)+A3(4)                                                 COLM0565
    ANSQ(4)=ANS(4)                                                     COLM0570
    Q=0.0                                                              COLM0575
225 DISTR=DISTR+5.0                                                    COLM0580
    GO TO 165                                                          COLM0585

230 RETURN                                                             COLM0590
    END                                                                COLM0595
```

Appendix B

The Differential Equation Writer Program

This code was originally written by David McIntyre (1965) for computing the total derivative of each species with respect to time and the partial derivatives of each species with respect to the other species. His code was written for an IBM-6000 computer; so in order to use this code on an IBM-7044 or 7094 computer it had to be rewritten. The author has made the modifications necessary for computing the codes for the formation sums $\sum F_i$ and the removal sums $\sum R_i$. The code consists of a main program and eight subroutines; DIFFEQ, SET8, DY, REFACT, DIFFER, FACTOR, OUT, CODER, and DECODE. Because the last two subprograms involve character manipulations that are not easily performed in the FORTRAN language, they are written in the MAP language.

The main program essentially controls the input and output. The input consists of a deck of cards containing the reactions in coded form. The output is a listing and a deck of cards containing the FORTRAN statements for subroutine SLOP. The only thing that has to be added to the deck is the proper COMMON statement. For the system described in this report, the following code applies. Since all of the species listed after the total density are not considered in the program, there is no output code for them.

Species	Input to Program	Output From Program	Species	Input to Program	Output From Program
e	5	Y(1)	N_2O	18	Y(14)
O^-	6	Y(2)	O	19	Y(15)
O_2^-	7	Y(3)	O_2	20	Y(16)
O_3^-	8	Y(4)	N_2	21	Y(17)
NO_2^-	9	Y(5)	Total Density	22	Y(18)
O^+	10	Y(6)	N^+	23	----
O_2^+	11	Y(7)	NO_3^+	24	----
N_2^+	12	Y(8)	N_2O^+	25	----
NO^+	13	Y(9)	O_3^+	26	----
NO	14	Y(10)	NO_3	27	----
N	15	Y(11)	N_4^+	28	----
NO_2	16	Y(12)	NO_2^+	29	----
O_3	17	Y(13)	NO^-	30	----

The reactions are punched on cards in coded form in FORMAT (20I4). Consider the reaction

$$O + O_2 + N_2 \xrightarrow{k_{139}} O_3 + N_2 \quad .$$

This reaction is coded as

$$Y(19) + Y(20) + Y(21) \longrightarrow Y(17) + Y(21) + O\ k_{139}$$

and punched as

bb19bb20bb21bb17bb21bbb0b139.

Each reaction must contain seven integers: three for the reactants, three for the products, and one for the reaction number. If a reaction contains less than three reactants or less than three products the corresponding subfields must contain zeros.

The output is a complete FORTRAN deck from the $IBFTC card to the END card with the exception of a COMMON statement. Although the subroutine is a general one, it still requires a COMMON statement compatible with the remainder of the code in which it is used. The following statements, beginning on page B4, are a listing of the main program.

Subroutine SET8 is called by the main program and assigns an identification to each of the reactions according to type. This is a digit from one to seven and is added to the coded reaction as the eighth integer. The maximum type of reaction that this program can handle is a three-reactant three-product process. The type identifiers as related to the reactions are shown in the comments of the FORTRAN program. The following statements, beginning on page B6, are a listing of subprogram SET8.

Subroutine DY scans the reactions searching for a particular species as a reactant or a product. If it finds the species as a reactant it transfers the reactants to the IREMOV region. The memory cell preceeding the cells containing the reactants is coded as -2 if the reaction type is one, as -3 if the reaction type is two, three, or four, and as -4 if the reaction type is five, six, or seven. If it finds the species as a product it transfers the reactants to the IFORM region. Now the memory cell preceeding the cells containing the reactants is coded as described above except that the sign of the digit is positive.

Upon a normal exit from this subroutine, all of the reactants for processes that form a particular species are in the IFORM region and all of the reactants for processes that remove this species are in the IREMOV region.

The following statements, beginning on page B7, are a listing of subroutine DY.

Subroutine DIFFER computes the partial derivatives of each differential equation with respect to the species for which the equation was written. Effectively, this routine factors out the given species from all the reactions that remove it so that the quantity $\sum R_j$ can be computed. The following statements, beginning on page B8, are a listing of this program.

Subroutine REFACT performs the factoring of species and leaves the reactions in the IFACT region in as highly a factored form as possible. The following statements, beginning on page B9, are a listing of the FORTRAN statements for this subprogram.

Subrouting FACTOR which is called very often by subroutine REFACT does the actual factoring. It scans a particular series of reactions set up by REFACT to find the species occurring most often in the series. It factors out this species and returns to REFACT with the factored version of the series fed to it by REFACT. The following statements, beginning on page B11, are a listing of this subroutine.

```
$IBFTC DIFG    LIST
      COMMON OUTPUT(2000)                                                    DIFQ0005
      DIMENSION IREAC(8,200),IPART(500),INSERT(500),IFACT(500),ICOR(500)DIFQ0010
     1,INSTIK(500),IIOLT(500),IFORM(500),IREMV(500)                          DIFQ0015
      DIMENSION ID(2000),G(5)
      DATA Q/6H+PNE  ,6H+PU2  ,6H+PU   ,6H+PN2  ,6H+PNO
      READ(5,110) NEQNS,NSPEC1
      READ(5,5) ((IREAC(I,J),I=1,7),J=1,NEQNS)                               DIFQ0035
    5 FORMAT(2014)                                                           DIFQ0040
      WRITE(6,90)                                                            DIFQ0045
      PUNCH 115
      NO=1
      N1=2
      CALL SET8(IREAC,NEQNS)                                                 DIFQ0065
      JSTOP=NSPEC1+4                                                         DIFQ0070
      DO 80 J=5,JSTOP                                                        DIFQ0075
      ICOR(1)=1001                                                           DIFQ0080
      IMARK=2                                                                DIFQ0085
      IPOINT=1                                                               DIFQ0090
      ICOUNT=0                                                               DIFQ0095
      CALL DY(J,NEQNS,IREAC,IFORM,IREMV,ISPEAR,LENDY)                        DIFQ0100
      LDEN=J                                                                 DIFQ0105
      CALL DIFFER(IREMV,LENDY,LDEN,IPART,LIPART,LFLAG,ICOR(IMARK),LICOR)DIFQ0110
      IMARK=IMARK+LICOR                                                      DIFQ0115
      JI=J-4                                                                 DIFQ0120
      LDENC=LDEN-4                                                           DIFQ0125
      IF(LIPART)10,30,10                                                     DIFQ0130
   10 INSERT(IPOINT)=J                                                       DIFQ0135
      INSERT(IPOINT+1)=LDEN                                                  DIFQ0140
      INSERT(IPOINT+2)=LDEN                                                  DIFQ0145
   15 IPOINT=IPOINT+3                                                        DIFQ0150
      ICOUNT=ICOUNT+1                                                        DIFQ0155
C                                                                            DIFQ0160
C  PUT PARENTHESIS AROUND IPART                                              DIFQ0165
      DO 20 I=1,LIPART                                                       DIFQ0170
      II=LIPART-I+1                                                          DIFQ0175
   20 IPART(II+1)=IPART(II)                                                  DIFQ0180
      IPART(LIPART+2)=-1001                                                  DIFQ0185
      IPART(1)=1001                                                          DIFQ0190
      LENGTP=LIPART+2                                                        DIFQ0195
      CALL REFACT(IPART,NSPEC1,IFACT,LENGTP)                                 DIFQ0200
      CALL OUT(IFACT,LENGTP,IARROW)                                          DIFQ0205
      DO 21 JJD=1,2000
   21 ID(JJD)=0
      CALL DECODE(OUTPUT,ID,IARROW,INUMB)                                    DIFQ0210
      WRITE(6,95) NO,LDENC,LDENC,N1
      PUNCH 95, NO,LDENC,LDENC,N1
      WRITE(6,25) LDENC,(ID(I),I=1,INUMB)                                    DIFQ0225
      PUNCH 25, LDENC,(ID(I),I=1,INUMB)
   25 FORMAT(6X,5HREMV(,I2,5H)    ,9A6/(5X,1H1,11A6))                       DIFQ0235
C                                                                            DIFQ0240
C  PUT PARENTHESIS AROUND IDYJDT                                             DIFQ0245
   30 DO 35 I=1,ISPEAR                                                       DIFQ0250
      II=ISPEAR-I+1                                                          DIFQ0255
   35 IFORM(II+1)=IFORM(II)                                                  DIFQ0260
      IFORM(ISPEAR+2)=-1001                                                  DIFQ0265
      IFORM(1)=1001                                                          DIFQ0270
      LENGTH=ISPEAR+2                                                        DIFQ0275
```

```
      CALL REFACT(IFORM,NSPECI,IFACT,LENGTH)                          DIFQ0280
      CALL OUT(IFACT,LENGTH,IARROW)                                   DIFQ0285
      DO 36 JJD=1,2000
   36 ID(JJD)=0
      IF(J .EQ. 5) GO TO 40                                           DIFQ0290
      IF(J .EQ. 13) GO TO 45                                          DIFQ0295
      IF(J .EQ. 12 ) GO TO 50                                         DIFQ0300
      IF( J .EQ. 14) GO TO 55                                         DIFQ0305
      IF(J .EQ. 15) GO TO 60                                          DIFQ0310
      GO TO 65                                                        DIFQ0315
   40 IARROW=IARROW+1                                                 DIFQ0320
      OUTPUT(IARROW)=Q(1)
      GO TO 65                                                        DIFQ0335
   45 IARROW=IARROW+1                                                 DIFQ0340
      OUTPUT(IARROW)=C(2)
      GO TO 65                                                        DIFQ0355
   50 IARROW=IARROW+1                                                 DIFQ0360
      OUTPUT(IARROW)=Q(3)
      GO TO 65                                                        DIFQ0375
   55 IARROW=IARROW+1                                                 DIFQ0380
      OUTPUT(IARROW)=C(4)
      GO TO 65                                                        DIFQ0395
   60 IARROW=IARROW+1                                                 DIFQ0400
      OUTPUT(IARROW)=C(5)
   65 CALL DECODE(OUTPUT,ID,IARROW,INUMB)                             DIFQ0415
      JI=J-4                                                          DIFQ0420
   70 WRITE(6,75) JI,(ID(I),I=1,INUMB)                                DIFQ0425
      PUNCH 75, JI,(ID(I),I=1,INUMB)
   75 FORMAT(6X,5HFORM(,I2,5H)     ,9A6/(5X,1H1,11A6))                DIFQ0435
      NO=N1                                                           DIFQ0440
      N1=NO+1                                                         DIFQ0445
   80 CONTINUE                                                        DIFQ0450
      WRITE(6,100) NO                                                 DIFQ0455
      PUNCH 100,NO
      WRITE(6,105)                                                    DIFQ0465
      PUNCH 105
   85 STOP                                                            DIFQ0475
   90 FORMAT(19H1IBFTC SLOP    LIST/6X,21HSUBROUTINE SLOP(KIND)/6X,23HDIDFQ0480
     1MENSION Y(250),C(173)/6X,34HEQUIVALENCE (Y(1),TREG(4)),(C,CON)/5X,
     2    11H CALL BALAN/6X,9HY(19)=DU2/6X,9HY(20)=DN2/6X,7HY(21)=D/6X,2
     33H1F(KIND .EQ. 2) GO TO 1/6X,11HCALL PRODUC)
   95 FORMAT(1H ,14,27H IF((KIND .EQ. 1 .AND. KEY(,I2,38H) .NE. 1) .OR. DIFQ0500
     1(KIND .EQ. 2 .AND. KEY(/5X,1H1,I2,17H) .NE. 2)) GO TO ,I3)      DIFQ0505
  100 FORMAT(1H ,I4,7H RETURN)                                        DIFQ0510
  105 FORMAT(6X,3HEND)                                                DIFQ0515
  110 FORMAT(2I4)
  115 FORMAT(19H1IBFTC SLOP    LIST/6X,21HSUBROUTINE SLOP(KIND)/6X,23HDIDFQ0480
     1MENSION Y(250),C(173)/6X,34HEQUIVALENCE (Y(1),TREG(4)),(C,CON)/5X,
     2    11H CALL BALAN/6X,9HY(19)=DU2/6X,9HY(20)=DN2/6X,7HY(21)=D/6X,2
     33H1F(KIND .EQ. 2) GO TO 1/6X,11HCALL PRODUC)
      END                                                             DIFQ0520
```

```
$IBFTC SET8    LIST                                                          SET80000
       SUBROUTINE SET8(IREAC,NEQNS)                                          SET80005
C                                                                            SET80010
C  SUBROUTINE IDENTIFIES THE TYPE OF EACH REACTION AND LOADS IREAC(8,K)       SET80015
C  WITH THE TYPE NO. FOR EACH EQUATION K=1,2,...,NEQNS L.T. 200.             SET80020
C  TYPE 1          I GOES TO J + K                                           SET80025
C  TYPE 2          I + J GOES TO K                                           SET80030
C  TYPE 3          I + J GOES TO K + L                                       SET80035
C  TYPE 4          I + J GOES TO K + L+M                                     SET80040
C  TYPE 5          I + J + K GOES TO L                                       SET80045
C  TYPE 6          I + J + K GOES TO L + M                                   SET80050
C  TYPE 7          I + J + K GOES TO L + M +N                                SET80055
       DIMENSION IREAC(8,200)                                                SET80060
       DO 40 II=1,NEQNS                                                      SET80065
       LMARK=0                                                               SET80070
       IRIGHT=0                                                              SET80075
       DO 10 IPT=1,3                                                         SET80080
       IF(IREAC(IPT,II))10,5,10                                              SET80085
     5 LMARK=LMARK+1                                                         SET80090
    10 CONTINUE                                                              SET80095
       DO 20 IPT=4,6                                                         SET80100
       IF(IREAC(IPT,II))20,15,20                                             SET80105
    15 IRIGHT=IRIGHT+1                                                       SET80110
    20 CONTINUE                                                              SET80115
       IF(LMARK-1)35,30,25                                                   SET80120
    25 IREAC(8,II)=1                                                         SET80125
       GO TO 40                                                              SET80130
    30 IREAC(8,II)=4-IRIGHT                                                  SET80135
       GO TO 40                                                              SET80140
    35 IREAC(8,II)=7-IRIGHT                                                  SET80145
    40 CONTINUE                                                              SET80150
       RETURN                                                                SET80155
       END                                                                   SET80160
```

```
$IBFTC DY      LIST                                                    DY000000
       SUBROUTINE DY(J,NEQNS,IREAC,IFORM,IREMOV,ISPEAR,IARROW)         DY500005
       DIMENSION IREAC(8,200),IFORM(500),IREMOV(500)                   DY500010
C  IFORM(1) TO IFORM(ISPEAR) CONTAIN IN CODED FORM DY(J)/DT FOR        DY500015
C  FORMATION PROCESSES.                                                DY500020
C  IREMV(1) TO IREMV(IARROW) CONTAIN IN CODED FORM DY(J)/DT FOR        DY500025
C  REMOVAL PROCESSES.                                                  DY500030
C  NEQNS=NO. OF REACTIONS CONSIDERED,NEQNS L. T. 200                   DY500035
C  LENGTH L.T. 500                                                     DY500040
C                                                                      DY500045
C  SEARCH TO FIND IF KTH EQN CONTAINS SPECIES J IN ROW I2              DY500050
       IARROW=0                                                        DY500055
       ISPEAR=0                                                        DY500060
       DO 90 I2=1,6                                                    DY500065
       DO 90 K=1,NEQNS                                                 DY500070
       IF(IREAC(I2,K)-J)90,5,90                                        DY500075
C                                                                      DY500080
C  KTH EQN CONTAINS Y(J) IN ROW I2                                     DY500085
    5  IF(I2-3)10,10,50                                                DY500090
C                                                                      DY500095
C  KTH EQN IS REMOVAL EQN FOR Y(J), CALCULATE REMOVAL TERM IN IREMOVE  DY500100
C  STARTING AT IARROW                                                  DY500105
   10  IARROW=IARROW+1                                                 DY500110
       IF(IREAC(8,K)-2)20,25,15                                        DY500115
   15  IF(IREAC(8,K)-4) 25,25,30                                       DY500120
   20  IREMOV(IARROW)=-2                                               DY500125
       GO TO 35                                                        DY500130
   25  IREMOV(IARROW)=-3                                               DY500135
       GO TO 35                                                        DY500140
   30  IREMOV(IARROW)=-4                                               DY500145
   35  IARROW=IARROW+1                                                 DY500150
       IREMOV(IARROW)=IREAC(7,K)                                       DY500155
       DO 45 II=1,3                                                    DY500160
       IF(IREAC(II,K))95,45,40                                         DY500165
   40  IARROW=IARROW+1                                                 DY500170
       IREMOV(IARROW)=IREAC(II,K)                                      DY500175
   45  CONTINUE                                                        DY500180
       GO TO 90                                                        DY500185
C                                                                      DY500190
C  KTH EQN IS FORMATION EQUATION FOR Y(J), CALCULATE FORMATION TERN IN DY500195
C  IFORM, STARTING AT ISPEAR                                           DY500200
   50  ISPEAR=ISPEAR+1                                                 DY500205
       IF(IREAC(8,K)-2)60,65,55                                        DY500210
   55  IF(IREAC(8,K)-4)65,65,70                                        DY500215
   60  IFORM(ISPEAR)=2                                                 DY500220
       GO TO 75                                                        DY500225
   65  IFORM(ISPEAR)=3                                                 DY500230
       GO TO 75                                                        DY500235
   70  IFORM(ISPEAR)=4                                                 DY500240
   75  ISPEAR=ISPEAR+1                                                 DY500245
       IFORM(ISPEAR)=IREAC(7,K)                                        DY500250
       DO 85 II=1,3                                                    DY500255
       IF(IREAC(II,K))95,85,80                                         DY500260
   80  ISPEAR=ISPEAR+1                                                 DY500265
       IFORM(ISPEAR)=IREAC(II,K)                                       DY500270
   85  CONTINUE                                                        DY500275
   90  CONTINUE                                                        DY500280
       GO TO 105                                                       DY500285
   95  WRITE(6,100)                                                    DY500290
  100  FORMAT(23H ERROR IN SUBROUTINE DY)                              DY500295
  105  RETURN                                                          DY500300
       END                                                             DY500305
```

```
$IBFTC DIFFER  LIST                                                    DIFR0000
      SUBROUTINE DIFFER(INPUT,INSTOP,LDEN,IPART,LENGTH,LFLAG,ICOR,LICOR)DIFR0005
      DIMENSION INPUT(500),IPART(500),ICOR(500)                        DIFR0010
C                                                                      DIFR0015
C  SUBROUTINE TAKES PORTION OF CORE FROM INPUT(1) TO ANS INCLUDING     DIFR0020
C  INPUT(INSTOP) AND CALCULATES PARTIAL DERIVATIVE OF IT W.R.T. Y(LDEN).DIFR0025
C  INSTOP L.T. 200                                                     DIFR0030
C  LDEN=5,6,...,NSPECI+4                                               DIFR0035
C  DERIVATIVE IS STORED IN IPART(1) TO IPART(LENGTH)                   DIFR0040
C  IF LENGTH = 0 , DERIVATIVE IS ZERO                                  DIFR0045
      IARROW=0                                                         DIFR0050
      IPOINT=0                                                         DIFR0055
      INDEX=1                                                          DIFR0060
C                                                                      DIFR0065
C  CHECK FOR END OF INPUT PORTION                                      DIFR0070
    5 IF(INPUT(INDEX))10,75,10                                         DIFR0075
   10 IF(IABS(INPUT(INDEX))-4)15,15,80                                 DIFR0080
   15 IF(IABS(INPUT(INDEX))-1)70,70,20                                 DIFR0085
C                                                                      DIFR0090
C  SCAN SERIES FOR LDEN                                                DIFR0095
   20 ISTOP=IABS(INPUT(INDEX))                                         DIFR0100
      ICOUNT=0                                                         DIFR0105
      DO 30 I=2,ISTOP                                                  DIFR0110
      II=INDEX+I                                                       DIFR0115
      IF(INPUT(II)-LDEN)30,25,30                                       DIFR0120
   25 ICOUNT=ICOUNT+1                                                  DIFR0125
   30 CONTINUE                                                         DIFR0130
      IF(ICOUNT-1)35,40,40                                             DIFR0135
C                                                                      DIFR0140
C  NO LDEN IN SERIES                                                   DIFR0145
   35 GO TO 70                                                         DIFR0150
C                                                                      DIFR0155
C  ONE LDEN IN SERIES,CALCULATE PARTIAL IN IPART,STARTING AT IARROW    DIFR0160
   40 IARROW=IARROW+1                                                  DIFR0165
      IPART(IARROW)=IABS(INPUT(INDEX))-1                               DIFR0170
      IPART(IARROW)=ISIGN(IPART(IARROW),INPUT(INDEX))                  DIFR0175
      IPART(IARROW+1)=INPUT(INDEX+1)                                   DIFR0180
      IARROW=IARROW+1                                                  DIFR0185
   45 DO 55 I=2,ISTOP                                                  DIFR0190
      II=INDEX+I                                                       DIFR0195
      IF(INPUT(II)-LDEN)50,55,50                                       DIFR0200
   50 IARROW=IARROW+1                                                  DIFR0205
      IPART(IARROW)=INPUT(II)                                          DIFR0210
   55 CONTINUE                                                         DIFR0215
      IF(ICOUNT-1)60,70,60                                             DIFR0220
C                                                                      DIFR0225
C   LDEN OCCURS MORE THAN ONCE IN SERIES,DIFFERENTIATE AGAIN           DIFR0230
   60 DO 65 LL=2,ICOUNT                                                DIFR0235
      IARROW=IARROW+1                                                  DIFR0240
   65 IPART(IARROW)=LDEN                                               DIFR0245
C                                                                      DIFR0250
C  RESET INDEX                                                         DIFR0255
   70 INDEX=INDEX+IABS(INPUT(INDEX))+1                                 DIFR0260
      IF(INDEX-INSTOP) 5,5,75                                          DIFR0265
   75 LENGTH=IARROW                                                    DIFR0270
      LICOR=IPOINT                                                     DIFR0275
      GO TO 90                                                         DIFR0280
   80 WRITE(6,85)                                                      DIFR0285
   85 FORMAT(27H ERROR IN SUBROUTINE DIFFER)                           DIFR0290
   90 RETURN                                                           DIFR0295
      END                                                              DIFR0300
```

```
$IBFTC REFACT  LIST                                                     REFA0000
      SUBROUTINE REFACT(INPUTO,NSPECI,INPUT,LENGTH,JK)                  REFA0005
      DIMENSION INPUTO(500),INPUT(500),IWORK(500),ISTICK(500),INPUTY(500REFA0010
     1)                                                                 REFA0015
C                                                                       REFA0020
C  SUBROUTINE COMPLETELY FACTORS INPUTO(0(1) TO INPUTO(LENGTH) ANS RETURNREFA0025
C  RESULT IN INPUT(1) TO INPUT(LENGTH).  LENGTH ID CHANGED IN ROUTINE   REFA0030
C  LENGTH 6. T. OR E. 1000                                              REFA0035
      DO 5 I=1,LENGTH                                                   REFA0040
    5 INPUT(I)=INPUTO(I)                                                REFA0045
      IBLOCK=0                                                          REFA0050
      IQUIT=1                                                           REFA0055
   10 DO 15 I=1,IQUIT                                                   REFA0060
   15 ISTICK(I)=0                                                       REFA0065
      IBEGIN=IBLOCK+1                                                   REFA0070
C                                                                       REFA0075
C  SEARCH INPUT AREA UNTIL FIND FIRST PAIR OF CLOSED PARENTHESIS TO     REFA0080
C  RIGHT OF IBLOCK                                                      REFA0085
      IF(IBLOCK-LENGTH+6)20,20,175                                      REFA0090
   20 DO 40 INDEX=IBEGIN,LENGTH                                         REFA0095
      IF((ABS(INPUT(INDEX))-1000)40,40,25                               REFA0100
   25 IF(INPUT(INDEX))30,35,35                                          REFA0105
   30 MARK=INDEX                                                        REFA0110
      GO TO 45                                                          REFA0115
   35 LMARK=INDEX                                                       REFA0120
   40 CONTINUE                                                          REFA0125
   45 ASSIGN 10 TO LSWTCH                                               REFA0130
C                                                                       REFA0135
C  NOW LH PARENTHEISS IS AT LMARK AND RH PARENTHEISI IS AR MARK,PICK    REFA0140
C  OFF EVERYTHING IN BETWEEN                                            REFA0145
      ISTART=LMARK +1                                                   REFA0150
      ISTOP=MARK-1                                                      REFA0155
      IF(ISTOP-ISTART-6)110,50,50                                       REFA0160
   50 II=0                                                              REFA0165
      DO 55 I=ISTART,ISTOP                                              REFA0170
      II=II+1                                                           REFA0175
   55 ISTICK(II)=INPUT(I)                                               REFA0180
      IQUIT=II                                                          REFA0185
C                                                                       REFA0190
C  FACTOR EVERYTHING BETWEEN LAMRK AND MARK                             REFA0195
      CALL FACTOR(ISTICK,IQUIT,NSPECI,IWORK,LMAX,ISPRED)                REFA0200
      IF(LMAX) 60,110,60                                                REFA0205
C                                                                       REFA0210
C  PUT FACTORED VERSION (IWORK) INTO INPUT BETWEEN LMARK AND MARK       REFA0215
   60 IHAVE =MARK-LMARK-1                                               REFA0220
      IF(ISPRED - IHAVE)65,100,85                                       REFA0225
C                                                                       REFA0230
C  TOO MUCH ROOM BETWEEN LMARK AND MARK,COLLAPSE SOME                   REFA0235
   65 IMOVE=IHAVE-ISPRED                                                REFA0240
      II=MARK-IMOVE                                                     REFA0245
      INPUT(II)=INPUT(MARK)                                             REFA0250
      INPUT(MARK)=0                                                     REFA0255
      ISTOP=LENGTH-MARK                                                 REFA0260
      IF(ISTOP)165,80,70                                                REFA0265
   70 DO 75 I=1,ISTOP                                                   REFA0270
      JJ=II+I                                                           REFA0275
      I2=MARK+I                                                         REFA0280
      INPUT(JJ)=INPUT(I2)                                               REFA0285
   75 INPUT(I2)=0                                                       REFA0290
```

```
   80 LENGTH=LENGTH-IMOVE                                                REFA0295
      GO TO 100                                                          REFA0300
C                                                                        REFA0305
C  NEED MORE ROOM,MOVE EVERYTHING TO RIGHT OF AND INCLUDING MARK TO      REFA0310
C  RIGHT ONE SPACE                                                       REFA0315
   85 ISTOP=LENGTH-MARK+1                                                REFA0320
   90 DO 95 I=1,ISTOP                                                    REFA0325
      II=LENGTH+1-I                                                      REFA0330
   95 INPUT(II+1)=INPUT(II)                                              REFA0335
      LENGTH=LENGTH+1                                                    REFA0340
      IHAVE=IHAVE+1                                                      REFA0345
      IF(ISPRED-IHAVE) 100,100,90                                        REFA0350
C                                                                        REFA0355
C  DONT HAVE TO MOVE THINGS AROUNG                                       REFA0360
  100 DO 105 I=1,ISPRED                                                  REFA0365
      II=LMARK+I                                                         REFA0370
  105 INPUT(II)=IWORK(I)                                                 REFA0375
      GO TO 10                                                           REFA0380
C                                                                        REFA0385
C  NEW SEARCH FOR )    )                                                 REFA0390
  110 DO 115 I=1,IQUIT                                                   REFA0395
  115 ISTICK(I)=0                                                        REFA0400
      IBLOCK=MARK                                                        REFA0405
      LMARK=MARK                                                         REFA0410
  120 IBEGIN = IBLOCK+1                                                  REFA0415
      DO 130 INDEX=IBEGIN,LENGTH                                         REFA0420
      IF(IABS(INPUT(INDEX))-1000)130,130,125                             REFA0425
  125 IF(INPUT(INDEX))135,175,175                                        REFA0430
  130 CONTINUE                                                           REFA0435
      GO TO 175                                                          REFA0440
  135 MARK=INDEX                                                         REFA0445
C                                                                        REFA0450
C  NOW ) IS AT LMARK AND ) IS AT MARK,PICK OFF EVERYTHING IN BETWEEN     REFA0455
      IF(MARK-LMARK-6)140,140,145                                        REFA0460
  140 LMARK=MARK                                                         REFA0465
      IBLOCK=MARK                                                        REFA0470
      GO TO 120                                                          REFA0475
C                                                                        REFA0480
C  NOT ENOUGH ROOM TO FACTOR                                             REFA0485
  145 ISTART=LMARK+1                                                     REFA0490
      ISTOP=MARK-1                                                       REFA0495
      II=0                                                               REFA0500
      DO 150 I=ISTART,ISTOP                                              REFA0505
      II=II+1                                                            REFA0510
  150 ISTICK(II)=INPUT(I)                                                REFA0515
      IQUIT=II                                                           REFA0520
      CALL FACTOR(ISTICK,IQUIT,NSPECI,IWORK,LMAX,ISPRED)                 REFA0525
      IF(LMAX) 160,155,160                                               REFA0530
C                                                                        REFA0535
C  CANT FACTOR )    ) GO BACK AND SEARCH FOR NEXT )    )                 REFA0540
  155 IBLOCK=MARK                                                        REFA0545
      LMARK=MARX                                                         REFA0550
      GO TO 110                                                          REFA0555
C                                                                        REFA0560
C  PUT FACTORED PART IN INPUT BETWEEM LMARK AND MARK                     REFA0565
  160 GO TO 60                                                           REFA0570
  165 WRITE(6,170)                                                       REFA0575
  170 FORMAT(27H ERROR IN SUBROUTINE REFACT)                             REFA0580
  175 RETURN                                                             REFA0585
      END                                                                REFA0590
```

```
$IBFTC FACTOR  LIST                                                    FACT0000
      SUBROUTINE FACTOR(INPUT,LENGTH,NSPECI,IFACT,LMAX,ISPRED)         FACT0005
      DIMENSION INPUT(500),IFACT(500),ICHECK(500),IMESS(500)           FACT0010
C                                                                      FACT0015
C  SUBROUTINE FACTORS OUT LMAX, THE SPECIE OCCURRING MOST OFTEN; FROM  FACT0020
C  INPUT(1) TO INPUT(LENGTH).   FACTORED VERSION IS RETURNED IN IFACT(1)FACT0025
C  TO IFACT(ISPRED).                                                   FACT0030
C  NSPECI=NO. OF SPECIES                                               FACT0035
C  LENGTH L.T. 500                                                     FACT0040
C  NSPECI L. T. 100                                                    FACT0045
      I5=NSPECI+4                                                      FACT0050
      DO 5 I=5,I5                                                      FACT0055
    5 ICHECK(I)=0                                                      FACT0060
      LCOUNT=0                                                         FACT0065
C                                                                      FACT0070
C  COUNT HOW MANY TIMES EACH SPECIES IS USED                           FACT0075
      INDEX=1                                                          FACT0080
   10 IF(INPUT(INDEX)) 15,70,15                                        FACT0085
C                                                                      FACT0090
C  IF HIT ZERO GET OUT                                                 FACT0095
   15 IF(IABS(INPUT(INDEX))-4)20,20,185                                FACT0100
   20 IF(IABS(INPUT(INDEX))-2)40,40,25                                 FACT0105
   25 ISTOP=IABS(INPUT(INDEX))                                         FACT0110
C                                                                      FACT0115
C  CHECK FOR REPEATED SPECIES IN TERM                                  FACT0120
      DO 35 I=2,ISTOP                                                  FACT0125
      II=INDEX+I                                                       FACT0130
      DO 35 J=2,ISTOP                                                  FACT0135
      JJ=INDEX+J                                                       FACT0140
      IF(I-J)30,35,30                                                  FACT0145
   30 IF(INPUT(II)-INPUT(JJ))35,60,35                                  FACT0150
   35 CONTINUE                                                         FACT0155
      GO TO 50                                                         FACT0160
C                                                                      FACT0165
C  NO REPEATED SPECIES IN TERM                                         FACT0170
   40 IF(IABS(INPUT(INDEX))-1)185,55,45                                FACT0175
   45 I=INPUT(INDEX+2)                                                 FACT0180
      ICHECK(I)=ICHECK(I)+1                                            FACT0185
      GO TO 65                                                         FACT0190
   50 DO 55 I=2,ISTOP                                                  FACT0195
      JJ=INDEX+I                                                       FACT0200
      II=INPUT(JJ)                                                     FACT0205
   55 ICHECK(II)=ICHECK(II)+1                                          FACT0210
      GO TO 65                                                         FACT0215
C                                                                      FACT0220
C  REPEATED SPECIES IS II FROM STATEMENT 4                             FACT0225
   60 II=INPUT(JJ)                                                     FACT0230
      ICHECK(II)=ICHECK(II)-1                                          FACT0235
      GO TO 50                                                         FACT0240
C                                                                      FACT0245
C  RESET INDEX TO HIT NEXT TERM                                        FACT0250
   65 INDEX=INDEX+ IABS(INPUT(INDEX))+1                                FACT0255
      IF(INDEX-LENGTH) 10,10,70                                        FACT0260
   70 LENGTH=INDEX-1                                                   FACT0265
C                                                                      FACT0270
C  NOW THE ICHECK(L) CONTAINS THE NUMBER OF TIMES THE SPECIES L OCCURS FACT0275
C  IN LOCATIONS INPUT(1) TO INPUT(LENGTH)                              FACT0280
C                                                                      FACT0285
C  SCAN THE ICHECK TO SEE IF FACTORABLE                                FACT0290
```

```
      NSTOP=NSPECI+4                                                    FACT0295
      DO 75 I=5,NSTOP                                                   FACT0300
      IF(ICHECK(I)-2)75,80,80                                           FACT0305
   75 CONTINUE                                                          FACT0310
                                                                        FACT0315
C  CANNOT BE FACTORED                                                   FACT0320
      LMAX=0                                                            FACT0325
      GO TO 195                                                         FACT0330
                                                                        FACT0335
C  SCAN THE ICHECK TO FIND LMAX,THE SPECIE OCCURRING MOST OFTEN         FACT0340
   80 DO 90 I=5,NSTOP                                                   FACT0345
      IF(ICHECK(I)-LCOUNT)90,90,85                                      FACT0350
   85 LMAX=I                                                            FACT0355
      LCOUNT=ICHECK(I)                                                  FACT0360
   90 CONTINUE                                                          FACT0365
                                                                        FACT0370
C  FACTOR AND ARRANGE PARENTHESIS                                       FACT0375
      IFACT(1)=LMAX                                                     FACT0380
      IFACT(2)=1001                                                     FACT0385
      IARROW=1                                                          FACT0390
      ISPEAR=3                                                          FACT0395
      INDEX=1                                                           FACT0400
   95 IF(INPUT(INDEX))100,175,100                                       FACT0405
  100 IF(IABS(INPUT(INDEX))-4)105,105,185                               FACT0410
  105 IF(IABS(INPUT(INDEX))-1)185,120,110                               FACT0415
  110 ISTOP=IABS(INPUT(INDEX))                                          FACT0420
      DO 115 I=2,ISTOP                                                  FACT0425
      II=INDEX+I                                                        FACT0430
      IF(INPUT(II)-LMAX)115,135,115                                     FACT0435
  115 CONTINUE                                                          FACT0440
      GO TO 125                                                         FACT0445
  120 ISTOP=1                                                           FACT0450
                                                                        FACT0455
C  NO LMAX IN THIS SERIES OF SPECIES,SHIP SERIES TO IMESS               FACT0460
  125 IMESS(IARROW)=INPUT(INDEX)                                        FACT0465
      DO 130 I=1,ISTOP                                                  FACT0470
      IARROW=IARROW+1                                                   FACT0475
      II=INDEX+I                                                        FACT0480
  130 IMESS(IARROW)=INPUT(II)                                           FACT0485
      IARROW=IARROW+1                                                   FACT0490
      GO TO 170                                                         FACT0495
                                                                        FACT0500
C  LMAX IS IN THIS SERIES, COLLAPSE SERIES INTO IFACT                   FACT0505
  135 IDUM=IABS(INPUT(INDEX))-1                                         FACT0510
      IFACT(ISPEAR)=ISIGN(IDUM,INPUT(INDEX))                            FACT0515
      ISPEAR=ISPEAR+1                                                   FACT0520
  140 IFACT(ISPEAR)=INPUT(INDEX+1)                                      FACT0525
      IDROP=0                                                           FACT0530
      DO 165 I=2,ISTOP                                                  FACT0535
      II=INDEX+I                                                        FACT0540
      IF(IDROP)150,145,150                                              FACT0545
  145 IF(INPUT(II)-LMAX)150,160,150                                     FACT0550
  150 ISPEAR=ISPEAR+1                                                   FACT0555
  155 IFACT(ISPEAR)=INPUT(II)                                           FACT0560
      GO TO 165                                                         FACT0565
  160 IDROP=1                                                           FACT0570
  165 CONTINUE                                                          FACT0575
      ISPEAR=ISPEAR+1                                                   FACT0580
                                                                        FACT0585
```

```
C  RESET INDEX                                                    FACT0590
  170 INDEX=INDEX+IABS(INPUT(INDEX))+1                            FACT0595
      IF(INDEX-LENGTH) 95,95,175                                  FACT0600
C                                                                 FACT0605
C  SET RIGHT PARENTHESIS                                          FACT0610
  175 IFACT(ISPEAR)=-1001                                         FACT0615
C                                                                 FACT0620
C  NOW SLAP IMESS ON REAR OF IFACT STARTING AT ISPEAR             FACT0625
      IARROW=IARROW-1                                             FACT0630
      DO 180 I=1,IARROW                                           FACT0635
      II=ISPEAR+I                                                 FACT0640
  180 IFACT(II)=IMESS(I)                                          FACT0645
      ISPRED=ISPEAR+IARROW                                        FACT0650
      GO TO 195                                                   FACT0655
  185 WRITE(6,190)                                                FACT0660
  190 FORMAT(27H ERROR IN SUBROUTINE FACTOR)                      FACT0665
  195 RETURN                                                      FACT0670
      END                                                         FACT0675
```

Subroutine OUT takes the highly factored version of the removal or formation reactions as required and prepares the hollorith output characters for each term of the factored equation. The species are coded as Y(X) and the rate constants as C(Z). Each term of the series is identified by a factorizing parenthesis as a species or a rate constant. Into the first word of the OUTPUT region is stored an equal sign. Whenever a factorizing parenthesis is found, be it open or closed, the BCD character for the open or closed parenthesis is stored in the next word of the OUTPUT region. The actual insertion of the BCD equivalent of the binary term is inserted between the parentheses of Y and C by subroutine CODER. This latter subroutine also inserts the BCD characters for addition (+) and for multiplication (*) into OUTPUT words as required.

Upon exit from OUT, the FORM or REMV terms preceeded by or followed by proper BCD arithmetic symbols are located in the OUTPUT region. Each word of this region contains one BCD coded quantity. It can be either a species Y, a rate constant C, an open or closed parenthesis, or a plus sign. The following statements, beginning on page B14, are a listing of the FORTRAN statements for subroutine OUT.

Subroutine CODER is called by OUT and sets up the BCD character for the information fed to it by OUT. OUT calls this subroutine if the word to be coded is a species or a rate constant, tells the subroutine if it is a species or a rate constant and whether the species should have an asterisk preceeding it or following it in the FORTRAN code. Upon return to OUT this BCD coded word is stored in the next word of the OUTPUT region.

The following statements, beginning on page B15, are a listing of the MAP code and can be used on an IBM-7044 as well as on an IBM-7094.

```
$IBFTC OOU        LIST
      SSUBROUTINE COUT(INPUT,LENGTH,JJARROW)                                    COUT00005
      COMMON OOUTPUT(200)                                                       COUT00010
      DIMENSION IINPUT(65C0),OUT(3)                                             COUT00015
      DATA OO/66HO(      ,66HY(      ,66H=       ,66H)       ,66H(       ,66H+       ,66H **
     1    //
      IARROW=1                                                                  COUT00030
      OOUTPUT(1)=OUT(3)
      IF(LENGTH .EQ. 2) GOTO 880                                                COUT00045
      INDEX=1                                                                   COUT00050
   55 IARROW=IARROW+1                                                           COUT00055
      IF(IABS(IINPUT(INDEX))-11)100,225,225,210                                 COUT00060
  100 IF(IINPUT(INDEX))105,885,220                                              COUT00065
  105 OOUTPUT(IARROW)=OUT(1)
      GOTO 720                                                                  COUT00080
  220 OOUTPUT(IARROW)=OUT(2)
      GOTO 720                                                                  COUT00095
  225 IF(IABS(IINPUT(INDEX))-44)830,830,855                                     COUT00100
  830 IIGOTO=IINPUT(INDEX+1)                                                    COUT00105
  835 OOUTPUT(IARROW)=OO(1)
      IARROW=IARROW+1                                                           COUT00120
      OMASK=OO(3)
      JJJ=1                                                                     COUT00140
      CALL CCODER (IIGOTO,OMASK,JJJ)                                            COUT00145
      OOUTPUT(IARROW)=OMASK                                                     COUT00150
      IF(IABS(IINPUT(INDEX))-11) 885,440,445                                    COUT00155
  440 INDEX=INDEX+1                                                             COUT00160
      GOTO 720                                                                  COUT00165
  445 ISTOP=IABS(IINPUT(INDEX))                                                 COUT00170
      GOTO 550 II=2,ISTOP                                                       COUT00175
      IARROW=IARROW+1                                                           COUT00180
      III=INDEX+1                                                               COUT00185
      IIGOTO=IINPUT(III)-44                                                     COUT00190
      OMASK=OO(2)
      JJJ=2                                                                     COUT00195
      CALL CCODER (IIGOTO,OMASK,JJJ)                                            COUT00200
      OOUTPUT(IARROW)=OMASK                                                     COUT00205
  550 CONTINUE                                                                  COUT00210
      INDEX=INDEX+IABS(IINPUT(INDEX))                                           COUT00215
      GOTO 720                                                                  COUT00220
  555 IF(IINPUT(INDEX+1))560,885,565
  560 OOUTPUT(IARROW)=OO(7)
      IARROW=IARROW+1                                                           COUT00235
  565 IIGOTO=IINPUT(INDEX)-4                                                    COUT00240
      OMASK=OO(2)
      JJJ=3                                                                     COUT00250
      CALL CCODER(IIGOTO,OMASK,JJJ)                                             COUT00255
      OOUTPUT(IARROW)=OMASK                                                     COUT00260
  770 INDEX=INDEX+1                                                             COUT00265
      IF(INDEX-LENGTH) 555,575,775                                              COUT00270
  775 JJARROW=IARROW                                                            COUT00275
      GOTO 895                                                                  COUT00280
  850 JJARROW=1                                                                 COUT00285
      GOTO 895                                                                  COUT00290
  885 WRITE(6,890)                                                              COUT00295
  890 FFORMAT(24H ERROR IN SSUBROUTINE OUT)                                     COUT00300
  895 RETURN                                                                    COUT00305
      END                                                                       COUT00310
```

```
SSBOMAP CCOOBR
CCOOBR  SSAVE   1122
        CCLA    3344
        SSTA    CCOOBA
        CCLA    4444
        SSTA    CCOOEB
        SSTA    CCOOED
        SSTA    CCOOEE
        SSTA    CCOOE6
        SSTA    CCOOB3
        SSTA    CCOOE7
        SSTA    CCOO64
        SSTA    CCOOE5
        CCLA    5544
        SSTA    CCOOEC
        SSTZ    TTEMP
CCOOEO  CCAL    ***
        AANA    MMASK
CCOOEE  SSLh    ***
CCOOBA  CCLA    ***
        TESX    .COEC/VA44
        LLCL    118
        LXXA    TTHREE,1
        CCLA    ZZERO
CCOOE1  LLCL    66
        TTNZ    **E2
        TTIX    CCOOE1,1,1
        LLCR    118
        XXCA
CCOOEB  CCRS    ***
        SSXA    TTEMP,1
        CCLA    FEQUR
        SSUB    TTEMP
        FPAX    CO,1
        CCLA    PPARAN
CCOOE2  TTIX    **E2,1,1
        TTRA    CCOOE3
        AABS    56
        AADD    #BLANK
        TTRA    CCOOE2
CCOOE3  CCRS    ***
CCOOEC  CCLA    ***
        FPAX    CO,1
        TTRA    ***4,1
        TTRA    CCOOE5
        TTRA    CCOOE4
CCOOE7  CCLA    ***
        TTRA    CCOOE6
CCOOE4  LDQ     ***
        CCLA    =MSTAR
        LLCR    56
        XXCA
        TTRA    CCOOE6
CCOOE5  CCLA    ***
        AADD    FEQUR
CCOOE6  STO     ***
        FRETURN CCOOER
TTHREE  OCT     CCCCCCCCCCC003
ZZERO   OCT     CCCCCCCCCCC000
FEQUR   OCT     CCCCCCCCCCC004
PPARAN  OCT     CCCCCCCCCC034
#BLANK  OCT     CCCCCCCCCC60
=MSTAR  OCT     CCCCCCCCCC054
MMASK   OCT     /7777CCCCCC00
TTEMP   BSS     11
        END
```

Upon exit from subroutine OUT, all of the terms of the given series are BCD coded and stored in the OUTPUT region. Since each word of this region contains one coded species, one rate constant, one equal sign, one plus sign, or an open or closed parenthesis, each word will contain some imbedded blank characters. The purpose of subroutine DECOD is to scan the OUTPUT region and remove any blank characters it finds and to collapse the remaining valid characters.

On exit from DECODE the required BCD coded equation is sitting in the ID region devoid of all imbedded blank characters and is ready to be printed and punched.

The following statements, beginning on page B17, are a listing of the MAP code for this subroutine and is acceptable to either the IBM-7044 or IBM-7094 computer.

```
$IBMAP DECODE
DECODE SAVE   1,2
       CLA    4,4
       STA    DECOD4
       CLA    5,4
       STA    **+1
       CLA    **
       PAX    0,1
       ADD    3,4
       STA    DECOD1
       CLA    6,4
       STA    DECOD8
       CLA    MASK
       STO    DECOD2
       STZ    COUNT
       LXA    SIX,2
       LXA    SIX,4
DECOD1 LDQ    **,1
       LGL    6
       CAS    BLANK
       TRA    **+2
       TRA    DECOD3
DECOD2 ALS    30
DECOD4 ORS    **
       CLA    DECOD2
       SUB    SIX
       STO    DECOD2
       TIX    DECOD3,2,1
       CLA    COUNT
       ADD    ONE
       STO    COUNT
       CLA    DECOD4
       SUB    ONE
       STO    DECOD4
       CLA    MASK
       STO    DECOD2
       LXA    SIX,2
DECOD3 CLA    ZERO
       TIX    DECOD1+1,4,1
       LXA    SIX,4
       TIX    DECOD1,1,1
       CLA    DECOD2
       STO    DECOD5
       CLA    DECOD4
       STO    DECOD6
DECOD7 CLA BLANK
DECOD5 ALS    **
DECOD6 ORS    **
       CLA    DECOD5
       SUB    SIX
       STO    DECOD5
       TIX    DECOD7,2,1
       CLA    COUNT
       ADD    ONE
DECOD8 STO    **
       RETURN DECODE
MASK   ALS    30
SIX    DEC    6
ONE    DEC    1
BLANK  OCT    000000000060
ZERO   OCT    000000000000
COUNT  BSS    1
       END
```

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY *(Corporate author)*
Hq AFCRL, OAR (CRU)
United States Air Force
Bedford, Massachusetts 01730

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP

3 REPORT TITLE
A Technique for Solving the General Reaction-Rate Equations in the Atmosphere

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Scientific Report. Interim.

5. AUTHOR(S) *(Last name, first name, initial)*
KENESHEA, Thomas J.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
April 1967	149	19

8a. CONTRACT OR GRANT NO.

b. PROJECT AND TASK NO. 8605-02

c. DOD ELEMENT 61445014

d. DOD SUBELEMENT 631310

9a. ORIGINATOR'S REPORT NUMBER(S)
AFCRL-67-0221
ERP No. 263

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* AFCRL-67-0221

10 AVAILABILITY/LIMITATION NOTICES
Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
Hq AFCRL, OAR (CRU)
United States Air Force
Bedford, Massachusetts 01730

13. ABSTRACT

With the availability of numerical techniques for solving an extensive set of nonlinear differential equations and high-speed computers for performing the calculations, interest in solving the unrestricted reaction-rate equations is growing among ionospheric researchers. In view of this, the author has continued to refine the techniques that he previously developed.

The computer code, as discussed here, is written to solve the photochemical behavior of 15 atmospheric species; these species are electrons, O^-, O_2^-, O_3^-, NO_2^-, O^+, O_2^+, N_2^+, NO^+, NO, N, NO_2, O_3, N_2O, and O. Built into the code are 168 reactions that can conceivably take place among these constituents. Several examples of the results obtained using the code are presented, including the buildup of ionization from zero concentrations at altitudes in the D and E regions and the deionization of an atmosphere with high initial electron densities. The diurnal variation of the atmospheric constituents is also presented along with profiles for the above-mentioned species from 60 km to 120 km.

The computer codes are included in their entirety with complete explanations on their usage.

DD FORM 1 JAN 64 1473

Unclassified
Security Classification

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Reaction rates						
Ionosphere						
Differential equations solution						
Diurnal variation						
Deionization						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year, or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U).*

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

Unclassified
Security Classification