08/07/80
2+D NTIS M's

UCID-18717

MASTER

DYNAVAC: A TRANSIENT-VACUUM-NETWORK ANALYSIS CODE

Gary A. Deis

July 8, 1980

Lawrence Livermore Laboratory

Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore Laboratory under Contract W-7405-Eng-48.

DYNAVAC: A TRANSIENT-VACUUM-NETWORK ANALYSIS CODE

ABSTRACT

This report discusses the structure and use of the program DYNAVAC, a new transient-vacuum-network analysis code implemented on the NMFECC CDC-7600 computer. DYNAVAC solves for the transient pressures in a network of up to twenty lumped volumes, interconnected in any configuration by specified conductances. Each volume can have an internal gas source, a pumping speed, and any initial pressure. The gas-source rates can vary with time in any piecewise-linear manner, and up to twenty different time variations can be included in a single problem. In addition, the pumping speed in each volume can vary with the total gas pumped in the volume, thus simulating the saturation of surface pumping. This report is intended to be both a general description and a user's manual for DYNAVAC.

INTRODUCTION

The experimental magnetic fusion energy program frequently presents one with the problem of designing, for a new experiment, a vacuum system which maintains a low pressure in the vicinity of the plasma while large amounts of gas are being injected into the vacuum via neutral beams and other gas sources. In the past, these design problems have been treated in basically two ways.

The first method is to assume that the vacuum system reaches equilibrium during the physics experiment, and design on that assumption. This has been done most frequently at LLL using the computer code GASBAL, written by Hoffman, Roose, and Carlson.[1] GASBAL performs a steady-state solution to obtain the pressures in the various volumes within the system, taking into account the influence of the plasma itself, direct streaming gas from neutral beams, both high and low energy gas, and many other factors (more than 28 in all). However,

---

1. T. R. Roose, M. A. Hoffman, and G. A. Carlson, "Analysis of the Steady-State Operation of Vacuum Systems for Fusion Machines," in *Proc. of the Sixth Symposium on Engineering Problems of Fusion Research*, San Diego, CA, Nov. 18-21, 1975, IEEE Publication number 75CH1097-5-NPS (1976).

GASBAL assumes a geometry in which pairs of volumes in series radiate out from a single, central, plasma chamber. Although (with some creativity on the part of their designers) many systems can be modelled in this general manner, tandem mirrors do not seem to be among them. In addition, experimental information from TMX indicates that the vacuum does not reach equilibrium during the experiment, which means that an assumption of conditions steady-state could lead to over-design of the vacuum system.

A second method of attacking the general problem is by solving the system differential equations manually. If a lumped model is assumed, the equations are straightforward but extremely tedious to solve, especially for large numbers of lumped volumes. Furthermore, in the design process where changes are frequent, many such solutions (each one different) are required. One tool which is currently available to help in this approach is the computer code DIP3.[2] This code is an interactive solver for a three-volume, differential-pumping system. The three volumes are required to be in series, and there can be only one gas source, located in the first volume. The output of this code consists of the three pressures and the flow of gas out of the third volume, at requested time steps. DIP3 produces exact results, since it merely supplies the input quantities to an extant general three-volume solution obtained by standard analytical means. This code has been useful in estimating saturation times for pumping surfaces within vacuum systems. Again, the geometry is restrictive, and although DIP3 is an extremely convenient method for handling simple problems, it cannot be used effectively for large, many-volume systems.

To solve this general vacuum system design problem satisfactorily, a tool is required that provides transient solutions to the general system equations for a large number of volumes, interconnected in any way. It must allow for variations in the properties of the system in order to account for time-variable gas sources and saturation of surface pumping, and it must be convenient and easy to use. Out of these requirements the code DYNAVAC was created. It has been constructed to be used easily in the design process, just as codes are used for stress, magnetic-field, and electric-field design. Furthermore, it is constructed in a modular fashion which will allow special features to be added as they are required.

---

2. L. Pittenger, "Analytic Solution for the Transient Response of 3-Stage Differentially-Pumped Vacuum Systems in the Molecular Flow Regime," Lawrence Livermore Laboratory Engineering Note ENE 75-4 (January 1975).

DESCRIPTION

DYNAVAC solves the general vacuum-network problem of N (≤ 20) lumped volumes interconnected in any configuration by conductances. In general, each lumped volume has four properties: volume (V), pumping speed (S), gas source rate (Q) and pressure (P). In addition, each volume can be connected to any other volume via specified conductances, which need not be the same in both directions; that is, the conductance from volume i to volume j need not be the same as that from volume j to volume i. With this information, one can write the continuity equation for volume i, to yield the general, first order, ordinary differential equation (ODE) for a volume:

$$V_i \frac{dP_i}{dt} = Q_i - S_iP_i - \sum_{j=1}^{N} (C_{ij} P_i - C_{ji}P_j) ,$$

where $C_{ii} \equiv 0$.

This equation indicates simply that the rate of increase of gas in the volume is equal to the rate at which it is added to the volume by a source within the volume, minus the rate at which it is pumped within the volume, minus the net rate at which it is conducted away to all other volumes. For a general system of N connected volumes, there are N similar equations, each containing N unknowns, the pressure in each of the volumes.

Within DYNAVAC, this system of first order ODE's is solved by LSODE, a solver descended from GEAR, and written by Alan C. Hindmarsh.[3] This powerful ODE solver is flexible enough that variations in the system properties can be made quite easily. For the current version of DYNAVAC, only two such variations have been incorporated. The first, time-variable gas source rates, is intended to allow simple simulation of neutral-beam sources and other gas loads, which may increase or decrease during the experiment. The second, variable pumping speeds, is intended to model the saturation effect of pumping surfaces. This is accomplished by permitting the pumping speed in each volume to vary with the total amount of gas pumped in that volume:

---

[3] A. C. Hindmarsh, *LSODE-Livermore Solver for Ordinary Differential Equations*, Lawrence Livermore Laboratory report in preparation.

$$S_i(t) = f\left(\int_{to}^{t} S_i \; P_i \; dt\right) .$$

The LSODE package performs all control of numerical error in the solution, under the condition that the error *per solver step* in P_i, ΔP_i is

$$\Delta P_i \leq 1 \times 10^{-10} + (1 \times 10^{-4}) \; P_i .$$

This provides both absolute (1×10^{-10}) and relative $\left[(1 \times 10^{-4}) P_i\right]$ error control, and in testing it has maintained at least three significant figures of accuracy in the output. It should be noted, however, that in problems having pressures on the order of 10^{-7}, results may be inaccurate. These parameters can be adjusted by changes in subroutine BAG, but it must be noted that such changes will influence the time required for solution.

AVAILABILITY

The executable code DYNAVAC, the source file DYNAVACS, and the input file for the sample problem in this report, DYNAVIN, are available by running FILEM as follows:

```
FILEM / t v
RDS . 3040 . VACUUM DYNAVAC DYNAVACS DYNAVIN
```

Before running the sample problem, the file name must be changed to VACIN, which can be done with utility routine switch SWITCH DYNAVIN VACIN / t v.

EXECUTION

DYNAVAC is run by simply executing the controllee DYNAVAC, and supplying an input file named VACIN. DYNAVAC outputs several lines of information to the terminal, concerning the code itself and the problem it is solving. It generates four different files which are left on disk. The printed output for the problem is in file VACPRT, along with an echo of the input information. The plotted output is left on disk with a file name of the form FX105/hhmm, where the last four digits (hhmm) are the time the file was generated. The two remaining files should be of little interest to the user. File SCRATCH contains the pressure vs time output data; this file could be used as input for other codes for various purposes. File VACERR contains information concerning the actual solution of the equations, including any messages sent by the solver LSODE. If DYNAVAC terminates during the solution of the equations, this information can be useful in determining the cause.

INPUT

All input to DYNAVAC is in the file VACIN, which must be present for DYNAVAC to execute successfully. If VACIN cannot be found, execution will be terminated by ORDERLIB ERROR 28*. All input is formatted as shown in square brackets below. Integer data must be right-justified and must not include a decimal. All other numerical data should be entered with a decimal point, unless the default input format (generally E10.3) is used; use of a decimal point will override the default format. If non-imbedded blanks are encountered, they are read as zeros. Numerical limits are indicated in parentheses below.

File VACIN is constructed as follows:

A. TITLE CARD [10A4]

Col 1-40 TITLE - up to 40 characters which will be output at the top of each page of output

B. MASTER CONTROL CARD [4I5]

Col 1-5 NTANK (≤20) - the number of lumped volumes in the system

Col 6-10 NCOND (≤380) - the number of conductance cards to be read

Col 11-15 NFUNQ (≤20) - the number of gas-source time functions (Q functions) to be entered (see Note 1)

Col 16-20 MODE - 0 or blank - for input-and-run mode
1 for data check mode

C. VOLUME CARDS - NTANK sets of cards required

1) VOLUME TITLE CARD [10A4]

Col 1-40 VTITLE - up to 40 characters to be output in the heading for the graph for the volume

2) **VOLUME DATA CARD** [4E10.3, 2I5]

Col 1-10 VOL (≥0.0) - the volume

Col 11-20 SPD (≥0.0) - the pumping speed (ignored if a speed table is used)

Col 21-30 Q (≥0.0) - the gas source rate

Col 31-40 P (≥0.0) - the initial pressure

Col 41-45 NPTS (≤20) - the number of points in the speed table (0 or blank for constant pumping speed)

Col 46-50 IFUNQ - the number of the Q function for this volume (0 or blank for constant gas source rate - see Note 1)

3) **SPEED TABLE CARDS** [2E10.3] - NPTS cards required (see Note 2)

Col 1-10 QINT (≥0.0) - total integrated gas pumped

Col 11-20 SPD (≥0.0) - the pumping speed at QINT

D. CONDUCTANCE CARDS [2I5, E10.3] - NCOND cards required

Col 1-5 I

Col 6-10 J

Col 11-20 C(I,J) (≥0.0) - The conductance from volume I to volume J. By default C(J,I) = C(I,J) unless both are input. If neither is input, the default is zero.

E. Q FUNCTION CARDS - **NFUNQ** sets of cards required

1) **Q FUNCTION CONTROL CARD** [I5]

Col 1-5 NPTQ (≤20) - the number of points in the Q function table

2) Q FUNCTION TABLE CARDS [2E10.3] - NPTQ cards required

Col 1-10 T - time

Col 11-20 FQ (≥0.0) - the Q function value at time T (see Note 1)

F. PROBLEM CONTROL CARDS - one and only one set required (see Note 3)

1) TIME BLOCK CONTROL CARD [I5, E10.3]

Col 1-5	NBLOCK ($\leq$10)	- the number of blocks of equally spaced time steps
Col 6-15	TO	- the initial time

2) TIME BLOCK CARDS [I5, E10.3] - NBLOCK cards required

Col 1-5	NSTEP	- the number of time steps in the block
Col 6-10	TSTEP	- the length of the time step

NOTES

1) Time-dependent gas-source rates are handled in the following manner if IFUNQ is greater than zero for a volume. (If IFUNQ is zero for a volume, the gas source is not time-dependent.) The gas source rate at any time is given by Q*FQ, where FQ, the Q function value, carried the time dependency. FQ is determined for a particular volume at a particular time by going to the Q function table for that volume and linearly interpolating between points. The first and last points of any table are extrapolated as constant values. Any number of volumes can use the same Q function, if it is desired to give them all the same time dependence.

2) Pumping speeds can be a function of the total integrated gas pumped in a volume, if NPTS is greater than zero. (If NPTS is equal to zero the pumping speed is constant.) The total integrated gas pumped in a volume is obtained by doing the stepwise integration

$$QINT = \sum Pi\ Si\ \Delta t ,$$

where Pi is the average pressure during the time step Δt in volume i. The pumping speed is then given as a function of QINT. The pumping speed at any time is determined by going to the volume's speed table with the current value of QINT, and linearly interpolating between points. The first and last points are extrapolated as constant values. If a speed table is used, SPD on the volume data card is ignored.

3) There must be less than a total of 1000 time steps per problem.

4) Any units can be used, but they must make a consistent set; that is, they must fit in the system equation correctly. A useful set is as follows: pressure in Torr, volume in liters, time in seconds, conductances and pumping speeds in liters per second, gas source rates in Torr-liters per second, integrated gas pumped in Torr-liters, and Q function unitless.

OUTPUT

TTY

When DYNAVAC is executed, three lines are normally sent to the user's TTY. The first line is information on the DYNAVAC version that is currently being used; it consists of eight integers separated by decimal points, each integer referring to the version number of one of the eight main DYNAVAC subroutines. If trouble is encountered in running a version of DYNAVAC just retrieved from public storage, check these numbers to see if changes have been made in the code itself. A further aid in this situation is the compilation date, also output at the TTY. Normally, the final line of output is the title line from the problem DYNAVAC is currently working on. However, if errors are encountered during execution, additional information will be output to the TTY, depending on the circumstances. If an error is found in the input data, it is printed on the TTY, and the rest of the input file is checked, regardless of how many errors are found.

Printed

File VACPRT contains the printed output from the problem. It consists of a formatted echo of the input file, followed by the pressure vs time data. If errors are encountered in the input, the entire input file is still checked, and information on errors is printed in VACPRT.

File VACERR contains information useful in the event of problems in the solution. For each output time step, VACERR contains the total number of steps taken internally by LSODE, the total number of evaluations of the system equations and the system Jacobian, and the last internal time step used successfully by LSODE. In addition, VACERR contains any messages sent by LSODE.

File SCRATCH contains only the pressure vs time data. This data may be useful as input for post-processing codes, but usually the file can be ignored.

Graphical

DYNAVAC produces FR80-format plotfiles with names of the form FX105/hhmm, where hhmm is the time the file was generated. These files can be output using NETPLOT with the F. option. If the L. option is also used, the plots will be of an appropriate size for an 8 1/2 × 11 page.

DYNVAC produces two types of graphs. The first set of graphs consists of the pressure vs time data for all volumes plotted to the same scale. In this set there are five plots per graph on as many graphs as are required. Following these plots are individual plots for each volume, each to an appropriate scale for that plot. The second set allows one to examine in detail the response of each volume on the individual plots and compare it to all volumes on the combined graphs.

The following example demonstrates many of the features of DYNAVAC. A schematic of the system model is shown in Fig. 1. This problem involves a TMX-like geometry, including all leakage paths from volume to volume as conductances. There are three gas sources: a gas box in the plug area, and neutral beams in the plug and center cell. In this example, the neutral beams operate simultaneously and the gas box comes on 0.005 seconds after them. Note that this is accomplished by using only two Q functions: the first for the gas box and the second for both neutral-beam clusters. Also note that although the Q function is the same for volumes 8 and 9, the actual gas loads are different (corresponding to 10 sources in the plug and 4 sources in the central cell) by means of the Q values entered on the volume data cards. For both the neutral beams and the gas box, the gas source is assumed to turn on instantly, remain on at a constant rate, and then ramp down to zero in 0.005 seconds (see Fig. 2).

In volumes 2, 3, and 5, the effect of saturation of surface pumping was modelled by using variable pumping speeds. It is assumed in all three cases that the pumping speed initially decreases linearly with QINT (the total gas pumped in the volume), and then levels off at a constant value (see Fig. 3). This can be entered with only two points in the speed table, since DYNAVAC interpolates between points, and extrapolates a constant value from the first and last points.

The time control is set up to output from time zero to 0.1 in steps of .001, from 0.1 to 0.3 in steps of .005, and from .3 to .5 in steps of .01.

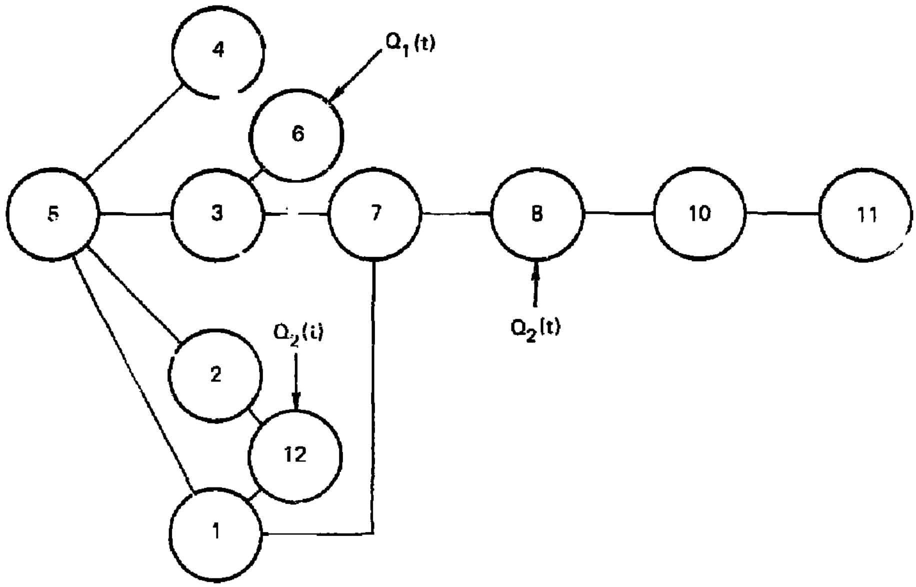


FIG. 1. Schematic of system model for example problem.

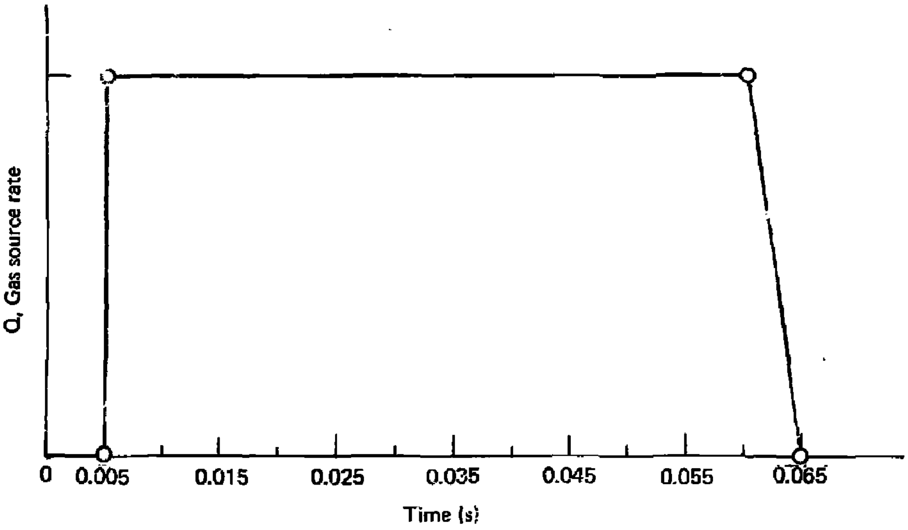


FIG. 2. Variation of gas-source rate for volume 6 with time. Circles denote input points.

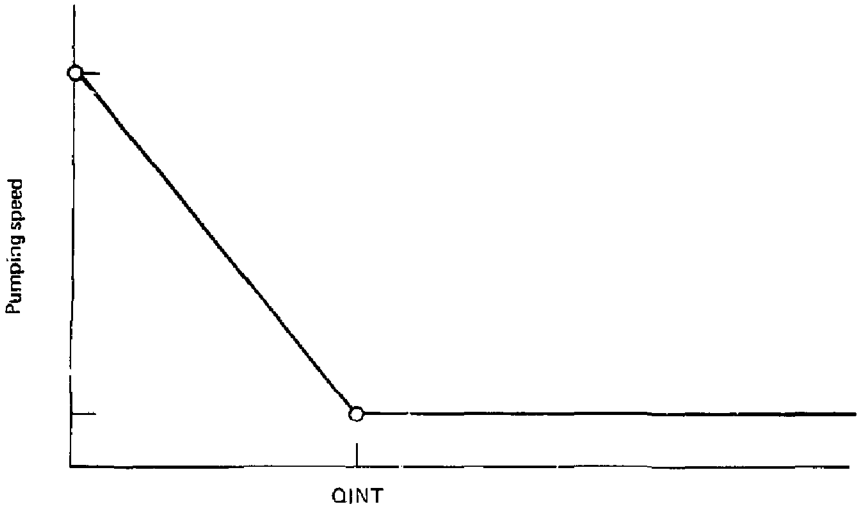


FIG. 3. Pumping speed characteristic for volumes 2, 3, and 5. Circles denote input points.

The units used in this example are Torr, liter, and seconds.

The input file, VACIN, for this example is shown in Fig. 4. Printed output from file VACPRT is shown in Fig. 5, a-c, with only one page of data included. The entire set of graphs produced for this problem is shown in Fig. 6, a-n. Finally, one page of printed output from file VACERR is shown in Fig. 7.

```
TEST CASE 8009 - COLD LINERS, GETTERING
  11   16   2    0
OUTSIDE OF OUTER LINER
   3000.0     5000.0       0.0    1.0E-07
OUTER ANNULUS
  18.2E+03                 0.0    1.0E-07    2
       0.0 5.511E+06
    7.6876 5.078E+05
INNER ANNULUS
  12.1E+03                 0.0    1.0E-07    2
       0.0 3.222E+06
    4.4947 3.622E+05
PLASMA CHAMBER
  0.92E+03  4.00E+04       0.0    1.0E-07    0    0
END FAN CHAMBER
  16.0E+03                 0.0    1.0E-07    2    0
       0.0 2.361E+06
     3.293 2.179E+05
GAS BOX
      13.0         0.0    10.0    1.0E-07    0    1
EAST FIXED DOME
  8.50E+03   3.1E+05       0.0    1.0E-07
CENTER CELL TANK
  6.50E+03   3.2E+05     120.0    1.0E-07    0    2
NEUTRAL BEAM SOURCES
  1.57E+03         0.0   300.0    1.0E-07    0    2
WEST FIXED DOME
  8.50E+03   3.1E+05       0.0    1.0E-07    0    0
DUMMY WEST TANK
  55.4E+03   16.E+05       0.0    1.0E-07    0    0
     1    2  9.77E+05
     1    5  2.87E+03
     1    7  2.50E+05
     2    3  4.41E+05
     2    5  2.80E+04
     3    4  3.15E+05
     3    5  3.20E+04
     3    6  1.10E+04
     3    7  7.88E+03
     4    5  4.10E+04
     6    7  1.10E+04
     7    8  6.62E+05
     8   10  6.62E+05
     9    1  9.46E+04
     9    2  7.94E+05
    10   11  2.69E+05
     4
   0.00499         0.0
     0.005         1.0
     0.055         1.0
     0.060         0.0
     2
     0.055         1.0
     0.060         0.0
     3      0.0
   100    0.001
    40    0.005
    20     0.01
```

FIG. 4. Input file (VACIN) for example problem.

```
1                     DYNAVAC OUTPUT
                ************************

          TEST CASE 8009 - COLD LINERS, GETTERING

          NUMBER OF VOLUMES       = 11
          NUMBER OF CONDUCTANCES = 16
          NUMBER OF Q FUNCTIONS   =  2

VOLUME  1 - OUTSIDE OF OUTER LINER
     VOLUME           =  3.000E+03
     INITIAL PRESSURE =  1.000E-07
     GAS SOURCE       =  0.
     PUMPING SPEED    =  5.000E+03

VOLUME  2 - OUTER ANNULUS
     VOLUME           =  1.820E+04
     INITIAL PRESSURE =  1.000E-07
     GAS SOURCE       =  0.
          PUMPING SPEED TABLE ( 2 POINTS)
             INT(S*P*DT)     SPEED
             -----------  -----------
               0.          5.511E+06
               7.688E+00   5.078E+05


VOLUME  3 - INNER ANNULUS
     VOLUME           =  1.210E+04
     INITIAL PRESSURE =  1.000E-07
     GAS SOURCE       =  0.
          PUMPING SPEED TABLE ( 2 POINTS)
             INT(S*P*DT)     SPEED
             -----------  -----------
               0.          3.222E+06
               4.495E+00   3.622E+05


VOLUME  4 - PLASMA CHAMBER
     VOLUME           =  9.200E+02
     INITIAL PRESSURE =  1.000E-07
     GAS SOURCE       =  0.
     PUMPING SPEED    =  4.000E+04

VOLUME  5 - END FAN CHAMBER
     VOLUME           =  1.600E+04
     INITIAL PRESSURE =  1.000E-07
     GAS SOURCE       =  0
          PUMPING SPEED TABLE ( 2 POINTS)
             INT(S*P*DT)     SPEED
             -----------  -----------
               0.          2.361E+06
               3.293E+00   2.179E+05

VOLUME  6 - GAS BOX
     VOLUME           =  1.300E+01
     INITIAL PRESSURE =  1.000E-07
     GAS SOURCE       =  1.000E+01
     Q FUNCTION       =          1
     PUMPING SPEED    =  0.



VOLUME  7 - EAST FIXED DOME
     VOLUME           =  8.500E+03
     INITIAL PRESSURE =  1.000E-07
     GAS SOURCE       =  0.
     PUMPING SPEED    =  3.100E+05

VOLUME  8 - CENTER CELL TANK
     VOLUME           =  6.500E+03
     INITIAL PRESSURE =  1.000E-07
     GAS SOURCE       =  1.200E+02
     Q FUNCTION       =          2
     PUMPING SPEED    =  3.200E+05

VOLUME  9 - NEUTRAL BEAM SOURCES
     VOLUME           =  1.570E+03
     INITIAL PRESSURE =  1.000E-07
     GAS SOURCE       =  3.000E+02
     Q FUNCTION       =          2
     PUMPING SPEED    =  0.

VOLUME 10 - WEST FIXED DOME
     VOLUME           =  8.500E+03
     INITIAL PRESSURE =  1.000E-07
     GAS SOURCE       =  0.
     PUMPING SPEED    =  3.100E+05

VOLUME 11 - DUMMY WEST TANK
     VOLUME           =  5.540E+04
     INITIAL PRESSURE =  1.000E-07
     GAS SOURCE       =  0.
     PUMPING SPEED    =  1.600E+06
```

(a)

FIG. 5. Printed output (VACPRT) for example problem.

```
     CONDUCTANCE MATRIX
     ------------------
0          9.770E+05 0.         0          2.870E+03 0          2.500E+05 0.         9.460E+04 0          0
9.770E+05  0.        4.410E+05  0          2.800E+04 0          0         0          7.940E+05 0.         0
0.         4.410E+05 0.         3.150E+05  3.200E+04 1.100E+04  7.880E+03 0.         0         0          0.
0          0.        3.150E+05  0.         4.100E+04 0          0         0          0         0          0
2.870E+03  2.800E+04 3.200E+04  4.100E+04  0         0.         0         0          0         0.         0.
0.         0.        1.100E+04  0.         0.        0.         1.100E+04 0          0         0          0
2.500E+05  0         7.880E+03  0          0.        1.100E+04  0.        6.620E+05  0         0          0.
0          0.        0          0          0.        0          6.620E+05 0          0         6.620E+05  0.
9.460E+04  7.940E+05 0.         0          0         0.         0.        0          0         0.         0.
0.         0.        0.         0.         0.        0.         0.        6.620E+05  0         0          2.690E+05
0.         0.        0.         0.         0.        0          0.        0          0         2.690E+05  0

     Q FUNCTION NUMBER  1. ( 4 POINTS)
             TIME           FQ
         ----------  ----------
          4.990E-03   0.
          5.000E-03   1.000E+00
          5.500E-02   1.000E+00
          6.000E-02   0.

     Q FUNCTION NUMBER  2. ( 2 POINTS)
             TIME           FQ
         ----------  ----------



          5.500E-02   1.000E+00
          6.000E-02   0.

     PROBLEM CONTROL
     ---------------
          NUMBER OF TIME BLOCKS  =  3
          INITIAL TIME           =  0.
                         NSTEP     TSTEP
                         -----  ----------
               BLOCK  1   100   1.000E-03
               BLOCK  2    40   5.000E-03
               BLOCK  3    20   1.000E-02
```

(b)

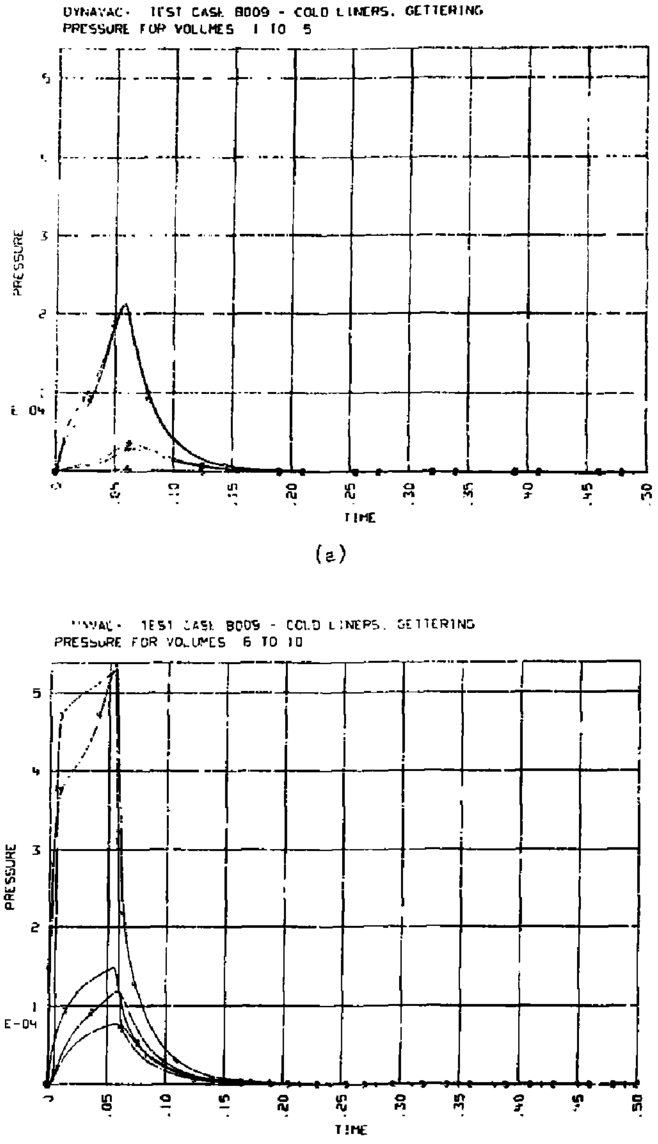
FIG. 5. (Continued)

```
1DYNAVAC-  TEST CASE 8009 - COLD LINERS, GETTERING

  TIME       P 1        P 2        P 3        P 4        P 5        P 6        P 7        P 8        P 9        P10
         P11        P
 0.         1.000E-07 1.000E-07 1.000E-07 1.000E-07 1.000E-07 1.000E-07 1.000E-07 1.000E-07 1.000E-07 1.000E-07
      1.000E-07
 1.000E-03 2.606E-06 3.140E-06 1.139E-07 9.571E-08 8.830E-08 2.285E-07 7.524E-07 1.645E-05 1.465E-04 7.275E-07
      9.820E-08
 2.000E-03 8.436E-06 9.259E-06 2.783E-07 1.177E-07 8.717E-08 8.523E-07 2.488E-06 2.936E-05 2.318E-04 2.320E-06
      1.019E-07
 3.000E-03 1.571E-05 1.589E-05 6.031E-07 2.027E-07 9.847E-08 1.997E-06 5.009E-06 3.976E-05 2.831E-04 4.535E-06
      1.148E-07
 4.000E-03 2.327E-05 2.203E-05 1.040E-06 3.667E-07 1.208E-07 3.548E-06 8.077E-06 4.830E-05 3.150E-04 7.130E-06
      1.387E-07
 5.000E-03 3.090E-05 2.736E-05 1.538E-06 6.060E-07 1.519E-07 9.207E-06 1.150E-05 5.548E-05 3.357E-04 9.940E-06
      1.748E-07
 6.000E-03 3.711E-05 3.191E-05 2.249E-06 9.283E-07 1.895E-07 3.791E-04 1.542E-05 6.164E-05 3.497E-04 1.205E-05
      2.235E-07
 7.000E-03 4.302E-05 3.577E-05 3.033E-06 1.355E-06 2.318E-07 4.481E-04 1.961E-05 6.705E-05 3.595E-04 1.578E-05
      2.844E-07
 8.000E-03 4.825E-05 3.909E-05 3.745E-06 1.842E-06 2.773E-07 4.640E-04 2.380E-05 7.190E-05 3.667E-04 1.869E-05
      3.574E-07
 9.000E-03 5.285E-05 4.201E-05 4.375E-06 2.347E-06 3.246E-07 4.688E-04 2.789E-05 7.629E-05 3.722E-04 2.154E-05
      4.416E-07
 1.000E-02 5.692E-05 4.458E-05 4.930E-06 2.840E-06 3.726E-07 4.715E-04 3.186E-05 8.032E-05 3.766E-04 2.431E-05
      5.365E-07
 1.100E-02 6.054E-05 4.694E-05 5.423E-06 3.307E-06 4.205E-07 4.730E-04 3.567E-05 8.404E-05 3.802E-04 2.698E-05
      6.414E-07
 1.200E-02 6.379E-05 4.914E-05 5.865E-06 3.742E-06 4.678E-07 4.759E-04 3.932E-05 8.750E-05 3.832E-04 2.955E-05
      7.553E-07
 1.300E-02 6.676E-05 5.124E-05 6.269E-06 4.143E-06 5.141E-07 4.779E-04 4.281E-05 9.073E-05 3.860E-04 3.202E-05
      8.774E-07
 1.400E-02 6.950E-05 5.326E-05 6.641E-06 4.514E-06 5.592E-07 4.798E-04 4.613E-05 9.377E-05 3.884E-04 3.439E-05
      1.007E-06
 1.500E-02 7.206E-05 5.524E-05 6.988E-06 4.856E-06 6.032E-07 4.816E-04 4.928E-05 9.662E-05 3.907E-04 3.665E-05
      1.143E-05
 1.600E-02 7.450E-05 5.725E-05 7.320E-06 5.174E-06 6.464E-07 4.834E-04 5.228E-05 9.932E-05 3.928E-04 3.881E-05
      1.286E-06
 1.700E-02 7.684E-05 5.927E-05 7.640E-06 5.473E-06 6.877E-07 4.850E-04 5.513E-05 1.019E-04 3.950E-04 4.087E-05
      1.433E-06
 1.800E-02 7.913E-05 6.132E-05 7.952E-06 5.757E-06 7.285E-07 4.869E-04 5.784E-05 1.043E-04 3.971E-04 4.283E-05
      1.586E-06
 1.900E-02 8.137E-05 6.342E-05 8.260E-06 6.029E-06 7.686E-07 4.880E-04 6.042E-05 1.065E-04 3.992E-04 4.470E-05
      1.742E-06
 2.000E-02 8.359E-05 6.556E-05 0.568E-06 6.293E-06 8.080E-07 4.894E-04 6.290E-05 1.087E-04 4.014E-04 4.648E-05
      1.902E-05
 2.100E-02 8.580E-05 6.777E-05 8.878E-06 6.552E-06 8.469E-07 4.909E-04 6.519E-05 1.108E-04 4.035E-04 4.818E-05
      2.065E-06
 2.200E-02 8.802E-05 7.006E-05 9.193E-06 6.809E-06 8.856E-07 4.921E-04 6.742E-05 1.127E-04 4.057E-04 4.980E-05
      2.231E-05
 2.300E-02 9.027E-05 7.244E-05 9.515E-06 7.066E-06 9.242E-07 4.933E-04 6.954E-05 1.145E-04 4.079E-04 5.134E-05
      2.398E-05
 2.400E-02 9.254E-05 7.490E-05 9.846E-06 7.326E-06 9.630E-07 4.945E-04 7.156E-05 1.163E-04 4.103E-04 5.280E-05
      2.567E-06
 2.500E-02 9.486E-05 7.747E-05 1.019E-05 7.589E-06 1.002E-06 4.957E-04 7.350E-05 1.180E-04 4.127E-04 5.420E-05
      2.738E-06
 2.600E-02 9.723E-05 8.015E-05 1.054E-05 7.859E-06 1.042E-06 4.968E-04 7.536E-05 1.196E-04 4.151E-04 5.552E-05
      2.909E-06
 2.700E-02 9.966E-05 8.294E-05 1.091E-05 8.136E-06 1.082E-06 4.979E-04 7.715E-05 1.211E-04 4.177E-04 5.679E-05
      3.080E-06
```

(c)

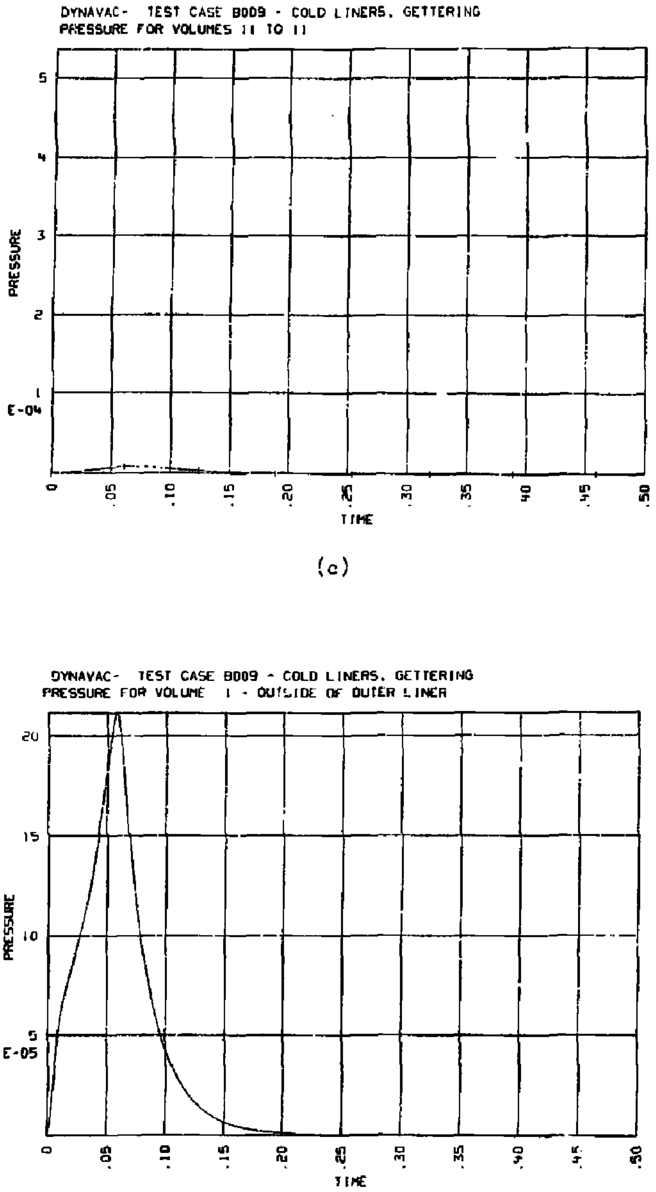
FIG. 5. (Continued)



(a)

(b)

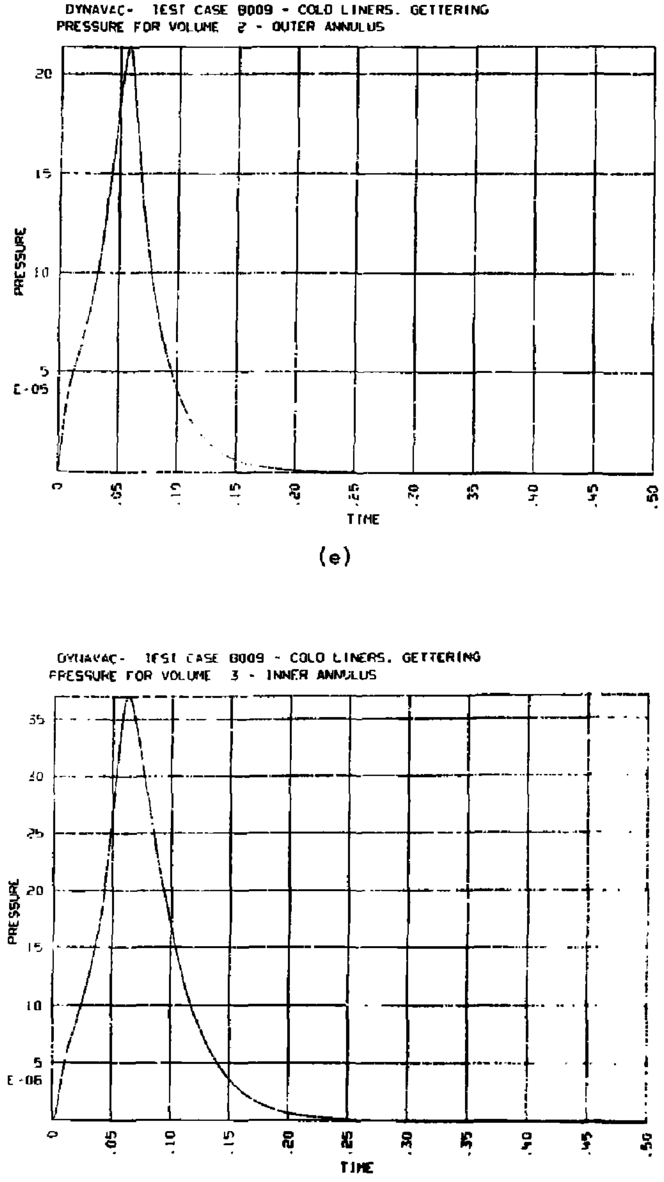
FIG. 6. Graphical output for example problem.



(c)

(d)

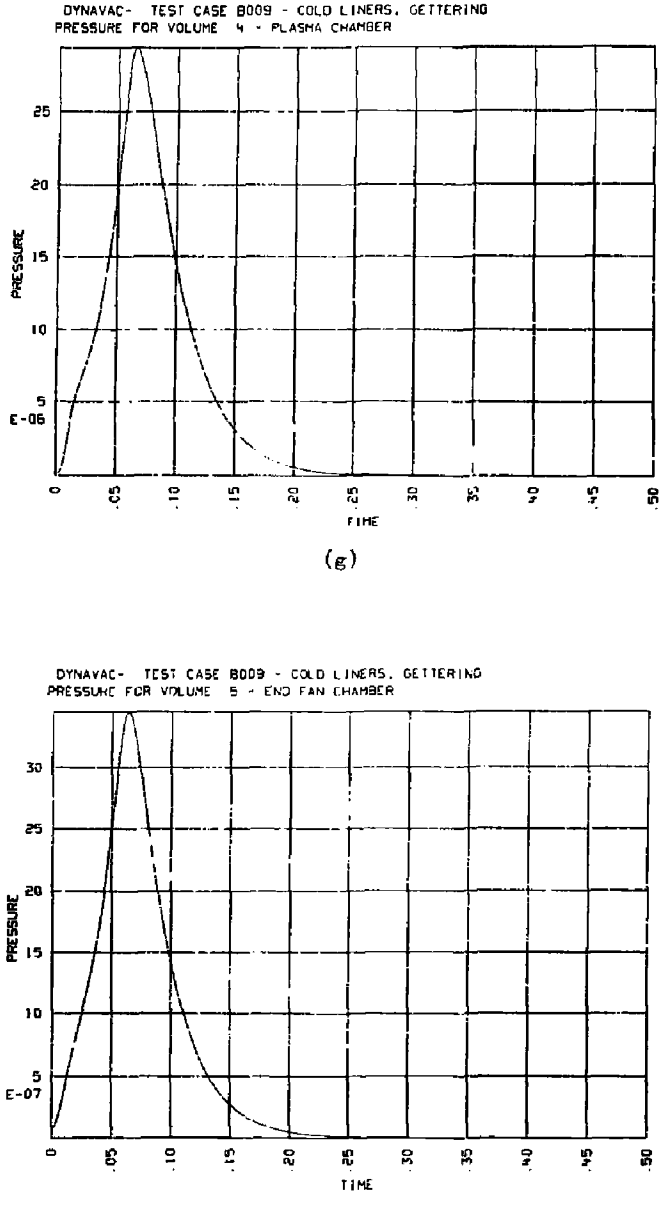
FIG. 6. (Continued)



(e)

(f)

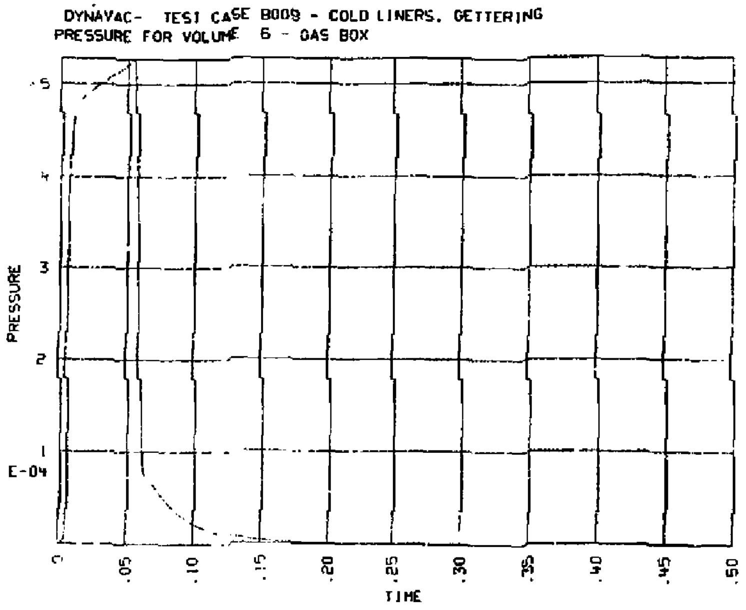
FIG. 6. (Continued)



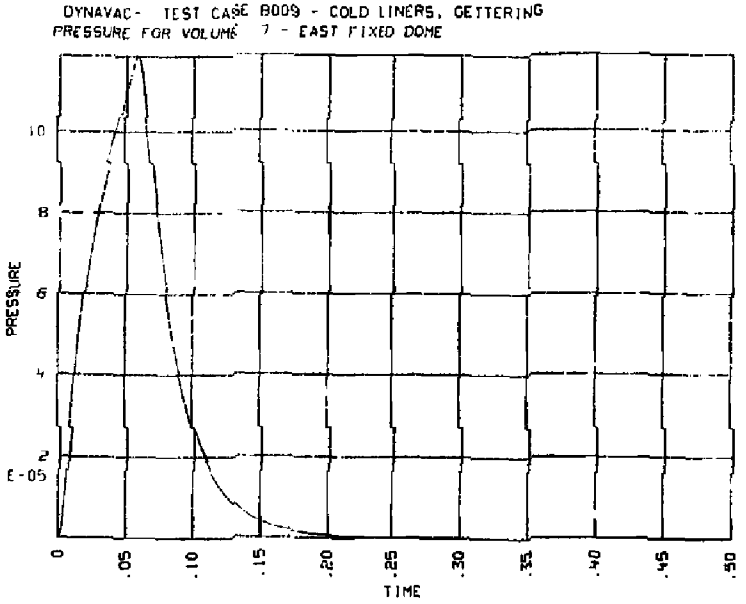
(g)

(h)

FIG. 6. (Continued)

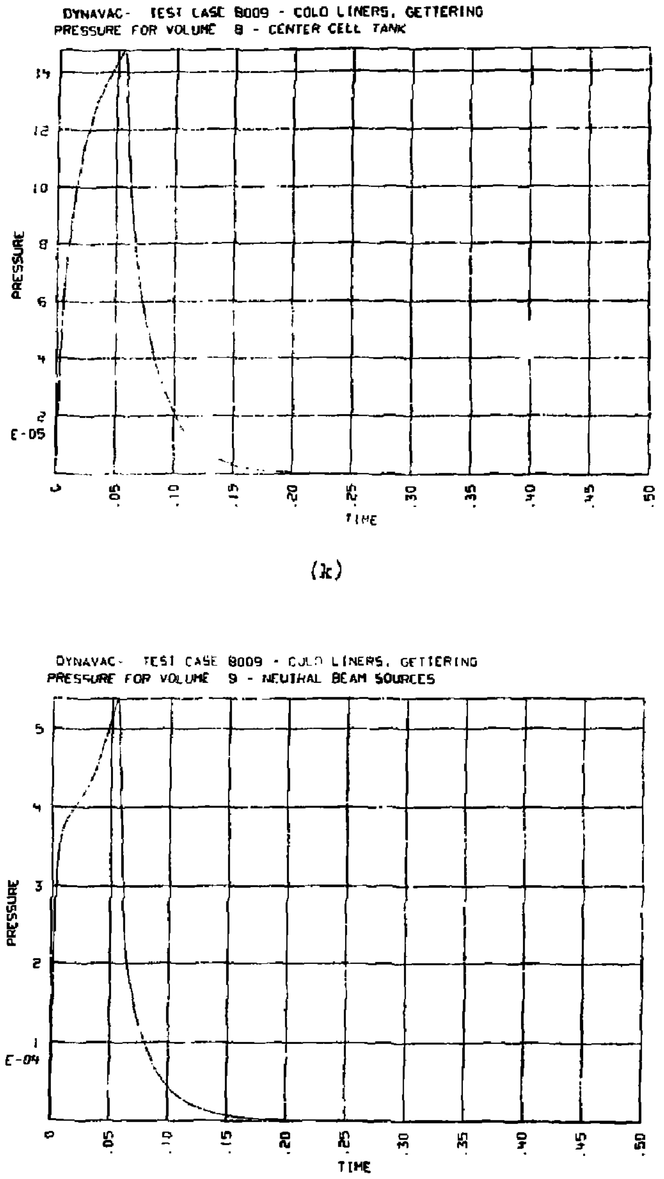


(i)



(j)

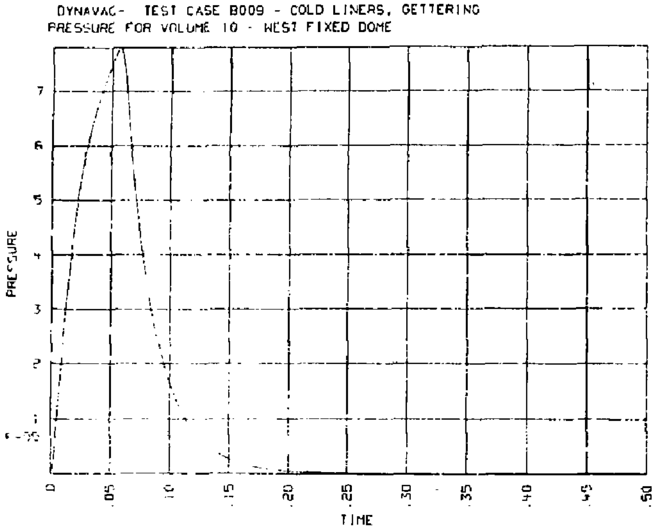
FIG. 6. (Continued)



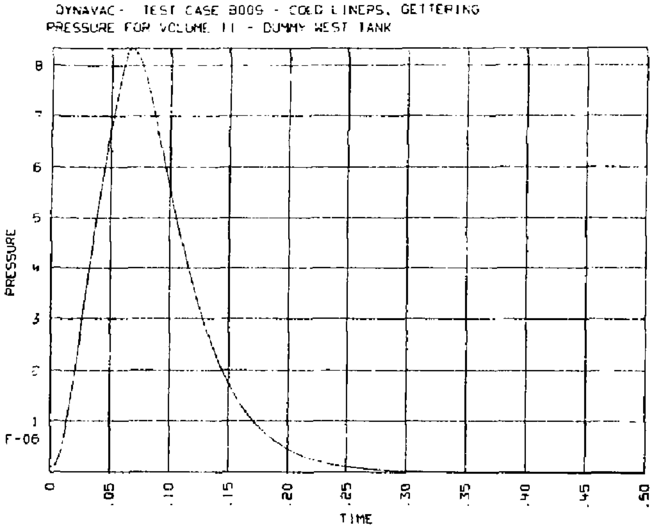
(k)

(l)

FIG. 6. (Continued)



(m)



(n)

FIG. 6. (Continued)

```
                 THIS IS FILE VACERR

  TSOLVER         0 STEPS      0FUNCTION      0 JACOBIAN       LAST TIME
                               EVALUATIONS     EVALUATIONS     STEP SIZE
----------      ----------    ----------     ----------      ----------
 1.000E-03          16            20              6          3.997E-05
 2.000E-03          26            31              9          4.576E-05
 3.000E-03          35            40             11          2.511E-04
 4.000E-03          39            45             11          2.468E-04
 5.000E-03          55            70             17          1.650E-06
 6.000E-03          79           102             25          9.411E-05
 7.000E-03          88           111             25          8.801E-05
 8.000E-03          97           120             27          5.650E-05
 9.000E-03         105           131             29          1.830E-04
 1.000E-02         111           138             30          6.496E-05
 1.100E-02         117           146             33          6.827E-05
 1.200E-02         123           154             36          4.026E-05
 1.300E-02         129           162             39          2.444E-04
 1.400E-02         134           170             39          1.749E-04
 1.500E-02         139           175             39          2.283E-04
 1.600E-02         145           184             42          3.209E-04
 1.700E-02         151           195             47          2.709E-04
 1.800E-02         157           205             51          2.901E-04
 1.900E-02         163           217             55          2.764E-04
 2.000E-02         169           228             59          2.038E-04
 2.100E-02         175           239             64          2.525E-04
 2.200E-02         181           25[illegible]   69          2.423E-04
 2.300E-02         187           26[illegible]   74          2.379E-04
 2.400E-02         193           272             79          2.414E-04
 2.500E-02         199           283             84          2.344E-04
 2.600E-02         205           294             89          2.408E-04
 2.700E-02         211           305             94          2.296E-04
 2.800E-02         217           316             99          2.411E-04
 2.900E-02         223           327            104          2.229E-04
 3.000E-02         229           338            109          2.432E-04
 3.100E-02         235           349            114          2.131E-04
 3.200E-02         239           357            116          2.274E-04
 3.300E-02         245           368            120          3.448E-04
 3.400E-02         251           379            124          1.490E-04
 3.500E-02         256           386            126          1.754E-04
 3.600E-02         261           393            128          5.930E-05
 3.700E-02         267           400            131          6.375E-05
 3.800E-02         273           408            134          2.964E-05
 3.900E-02         279           414            137          2.121E-04
 4.000E-02         283           420            138          2.453E-04
 4.100E-02         289           431            143          2.796E-04
 4.200E-02         293           438            145          6.934E-05
 4.300E-02         298           444            147          2.514E-04
 4.400E-02         302           450            149          1.387E-04
 4.500E-02         307           456            150          4.451E-04
 4.600E-02         313           467            154          1.265E-05
 4.700E-02         319           473            157          2.556E-04
 4.800E-02         323           479            159          1.207E-04
 4.900E-02         328           485            161          3.397E-05
 5.000E-02         334           491            164          4.444E-05
 5.100E-02         339           496            166          5.595E-04
 5.200E-02         342           502            167          1.063E-04
 5.300E-02         347           509            168          3.620E-04
 5.400E-02         350           513            168          2.760E-04
```

FIG. 7 Printed output (VACERR) for example problem.

APPENDIX A--INTERNAL DYNAVAC STRUCTURES

DYNAVAC has been written in a modular fashion to facilitate modifications and additional capabilities, and the following discussion of its structure will be of interest to users who intend to make such changes.

DYNAVAC consists of a short main program, 7 other DYNAVAC subroutines, and the LSODE package which consists of 21 subroutines in all. The main program, DYNAVAC, serves mainly as a calling routine, with some minor problem control responsibilities. It first calls subroutine HOSE, which handles all input, error checking of input, and the echoing of the input data. All data is passed to other subroutines via COMMON's. When HOSE returns control to DYNAVAC, subroutine BAG is called. This subroutine actually obtains the solution at the desired output times by repeated calls to LSODE. In addition, BAG checks for indications of trouble from LSODE, and performs a stepwise integration to calculate QINT for each volume. The pressure vs time output data is written into file SCRATCH in a form suitable for later use. When the solution is complete, control is returned to DYNAVAC. The next subroutine called is EXHAUST, which handles all of the output functions. It reads all of the output data from file SCRATCH, and then produces all of the printed and plotted output. Control is then returned to DYNAVAC, which terminates the run. If at any point an error occurs which has been anticipated in the coding (such as a negative volume input, or problems with LSODE), control is transferred at an appropriate time from whatever subroutine is executing, through DYNAVAC to subroutine PLUG. Currently, this subroutine just prints a brief message with suggestions to help in fixing the problem, and then terminates the run gracefully.

These five routines determine the main flow of operation of DYNAVAC. There are, however, three routines supplied besides LSODE and its subroutines. These three routines are called by LSODE to provide information needed for the solution. The first of these, PDERIV, is basically an explicit coding of the system equations. It merely loads array PDOT by using array P, the time T, and the system properties. This subroutine also performs the determination of the gas-source rates, by interpolation or extrapolation of the appropriate Q function. Subroutine PDERIV also calls subroutine CONDUCTANCE, which loads all of the system properties into the array C. This simplifies the coding in

PDERIV. CONDUCTANCE actually determines the pumping speeds (where they are variable) by using QINT, as calculated in BAG. The final DYNAVAC subroutine which may be of interest is subroutine PJAC, which supplies the system Jacobian matrix, PD, defined as

$$\mathrm{PD}\ (\mathrm{I},\mathrm{J}) = \frac{\partial}{\partial \mathrm{P}(\mathrm{J})}\left(\frac{\partial \mathrm{P}(\mathrm{J})}{\partial t}\right) .$$

This matrix is equal to the C matrix (from subroutine CONDUCTANCE) divided by the volumes. It is not necessary for PJAC to call CONDUCTANCE, since PJAC is never called unless PDERIV is called first.

The only other subroutines used are those in the LSODE package, and a detailed discussion of these routines will not be attempted here. Reference 3 should be consulted for such a discussion, if LSODE modifications are contemplated. For a discussion of the general user interface requirements for LSODE, see the comment lines below line 651 in the source listing, DYNAVACS. In the following pages, the source listing for LSODE and its subroutines has been omitted, leaving only the eight DYNAVAC subroutines. In DYNAVACS, the LSODE package begins below line 651.

```
      PROGRAM DYNAVAC (VACIN,TAPE7=VACIN,VACPRT,TAPE5=VACPRT,
     1SCRATCH,TAPE9=SCRATCH,VACERR,TAPE3=VACERR,TAPE59)
C---/--------------------------------------------------------------------
C THIS IS THE 8 MAY 1980 VERSION OF
C
C       DYNAVAC -- A TRANSIENT VACUUM NETWORK ANALYSIS CODE
C
C                    AUTHOR AND CONTACT:
C                               GARY A. DEIS, L-540
C                                TMX/BETA-II GROUP,
C                                MAGNETIC FUSION ENGINEERING DIVISION
C                                LAWRENCE LIVERMORE LABORATORY
C                                PHONE: 422-1479
C
C DYNAVAC SOLVES THE INITIAL VALUE PROBLEM OF A SYSTEM OF
C ODE'S OF THE FORM:
C     DP(I)/DT=(Q(I)-SPD(I)*P(I)-SUM(C(I,J)*P(I)-C(J,I)*P(J)))/VOL(I)
C WHERE:
C       P = PRESSURE (TORR)
C       Q = GAS SOURCE RATE (TORR-L/SEC)
C     SPD = PUMPING SPEED (L/SEC)
C C(I,J) = CONDUCTANCE FROM I TO J (L/SEC), ASSUMED EQUAL TO
C              C(J,I) UNLESS BOTH ARE INPUT
C     VOL = VOLUME (L)
C SHOWN IN PARENTHESES ARE REPRESENTATIVE UNITS.  ANY UNITS
C CAN BE USED, BUT THEY MUST BE INTERNALLY CONSISTANT.
C THAT IS, (P*SPD/Q), (P*C/Q), AND (P*V/Q*TIME) MUST
C BE UNITLESS.
C-----------------------------------------------------------------------
      COMMON /BLOCK1/ VOL(20),SPD(20),Q(20),CI(20,20)
      COMMON /BLOCK2/ IEX,IPLUG
      COMMON /BLOCK3/ P(20),NSTEP(10),TSTEP(10),T0,NBLOCK,IBAG,IHOSE
      COMMON /BLOCK4/ C(20,20),NTANK,IPD,IPJAC,ICONDU,ISTAT
      COMMON /BLOCK5/ RWORK(620),IWORK(50)
      COMMON /BLOCK7/ NFUNQ,IFUNQ(20),NPTQ(20),XQ(20,20),YQ(20,20),
     1FQ(20)
      COMMON /BLOCK8/ NPTS(20),QINT(20),XS(20,20),YS(20,20)
      DATA IDYN /2/
      CALL CHANGE (8H+DYNAVAC)
      WRITE (59,100) IDYN,IHOSE,IBAG,IEX,IPLUG,ICONDU,IPD,IPJAC
  100 FORMAT(" DYNAVAC VERSION ",8(I2,"."))
      WRITE (59,200)
  200 FORMAT("COMPILED 5/8/80")
      KOUNT=0
      ISTAT=0
      MODE=0
      CALL HOSE(MODE)
C---/-------------------------------------------------------------------
C SUBROUTINE HOSE HANDLES ALL INPUT.
C MODE=1 SIGNALS DATA CHECK MODE.
C ISTAT .LT. 0 AT ANY POINT SIGNALS AN ERROR
C---/-------------------------------------------------------------------
      CALL EMPTY(5)
      IF(MODE.EQ.1)CALL EXIT
      IF(ISTAT.LT.0) GO TO 10
      CALL BAG(KOUNT)
C---/--------------------------------------------------------------------
C SUBROUTINE BAG HANDLES THE ACTUAL SOLUTION, INCLUDING
C TIME STEPPING, ETC.  KOUNT IS THE TOTAL NUMBER OF DATA
C POINTS OUTPUT BY BAG.
```

```
C-----------------------------------------------------------------------
      IF(ISTAT.LT.0) GO TO 20
      CALL EXHAUST(KOUNT)
C-----------------------------------------------------------------------
C SUBROUTINE EXHAUST HANDLES ALL OUTPUT, INCLUDING PRINTOUTS, PLOTS, ETC.
C-----------------------------------------------------------------------
      IF(ISTAT.LT.0) GO TO 30
      CALL EXIT
   10 CALL PLUG(1)
C-----------------------------------------------------------------------
C SUBROUTINE PLUG HANDLES ALL ERROR PROBLEMS WHICH HAVE BEEN ANTICIPATED
C-----------------------------------------------------------------------
      CALL EXIT
   20 CALL PLUG(2)
      CALL EXIT
   30 CALL PLUG(3)
      CALL EXIT
      END
C
C
C
      SUBROUTINE HOSE(MODE)
C-----------------------------------------------------------------------
C SUBROUTINE HOSE PERFORMS INPUT OF ALL DATA FROM FILE "VACIN".
C ALL PARAMETERS ARE READ IN, ECHOED, AND CHECKED FOR ERRORS.  THE
C ENTIRE INPUT FILE IS CHECKED, REGARDLESS OF HOW MANY ERRORS ARE
C ENCOUNTERED.
C-----------------------------------------------------------------------
      COMMON /BLOCK1/ VOL(20),SPD(20),Q(20),CI(20,20)
      COMMON /BLOCK3/ P(20),NSTEP(10),TSTEP(10),T0,NBLOCK,IDUM,IHOSE
      COMMON /BLOCK4/ DUM(20,20),NTANK,IDUM2(3),ISTAT
      COMMON /BLOCK6/ TITLE(10),VTITLE(20,10)
      COMMON /BLOCK7/ NFUNQ,IFUNQ(20),NPTQ(20),XQ(20,20),YQ(20,20),
     1FQ(20)
      COMMON /BLOCK8/ NPTS(20),QINT(20),XS(20,20),YS(20,20)
      DATA IHOSE /2/
      READ (7,101) (TITLE(I),I=1,10)
  101 FORMAT(10A4)
      WRITE (5,600) (TITLE(I),I=1,10)
  600 FORMAT("1",20X,"DYNAVAC OUTPUT"/15X,24(1H*)//10X,10A4/)
  102 FORMAT("1","DYNAVAC-   ",10A4)
      WRITE(59,102) (TITLE(I),I=1,10)
      READ (7,100) NTANK,NCOND,NFUNQ,MODE
      WRITE (5,605) NTANK,NCOND,NFUNQ
  605 FORMAT(/10X,"NUMBER OF VOLUMES",6X,"=",I3/
     110X,"NUMBER OF CONDUCTANCES =",I3/
     210X,"NUMBER OF Q FUNCTIONS  =",I3/)
  100 FORMAT(4I5)
C-----------------------------------------------------------------------
C THE FOLLOWING LOOP INPUTS ALL DATA RELATING TO THE VOLUMES
C-----------------------------------------------------------------------
      DO 10 I=1,NTANK
      P(I)=0.0
      SPD(I)=0.0
      Q(I)=0.0
      FQ(I)=1.0
      VOL(I)=0.0
      DO 11 J=1,NTANK
      CI(I,J)=0.0
   11 CONTINUE
```

```
      READ (7,103) (VTITLE(I,J),J=1,10)
      WRITE (5,610) I,(VTITLE(I,J),J=1,10)
  610 FORMAT(/"VOLUME",I3," - ",10A4)
  103 FORMAT(10A4)
      READ (7,200) VOL(I),SPD(I),Q(I),P(I),NPTS(I),IFUNQ(I)
  200 FORMAT(4E10.3,2I5)
      WRITE (5,615) VOL(I),P(I),Q(I)
  615 FORMAT(5X,"VOLUME",11X,"= ",E10.3/5X,"INITIAL PRESSURE = ",
     1E10.3/5X,"GAS SOURCE",7X,"= ",E10.3)
      IF(IFUNQ(I).GE.1) WRITE (5,620) IFUNQ(I)
  620 FORMAT(5X,"Q FUNCTION",7X,"= ",8X,I2)
      IF(NPTS(I).LT.1) GO TO 625
      READ (7,205) (XS(I,J),YS(I,J),J=1,NPTS(I))
  205 FORMAT(2E10.3)
      WRITE (5,630) NPTS(I),(XS(I,J),YS(I,J),J=1,NPTS(I))
  630 FORMAT(10X,"PUMPING SPEED TABLE (",I2," POINTS)"/13X,
     1"INT(S*P*DT)",5X,"SPEED"/13X,11(1H-),2X,11(1H-)/
     220(14X,E10.3,2X,E10.3/))
      GO TO 10
  625 WRITE (5,635) SPD(I)
  635 FORMAT(5X,"PUMPING SPEED",4X,"= ",E10.3)
   10 CONTINUE
C-----------------------------------------------------------------------
C THE FOLLOWING LOOP CHECKS ALL THE VOLUME DATA FOR ERRORS
C-----------------------------------------------------------------------
      DO 15 I=1,NTANK
      IF(VOL(I).GT.0.0) GO TO 16
      ISTAT=-1
      WRITE (5,210) I,VOL(I)
      WRITE (59,210) I,VOL(I)
  210 FORMAT("***ERROR***  VOLUME FOR TANK ",I2," IS ",E10.3)
   16 IF(P(I).GE.0.0) GO TO 17
      ISTAT=-1
      WRITE (5,211) I,P(I)
      WRITE (59,211) I,P(I)
  211 FORMAT("***ERROR*** PRESSURE FOR TANK ",I2," IS ",E10.3)
   17 IF(SPD(I).GE.0.0) GO TO 18
      ISTAT=-1
      WRITE (5,212) I,SPD(I)
      WRITE (59,212) I,SPD(I)
  212 FORMAT("***ERROR*** SPEED FOR TANK ",I2," IS ",E10.3)
   18 IF(Q(I).GE.0.0) GO TO 770
      ISTAT=-1
      WRITE (5,213) I,Q(I)
      WRITE (59,213) I,Q(I)
  213 FORMAT("***ERROR*** GAS SOURCE FOR TANK ",I2," IS ",E10.3)
  770 IF(NPTS(I).LT.1) GO TO 15
      IF(NPTS(I).EQ.1) GO TO 705
      DO 710 J=1,NPTS(I)-1
      IF(YS(I,J).GE.0.0) GO TO 715
      WRITE (5,670) J,I,YS(I,J)
      WRITE (59,670) J,I,YS(I,J)
  670 FORMAT("***ERROR*** SPEED",I3," IN SPEED TABLE",I3," IS ",E10.3)
      ISTAT=-1
  715 IF(XS(I,J).GE.0.0) GO TO 720
      WRITE (5,685) J,I,XS(I,J)
      WRITE (59,685) J,I,XS(I,J)
  685 FORMAT("***ERROR*** QINT",I3," IN SPEED TABLE",I3," IS ",E10.3)
      ISTAT=-1
  720 IF(XS(I,J).LT.XS(I,J+1)) GO TO 710
```

```
      WRITE (5,675) I,J,XS(I,J),J+1,XS(I,J+1)
      WRITE (59,675) I,J,XS(I,J),J+1,XS(I,J+1)
  675 FORMAT("***ERROR*** IN SPEED TABLE",I3,
     1". POINTS ARE OUT OF SEQUENCE"/10X,"POINT",I3," = ",E10.3,
     2", POINT",I3," = ",E10.3)
      ISTAT=-1
  710 CONTINUE
      IF(YS(I,NPTS(I)).GE.0.0) GO TO 15
      WRITE (5,670) NPTS(I),I,YS(I,NPTS(I))
      WRITE (59,670) NPTS(I),I,YS(I,NPTS(I))
      ISTAT=-1
      GO TO 15
  705 WRITE (5,680) I
      WRITE (59,680) I
  680 FORMAT("***ERROR*** ONLY ONE POINT IN SPEED TABLE ",I2)
      ISTAT=-1
   15 CONTINUE
C-----------------------------------------------------------------------
C CONDUCTANCES ARE INPUT HERE. CI IS THE INPUT CONDUCTANCE
C ARRAY.  CI(J,K) IS SET EQUAL TO CI(K,J) UNLESS BOTH ARE
C INPUT. IN WHICH CASE THEY CAN HAVE DIFFERENT VALUES.
C-----------------------------------------------------------------------
      DO 20 L=1,NCOND
      READ (7,300) J,K,(CI(J,K))
  300 FORMAT(2I5,E10.3)
      IF(J.NE.K) GO TO 22
      ISTAT=-1
      WRITE(3,310) J,K,CI(J,K)
   22 IF(CI(J,K).GE.0.0) GO TO 21
      ISTAT=-1
      WRITE (5,310) J,K,CI(J,K)
      WRITE (59,310) J,K,CI(J,K)
  310 FORMAT("***ERROR*** CONDUCTANCE ",2I3," IS ",E10.3)
   21 IF(CI(K,J).EQ.0.0) CI(K,J)=CI(J,K)
   20 CONTINUE
      WRITE (5,640)
  640 FORMAT(/5X,"CONDUCTANCE MATRIX"/5X,18(1H-))
      DO 30 I=1,NTANK
      WRITE (5,400) (CI(I,J),J=1,NTANK)
  400 FORMAT(20E10.3)
   30 CONTINUE
      IF(NFUNQ.LT.1) GO TO 51
C------------------------------------------------------------------
C THE FOLLOWING LOOP INPUTS Q FUNCTIONS.  THE GAS SOURCE
C RATE AT ANY TIME IS COMPUTED AS Q=QL*FQ WHERE QL IS THE
C INITIAL RATE (INPUT WITH EACH VOLUME), AND FQ IS THE
C Q FUNCTION VALUE FOR THAT TIME.
C------------------------------------------------------------------
      DO 50 I=1,NFUNQ
      READ (7,401) NPTQ(I)
      READ (7,402) (XQ(I,J),YQ(I,J),J=1,NPTQ(I))
      WRITE (5,645) I,NPTQ(I)
      WRITE (5,650) (XQ(I,J),YQ(I,J),J=1,NPTQ(I))
  645 FORMAT(//5X,"Q FUNCTION NUMBER ",I2,", (",I2," POINTS)"/
     113X,"TIME",9X,"FQ"/10X,10(1H-),2X,10(1H-))
  650 FORMAT(10X,E10.3,2X,E10.3)
  401 FORMAT(I5)
  402 FORMAT(2E10.3)
      IF(NPTQ(I).LT.2) GO TO 815
      DO 810 J=1,NPTQ(I)-1
```

```
      IF(YQ(I,J).GE.0.0) GO TO 815
      WRITE (5,690) J,I,YQ(I,J)
      WRITE (59,690) J,I,YQ(I,J)
  690 FORMAT("***ERROR*** FQ POINT",I3," IN Q FUNCTION",I3," IS ",E10.3)
      ISTAT=-1
  815 IF(XQ(I,J).LT.XQ(I,J+1)) GO TO 810
      WRITE (5,695) I,J,XQ(I,J),J+1,XQ(I,J+1)
      WRITE (59,695) I,J,XQ(I,J),J+1,XQ(I,J+1)
  695 FORMAT("***ERROR*** IN Q FUNCTION",I3,
     1", POINTS ARE OUT OF SEQUENCE"/
     210X,"POINT",I3," = ",E10.3,", POINT",I3," = ",E10.3)
      ISTAT=-1
  810 CONTINUE
      IF(YQ(I,NPTQ(I)).GE.0.0) GO TO 50
      WRITE (5,690) NPTQ(I),I,YQ(I,NPTQ(I))
      WRITE (59,690) NPTQ(I),I,YQ(I,NPTQ(I))
      ISTAT=-1
      GO TO 50
  805 WRITE (5,820) I
      WRITE (59,820) I
  820 FORMAT("***ERROR*** ONLY ONE POINT IN Q FUNCTION ",I2)
      ISTAT=-1
   50 CONTINUE
   51 READ (7,500) NBLOCK,TO
  500 FORMAT(I5,E10.3)
      WRITE (5,655) NBLOCK,TO
  655 FORMAT(/5X,"PROBLEM CONTROL"/5X,15(1H-)/10X,
     1"NUMBER OF TIME BLOCKS = ",I2/10X,"INITIAL TIME",
     210X,"= ",E10.3/25X,"NSTEP",4X,"TSTEP"/25X,5(1H-),
     32X,10(1H-))
      READ (7,500) (NSTEP(I),TSTEP(I),I=1,NBLOCK)
      DO 665 I=1,NBLOCK
      WRITE (5,660) I,NSTEP(I),TSTEP(I)
  660 FORMAT(15X,"BLOCK",I3,1X,I5,2X,E10.3)
      IF(NSTEP(I).GT.0) GO TO 755
      WRITE (5,750) I,NSTEP(I)
      WRITE (59,750) I,NSTEP(I)
  750 FORMAT("***ERROR*** IN TIME BLOCK",I3,", NSTEP IS ",I5)
      ISTAT=-1
  755 IF(TSTEP(I).GT.0.0) GO TO 665
      WRITE (5,760) I,TSTEP(I)
      WRITE (59,760) I,TSTEP(I)
  760 FORMAT("***ERROR*** IN TIME BLOCK",I3,", TSTEP IS ",E10.3)
      ISTAT=-1
  665 CONTINUE
      CALL EMPTY(5)
      RETURN
      END
C
C
C
      SUBROUTINE BAG(KOUNT)
C-----------------------------------------------------------------------
C SUBROUTINE BAG ACTUALLY OBTAINS THE SOLUTION TO THE PROBLEM AT
C THE DESIRED TIME STEPS. THE ACTUAL ODE SOLVER USED IS LSODE
C (21 JAN. 1980 VERSION), AVAILABLE FROM ALAN C. HINDMARSH, L-300, LLL.
C LSODE IS CALLED ONCE FOR EACH OUTPUT POINT.
C-----------------------------------------------------------------------
      EXTERNAL PDERIV,PJAC
      COMMON /BLOCK1/ DUM1(20),SPO(20),DUM2(20),DUM3(20,20)
```

```
      COMMON /BLOCK2/ IDUM3(2)
      COMMON /BLOCK3/ P(20),NSTEP(10),TSTEP(10),T0,NBLOCK,IBAG,KOUNT
      COMMON /BLOCK4/ C(20,20),NTANK,IDUM2(3),ISTAT
      COMMON /BLOCK5/ RWORK(620),IWORK(50)
      COMMON /BLOCK8/ NPTS(20),QINT(20),XS(20,20),YS(20,20)
      DIMENSION PLAST(20)
      DATA IBAG /2/
C------------------------------------------------------------------------
C ITOL, RTOL, AND ATOL ARE ERROR CONTROL PARAMETERS WHICH FORCE THE
C ERROR PER TIME STEP (NOT PER OUTPUT POINT) TO BE LESS THAN
C       ATOL+RTOL*P(I)
C THIS PROVIDES BOTH RELATIVE AND ABSOLUTE ERROR CONTROL.
C------------------------------------------------------------------------
      ITOL=1
      RTOL=1.0E-04
      ATOL=1.0E-10
      ITASK=4
      ISTATE=1
      IOPT=0
      MF=21
      LRW=620
      LIW=50
      TOUT=T0
      KOUNT=1
      WRITE (9,100) T0,(P(K),K=1,NTANK)
      TLAST=T0
      DO 5 I=1,NTANK
      QINT(I)=0.0
      PLAST(I)=P(I)
    5 CONTINUE
      WRITE (3,500)
  500 FORMAT("1",20X,"THIS IS FILE VACERR"///9X,"TSOLVER",
     18X,"0 STEPS",6X,"0FUNCTION",5X,"0 JACOBIAN",7X,"LAST TIME"/
     237X,"EVALUATIONS",4X,"EVALUATIONS",5X,"STEP SIZE"/5X,
     35(2X,10(1H-),3X))
      DO 10 I=1,NBLOCK
      DO 10 J=1,NSTEP(I)
      TOUT=TOUT+TSTEP(I)
      RWORK(1)=TOUT
      CALL LSODE(PDERIV,NTANK,P,T0,TOUT,ITOL,RTOL,ATOL,ITASK,ISTATE,
     !IOPT,RWORK,LRW,IWORK,LIW,PJAC,MF)
      WRITE (3,501) RWORK(13),IWORK(11),IWORK(12),IWORK(13),RWORK(11)
  501 FORMAT(8X,E10.3,7X,I5,2(10X,I5),8X,E10.3)
      IF(ISTATE.LT.0) GO TO 30
C------------------------------------------------------------------------
C DATA IS OUTPUT TO FILE "SCRATCH" IN A FORM SUITABLE
C FOR READING BY EXHAUST.
C------------------------------------------------------------------------
   30 WRITE (9,100) T0,(P(K),K=1,NTANK)
  100 FORMAT(11E10.3/5X,10E10.3)
      KOUNT=KOUNT+1
C-------------------------------------------------------
C THE REQUIRED INTEGRATION FOR THE PUMPING SPEED
C CALCULATION IS PERFORMED HERE. THE INTEGRATION IS
C INTEGRAL(SPD(I)*P(I)*DT). THE RESULT, QINT(I), IS
C USED IN SUBROUTINE CONDUCTANCE.
C-------------------------------------------------------
      DO 40 K=1,NTANK
      QINT(K)=QINT(K)+0.5*(P(K)+PLAST(K))*SPD(K)*(T0-TLAST)
      PLAST(K)=P(K)
```

```
   40 CONTINUE
      TLAST=TO
   10 CONTINUE
      CALL EMPTY(9)
      RETURN
   30 ISTAT=-1
      CALL EMPTY(9)
      WRITE (3,502) ISTATE,TO
  502 FORMAT("ISTATE = ",I2," AT T = ",E10.3)
      IF(ISTATE.LT.-1) RETURN
      ISTATE=2
      GO TO 31
      END
C
C
C
      SUBROUTINE EXHAUST(KOUNT)
C-----------------------------------------------------------------
C SUBROUTINE EXHAUST PERFORMS ALL OUTPUT FUNCTIONS.
C PRESSURE VS. TIME DATA IS READ FROM FILE "SCRATCH".
C FIRST, PLOTS ARE PRODUCED FOR ALL VOLUMES AT THE SAME SCALE.
C FIVE LINES PER PLOT.  NEXT A PLOT IS MADE FOR EACH VOLUME,
C USING THE LARGEST POSSIBLE SCALE. IN ADDITION, PRINTED DATA
C IS OUTPUT TO FILE VACPRT, 50 LINES TO A PAGE, WITH APPROPRIATE
C HEADINGS AT THE TOP OF EACH PAGE.
C-----------------------------------------------------------------
      DIMENSION T(1000),P(20,1000),PPLOT(1000)
      DIMENSION PMIN(20),PMAX(20)
      COMMON /BLOCK2/ IEX,IDUM
      COMMON /BLOCK4/ DUM(20,20),NTANK,IDUM2(3),ISTAT
      COMMON /BLOCK5/ RWORK(620),IWORK(50)
      COMMON /BLOCK6/ TITLE(10),VTITLE(20,10)
      DATA IEX /2/
      CALL REWIND(9)
      DO 90 I=1,NTANK
      PMIN(I)=1.0E+50
      PMAX(I)=0.0
   90 CONTINUE
      WRITE (5,104) (TITLE(I),I=1,10)
  104 FORMAT("1","DYNAVAC-   ",10A4//)
      WRITE (5,105) (I,I=1,NTANK)
  105 FORMAT(3X,"TIME",3X,10(3X,"P",I2,4X)/5X,10(3X,"P",I2,4X))
      IKOUNT=0
C-----------------------------------------------------------
C DATA IS READ IN FROM FILE "SCRATCH" AND LOADED INTO
C ARRAY P.  THE FIRST SUBSCRIPT IS THE VOLUME NUMBER,
C AND THE SECOND IS THE TIME STEP NUMBER (TO=STEP1).
C PRINTED DATA IS OUTPUT TO FILE "VACPRT"
C---------------------------------------------------------
      DO 10 I=1,KOUNT
      READ (9,100) T(I),(P(J,I),J=1,NTANK)
  100 FORMAT(11E10.3/5X,10E10.3)
      WRITE (5,100) T(I),(P(J,I),J=1,NTANK)
      IKOUNT=IKOUNT+1
      IF(IKOUNT.LT.50) GO TO 15
      WRITE (5,104) (TITLE(K),K=1,10)
      WRITE (5,105) (K,K=1,NTANK)
      IKOUNT=0
C-----------------------------------------------------------
C MAXIMUM AND MINIMUM PRESSURE FOR EACH VOLUME IS LOCATED.
```

```
C AS WELL AS THE MAXIMUM AND MINIMUM FOR ALL VOLUMES (PMAXX
C AND PMINX). THESE ARE USED FOR SCALING PURPOSES.
C----------------------------------------------------------------------
   15 DO 20 J=1,NTANK
      IF(P(J,I).LT.PMIN(J)) PMIN(J)=P(J,I)
      IF(P(J,I).GT.PMAX(J)) PMAX(J)=P(J,I)
   20 CONTINUE
   10 CONTINUE
      PMINX=PMIN(1)
      PMAXX=PMAX(1)
      DO 30 I=2,NTANK
      IF(PMIN(I).LT.PMINX) PMINX=PMIN(I)
      IF(PMAX(I).GT.PMAXX) PMAXX=PMAX(I)
   30 CONTINUE
C----------------------------------------------------------------------
C BEGIN PLOTTING
C----------------------------------------------------------------------
      CALL KEEP80(1,3)
      CALL FR80ID
      CALL FRAME
      CALL SETCHM(1)
C----------------------------------------------------------------------
C THE FOLLOWING LOOP PLOTS DATA FOR ALL VOLUMES AT THE SAME
C SCALE, FIVE PLOTS PER GRAPH. AS MANY GRAPHS ARE GENERATED
C AS REQUIRED.
C----------------------------------------------------------------------
      DO 40 I=1,NTANK,5
      CALL MAPG(T(1),T(KOUNT),PMINX,PMAXX,0.125,0.925,0.300,0.878)
      CALL SETCH(1.,40.,1,0,1,0)
      WRITE (100,200) (TITLE(J),J=1,10)
  200 FORMAT(" DYNAVAC-   ",10A4)
      MMAX=NTANK
      IF(NTANK.GT.(I+4)) MMAX=I+4
      WRITE (100,300) I,MMAX
  300 FORMAT(" PRESSURE FOR VOLUMES ",I2," TO ",I2)
      CALL SETCH(44.,10.6,1,0,1,0)
      WRITE (100,400)
  400 FORMAT(" TIME")
      CALL SETCH(3.5,46.,1,0,1,1)
      WRITE (100,500)
  500 FORMAT(" PRESSURE")
      DO 50 J=I,MMAX
      DO 60 K=1,KOUNT
      PPLOT(K)=P(J,K)
   60 CONTINUE
      GO TO (1,2,3,4,5),(J-I+1)
    1 CALL TRACEC(1H1,T,PPLOT,KOUNT)
      GO TO 50
    2 CALL TRACEC(1H2,T,PPLOT,KOUNT)
      GO TO 50
    3 CALL TRACEC(1H3,T,PPLOT,KOUNT)
      GO TO 50
    4 CALL TRACEC(1H4,T,PPLOT,KOUNT)
      GO TO 50
    5 CALL TRACEC(1H5,T,PPLOT,KOUNT)
   50 CONTINUE
      CALL FRAME
   40 CONTINUE
C----------------------------------------------------------------------
C THE FOLLOWING LOOPS PLOT THE DATA FOR EACH VOLUME, ONE PLOT PER
```

```
C PAGE, TO THE CORRECT SCALE FOR THAT VOLUME.  NTANK PLOTS ARE
C PRODUCED.
C-----------------------------------------------------------------------
      DO 70 I=1,NTANK
      DO 80 J=1,KOUNT
      PPLOT(J)=P(I,J)
   80 CONTINUE
      CALL MAPG(T(1),T(KOUNT),PMIN(I),PMAX(I),0.125,0.925,0.300,0.878)
      CALL SETCH(11.,40.,1,0,1,0)
      WRITE (100,200) (TITLE(J),J=1,10)
      WRITE (100,600) I,(VTITLE(I,J),J=1,10)
  600 FORMAT("PRESSURE FOR VOLUME ",I2," - ",10A4)
      CALL SETCH(44.,10.6,1,0,1,0)
      WRITE (100,400)
      CALL SETCH(3.5,46.,1,0,1,1)
      WRITE (100,500)
      CALL TRACE(T,PPLOT,KOUNT)
      CALL FRAME
   70 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE PLUG(N)
C-----------------------------------------------------------------------
C SUBROUTINE PLUG IS CALLED IN THE EVENT OF AN ANTICIPATED ERROR.
C IT CURRENTLY DOES NOTHING.
C-----------------------------------------------------------------------
      COMMON /BLOCK2/ IDUM,IPLUG
      DATA IPLUG /2/
      IF(N.NE.1) GO TO 10
      WRITE (5,100)
      WRITE (59,100)
  100 FORMAT("ERROR DETECTED BY INPUT SUBROUTINE"/5X,
     1"FURTHER EXECUTION TERMINATED"/5X,
     2"CHECK FILE VACPRT AND CORRECT FILE VACIN")
      GO TO 40
   10 IF(N.NE.2) GO TO 20
      WRITE (5,200)
      WRITE (59,200)
  200 FORMAT("ERROR DETECTED BY SOLVER SUBROUTINE"/5X,
     1"FURTHER EXECUTION TERMINATED"/5X,
     2"CHECK FILE VACERR AND VACPRT FOR FURTHER INFORMATION")
      GO TO 40
   20 IF(N.NE.3) GO TO 30
      WRITE (5,300)
      WRITE (59,300)
  300 FORMAT("ERROR DETECTED BY OUTPUT ROUTINE")
      GO TO 40
   30 WRITE (5,400)
      WRITE (59,400)
  400 FORMAT("ERROR ENCOUNTERED - LOCATION UNKNOWN")
   40 CALL EMPTY(3)
      CALL EMPTY(5)
      CALL EMPTY(9)
      RETURN
      END
C
C
```

```
C
      SUBROUTINE CONDUCTANCE
C-----------------------------------------------------------------------
C SUBROUTINE CONDUCTANCE LOADS THE ARRAY C, SUCH THAT EACH OFF-
C DIAGONAL TERM IS THE CORRESPONDING CONDUCTANCE, WHILE THE
C DIAGONAL TERMS ARE
C                -1.*(SPD(I)+CI(I,1)+CI(I,2)+CI(I,3).....)
C-----------------------------------------------------------------------
      COMMON /BLOCK1/ VOL(20),SPD(20),DUM1(20),CI(20,20)
      COMMON /BLOCK4/ C(20,20),NTANK,IDUM1(2),ICONDU,IDUM2
      COMMON /BLOCK8/ NPTS(20),QINT(20),XS(20,20),YS(20,20)
      DATA ICONDU /2/
      DO 10 I=1,NTANK
      SUM=0.0
      DO 20 J=1,NTANK
      IF(I.EQ.J) GO TO 20
      C(I,J)=CI(I,J)
      SUM=SUM+C(I,J)
   20 CONTINUE
C-------------------------------------------------------------
C THE FOLLOWING STATEMENTS CALCULATE THE PUMPING SPEED FOR
C EACH VOLUME AS A FUNCTION OF INTEGRAL(SPD*P*DT). THE
C ACTUAL INTEGRATION IS PERFORMED IN SUBR. BAG, AND THE
C RESULT IS PASSED AS QINT(I).
C-------------------------------------------------------------
      IF(NPTS(I).LT.1) GO TO 50
      IF(QINT(I).LT.XS(I,1)) GO TO 35
      DO 30 K=1,NPTS(I)-1
      IF(QINT(I).GE.XS(I,K).AND.QINT(I).LT.XS(I,K+1)) GO TO 40
   30 CONTINUE
      SPD(I)=YS(I,NPTS(I))
      GO TO 50
   35 SPD(I)=YS(I,1)
      GO TO 50
   40 DELYS=YS(I,K+1)-YS(I,K)
      DELXS=XS(I,K+1)-XS(I,K)
      SPD(I)=YS(I,K)+(QINT(I)-XS(I,K))*DELYS/DELXS
   50 C(I,I)=-1.0*(SPD(I)+SUM)
   10 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE PDERIV(NTANK,T,P,PDOT)
C-----------------------------------------------------------------------
C SUBROUTINE PDERIV IS CALLED BY THE LSODE PACKAGE TO ACTUALLY
C LOAD THE FIRST DERIVATIVES INTO THE AARRAY PDOT.  ESSENTIALLY,
C THIS IS JUST AN EXPLICIT CODING OF THE SYSTEM EQUATIONS.
C ALSO PDERIV HANDLES TIME-DEPENDANT GAS SOURCES, IF THEY
C ARE INCLUDED.
C-----------------------------------------------------------------------
      DIMENSION P(20),PDOT(20)
      COMMON /BLOCK1/ VOL(20),DUM2(20),Q(20),DUM3(20,20)
      COMMON /BLOCK4/ C(20,20),IDUM(2),IPD,IDUM2(2)
      COMMON /BLOCK7/ NFUNQ,IFUNQ(20),NPTQ(20),XQ(20,20),YQ(20,20),
     1FQ(20)
      DATA IPD /2/
      CALL CONDUCTANCE
      IF(NFUNQ.LT.1) GO TO 60
```

```
C------------------------------------------------------------
C THE FOLLOWING STATEMENTS DETERMINE THE GAS-SOURCE TIME
C FUNCTION, WHICH IS MULTIPLIED BY THE INITIAL GAS SOURCE
C RATE TO DETERMINE THE GAS SOURCE RATE AS A FUNCTION OF
C TIME.
C------------------------------------------------------------
      DO 30 I=1,NTANK
      IF(IFUNQ(I).LT.1) GO TO 30
      IQ=IFUNQ(I)
      IF(T.LE.XQ(IQ,1)) GO TO 45
      DO 40 J=1,(NPTQ(IQ)-1)
      IF(T.GE.XQ(IQ,J).AND.T.LT.XQ(IQ,J+1)) GO TO 50
   40 CONTINUE
      FQ(I)=YQ(IQ,NPTQ(IQ))
      GO TO 30
   45 FQ(I)=YQ(IQ,1)
      GO TO 30
   50 DELYQ=YQ(IQ,J+1)-YQ(IQ,J)
      DELXQ=XQ(IQ,J+1)-XQ(IQ,J)
      FQ(I)=YQ(IQ,J)+(T-XQ(IQ,J))*DELYQ/DELXQ
   30 CONTINUE
   60 DO 10 I=1,NTANK
      PDOT(I)=Q(I)*FQ(I)
      DO 20 J=1,NTANK
      PDOT(I)=PDOT(I)+C(J,I)*P(J)
   20 CONTINUE
      PDOT(I)=PDOT(I)/VOL(I)
   10 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE PJAC(NTANK,T,P,ML,MU,PD,NROWPD)
C------------------------------------------------------------------------
C SUBROUTINE PJAC IS CALLED BY THE LSODE PACKAGE TO SUPPLY
C THE JACOBIAN MATRIX
C               D(PDOT(I))/DP(J)=PD(I,J)
C THIS IS VERY NEARLY EQUAL TO THE C ARRAY.
C------------------------------------------------------------------------
      DIMENSION P(20),PD(NROWPD,20)
      COMMON /BLOCK1/ VOL(20),DUM(40),DUM2(20,20)
      COMMON /BLOCK4/ C(20,20),IDUM1,IPJAC,IDUM2(3)
      DATA IPJAC /2/
      DO 10 I=1,NROWPD
      DO 20 J=1,NROWPD
      PD(I,J)=C(J,I)/VOL(I)
   20 CONTINUE
   10 CONTINUE
      RETURN
      END
```

ACKNOWLEDGMENTS

The author gratefully acknowledges the assistance and suggestions of Robert Wong, R. Paul Drake, and especially Mike Richardson, all of whom supplied direction and useful suggestions during the conception and writing of DYNAVAC.

BO/jmp