1976-06

Computer simulation of a propellant feed system for a liquid propellant gun

Dampier, Craig Richard

Monterey, California. Naval Postgraduate School

COMPUTER SIMULATION OF A PROPELLANT FEED SYSTEM FOR A LIQUID PROPELLANT GUN

Craig Richard Dampier

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

COMPUTER SIMULATION OF A PROPELLANT
FEED SYSTEM FOR A LIQUID PROPELLANT GUN

by

Craig Richard Dampier

June 1976

Thesis Advisor: T. M. Houlihan

Approved for public release; distribution unlimited.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Computer Simulation of a Propellant Feed System for a Liquid Propellant Gun		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis June 1976
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Craig Richard Dampier		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1976
		13. NUMBER OF PAGES 71
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Liquid Propellant Gun
Computer Simulation
Fluid Dynamics
Lumped Parameter Approach

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A computer model was developed to simulate a projectile ram-propellant feed system for a Liquid Propellant Gun. Using a lumped parameter approach, a set of simultaneous differential equations was derived for the complex interaction of the propellant fluid, the driving injector and the projectile. The computer model was verified against a 20 mm experimental apparatus. Injector displacement, projectile displacement,

and chamber pressure were compared for a nominal driving pressure of 140 psi. The important system parameters affecting projectile ram time and chamber pressure oscillations were investigated and potential problem areas for testing with actual propellant were identified.

COMPUTER SIMULATION OF A PROPELLANT FEED SYSTEM FOR A LIQUID PROPELLANT GUN

by

Craig Richard Dampier
Lieutenant, United States Navy
B.S., United States Naval Academy, 1970

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN APPLIED SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June, 1976

ABSTRACT

A computer model was developed to simulate a projectile ram-propellant feed system for a Liquid Propellant Gun. Using a lumped parameter approach, a set of simultaneous differential equations was derived for the complex interaction of the propellant fluid, the driving injector and the projectile. The computer model was verified against a 20 mm experimental apparatus. Injector displacement, projectile displacement, and chamber pressure were compared for a nominal driving pressure of 140 psi. The important system parameters affecting projectile ram time and chamber pressure oscillations were investigated and potential problem areas for testing with actual propellant were identified.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to express my deep appreciation to my thesis advisor, Professor Thomas M. Houlihan, for the many hours he spent keeping me on firm theoretical ground and showing me the "ropes" in the Liquid Propellant Gun business. I would also like to offer my special thanks to my wife, Ingrid, who spent innumerable hours keypunching this thesis into the Naval Postgraduate School's Text Processing system and acting as the "real" first reader.

I. INTRODUCTION

As the state of the art of weapons technology improves, more complex systems are continually being developed. To meet future needs these systems must have improved performance, less weight and volume, less complicated logistics, and at the same time be more cost effective than current systems. One candidate for these future weapon systems is the liquid propellant gun (LPG). Numerous investigators over the past thirty years have demonstrated its potential if the technology can be developed to utilize a liquid propellant in a large caliber gun.

One important area that needs investigation is the aspect of fluid propellant handling. The ordnance designer must know the necessary propellant flow rates and pressure regimes needed to achieve the desired system performance. These parameters must be within allowable safety margins for the handling and use of the explosive propellant. One possible design of an LPG which would allow a very high firing rate utilizes the propellant to ram the loaded projectile. Combining the normally separate propellant load and ram cycles decreases the required movement of mechanical parts. Thus, using this method, it is feasible that firing rates in excess of four to five times present rates could be achieved. This thesis was directed toward identifying the important fluid dynamic parameters involved with such a projectile ram feed system.

II. BACKGROUND

During 1974 and early 1975, an investigation was conducted at the Naval Postgraduate School to study, both analytically and experimentally, the fluid dynamics of a liquid propellant under conditions similar to those which would exist in a rapid-fire LPG feed system. The results of this investigation were to be used to establish such LPG design and performance parameters as time-to-load, injection supply pressure, injection system configuration, ullage, charge-to-mass ratio, caliber size, and projectile mass. It was hoped that this investigation would identify any potential problem areas for further detailed research.

The basic objective of the experimental portion of the investigation was to identify what fluid dynamic characteristics of a liquid propellant feed system would limit loading times and hence rates of fire. A 20 mm experimental model of a basic propellant feed system was designed and built. Data on injector displacement, breech chamber pressure, and ram gas pressure (input driving pressure), were recorded for driving pressures between 50 and 220 psig in 10 psi increments. As reported in Ref. 1, it was found that the instantaneous behavior of the chamber pressure was the result of a complex interaction of inertia forces, viscous forces, and the unsteady motion of the fluid. The experiments demonstrated that frictional and inertial effects were significant during the movements of the injector and the projectile slug. Once the projectile slug stopped, the effect of entrapped gas in the fluid caused large breech chamber pressure oscillations. On several runs, sub-atmospheric pressures were experienced which suggested the possibility of cavitation and hence vapor-phase ignition. Over all, it was found that the ram time had a quadratic dependence upon ram pressure.

The analytical portion of the investigation was directed toward predicting the pressure and flow rate of the propellant and the projectile slug motion during an LPG loading cycle. The mathematical model which was developed and programmed on an analog computer predicted the position, velocity, and acceleration of the projectile and the pressure at various points in the system as functions of time. The model was tested against experimental results and found to be adequate for the prediciton of projectile ram time. An analysis was performed using the model to indicate the areas of system redesign likely to be most profitable and to obtain preliminary predictions of LPG loading system performance under a variety of design conditions. The results of the analytical study, as well as a summary of the experimental work is contained in Ref. 2.

The aforementioned model was the starting point for the present study. To better understand the derivations in the following sections, Figure (1) indicates the geometry of a basic projectile ram propellant feed system. An injector chamber is filled with propellant, which is then pumped into the gun breech by applying ram pressure to the ram side of the injector piston. This can be accomplished by using high pressure gas from an accumulator or by using an hydraulic drive system. The force exerted by the injector piston drives the fluid propellant through the connecting line into the gun breech. The rapidly accelerated propellant drives the already loaded projectile to its seated position ready for firing.

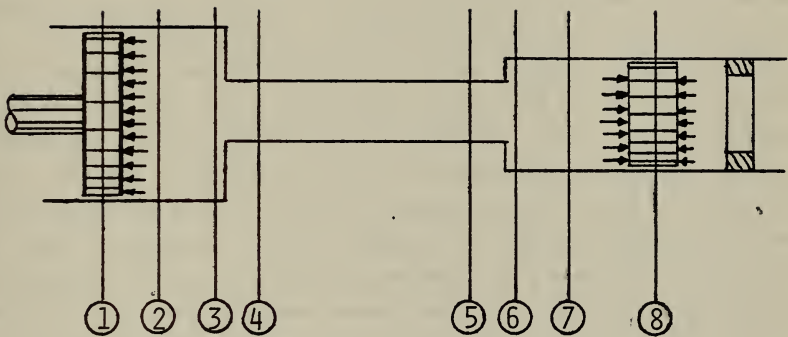


Figure 1 - BASIC GEOMETRY

The complex nature of this dynamic ram process is not easy to describe with mathematical equations. Many researchers have approached the problem using wave mechanics and partial differential equations. These equations were then modified according to the propagation characteristics of the system and the various boundary conditions encountered therein. This usually led to involved finite difference methods of solution on a digital computer.

Another possible approach considers the kinetic energy involved in the ram process. This is the approach that was used to derive the governing equations for the previously described analog computer model. Beginning with the input side, a force balance on the injector piston yielded an equation for the pressure in the injector chamber in terms of the injector motion. Writing Bernoulli's equations for head losses at the expansion to the breech chamber, the contraction to the connecting line, and an orifice in the connecting line resulted in an equation for the pressure drop in the connecting line in terms of the square of the fluid velocity and the fluid acceleration. A force balance on the projectile slug yielded a third equation which described the breech chamber pressure in terms of the slug motion. To solve these equations, some method of relating the motions in the injector, connecting line, and breech chamber had to be established. It was assumed that the fluid was incompressible and therefore these motions were identical. To facilitate the analog simulation it was also assumed that the input ram pressure was a step input. In the process of converting these equations to an integral form for wiring on an analog computer, all variables were normalized to permit scaling.

Despite these limiting assumptions, the analog computer model was able to predict the time to ram the projectile and the chamber pressures which occur during the time that the injector is in motion. Due to the incompressibility assumption, the model was unable to account for the pressure transients or "water hammer" type pressure oscillations which occurred when the projectile was seated and the propellant fluid was being decelerated. These pressure oscillations are important to the system designer because the peak breech chamber load pressure is experienced during these oscillations. It is also possible that these oscillations could interfere with the uniformity of the ignition and subsequent combustion of the propellant.

A straight forward attempt was made to extend this analog model by first converting it to a digital computer program. This allowed an accurate representation of the input ram pressure and the use of unnormalized variables to be incorporated. These improvements increased the accuracy of the model but the effect of the compressibility of the propellant fluid still was not taken into account. Trying to add compressibility effects to this model by adding time derivatives of the pressure terms to the governing equations unlinked the motions of the injector, the connecting line fluid, and the projectile slug causing a problem with too many unknown variables for the number of equations involved.

It was decided that a new approach should be tried which would adequately describe the system pressure oscillations. The approach would feature an engineering model which would be easy to use, adaptable to any LPG feed system, and not obscure the interaction of system parameters by complicated mathematics.

III. DERIVATION OF COMPUTER MODEL

A. LUMPED PARAMETER APPROACH

The approach that showed the greatest promise for modeling the LPG feed system was that of the fluid transmission line concept. This approach, which has become popular in the last five to ten years, is based on a pressure-voltage and flow velocity-current analogy with Electrical Engineering determinations. It is an outgrowth of the large amount of effort that has been devoted to investigating fluid line transients. Reference 3 is a good survey of this field.

For complex systems this approach is usually simplified by using the approximation of lumped parameters. The effects of fluid inertia, capacitance, and resistance are "lumped" and considered to act only in discreet areas of the system. This results in a reduction of the unknown parameters due to the assumed lack of interaction of the different fluid effects. This approximation, however, results in the necessity of using several empirical constants which must be determined by fitting experimental data. The ordinary differential equations that are derived from this method can be solved either by Laplace transformation or by computer integration.

To consider the effect of fluid inertia it is assumed that only pressure and inertia forces are present and that compressibility effects are negligible in the volume under consideration. Then,

$$P_1 - P_2 = \rho L \frac{dV}{dT}$$

where ρ is the propellant fluid density, L is a characteristic length and $V_1 = V_2 = V$ because the flow is considered incompressible for this building block (Fig.2A).

To consider the effect of fluid capacitance, it is assumed that only compressibility effects are important, and that inertia and resistance effects may be neglected in the volume under consideration. Therefore,

$$V_1 - V_2 = \frac{L}{K} \frac{dP}{dT}$$

where K is the effective bulk modulus of the propellant fluid, L is a characteristic length and $P_1 = P_2 = P$ (Fig. 2B).

Because fluid resistance can be affected by so many different parameters, it is impossible to write a general equation describing the pressure drop due to fluid resistance. It is best to treat it empirically using an experimentally derived figure. Thus, in the volume under consideration,

$$P_1 - P_2 = Rv\ V$$

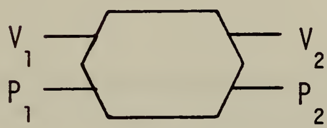
where Rv is a function of fluid velocity and $V_1 = V_2 = V$ (Fig. 2C). Changes in cross sectional area can be accounted for by using appropriate area ratios.

These results specify three building blocks which can be combined in any sequence to model the dynamic characteristics of a system. The complexity of the model can be increased to any degree necessary by including more and more combinations of these three basic building blocks.

A) INERTIA

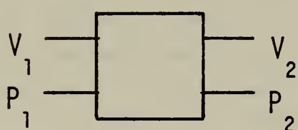
$$P_1 - P_2 = \rho L \frac{dV}{dT} \qquad V_1 = V_2 = V$$

B) CAPACITANCE



$$V_1 - V_2 = \frac{L}{K} \frac{dP}{dT} \qquad P_1 = P_2 = P$$

C) RESISTANCE



$$P_1 - P_2 = Rv\ V \qquad V_1 = V_2 = V$$

Figure 2 - LUMPED PARAMETER MODULES

B. LPG SYSTEM GOVERNING EQUATIONS

Several different combinations of building blocks were tried in attempting to model the experimental LPG system. The experimental results showed that inertial and resistance blocks should be included in the model. The fact that pressure transients occurred in the breech chamber indicated that a capacitance block should also be included. To minimize the use of computer time, it was decided to start with the most simple model thought adequate and work toward more complex models as the quality of the computer results dictated.

The first attempt considered that compressibilty effects would dominate in the injector chamber as the accelerating piston interacted with the propellant fluid. The resistance effects of changes in the feedline cross sectional diameter and associated orifices were modeled as one resistance block at the beginning of the connecting line. Inertial effects were thought to dominate in the connecting line. This would be particularly true in large scale models where the length of the connecting line would be very large. It was thought that compressibility effects should again dominate in the breech chamber as the accelerating propellant fluid drove the projectile down the breech chamber. Results in this case were governed by inertial effects because the calculated connecting line velocity was too great. This led to a decrease in the chamber pressure as the projectile accelerated until a large negative chamber pressure existed. In addition to this deficiency, the computed system pressure oscillations were not sufficiently damped.

To try to adjust the connecting line velocity, another resistance block was added. As will be seen in the section discussing the parameters which affect the pressure oscillations, the value of this resistance coefficient is proportional to the amount of damping in the pressure oscillations. Even with a value which gave the proper damping, the breech chamber pressure was dominated by the pressure loss due to the accelerating fluid in the connecting line.

To reduce the dominance of the fluid inertia effects, it was decided to restructure the model. To model the injector chamber it was decided to consider the effect of fluid capacitance and inertia. The connecting line was modeled by two resistance blocks separated by a fluid capacitance block. The breech chamber was modeled using a fluid capacitance and a fluid inertia block, as was done for the injector chamber. This model gave the final results described in the section comparing computer and experimental data. By combining these building blocks, the governing equations became: (See Fig. (1) for notation)

For the injector:

1. $P_R A_1 - P_1 A_1 = M_P \dot{V}_1 + K_{FS} V_1 + P_{DI} A_1$

2. $V_1 - V_2 = \frac{L_1}{K} \dot{P}_1$

3. $P_2 - P_3 = \rho L_1 \dot{V}_2$

For the connecting line:

4. $P_3 - P_4 = Rv_1 V_4$

5. $V_4 - V_5 = \frac{L_2}{K} \dot{P}_4$

6. $P_5 - P_6 = Rv_2 V_5$

For the breech chamber:

7. $P_6 - P_7 = \rho L_3 \dot{V}_6$

8. $V_7 - V_8 = \frac{L_3}{K} \dot{P}_8$

9. $P_8 A_3 - P_{DS} A_3 = M_S \dot{V}_8 + K_{FS} V_8$

These equations were subsequently modified to account for changes in cross-sectional area and changing geometry as the injector chamber decreased in volume and the breech chamber increased. Once the projectile reached the end of the breech chamber, the force balance equation was no longer considered applicable. At this time, it was noted that the remaining equations for the connecting line and the breech chamber could be combined to form a second order differential equation for the breech chamber pressure which described the "water hammer" type pressure oscillations displayed by the experimental data. Thus,

$$10.\ \ddot{P}_8 + \frac{A_3}{A_2}\frac{Rv_2}{\rho L_3}\dot{P}_8 + \frac{K}{\rho L_3 L_3}P_8 = \frac{K}{\rho L_3 L_3}P_4$$

Once the injector piston reached the end of its travel, the injector chamber no longer existed and the governing equations were considered as no longer applicable. Ultimately, the injector chamber pressure was propagated down the connecting line as the propellant fluid came to rest. Throughout these analyses, input ram pressure was modeled using exponential terms to fit experimental data.

The equations from the lumped parameter approach for the propellant fluid were combined with the force balance equations for the injector piston and the projectile using State Variable methods. Subsequently, a computer program was constructed to solve these state variable relations using a fourth order Runga-Kutta integration routine for simultaneous first order differential equations which was developed at the Naval Postgraduate School. A brief description of the state variable method and a listing of the computer program can be found in Appendix A.

The following is a listing of the nomenclature and the values of constants used in the derivation of the governing equations and the computer program:

A1	Injector Cross-Sectional Area	1.77 in^2	11.42 cm^2
A2	Connecting Line Cross-Sectional Area	.255 in^2	1.65 cm^2
A3	Breech Chamber Cross-Sectional Area	.49 in^2	3.16 cm^2
AR	Ram Piston Cross-Sectional Area	1.77 in^2	11.42 cm^2
L1	Injector Length	1.60 in	4.22 cm

L2	Connecting Line Length	16.0	in	40.6	cm
L3	Breech Chamber Length	5.0	in	12.7	cm
MP	Injector Piston Mass	.0052	$\frac{lb\text{-}sec^2}{in}$	913.8	gm
MS	Projectile Mass	.00053	$\frac{lb\text{-}sec^2}{in}$	93.1	gm
KFP	Injector Friction Factor	.001	$\frac{lb\text{-}sec}{in}$	175.7	gm/sec
KFS	Projectile Friction Factor	.001	$\frac{lb\text{-}sec}{in}$	175.7	gm/sec
PDI	Injector Back Pressure	24.0	PSI	163.3	kPa
PDS	Projectile Back Pressure	.50	PSI	3.4	kPa
K	Effective Bulk Modulus	3200	PSI	6808	kPa
RHC	Propellant Density	.000093	$\frac{lb\text{-}sec^2}{in^4}$	.994	gm/cm^3
PR	Ram Pressure			Computed	
P1	Injector Chamber Pressure			Computed	
V1	Injector Piston Velocity			Computed	
V2	Injector Chamber Exit Velocity			Computed	
P4	Connecting Line Pressure			Computed	
V6	Breech Chamber Entrance Velocity			Computed	
P8	Breech Chamber Pressure			Computed	
V8	Projectile Velocity			Computed	

The treatment of several areas of the computer model continually reappeared as requiring refinement. The governing differential equations worked well during the dynamic portion of the feed cycle but experienced difficulties during the initial and final static periods. The handling of static friction and back pressure as a constant value created the possibility of negative velocities until the driving ram pressure overcame the system back pressure. These negative values never occurred in the real system due to the geometric restraints on the injector piston and projectile; therefore, the computer model had to be manipulated to maintain this condition. Unfortunately the need for simultaneous solution of the governing equations made it difficult to manipulate the initial conditions without greatly increasing the complexity of the computer model. After the projectile stopped, the transition from the force balance equation on the projectile to an equation describing the fluid velocity in response to the pressure transients was awkward. No simple differential equation describes the complexity of the wave mechanics involved with the reflection of the pressure waves in the system. In this regard several alternate approaches involving sequential alterations of the governing equations were tried with varying degrees of success.

IV. DESCRIPTION OF EXPERIMENTAL APPARATUS

In order that the comparison of the experimental and computer generated data can be fully understood, a brief description of the NPS experimental apparatus and the conduct of the associated experiments is included (See Ref. 1, pages 17-27).

The test chamber was fabricated from a three-inch O.D. Lucite cylinder, 18 inches long, bored to a 20 mm inside diameter and fitted with aluminum end caps. The chamber was loaded with a brass slug weighing 93 grams which rode on two graphite filled Teflon sealing rings. The brass slug, which simulated the projectile, was cycled from the breech end of the chamber to the barrel end and returned to the breech end, completing one hypothetical firing cycle.

Because of the desire to vary the charge to mass ratio, a variable chamber velocity was necessary. To accomplish this with one chamber, a volume control retaining rod was designed into the system. This brass rod, bored to allow gas to pass its length, was threaded through a plate which was attached to the barrel end cap holding the rod in the chamber. The rod, which has a Teflon disc on the end, not only established the volume of the test chamber, but provided a buffer stop for the slug at the end of its forward motion. Another Teflon buffer was affixed to the breech end cap to cushion the slug in return motion.

The LPG simulator started a simulated firing cycle with the slug at the breech end of the empty chamber, as shown in Fig. (3).

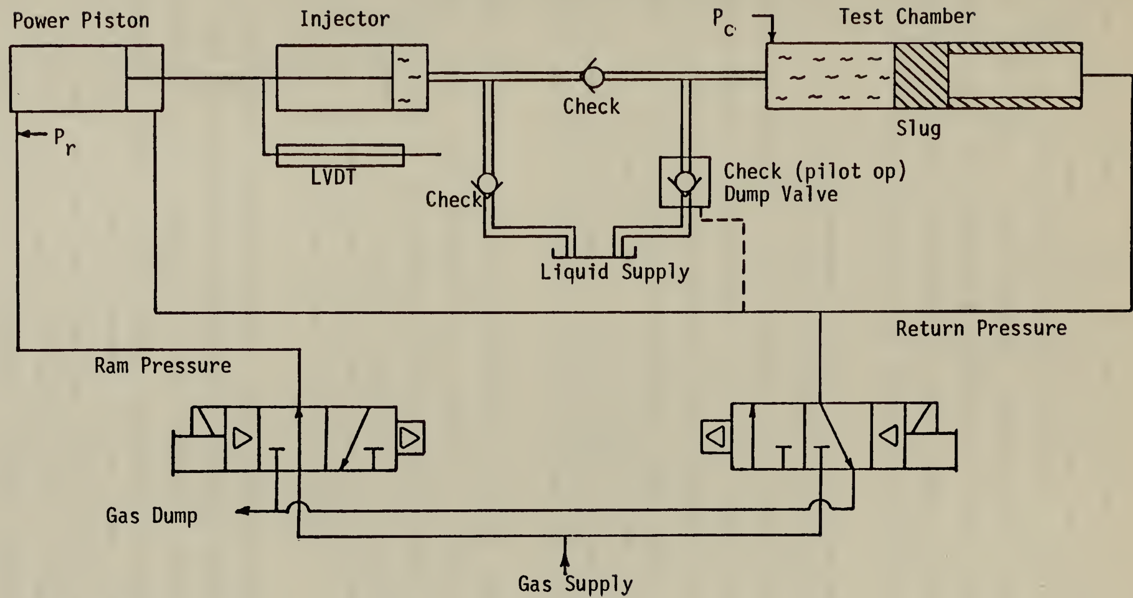


Figure 3 - PNEUMATIC CONTROL SYSTEM IN RAM POSITION

This is the ready-to-ram position. The simulated propellant, distilled water, was then introduced, ramming the slug to the opposite (barrel) end of the chamber as the chamber was filled. This was accomplished by applying gas pressure to a power piston which drove the injector piston. The injector piston forced the propellant past a flow check valve and into the chamber. This placed the slug in the ready-to-fire or in-battery position. In an actual gun, the propellant would be ignited at this time in the cycle. Due to laboratory constraints, an expulsion system was used.

An LVDT displacement transducer was manufactured and mounted next to the injector piston. The LVDT was attached to the connecting rod, between the power piston and the injector piston. The volume of the liquid being placed under pressure during each shot was measured by filling the system in the ready-to-ram position and then draining it into a graduated beaker. By measuring the displacement of the injector piston head with the LVDT, the volumetric rate of fluid injection during the ram stroke was obtained.

Two pressure taps were drilled in the test chamber. These taps were located as close to the breech end as possible, one at 20 degrees from top center and the other at 20 degrees from bottom center in a counterclockwise direction, as seen from the breech end of the chamber. Only the bottom location was used to record data.

A 4-channel Hewlett-Packard 3960 Magnetic Tape Recorder was used to FM record desired data during system operation. For each data run signals were recorded from two Kaman diaphragms type (1000 psi) pressure transducers, one connected to the breech pressure tap on the test chamber, and the other to the gas injection side of the power piston. The pressure signals were processed with a Kaman Digi-Vit Readout Unit which also provided a visual (digital) display.

Data from the IVDT displacement transducer was also recorded on the tape.

A Brush Recorder (Mark 280) was used to obtain a visual display of the recorded data. By transcribing the desired signals on the Magnetic Tape Recorder at a tape speed of 15 feet per minute and playing them back into the Brush Recorder at 3-3/4 feet per minute, the time scale of the output was expanded by a factor of four on the Brush recordings (viz., from a real-time maximum of 200 mm/sec to a delayed time maximum of 800 mm/sec.)

V. COMPARISON OF COMPUTER AND EXPERIMENTAL DATA

As mentioned previously, experimental data was taken for ram gas pressures ranging from 50 to 220 psig. From this collection of data, one run at 140 psig was selected as representative of system performance. It was felt that any model which would adequately describe system operation at this intermediate pressure would be valid for the entire range of expected LPG driving pressures.

Figure (4) is a comparison of computed and recorded data for injector piston displacement. As can be seen, the agreement is very good. This is not very significant in that all models tried, as well as the original analog computer model, were able to correctly predict injector piston motion.

Figure (5) is a comparison of analytical results and experimental data for breech chamber pressure. The computer model follows the shape of the experimental curve for the duration of the time that the projectile is in motion (0-30 msec). It oscillates very rapidly but does not fall off to a zero value at the end of the projectile motion. The frequency of these initial oscillations is approximately 1.0 khz which is within the frequency range of the Kaman pressure transducers used in testing. However, the mounting of these transducers within a connecting cavity instead of flush with the chamber wall could have led to them experiencing a reduced, lagging frequency response. It is felt that some, if not all, of the computed pressure oscillations must exist as evidenced by the close correspondence of the first two peaks in Fig. (5). The accuracy of the experimental pressure reading could also be considered as being reduced by the location of the pressure transducer in the test setup. It is possible that the monitoring pressure tap was too close to the end of the breech chamber and may not have sensed the full chamber pressure during the latter part of the projectile's travel when the fluid velocity is greatest. It is felt that to adequately describe the pressure decrease toward zero, it would be necessary to completely account for exact changes in system geometry. To do this would require using a distributed parameter approach featuring variable lengths for the injector chamber and breech chamber.

The pressure transients which occur after the projectile stops are complex interference phenomena which are not fully described by the model. However, the peak pressure and the natural frequency of oscillation of the computer model compare favorably with the experimental data. Unfortunately, the damping characteristics of the model do not follow the experimental data well. The system parameters which determine these values will be discussed in the next section.

It should be reiterated that the lumped parameter approach is an approximation. In a distributed parameter approach, pressure and velocity would vary continuously with distance within the system as well as with time. In the lumped parameter approach, these changes are assumed to occur only at the input and output of a building block; therefore, pressure and velocity are considered constant within the building block. This will always lead to some discrepancies when comparing model data to experimental data taken at <u>at a fixed point.</u>

Figure (6) shows the simultaneous pressure history for several areas of the LPG feed system. The input ram pressure, injector chamber pressure, connecting line pressure, and breech chamber pressure are graphed as functions of time.

Figures (7) and (8) show the simultaneous velocities at several points as predicted by the computer model. The injector piston velocity, fluid propellant velocities at the beginning and end of the connecting line, and projectile velocity are plotted as functions of time.

The injector and projectile displacements as functions of time are included as Figure (9). As can be seen from Figures (7), (8), and (9), the movement of the injector piston is much more rapid than that of the projectile. It is felt that this velocity difference is very much a function of system geometry and should not be considered as a generalization for all LPG feed systems. The NPS experimental apparatus being modeled had only small pressure drops between injector and breech chambers. In addition, the projectile slug was an order of magnitude lighter than the injector piston. In large scale systems, with significant pressure drops and very massive projectiles, it is quite possible that the injector piston's full stroke will occur significantly before the projectile has seated.

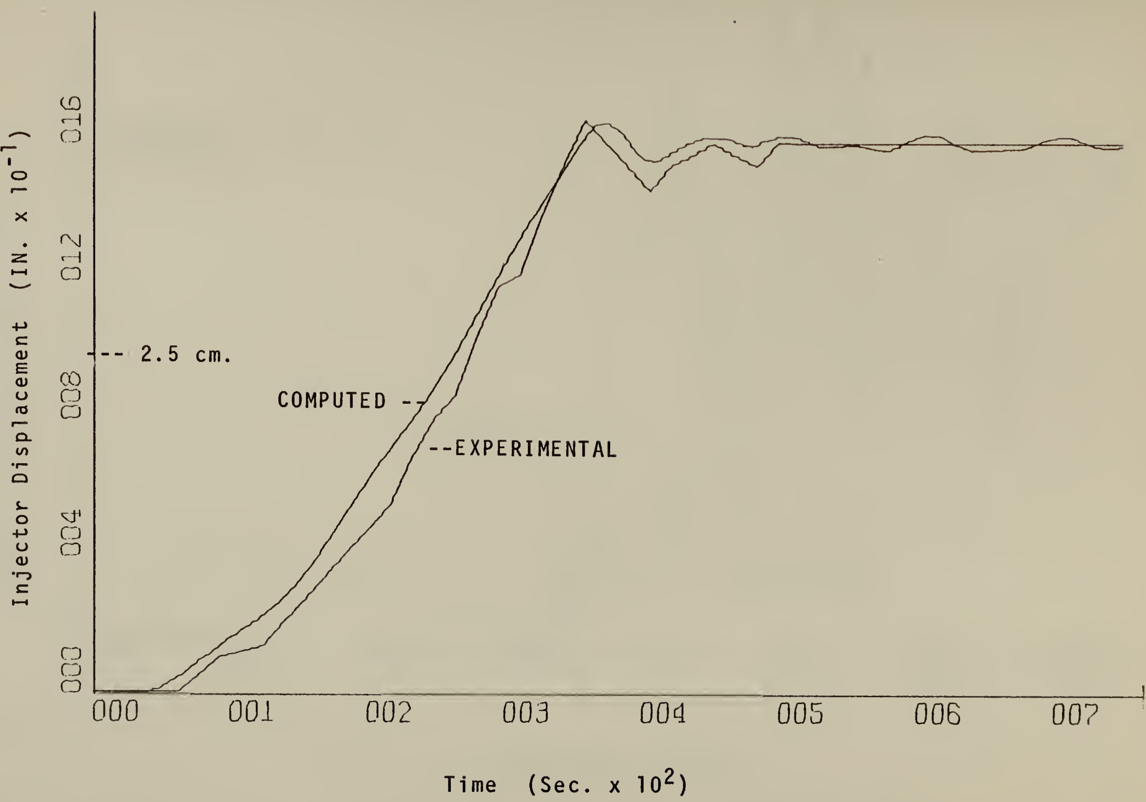


Figure 4 - COMPARISON OF INJECTOR PISTON DISPLACEMENT

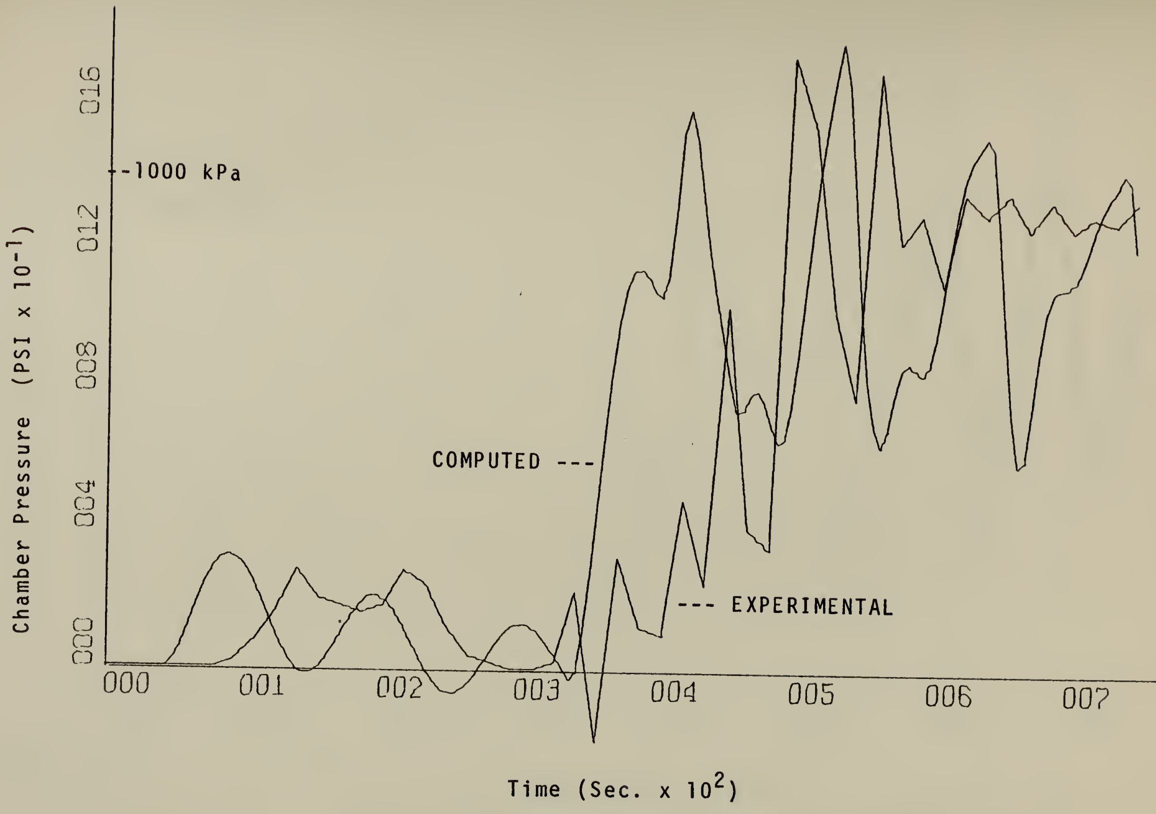


Figure 5 - COMPARISON OF BREECH CHAMBER PRESSURE

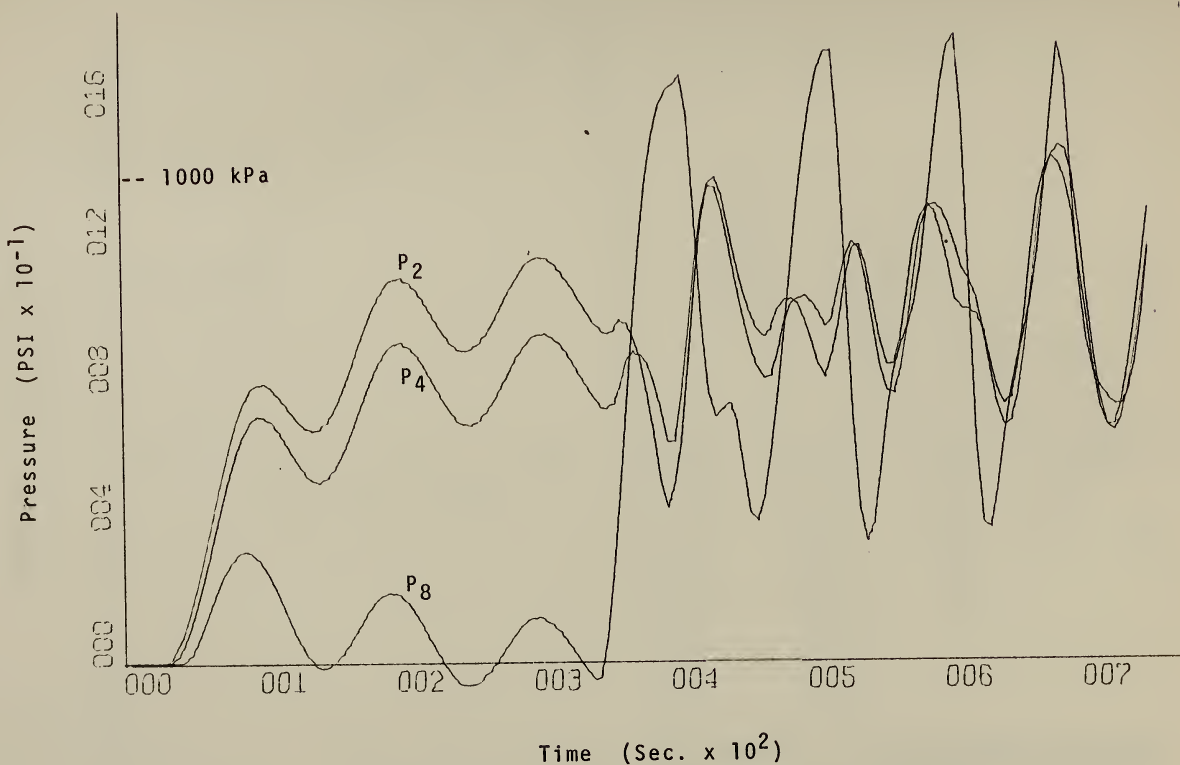


Figure 6 - SIMULTANEOUS PRESSURE HISTORIES

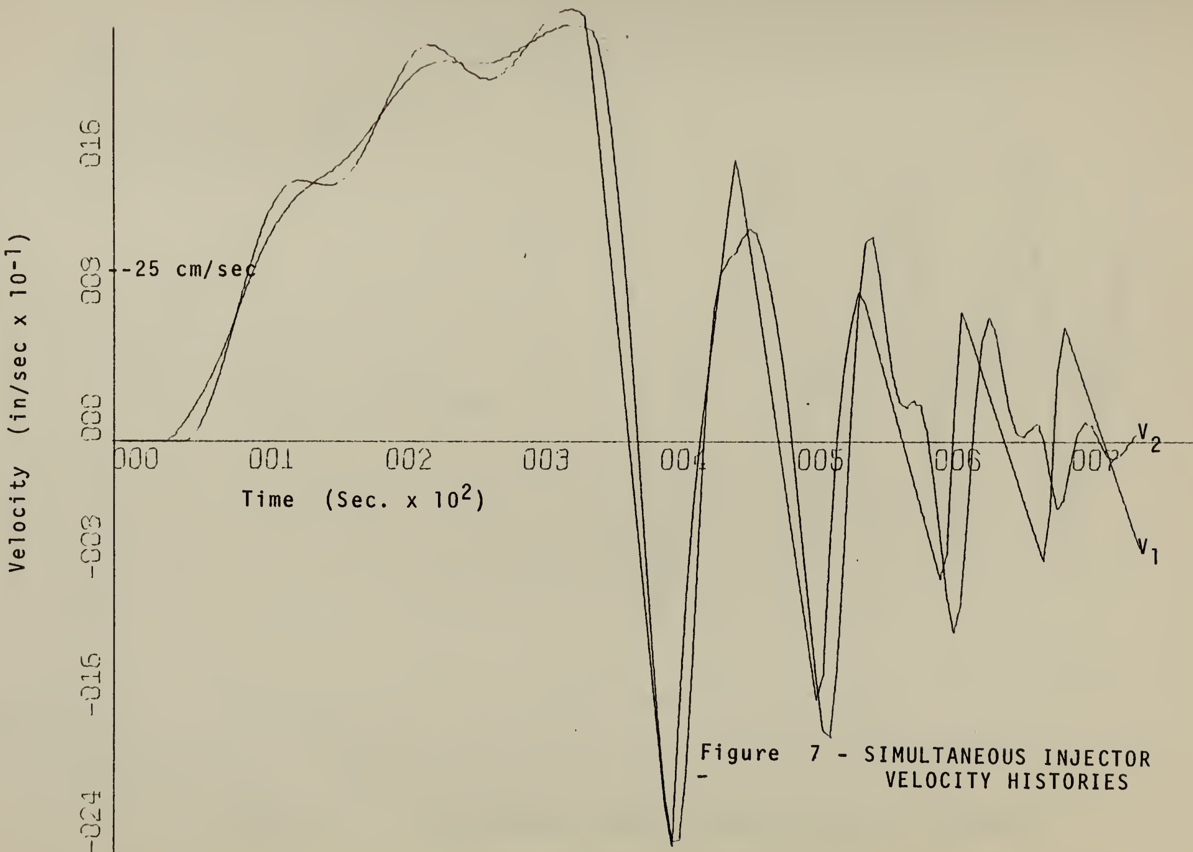


Figure 7 - SIMULTANEOUS INJECTOR VELOCITY HISTORIES

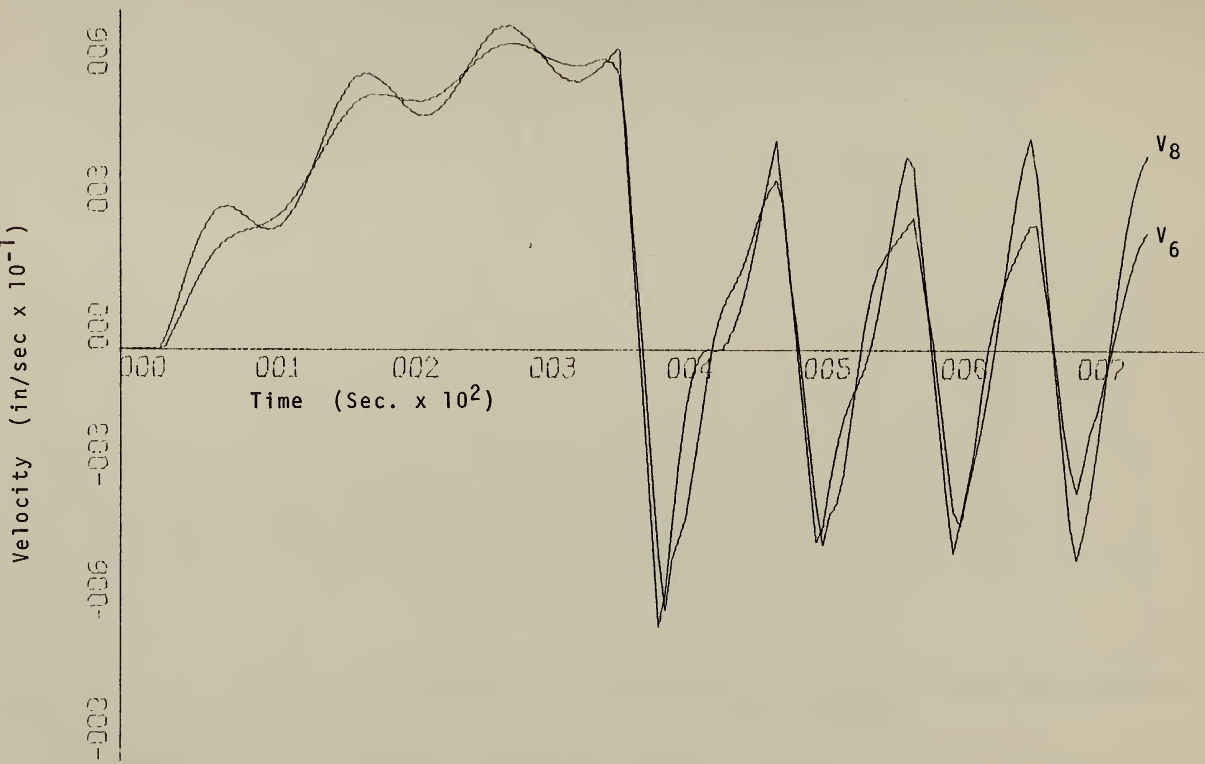


Figure 8 - SIMULTANEOUS BREECH CHAMBER VELOCITY HISTORIES

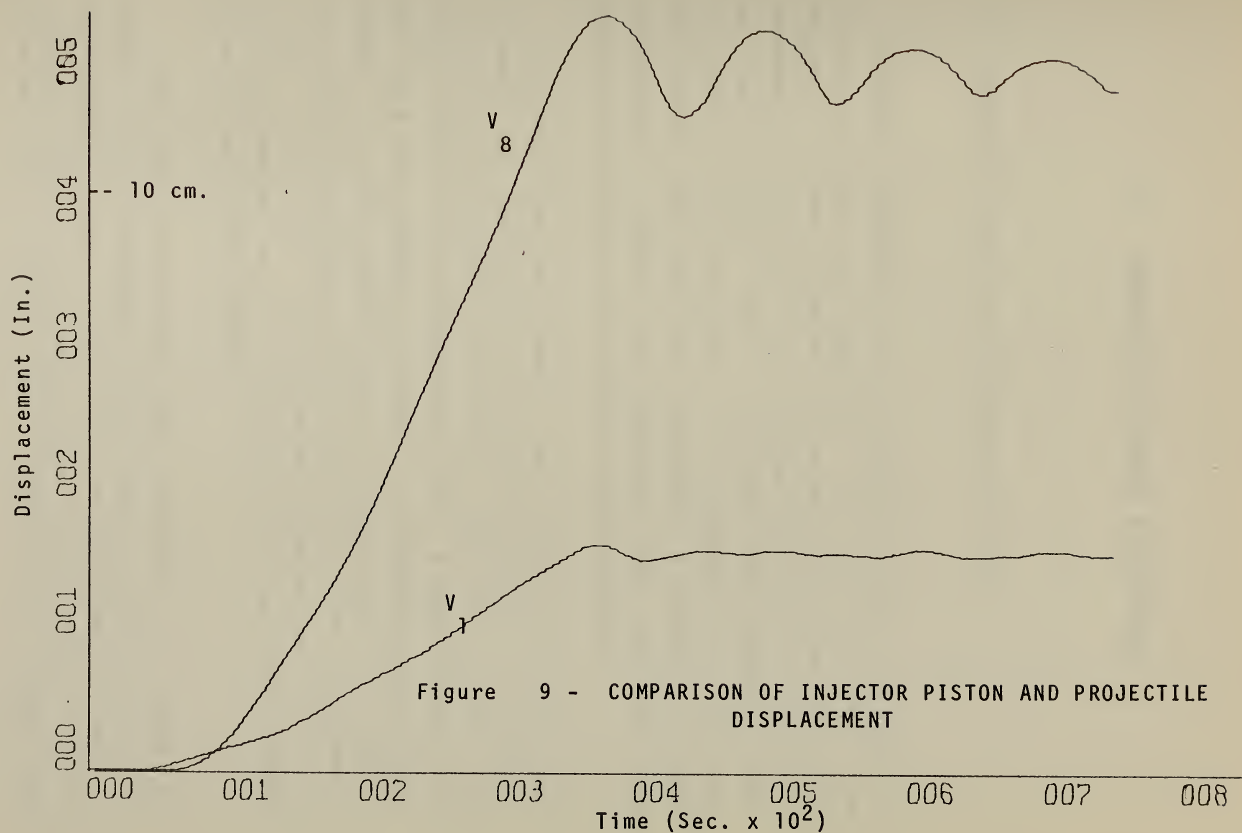


Figure 9 - COMPARISON OF INJECTOR PISTON AND PROJECTILE DISPLACEMENT

VI. ANALYSIS OF SYSTEM PERFORMANCE

A. PARAMETERS AFFECTING PROJECTILE RAM TIME

Several parameters can be varied to "tune" the model to correct ram time. In equations (1) and (9) (the force balances on the injector piston and projectile) empirical constants are included which account for any static friction and back pressure (PDI = injector back pressure, PDS = projectile back pressure). In equations (4) and (6), resistance coefficients are introduced to account for system resistances. Finally in equations (2), (5), and (8), propellant effective bulk modulus values appear.

The value for the injector and projectile total back pressure is difficult to determine precisely. Reference 1 cited an experimentally determined value of 24 psi. The analog computer model presented in Ref. 2 used a value of 46 psi for a ram pressure of 140 psi. The present model uses a value of 24 psi for PDI and 0.5 psi for PDS, or a total of 24.5 psi back pressure, which agrees with the experimental value. The value of system back pressure will vary with driving pressure and the configuration of each LPG system's injector and breech chambers.

Some basic research has been conducted to relate the value of the resistance coefficients to fluid properties. Unfortunately these studies which are described in Refs. 4 and 5, dealt with steady state fluid flow in constant diameter lines without flow restrictions, making their results inapplicable to the present LPG feed system problem. The value of RV_2 can be related to the damping ratio of the "water hammer" pressure oscillations. The value of RV_1 was determined by an iterative process to obtain the best fit to experimental data. Figure (10) shows the dependence of ram-time on RV_1 and RV_2, keeping all other variables constant. As expected intuitively, increasing the fluid resistance creates a larger system pressure drop, resulting in a lower pressure exerted on the face of the projectile and hence slower ram times. The slope of the two curves are almost identical so that no significant advantage would be achieved by system designs which try to minimize either RV_1 or RV_2 to the detriment of the other. It should be noted, however, that higher values for RV_2 do tend to slow down the ram time more than high values for RV_1. Since the magnitude of RV_2 would probably depend mostly on the pressure drop at the gun valve, which seals the breech chamber, its design should be closely watched to ensure rapid ram times. Likewise the system designer will have to pay close attention to the design of the piping and valves in a large scale LPG feed system to achieve optimum performance.

The value of the effective bulk modulus was determined from the undamped natural frequency of the "water hammer" pressure oscillations as will be discussed in the next section. The bulk modulus is a fluid parameter which characterizes the spring effect of a liquid. The bulk modulus can be substantially lowered by the elasticity of the chamber and connecting line walls and the amount of entrapped gas present in the propellant fluid. As can be seen from Figure (11), the value of the bulk modulus has only a small effect on ram time. Between bulk modulus values of 10,000 and 100,000 psi (the expected operating region of an operational system) the ram time is almost constant, varying less than a quarter of a millisecond. This is an encouraging result since minimizing entrapped gas is a difficult and costly design constraint.

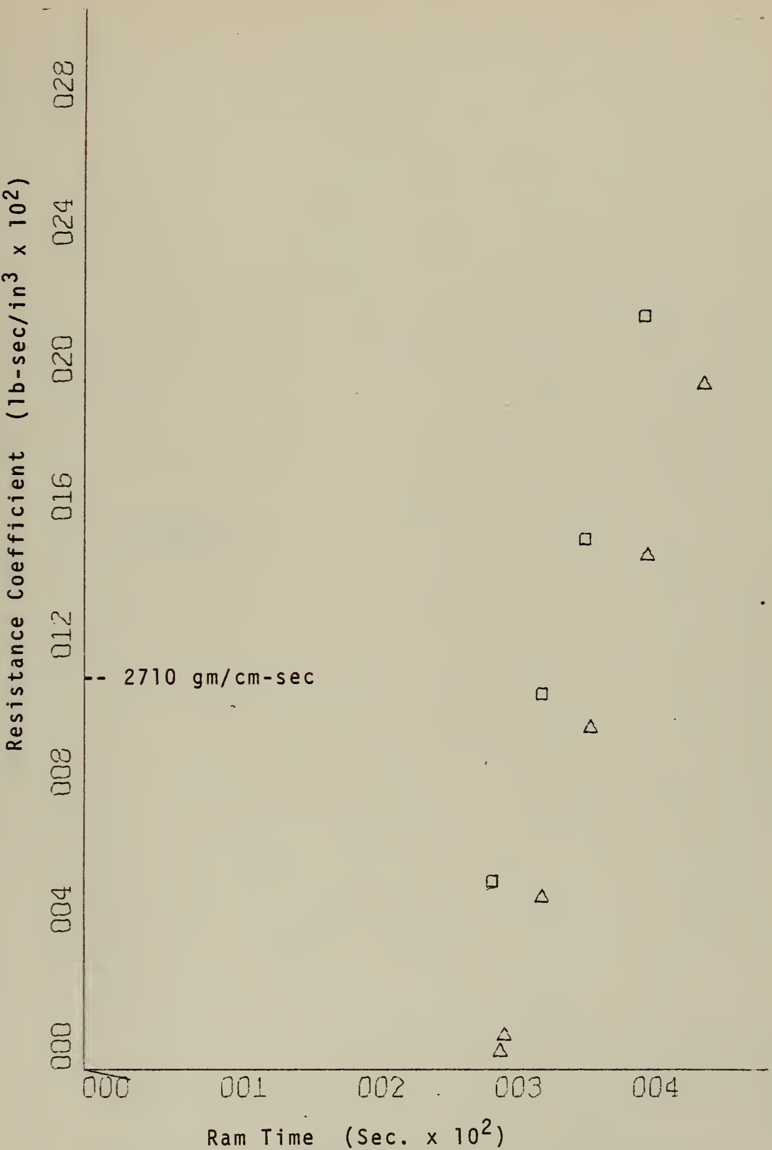


Figure 10 - RESISTANCE COEFFICIENTS VS RAM TIME

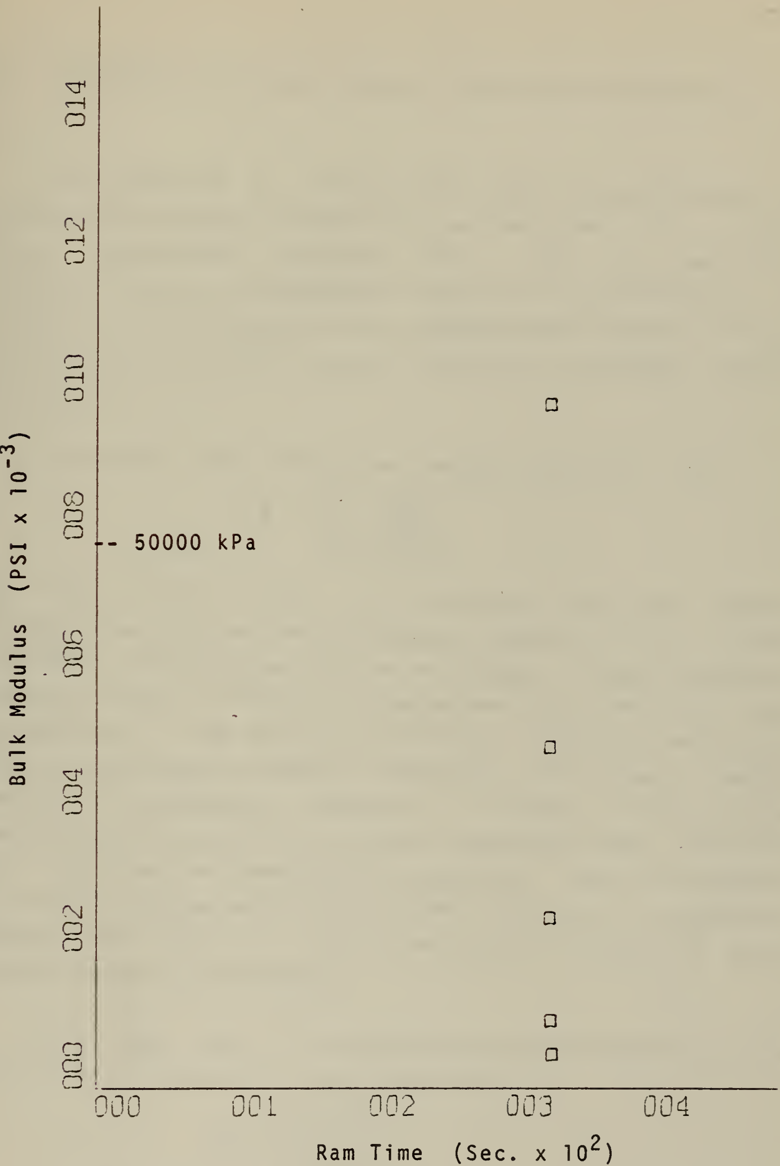


Figure 11 - BULK MODULUS VS RAM TIME

B. PARAMETERS AFFECTING CHAMBER PRESSURE OSCILLATIONS

As explained in Section III, after the projectile has stopped the chamber pressure can be described by a second order differential equation (Eqn. 10). If the system described by this differential equation is underdamped, the solution to the equation is an exponentially damped sinusoid with a characteristic damping ratio and undamped natural frequency.

From Eqn. (10), the system damping ratio, γ is:

$$\gamma = \frac{A_3}{A_2} \frac{Rv_2}{2\sqrt{\rho K}}$$

If this damping ratio is increased, the peak chamber pressure as well as the dissipation period of the system pressure oscillations will be decreased. From a designer's point of view, it would be advantageous for both of these quantities to be as low as possible. Hence, it would seem that an increase in system resistance to enhance damping would be desirable. However, as shown in the previous section,an increase in the system resistance also increases the ram time- an undesirable occurrence. Thus, alterations in system configuration which affect resistance coefficient values will have to be accomplished carefully to insure optimal system performance.

From Eqn. (10), the undamped natural frequency of the pressure oscillations in the LPG system is:

$$W_N = \sqrt{\frac{K}{\rho L_3 L_3}}$$

The primary effect of ullage would be to decrease the value of effective bulk modulus and consequently to decrease the natural frequency of pressure oscillations. Hence, should it be found in future tests that the actual liquid propellant is highly sensitive to pressure oscillations, changes in ullage may be effected to decrease system ringing. Fortunately, as shown in the previous section, the effect of ullage variations on ram time is minimal.

VII. CONCLUSIONS

Several important conclusions can be drawn from the output of the LPG ram feed computer model. Thus, it is realized that serious design consideration must be given to the values of fluid resistances in the feed system in order to minimize ram time and peak pressures, and to optimize the damping of system pressure transients. Fortuantely, concommitant model results also show that the effect of entrapped gas is minimal for the overall operation of a LPG feed system. The model was unable to account for any cavitation effects so this is an area that must be a subject of further studies.

The fact that large pressure transients occur during the projectile ramming cycle indicates the difficulties that might arise with proposed designs for liquid propellant guns with variable propellant volumes and projectile displacements. The pressure transients experienced in these systems will undoubtedly be very complex in nature.

If further improvements are desired in the modeling of LPG feed systems, the next step should be the inclusion of thermodynamic effects. For any real fluid, resultant feed system pressure and velocity changes will cause temperature changes which may substantially affect the propellant fluid density. Consequently, propellant ignition characteristics at high rates of fire will undoubtedly be affected.

The lumped parameter LPG feed system computer model has been shown to be in favorable agreement with experimental data. It is felt that this model has been sufficiently validiated tc allow its use in more complicated LPG feed system designs. It is expected that this model will be used by the Naval Crdnance Station, Indian Head, Maryland to assist in studying the fluid dynamic characteristics of a 30 mm scale model LPG feed system becoming operational in June 1976. It will be necessary to make the obvious changes to describe the different geometry of the 30 mm scale model. The additicnal line lengths, and the larger pressure drops due to complex valve arrangements in the 30 mm system will require different values for model back pressure and resistance cceffícients. The presence of an accumulator near the breech chamber will affect the value of the system damping ratic. It is possible that the different gecmetry and mass characteristics of the 30 mm model could result in the injector piston being slower than the projectile. However by incorporating all of these alteraticns, the manner in which each of the system parameters affect total system performance will be able to be predicted by the versatile lumped parameter computer model developed in this study.

As such, computer simulation is a useful design tool. However, tc remain useful, it must be viewed in its proper perspective. Full scale prototype testing is the only conclusive method of demonstrating ordnance system performance. Unfortunately, testing is both costly and time consuming. Thus, computer simulation, no matter how simple or complex, can be used to designate critical testing instances and to identify those areas where design efforts will be most productive. In this way, the time involved in optimizing system performance as well as subsequent production costs can be reduced. In explosives research and ordnance design, which is still an empirical and necessarily hazardous science, computer simulation can be particularly useful in specifying the correct approach for design testing practices. In this respect, such models as the lumped parameter model developed in this study can be considered as critical signposts at some of the many crossroads in systems development. However, it must be realized that as such they only point the way. The vehicle for arriving at an operational system can only be diligent research and engineering based on a measured progression of system demonstrations overseen by dedicated project managers aware of the many pitfalls along the way.

APPENDIX A

COMPUTER PROGRAM LISTING

The basis of the state variable method of systems analysis is the interpretation that the energy state of the components of a system completely describes the condition of a system. As defined in Ref. 6, the state of a system is the set of variables, the "state variables", which contain sufficient information about the present condition of the system to permit the determination of all future time history of the system - provided that all future inputs to the system are known. Therefore the energy state of those elements which store energy completely describes the system. In the case of the lumped parameter model, these elements are the fluid inertia and capacitance and the inertia of the injector piston and projectile. The energy state of these storage elements can be described as a function of time in terms of the state variables - pressure, velocity, and the system input pressure. Only those variables required to completely specify the state of the system need to be included.

The first step in arriving at the computer program was to take equations (1) through (9) and normalize, or solve them for the highest derivative. The state variables then become the pressure or velocity associated with these first derivatives. The computer program was developed by combining equations and defining new variables as follows:

$X(1) = V_1$

$X(2) = P_1$
$X(3) = V_2$
$X(4) = P_4$
$X(5) = V_6$
$X(8) = P_8$
$X(9) = V_8$

Equation (1) becomes:

$$XDOT(1) = \frac{-K_{FP}}{M_P} X(1) - \frac{A_1}{M_P} X(2) + \frac{A_R}{M_P} P_R - \frac{A_1}{M_P} P_{DI}$$

In the computer program, PR is represented by X(30) and KFP and PDI become program constants, C(7) and C(2), respectively. The value of C(2) and C(7) are input on data cards. Using the fact that $P_1=P_2$, equation (2) becomes:

$$XDOT(2) = \frac{K}{L_1} X(1) - \frac{K}{L_1} X(3)$$

where K is input to the computer program as C(3). By combining equations (3) and (4) and using the incompressible flow assumption $A_1V_2=A_2V_4$ the next equation is derived:

$$XDOT(3) = \frac{1}{R_{HO}\ L_1} \left(X(2) - R_V \frac{A_1}{A_2} X(3) - X(4) \right)$$

where RV_1 becomes C(4). Next equation (5) and the incompressible flow assumptions $A_1V_2=A_2V_4$ and $A_2V_5=A_3V_6$ yield:

$$XDOT(4) = \frac{A_1}{A_2} \frac{K}{L_2} X(3) - \frac{A_3}{A_2} \frac{K}{L_2} X(4)$$

and as before, C(3)=K. Equations (6) and (7) combine with the flow velocity equation $A_2V_5=A_3V_6$ to yield:

$$XDOT(5) = \frac{1}{R_{HO}\ L_3} \left(X(4) - R_{V_2} \frac{A_3}{A_2} X(5) - X(8) \right)$$

where RV_2 becomes C(5). Since $V_6=V_7$, equation (8) becomes:

$$XDOT(8) = \frac{K}{L_3} X(5) - \frac{K}{L_3} X(9)$$

where K again becomes C(3). Finally, equation (9) becomes:

$$XDOT(9) = \frac{-K_{FS}}{M_S} X(9) + \frac{A_3}{M_S} X(8) - \frac{A_3}{M_S} P_{DS}$$

where KFS and PDS become C(8) and C(9) respectively. The auxiliary equations:

$$XDOT(6) = X(1)$$

$$XDOT(7) = X(9)$$

are used to calculate injector piston displacement, X(6), and projectile displacement, X(7).

```
      DIMENSION X(30),XDOT(30),C(15)
      REAL  MS,MP,L1,L2,L3
      C(10)=1.0
       IFLAG=0
    1 CALL CAMP1(T,X,XDOT,C)
C
C   THE FOLLOWING CONSTANTS DESCRIBE SYSTEM GEOMETRY AND FLUID PROPERTIES
C
      AR=1.77
      A1=1.77
      A2=0.255
      A3=0.49
      MS=5.30E-04
      MP=5.20E-03
      RHO=9.30E-05
      L1=1.60
      L2=16.0
      L3=5.00
C
C        X(30) DESCRIBES THE RAM PRESSURE INPUT AS A FUNCTION OF TIME.
C        THIS IS DERIVED FROM FITTING CURVES TO EXPERIMENTAL DATA.
      IF(T.LE.0.00625) X(30)=C(1)*(1.0-EXP(-125.0*T))
      IF(T.GT..00625) X(30)=C(1)*(1.-EXP(-125.*T))-EXP(75.0*(T-.00625))
      IF(T.GT.0.0375) X(30)=C(1)
C
C
C        THIS SECTION DESCRIBES THE INJECTOR MOTION
       IF(X(6).GE.L1)  GO TO 20
       IF(IFLAG.NE.0)    C(2)=0.0
      XDOT(1)=-(C(7)/MP)*X(1)-(A1/MP)*X(2)+(AR/MP)*X(30)-(A1/MP)*C(2)
       IF(T.LE.0.010)   X(1)=AMAX1(0.0,X(1))
       X1M=X(1)
       GO TO 25
   20 XDOT(1)=-X1M/1.7625E-03
      IFLAG=1
C
   25 XDOT(2)=(C(3)/L1)*X(1)-(C(3)/L1)*X(3)
       IF(X(2).EQ.0.0)   X(4)=0.0
      XDOT(3)=(1.0/(RHO*L1))*(X(2)-(C(4)*((A1/A2)*X(3)))-X(4))
      XDOT(4)=(A1/A2)*(C(3)/L2)*X(3)-(A3/A2)*(C(3)/L2)*X(5)
       IF(X(4).EQ.0.0)   X(8)=0.0
      XDOT(5)=(1.0/(RHO*L3))*(X(4)-(C(5)*((A3/A2)*X(5)))-X(8))
      XDOT(6)=X(1)
       IF(X(5).EQ.0.0)   X(9)=0.0
         IF(T.LE.0.010) X(9)=AMAX1(0.0,X(9))
      XDOT(8)=(C(3)/L3)*(X(5)-X(9))
       IF(X(7).GE.L3)  GO TO 50
```

```
      XDOT(9)=(-C(8)/MS)*X(9)+(A3/MS)*X(8)-C(6)*A3/MS
       XSM=X(9)
      GO TO 100
   50 XDOT(9)=-XSM/3.8530E-03
  100 XDOT(7)=X(9)
      C(11)=100.0
      C(12)=50.0
C
      GO TO 1
      END


      SUBROUTINE DAMP1(/TC/,/XC/,/DX/,/C/)
      REAL*8 ITITLE(12),JTITLE(8),KTITLE(8),IBLANK/'        '/
      DIMENSION X(30),BX(30),XC(30),C(15),IP(10),IG(10),PR(10),GR(10),
     1TX(5),TY(5),X1(900),Y1(900),X2(900),Y2(900),X3(900),Y3(900),
     2X4(900),Y4(900)
      REAL LABEL,RUN(2)/' RUN',' '/,BT(9)/' 1  ',' 2  ',' 3  ',' 4  '
     1,' 5  ',' 6  ',' 7  ',' 8  ',' 9  '/
      EQUIVALENCE (ITITLE(7),RUN(1))
      INDIC = C(10)+0.0000001
      GO TO (1, 2000, 50, 58, 88, 88),INDIC
C
C        READ DATA AND PRINT RECORD.
C
    1 READ(5,100) (ITITLE(I), I=1,6)
  100 FORMAT (10A8)
      READ (5,101)NR
  101 FORMAT (I1)
      READ (5,102)NN
  102 FORMAT (I2)
      NRC = 0
      IF(NN.LE.30) GO TO 1000
      WRITE(6,200)
  200 FORMAT (////,48H ERROR IN ORDER OF EQUATION. MUST NOT EXCEED 30. )
      STOP
 1000 NRC = NRC + 1
      WRITE (6,201) (ITITLE(I),I=1,6)
  201 FORMAT (1H1,/////,36X,6A8)
      IF(NRC.EQ.1.AND.NR.EQ.1) GO TO 5
      WRITE(6,202)NR
  202 FORMAT (/,37X,I1,20H RUNS ARE CALLED FOR )
      GO TO 6
    5 WRITE(6,203)
  203 FORMAT (/,37X,21HONE RUN IS CALLED FOR ,///,18H INPUT DATA RECORD)
      GO TO 7
    6 WRITE(6,204)NRC
```

```
204   FORMAT (///,34H INPUT DATA RECORD FOR RUN NUMBER   ,I1)           LPG00940
    7 WRITE(6,205)NN                                                    LPG00950
205   FORMAT (//,22H ORDER OF EQUATIONS = ,I2)                          LPG00960
      READ (5,103)TI,DT,TF1,DT2,TF2,DT3,TF3                             LPG00970
103   FORMAT (8F10.4)                                                   LPG00980
      TF = TF1                                                          LPG00990
      IF(DT2.NE.0.) GO TO 9                                             LPG01000
      WRITE(6,206)TI,TF                                                 LPG01010
206   FORMAT (22H INITIAL TIME         = ,E10.4,/                       LPG01020
     1        22H FINAL TIME           = ,E10.4)                        LPG01030
      WRITE(6,207)DT                                                    LPG01040
207   FORMAT (22H STEP SIZE            = ,E10.4)                        LPG01050
      GO TO 12                                                          LPG01060
    9 IF(DT3.NE.0.) GO TO 11                                            LPG01070
      TF = TF2                                                          LPG01080
      WRITE(6,206) TI,TF                                                LPG01090
      WRITE(6,208)DT,TI,TF1,DT2,TF1,TF                                  LPG01100
208   FORMAT (22H STEP SIZE            = ,E10.4,13H BETWEEN T = ,E10.4, LPG01110
     1        9H AND T = ,E10.4)                                        LPG01120
      GO TO 12                                                          LPG01130
   11 TF=TF3                                                            LPG01140
      WRITE(6,208) DT,TI,TF1,DT2,TF1,TF2,DT3,TF2,TF                     LPG01150
   12 READ(5,103) (C(I),I=1,8)                                          LPG01160
      READ (5,103)(X(I),I=1,NN)                                         LPG01170
      J = 0                                                             LPG01180
      DO 14 I=1,8                                                       LPG01190
      IF(C(I).NE.0.) J=J+1                                              LPG01200
   14 CONTINUE                                                          LPG01210
      K = 0                                                             LPG01220
      DO 16 I=1,NN                                                      LPG01230
      IF(X(I).NE.0.) K=K+1                                              LPG01240
   16 CONTINUE                                                          LPG01250
      IF(J - 1)17,18,19                                                 LPG01260
   17 WRITE (6,209)                                                     LPG01270
209   FORMAT (/,34H ALL THE CONSTANTS, C(I), ARE ZERO )                 LPG01280
      GO TO 423                                                         LPG01290
   18 WRITE (6,210)                                                     LPG01300
210   FORMAT (/,30H THE ONLY NON-ZERO CONSTANT IS )                     LPG01310
      GO TO 420                                                         LPG01320
   19 WRITE (6,211)                                                     LPG01330
211   FORMAT (/,35H THE NON-ZERO CONSTANTS, C(I), ARE )                 LPG01340
420   DO 422 I=1,8                                                      LPG01350
      IF(C(I).NE.0.) WRITE(6,212) I,C(I)                                LPG01360
212   FORMAT (14X,2HC(,I2,4H) = ,E10.4)                                 LPG01370
422   CONTINUE                                                          LPG01380
423   IF(K - 1)424,425,426                                              LPG01390
424   WRITE (6,1209)                                                    LPG01400
1209  FORMAT (/,36H ALL THE INITIAL CONDITIONS ARE ZERO )               LPG01410
```

```
      GO TO 20                                                          LPG01420
  425 WRITE (6,1210)                                                    LPG01430
 1210 FORMAT (/,39H THE ONLY NON-ZERO INITIAL CONDITION IS )            LPG01440
      GO TO 427                                                         LPG01450
  426 WRITE (6,1211)                                                    LPG01460
 1211 FORMAT (/,36H THE NON-ZERO INITIAL CONDITIONS ARE )               LPG01470
  427 DO 429 I=1,NN                                                     LPG01480
      IF(X(I).NE.0.) WRITE(6,1212) I,X(I)                               LPG01490
 1212 FORMAT (14X,2HX(,I2,4H) = ,E10.4)                                 LPG01500
  429 CONTINUE                                                          LPG01510
   20 READ (5,104) (JTITLE(I),IP(I),I=1,8)                              LPG01520
  104 FORMAT(8(A8,I2))                                                  LPG01530
C                                                                       LPG01540
C     CHECK FOR THE NUMBER OF COLUMNS CALLED FOR BY LOCATING FIRST      LPG01550
C     BLANK COLUMN HEADING                                              LPG01560
C                                                                       LPG01570
      DO 21 J=1,8                                                       LPG01580
      IF(JTITLE(J).EQ.IBLANK) GO TO 22                                  LPG01590
   21 CONTINUE                                                          LPG01600
      J = 9                                                             LPG01610
   22 JJ = J - 1                                                        LPG01620
C                                                                       LPG01630
C     JJ IS NOW THE NUMBER OF COLUMNS. REPEAT WITH THE GRAPHS.          LPG01640
C                                                                       LPG01650
      READ (5,105)(KTITLE(I),KTITLE(I+1),IG(I),IG(I+1),I=1,7,2)         LPG01660
  105 FORMAT (4(2A8,2I2))                                               LPG01670
      DO 24 K=1,7,2                                                     LPG01680
      IF(KTITLE(K).EQ.IBLANK.AND.KTITLE(K+1).EQ.IBLANK) GO TO 25        LPG01690
   24 CONTINUE                                                          LPG01700
      K = 8                                                             LPG01710
   25 KK = K/2                                                          LPG01720
      KKK = KK*2                                                        LPG01730
      MULTIP = 0                                                        LPG01740
      IF(KK.NE.1) GO TO 306                                             LPG01750
      IF(IG(3) + IG(4).EQ.0) GO TO 306                                  LPG01760
      IF(IG(5) + IG(6).NE.0) GO TO 303                                  LPG01770
      MULTIP = 2                                                        LPG01780
      KKK = 4                                                           LPG01790
      GO TO 306                                                         LPG01800
  303 IF(IG(7) + IG(8).NE.0) GO TO 305                                  LPG01810
      MULTIP = 3                                                        LPG01820
      KKK = 6                                                           LPG01830
      GO TO 306                                                         LPG01840
  305 MULTIP = 4                                                        LPG01850
      KKK = 8                                                           LPG01860
C                                                                       LPG01870
C     IF MULTIP = 0, KK IS THE NUMBER OF SINGLE CURVE GRAPHS. OTHERWISE LPG01880
C     MULTIP IS THE NUMBER OF CURVES ON A SINGLE GRAPH.                 LPG01890
```

```
C                                                                       LPG01900
  306 IF(JJ.EQ.0) GO TO 27                                              LPG01910
      WRITE(6,214) (JTITLE(I),IP(I),I=1,JJ)                             LPG01920
  214 FORMAT (///,56H THE COLUMN HEADINGS AND THE CORRESPONDING VARIABLELPG01930
     1S ARE ,//,(10X,A8,4X,2HX(,I2,1H)))                                LPG01940
      GO TO 28                                                          LPG01950
   27 WRITE(6,215)                                                      LPG01960
  215 FORMAT (///,25H NO PRINTOUT IS REQUIRED   )                       LPG01970
   28 IF(KK.EQ.0) GO TO 308                                             LPG01980
      IF(MULTIP.NE.0) GO TO 309                                         LPG01990
      IF(KK.NE.1) GO TO 307                                             LPG02000
      WRITE (6,216) KTITLE(1),KTITLE(2),IG(1),IG(2)                     LPG02010
  216 FORMAT (///,52H THE GRAPH TITLE AND THE CORRESPONDING VARIABLES ARLPG02020
     1E ,//,10X,2A8,4X,2HX(,I2,8H) VS. X(,I2,1H))                       LPG02030
      GO TO 31                                                          LPG02040
  307 WRITE (6,217) (KTITLE(I),KTITLE(I+1),IG(I),IG(I+1),I=1,KKK,2)     LPG02050
  217 FORMAT (///,64H THE INDIVIDUAL GRAPH TITLES AND THE CORRESPONDING LPG02060
     1VARIABLES ARE ,//,(10X,2A8,4X,2HX(,I2,8H) VS. X(,I2,1H)))         LPG02070
      GO TO 31                                                          LPG02080
  308 WRITE (6,1217)                                                    LPG02090
 1217 FORMAT(///,24H NO GRAPHS ARE REQUIRED   )                         LPG02100
      GO TO 31                                                          LPG02110
  309 WRITE(6,1220)                                                     LPG02120
 1220 FORMAT (///,52H THE GRAPH TITLE AND THE CORRESPONDING VARIABLES ARLPG02130
     1E ,/)                                                             LPG02140
      WRITE(6,1221) KTITLE(1),KTITLE(2),(IG(I),IG(I+1),I=1,KKK,2)       LPG02150
 1221 FORMAT (10X,2A8,4X,2HX(,I2,8H) VS. X(,I2,1H),/, (30X,2HX(,I2,     LPG02160
     1         8H) VS. X(,I2,1H)))                                      LPG02170
C                                                                       LPG02180
C     THIS ENDS THE BOOK-KEEPING. INITIALIZE BEFORE ENTERING MAIN LOOP. LPG02190
C                                                                       LPG02200
   31 IPAGE = 0                                                         LPG02210
      T = TI                                                            LPG02220
      NCPTS = 0                                                         LPG02230
      NUMPTS = 0                                                        LPG02240
      ITITLE(8)=IBLANK                                                  LPG02250
      ITITLE(11)=IBLANK                                                 LPG02260
      ITITLE(12)=IBLANK                                                 LPG02270
      RUN(2)=ET(NRC)                                                    LPG02280
      C(11) = 20.                                                       LPG02290
      C(12) = 5.                                                        LPG02300
      C(13) = DT                                                        LPG02310
      DO 42 I=1,NN                                                      LPG02320
   42 XC(I) = X(I)                                                      LPG02330
      TC = T                                                            LPG02340
      C(10) = 2.                                                        LPG02350
      RETURN                                                            LPG02360
C                                                                       LPG02370
```

```
 2000 IF(JJ.EQ.0) GO TO 54
C
      INCPR = C(11)+0.0000001
      C(11) = 20.
      IF( MOD (NOPTS,50*INCPR).EQ.0) GO TO 46
      IF( MOD (NOPTS,10*INCPR).EQ.0) GO TO 47
      IF( MOD (NOPTS,    INCPR))54,48,54
   46 IPAGE = IPAGE + 1
      IF(NR.EQ.1) GO TO 1047
      WRITE (6,218) (ITITLE(I),I=1,6),IPAGE,ITITLE(7),(JTITLE(I),I=1,8)
      WRITE(6,219)
      GO TO 47
 1047 WRITE (6,1218)(ITITLE(I),I=1,6),IPAGE,(JTITLE(I),I=1,8)
      WRITE (6,219)
   47 WRITE(6,219)
  218 FORMAT (1H1,///,20X,6A8,10X,5HPAGE ,I1,14H OF OUTPUT FOR,A8,/////,
     1        11X,8(A8,5X))
 1218 FORMAT (1H1,///,20X,6A8,30X,5HPAGE ,I1,/////,11X,8(A8,5X))
  219 FORMAT (1H   )
   48 DO 49 I=1,NN
   49 XC(I) = X(I)
      TC = T
      C(10) = 3.
      RETURN
C
   50 DO 53 I=1,JJ
C
      PR(I) = T
      IF(IP(I).NE.0) PR(I)=XC(IP(I))
   53 CONTINUE
      WRITE (6,220)(PR(I),I=1,JJ)
  220 FORMAT (7X, 8E13.5)
   54 IF(KK.EQ.0) GO TO 62
      INCGR = C(12)+0.0000001
      C(12) = 5.
      IF( MOD (NOPTS, INCGR).NE.0) GO TO 62
      DO 57 I=1,NN
   57 XC(I) = X(I)
      TC = T
      C(10) = 4.
      RETURN
C
   58 DO 61 I=1,KKK
C
      GR(I) = T
      IF(IG(I).NE.0) GR(I)=XC(IG(I))
   61 CONTINUE
      IF (KKK .GE. 8) GO TO 1610
```

```
      KP1 = KKK + 1                                                     LPG02860
      DO 1612 I=KP1,8                                                   LPG02870
 1612 GR(I) = 0.                                                        LPG02880
 1610 NUMPTS = NUMPTS + 1                                               LPG02890
      Y1(NUMPTS) = GR(1)                                                LPG02900
      X1(NUMPTS) = GR(2)                                                LPG02910
      Y2(NUMPTS) = GR(3)                                                LPG02920
      X2(NUMPTS) = GR(4)                                                LPG02930
      Y3(NUMPTS) = GR(5)                                                LPG02940
      X3(NUMPTS) = GR(6)                                                LPG02950
      Y4(NUMPTS) = GR(7)                                                LPG02960
      X4(NUMPTS) = GR(8)                                                LPG02970
   62 NOPTS = NOPTS + 1                                                 LPG02980
      IF(NUMPTS.LT.900) GO TO 64                                        LPG02990
      WRITE (6,221)                                                     LPG03000
  221 FORMAT (//////,25H STOP AT 900 GRAPH POINTS    )                  LPG03010
      GO TO 91                                                          LPG03020
   64 IF(NOPTS.LT.450000) GO TO 66                                      LPG03030
      WRITE (6,222)                                                     LPG03040
  222 FORMAT (//////,31H STOP AT 4500 INTEGRATION STEPS   )             LPG03050
      GO TO 91                                                          LPG03060
   66 IF(IPAGE - 9)69,67,68                                             LPG03070
   67 IF( MOD (NOPTS , 50*INCPR).NE.0) GO TO 69                         LPG03080
   68 WRITE (6,223)                                                     LPG03090
  223 FORMAT (//////,27H STOP AT 9 PAGES OF OUTPUT  )                   LPG03100
      GO TO 91                                                          LPG03110
   69 DO 70 I=1,NN                                                      LPG03120
      IF(ABS(X(I)).GT.1.E+12) GO TO 71                                  LPG03130
   70 CONTINUE                                                          LPG03140
      GO TO 72                                                          LPG03150
   71 WRITE (6,224)                                                     LPG03160
  224 FORMAT (//////,76H STOP WITH THE ABSOLUTE VALUE OF A DEPENDENT VARLPG03170
     1IABLE GREATER THAN 1.0E+12.   ,/,57H INTEGRATION PROBABLY UNSTABLE.LPG03180
     2 TRY A SMALLER STEP SIZE. ,26HNO GRAPHS WILL BE PLOTTED.        )  LPG03190
      GO TO 330                                                         LPG03200
   72 DT = C(13)                                                        LPG03210
      IF(TI.GT.TF) GO TO 80                                             LPG03220
   73 IF(T.LT.TF) GO TO 75                                              LPG03230
   74 WRITE (6,225)                                                     LPG03240
  225 FORMAT (//////,26H NORMAL STOP AT FINAL TIME    )                 LPG03250
      GO TO 91                                                          LPG03260
   75 IF(T.GE.TF1) GO TO 77                                             LPG03270
   76 C(13) = DT                                                        LPG03280
      GO TO 87                                                          LPG03290
   77 IF(T.GE.TF2) GO TO 79                                             LPG03300
   78 C(13) = DT2                                                       LPG03310
      GO TO 87                                                          LPG03320
   79 C(13) = DT3                                                       LPG03330
```

```
      GO TO 87                                                          LPG03340
   80 IF(TF.GE.T) GO TO 74                                              LPG03350
      IF(TF1.LT.T) GO TO 76                                             LPG03360
      IF(TF2 - T)78,79,79                                               LPG03370
   87 C(10) = 5.                                                        LPG03380
C                                                                       LPG03390
   88 CALL RKUTTA (NN,T,X,DT,C,TC,XC,DX)                                LPG03400
C                                                                       LPG03410
      IF(C(10).EQ.6.) RETURN                                            LPG03420
   90 T = T + DT                                                        LPG03430
      GO TO 2000                                                        LPG03440
   91 IF(KK.EQ.0) GO TO 330                                             LPG03450
      IF(MULTIP.NE.0) GO TO 97                                          LPG03460
C                                                                       LPG03470
C     PRINT PLOT UP TO 4 INDIVIDUAL CURVES                              LPG03480
C                                                                       LPG03490
      NUMPTS=-NUMPTS                                                    LPG03500
      DO 310 II=1,KK                                                    LPG03510
      WRITE(6,9998)                                                     LPG03520
 9998 FORMAT(1H1)                                                       LPG03530
      ITITLE(9)=KTITLE(2*II-1)                                          LPG03540
      ITITLE(10)=KTITLE(2*II)                                           LPG03550
      GO TO (311,312,313,314),II                                        LPG03560
  311 CALL  PLOTP(X1,Y1,NUMPTS,0)                                       LPG03570
      GO TO 310                                                         LPG03580
  312 CALL  PLOTP(X2,Y2,NUMPTS,0)                                       LPG03590
      GO TO 310                                                         LPG03600
  313 CALL  PLOTP(X3,Y3,NUMPTS,0)                                       LPG03610
      GO TO 310                                                         LPG03620
  314 CALL  PLOTP(X4,Y4,NUMPTS,0)                                       LPG03630
  310 WRITE(6,9999) ITITLE                                              LPG03640
 9999 FORMAT(1H0,8X,12A8)                                               LPG03650
      GO TO 330                                                         LPG03660
C                                                                       LPG03670
C     PLOT DUMMY CURVE ALONG AXES TO SET SCALES FOR MULTIPLE PLOT       LPG03680
C                                                                       LPG03690
   97 BIGX = 0.                                                         LPG03700
      BIGY = 0.                                                         LPG03710
      SMLX = 0.                                                         LPG03720
      SMLY = 0.                                                         LPG03730
      DO 1970 I=1,NUMPTS                                                LPG03740
      XMAX =AMAX1  ( X1(I), X2(I), X3(I), X4(I))                        LPG03750
      YMAX =AMAX1  ( Y1(I), Y2(I), Y3(I), Y4(I))                        LPG03760
      XMIN =AMIN1  (X1(I), X2(I), X3(I), X4(I))                         LPG03770
      YMIN =AMIN1 ( Y1(I), Y2(I), Y3(I), Y4(I))                         LPG03780
      IF(BIGX.LT.XMAX) BIGX=XMAX                                        LPG03790
      IF(BIGY.LT.YMAX) BIGY=YMAX                                        LPG03800
      IF(SMLX.GT.XMIN) SMLX=XMIN                                        LPG03810
```

```
      IF(SMLY.GT.YMIN) SMLY=YMIN                                       LPG03820
 197C CONTINUE                                                         LPG03830
      TX(1) = 0.                                                       LPG03840
      TX(2) = 0.                                                       LPG03850
      TX(3) = 0.                                                       LPG03860
      TX(4) = SMLX                                                     LPG03870
      TX(5) = BIGX                                                     LPG03880
      TY(1) = BIGY                                                     LPG03890
      TY(2) = SMLY                                                     LPG03900
      TY(3) = 0.                                                       LPG03910
      TY(4) = 0.                                                       LPG03920
      TY(5) = 0.                                                       LPG03930
      WRITE(6,9998)                                                    LPG03940
      ITITLE(9) = KTITLE(1)                                            LPG03950
      ITITLE(10)= KTITLE(2)                                            LPG03960
      NT=-5                                                            LPG03970
      CALL FLCTP(TX,TY,NT,1)                                           LPG03980
      MODCUR  = 2                                                      LPG03990
      DO 410 II=1,MULTIP                                               LPG04000
      IF(II.EQ.MULTIP) MODCUR=3                                        LPG04010
      GO TO  (411,412,413,414),II                                      LPG04020
  411 CALL FLCTP(X1,Y1,NUMPTS,MODCUR)                                  LPG04030
      GO TO 410                                                        LPG04040
  412 CALL FLCTP(X2,Y2,NUMPTS,MODCUR)                                  LPG04050
      GO TO 410                                                        LPG04060
  413 CALL FLCTP(X3,Y3,NUMPTS,MODCUR)                                  LPG04070
      GO TO 410                                                        LPG04080
  414 CALL FLCTP(X4,Y4,NUMPTS,MODCUR)                                  LPG04090
  410 CONTINUE                                                         LPG04100
      WRITE(6,9999) ITITLE                                             LPG04110
C                                                                      LPG04120
  330 IF(NRC.NE.NR) GO TO 1000                                         LPG04130
      IF(NR.GT.1) GO TO 333                                            LPG04140
      WRITE(6,226)                                                     LPG04150
  226 FORMAT(//,43H THE ONE RUN CALLED FOR HAS BEEN COMPLETED. ,//)    LPG04160
      STOP                                                             LPG04170
  333 WRITE(6,227)NR                                                   LPG04180
  227 FORMAT(//,5H THE ,I1,37H RUNS CALLED FOR HAVE BEEN COMPLETED.,//) LPG04190
      STOP                                                             LPG04200
      END                                                              LPG04210


      SUBROUTINE RKUTTA(/NN/,/T/,/X/,/DT/,/C/,/TC/,/XC/,/DX/)          LPG04220
      DIMENSION X(30), C(15), XC(30), DX(30), CT(4), AK(4,30)          LPG04230
      REAL*8 AK,CT                                                     LPG04240
      INDIC = C(10) - 4.0+0.0000001                                    LPG04250
      IF(INDIC.GT.1) GO TO 3                                           LPG04260
```

```
      CT(1) =  0.0D0                                                    LPG04270
      CT(2) =  0.5D0                                                    LPG04280
      CT(3) =  0.5D0                                                    LPG04290
      CT(4) =  1.0D0                                                    LPG04300
      II=0                                                              LPG04310
    7 II=II+1                                                           LPG04320
      TC = T + CT(II)*DT                                                LPG04330
      DO 2 J=1,NN                                                       LPG04340
    2 X(J) = X(J) + CT(II)*AK(II-1, J)                                  LPG04350
      C(10) = 6.0                                                       LPG04360
      RETURN                                                            LPG04370
    3 DO 4 J=1,NN                                                       LPG04380
    4 AK(II, J) = DT*DX(J)                                              LPG04390
      IF(II.LT.4) GO TO 7                                               LPG04400
      DO 5 J=1,NN                                                       LPG04410
    5 X(J)=X(J)+(AK(1,J)+2.0*(AK(2,J)+AK(3,J))+AK(4,J))/6.0             LPG04420
      C(10) = 7.0                                                       LPG04430
      RETURN                                                            LPG04440
      END                                                               LPG04450


      SUBROUTINE PLOTP(X,Y,NN,MODCUR   )                                PLOP1630
      DIMENSION X( 1), Y( 1), RANGE(4)                                  PLOP1640
      EQUIVALENCE (RANGE(1),XMAX), (RANGE(2),XMIN),(RANGE(3),YMAX),     PLOP1650
     1          (RANGE(4),YMIN)                                         PLOP1660
      DATA JERR/0/                                                      PLOP1670
C                                                                       PLOP1680
      KKZ=1                                                             PLOP1690
      GO TO 20                                                          PLOP1700
      ENTRY DPLTP(X,Y,NN,MODCUR)                                        PLOP1710
      KKZ=2                                                             PLOP1720
   20 KN=IABS(NN)                                                       PLOP1730
      KDATA=KN*KKZ                                                      PLOP1740
      MMC=MOD(MODCUR,4)                                                 PLOP1750
      IF(MMC.GT.1.AND.JERR.GT.0) GO TO 885                              PLOP1760
      ISCT=MODCUR/4+1                                                   PLOP1770
      IF(NN.LT.0) ISCT=2                                                PLOP1780
      IF(MMC    .GT.1) GO TO 5                                          PLOP1790
C                                                                       PLOP1800
C     FIND MAX & MIN FOR SCALE COMPUTATIONS                             PLOP1810
C                                                                       PLOP1820
      JERR=0                                                            PLOP1830
      XMAX=-1.E20                                                       PLOP1840
      XMIN=1.E20                                                        PLOP1850
      YMAX=-1.E20                                                       PLOP1860
      YMIN=1.E20                                                        PLOP1870
      DO 1 I=1,KDATA,KKZ                                                PLOP1880
```

```
      IF(X(I).LT.XMAX) GO TO 6                                          PLOP1890
      XMAX=X(I)                                                         PLOP1900
    6 IF(X(I).GT.XMIN) GO TO 7                                          PLOP1910
      XMIN=X(I)                                                         PLOP1920
    7 IF(Y(I).LT.YMAX) GO TO 8                                          PLOP1930
      YMAX=Y(I)                                                         PLOP1940
    8 IF(Y(I).GT.YMIN) GO TO 1                                          PLOP1950
      YMIN=Y(I)                                                         PLOP1960
    1 CONTINUE                                                          PLOP1970
C                                                                       PLOP1980
C     IF NOT AUTOSCALE GO TO CALL UTPLOT                                PLOP1990
C                                                                       PLOP2000
      IF(XMAX.EQ.XMIN) GO TO 887                                        PLOP2010
      IF(YMAX.EQ.YMIN) GO TO 889                                        PLOP2020
      IF(ISCT.EQ.1) GO TO 5                                             PLOP2030
C                                                                       PLOP2040
C     COMPUTE X-SCALE & NEW XMAX AND XMIN                               PLOP2050
C                                                                       PLOP2060
      CALL PSCALE(XMAX,XMIN,4,ISCT)                                     PLOP2070
C                                                                       PLOP2080
C     COMPUTE Y-SCALE & NEW YMAX AND YMIN                               PLOP2090
C                                                                       PLOP2100
      CALL PSCALE(YMAX,YMIN,6,ISCT)                                     PLOP2110
C                                                                       PLOP2120
C     PLOT CURVE                                                        PLOP2130
C                                                                       PLOP2140
    5 CALL UTPLOT(X,Y,KN,RANGE,KKZ,MMC)                                 PLOP2150
      IF(MMC   . EQ.1.OR.MMC   .EQ.2) RETURN                            PLOP2160
C                                                                       PLOP2170
C     PRINT SCALES WHEN LAST CURVE PLOTTED                              PLOP2180
C                                                                       PLOP2190
      XS=(XMAX-XMIN)/80.                                                PLOP2200
      YS=(YMAX-YMIN)/60.                                                PLOP2210
      WRITE(6,100) XS,YS                                                PLOP2220
  100 FORMAT(  15X,'X-SCALE:  "*"=',E10.3,' UNITS'//                    PLOP2230
     1         15X,'Y-SCALE:  "*"=',E10.3,' UNITS')                     PLOP2240
      RETURN                                                            PLOP2250
  889 WRITE(6,888)                                                      PLOP2260
  888 FORMAT(' ALL Y VALUES=0. OR THE SAME VALUE.  CANNOT SETUP PLOT GRIPLOP2270
     1D. CHECK MAX AND MIN Y FOR INITIAL CALL TO PLOTP/DPLTP')          PLOP2280
      JERR=10                                                           PLOP2290
      RETURN                                                            PLOP2300
  887 WRITE(6,886)                                                      PLOP2310
  886 FORMAT(' ALL X VALUES=0. OR THE SAME VALUE.  CANNOT SETUP PLOT GRIPLOP2320
     1D. CHECK MAX AND MIN X FOR INITIAL CALL TO PLOTP/DPLTP')          PLOP2330
      JERR=10                                                           PLOP2340
      RETURN                                                            PLOP2350
  885 WRITE(6,884)                                                      PLOP2360
```

```
884 FORMAT(' GRID NOT SETUP WHEN PLOT INITIALIZED.    NO PLOT UNTIL GRPLOP2370
   1ID PROPERLY SETUP')                                                 PLOP2380
      RETURN                                                            PLOP2390
      END                                                               PLOP2400


      SUBROUTINE PSCALE(XMAX,XMIN,IDIV,ISCT)                            PLOP2410
      DIV=IDIV                                                          PLOP2420
C     ROUND MAXIMUM TO NEXT HIGHEST 2 SIG FIGS                          PLOP2430
      IF(ISCT.LT.3) XMAX=AMAX1(0.,XMAX)                                 PLOP2440
      CALL ROUND (XMAX,IMX,FMX)                                         PLOP2450
      IF(ABS(XMX).LE.1.E-7) XMX=0.0                                     PLOP2460
      IMX=IMX-1                                                         PLOP2470
      FMX=FMX*10.                                                       PLOP2480
    3 XMX=FMX*10.**IMX                                                  PLOP2490
      IF(XMX.GE.XMAX) GO TO 2                                           PLOP2500
      FMX=FMX+1.                                                        PLOP2510
      IMM=FMX                                                           PLOP2520
      FMX=IMM                                                           PLOP2530
      GO TO 3                                                           PLOP2540
C     ROUND MINIMUM TO NEXT LOWEST 2 SIG FIGS                           PLOP2550
    2 IF(ISCT.LT.3) XMIN=AMIN1(0.,XMIN)                                 PLOP2560
      CALL ROUND (XMIN,IMN,FMN)                                         PLOP2570
      IMN=IMN-1                                                         PLOP2580
      FMN=FMN*10.                                                       PLOP2590
   14 XMN=FMN*10.**IMN                                                  PLOP2600
      IF(XMIN.GE.XMN) GO TO 11                                          PLOP2610
      FMN=FMN-1.                                                        PLOP2620
      IMM=FMN                                                           PLOP2630
      FMN=IMM                                                           PLOP2640
      GO TO 14                                                          PLOP2650
C     ROUND MAX & MIN TO 1. OR .1 IF RANGE LARGE                        PLOP2660
   11 IF(XMN.LE.1.E-7) XMN=0.                                           PLOP2670
      XSC=XMX-XMN                                                       PLOP2680
      IM=0                                                              PLOP2690
      SM=1.                                                             PLOP2700
    9 IF(XSC/DIV.LE.SM) GO TO 12                                        PLOP2710
      IF(ABS(XMN).LT.SM.AND.ABS(XMN).GT.0.) XMN=SIGN(SM,XMN)            PLOP2720
      IF(ABS(XMX).LT.SM.AND.ABS(XMX).GT.0.) XMX=SIGN(SM,XMX)            PLOP2730
   12 IF(IM.GT.0) GO TO 19                                              PLOP2740
      SM=.1                                                             PLOP2750
      IM=IM+1                                                           PLOP2760
      GO TO 9                                                           PLOP2770
C     ROUND RANGE (MAX-MIN) TO 2 SIG FIGS                               PLOP2780
   19 XSC=XMX-XMN                                                       PLOP2790
      CALL ROUND (XSC,IS10,FACTX)                                       PLOP2800
C     FIND FACTOR WHICH IS MULTIPLE OF IDIV                             PLOP2810
```

```
      FACTX=FACTX*10.                                                   PLOP2820
      OFAC=FACTX                                                        PLOP2830
      IS10=IS10-1                                                       PLOP2840
      IFX=FACTX                                                         PLOP2850
      FACTX=IFX                                                         PLOP2860
   20 IF(MOD(IFX,IDIV).EQ.0.AND.FACTX.GE.OFAC) GO TO 10                 PLOP2870
      IFX=IFX+1                                                         PLOP2880
      FACTX=IFX                                                         PLOP2890
      GO TO 20                                                          PLOP2900
   10 IF(IDIV.GT.4) GO TO 15                                            PLOP2910
C     IF X SCALE BETWEEN 8. AND 10., ROUND TO 10.                       PLOP2920
      FFX=ABS(FACTX/10.)                                                PLOP2930
      IF(FFX.GT.8..AND.FFX.LT.10.) FFX=10.                              PLOP2940
      IF(FACT X.LT.0.) FXX=-10.                                         PLOP2950
      FACTX=FFX*10.                                                     PLOP2960
   15 XSC=FACTX*10.**IS10                                               PLOP2970
C     COMPUTE NEW MAX & MIN FROM ROUNDED SCALE                          PLOP2980
      IF(XM N*XM X.NE.0.) GO TO 4                                       PLOP2990
      IF(XM N.LT.0.) XMIN=-XSC                                          PLOP3000
      IF(XM X.GT.0.) XMAX=XSC                                           PLOP3010
      RETURN                                                            PLOP3020
    4 XMAX=XSC+XMN                                                      PLOP3030
      XMIN=XMN                                                          PLOP3040
      RETURN                                                            PLOP3050
      END                                                               PLOP3060


      SUBROUTINE ROUND(ANUM,IS,FACT)                                    PLOP3070
C                                                                       PLOP3080
C     EXPRESS ANUM IN SCIENTIFIC NOTATION WHERE                         PLOP3090
C           ANUM=FACT*10.**IS WHERE FACT IS BETWEEN 1. AND 9.9          PLOP3100
C                                                                       PLOP3110
      IF(ANUM.EQ.0.) GO TO 15                                           PLOP3120
      BNUM=ANUM                                                         PLOP3130
      IF(BNUM.LT.0.) BNUM=-BNUM                                         PLOP3140
      IS=ALOG(BNUM)*.43429448                                           PLOP3150
      FACT=BNUM/10.**IS                                                 PLOP3160
C     FIND POWER OF 10                                                  PLOP3170
      IDD=-3                                                            PLOP3180
      R2=0                                                              PLOP3190
      DO 10 II=1,5                                                      PLOP3200
      IDD=IDD+1                                                         PLOP3210
      R1=R2                                                             PLOP3220
      R2=10.**(IDD+1)                                                   PLOP3230
      IF(FACT.GE.R1.AND.FACT.LT.R2) GO TO 8                             PLOP3240
   10 CONTINUE                                                          PLOP3250
    8 FACT=FACT*10.**(-IDD)                                             PLOP3260
```

```
      IS=IS+ICD                                                         PLOP3270
C     ROUND MANTISSA TO 2 SIG FIGS                                      PLOP3280
      IFAC=FACT*10.+.05                                                 PLOP3290
      FACT=IFAC                                                         PLOP3300
      FACT=FACT/10.                                                     PLOP3310
      IF(FACT.LT.10.) GO TO 20                                          PLOP3320
C     SET TO 1 IF LESS THAN 10.                                         PLOP3330
      FACT=1.                                                           PLOP3340
      IS=IS+1                                                           PLOP3350
C     IF INPUT NEGATIVE, SET MANTISSA NEGATIVE                          PLOP3360
   20 IF(ANUM.LT.0.) FACT=-FACT                                         PLOP3370
      RETURN                                                            PLOP3380
C     SET TO 0. IF 0.                                                   PLOP3390
   15 FACT=0.                                                           PLOP3400
      IS=0                                                              PLOP3410
      RETURN                                                            PLOP3420
      END                                                               PLOP3430
```

```
     CAMPIER,CR  LPG SIMULATION
1
05
   C.0     .00001    .075
 140.0     24.0       3200.    0.040       .180     0.5      .001       .001

TIME  00   XI    06   XS   07    PS   08   PI   02   VS   09   FL   04   V6  05
XI VS TIME     0600  V6 VS TIME    0500  XS VS TIME    0700  PS VS TIME    0800
```

```
C                                                                      INT10020
C     SUBROUTINE DAMP1
C                                                                      INT10050
C     DESCRIPTION OF PARAMETERS AND DATA CARDS                         INT10660
C                                                                      INT10670
C                                                                      INT10680
C        EQUATION STATEMENTS                                           INT10690
C                                                                      INT10700
C        THE NUMBER OF XDOT EQUATIONS IS N, WHICH MUST NOT EXCEED 20.  INT10710
C        THESE EQUATIONS ARE SUPPLIED BY THE USER, WHO DEFINES EACH    INT10720
C        'XDOT' IN TERMS OF THE DEPENDENT VARIABLES X(1) THROUGH X(N)  INT10730
C        AND THE INDEPENDENT VARIABLE 'T'. IN WRITING THESE EQUATIONS  INT10740
C        THE USER MAY INTRODUCE AT HIS CONVENIENCE:                    INT10750
C                                                                      INT10760
C        A)  ANY UNSUBSCRIPTED VARIABLES                               INT10770
C                                                                      INT10780
C        B)  THE SUBSCRIPTED VARIABLES X(I),  (N.LT.I.LE.20).          INT10790
C                                                                      INT10800
C        C)  THE CONSTANTS C(I), (I .LE. 8, TO BE ENTERED AS DATA)     INT10810
C                                                                      INT10820
C        D)  ANY NORMAL FORTRAN TECHNIQUE OR FUNCTION                  INT10830
C                                                                      INT10840
C        E)  ROUTINES FROM ANY SOURCE LIBRARY OR USER-SUPPLIED         INT10850
C            SUBROUTINES                                               INT10860
C                                                                      INT10870
C        NOTE 1:   THE USE OF 'AUXILIARY' X(I) (WHEN I .GT. NUMBER OF  INT10880
C              EQUATIONS TO BE INTEGRATED) DOES NOT ALTER THE VALUE OF INT10890
C              N, THE ORDER OF THE EQUATIONS.                          INT10900
C                                                                      INT10910
C        NOTE 2:   LOOPS, EITHER WITH OR WITHOUT A DO STATEMENT ARE    INT10920
C              BEST AVOIDED.                                           INT10930
C                                                                      INT10940
C        NOTE 3:   'IF' STATEMENTS, PROVIDED THAT THEY DO NOT CREATE A INT10950
C              LOOP, CAN BE USED TO TRANSFER CONTROL WITHIN THE USER'S INT10960
C              EQUATIONS. FOR EXAMPLE, THE STATEMENT                   INT10970
C              IF (T .GT. 10.) C(3) = 0.                               INT10980
C              WOULD CAUSE C(3) TO TAKE THE VALUE ZERO FOR ALL T GREATERINT10990
C              THAN 10.                                                INT11000
C                                                                      INT11010
C        CONSTANTS                                                     INT11020
C                                                                      INT11030
C        THE CONSTANTS C(1) THROUGH C(8) MAY BE USED AS DESCRIBED IN   INT11040
C        THE ABOVE EXAMPLES, AND ARE READ IN FROM A DATA CARD.         INT11050
C                                                                      INT11060
C        C(10) MUST NEVER BE USED, EXCEPT AS INDICATED IN THE STANDARD INT11070
C        OS/360 DECK ABOVE. C(11) AND C(12) CONTROL THE OUTPUT. FOR    INT11080
C        EXAMPLE, IF THE STATEMENTS                                    INT11090
C                                                                      INT11100
```

```
C         C(11) = 10.                                                      INT11110
C         C(12) =  2.                                                      INT11120
C                                                                          INT11130
C         ARE ADDED TO THE USERS FORTRAN EQUATIONS, EVERY TENTH INTEGRA-   INT11140
C         TION STEP WILL BE PRINTED OUT, AND EVERY SECOND STEP WILL BE     INT11150
C         PLOTTED. (IF NOT SET BY THE USER, DEFAULT VALUES OF 20 AND 5     INT11160
C         APPLY.) C(13) CAN BE SIMILARLY USED TO MODIFY THE STEP SIZE OF   INT11170
C         THE NUMERICAL INTEGRATION WHICH IS MORE USUALLY DEFINED ON A     INT11180
C         DATA CARD.                                                       INT11190
C                                                                          INT11200
C         OUTPUT                                                           INT11210
C                                                                          INT11220
C         ONLY THE INDEPENDENT VARIABLE 'T' AND THE VARIABLES X(1)         INT11230
C         THROUGH X(30) CAN BE OUTPUT. THEREFORE, TO OUTPUT A QUANTITY     INT11240
C         WHICH IS NOT ONE OF THE 'N' DEPENDENT VARIABLES, THE USER        INT11250
C         MUST INTRODUCE AN 'AUXILIARY' VARIABLE X(I),  (N.LT.I.LE.        INT11260
C         30). FOR EXAMPLE, BY ADDING THE FORTRAN EQUATION                 INT11270
C                                                                          INT11280
C         X(27) = X(3)*X(3)                                                INT11290
C                                                                          INT11300
C         THE SQUARE OF X(3) CAN BE OUTPUT BY THE PROGRAM AS X(27).        INT11310
C         NORMALLY, A LINE OF PRINTOUT IS GENERATED AFTER EVERY 20 INTE-   INT11320
C         GRATION STEPS, AND ONE GRAPH POINT (PER CURVE) AFTER EVERY 5     INT11330
C         INTEGRATION STEPS. THESE VALUES CAN BE MODIFIED AS INDICATED ININT11340
C         THE PREVIOUS SECTION. A RUN IS TERMINATED IF THE RUN EXCEEDS     INT11350
C         450 LINES, OR IF MORE THAN 900 GRAPH POINTS ARE GENERATED (PER   INT11360
C         CURVE).                                                          INT11370
C                                                                          INT11380
C                                                                          INT11560
C         DATA CARDS                                                       INT11570
C                                                                          INT11580
C         FIRST:    USER'S JOB IDENTIFICATION LABEL IN COLUMNS 1-48.       INT11590
C         NOTE:   THIS PROVIDES THE ONLY IDENTIFICATION OF GRAPHS OUTPUT   INT11600
C                                                                          INT11620
C         SECOND:   NUMBER OF RUNS TO BE PROCESSED (.LE. 9) IN COLUMN 1.   INT11630
C         THE RUN NUMBER TOGETHER WITH THE USER'S JOB IDENTIFICATION       INT11640
C         LABEL IS PLACED ON ALL OUTPUT.                                   INT11650
C                                                                          INT11660
C         THIRD:    THE ORDER OF THE DIFFERENTIAL EQUATIONS, N (.LE. 30)   INT11670
C         IS PUNCHED WITH TWO DIGITS IN COLUMNS 1 AND 2. FOR EXAMPLE,      INT11680
C         '03'.                                                            INT11690
C                                                                          INT11700
C         FOURTH:   INITIAL AND FINAL VALUES OF THE INDEPENDENT VARIABLE   INT11710
C         (TI AND TF) AND THE STEP SIZE (DT) APPEAR IN THE ORDER           INT11720
C         TI - DT - TF AND ARE PUNCHED, WITH DECIMAL POINTS, IN COLUMNS    INT11730
C         1-10, 11-20 AND 21-30.                                           INT11740
C                                                                          INT11750
C         IT IS ALSO POSSIBLE TO PROCESS THE INTEGRATION IN EITHER TWO     INT11760
```

```
C         OR THREE SEGMENTS OF DIFFERING STEP SIZE. THE DATA, IN COLUMNS INT11770
C         OF 1C AS ABOVE, THEN TAKES THE FORM                             INT11780
C         TI - DT1 - TF1 - DT2 - TF     OR                                INT11790
C         TI - DT1 - TF1 - DT2 - TF2 - DT3 - TF                           INT11800
C                                                                         INT11810
C         FIFTH:    THE VALUES OF THE CONSTANTS C(1),C(2),...,C(8) ARE    INT11820
C         PUNCHED, WITH DECIMAL POINTS, IN COLUMNS 1-10, 11-20,...,71-80.INT11830
C         TEN BLANK COLUMNS MAY BE USED FOR A ZERO OR UNUSED CONSTANT.    INT11840
C         IF NO CONSTANTS ARE USED, THIS ENTIRE CARD WILL BE BLANK ---    INT11850
C         BUT IT MUST NOT BE OMITTED.                                     INT11860
C                                                                         INT11870
C         SIXTH:    THE INITIAL VALUES OF X(1),X(2),...,X(N) ARE PUNCHED,INT11880
C         WITH DECIMAL POINTS, IN COLUMNS 1-10,11-20,... ADDITIONAL       INT11890
C         CARD(S) ARE REQUIRED IF N.GT.8. TEN BLANK COLUMNS MAY BE USED  INT11900
C         FOR A ZERO INITIAL VALUE.                                       INT11910
C                                                                         INT11920
C         NEXT TO LAST:       THIS CARD CONTROLS THE CHOICE OF VARIABLES  INT11930
C         FOR PRINTOUT. EACH GROUP OF 10 COLUMNS, 1-10,11-20,... UP TO    INT11940
C         71-80 MAY BE USED TO SPECIFY A COLUMN HEADING (8 CHARACTERS)    INT11950
C         AND A 2 DIGIT, RIGHT JUSTIFIED SUBSCRIPT IDENTIFYING THE COR-   INT11960
C         RESPONDING VARIABLE.                                            INT11970
C                                                                         INT11980
C         FOR EXAMPLE:                                                    INT11990
C                                                                         INT12000
C           TIME  00    AND                                               INT12010
C           SPEED 03    IN COLUMNS 1-10 AND 11-20 WOULD CAUSE THE INDEP-  INT12020
C         ENDENT VARIABLE 'T' (REPRESENTED ON THIS CARD BY THE SUBSCRIPT  INT12030
C         00) TO BE PRINTED OUT UNDER THE COLUMN HEADING 'TIME', AND THE  INT12040
C         VARIABLE X(3) TO BE PRINTED OUT UNDER THE HEADING 'SPEED'.      INT12050
C         NO MORE THAN 8 VARIABLES CAN BE OUTPUT DURING ONE RUN. IF NO    INT12060
C         PRINTOUT IS DESIRED, THIS CARD MUST BE BLANK.                   INT12070
C                                                                         INT12080
C         LAST:     THIS CARD CONTROLS THE CHOICE OF VARIABLES FOR GRAPH  INT12090
C         OUTPUT. UP TO 4 CURVES CAN BE PLOTTED, EITHER ALL ON SEPARATE   INT12100
C         GRAPHS OR ALL ON A SINGLE GRAPH. EACH GROUP OF 20 COLUMNS 1-20,INT12110
C         21-40,... SPECIFIES A CURVE. FOR EXAMPLE:                       INT12120
C          SPEED VS. TIME 0300 (IN COLUMNS 1-20) WOULD CAUSE X(3) TO BE   INT12130
C         PLOTTED VERTICALLY AGAINST THE INDEPENDENT VARIABLE 'T' HORIZ-  INT12140
C         ONTALLY. (AGAIN THE SUBSCRIPT 00 REPRESENTS THE INDEPENDENT     INT12150
C         VARIABLE.) THE GRAPH OUTPUT WOULD BE LABELED 'SPEED VS.TIME'.   INT12160
C                                                                         INT12170
C         NOTE THAT THE FIRST 16 COLUMNS OF EACH GROUP ARE USED FOR THE   INT12180
C         LABEL, THE 17TH AND 18TH FOR THE 'Y' ORDINATE, AND THE LAST     INT12190
C         TWO FOR THE X ORDINATE.                                         INT12200
C                                                                         INT12210
C         IF THE CURVES ARE TO BE ON SEPARATE GRAPHS, EACH GRAPH MUST     INT12220
C         HAVE A LABEL. IF THE CURVES ARE TO BE ALL ON ONE GRAPH, ONLY    INT12230
C         THE FIRST LABEL MUST BE PROVIDED (IN COLUMNS 1-16). THE OTHER   INT12240
```

```
C     LABELS MUST BE BLANK. THE ENTIRE CARD MUST BE BLANK IF NO GRAPH
C     OUTPUT IS REQUIRED.
C
C     MULTIPLE RUNS
C
C     IF SEVERAL SOLUTIONS OF THE SAME EQUATIONS (WITH DIFFERENT
C     CONSTANTS OR INITIAL CONDITIONS) ARE REQUIRED, THE NUMBER OF
C     RUNS IS SPECIFIED ON THE SECOND DATA CARD. FOR ALL RUNS AFTER
C     THE FIRST, ONLY THE FOURTH THROUGH THE LAST DATA CARDS ARE TO
C     BE SUPPLIED. EXCEPT FOR THE DATA ON THE FIRST THREE CARDS, AND
C     THE EQUATIONS THEMSELVES, NO INFORMATION IS RETAINED BETWEEN
C     RUNS.
C
C
C     SUBROUTINE PLOTP/CPLTP
C
C     CALLING SEQUENCE
C
C        CALL PLOTP(X,Y,N,MOCCUR)
C
C        CALL CPLTP(X,Y,N,MOCCUR)
C
C     DESCRIPTION OF ARGUMENTS
C
C     X      VECTOR OF ABSCISSAE (REAL*4 WHEN CALLING PLOTP AND REAL*8
C            WHEN CALLING CPLTP)
C
C     Y      VECTOR OF ASSOCIATED ORDINATES (REAL*4 WHEN CALLING PLOTP
C            AND REAL*8 WHEN CALLING CPLTP)
C
C     N      NUMBER OF (X,Y) PAIRS
C
C     MOCCUR CONTROLS THE NUMBER OF CURVES ON ONE GRAPH.  SEE BELOW
C            FOR EXPLANATION OF MULTIPLE VALUES OF MOCCUR.
C
C            =0, 4, OR  8   THERE IS ONLY ONE CURVE ON THIS GRAPH
C            =1, 5, OR  9   THIS IS THE FIRST OF TWO OR MORE CURVES ON
C                           THIS GRAPH.
C            =2, 6, OR 10   THIS IS AN INTERMEDIATE CURVE ON THIS GRAPH.
C            =3, 7, OR 11   THIS IS THE LAST CURVE ON THIS GRAPH.
C
C     SCALING
C
C        SCALING IS PERFORMED ONLY ON THE FIRST SET OF POINTS (WHEN
C     MOCCUR IS 0, 1, 4, 5, 8 OR 9.) FOR A MULTIPLE CURVE GRAPH,
C     THE USER SHOULD CALL THE SUBROUTINE INITIALLY WITH THE LARGEST
C     CURVE TO INSURE OPTIMUM SCALING.
```

```
C         PLOTP/DPLTP USES ONE OF THREE OPTIONAL METHODS FOR              PLOP0440
C      DETERMINING THE APPROPRIATE SCALE FACTORS.                        PLOP0450
C                                                                        PLOP0460
C    METHOD 1:  PLOTP SCANS THE X AND Y ARRAYS TO DETERMINE THE MAXI-    PLOP0470
C               MUM AND MINIMUM VALUES.  SCALE FACTORS ARE CHOSEN SO     PLOP0480
C               THAT MAXIMUM X AND MINIMUM X ARE SLIGHTLY WITHIN THE     PLOP0490
C               LEFT AND RIGHT, RESPECTIVELY OF THE GRAPH BOUNDARIES     PLOP0500
C               AND, SIMILARLY, THAT MAX Y AND MIN Y ARE SLIGHTLY        PLOP0510
C               BELOW AND ABOVE, RESPECTIVELY, THE GRAPH BOUNDARIES.     PLOP0520
C               UTPLOT IS THEN CALLED TO PLOT THE GRAPH.                 PLOP0530
C                                                                        PLOP0540
C    METHOD 1 IS USED WHEN MODCUR = 0, 1, 2, OR  3                       PLOP0550
C                                                                        PLOP0560
C    METHOD 2:  PLOTP DETERMINES THE MAXIMUM AND MINIMUM VALUES OF X     PLOP0570
C               AND Y AS IN METHOD 1. IF THE MINIMUM IS GREATER THAN     PLOP0580
C               ZERO, IT IS SET TO ZERO, AND IF THE MAXIMUM IS LESS      PLOP0590
C               THAN ZERO IT IS SET TO ZERO. THE MAXIMUM IS THEN         PLOP0600
C               ROUNDED UP TO THE NEXT HIGHEST VALUE CONTAINING 2        PLOP0610
C               SIGNIFICANT FIGUES AND THE MINIMUM IS ROUNDED DOWN TO    PLOP0620
C               THE NEXT LOWEST VALUE CONTAINING 2 SIGNIFICANT FIGURES.PLOP0630
C               A RANGE IS THEN COMPUTED WHICH IS THE DIFFERENCE BE-     PLOP0640
C               TWEEN THE MAXIMUM AND THE MINIMUM. THE RANGE IS AD-      PLOP0650
C               JUSTED UNTIL, IN THE X-DIRECTION, IT IS A MULTIPLE OF    PLOP0660
C               4, AND IN THE Y-DIRECTION, IT IS A MULTIPLE OF 6.        PLOP0670
C               ROUNDED MAXIMUM AND MINIMUMS ARE THEN USED TO CALL       PLOP0680
C               SUBROUTINE UTPLOT WHICH PLOTS THE GRAPH.                 PLOP0690
C               THIS ALGORITHM IS USED TO "RATIONALIZE" THE AXES AND     PLOP0700
C               SCALE FACTORS.                                           PLOP0710
C                                                                        PLOP0720
C    METHOD 2 IS USED WHEN MODCUR = 4, 5, 6, OR 7                        PLOP0730
C                                                                        PLOP0740
C    METHOD 3:  THIS IS THE SAME AS METHOD 2 EXCEPT WHEN ALL VALUES      PLOP0750
C               OF X OR Y ARE POSITIVE OR NEGATIVE (NO ZERO), THE        PLOP0760
C               ORIGIN OF X OR Y (RESPECTIVELY) DOES NOT APPEAR ON THE PLOP0770
C               GRAPH.                                                   PLOP0780
C                                                                        PLOP0790
C    METHOD 3 IS USED WHEN MODCUR = 8, 9, 10, OR 11                      PLOP0800
C                                                                        PLOP0810
C                                                                        PLOP0820
C         IN EITHER METHOD, EACH PRINT POSITION IN THE X-DIRECTION       PLOP0830
C      WILL BE EQUAL TO (XMAX-XMIN)/80. AND EACH PRINT POSITION IN THEPLOP0840
C      Y-DIRECTION WILL BE EQUAL TO (YMAX-YMIN)/60.                      PLOP0850
C                                                                        PLOP0860
C    GRID LABELLING                                                      PLOP0870
C                                                                        PLOP0880
C         THE DATA TO BE GRAPHED WILL BE FIT INTO AN 80 COLUMN BY        PLOP0890
C      BY 6C ROW GRID. THE GRID WILL BE LABELLED AS FOLLOWS:             PLOP0900
C                                                                        PLOP0910
```

```
C        IN THE X DIRECTION (COLUMN-WISE), THERE WILL BE 5 VALUES:         PLOP0920
C        THE MAXIMUM, THE MINIMUM, AND 3 INTERMEDIATE EQUALLY SPACED       PLOP0930
C        VALUES.                                                           PLOP0940
C                                                                          PLOP0950
C        IN THE Y DIRECTION (ROW-WISE) THERE WILL BE 7 VALUES: THE MAXI-PLOP0960
C        MUM, THE MINIMUM, AND 5 INTERMEDIATE EQUALLY SPACED VALUES.      PLOP0970
C                                                                          PLOP0980
C        IF THE LABELS HAVE A VALUE BETWEEN 1. AND 10**8, THEY WILL BE     PLOP0990
C        PRINTED IN AN F11.2 FORMAT, OTHERWISE THEY WILL BE PRINTED IN APLOP1000
C        1PE10.3 FORMAT.                                                   PLOP1010
C                                                                          PLOP1020
C     PLOTTING                                                             PLOP1030
C                                                                          PLOP1040
C            FOUR CHARACTERS ARE USED FOR PLOTTING CURVES, ".","+","*",    PLOP1050
C        AND "X". WHEN MORE THAN ONE CURVE APPEARS ON A GRAPH, EACH IS     PLOP1060
C        USED IN TURN IN THE ABOVE ORDER.  IF MORE THAN 4 CURVES ARE       PLOP1070
C        PLOTTED THE ABOVE CYCLE OF CHARACTERS IS REPEATED.  IF A NEW      PLOP1080
C        CURVE IS TO BE PLACED IN THE PLOTTING GRID WHERE AN OLD CURVE     PLOP1090
C        EXISTS, THE NEW CURVE CHARACTER REPLACES THE OLD ONE.  THUS, IFPLOP1100
C        3 IDENTICAL CURVES ARE PLOTTED, THEY WILL APPEAR AS ONE CURVE     PLOP1110
C        COMPOSED OF "*"'S.                                                PLOP1120
```

LIST OF REFERENCES

1. Gibson, E.J., *Experimental Investigation of the Fluid Dynamic Characteristics of Liquid Propellant Gun Loading Systems* ,M.S. Thesis, Naval postgraduate School, 1975.

2. Naval Postgraduate School Report NPS-59Nn75061A *Dynamics of Liquid Propellant Gun Loading Systems, Annual Progress Report* , by R.H. Nunn and E.J. Gibson, 1 June, 1975.

3. Goodson, R.E. and Leonard, R.G., " A Survey of Modeling Techniques for Fluid Line Transients", *Journal of Basic Engineering*, v. 94, p. 474 to 482, June, 1972.

4. Leonard, R. G., *A Simplified Model for a Fluid Transmission Line*, Ph.D. Thesis, Pennsylvania State University, 1970

5. Karam, J. T. and Tindall, R. R., " A Refinement of a Simplified Model for Transients in Fluid Lines", *Journal of Fluids Engineering*, v. 97, p. 376 to 377, Sept, 1975.

6. Martens, H. R. and Allen, D. R., *Introduction to Systems Theory*, p. 71, Charles E. Merrill Publishing Co., 1969.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
2.	Department Chairman, Code 59 Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	2
3.	Associate Professor Robert H. Nunn, Code 59Nn Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Associate Professor Thomas M. Houlihan, Code 59Hm Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
5.	Curricular Officer Weapons Engineering Programs Naval Postgraduate School Monterey, California 93940	1
6.	Lt. Craig R. Campier Class 53,SWOSCOLCOM Building 446,NETC Newport, Rhode Island 02840	1