BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D.C. 20024

B70 07016

SUBJECT: Computer Program for Launch Vehicle Sizing and Sensitivities
Case 105-4

DATE: July 7, 1970

FROM: A. E. Marks

ABSTRACT

A computer program was written to evaluate various launch vehicle configurations, payload variations, missions, and sizing criteria. The program analyzes two stage vehicles, considering the variation in specific impulse (I_s) with altitude, and the effects of vehicle size on mass fraction. Given the total impulsive velocity (ΔV) requirement from ground to orbit, the program will:

a) produce parametric sizing data over a range of ΔV splits between the first and second stage,

b) define the ΔV split required to produce minimum gross weight on the pad,

c) define the ΔV split required to produce equal size stages,

d) define the payload capability for given stage sizes, and

e) define performance sensitivities to various design parameters.

One and two stage vehicles, either recoverable or expendable, can be analyzed with the option to place expendable drop tanks on any configuration. The stages can be either conventional expendable stages, lifting bodies, or ballistic bodies. Provisions are made for different up and down payloads if desired. This memo contains first level documentation of the program computational techniques and its input/output characteristics.

N79-72185

(NASA-CR-113376) COMPUTER PROGRAM FOR LAUNCH VEHICLE SIZING AND SENSITIVITIES (Bellcomm, Inc.) 40 p

Unclas
00/15 12797

BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D. C. 20024

B70 07016

SUBJECT: Computer Program for Launch Vehicle Sizing and Sensitivities Case 105-4

DATE: July 7, 1970

FROM: A. E. Marks

MEMORANDUM FOR FILE

1.0 INTRODUCTION

To aid in launch vehicle analyses, a computer program was written to evaluate various configurations, payload variations, missions, and sizing criteria. In addition, the sensitivity of the payload to various parametric changes can be calculated. The program is general in nature, with the capability of sizing two stage vehicles with multiple burns and varying specific impulses. The primary use of the program to date has been with earth launch vehicles, however, and the program is described for this application.

The program will consider the variation in I_s with altitude when applicable, and the effects of vehicle size on mass fraction. Given the total impulsive ΔV requirements, the program will:

a) produce parametric sizing data over a range of ΔV splits between the first and second stage,

b) define the ΔV split required to produce minimum gross weight,

c) define the ΔV split required to produce equal size stages,

d) define the payload capability for given stage sizes, and

e) define performance sensitivities to various design parameters.

One and two stage vehicles, either recoverable or expendable, can be analyzed, with the option to place expendable drop tanks on any configuration. Provisions are made for different up and down payloads. Figure 1 shows the representative configurations.

The terms booster and orbiter are used to indicate the first and second stages respectively. Both booster and orbiter recovery can be analyzed, however, only the orbiter can bring back discretionary payload.

The various burns of a two stage system are indicated on Figure 2. For the booster, the first burn is its portion of the ascent ΔV. The next burn is any separation propulsion or propulsion necessary for stage return to the launch site. The third burn is any propulsion required at the launch site for landing. If the booster should go to orbit, a provision is made for a de-orbit burn if it is required.

The orbiter burn starts at booster burnout and brings itself and the payload to orbit. It will then provide any on-orbit maneuvering the payload may require. If the stage is recoverable, it will then fire the engines to de-orbit itself and any down payload. If any cross range is required, necessitating propulsion, it is performed at vacuum specific impulse right after the de-orbit burn. Landing propulsion is then accomplished in the same manner as the booster.

2.0 PROGRAM EQUATIONS

The basic rocket equation is used in both the sizing and sensitivity programs. This equation is:

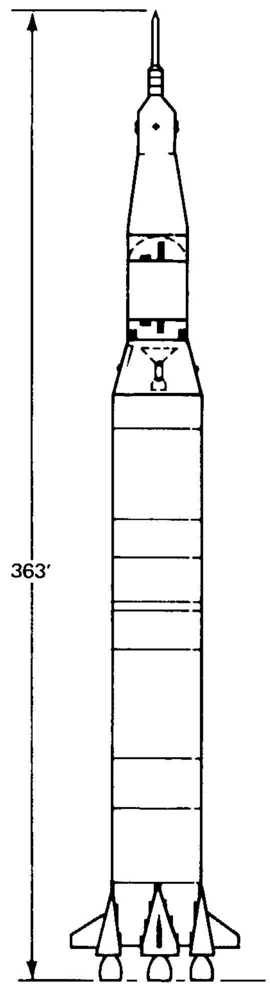
(1) $$\Delta V_i = I_{s_i} \, g \ln \frac{W_{o_i}}{W_{f_i}}, \text{ where}$$

ΔV = velocity increment during the rocket burn, fps
I_s = engine specific impulse, sec
W_o = initial gross weight before the rocket burn, lbs
W_f = final gross weight after the rocket burn, lbs

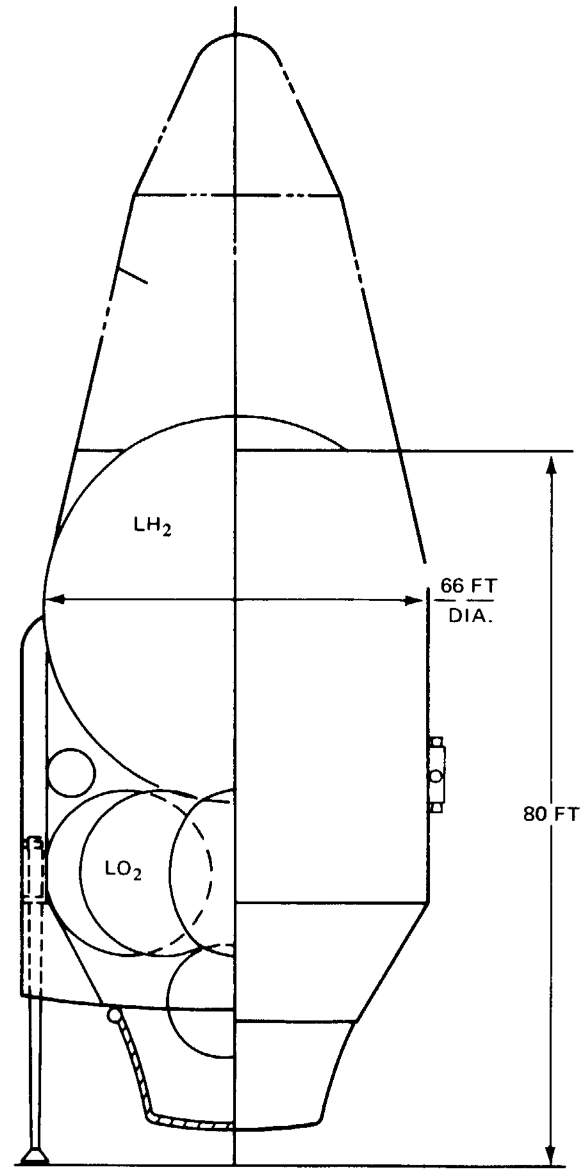
The equations for the booster are thus:

(2) $$\Delta V_1 = I_{s_1} \, g \ln \frac{GW_B + GW_O + PL_1}{W_{B_{B.O._1}} + GW_O + PL_1} = \text{Booster Ascent } \Delta V$$

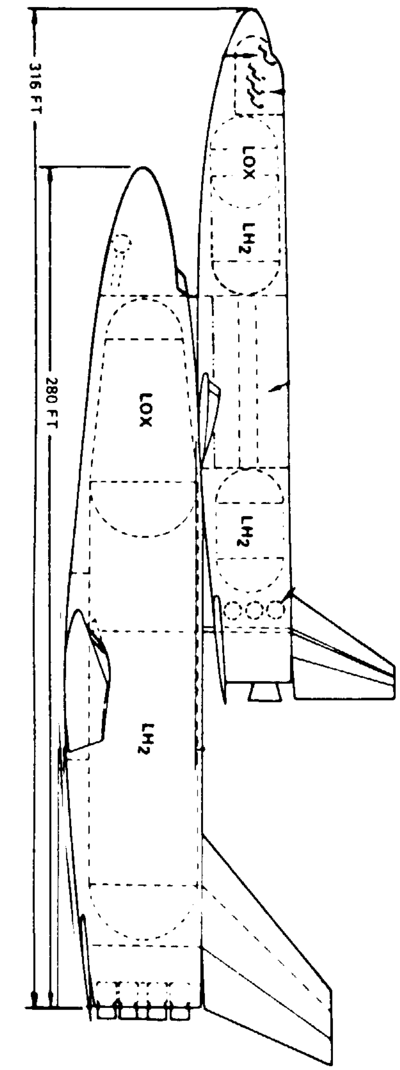
(3) $$\Delta V_2 = I_{s_2} \, g \ln \frac{W_{B_{B.O._1}}}{W_{B_{B.O._2}}} = \text{Booster Stage Separation or Recovery } \Delta V$$



SATURN V
EXPENDABLE



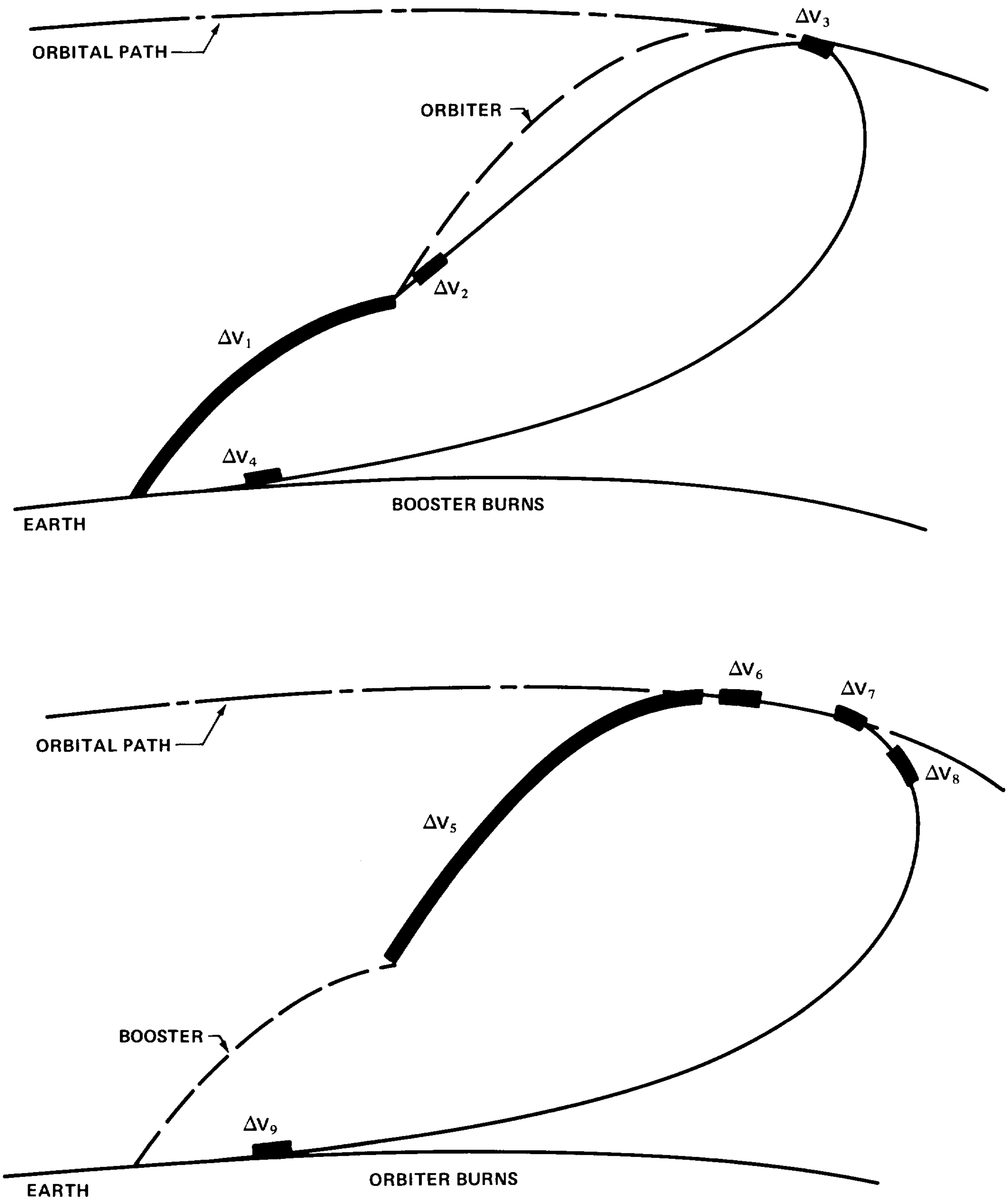
SASSTO
RECOVERABLE BALLISTIC



SPACE SHUTTLE
RECOVERABLE LIFTING BODIES

FIGURE 1 - REPRESENTATIVE CONFIGURATIONS

FIGURE 2 - ROCKET BURNS



(4) $$\Delta V_3 = I_{s_3}\, g \ln \frac{W_{B_{B.O._2}}}{W_{B_{B.O._3}}}$$ = Booster De-orbit ΔV if Booster Goes on to Orbit

(5) $$\Delta V_4 = I_{s_4}\, g \ln \frac{W_{B_{B.O._3}}}{(1-\lambda_B)GW_B}$$ = Booster Landing ΔV

For the booster, I_{s_1} is an integrated average specific impulse over the ascent burn. I_{s_2} is the specific impulse at booster burnout if sub-orbital recovery is used, or it is the average specific impulse for the burn to orbit. The de-orbit ΔV_3 is burned with vacuum specific impulse, and the landing ΔV_4 at sea level specific impulse. The last 3 ΔV's may be zero if expendable systems are considered.

The equations for the orbiter are:

(6) $$\Delta V_5 = I_{s_5}\, g \ln \frac{GW_O + PL_1}{W_{O_{B.O._1}} + PL_1}$$ = Orbiter Ascent ΔV

(7) $$\Delta V_6 = I_{s_6}\, g \ln \frac{W_{O_{B.O._1}} + PL_1}{W_{O_{B.O._2}} + PL_1}$$ = On-orbit ΔV

(8) $$\Delta V_7 = I_{s_7}\, g \ln \frac{W_{O_{B.O._2}} + PL_2}{W_{O_{B.O._3}} + PL_2}$$ = De-orbit ΔV

(9) $$\Delta V_8 = I_{s_8}\, g \ln \frac{W_{O_{B.O._3}} + PL_2}{W_{O_{B.O._4}} + PL_2}$$ = Cross Range ΔV

(10) $$\Delta V_9 = I_{s_9}\, g \ln \frac{W_{O_{B.O._4}} + PL_2}{(1-\lambda_O)GW_O + PL_2}$$ = Landing ΔV

I_{s_5} is the integrated average specific impulse of the orbiter during its ascent burn. I_{s_6}, I_{s_7}, and I_{s_8} are burned at vacuum, while I_{s_9} is at sea level.

Equations 2-5 and 6-10 are solved simultaneously for GW_B and GW_O after the ΔV's, PL's and I_s's are all input.

3.0 PROGRAM LOGIC

The better the mass fraction, the smaller the propulsion stage required to provide a given ΔV to a given payload. This is shown as line A on Figure 3, and constitutes a performance line for a particular propulsive job.

At the same time, as propulsive stages get bigger it becomes easier to design the stages with high mass fractions. This is shown as line B on Figure 2. The point where the two lines cross represents the point where the design capability satisfies the performance requirements. Any point on the design line above the performance line will meet the performance requirements, however, the lightest system corresponds to the intersection. This is the point selected by the program.

A schematic of the program logic is shown in Figure 4. The specific impulses of each stage, the ΔV's required of the vehicle, and the up and down payloads are input to the program. An ascent velocity split for the first and second stages is then selected. The average I_s for each stage during ascent is then computed using a curve fit of I_s vs. impulsive ΔV. The derivation of this curve will be discussed later. The sizing calculation is started with an initial estimate of the stage mass fraction (λ_1). The stage gross weight is then computed from the performance equations above. This stage gross weight is in turn used to compute a second mass fraction (λ_2) from a design curve for that particular type of vehicle. This curve fit will also be discussed in a later section. If λ_1 and λ_2 agree within a predetermined accuracy, the gross weight calculated is the final output. If the difference between the two λ's is greater than that allowable, a new λ, taken as a weighted average of the two old λ's, is used to compute a new gross weight. The iteration procedure is continued until the last two computed mass fractions are sufficiently close. This procedure is done for the orbiter first and then the booster.

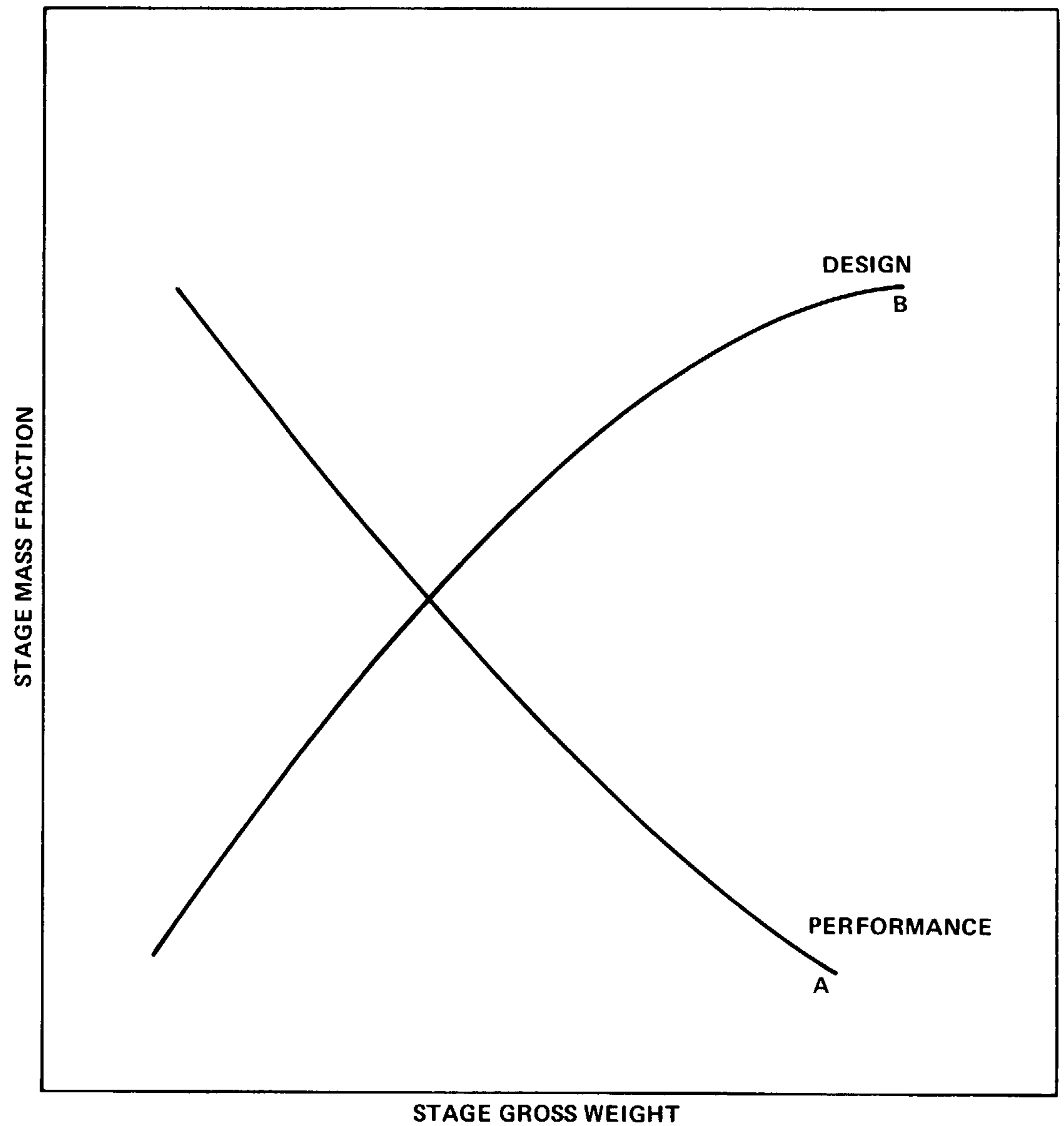


FIGURE 3 - PROGRAM SOLUTION SCHEMATIC

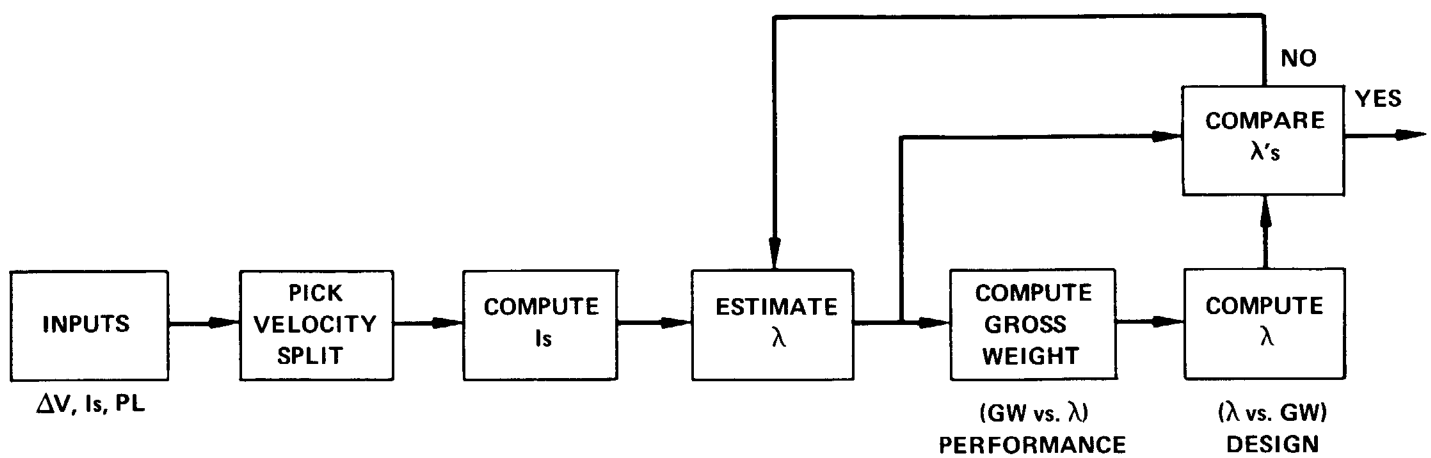


FIGURE 4 - SIZING PROGRAM LOGIC

4.0 PROGRAM FEATURES

4.1 Mass Fraction

The program uses a correlation between vehicle gross weight, W_g, and mass fraction, λ, of the form:

$$\lambda = s - \frac{T}{W_g^{1/3}} - \frac{U}{W_g} \tag{11}$$

where S, T, and U are constants. This equation form was derived on the assumption that the inert weight, W_I, of a vehicle could be described by an equation of form:

$$W_I = U + S_O W_g + T W_g^{2/3} \tag{12}$$

U represents those weights that are fixed independent of vehicle size, such as avionics and crew support systems. S_O represents those systems that are directly dependent on gross weight such as engines and tankage. T represents those systems that are dependent on vehicle surface area, such as insulation and interstage structure. Many systems obviously don't fit any of these categories, however, it was felt that these three groupings would accommodate the majority of the heavy systems. An empirical fit to parametric sizing data was the final objective. By definition, mass fraction is:

$$\lambda = \frac{W_{propellant}}{W_g} = \frac{W_g - W_I}{W_g} = 1 - \frac{W_I}{W_g} \tag{13}$$

Equations (12) and (13) lead to Equation (11) with $S = 1 - S_O$. If T and U are input as zero, the equation produces a constant λ. Empirical fits to data are shown in Figures 5 and 6. Data for these curves were taken from various studies of ballistic boosters and from the ILRV studies, (References 1 to 6).

New coefficients may be input directly as calculated from λ vs. gross weight data by calling a special subroutine. Three data points of λ vs. gross weight must be input and the coefficients of the curve fit will be computed. These coefficients are identified in the program as s(i), t(i), and u(i).

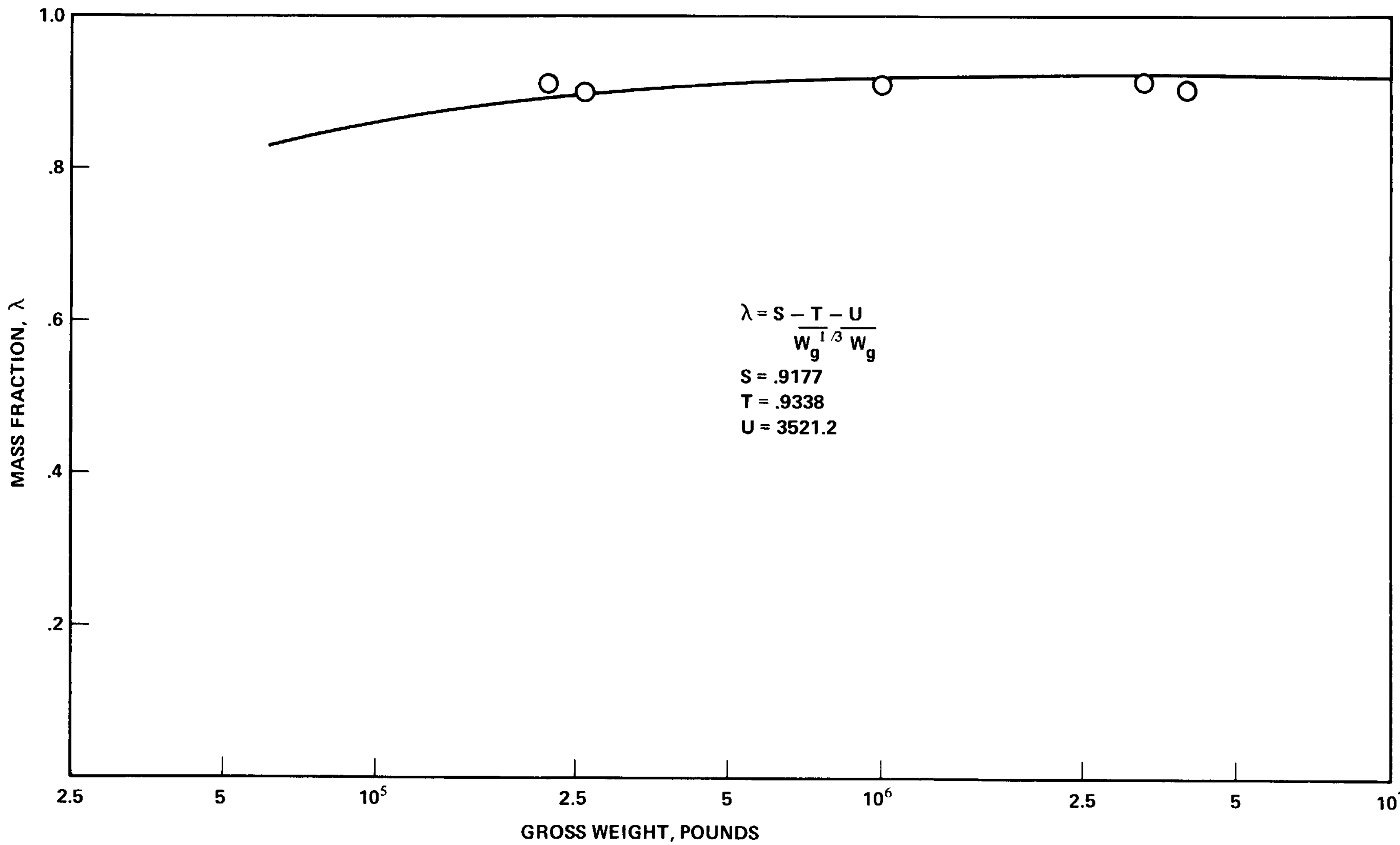


FIGURE 5 - MASS FRACTION VS. GROSS WEIGHT (NO PAYLOAD) FOR BALLISTIC BODIES

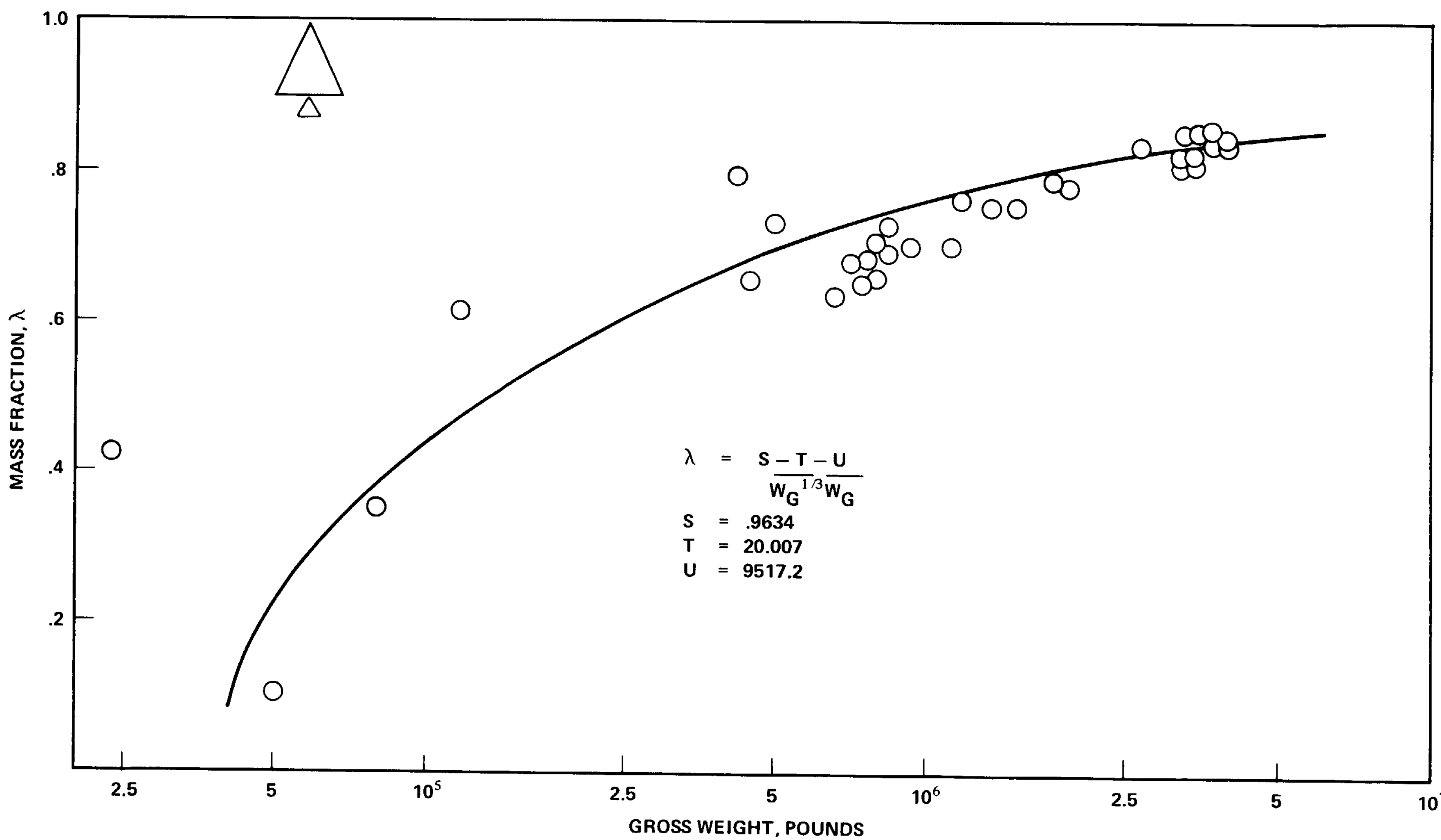


FIGURE 6 - MASS FRACTION VS. GROSS WEIGHT (NO PAYLOAD) FOR LIFTING BODIES

4.2 Specific Impulse

The program makes an allowance for the variation in specific impulse with altitude. Since the program does not actually "fly" a trajectory, but uses mission descriptors in terms of impulsive ΔV; a curve fit of specific impulse vs. impulsive ΔV is used.

Some computer runs were made with existing trajectory analysis programs to determine the specific impulse vs. impulsive ΔV (ideal ΔV + gravity losses + drag losses) for a number of different type engines. Referring back to the method of computation, the mass ratio of a stage from equation 1 is, for a given ΔV, $\mu = e^{\frac{\Delta V}{I_s g}}$. If the I_s varies during execution of this ΔV, however, some average value of I_s must be used. If the total ΔV is broken down into small segments, ΔV_i, such that I_s may be considered constant over ΔV_i, then

(14) $$\mu_i = \exp\left(\frac{\Delta V_i}{I_s g}\right) .$$

The ratio for the total ΔV is $\mu_t = \mu_1 \cdot \mu_2 \cdot \mu_3 \cdots \mu_n =$

(15) $$\exp\left\{\frac{1}{g}\left(\frac{\Delta V_1}{I_{s_1}} + \frac{\Delta V_2}{I_{s_2}} + \frac{\Delta V_3}{I_{s_3}} + \cdots \frac{\Delta V_n}{I_{s_n}}\right)\right\} \text{ or } \mu_t = \exp\left(\frac{1}{g} \sum_{i=1}^{n} \frac{\Delta V_i}{I_{s_i}}\right)$$

The average I_s can be defined by

(16) $$\mu_t = \exp\left(\frac{\Delta V}{\bar{I}_s g}\right), \text{ or } \frac{\Delta V}{\bar{I}_s g} = \frac{1}{g} \sum_{i=1}^{n} \frac{\Delta V_i}{I_{s_i}} ;$$

Thus,

(17) $$\bar{I}_s = \frac{\Delta V}{\sum_{i=1}^{n} \frac{\Delta V_i}{I_{s_i}}} .$$

The denominator is simply the area under the curve on a plot of $\frac{1}{I_s}$ vs. ΔV. Using some typical launch trajectories, plots of $\frac{1}{I_s}$ vs. ΔV were determined, and are shown in Figures 7 and 8 for HiP_c and annular engines, respectively. Also included on these figures are linearized approximations to the actual curves.

FIGURE 7 - SPECIFIC IMPULSE HiP_c ENGINE

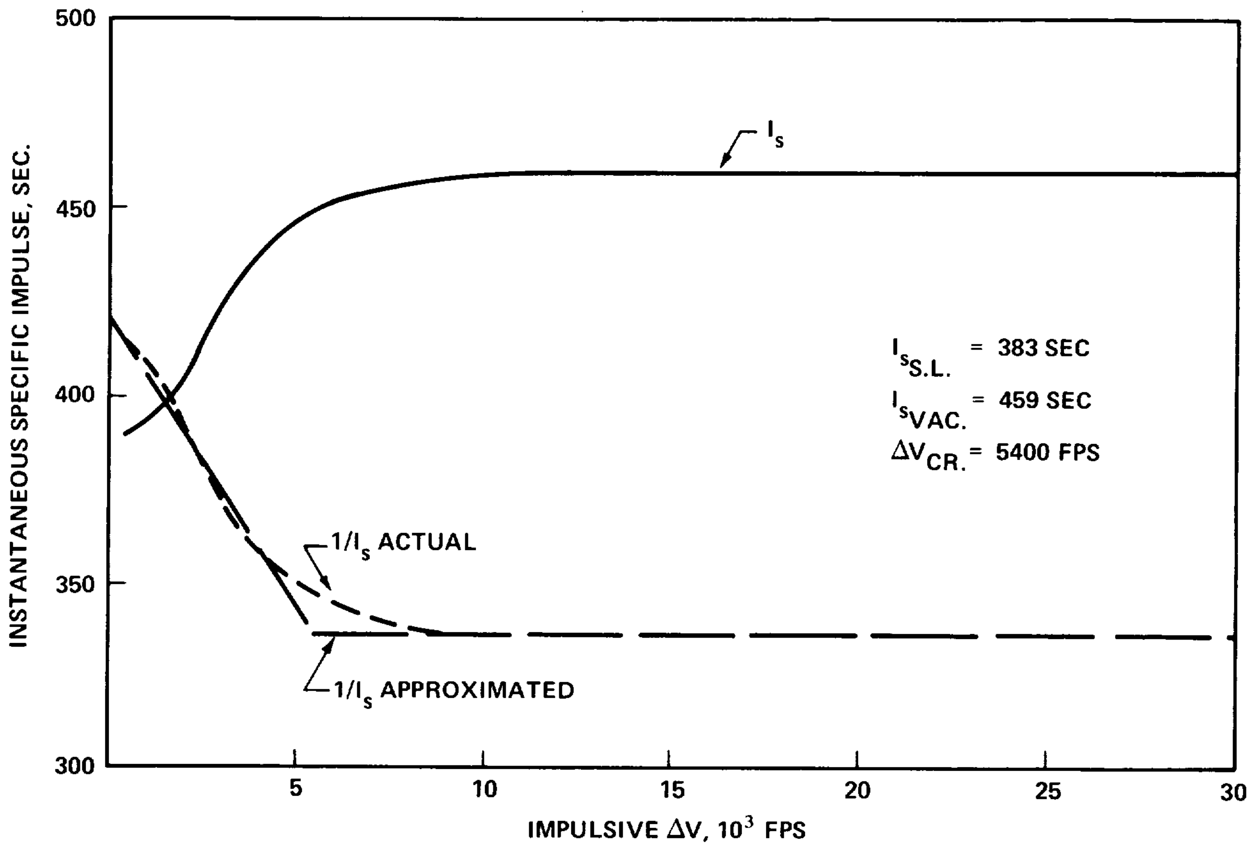
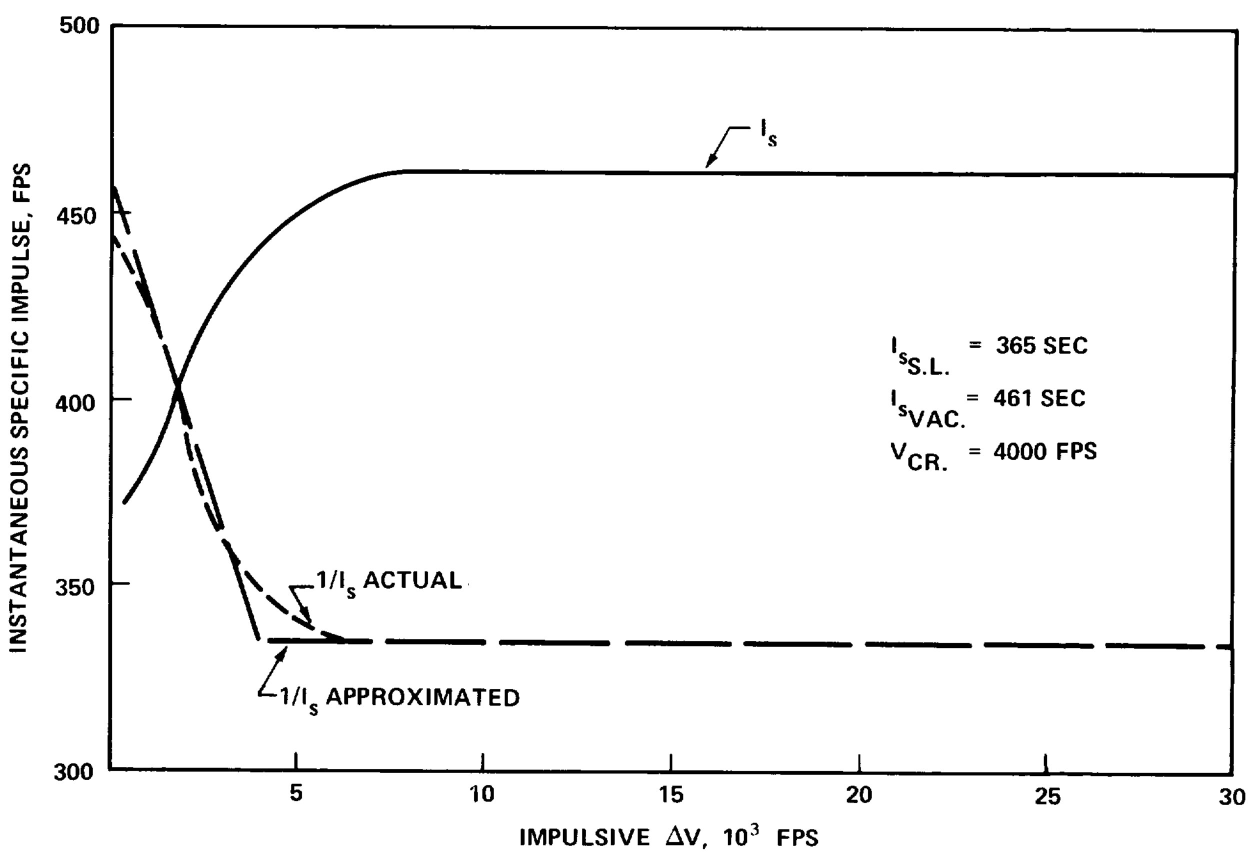


FIGURE 8 - SPECIFIC IMPULSE ANNULAR ENGINE



The linearized curves can be defined by three values: 1) the starting point corresponding to sea level I_s, 2) the ΔV at which I_s becomes essentially vacuum I_s, called the breakpoint ΔV, and 3) the vacuum I_s corresponding to the horizontal portion of the curve.

The average I_s can thus be computed for a given ΔV. A comparison between the average effective I_s taken from the trajectory, and the results of the linearized model are shown in Figure 9. Agreement is excellent.

4.3 Impulsive Return to Launch Site

Ballistic boosters do not have the flyback characteristics of lifting-bodies and must therefore use propulsion to return to the launch site. Calculations indicate that the ΔV required to accomplish this is proportional to the ΔV at booster burnout. This maneuver, sometimes called lob-retro, is analyzed and described in Reference 7. The ΔV required for lob-retro as a function of burnout ΔV is shown in Figure 10. A curve fit for the fraction of the burnout ΔV needed for impulsive return to the launch site is

(18) $$\frac{\Delta V_{\text{lob-retro}}}{\Delta V_{\text{burnout}}} = \Delta V_{\text{burnout}} \times .007\sqrt{\Delta V_{\text{burnout}}} \, .$$

This formula is included in the program. If the ballistic booster delivers much more than half the ΔV to orbit, it may be cheaper (in terms of propellant consumed) to fly on to orbit and reenter ballistically to the launch site, and not lob-retro.

The sizing program analyzes both the lob-retro and on-to-orbit situations for a ballistic booster and chooses the mode requiring the smallest mass ratio. Variations in the fraction of the burnout ΔV used for lob-retro can also be input. If zero is used, the analysis corresponds to a down range landing.

4.4 Optimizations

The capability to optimize any shuttle configuration either for minimum gross lift-off weight or for equal size stages is provided. The latter is attractive for two stage vehicles where both stages are of the same type (ballistic, lifting-body, or expendable). This could provide significant cost savings in a development program.

When optimizing for minimum gross liftoff weight, the program searches for the velocity split at which the sum of the booster and orbiter gross weight is a minimum.

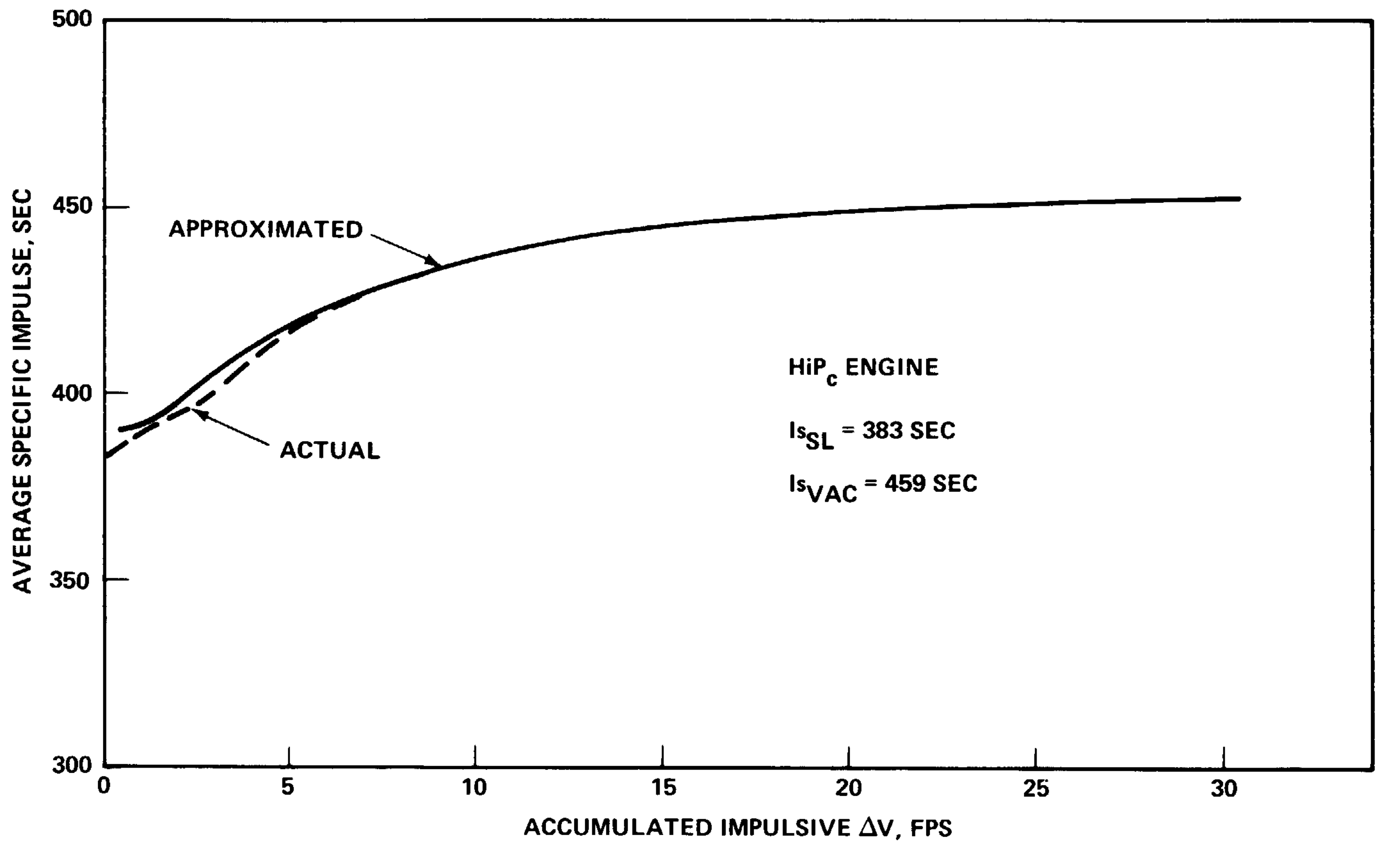


FIGURE 9 - AVERAGE SPECIFIC IMPULSE VS. ACCUMULATED ΔV

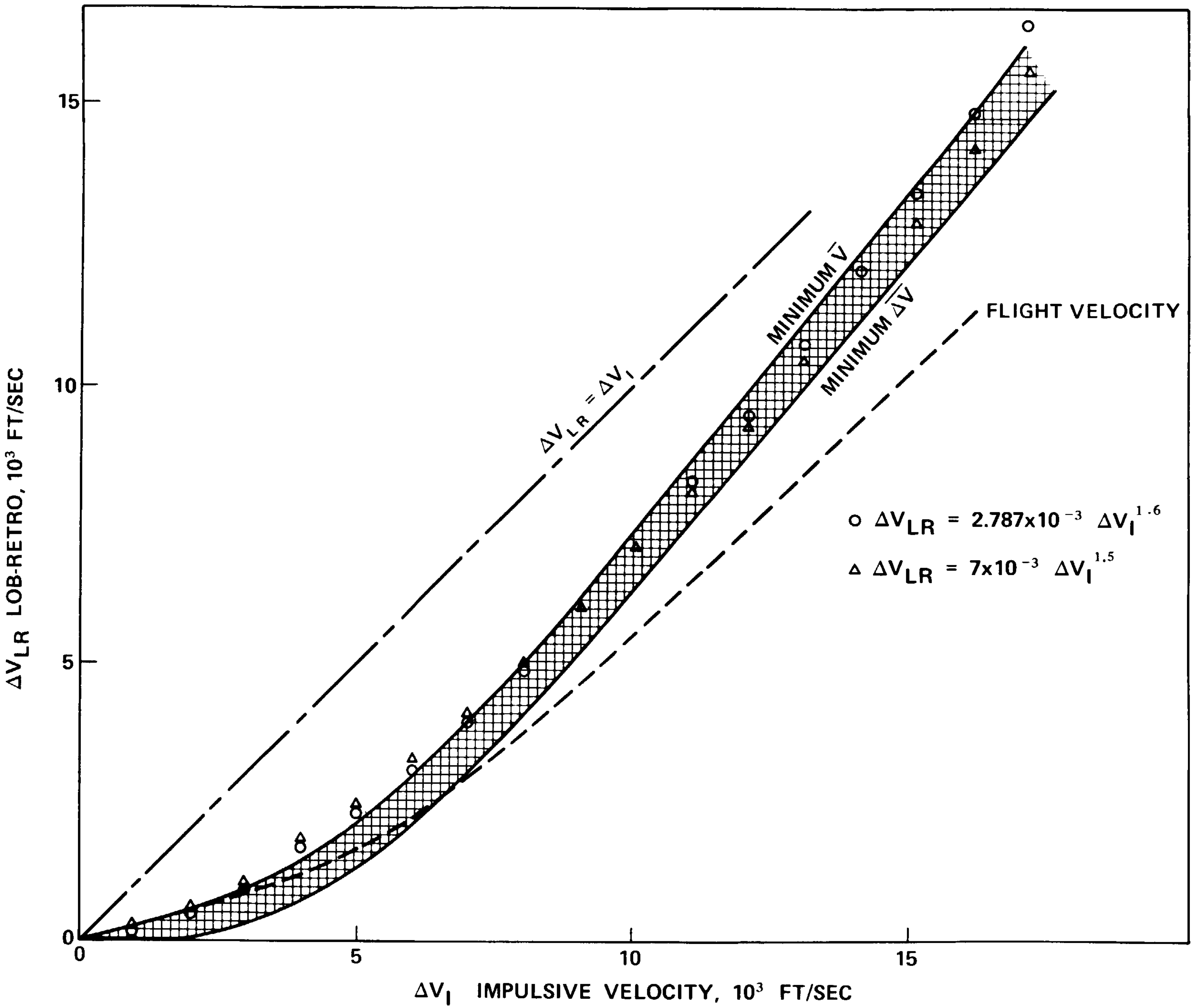


FIGURE 10 - LOB-RETRO REQUIREMENTS

Configurations can also be analyzed at single points by inputing a desired velocity split, or produce parametric data for W_g vs. ΔV splits.

4.5 Expendable Drop Tanks

Once a design has been defined using the sizing program, additional expendable propellant tanks (drop or tip-tanks) can be added for additional performance. The new desired payload and ΔV must be known, plus an estimation of the mass fraction of the tip tanks. With this information, new velocity splits are calculated for the orbiter, booster, and drop tanks and the gross weight of the drop tanks are calculated. The specific impulse of the booster engines are used during drop tank operations since the drop tanks supply propellant to these engines.

5.0 INPUTS

5.1 Input Program

Inputs are demanded by the program with variable names that are used in the calculations. When single subscripts are used, the subscript 1 refers to tip tanks, 2 refers to the booster, and 3 to the orbiter. In the case of payloads, the subscripts 1 and 3 are up payloads, and 2 and 4 are down payloads. The payload subscripts 3 and 4 are only used with tip tank calculations. The specific impulse subscripts are 1 and 3 for orbiter and booster sea level values, respectively, while 2 and 4 are orbiter and booster vacuum values.

Double subscripted variables are used for convenience in the program and are not necessarily related to any particular stage or maneuver.

5.1.1 Vehicle Type

Variable: B(i)

B(2) = booster configuration
B(3) = orbiter configuration

Use: Flag for each stage configuration

Values: 0 = expendable stage
1 = ballistic stage
2 = lifting body stage

Comments: The flag, B(i), calls a subroutine to use stored mass fraction vs. gross weight correlation coefficients corresponding to the particular configuration desired. In addition, it sets certain maneuvers that are not needed for the various configurations to zero.

5.1.2 Ideal Velocities

Variable: V(i,j)

V(1,0) = total ascent ΔV, fps
V(2,0) = on-orbit ΔV, fps
V(3,6) = booster de-orbit ΔV, fps
V(3,7) = orbiter de-orbit ΔV, fps
V(4,0) = cross-range ΔV, fps
V(5,2) = booster hover ΔV, fps
V(5,3) = orbiter hover ΔV, fps

Use: Inputs non-variant velocity data to describe type and magnitude of mission maneuvers.

Values: Total impulsive velocity increments in fps.

Comments: All the ΔV's required of the shuttle must be input. The first input is the total ΔV to orbit. This velocity must include gravity and drag losses as well as the actual flight velocity change.

The on-orbit ΔV is any velocity increment that must be added to the payload by the orbiter after it achieves orbit. This might include such maneuvers as circularization, or Hohmann transfer to a different altitude.

The de-orbit ΔV's are the velocities required for the stages reaching orbit to de-orbit for reentry. This maneuver is done for the orbiter with down payload only, and for the booster if it reaches orbit.

The DOD has missions which require significant cross-range from the de-orbiting shuttle. A lifting-body can achieve this cross-range aerodynamically, while a ballistic vehicle must accomplish this propulsively. Ths cross-range ΔV can be input into the program.

Lifting body vehicles will fly back from orbit to the landing site aerodynamically. Ballistic vehicles will reenter ballistically. The last portion of the terminal velocity will be taken out propulsively. Also, the ballistic vehicles will land vertically and some hover time must be allowed. Therefore, the total ΔV needed to cancel the terminal velocity and to hover are input to the program for a ballistic vehicle.

5.1.3 Adjustment Factors

Variable: F(j,k), S(i)

F(3,2) = inert weight factor, orbiter
F(2,2) = inert weight factor, booster
S(2) = mass fraction coefficients, expendable booster
S(3) = mass fraction coefficients, expendable orbiter

Use: To adjust the inert weight or mass fractions that are obtained from the curve fits.

Values: The F(i,j)'s are used as multiplication factors. Thus, a 10 percent increase in inert weight can be achieved by entering F(i,j)=1.1. The S(i) factors are for a constant mass fraction for expendable stages only. If curve fits are to be used, set S(i)=0.

Comments: The mass fractions of all the stages are determined from curve fits. The inert weight determined from these curves can be adjusted by multiplication factors F(3,2) and F(2,2). Thus, the orbiter and booster inert weights can be changed such that a 10 percent increase in booster or orbiter inert weight is attained by using F(j,2)=1.1. If the inert weights are to be unchanged from the curve fits, factors of 1 should be input.

For expendable stages, constant mass fractions can be input for the booster and orbiter. The inputs S(2) and S(3) refer to the booster and orbiter mass fractions, respectively. If the mass fraction curves are to be used, S(i) should be set to zero.

If a different mass fraction curve fit is desired, a sub-routine in the input program can be used. Three data points, stage weight vs. mass fraction, are needed to generate this new curve.

5.1.4 Specific Impulse

Variables: I(k), V(7,0)

I(1) = orbiter sea level I_s
I(2) = orbiter vacuum I_s
I(3) = booster sea level I_s
I(4) = booster vacuum I_s
V(7,0) = breakpoint ΔV for curve fit

Use: Inputs the vacuum and sea level I_s of both the booster and orbiter, and the impulsive velocity at which the I_s reaches vacuum performance. These values are then used to help compute average I_s values for various ΔV burns.

Values: The I_s in seconds and the ΔV in fps are input.

Comments: The specific impulse of the booster and orbiter at both sea level and vacuum can be input. A straight line curve fit of $1/I_s$ vs. ΔV is then used from sea level to the point at which the engines operate at essentially vacuum specific impulse. The ΔV, V(7,0), at which this vacuum performance occurs must be input. This generally occurs between 4,000 and 6,000 fps.

The program has the projected High P_c engine performance built in. If this is to be used for both stages, -1 should be input for i(1). A constant I_s can be obtained by inputing all the I_s values equal to the desired I_s.

5.2 Sizing Program

5.2.1 Type of Calculation

Variable: J

Use: Set flag for type of calculation to be performed.

Values: J = 1 parametric calculations
J = 0 minimum gross weight optimization
J = -1 equal stage optimization

Comments: The first input, J, indicates the type of calculation to be performed. If either a single point or parametric analysis is to be conducted, J = 1. If it is desired to have the program search for the minimum gross-lift-off-weight, J = 0. If it is desired to find the point at which equal size stages occur, J = -1.

5.2.2 Payloads

Variable: P(n)
P(1) = up payload, lbs
P(2) = down payload, lbs

Use: Inputs the up and down payload for the vehicle design.

Values: The payloads are in pounds.

Comments: Varying amounts of up and down payload can be input. The up payload, P(1), is propelled through ascent and on-orbit ΔV's. The down payload is put through de-orbit, cross range and hover ΔV. After booster burnout, all remaining booster velocities are accomplished without payload.

5.2.3 Ascent Velocity Split

Variables: X, Y, Z

When J = 1:

X = initial velocity split
Y = increment size for velocity split
Z = final velocity split

When J = o or -1:

X = starting point velocity split
Y = initial increment size (usually .1).

Use: X,Y, and Z are used to initialize and bound the desired calculations, and set the increment size when optimizing.

Values: Since X,Y, and Z are expressed in fractions, they must be from 0 to 1. X cannot be larger than Z.

Comments: Parametric calculations are controlled by three input variables labeled X,Y, and Z. Coupled with the index, J, these variables also control the iterative search for optimum sizes.

The initial velocity split, the portion of the ascent velocity put on by the booster, is input as X. for point calculations, this will be the point at which the vehicle will be sized. For runs to be optimized, this is the point at which the optimization will begin.

A series of point runs can be made using X,Y, and Z. X will be the initial point, Z will be the final point, and Y will be the increment size or step to be used going from X to Z. For a single calculation, set Z equal to X. For optimization, Z is not needed, and Y is the initial increment size. Y will be reduced at least two orders-of magnitude to get a more accurate result during the optimization procedure.

Y can be set to zero for point calculations and a limited printout will result. Only the stage weights and the gross liftoff weight will be printed out.

5.2.4 Expendable Drop Tanks

Variables: V(6,0), P(m), S(1)

V(6,0) = ascent ΔV with tip tanks, fps
P(3) = up payload, lbs
P(4) = down payload, lbs
S(1) = tip tank mass fraction

Use: These variables describe the payload and mission characteristics for the previously designed stage to use with tip tanks added.

Values: The ascent ΔV should be input in fps, the payloads in pounds, and the mass fraction as a fraction.

Comments: When the sizing program has sized a vehicle, a call for V(6,0) is made. This is the ascent ΔV to be used with tip tanks added to the vehicle. If no tip tanks are desired, set V(6,0) = 0. If V(6,0) is input, the payload up and down and the mass fraction of the tip tanks must be input.

5.3 Sensitivity Program

5.3.1 Type of Calculation

Variable: X

Use: Sets flag for type of calculation desired.

Values: X = 1 vary up payload only
X = 0 equal up and down payload

Comments: The sensitivity program analyzes the variation in payload for a given change in some parameter. This payload variation can be in two forms. For cases where there is no down payload, the variation in up payload is the only concern. When equal up and down payload is desired, the variation in both is calculated. Thus, X = 1 will vary only the ascent payload, while X = 0 will solve for equal up and down payload.

5.3.2 Variation Factor

Variable: A

Use: To set the variation increment with which to approximate partial derivatives.

Value: A = 0 define payload for given vehicle(s)

A = 1.n defines sensitivities

Comments: The sensitivity of the vehicle to a number of parameters is built into the program. The amount these parameters are varied is required as an input. In general, a one percent variation is sufficient. Thus, A = 1.01 should be used. The sensitivity of the payload to changes in propellant weight, structural weight, specific impulses, and velocities is then calculated.

The sizing program determines vehicle sizes for given payloads. If the vehicle size is known but the payload is unknown, the sensitivity program can be used to calculate this payload. A should be set to 0 in this case. In addition to vehicle size, all velocities must also be known.

5.4 Input Listing

A list of the inputs as they appear in the program is presented.

5.4.1 Input Program

B(2) - Booster vehicle type — 2 = lifting body; 1 = ballistic; 0 = expendable

B(3) - Orbiter vehicle type

V(1,0) - Total ascent velocity
V(2,0) - On-orbit ΔV
V(3,6) if B(2) = 1 - booster de-orbit ΔV
V(3,7) if B(3) = 0 - orbiter de-orbit ΔV
V(4,0) if B(3) = 1 - cross-range ΔV
V(5,0) if B(2) = 1 - booster hover ΔV
F(0,3) if B(2) = 1 - lob-retro factor
V(5,3) if B(3) = 1 - orbiter hover ΔV
S(2) if B(2) = 0 - booster mass fraction — 0 if variable
S(3) if B(3) = 0 - orbiter mass fraction
F(2,2) - booster structural weight factor
F(3,2) - orbiter structural weight factor
I(1) - sea level I_s, orbiter
I(2) - vacuum I_s, orbiter — enter I(1) = -1 for HiP$_C$ engine performance of both stages
I(3) - sea level I_s, booster

I(4) - vacuum I_s, booster
V(7,0) - breakpoint ΔV

5.4.2 Sizing Program

J - calculation type: 1 - single point; 0 - minimum gross weight; -1 - equal size stages

P(1) - up payload
P(2) - down payload
X - initial velocity split fraction
Y - velocity split increment
Z - final velocity split fraction

V(6,0) - ascent ΔV if tip tanks desired [0 if no tip tanks]
P(3) - up payload with tip tanks
P(4) - down payload with tip tanks
S(1) - mass fraction of tip tanks

5.4.3 Sensitivity Program

X - type of payload variation: 1 - up payload only; 0 - equal up and down payload

A - variation factor

6.0 OUTPUTS

6.1 Input Data

Some of the inputs are displayed before the sizing program is used. The outputs are all the ΔV's and the sea level and vacuum specific impulse of both stages. A sample output is shown below.

To Orbit	On-Orbit	Cross-Range
30647.00	350.00	.00

	Booster	Orbiter
De-Orbit Delta V's	500.00	500.00
Hover Delta V's	1000.00	1000.00
Sea Level Specific Impulse	383.0	383.0
Vacuum Specific Impulse	459.0	459.0

6.2 Sizing Program

The printout for the sizing program shows the sizing criteria used (i.e., minimum gross weight, equal size stages),the up and down payload, and the type of stages used. If the payload is integrated into the orbiter, it is indicated.

The ascent ΔV's of both stages are shown, with the corresponding fraction of the total ascent ΔV. The average specific impulse of both stages during ascent is also shown.

Each stage mass fraction and gross weight is presented, as well as the total lift-off gross weight.

A sample of this printout is shown below:

Minimum Gross Weight		
Payload (Up - Down)	43000.00	43000.00
	Booster	Orbiter
Booster - Ballistic		
Orbiter - Lifting Body (Integrated)		
Ascent Delta V's	11412.94	19234.06
Fraction of Boost Velocity	.3724	.6276
Specific Impulse	438.42	459.00
Stage Mass Fraction	.9101	.7690
Stage Gross Weight	3.013822 06	1.232148 06
Lift-Off Gross Weight	4.288971 06	

An example of the abbreviated printout resulting by setting Y = 0 is shown below:

Stage Gross Weight	7.050019 05	7.032469 05
Lift-Off Gross Weight	1.489259 06	

6.3 <u>Sizing Program With Drop Tanks</u>

When expendable drop tanks are added to a vehicle design, the printout presents the new up and down payloads. The ascent velocity splits between the drop tanks, booster, and orbiter are then shown, with the corresponding weights of each. The lift-off gross weight is also presented. A sample of this printout is shown below.

Performance with Tip Tanks			
Payload (Up - Down)		165000.00	.00
	Tip Tanks	Booster	Orbiter
Ascent Delta V's	3481.40	10390.79	16087.81
Stage Gross Weight	1.4457 06	3.0138 06	1.2321 06
Lift-Off Gross Weight		5.856633 06	

6.4 <u>Sensitivity Program</u>

The output of the sensitivity program shows variations in payload with changes in various parameters of both stages individually and combined. Changes in payload due to changes in propellant weight and structural weight are presented in pounds of payload per percent change. The sensitivity to structural weight changes is presented for each stage individually and for the whole vehicle.

The changes in sea level and vacuum specific impulse for each stage are presented. Also, sensitivity to all the various ΔV's required by both stages is presented. A smaple printout is shown below.

Vehicle Sensitivities

Propellant Weight (lbs PL/%)	393.13		530.21
Structural Weight (lbs PL/%)	-125.48		-368.85
Sea Level I(s) (lbs PL/SEC)	44.7		19.7
Vacuum I(S) (lbs PL/FPS)	92.2		307.5
On-Orbit Delta V (lbs PL/FPS)			-6.3
Cross-Range Delta V (lbs PL/FPS)			0
Hover Delta V (lbs PL/FPS)	-1.4		-7.5
Lob-Retro Delta V (lbs PL/FPS	-1.1		
Total Impulsive Delta V (lbs PL/FPS)		-6.24	
De-Orbit Delta V (lbs PL/FPS)		-6.4	
Total Vehicle Structural Weight (lbs PL/%)		-493.59	

7.0 PROGRAM LISTING

A listing of the input, sizing, and sensitivity programs is presented.

1012-AEM-nma

A. E. Marks

INPUT PROGRAM

```
1.001 DEMAND B(2).
1.002 DEMAND B(3).
1.003 DEMAND V(1,0).
1.004 DEMAND V(2,0).
1.005 DEMAND V(3,6) IF B(2)=1.
1.006 DEMAND V(3,7) IF B(3)>0.
1.007 DEMAND V(4,0) IF B(3)=1.
1.008 DEMAND V(5,2) IF B(2)=1.
1.0085 DEMAND F(0,3) IF B(2)=1.
1.009 DEMAND V(5,3) IF B(3)=1.
1.01 DEMAND S(2) IF B(2)=0.
1.011 DEMAND S(3) IF B(3)=0.
1.012 DEMAND F(2,2).
1.013 DEMAND F(3,2).
1.03 TO STEP 1.08 IF B(2)>0.
1.04 SET F(0,1)=0.
1.046 SET T(2)=0.
1.047 SET U(2)=0.
1.05 DO PART 10 FOR N=2 IF S(2)=0.
1.06 SET V(3,6)=0.
1.07 SET V(5,2)=0.
1.075 TO STEP 1.2.
1.08 TO STEP 1.14 IF B(2)=2.
1.11 DO PART 10 FOR N=2.
1.13 TO STEP 1.20.
1.14 SET V(5,2)=0.
1.15 SET F(0,1)=0.
1.16 SET V(3,6)=0.
1.17 DO PART 9 FOR N=2.
1.2 TO STEP 1.27 IF B(3)>0.
1.22 SET V(3,7)=0.
1.23 SET V(4,0)=0.

1.24 SET V(5,3)=0.
1.244 SET T(3)=0.
1.246 SET U(3)=0.
1.25 DO PART 10 FOR N=3 IF S(3)=0.
1.265 TO STEP 1.4.
1.27 TO STEP 1.33 IF B(3)=2.
1.29 DO PART 10 FOR N=3.
1.31 TO STEP 1.4.
1.33 SET V(4,0)=0.
1.34 SET V(5,3)=0.
1.35 DO PART 9 FOR N=3.
1.4 SET I(5)=I(1).
1.41 DEMAND I(1).
1.42 TO STEP 1.5 IF I(1)<0.
1.43 DEMAND I(2).
1.431 DEMAND I(3).
1.432 DEMAND I(4).
1.44 DEMAND V(7,0).
1.48 TO STEP 1.6.
1.5 SET I(1)=I(5).
1.6 TO PART 7.

7.0 TYPE "                    INPUT DELTA V'S".
7.02 TYPE "    TO ORBIT          ON-ORBIT          CROSS-RANGE".
7.03 TYPE V(1,0),V(2,0),V(4,0) IN FORM 5.
7.09 TYPE FORM 1.
```

```
7.10 TYPE V(3,6),V(3,7) IN FORM 8.
7.12 TYPE V(5,2), V(5,3) IN FORM 2.
7.14 TYPE I(3),I(1) IN FORM 3.
7.16 TYPE I(4),I(2) IN  FORM 6.
7.164 TO STEP 7.43.
7.165 TYPE "MASS FRACTION COEFFICIENTS".
7.17 TYPE S(1),S(2),S(3) IN FORM 7.
7.18 TYPE T(1),T(2),T(3) IN FORM 7.
7.19 TYPE U(1),U(2),U(3) IN FORM 7.
7.43 DELETE ALL FORMS.
7.5 DELETE ALL PARTS.

8.005 DEMAND P(9).
8.006 TYPE N.
8.01 DEMAND M(1).
8.02 DEMAND L(1).
8.03 DEMAND M(2).
8.04 DEMAND L(2).
8.05 DEMAND M(3).
8.06 DEMAND L(3).
8.6 SET C(1,1)=1/M(2)**(1/3)-1/M(1)**(1/3).
8.7 SET C(1,2)=1/M(3)**(1/3)-1/M(2)**(1/3).
8.8 SET C(1,3)=1/M(3)-1/M(2).
8.9 SET C(1,4)=1/M(2)-1/M(1).
8.91 SET T(N)=(L(1)-L(2))*(C(1,3)/C(1,4))-(L(2)-L(3)).
8.92 SET T(N)=T(N)/(C(1,1)*C(1,3)/C(1,4)-C(1,2)).
8.93 SET U(N)=((L(2)-L(3))-T(N)*C(1,2))/C(1,3).
8.94 SET S(N)=L(3)+T(N)/(M(3)**(1/3))+U(N)/M(3).

8.95 TYPE S(N).
8.96 TYPE T(N).
8.97 TYPE U(N).
8.971 TO STEP 8.992 IF P(9)=0.
8.975 TYPE "        GROSS WEIGHT    MASS FRACTION".
8.98 DO STEP 8.99 FOR W(N)=M(3)((M(1)-M(3))/10)M(1).
8.99 TYPE W(N),Q IN FORM 4.
8.991 LINE.
8.992 LINE.
8.993 DELETE C.
8.994 DELETE P(9).

9.0 SET S(N)=.963415727.
9.1 SET T(N)=20.0070368.
9.2 SET U(N)=9517.1637.

10.0 SET S(N)=.917686451.
10.1 SET T(N)=.933803733.
10.2 SET U(N)=3521.18291.

FORM 1:
                                         BOOSTER           ORBITER

FORM 2:
HOVER DELTA V'S                          _____.__          _____.__

FORM 3:
SEA LEVEL SPECIFIC IMPULSE                ____._            ____._

FORM 4:
      ...........     _.____

FORM 5:
    _____.__       _____.__        _____.__
```

FORM 6:
VACUUM SPECIFIC IMPULSE ____.__ ____.__

FORM 7:
____.______ ____.______ ____.______

FORM 8:
DE-ORBIT DELTA V'S ____.__ ____.__

```
G:  S(N)-T(N)/(W(N)**(1/3))-U(N)/W(N)
Q:  1/(1+F(N,2)*(1/G-1))

    C =        4
    N =        3

 B(2) =        1
 B(3) =        1

 D(1) =         .00261096606

 D(2) =         .00217864924
 D(3) =         .00261096606
 D(4) =         .00217864924

 I(1) =      383
 I(2) =      459
 I(3) =      383
 I(4) =      459
 I(5) =      383

 S(1) =         .95
 S(2) =         .917686451
 S(3) =         .917686451

 T(1) =        0
 T(2) =         .933803733
 T(3) =         .933803733

 U(1) =        0
 U(2) =     3521.18291
 U(3) =     3521.18291

F(0,1) =       1
F(0,2) =        .177107248
F(0,3) =       1
F(2,1) =       -.00244098886
F(2,2) =       1
F(3,1) =       -.00244098886
F(3,2) =       1

V(1,0) =   30647
V(2,0) =     350
V(3,0) =     500
V(3,6) =     500
V(3,7) =     500
V(4,0) =       0
V(5,2) =    1000
V(5,3) =    1000
V(6,0) =       0
V(7,0) =    5400
```

SIZING PROGRAM

```
1.001 DO STEP 1.002 FOR C=1(1)4.
1.002 SET D(C)=1/I(C).
1.003 SET F(0,2)=V(7,0)/V(1,0).
1.004 SET F(3,1)=(D(2)-D(1))/F(0,2).
1.005 SET F(2,1)=(D(4)-D(3))/F(0,2).
1.01 DEMAND J.
1.05 DEMAND P(1).
1.06 DEMAND P(2).
1.0805 SET W(1)=10**9.
1.081 DEMAND X.
1.0815 TO STEP 1.05 IF X<0.

1.082 DEMAND Y.
1.0825 DO PART 2 FOR F(0,1)=X IF J<1.
1.0827 TO STEP 1.081 IF J<1.
1.083 DEMAND Z IF Y>0.
1.1 DO PART 2 FOR F(0,1)=X IF Y=0.
1.2 DO PART 2 FOR F(0,1)=X(Y)Z IF Y>0.
1.21 DEMAND V(6,0).
1.22 DO PART 5 IF V(6,0)>0.
1.3  TO STEP 1.081.

2.08 SET N=3.
2.09 SET V(3,1)=V(1,0)*(1-F(0,1)).
2.1 SET V(3,2)=V(3,7)+V(4,0).
2.2 SET V(3,3)=V(5,3).
2.3 SET R(3,1)=EXP(V(2,0)/(32.2*I(2))).
2.31 SET D(5)=D(2)-F(3,1)*((F(0,2)-F(0,1))**2)/(2.000001-2*F(0,1)).
2.32 SET D(5)=D(2) IF F(0,1)>=F(0,2).
2.325 SET I(5)=1/D(5).
2.33 SET R(3,1)=R(3,1)*EXP(V(3,1)/(32.2*I(5))).
2.4 SET R(3,2)=EXP(V(3,2)/(I(2)*32.2)).
2.5 SET R(3,3)=EXP(V(3,3)/(I(1)*32.2)).
2.6 SET R(3,0)=R(3,1)*R(3,2)*R(3,3).
2.63 SET L(4)=1-1/R(3,0).
2.645 SET L(5)= 1/(1+F(3,2)*(1/S(3)-1)).
2.6455 TO STEP 6.082 IF L(5)<L(4).
2.647 SET L(3)=(L(4)+L(5))/2.
2.648 SET L(3)=S(3) IF T(3)=0.
2.7 SET W(3)=((R(3,1)-1)*P(1)+(R(3,0)-R(3,1))*P(2))/(1-R(3,0)*(1-L(3))).
2.702 TO STEP 2.8 IF W(N)=0.
2.70205 DO PART 7 FOR N=3.
2.7021 SET L(5)=L(3) IF L(3)>Q.
2.7022 SET L(4)=L(3) IF L(3)<Q.
2.704 TO STEP 2.647 IF |(Q-L(3))|>.0001.
2.8 TO PART 3.

3.001 SET N=2.
3.01 SET V(2,1)=V(1,0)-V(3,1).
3.02 SET V(2,2)=V(2,1)*.007*(V(2,1)**(1/2)).
3.03 SET V(2,3)=V(5,2).
3.04 SET V(2,4)=V(3,1).
3.045 TO STEP 3.06 IF F(0,1)=0.
3.05 SET D(6)=(D(4)*(F(0,1)-F(0,2))+(D(3)+D(4))*F(0,2)/2)/F(0,1).
3.06 SET D(6)=D(3)+F(2,1)*F(0,1)/2 IF F(0,1)<F(0,2).
3.07 SET I(6)=1/D(6).
3.4 SET R(2,1)=EXP(V(2,1)/(32.2*I(6))).
3.41 SET D(7)=D(4)-F(3,1)*((F(0,2)-F(0,1))**2)/(2.000001-2*F(0,1)).
3.42 SET D(7)=D(4) IF F(0,1)>=F(0,2).
```

```
3.425 SET I(7)=1/D(7).
3.43 SET R(2,4)=EXP(V(2,4)/(32.2*I(7))+V(3,6)/(32.2*I(4))).
3.44 SET D(8)=D(3)+F(2,1)*F(0,1).
3.45 SET D(8)=D(4) IF F(0,1)>F(0,2).
3.455 SET I(8)=1/D(8).
3.46 SET R(2,2)=EXP(V(2,2)/(32.2*I(8))).
3.47 SET R(2,2)=R(2,4) IF R(2,4)<R(2,2).

3.48 SET R(2,2)=1.0 IF V(5,2)=0.
3.6 SET R(2,3)=EXP(V(2,3)/(I(3)*32.2)).
3.7 SET R(2,0)=R(2,1)*R(2,2)*R(2,3).
3.72 SET L(6)=1-1/R(2,0).
3.73 SET L(7)=1/(1+F(2,2)*(1/S(2)-1)).
3.735 TO STEP 6.082 IF L(7)<L(6).
3.745 SET L(2)=(L(6)+L(7))/2.
3.746 SET L(2)=S(2) IF T(2)=0.
3.8 SET W(2)=(R(2,1)-1)*(P(1)+W(3))/(1-R(2,0)*(1-L(2))).
3.802 TO STEP 3.9 IF W(N)=0.
3.8025 DO PART 7 FOR N=2.
3.803 SET L(7)=L(2) IF L(2)>Q.
3.804 SET L(6)=L(2) IF L(2)<Q.
3.807 TO STEP 3.745 IF |(Q-L(2))|>.0001.
3.9 TO PART 4 IF J<1.
3.91 TO PART 6 IF Y=0.
3.92 TO PART 6 IF J=1.


4.1 TO STEP 4.5 IF J<0.
4.2 TO STEP 4.4 IF W(1)<=W(2)+W(3)+P(1).
4.25 SET W(1)=W(2)+W(3)+P(1).
4.3 SET F(0,1)=F(0,1)+Y.
4.34 SET F(0,1)=1 IF F(0,1)>1.
4.35 TO PART 2.
4.4 TO PART 6 IF |Y|<.001.
4.41 SET Y=-Y/10.
4.45 TO STEP 4.25.
4.5 SET F(0,1)=F(0,1)*(W(3)/W(2))**.3.
4.6 TO STEP 2.08 IF |W(3)-W(2)|>500.
4.7 TO PART 6.


5.001 DEMAND P(3).
5.002 DEMAND P(4).
5.003 DEMAND S(1).
5.01 SET V(3,1)=((P(3)-P(4))+R(3,2)*R(3,3)*(P(4)+(1-L(3))*W(3))).
5.02 SET V(3,1)=I(2)*32.2*LOG((P(3)+W(3))/V(3,1)).
5.03 SET V(3,4)=V(3,1)-V(2,0).
5.04 SET V(2,2)=V(3,4)+V(3,6).
5.05 SET V(2,2)=V(1,0)-V(3,4) IF V(3,4)>(V(1,0)+V(3,6))/2.
5.06 SET R(2,2)=EXP(V(2,2)/(I(4)*32.2)).
5.07 SET V(2,1)=P(3)+W(3)+R(2,2)*R(2,3)*(1-L(2))*W(2).
5.08 SET V(2,1)=I(4)*32.2*LOG((P(3)+W(3)+W(2))/V(2,1)).
5.09 SET V(1,1)=V(6,0)-V(3,4)-V(2,1).
5.091 SET F(0,1)=V(1,1)/V(1,0).
5.092 DO STEP 3.05.
5.093 DO STEP 3.06.
5.094 DO STEP 3.07.
5.10 SET R(1,1)=EXP(V(1,1)/(I(6)*32.2)).
5.11 SET L(1)=S(1).
5.12 SET W(1)=(P(3)+W(3)+W(2))*(R(1,1)-1)/(1-R(1,1)*(1-L(1))).
```

```
5.13 TO STEP 6.082 IF V(3,1)<V(2,0).
5.20 TYPE "                    PERFORMANCE WITH TIP TANKS".
5.21 TYPE P(3),P(4) IN FORM 7.
5.22 TYPE FORM 8.
5.23 TYPE V(1,1),V(2,1),V(3,1)-V(2,0) IN FORM 12.
5.24 TYPE W(1),W(2),W(3) IN FORM 10.
5.25 TYPE W(1)+W(2)+W(3)+P(3) IN FORM 4.

6.01 TYPE "                                  MINIMUM GROSS WEIGHT" IF J=0.
6.02 TYPE "                                     EQUAL SIZE STAGES" IF J=-1.
6.023 TO STEP 6.03 IF Y=0.
6.024 TYPE P(1),P(2) IN FORM 7.
6.0245 TYPE FORM 11.
6.025 TYPE "BOOSTER  -  EXPENDABLE" IF B(2)=0.
6.026 TYPE "BOOSTER  -  BALLISTIC" IF B(2)=1.
6.027 TYPE "BOOSTER  -  LIFTING BODY" IF B(2)=2.
6.028 TYPE "ORBITER  -  EXPENDABLE" IF B(3)=0.
6.0285 TO STEP 6.02951 IF P(2)>0.
6.029 TYPE "ORBITER  -  BALLISTIC" IF B(3)=1.
6.0295 TYPE "ORBITER  -  LIFTING BODY" IF B(3)=2.
6.029501 TO STEP 6.0296.
6.02951 TYPE "ORBITER  -  BALLISTIC (INTEGRATED)" IF B(3)=1.
6.02952 TYPE "ORBITER  -  LIFTING BODY (INTEGRATED)" IF B(3)=2.
6.0296 LINE.
6.03 TYPE V(2,1),V(3,1) IN FORM 1 IF |Y|>0.
6.04 TYPE F(0,1),(1-F(0,1)) IN FORM 6 IF |Y|>0.
6.05 TYPE I(6),I(5) IN FORM 2 IF |Y|>0.
6.06 TYPE L(2),L(3) IN FORM 5 IF |Y|>0.
6.07 TYPE W(2),W(3) IN FORM 3.
6.08 TYPE W(2)+W(3)+P(1) IN FORM 4.
6.081 TO STEP 6.09.
6.082 TYPE "                                  CALL I.M.F.".
6.09 LINE.

7.0 SET W(4)=W(N).
7.1 SET W(5)=W(N)*G+F(N,2)*W(N)*(1-G).
7.2 SET W(N)=W(N)+(W(4)-W(5)).
7.3 TO STEP 7.1 IF |W(4)-W(5)|>1.
7.4 SET Q=W(N)*G/W(5).
7.5 SET W(N)=W(5).

10.2 DELETE ALL FORMS.
10.3 DELETE ALL FORMULAS.
10.5 DELETE ALL PARTS.

FORM 1:
ASCENT DELTA V'S
                                        _________.__        _________.__

FORM 2:
SPECIFIC IMPULSE
                                             _____.__            _____.__

FORM 3:
STAGE GROSS WEIGHT
                                        ............        ............

FORM 4:
LIFT-OFF GROSS WEIGHT
                                                 ............
```

FORM 5:
STAGE MASS FRACTION　　_.____　　_.____

FORM 6:
FRACTION OF BOOST VELOCITY　　_.____　　_.____

FORM 7:
PAYLOAD (UP - DOWN)　　________._　　________._

FORM 8:

	TIP TANKS	BOOSTER	ORBITER

FORM 10:

STAGE GROSS WEIGHT	....	....	....

FORM 11:

	BOOSTER	ORBITER

FORM 12:

ASCENT DELTA V'S	_______._	_______._	_______._

```
G:  S(N)-T(N)/(W(N)**(1/3))-U(N)/W(N)
```

SENSITIVITY PROGRAM

```
1.005 TO PART 4.
1.15 DEMAND X.
1.151 DEMAND P(2) IF X>0.
1.3 DEMAND A.
1.301 TO STEP 1.6 IF A=0.
1.302 SET B=(A-1)/A.
1.3021 SET M=100*(A-1).
1.3023 SET E(1)=0.
1.3024 SET E(2)=0.
1.3025 TO STEP 1.3193 IF W(2)=0.
1.31 DO PART 2 FOR W(4)=W(4)*A.
1.311 SET E(1)=(P(1)-C)/M.
1.312 SET W(4)=W(4)/A.
1.315 DO PART 2 FOR W(5)=W(5)*A.
1.316 SET E(2)=(P(1)-C)/M.
1.317 SET W(5)=W(5)/A.
1.3193 SET E(3)=0.
1.3194 SET E(4)=0.
1.3195 TO STEP 1.33 IF W(3)=0.
1.32 DO PART 2 FOR W(6)=W(6)*A.
1.321 SET E(3)=(P(1)-C)/M.
1.322 SET W(6)=W(6)/A.
1.325 DO PART 2 FOR W(7)=W(7)*A.
1.326 SET E(4)=(P(1)-C)/M.
1.327 SET W(7)=W(7)/A.
1.329 SET D(1)=D(1)/A.
1.33 DO PART 2 FOR I(1)=I(1)*A.
1.331 SET E(05)=(P(1)-C)/(I(1)*B).
1.3315 SET D(1)=D(1)*A.
1.332 SET I(1)=I(1)/A.
1.333 SET D(2)=D(2)/A.
1.335 DO PART 2 FOR I(2)=I(2)*A.
1.336 SET E(06)=(P(1)-C)/(I(2)*B).
1.337 SET I(2)=I(2)/A.
1.338 SET D(2)=D(2)*A.
1.339 SET D(3)=D(3)/A.
1.340 DO PART 2 FOR I(3)=I(3)*A.
1.341 SET E(07)=(P(1)-C)/(I(3)*B).

1.342 SET I(3)=I(3)/A.
1.343 SET D(3)=D(3)*A.
1.344 SET D(4)=D(4)/A.
1.345 DO PART 2 FOR I(4)=I(4)*A.
1.346 SET E(08)=(P(1)-C)/(I(4)*B).
1.347 SET I(4)=I(4)/A.
1.348 SET D(4)=D(4)*A.
1.350 DO PART 2 FOR V(1,0)=V(1,0)*A.
1.351 SET E(09)=(P(1)-C)/(V(1,0)*B).
1.352 SET V(1,0)=V(1,0)/A.
1.353 SET E(10)=0.
1.354 TO STEP 1.358 IF V(2,0)=0.
1.355 DO PART 2 FOR V(2,0)=V(2,0)*A.
1.356 SET E(10)=(P(1)-C)/(V(2,0)*B).
1.357 SET V(2,0)=V(2,0)/A.
1.358 SET E(11)=0.
1.359 TO STEP 1.363 IF V(3,7)=0.
1.36 DO PART 2 FOR V(3,7)=V(3,7)*A.
1.361 SET E(11)=(P(1)-C)/(V(3,0)*B).
1.362 SET V(3,7)=V(3,7)/A.
1.363 SET E(12)=0.
```

```
1.364 TO STEP 1.368 IF V(4,0)=0.
1.365 DO PART 2 FOR V(4,0)=V(4,0)*A.
1.366 SET E(12)=(P(1)-C)/(V(4,0)*B).
1.367 SET V(4,0)=V(4,0)/A.
1.368 SET E(13)=0.
1.369 TO STEP 1.374 IF V(5,3)=0.
1.37 DO PART 2 FOR V(5,3)=V(5,3)*A.
1.371 SET E(13)=(P(1)-C)/(V(5,3)*B).
1.372 SET V(5,3)=V(5,3)/A.
1.374 SET E(14)=0.
1.375 TO STEP 1.379 IF V(5,2)=0.
1.376 DO PART 2 FOR V(5,2)=V(5,2)*A.
1.377 SET E(14)=(P(1)-C)/(V(5,2)*B).
1.378 SET V(5,2)=V(5,2)/A.
1.379 SET E(15)=0.
1.38 TO STEP 1.385 IF V(3,1)<(V(1,0)-V(3,0))/2.
1.3805 SET E(15)=0.
1.3806 TO STEP 1.385 IF V(2,2)=0.
1.3808 SET Q=1.
1.381 DO PART 2 FOR F(0,3)=F(0,3)*A.
1.382 SET E(15)=(P(1)-C)/(V(2,2)*B).
1.383 SET V(2,2)=V(2,2)/A.
1.3835 SET F(0,3)=F(0,3)/A.
1.384 SET Q=0.
1.385 SET W(5)=W(5)*A.
1.386 DO PART 2 FOR W(7)=W(7)*A.
1.387 SET E(16)=(P(1)-C)/M.
1.388 SET W(5)=W(5)/A.
1.389 SET W(7)=W(7)/A.
1.5 TYPE "                                VEHICLE SENSITIVITIES".
1.55 TO PART 5.
1.6 STOP.


2.001 SET W(3)=W(6)+W(7).
2.002 SET W(2)=W(4)+W(5).
2.01 SET V(2,1)=V(1,0)-V(3,1).
2.011 SET V(2,1)=0 IF W(2)=0.
2.015 SET V(3,1)=H IF W(2)=0.
2.0155 TO STEP 2.02 IF W(2)>0.
2.016 SET V(3,1)=V(1,0) IF V(1,0)>H.
2.018 SET F(0,1)=V(2,1)/V(1,0).
2.02 SET D(7)=D(2)-F(3,1)*((F(0,2)-F(0,1))**2)/(2.000001-2*F(0,1)).
2.03 SET D(7)=D(2) IF F(0,1)>=F(0,2).
2.035 SET I(7)=1/D(7).
2.04 SET R(4)=EXP(V(3,1)/(I(7)*32.2)).
2.05 SET R(5)=EXP(V(2,0)/(I(2)*32.2)).
2.06 SET R(6)=EXP((V(3,7)+V(4,0))/(I(2)*32.2)).
2.07 SET R(7)=EXP(V(5,3)/(I(1)*32.2)).
2.08 SET R(8)=R(4)*R(5)*R(6)*R(7).
2.081 TO STEP 2.09 IF X>0.
2.082 SET P(1)=W(6)/(R(8)-1)-W(7).
2.084 TO STEP 2.11.
2.09 SET P(1)=(W(6)+(1-R(8))*W(7))/(R(4)*R(5)-1).
2.10 SET P(1)=P(1)+(R(4)*R(5)-R(8))*P(2)/(R(4)*R(5)-1).
2.11 SET P(5)=P(1).
2.12 TO PART 3 IF W(2)>0.

3.001 SET D(5)=(D(4)*(F(0,1)-F(0,2))+(D(3)+D(4))*F(0,2)/2)/F(0,1).
3.002 SET D(5)=D(3)+F(2,1)*F(0,1)/2 IF F(0,1)<F(0,2).
3.0025 SET I(5)=1/D(5).
3.003 SET D(6)=D(3)+F(2,1)*F(0,1).
```

```
3.0035 SET D(6)=D(4) IF F(0,1)>F(0,2).
3.0036 SET I(6)=1/D(6).
3.004 SET D(8)=D(4)-F(3,1)*((F(0,2)-F(0,1))**2)/(2.000001-2*F(0,1)).
3.005 SET D(8)=D(4) IF F(0,1)>=F(0,2).
3.006 SET I(8)=1/D(8).
3.01 SET R(1)=EXP(V(2,1)/(I(5)*32.2)).
3.015 SET V(2,2)=V(2,1)*F(0,3) IF V(3,1)>=(V(1,0)-V(3,6))/2.
3.02 SET R(2)=EXP(V(2,2)/(I(6)*32.2)).
3.03 SET R(2)=EXP((V(3,1)+V(3,6))/(I(8)*32.2)) IF V(3,1)<(V(1,0)-V(3,6))/2.
3.035 SET R(2)=1 IF V(5,2)=0.
3.04 SET R(3)=EXP(V(5,2)/(I(3)*32.2)).
3.05 SET R(9)=R(1)*R(2)*R(3).
3.06 SET P(1)=(W(4)-(R(9)-1)*W(5))/(R(1)-1)-W(3).
3.07 SET P(6)=P(1).
3.09 TO STEP 3.2 IF |P(5)-P(6)|<1.
3.10 SET J=V(3,1) IF P(5)>=P(6).
3.11 SET K=V(3,1) IF P(6)>P(5).
3.12 SET V(3,1)=(J+K)/2.
3.13 TO PART 2.
3.2 SET J=0.
3.21 SET K=H.
3.22 SET P(1)=(P(5)+P(6))/2.

4.01 SET C=P(1).
4.02 SET K=V(1,0).
4.03 SET J=0.

4.04 SET G=V(3,1).
4.05 SET H=V(1,0).
4.10 SET W(4)=L(2)*W(2).
4.11 SET W(5)=W(2)-W(4).
4.12 SET W(6)=L(3)*W(3).
4.13 SET W(7)=W(3)-W(6).
4.15 SET V(8,0)=V(7,0).
4.18 SET Q=0.
4.20 TO STEP 1.15.

5.01 TYPE E(1),E(3) IN FORM 1.
5.02 TYPE E(2),E(4) IN FORM 2.
5.03 TYPE E(7), E(5) IN FORM 3.
5.04 TYPE E(8),E(6) IN FORM 4.
5.05 TYPE E(10) IN FORM 5.
5.07 TYPE E(12) IN FORM 7.
5.08 TYPE E(14),E(13) IN FORM 8.
5.085 SET E(15)=0 IF V(5,2)=0.
5.09 TYPE E(15) IN FORM 9.
5.10 TYPE E(9) IN FORM 10.
5.11 TYPE E(11) IN FORM 6.
5.12 TYPE E(16) IN FORM 11.

FORM 1:
PROPELLANT WEIGHT (LBS PL/%)                      _____.__        _____.__

FORM 2:
STRUCTURAL WEIGHT (LBS PL/%)                      _____.__        _____.__

FORM 3:
SEA LEVEL I(S) (LBS PL/SEC)                       _____._         _____._

FORM 4:
VACUUM I(S) (LBS PL/SEC)                          _____._         _____._
```

FORM 5:
ON-ORBIT DELTA V (LBS PL/FPS) ______.__

FORM 6:
DE-ORBIT DELTA V (LBS PL/FPS) ______.__

FORM 7:
CROSS-RANGE DELTA V (LBS PL/FPS) ______.__

FORM 8:
HOVER DELTA V (LBS PL/FPS) ______.__ ______.__

FORM 9:
LOB-RETRO DELTA V (LBS PL/FPS) ______.__

FORM 10:
TOTAL IMPULSIVE DELTA V (LBS PL/FPS) ______.___

FORM 11:
TOTAL VEHICLE STRUCTURAL WEIGHT (LBS PL/%) ______.___

REFERENCES

1. Study of Integral Launch and Reentry Vehicle Systems, North American Rockwell, Space Division, SD 69-573-1, December, 1969.

2. Space Shuttle Final Technical Report, General Dynamics Convair Division, GDC-DCB69-046, October 31, 1969.

3. Integral Launch and Reentry Vehicle Systems, McDonnell Douglas Corporation, MDC E0049, November, 1969.

4. NASA Space Shuttle Task Group Report, June 12, 1969.

5. Study of Advanced Multipurpose Large Launch Vehicles, Boeing Company, CR 73154, January, 1968.

6. Project SERV, A Space Shuttle Feasibility Study, Chrysler Corporation Space Division, AE-PB-69-51, November 19, 1969.

7. Launch Site Recovery of Ballistic Boosters, E. D. Marion, Bellcomm Memorandum for File, B70-06019, June 9, 1970

BELLCOMM, INC.

Subject: Computer Program for Launch Vehicle Sizing and Sensitivities Case 105

From: A. E. Marks

DISTRIBUTION LIST

NASA Headquarters

W. O. Armstrong/MTX
P. E. Culbertson/MT
S. S. Dimaggio/MAR
C. J. Donlan/MD-T
E. W. Hall/MTG
T. A. Keegan/MA-2
B. Milwitzky/MAL
A. D. Schnyer/MTE
A. O. Tischler/RP
M. G. Waugh/MT-1
J. W. Wild/MTE

MSC

D. E. Fielder/HC

MSFC

W. G. Huber/PD-SA-DIR

Bellcomm

G. M. Anderson
D. O. Baechler
A. P. Boysen
D. R. Hagner
E. E. Hillyard
N. W. Hinners
P. F. Long
J. Z. Menard
J. M. Nervik
J. W. Timko
M. P. Wilson
All Members, Depts. 1011
 1012, 1013
Central Files
Department 1024 File
Library

Abstract Only

I. M. Ross
R. L. Wagner