N65-35612

COMPUTERS FOR SIMULATION OF SPACE VEHICLE SYSTEMS

by

R. M. Howe

The University of Michigan

1. Introduction

The importance of simulation in the design and testing of space vehicle systems is self evident. Simulation also plays a vital role in the training of both astronauts and ground crews. It may even form a vital part of the mechanization of on-board displays. One of the most important parts of any space-system simulator is the computer which calculates the simulated performance of the vehicle. In previous years the simulation of aircraft and missiles in six degrees of freedom has posed challenging computational problems. These problems become even more severe in the simulation of space vehicles, since the dynamic range of problem variables becomes much larger and the flight times become much longer. For example, simulation of the full Apollo mission requires accurate computation of vehicle trajectories over the earth-moon distance, more than 10^9 feet, for times of many hours or even days. Yet the same computer is required to simulate docking maneuvers with resolution of better than one foot, as well as reentry maneuvers where vehicle pitching and yawing frequencies up to several cps and higher are encountered. In fact, accurate simulation of jet-reaction attitude control systems, including precise computation of fuel usage, may require computer time resolution in the millisecond region.

Although the above example is a severe one, it illustrates the type of problems that may face the simulator designer in choosing the appropriate computer system for the task. The purpose of this paper is to outline some of the performance considerations of analog and digital computers which are important in space-vehicle simulation. The choice of axis systems such that computer performance requirements will be less severe will also be described, along with some typical computer results.

In considering the simulation of space vehicles there is a very wide range of problem types which one encounters. We have already mentioned a full-mission Apollo simulation in real time, which might be used for mission studies or actual astronaut training. On the other hand one might be interested in simulating just the rotational degrees of freedom of one stage of a space-vehicle booster system as part of an autopilot-design study. Such a simulation need not be in real time unless actual flight hardware is substituted for parts of the computer as the design progresses. Or repetitive simulation of space-vehicle reentry trajectories at speeds thousands of times faster than real time may be required to implement a predictor display, or to mechanize a com= putation of the optimum reentry trajectory. Most of the discussion that follows will concentrate on the computational requirements for the xis-degree-of-freedom full mission type of simulation, since other simulation tasks are usually part of this overall, larger task.

2. Scope of the Full-Mission Simulation. Analog, Hybrid, or Digital Computation

In a simulation that involves lunar or planetary missions accurate trajectory computation requires that a high-precision digital computer be used. This is because of the enormous range in force magnitudes which act on the vehicle, and because analog computers have precision limited to the order of 0.01 percent of full scale. In fact, care must be taken in a digital simulation of such trajectories to make certain that roundoff and truncation errors do not generate unacceptably large errors. For example, in Figure 1 are shown the results of a study performed by the Link Division of General Precision, Inc., where a highly eccentric earth-satellite orbit with apogee equal approximately to the earth-moon distance has been computed using a fourth-order Runge-Kutta integration formula. The abscissa is essentially proportional to the vehicle distance from the earth, with check point 60 representing lunar distance. Hence, except for the lack of inclusion of the lunar gravity, the results are representative of an Apollo lunar trajectory. The ordinate is position error, which builds up to 30,000 feet using a 24 bit word length (single precision). This would be unacceptably large in many cases. On the other hand, using double precision reduces the error to less than 2000 feet, an acceptable number. In many cases simulation of space-vehicle trajectories may have to be accomplished using two simultaneous integrations, one at low iteration rate using double precision and a second at high iteration rate using single precision. The second integration provides trajectory information at high enough rates to be used to drive visual displays, but gets updated at the end of each computational cycle of the first integration.

Although the space-vehicle trajectory may require digital solution, one may wish to solve the rotational equations on an analog computer. This is because the accuracy requirements of the rotational equations are much less severe, but at the same time the frequency components are much higher.

Although current digital computers are capable of solving both rotational and translational equations in real time, the initial programming of the equations is extremely expensive as compared with an analog computer. Also, one may wish to solve the equations faster than real time in order to avoid long solution times in engineering studies, in which case an analog computer must be used. Figures 2 and 3 show block diagrams of the equations which might be solved on the analog and digital portions, respectively, of such a simulator. They also illustrate the complexity of the equations which must be solved in such a six-degree-of-freedom simulation. When both analog and digital computers are combined in such a simulation, we have a hybrid computer, which requires analog-to-digital converters and digital-to-analog converters for communication back and forth between the machine elements. Actually, an all-digital simulation requires such converters to serve as input-output devices, in any case. Although hybrid computers offer an obviously efficient solution to the six-degree-of-freedom space-vehicle simulation problem, programming the computers and interface equipment is by no means a trivial problem. For this reason, it is probably well to avoid hybrid simulations of this type unless there is a sizeable economic advantage or unless it is required because of performance considerations.

One of the computational areas where the digital computer has a sizeable advantage over the analog is in generation of multivariable functions. For example, suppose that the rotational equations of motion are to be solved on the analog computer, except that the digital computer is to be used to calculate the aerodynamic pitching moment, since it is a complicated function of Mach number M, altitude h, and angle of attack α. A block diagram of the pitch loop is shown in Figure A, where the pitch acceleration $\ddot{\theta}$ is computed in analog voltage form by dividing the total pitching moment

$M_a + M_c$ by the pitching inertia I_{yy}. $\ddot{\theta}$ is then integrated twice to obtain an analog voltage θ representing the vehicle attitude. The flight-path angle γ is subtracted from this to compute the angle of attack α, which is then converted to digital form and fed, along with Mach number and altitude, into a digital computer which calculates the aerodynamic pitching moment M_a. This is in turn converted back to analog form and added to the voltage representing the jet-reaction pitching moment M_c, thus closing the computational loop.

Let us now consider the effect of the finite calculation time T required by the digital computer to calculate the pitching moment M_a. To simplify the analysis we will assume that the flight-path angle γ remains constant, as well as the Mach number M and the altitude h. Furthermore, for these constant values let us make the assumption that the aerodynamic pitching moment M_a is approximately a linear function of α. Then the computer loop as shown behaves as a mass-spring system with a natural frequency which we shall denote by ω_p. This is just the approximate transient frequency of the vehicle pitching dynamics under these flight conditions.

By means of the method of Z-transforms, such as is used in analyzing sampled data systems, it can be shown that the computing loop in Figure 4 has a transient frequency ω and damping ratio ζ (fraction of critical damping) given approximately by[1]

$$\omega = \omega_p \left(1 - \frac{7}{12} \omega_p^2 T^2\right),$$

$$\zeta = -\frac{3}{4} \omega_p T, \quad T << \frac{1}{\omega_p} \tag{2.1}$$

[1] Howe, R. M., Error Analysis of Combined Analog-Digital Computer Systems, Information and Control Engineering Program, The University of Michigan, Ann Arbor, Michigan, May, 1964.

Here ω_p is the ideal oscillatory frequency. Ideally, the damping ratio ζ - 0.T is the digital cycle time (1/T is the number of computations of M_a per second). The effect of the digital computer cycle time T is second order on the frequency, but is first order on the damping. The computer loop as shown will have a transient which exhibits a fractional growth in amplitude per cycle equal to $-2\pi\zeta$. A more complete simulation of the pitch system would include a slight amount of aerodynamic damping (moment proportional to $\dot{\theta}$), but in any event the damping-ratio error will be approximately that given in Eq. (2.1). For example, if ω_p is 10 radians per second, the digital cycle time T must be in 1 millisecond to have a damping-ratio error less than 1 percent.

The accuracy of the computing loop in Figure 4 can be improved remarkably by updating the angle of attack into the A-D converter by the sum of the time-delay exhibited by the digital computer (t seconds) and the D-A converter (0.5 T seconds on the average, assuming a zero-order hold). Noting that $\dot{\alpha} = \dot{\theta}$ ($\dot{\gamma}$ is negligible), we feed $\alpha + 1.5\dot{\theta}T$ into the A-D converter. Under these conditions one can show that the grequency ω and damping ratio ζ are given by[2]

$$\omega = \omega_p \left(1 + \frac{13}{24} \omega_p^2 T^2\right),$$

$$\zeta = -\omega_p^4 T^4 , \quad T < < \frac{1}{\omega_p} \qquad (2.2)$$

Here the damping ratio error is fourth-order in T, representing an enormous improvement over the result in Eq. (2.1), where we failed to update the analog input to the A-D converter.

Analyses similar to the above can be applied to all-digital computation

---

[2] Howe, R. M., op. cit.

of second-order computing loops, where the error in frequency and damping using various numerical integration schemes is determined as a function of iteration period T. Although actual three and six-degree-of-freedom flight equations are much more complicated, those portions of the problem with the highest frequency outputs usually behave approximately like the loop in Figure 4. Thus the simplified analysis as presented here yields valuable insight into the computational speeds required to obtain given accuracy.

Most pure analog elements, such as summing amplifiers, integrating amplifiers, coefficient potentiometers, function-generators, etc., have dynamic behavior in the problem-frequency range which approximates that of a first-order linear system, i.e., with a transfer operator of the form $K(1 + \tau_p)^{-1}$, where τ is the equivalent time constant.[3] For typical state-of-the-art analog computer components τ is the order of magnitude of 1 microsecond. In implementing computing loops similar to Figure 4 using analog components exclusively, one can show that the damping-ratio error is approximately given by

$$\zeta \cong \frac{\omega_p}{2} \sum_{i=1}^{N} \tau_i \qquad (2.3)$$

where the τ_i represent, respectively, the time constant of each of the N analog elements around the loop. A typical value for $\Sigma \tau_i$ would be 10^{-5} seconds, in which case ω_p values up to 2000 radians per second (320 cycles per second) could be handled with less than one percent damping error. For many problems this means that the solution can be run at much faster speeds than real time.

---

[3] Howe, R. M. Design Fundamentals of Analog Computer Components, Chapter 2, D. Van Nostrand, 1961.

3. Axis Systems for Computation of Space and Reentry Trajectories

In writing the translational equations of motion to be solved by the computer, either analog or digital, the forces, including inertial, are normally summed along each of three axes and the resulting accelerations are integrated twice to obtain velocity and position coordinates. The importance of the choice of this axis system is readily illustrated by comparing trajectory computation with the accelerations and velocities referred to flight path axes and conventional body axes. For a near-circular satellite the velocity along the x flight-path axis is relatively constant, and velocity components along the y and z axes are zero by definition. The velocity along the flight path can be computed as a small difference from circular-orbit velocity. If, however, the equations of motion are solved in body axes and the vehicle tumbles, then the components of velocity along the x, y, and z body axes may be either positive or negative and the magnitude of any component may be as great as the total vehicle velocity. Therefore, each velocity component must be scaled for at least twice the maximum flight-path velocity. Furthermore, each velocity component must be capable of changing from maximum positive to maximum negative in the time it takes for half a body revolution. The equivalent artificial acceleration due to the rotating axes can be many times as large as the net gravitational and inertial acceleration acting on the body; hence the true acceleration might be masked by errors in computing accelerations along the body axes. Using the same computer, the flight-path axis computation will obviously be much more accurate than the body-axis computation.

It has been common practice to express the trajectory equations of motion directly in a rectnagular cartesian inertial frame with origin at the center of the earth. In such a coordinate system, it is apparent that the x, y, and z distances must be scaled to range through at least twice the apogee radius from the center of the earth, the x, y, and z velocity components must be scaled to range through at least twice the orbital velocity, and the x, y, and z acceleration components must be scaled to range through at least twice the gravitational acceleration. Since a velocity error of more than a few feet per second cannot be tolerated, it is obvious that great computational accuracy is required to avoid unacceptable error build-up over the integration period of hours or even days, when such unfavorable scaling is used. The classical method of obtaining good scaling of the computation is to write the equations of motion in terms of the elements of the osculating orbit. This "*method of variation of parameters*" has several serious drawbacks for real time simulation. The equations are more complicated than the usual Newtons' Laws equations. There are a number of bothersome singularities in them, and the results of the computation are obtained in terms of orbit elements, rather than conventional length or angle coordinates, hence require further processing if one wishes to display a trajectory.

A basic objective in selection of axes for the translational equations, then, is to choose the axes to obtain some of the scaling benefits of the method of variation of parameters but to retain direct computation of position coordinates for display purposes. The selection of the reference frame in which the equations of motion are to be expressed requires a compromise between good scaling, simplicity of the equations and convenience of use of the computed results. For best scaling, it is apparent that the reference direction should be aligned with the velocity vector, $\vec{V}_p$. The equations of motion are somewhat simpler, however, if the reference direction is aligned with the horizontal component of the velocity vector. For orbits of small eccentricity there is little difference in scaling between the two systems of equations, so the latter reference frame, called the H-frame, has proven to be preferable.[4]

The H-frame is defined as a rectangular cartesian coordinate system, x_h, y_h, z_h with origin at the vehicle center of gravity. The x_h axis is normal to the radius from the center of the earth and points forward in the plane of the motion. The y_h axis is horizontal and normal to the plane of the motion. The z_h axis is along the radius from the center of the earth, positive downward.

Considering the components of the acceleration with respect to inertial space along the x_h, y_h, z_h directions, we obtain the following equations of motion for a vehicle of mass m, velocity components U_h, V_h, W_h ($V_h = 0$ by definition) and external forces X_h, Y_h, Z_h:[4]

$$\dot{U}_h - \frac{U_h W_h}{r} = \frac{X_h}{m} \tag{3.1}$$

$$r_h U_h = \frac{Y_h}{m} \tag{3.2}$$

$$\dot{W}_h = -\ddot{r} = \frac{U_h^2}{r} - \frac{g_0 r_0^2}{r^2} + \frac{Z_h}{m} \tag{3.3}$$

In these equations r is the radial distance from the center of gravitational attraction, g_0 is the gravity acceleration at a nominal distance r_0, and r_h is the z_h or radial component of H-frame angular velocity.

Note that Eqs. (3.1) and (3.3) can be solved for the in-plane motion

---

[4] Fogarty, L. E., and Howe, R. M., "Flight Simulation of Orbital and Reentry Vehicles" - IRE Transactions on Electronic Computer Vol. EC-11, Aug. 1962.

independently of Eq. (3.2). Eq. (3.1) can be rewritten as

$$r\dot{U}_h + \dot{r}U_h = \frac{rX_h}{m}$$

which can be integrated directly and solved for U_h. Thus

$$U_h = \frac{1}{r}\left[\int_0^t \frac{rX_h}{m}\, dt + r(0) \quad U_h\,(0)\right] \tag{3.4}$$

In the absence of external force X_h, Eq. (3.4) allows the horizontal velocity U_h to be solved directly as an algebraic function of radial distance r. Thus an open-ended integration of $\dot{U}_h$ to obtain U_h, as would be computed from Eq. (3.4), is avoided and much higher computational accuracy results. The time duration over which sizeable non central-force field external forces X_h exist is relatively short (e.g., thrusting and reentry) and the open-ended integration in Eq. (3.4) gives no problem. Actually, Eq. (3.4) is just the statement of conservation of angular momentum.

A further scaling advantage is obtained by writing Eqs. (3.3) and (3.4) in terms of the variation δr in radial distance from the fixed mean radius r_0 and the variation δU_h in horizontal velocity component from the circular orbit velocity U_{h_0} for radius r_0 ($U_{h_0} = \sqrt{g_0 r_0}$). Thus we let

$$r = r_0 + \delta r \text{ and } U_h = \sqrt{g_0 r_0} + \delta U_h \tag{3.5}$$

Rewriting Eq. (3.3) in terms of δr and δU_h, we obtain

$$\delta\ddot{r} = \frac{2\sqrt{g_0 r_0}\,\delta U_h + (\delta U_h)^2}{(r_0 + \delta r)^2} - \frac{Z_h}{m} \qquad (3.6)$$

For near circular satellite orbits the net radial acceleration $\delta\ddot{r}$ is the small difference between gravity acceleration $g_0 r_0^2/r^2$ and centrifugal acceleration U_h^2/r. The first term on the right side of Eq. (3.6) represents this difference directly and hence allows much more favorable computer scaling. In fact, Eqs. (3.4) and (3.6) can be solved with satisfactory accuracy using an analog computer, whereas equations based on more conventional coordinate systems are completely unsuitable for analog solution.

Consider, for example, a low eccentricity orbit with the analog computer scaled to allow a range in δr of $\pm$ 80 statute miles. We assume a pure central force gravity field with no other external forces.[5] Several typical analog solutions starting with the vehicle at perigee are shown in Figure 5. For the case where the initial perigee altitude is at 16 miles the first apogee is within 200 feet of the correct value of 146.5 miles and the second perigee is within 200 feet of the initial value, indicating orbit closure to that accuracy.

Also shown is a case representing, ideally, injection into a circular orbit. The computer solution, blown up by a factor of 200, is shown in the figure and indicates that the altitude holds to within 200 feet of the initial value over one orbital distance. The results in Figure 5 were computed at 100 times real time. Comparable results were obtained in real time and at 10 times real time. In the computer circuit used for these recordings the horizontal velocity deviation δU_h was represented to allow a full-scale

---

[5] Fogarty, L. E., and Howe, R. M., op. cit.

excursion of $\pm U_{h_0}$ (corresponding to full circular-orbit velocity) even though the actual range in δU_h in the solution shown in Figure 5 is a small fraction of this. Similarly, W_h was scaled to a maximum of $\pm U_{h_0}/5$ even though the actual range for the near-circular orbits shown in much smaller. This was to allow for utilization of the same scaling in the computation of ascent and reentry trajectories. Despite this unfavorable scaling for the near circular orbit case, the results using the H-frame coordinates appear to be quite favorable. The H-frame has proven to be extremely advantageous for digital computation, too.

In the above example the angular momentum integral was used to compute horizontal velocity directly, as indicated in Eq. (3.4). One can utilize conservation of energy as a constraint, too.[6] In this scheme the total energy, potential and kinetic, as determined from the position and velocity coordinates, is subtracted from the known total energy as computed from the initial energy plus the time integral of the energy rate of change due to external forces. This difference defines the energy error ε. A term proportional to $\varepsilon\,\dot{r}$ is then added to the right side of Eq. (3.6) to drive $\delta\ddot{r}$ in such a direction as to make $\varepsilon = 0$.

The effectiveness of the use of energy constraint is illustrated by considering the analog solution of a highly-eccentric satellite orbit. Dimensionless radius $\rho = r/r_0$ is shown as a function of dimensionless time τ in Figure 6, where the starting perigee = $0.12r_0$ and the resulting apogee should be $1.88r_0$. In this case the initial horizontal velocity $U_h = 3.96U_{h_0}$ which should yield a minimum horizontal velocity at apogee of $0.252U_{h_0}$. The analog computer exhibited the periodic solution shown in Figure 6 with a

---

[6] Fogarty, L. E., and Howe, R. M. Axis Systems for Analog and Digital Computation of Space and Reentry Trajectories, Application Report, Applied Dynamics, Inc., Ann Arbor, Michigan, September, 1963

measured apogee of $\rho = 1.877$ compared with the ideal value of 1.880 and a horizontal velocity at apogee equal to $0.256U_{h_0}$ instead of the ideal value of $0.252U_{h_0}$. Both results agree with the corresponding theoretical values to the order of 0.1 percent of full scale, which is reasonable considering that the required multiplications were carried out using quarter-square multipliers with accuracies of approximately 0.05 percent.

Figure 7 shows a number of cycles with the energy-correction term removed from the circuit. The resulting solution decays slowly with time. After some 75 orbits the energy correction was repatched to the $\ddot{r}$ integrator, with the result that the solution returns within one cycle to its correct amplitude.

As a final example of space-vehicle trajectories, consider the simulation in three dimensions of a lifting reentry vehicle. The problem was set up on an analog computer using the H-frame equations as described earlier. The three dimensional equations were solved for a number of different bank-angles, ϕ for fixed ballistic coefficient values, and for fixed lift-to-drag ratios L/D. A typical recording of altitude versus downrange distance is shown in Figure 8. Figure 9 shows a plot of altitude L versus longitude λ for a nominal equatorial trajectory, and a number of additional trajectories, each with a different fixed bank angle.[7] Although these computer runs were made using an x-y plotter (approximately 800 times real time) the problem was also solved in repetitive operation at 80,000 times real time and the different values of bank angle were set for each trajectory using a sample-hold circuit driven by a low frequency sawtooth wave. The result, when displayed as latitude L versus longitude λ on the face of a cathode-ray oscilloscope, appears as a

---

[7] Fogarty, L. E., and Howe, R. M., Op. Cit.

more-or-less continuous picture similar to Figure 9. If the L/D ratio is now varied slowly, one can sweep out very quickly the footprint of possible landing points for the lifting reentry vehicle.

It should be noted that solution of lifting reentry trajectories at these high speeds requires analog computing components of considerable bandwidth, since the trajectory oscillations near the end of the reentry have a frequency of approximately 1 kilocycle.

4. Summary

We have seen that the simulation of space-vehicle systems poses sizable computer problems, depending on the scope of the particular simulation being undertaken. Many simulation problems, including some real-time problems, are amenable to all-digital solution. Other simulation problems are more suitable for all analog mechanization. There is a sizable class of problems which is best solved on hybrid computer systems, although the programming complexity in such a mechanization should not be overlooked. Whatever type of computer system is used, it is important to choose appropriate axis systems in writing equations of motion so that required computer precision and speed is minimized. Long term future trends in space-vehicle simulation will probably see digital computers used more for real-time simulation of computer problems, whereas analog computers will be used for faster-than-real-time problems, such as trajectory optimization studies and predictor displays. Combined analog-digital systems will probably be used less for real-time simulation but very much more for optimization and other design studies, where the speed of the analog and the precision and function-storage capability of the digital can be combined to advantage.

Single Precision Error Vs. Double Precision Error
(Runge-Kutta Fourth Order)

Step Size 107 Seconds

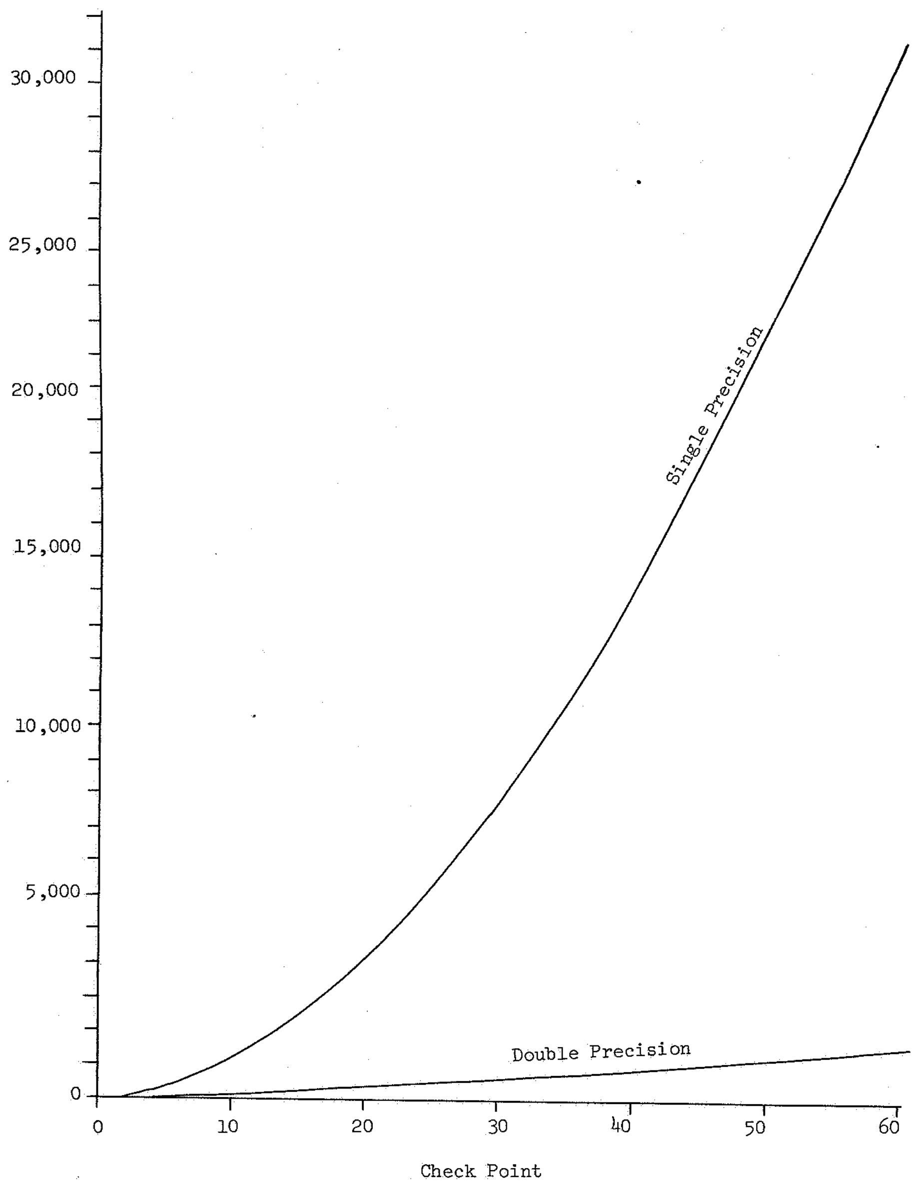


Figure 1. Single precision versus double precision error in computing an earth-moon trajectory

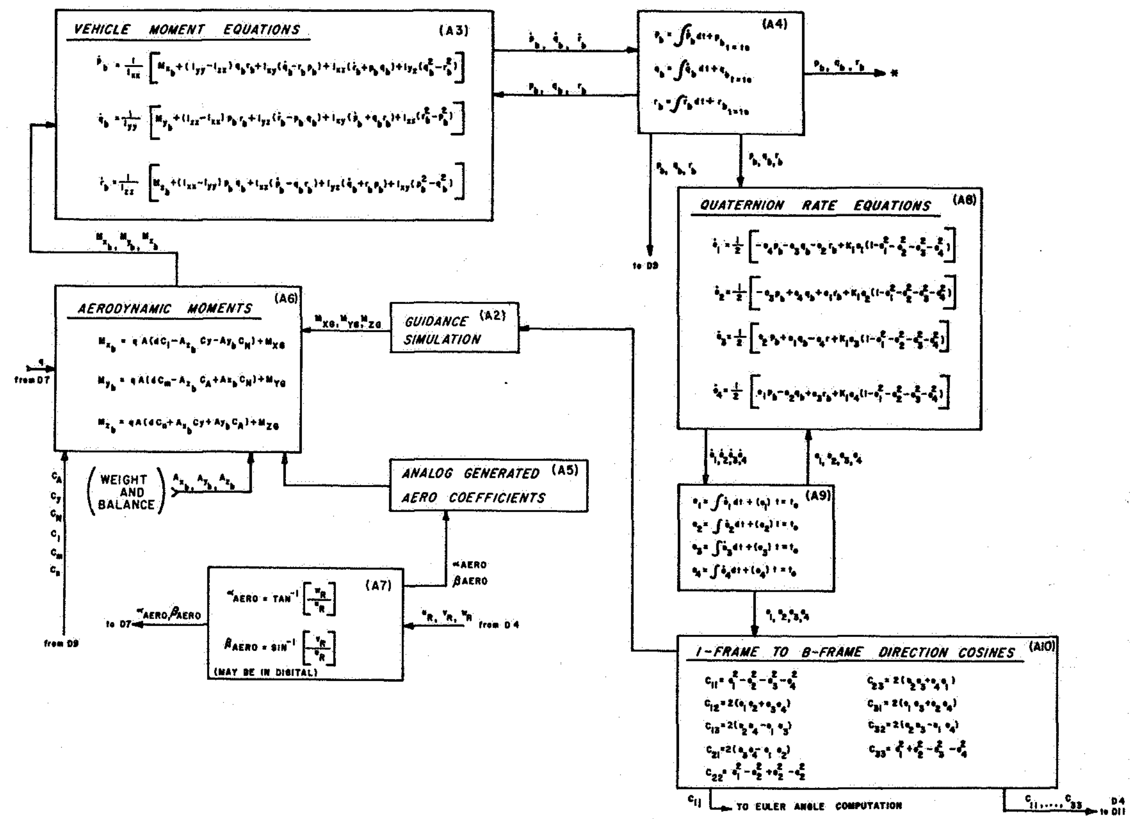


Figure 2. Equations to be solved on the analog computer for a hybrid six-degree-of-freedom space vehicle simulation.

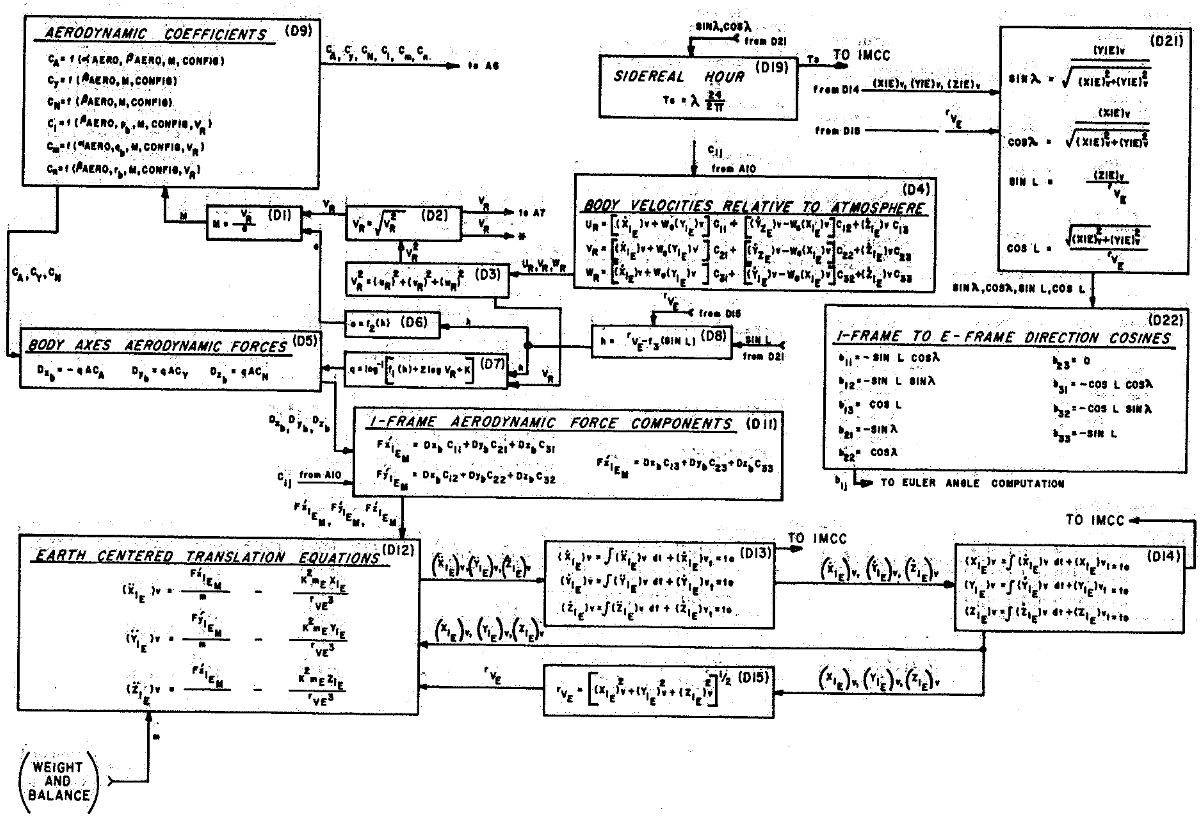
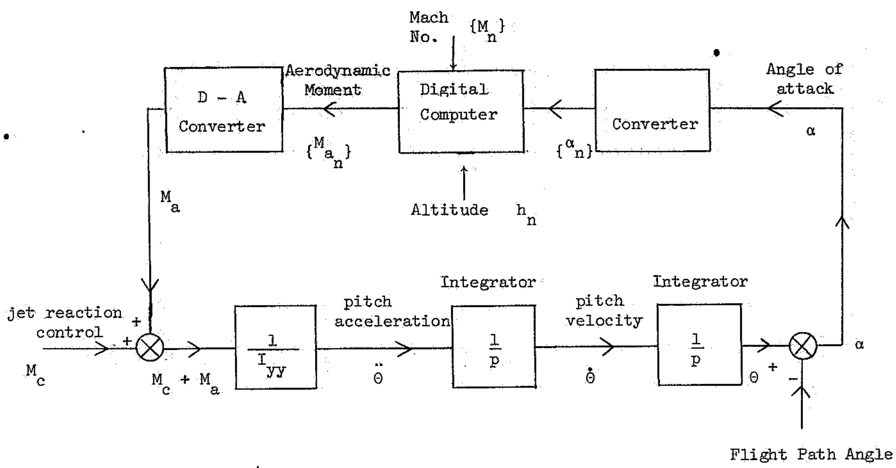


Figure 3. Equations to be solved on the digital computer for a hybrid ; six-degree of freedom simulation.



$$\ddot{\theta} = \frac{M_a + M_c}{I_{yy}} \quad ; \quad \alpha = \theta - \gamma$$

Figure 4. Block diagram of hybrid computing loop for calculating pitch angle θ.

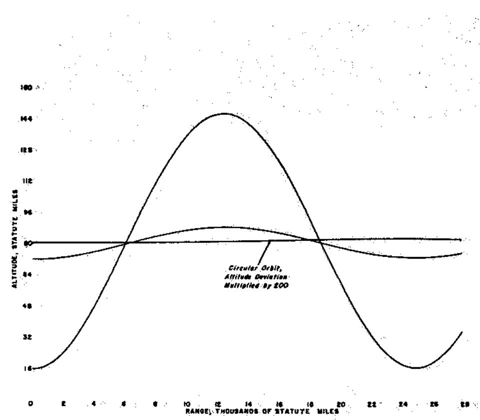


Figure 5. Analog solutions for low-eccentricity orbits.

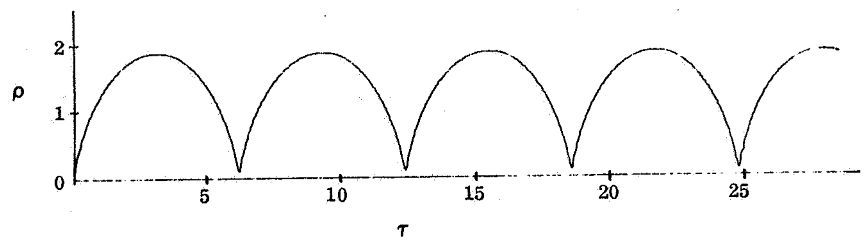


Figure 6. Typical solution for a highly-eccentric orbit.

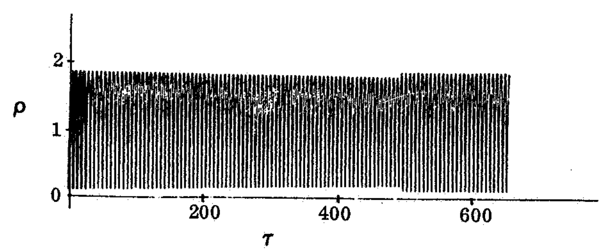


Figure 7. Periodic solution with no energy constraint initially. Energy constraint is added after about 75 cycles.

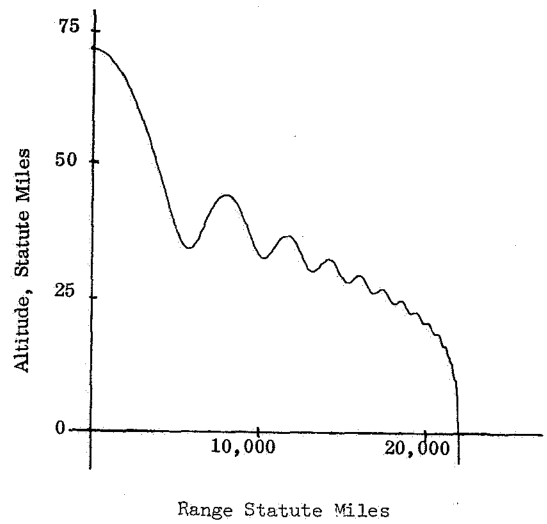


Figure 8. Typical reentry profile, L/D = 2.0

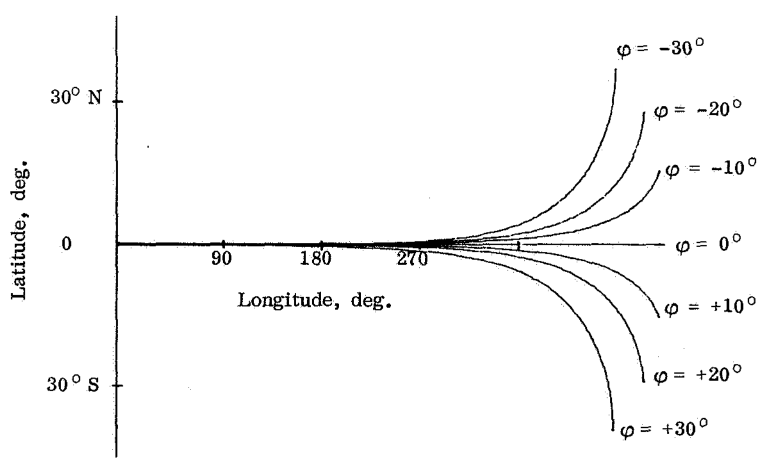


Figure 9. Typical impact footprint, L/D = 2.0