NASA/TM—2015-218753 AIAA–2014–3671

DUKSUP: A Computer Program for High Thrust Launch Vehicle Trajectory Design and Optimization

O. Frank Spurlock and Craig H. Williams
Glenn Research Center, Cleveland, Ohio

May 2015

NASA STI Program . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program plays a key part in helping NASA maintain this important role.

The NASA STI Program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI Program provides access to the NASA Technical Report Server—Registered (NTRS Reg) and NASA Technical Report Server—Public (NTRS) thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counter-part of peer-reviewed formal professional papers, but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., "quick-release" reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Fax your question to the NASA STI Information Desk at 757-864-6500

- Telephone the NASA STI Information Desk at 757-864-9658

- Write to:
  NASA STI Program
  Mail Stop 148
  NASA Langley Research Center
  Hampton, VA 23681-2199

NASA/TM—2015-218753 AIAA–2014–3671

DUKSUP: A Computer Program for High Thrust Launch Vehicle Trajectory Design and Optimization

O. Frank Spurlock and Craig H. Williams
Glenn Research Center, Cleveland, Ohio

Prepared for the
50th Joint Propulsion Conference
cosponsored by the AIAA, ASME, SAE, and ASEE
Cleveland, Ohio, July 28–30, 2014

National Aeronautics and
Space Administration

Glenn Research Center
Cleveland, Ohio 44135

May 2015

Acknowledgments

Mr. Frank Spurlock (first author) is deceased, passing away only 2 months prior to this conference (AIAA 50th Joint Propulsion Conference), and only 3 weeks after completing the initial draft of this paper. Portions of this paper may appear to be written in one of two very different tones: sometimes humble and modest, while other times self-congratulatory to the point of boastful. The former is the product of the first author, the creator of DUKSUP, who had an overly modest and quiet persona. The latter is the product of the second author, someone who had worked for Frank for decades, believes in bestowing credit to those deserving, and felt Frank never got sufficient credit and recognition. That feeling is shared by many who worked for and with him (a few of whom contributed to the final review of this paper). NASA Lewis Research Center (now Glenn) senior management frequently acknowledged his accomplishments and his admirable character. His colleagues at General Dynamics greatly valued his technical skills and intellect. Frank also had many close friends and colleagues in the unmanned payload community, especially JPL. But until the mid-1980s with Shuttle/Centaur, the Centaur program was largely isolated within NASA, totally overshadowed by the manned program. Had Frank Spurlock been located at a Code M NASA Center performing comparable feats, he would have been front and center and a recipient of accolades from NASA Headquarters. He did receive many awards from LeRC, including NASA's prestigious Outstanding Leadership Medal (NASA's prestigious Outstanding Leadership Medal (for identifying, recruiting, and training young engineers); NASA's Certificate of Appreciation in recognition of outstanding contribution to the NASA Lewis Research Center College Relations Program with Howard University; numerous awards for supporting space missions beginning with Surveyor (first U.S. soft landing on the Moon) to Galileo (mission to Jupiter)). He began his career at LeRC in 1962 working in trajectory optimization and mission design for the Atlas/Centaur vehicle. Before retiring from the Center in 1997, he had done mission design and trajectory optimization for not only Atlas/Centaur, but also for Titan/Centaur, Shuttle/Centaur, and wide variety of other vehicles, many of which were never built. He managed a mission design and trajectory optimization section for the launch vehicle project at LeRC as well as managing mission design for specific missions. He supported the launch of the many vehicles that sent payloads to all of the planets (except Pluto), the Moon, and a wide variety of Earth orbits. His last position at LeRC was Deputy Division Chief for the Advanced Space Analysis Office, with primary responsibility for the analytic capabilities and activities of the group. Since retiring in 1997, Mr. Spurlock had consulted for government and commercial companies in the areas of mission design, trajectory optimization, space transportation architecture, and related disciplines. He had served on numerous review teams for launch vehicles, and most recently was a member of the team that considered the options for deflecting near-Earth objects that might impact Earth. Though not covered in this technical paper, Frank had another legacy—his positive impact on the lives of others. He mentored and helped propel the careers of many young engineers, within his group and beyond, and recruited workers from nontraditional sources. He also took a strong interest in the whole person, both at and outside the office, including their families. Frank was a superb role model and supporter of dreams and initiatives of others. Frank Spurlock was renowned at LeRC and within the broader unmanned space launch and spacecraft community for his exceptional technical skills, demonstrated accomplishments, and mentoring of staff. This paper is a modest attempt to secure for him the respect which was never widely bestowed on him—and which he never sought. Though DUKSUP brought him the initial recognition, it was only one of the lasting legacies of his career.

Level of Review: This material has been technically reviewed by technical management.

DUKSUP: A Computer Program for High Thrust Launch Vehicle Trajectory Design and Optimization

O. Frank Spurlock* and Craig H. Williams
National Aeronautics and Space Administration
Glenn Research Center
Cleveland, Ohio 44135

Abstract

From the late 1960s through 1997, the leadership of NASA's Intermediate and Large class unmanned expendable launch vehicle projects resided at the NASA Lewis (now Glenn) Research Center (LeRC). One of LeRC's primary responsibilities—trajectory design and performance analysis—was accomplished by an internally-developed analytic three dimensional computer program called DUKSUP. Because of its Calculus of Variations-based optimization routine, this code was generally more capable of finding optimal solutions than its contemporaries. A derivation of optimal control using the Calculus of Variations is summarized including transversality, intermediate, and final conditions. The two point boundary value problem is explained. A brief summary of the code's operation is provided, including iteration via the Newton-Raphson scheme and integration of variational and motion equations via a 4th order Runge-Kutta scheme. Main subroutines are discussed. The history of the LeRC trajectory design efforts in the early 1960s is explained within the context of supporting the Centaur upper stage program. How the code was constructed based on the operation of the Atlas/Centaur launch vehicle, the limits of the computers of that era, the limits of the computer programming languages, and the missions it supported are discussed. The vehicles DUKSUP supported (Atlas/Centaur, Titan/Centaur, and Shuttle/Centaur) are briefly described. The types of missions, including Earth orbital and interplanetary, are described. The roles of flight constraints and their impact on launch operations are detailed (such as jettisoning hardware on heating, Range Safety, ground station tracking, and elliptical parking orbits). The computer main frames on which the code was hosted are described. The applications of the code are detailed, including independent check of contractor analysis, benchmarking, leading edge analysis, and vehicle performance improvement assessments. Several of DUKSUP's many major impacts on launches are discussed including Intelsat, Voyager, Pioneer Venus, HEAO, Galileo, and Cassini.

Nomenclature

C	first integral of Euler-Lagrange equations
E	energy per unit mass
F	functional defined by Equation (3)
G	spherical Earth gravity constant
J	functional to be minimized
N	total number of stages
S	variational switching function
T	thrust
e	eccentricity
f	thrust direction
g	gravity acceleration
h	angular momentum per unit mass
m	mass
r	radius
t	time
v	velocity
x	state variable
z	vector pointing at north pole
β	mass flow rate
ε	jump factor
η	fourth Lagrange multiplier
λ	first Lagrange multiplier
μ	second Lagrange multiplier
σ	third Lagrange multiplier

Superscripts

0	initial
f	final
$\dot{}$	time derivative
$\hat{}$	unit vector

Subscripts

0	initial
f	final
i,j,k,l,m,n	stage numbers
p	perigee
1, 2	components of oblate gravity acceleration

* Retired.

1.0 Introduction

This paper is a history of the rationale for, the development of, and the use of, a trajectory analysis code called DUKSUP. It should be noted that it was very much a product of its time—over 50 years ago. While there is limited utility in describing it in detail since at this point it is of historical interest largely to students of trajectory analysis and optimization. Thus, this paper is not a "Users Guide". There is value, however, in documenting how such a product of largely one individual had such a profound and sustained impact on so many of this nation's launches of spacecraft. Though largely unknown to those outside of the launch vehicle trajectory design and performance optimization community, it was relied upon by NASA management to design the trajectories for most of the Atlas/Centaur, Titan/Centaur, and (anticipated) Shuttle/Centaur missions from the late 1960s until 1997 with the launch of the Cassini mission to Saturn. Originally written in FORTRAN IV, the only technical computer language available at that time, DUKSUP retained that language's coding through its lifetime.

Because of its perceived value to the history of the NASA Lewis Research Center (LeRC) (now Glenn Research Center (GRC)), management included a hardcopy FORTRAN listing of the source code in the time capsule in front of the NASA GRC Administration Building 3; slated for opening in the year 2041, 50 years after its sealing.

2.0 Background

The Atlas/Centaur rocket was transferred to LeRC in Cleveland, Ohio, in the fall of 1962 following the failure of its maiden flight (designated "F-1") managed by the Marshall Space Flight Center (MSFC) in Huntsville, Alabama. This took place within the backdrop of President Kennedy's 1961 commitment to the United States landing a man on the Moon by the end of the decade. Werner von Braun and the MSFC were focused largely on responding to the challenge of building the launch vehicle to carry the crew to the Moon (Ref. 1). The responsibility for solving Atlas/Centaur problems was then transferred to LeRC, due in large part to the technical expertise and determination of Dr. Abe Silverstein. Centaur became known as "Abe's Baby". The sole mission of the Atlas/Centaur at that time was to get a soft lander (Surveyor) to the Moon prior to the attempt to land humans on the Moon. A project office was established at LeRC to manage the development of the vehicle. Within 14 months of the transfer of the Centaur program to LeRC, the problems were rectified and the first successful launch of an Atlas/Centaur launch vehicle (LeRC's first attempt to launch an Atlas/Centaur), designated "AC-2," occurred on November 27, 1963 (Figure 1).

Figure 1.—Launch of Atlas/Centaur-2

The LeRC was the Power, Propulsion, and Communication technology research center for NASA. LeRC did not have experience with large projects such as this and had to find the staff mostly from research groups to meet the challenge. In anticipation that the Russians would be first to land humans on the Moon due to the perceived lead that they had in the "race," LeRC was doing the research on the propulsion for the vehicles to get to Mars. In 1961, a branch was created and staffed to provide the analytic resources to "guide" the research. Among the technologies being developed were a large hydrogen/oxygen engine, a large solid rocket motor, and a nuclear thermal rocket. Electric propulsion had been invented at LeRC and that research also continued. The analytic branch consisted of several sections that included thermal, structural, nuclear, and performance analysis capability (later control analysis was added). This analytic capability was in place or under development when the responsibility for the Atlas/Centaur was transferred to LeRC. As stated earlier, the research divisions were "raided" to staff the new project office. Significant parts of the analytic capability were transferred to the project office to support the challenging schedule of landing Surveyor on the Moon prior to the human landing attempt. At that point, the performance analysis capability was not as mature as the other disciplines and was left in the Propulsion Research Division until sometime later.

Initially, performance analysis for the Atlas/Centaur was done by the contractor, General Dynamics. For all practical purposes, LeRC had no capability to do vehicle performance analysis and it had to be developed from codes developed for other purposes, such as low thrust trajectory analysis for electric propulsion missions to Mars and an N-Body code available at LeRC (Ref. 2). We[1] scrambled to develop vehicle performance capability at LeRC. The analysis for the Surveyor mission was done virtually entirely by the contractor using capability developed to support the Atlas ICBM program. They had analyzed the Surveyor mission with the Centaur for several years.

By the late 1960s, LeRC had developed the most sophisticated mission analysis and mission design capability in NASA and perhaps in the United States. Obviously, we did not have access to what the classified world's capabilities were. We were able to model the Centaur vehicles with such precision that post flight analysis verified performance capability to such accuracy that we could launch with minimal reserves. The trajectory designs were such that we could maximize the payload weights available to the spacecraft to the desired orbits. This was particularly true for geostationary orbit (GSO) missions where payload translated to profit or useful time on station.

For the missions managed by LeRC, there evolved a philosophy and capability committed to extracting all the potential payload mass available from flying these optimum trajectories. In addition to optimizing the trajectories, constraints, such as instantaneous heating, were explicitly incorporated into the optimization. Applying constraints optimally can lower technical risk while minimizing the impact on payload capability. Fundamentally, this paper documents the history of how the LeRC's launch vehicle projects designed and flew the trajectories such that all the payload capability available from the vehicle was made available to the mission.

3.0 Derivation of Optimum Control: the Calculus of Variations

The following section outlines general mathematical theory on which DUKSUP was based. The Calculus of Variations (CoV) derivation as applied to this problem was previously published by the lead author (Ref. 3). The appendix to that paper is included here for completeness. That derivation was the basis for the code. As new missions were added to the Atlas/Centaur manifest, the CoV analysis was extended to incorporate new requirements.

3.1 Why Calculus of Variations?

The decision to take advantage of all of the potential payload capability obtainable from trajectory optimization for operational missions developed from the circumstances that existed when the Atlas/Centaur project came to LeRC in 1962. During the late fifties, LeRC was developing electric propulsion as a possible propulsion system for Mars missions. To obtain data to make informed technology decisions, the trajectory analysts used the CoV to construct a simple code. The computer available at that time was an IBM 704, which had 8,000 36 bit words. Formulating a CoV solution was the only analytic technique that would obtain the data required and fit. In 1961 President Kennedy made the decision that the United States would beat the Russians to the Moon with humans. Had the Russians beaten us to the Moon, the U.S. planned to beat them to Mars. Consequently, two things happened at LeRC that affected analytic capability: (1) The Atlas/Centaur Project came to LeRC to support the Surveyor project to soft land robotic spacecraft on the Moon as precursors to human landings, and (2) Research on very large propulsion systems was being conducted in case it were needed to build the launch vehicles to send humans to Mars. The simplest analytic technique for such problems is to use methods known as steepest descent to maximize the payload to the desired state. That approach solved the optimization problem by iterating on initial conditions to maximize the payload consistent with the desired end conditions. But the computers available in 1962 were too small and too slow to implement such a technique. Of necessity, the analysts responded by developing CoV launch vehicle trajectory codes to support the LeRC technology groups working on large engines. Different CoV codes were developed to support the Atlas/Centaur project, whose only mission at that time was Surveyor. There were no commercially available codes; all had to be developed in-house. Serendipitously, being forced to use the CoV taught the analysts a great deal about the nature of optimum solutions for launch vehicle trajectory problems. Such insight is not readily available from steepest descent techniques. The CoV theory showed that optimum solutions maximizing payload are characterized by continuous thrust vectors, unless a constraint is imposed, in which case the thrust vectors are discontinuous. Initially, analysts had not learned how to impose constraints. It soon became apparent that the CoV technique yielded the maximum payload to orbit to within about one hundredth of a pound. This was verified by obtaining parametric data. We also learned the level of fidelity of vehicle models required to predict payload accurately. A fairly simple, three dimensional (3D) model was all that was required to accurately predict payload, and to design the trajectories that would yield

[1] The primary author uses "we" to indicate the core group of LeRC launch vehicle mission designers (and supporting management personnel) at the time of the event described.

that payload. Very detailed post-flight trajectory reconstructions of early Atlas/Centaur flights verified the results of the simplified modeling and the optimization.

3.2 Problem Setup

The optimization of a trajectory to a circular synchronous equatorial orbit may be considered as the problem of optimizing a multistage launch vehicle to a particular final orbit. The optimization problem to be considered here begins at booster jettison, which is assumed to be a fixed position and velocity. The sustainer portion of the Atlas continues until propellant depletion. The sustainer is jettisoned and a few seconds later the first Centaur burn begins. Its duration is variable and must be optimized. The perigee radius of the parking orbit is fixed. The duration of the parking orbit coast may or may not be optimized. The parking orbit is not a true coast since a small acceleration is maintained for propellant retention. The duration and direction of the second Centaur burn must be optimized, followed by an optimum transfer orbit coast (a true coast), and a final burn of fixed total impulse. The analysis presented here to solve this problem is a special case of the analysis derived in a prior work by the author (Ref. 4), with an additional constraint—parking orbit perigee radius.

The variational problem to be solved is to find the steering program and various stage durations which maximize the payload capability of a multistage launch vehicle to a specified final orbit. The trajectory must satisfy certain initial, final, and intermediate conditions on the state variables. The thrust, propellant flow rate, and jettison weight for each stage are assumed to be constant. The equations of motion and constraints for each stage may be written as

$$\dot{\mathbf{v}} - \mathbf{g}(\mathbf{r}) - T_i \hat{f} / m = 0 \tag{1a}$$

$$\dot{\mathbf{r}} - \mathbf{v} = 0 \tag{1b}$$

$$\dot{m} + \beta_i = 0 \tag{1c}$$

$$\hat{f} \cdot \hat{f} - 1 = 0 \tag{1d}$$

where $\hat{f}$ is the unit thrust direction and $\mathbf{g}(\mathbf{r})$ is the oblate Earth gravity acceleration, which may also be written

$$\mathbf{g}(\mathbf{r}) = g_1(r, \mathbf{r}\cdot\hat{z})\hat{r} + g_2(r, \mathbf{r}\cdot\hat{z})\hat{z} \tag{2}$$

Suppose that each stage of the vehicle is numbered consecutively starting with the booster. For analysis purposes a stage change occurs when the thrust and/or propellant flow rate changes and/or a mass is jettisoned. A Bolza formulation of the variational problem is used, and the functional to be minimized is written in an earlier source (Ref. 4) as

$$J = -m_f + \sum_{i=2}^{N} \int_{t_{i-1}}^{t_i} F_i \, dt \tag{3}$$

where the functional F_i for each stage is

$$\begin{aligned} F_i = {} & \boldsymbol{\lambda} \cdot \left[\dot{\mathbf{v}} - \mathbf{g} - T_i \hat{f} / m\right] + \boldsymbol{\mu} \cdot \left[\dot{\mathbf{r}} - \mathbf{v}\right] \\ & + \sigma(\dot{m} + \beta_i) + \eta\left(\hat{f} \cdot \hat{f} - 1\right) \end{aligned} \tag{4}$$

The resulting Euler-Lagrange equations are

$$\dot{\boldsymbol{\lambda}} + \boldsymbol{\mu} = \mathbf{0} \tag{5a}$$

$$\dot{\boldsymbol{\mu}} + g_1 \boldsymbol{\lambda}/r + (\boldsymbol{\lambda}\cdot\hat{r})\nabla_{\mathbf{r}} g_1 - g_1(\boldsymbol{\lambda}\cdot\hat{r})\hat{r}/r + (\boldsymbol{\lambda}\cdot\hat{z})\nabla_{\mathbf{r}} g_2 = 0 \tag{5b}$$

$$\dot{\sigma} - (T_i / m^2)\boldsymbol{\lambda}\cdot\hat{f} = 0 \tag{5c}$$

$$2\eta\hat{f} - (T_i/m)\boldsymbol{\lambda} = \mathbf{0} \tag{5d}$$

The optimum thrust direction $\hat{f}$ is obtained by combining Equations (1d) and (5d) and using the Weierstrass E-test. This procedure results in

$$\hat{f} = \hat{\lambda} \tag{6}$$

Since F does not explicitly depend on time, an integral of the motion is

$$C + \boldsymbol{\lambda}\cdot\mathbf{g} + \boldsymbol{\mu}\cdot\mathbf{v} + (T_i/m)\lambda - \sigma\beta_i = 0 \tag{7}$$

When a spherical gravity model is assumed (i.e., $\mathbf{g}(\mathbf{r}) = G/r^3\,\hat{\mathrm{r}}$), three additional integrals of the motion exist which are given by $\boldsymbol{\lambda}\times\mathbf{v} + \boldsymbol{\mu}\times\mathbf{r} = \text{const}$. Since, $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, $\mathbf{r}$, and $\mathbf{v}$ are all continuous except where an intermediate boundary condition is imposed (as will be shown later), the three integrals are constant across staging points where continuity holds. However for the oblate gravity model used in this analysis, only a single component of the previous vector integral is constant, as can be verified by differentiation with respect to time.

$$(\boldsymbol{\lambda}\times\mathbf{v} + \boldsymbol{\mu}\times\mathbf{r})\cdot\hat{z} = \text{const} \tag{8}$$

3.3 Transversality Equation

The transversality equation for this problem is

$$dJ = \sum_{i=2}^{N} \left(C\,dt + \boldsymbol{\lambda}\cdot d\mathbf{v} + \boldsymbol{\mu}\cdot d\mathbf{r} + \sigma\,dm\right)_{t_{i-1}}^{t_i} - dm_f \tag{9}$$

which is set equal to zero for an optimal solution. An earlier source (Ref. 4) shows that:

- $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are continuous everywhere if there are no intermediate boundary conditions. If the intermediate boundary condition (assumed to occur at a staging point) is expressed as

$$r_p - r_{p,\text{desired}} = 0 \tag{10}$$

  A previous source (Ref. 5) shows that the discontinuities in $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are $\varepsilon\nabla_{\mathbf{v}} r_p$ and $\varepsilon\nabla_{\mathbf{r}} r_p$, respectively. The variable ε is used as an initial condition in the two-point boundary value problem to satisfy the intermediate boundary condition Equation (10).
- The equations that must be satisfied to optimize the duration of the powered and coast stages are derived in a previous source (Ref. 4), and the applicable results are presented here. Let j be the first optimized powered stage. Then for constant jettison weight the equation for optimizing stage l is

$$\sum_{i=j}^{l-1}\left(S_i{}^f - S_{i+1}{}^0\right) = 0 \tag{11}$$

  where 0 and f refer to initial and final values and the S functions are defined as

$$S_i = C/\beta_i - \sigma = -\left[\boldsymbol{\lambda}\cdot\mathbf{g} + \boldsymbol{\mu}\cdot\mathbf{v} + (T_i/m)\lambda\right](1/\beta_i) \tag{12a}$$

$$S_i \equiv 0,\ \beta_i = 0 \tag{12b}$$

  where the right side of Equation (12a) is obtained by using Equation (7) (note: $\beta_i \neq 0$). For coasting stages ($\beta_i = T_i = 0$) to be optimized, the equation

$$C_i = \left(\boldsymbol{\lambda}\cdot\mathbf{g} + \boldsymbol{\mu}\cdot\mathbf{v}\right) = 0 \tag{13}$$

  must be satisfied for maximum payload.
- For free initial or final state variable x, the required or final condition for maximum payload is given from a previous publication as (Ref. 6)

$$\boldsymbol{\lambda}\cdot\left(\partial\mathbf{v}/\partial x\right) + \mathbf{u}\cdot\left(\partial\mathbf{r}/\partial x\right) = 0 \tag{14}$$

If the initial position and velocity are specified, the initial values of any five of the six $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ and may be used as variable initial conditions in order to satisfy the required final conditions of the two-point boundary value problem. In order to eliminate the difficulty associated with guessing at values of the multipliers, the values of $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ may be used as variable initial conditions in order to satisfy the required final conditions of the two-point boundary value problem. In order to eliminate the difficulty associated with guessing at values of the multipliers, the values of $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ can be expressed in terms of pitch and yaw attitudes and their time rates of change. These equations may be found in a previous publication (Ref. 6). The values of $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are then calculated from:

$$\boldsymbol{\lambda} = \lambda\hat{\lambda} \tag{15a}$$

$$\boldsymbol{\mu} = -\lambda\hat{\lambda} - \dot{\lambda}\hat{\lambda} \tag{15b}$$

The value of λ can be set equal to unity without loss of generality. The initial value of $\dot{\lambda}$ can be calculated in closed form, as will be shown by the following development.

3.4 Final Conditions

Final conditions for both the conventional and unconventional synchronous equatorial orbit mission require a circular orbit at synchronous orbit altitude with prescribed inclination. If the required inclination is nonzero, both the longitude of the ascending node and the injection point in the final orbit are free for optimization. As shown in a previous publication (Ref. 6), the corresponding auxiliary variational final conditions are

$$\left(\boldsymbol{\lambda}\times\mathbf{v} + \boldsymbol{\mu}\times\mathbf{r}\right)\cdot\hat{z} = 0 \tag{16a}$$

and

$$\left(\boldsymbol{\lambda}\times\mathbf{v} + \boldsymbol{\mu}\times\mathbf{r}\right)\cdot\left(\mathbf{r}\times\mathbf{v}\right) = \mathbf{0} \tag{16b}$$

If the desired inclination is zero, Equations (16a) and (16b) degenerate into one equation (zero inclination is equivalent to two final conditions, $\mathbf{r}\cdot\hat{z} = 0$ and $\mathbf{v}\cdot\hat{z} = 0$, and only Equation (16a) must be satisfied.

Since Equation (16a) is a constant of the motion Equation (8), it may be satisfied at the beginning of the trajectory, and used to calculate $\dot{\lambda}$. However, it must first be verified that jump discontinuities in $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ at the intermediate boundary point do not change the value of the constant. This requires that

$$\left(\nabla_{\mathbf{v}} r_p \times \mathbf{v} + \nabla_{\mathbf{r}} r_p \times \mathbf{r}\right)\cdot\hat{z} = \mathbf{0} \tag{17}$$

It will be shown later than Eq. (17) is satisfied.

The calculation of $\dot{\lambda}$ proceeds as follows:

$$\left(\boldsymbol{\lambda}\times\mathbf{v} + \boldsymbol{\mu}\times\mathbf{r}\right)\cdot\hat{z} = \left(\lambda\hat{\lambda}\times\mathbf{v}\right)\cdot\hat{z} - \left(\lambda\dot{\lambda}\times\mathbf{r}\right)\cdot\hat{z} - \left(\dot{\lambda}\hat{\lambda}\times\mathbf{r}\right)\cdot\hat{z} = 0 \tag{18a}$$

$$\dot{\lambda} = \lambda\left(\hat{\lambda}\times\mathbf{v} - \dot{\hat{\lambda}}\times\mathbf{r}\right)\cdot\hat{z}\,/\left[\left(\hat{\lambda}\times\mathbf{r}\right)\cdot z\right] \tag{18b}$$

Computing $\dot{\lambda}$ with Equation (18b) guarantees that Equation (16a) will be satisfied.

3.5 Intermediate Conditions

As explained earlier, it is necessary to constrain the perigee radius at injection into the first parking orbit. Otherwise, the optimum solution would result in the parking orbit injection and/or the equator crossing occurring at very low altitudes, thus, violating spacecraft heating constraints. Therefore, the intermediate constraint is given by Equation (10), where the desired value corresponds to the perigee altitude.

The required gradients are calculated to be (where e is the orbital eccentricity):

$$\nabla_{\mathbf{v}} r_p = h\left(\hat{h}\times\mathbf{r} - r_p^2\mathbf{v}\right)\Big/ eG \tag{19a}$$

$$\nabla_{\mathbf{r}} r_p = \left[(h/G)\left(\mathbf{v}\times\hat{h}\right) - \mathbf{r}\left(r_p^2/r^2\right)\right](1/e) \tag{19b}$$

It is easily shown that Equation (17) is satisfied for the gradients in Equations (19a) and (19b). Hence the value of $\boldsymbol{\lambda}\times\mathbf{v} + \boldsymbol{\mu}\times\mathbf{r}$ is unaffected by the jump in $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$.

3.6 Boundary Value Problem

For the GSO missions, both fixed and optimum parking orbit coast time were considered. The transfer orbit coast time was always optimized, however, along with the durations of the first and second Centaur burns. Based on the preceding discussion of the transversality equation, the initial and final conditions for the two point boundary value problem (2PBVP) are shown in Table 1 for the case where the parking orbit coast time was optimized.

If the desired final inclination is nonzero, then $(\mathbf{r}\cdot\hat{z})$ and $(\mathbf{v}\cdot z') = 0$ are replaced by i_{desired} and $(\boldsymbol{\mu}\times\mathbf{r} + \boldsymbol{\lambda}\times\mathbf{v})\cdot(\mathbf{r}\times\mathbf{v}) = 0$. If the parking orbit coast time is fixed, then an initial and final condition is removed. These are t_k and

$$\sum_{i=j}^{k-1}\left(S_i^f - S_{i+1}{}^0\right) = 0 \tag{20}$$

It should be recognized that there may be any number of fixed stages between t_j and t_k, etc. Also, the last three final conditions are evaluated at intermediate points in the trajectory.

4.0 Two Point Boundary Value Problem

The following technique was devised to systematically proceed from a simple, easily converged problem to the solution of the two point boundary value problem (2PBVP) for a circular synchronous equatorial orbit.

TABLE 1.—INITIAL AND FINAL CONDITIONS FOR 2PBVP

Initial conditions	Final conditions
Pitch attitude	Energy/unit mass
Pitch attitude	$\mathbf{r}_{\text{desired}}$
Yaw attitude	$\dot{\mathbf{r}}_{\text{desired}}$
Yaw attitude rate	$\mathbf{r}_{p,\text{desired}}$
ε (jump factor)	$(\mathbf{r}\bullet\hat{\mathbf{z}}) = 0$
t_j (1st Centaur burn)	$(\mathbf{v}\bullet\hat{\mathbf{z}}) = 0$
t_k (park orbit coast)	$\sum_{i=j}^{k-1}\left(S_i^f - S_{i+1}{}^0\right) = 0$
t_l (2nd Centaur burn)	$\sum_{i=j}^{l-1}\left(S_i^f - S_{i+1}{}^0\right) = 0$
t_m (transfer orbit coast)	$C_i = (\lambda\cdot g + \mu\cdot v) = 0$

A trajectory is obtained to a slightly elliptical (parking) orbit with the desired perigee radius without plane change with a 90° launch azimuth. This problem converges easily. Then the ascent burn time is fixed at the value obtained and a variable length parking orbit coast, a fixed parking orbit perigee radius and second burn are added. This problem is targeted to the desired apogee and 180° argument of perigee for first equator crossing second burn. An inclination decrease of about 2° is then added to these final conditions and the problem is retargeted to the augmented final conditions. Now the transfer orbit coast (variable) and apogee burn (fixed or variable) are added. This trajectory is integrated to the end with the converged initial guesses from the last step. The final conditions achieved will frequently be far from a circular synchronous equatorial orbit. However, specify the final conditions actually achieved as the desired ones, and optimize the problem. The parking orbit coast, second burn, and transfer orbit coast durations will change. Now alter the achieved final conditions toward the desired ones judiciously in steps, retargeting at each step. In this manner, the desired final orbit conditions may be obtained. Now the ascent burn duration may be optimized. Any sizable change in a constraint or final condition is best achieved by proceeding in steps. The problem is quite nonlinear. Attempts to plot initial conditions as functions of the final conditions for extrapolation purposes were made. They were generally unsuccessful.

The most challenging aspect was solving the two point boundary value problem. In the worst case, there were 18 initial conditions to satisfy 18 final conditions. The problem was highly nonlinear and to achieve convergence, very accurate derivatives were necessary. We never encountered problems that we couldn't converge, but "sneaking" up on the solutions was mandatory. Experience was of great value in formulating a strategy for achieving convergence. It usually involved converging a problem that

was "easy" and adding complexity to the problem and proceeding step-by-sometimes small step to get a solution. If a converged solution to a similar problem were available, it was possible to "move" in small steps to converge the new problem. As faster and faster machines became available, these techniques and/or strategies became less of a handicap. However, when we had the slower machines, we achieved convergence, but the computer time required reduced productivity.

However, at the beginning with the small and slow machine we had to work with, the CoV/two point value problem technique was faster and the machine size limitation less onerous. Steepest descent techniques were slower and more of a problem with the small computer. A series type technique very quickly swamped the machine.

A problem with any technique is avoiding local maxima. The larger the number of independent variables, the probability of settling on a local maximum is higher. This is probably much higher with a steepest descent technique because the hyper-space is so large and complex. The CoV has many fewer initial variables and the hyper-space is not nearly so complex. Local maxima are obtainable with the CoV; in fact, it is not hard to select initial conditions which lead to those locals. Since the problems are so hard to converge and good initial "guesses" so important to have any chance of convergence, locals are not usually a problem. A few cases were encountered where a local was so close to a global that the problem converged to the local. These can usually be identified by examining the solution and from experience, deducing that it is, in fact, a local that has been identified. And usually, the payload capabilities are only a few pounds different. If the solution was obtained by "sneaking up" on the solution and if the initial conditions in the "sneak" attack are plotted, often there is a discontinuity which suggests that a local optimum has been encountered. In "esoteric" problems (such as the HEAO trajectory, Sec. 11.7) it was important to plot the evolving trajectory characteristics and to "reason" that from energy considerations, the solution was almost certainly a global. It was not unusual to have the CoV "discover" a solution that was counterintuitive. After hand plotting trajectory characteristics and the variation in initial conditions, it was possible to understand the "tradeoffs" the calculus had made.

5.0 DUKSUP

5.1 Early NASA LeRC Trajectory Design Code Development

LeRC had never analyzed launch trajectories prior to the transfer of the Centaur Program. It had, however, studied interplanetary trajectories. Art Zimmerman had developed a computer code in polar coordinates based on the calculus of variations to study low-thrust electric propulsion research options (Ref. 7). That code gave us our first exposure to trajectory optimization using the calculus of variations. By replacing the gravitational constant of the Sun with that of the Earth, we could use the code to analyze launch trajectories after the vehicle had left the atmosphere. We had no code to simulate the atmospheric portion of the trajectory. Due to bending loads and atmospheric heating, launch vehicle trajectories are near zero-angle-of-attack in critical periods until the vehicle leaves the atmosphere. Until we could develop a digital capability, we simulated the atmospheric portion of a trajectory on an analog computer and fed the state vector from that part of the "flight" into the simple jury rigged version of the available low thrust code. While using this "primitive" technique temporarily, we were taking parts of the N-Body code (Ref. 2), replacing the gravitational constant of the Sun with that of the Earth, and cobbling together a digital code to "fly" the atmospheric portion of the trajectory. We then took the state vector from the digital replacement and manually input it into the low-thrust code. This was a cumbersome at best, temporary solution to analyzing the performance of multi-stage vehicles. The goal was to be able to maximize the payload (not the burn-out weight) capability of four or five stage vehicles where the last stage could be nuclear thermal. In the 1960s, it was assumed that we could use a nuclear thermal rocket in the atmosphere—so the last stage was sometimes nuclear thermal. We also assumed that we could "relight" the last stage after a coast to put it on an interplanetary trajectory, usually to Mars.

To perform the analyses to support the rapidly expanding needs of the Propulsion Research Division, a new code was needed to quickly assess the payload impact of technology options for engines, both liquid and solid. Still using the polar coordinate calculus of variations formulation of the problem, Fred Teren derived the equations for optimizing five stages where the jettisoned stage weight of each stage was a linear function of the propellant "burned" by that stage. That included the last stage, where the payload was the burnout weight minus the stage weight. This code was developed, documented, and provided to the community prior to the development of DUKSUP. Its lessons learned heavily influenced the design of DUKSUP. One of the challenges with the earlier code was the difficulty in converging the two-point boundary value problem (see Sec. 4.0) inherent in the calculus of variations formulation. The partial derivatives required were obtained by perturbing each of the initial conditions and obtaining the differences between the perturbed cases and the nominal. It was soon apparent that the perturbation size had an enormous effect on the "accuracy" of the partial derivatives (really secants). A scheme was incorporated which assessed "accuracy" and adjusted the perturbation size to obtain partial derivatives that were sufficiently accurate to permit convergence. This scheme worked but was clumsy and somewhat arbitrary. But obtaining analytical partials was initially thought impractical and would probably make the code so large it could not fit on core. Segmenting the code was possible, but would make it so slow that it would approach the unusable.

From the experience Fred Teren and I gained from our initial development of codes designed to support the Propulsion Research Division, we knew that we had to obtain analytic partial

derivatives or converging a large two point boundary problem would prove very difficult if not impossible. This challenge was twofold. Derivation of the equations for calculating these derivatives was necessary (they had to be right) and incorporating them into a code (correctly) was the second challenge. We also could not be sure that all of this new code would "fit" onto our small machine until we tried it, because we could not be sure of how the compiler would proceed. In addition, these derivatives had to be checked for correctness, otherwise the solution of the two point boundary value problem would difficult if not impossible. These were checked for calculating "partials" by varying initial conditions slightly and calculating the resulting "secant" partials. This was a laborious and time consuming process as there were inevitable errors both in the derivations and the coding. But the errors were found and corrected.

5.2 DUKSUP: A New Code for a New Centaur Mission

When the Atlas/Centaur was assigned to LeRC in the fall of 1962, it had only the one mission assigned to it—the Surveyor mission to soft land the first U.S. spacecraft on the Moon in support of the human landing to follow. These launches took place between June 1966 and January 1968. Centaur's manufacturer General Dynamics did the trajectory analysis for the Surveyor missions, as LeRC's capability was not yet in place. General Dynamics had been performing trajectory design supporting the Centaur program from the start, after having done suborbital ballistic analysis for the Atlas as far back as the inception of that program in 1955. But by the mid-1960s a new class of missions were being planned for Atlas/Centaur. These were GSO missions, characterized by two coast periods—a relatively short one (~ 15 min) to the equator and the other longer one (~ 5¼ hr) from low Earth orbit (LEO) to geostationary altitude—punctuated by two Centaur burns. The trajectories were not planar, so a 3D formulation of the problem was required. They were to be the first missions that LeRC analyzed, and they were more complex, and of much longer duration than the lunar missions or any other interplanetary launch. DUKSUP was developed to perform the trajectory design and performance analysis for those missions.

These missions were initially handled by LeRC and General Dynamics as two separate optimization problems. The first target was to get the Centaur and payload into a parking orbit, 90 nmi altitude circular, followed by a coast, then a second target to a transfer coast to GSO. From this, the change in velocity (ΔV) to injection into the final orbit could be calculated and the size of the spacecraft's solid motor determined. This was a laborious, awkward process, obviously not optimum. That is to say, more payload could be obtained if the trajectory could be optimized from liftoff to injection into the final GSO. It was also obvious to us that optimizing the launch trajectory (that is, getting all of the payload capability from the launch vehicle) was the cheapest and most reliable means of increasing payload

Figure 2.—Launch of Advanced Technology Satellite (AC-17)

over the piecemeal technique which was currently in use. The challenge was to develop a code that could maximize payload to a desired GSO within less than 1 lb. This goal was suggested by the economics of geostationary spacecraft. Adding as much as 5 lb could significantly increase the life time of the spacecraft, thus generating significantly more revenue.

One of the first of these new, other-than lunar missions in the mid-1960s for Atlas/Centaur was the Advanced Technology Satellite (ATS), first launched in August 1968 (see Figure 2 AC-17 launch). ATS was a GSO mission that had "out-grown" its original launch vehicle. Before new trajectory tools could be constructed and after closed-form approximate solutions were obtained, we were forced to try to maximize the payload "piecemeal," i.e., optimizing parts of the trajectory sequentially, and then "hooking" them together. We then iterated "manually" to try to find the maximum payload for a mission profile which required two ignitions of the Centaur main engines, followed by a burn of the spacecraft solid motor (Ref. 8). There were two coast periods, with the first about 20 min between the Centaur burns (to near the equator), and the second after the Centaur second burn. The spacecraft then separated from Centaur and coasted for a 5½ hr period to geostationary altitude, culminating in the burn of the spacecraft solid motor at apogee. This motor, integral to the spacecraft, had fixed total impulse to circularize the orbit and reduce orbit inclination to near-zero. The Centaur burns were primarily in-plane, while the circularization burn an apogee of the transfer orbit performed most of the plane change. Figure 3 and Figure 4 illustrate the in-plane and out-of-plane views of the geostationary mission.

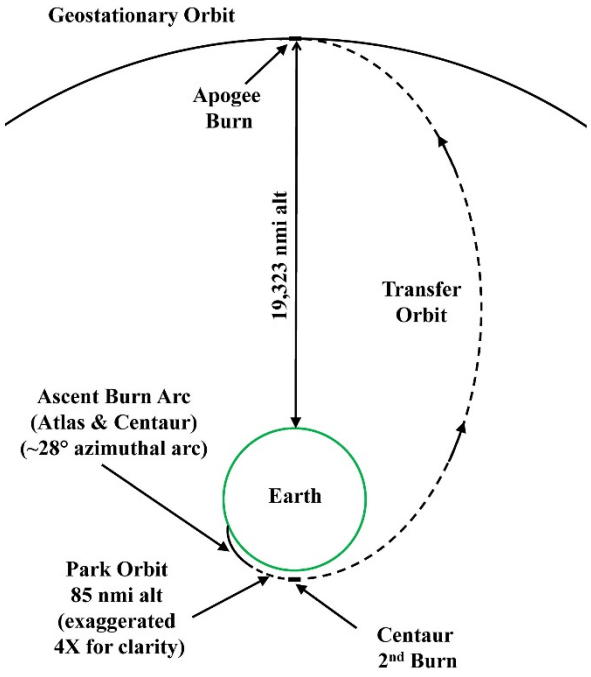

Figure 3.—Geostationary Mission (In-Plane)
(Drawn to scale unless noted)

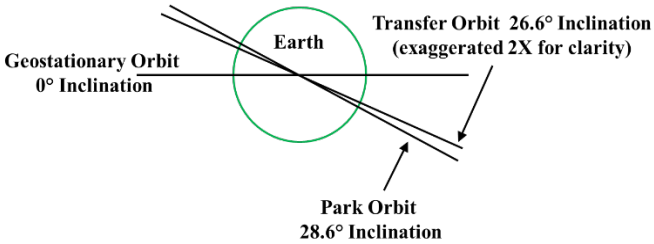

Figure 4.—Geostationary Mission (Out-of-Plane)
(Drawn to scale unless noted)

Soon after, we were also asked to launch Intelsat GSO missions, which were similar to ATS missions, except that the spacecraft solid motors were not yet selected and the total impulses of those motors could still be optimized. "Piece-meal" optimization of these trajectories to yield maximum payload to GSO was nearly impossible because of the very large number of variables. There was also enormous pressure to optimize the trajectories to yield maximum payload. This was unlike the ATS case, where only enough extra launch vehicle capability was needed to match the solid motor total impulse. For the commercial Intelsat missions, any extra payload to GSO could be used to add capability to the spacecraft, or provide extra propellant to the stationkeeping tanks. This extra propellant could add months or years to orbit lifetime, which was highly desirable because spacecraft lifetime was directly related to revenue.

It was not clear that an Atlas/Centaur code with the desired precision was possible on the small, slow computer available to us. Abandoning the calculus of variations was considered. At first, using a steepest descent technique was considered, but this was discarded when it became clear that this technique would be much slower than the calculus of variations. Each of the variables would require a separate trajectory to utilize such a technique. The accuracy of such a technique was unknown and capturing a misleading local (not global) maximum was considered a real possibility. The calculus of variations formulation also had a known risk of local maxima, but the threat seemed more manageable. An infinite series approach to the problem was also tried, but it soon swamped the machine and for our era, seemed impractical if not impossible. This technique would probably have worked with a simple optimization problem, but it did not work with the complexity of the model required to analyze our problem. So the calculus of variations formulation was rederived in spherical coordinates in the belief that the numerical integration of the coast periods would open up the numerical integration step size and reduce the computer time required per problem. It did, but the spherical coordinate formulation increased the size of the code and did not lend itself to facilitating FORTRAN do-loops. Thus, it was concluded to formulate the CoV problem in rectangular coordinates which did lend itself to the use of "do-loops," reducing code size. In retrospect, this last decision with the resulting derivations and coding represented the first version of what would be called DUKSUP.

Consequently, the CoV-based DUKSUP code was written which optimized the GSO trajectories from liftoff to injection into GSO (Ref. 8). To our knowledge, no one had attempted to do that before. This meant not only optimizing the trajectory "path," but also optimizing the duration of the Centaur burns (the sum of the burn times being a constant dictated by the maximum propellant load of the Centaur), the durations of the two coasts, and in the case of the Intelsat missions, the total impulse of the spacecraft solid motor. This proved to be a formidable task. Three problems had to be overcome immediately (Ref. 8). The first problem was that the CoV technique required the solution of a multi-variable two-point boundary value problem—that is, it was necessary to vary initial conditions until final conditions were satisfied. The second problem was that to obtain optimum solutions for such a long-duration mission (covering a travel arc of about 180° for the transfer orbit coast), it was necessary to include the first harmonic of the Earth's oblateness in the CoV formulation. Ignoring the first harmonic was demonstrated to yield nonoptimum solutions. The third problem was that intermediate constraints had to be imposed explicitly in the CoV formulation to model the problem. Specifically, the altitude of the parking orbit had to be constrained to at least an acceptable value, else the trajectories would unacceptably "dip" during the first coast.

To address the first problem, up to this point, we had obtained the derivatives to solve the two-point boundary value problem using perturbed trajectories. This had been a cumbersome but simple and adequate technique for the relatively short duration problems we had solved earlier. However, for these long duration, complicated problems with near circular parking orbits,

we had to derive and code equations to obtain analytic derivatives to get solutions.

For the second problem, it turned out that including the first harmonic of the oblateness in the CoV solution was feasible, and easy, and did not really complicate the formulation. Including the first harmonic eliminated nonoptimum solutions. By ignoring the harmonic and comparing that trajectory with one including it, we demonstrated that for such a long duration and arc, the harmonic had a significant effect on the trajectory. As will be discussed later, understanding the effects of the harmonic on trajectories provided valuable experience in rendezvous and round trip problems, enabling the code to be modified to accommodate space-tug missions.

The third problem introduced the challenge of imposing constraints in the middle of a trajectory, specifically to limit the perigee altitude of the parking orbit such that the trajectory would not dip back into the sensible atmosphere, which it would do "optimally." Our first solution to the problem was to force the Centaur first burn to "target" to a circular parking orbit at a specified altitude. Due to oblateness, the orbit did not stay perfectly circular, but close. We were able to obtain solutions for both the ATS and Intelsat missions. For Intelsat, the entire mission was optimized from liftoff to apogee motor burnout, resulting in maximum payload to GSO (verified by perturbation). In particular, the amount of plane change performed by the Centaur second burn was optimized (about ~2°). Circularization and the final plane change at GSO were performed by an optimally sized solid motor on the spacecraft. As mentioned earlier for ATS, the desired spacecraft weight had grown, and the original launch vehicle was no longer able to place the heavier spacecraft in the desired geostationary transfer orbit. The apogee motor was too small, having been sized for a lower payload and a less capable launch vehicle. Therefore, the Atlas/Centaur had to reduce the ΔV required of the fixed total impulse motor to match the ΔV required for the desired final maneuver into GSO. A solution was obtained at apogee by decreasing the inclination to be removed by the apogee motor and increasing the inclination to be removed by the Centaur second burn. Later, we would further examine the trajectory design and realize that there was a better solution; namely, the perigee of the transfer orbit could also be raised to lower the in-plane ΔV required at apogee. But this meant that the parking orbit was no longer near-circular, but instead would be elliptical. The solution that would maximize the payload in the final orbit would reduce the ΔV at apogee by reducing the inclination change required of the apogee motor, and also reduce the ΔV to circularize the orbit by raising the perigee altitude of the transfer orbit. The optimum combination of these two maneuvers would yield the maximum final payload.

Optimizing the problem from liftoff until injection into GSO meant that the optimization was one problem, not two or more. That introduced intermediate boundary conditions at injection into the parking orbit—initially three: energy, radius, and flight path angle. If properly calculated, these three conditions (these were not the only three) would deliver a circular orbit with an optimum inclination and node. But these intermediate boundary conditions introduced discontinuities in the adjoint equations and their derivatives. Each of them introduced a discontinuity which had to be calculated, accompanied by an additional initial variable to be added to the two point boundary value problem. Beyond that, optimum duration burns and coasts introduced other equations that had to be satisfied for optimality. These, too, introduced more initial variables and final conditions that added to the two point boundary value problem.

As these capabilities were added to the code, the optimality of the result was checked parametrically. That is, the code determined an optimum value to the first burn of the Centaur. The code was flexible enough to allow us to fix the duration of the first burn and vary it around the "optimum" value. We found that the "optimum" was not the optimum. We had assumed that it would not be necessary to include oblateness in the derivation of the variational equations. Including the first harmonic of the oblateness model did not add significantly to the complexity or length of the code, so it was included. With that addition, the "optimum" was determined to be the optimum within less than two-tenths of a pound. This amazed us, because we had no idea what to expect from the theory or the code. On reflection, it should not have been surprising, since the modeling was simple and did not introduce "noise" into the application of the theory.

In summary: DUKSUP was designed to optimize the thrust direction consistent with constraints, the duration of burn times that were free for optimization, and to optimize the intermediate and final injection orbit elements that were not specified to maximize the payload available to the spacecraft. We pursued the goal of extracting every bit of payload for any mission that could be obtained from by optimizing the trajectory and mission design. If the vehicle could provide more payload than needed by the spacecraft, the excess capability was used to improve range safely or to provide the spacecraft with better telemetry coverage, etc. We had enough confidence in the reliability of the vehicle that we strived to make available to the mission the benefit of using all the launch vehicle propellant.

5.3 Limitations of FORTRAN and Its Impacts

FORTRAN was the only scientific and engineering programming language available to us. It was not yet standardized (still in the process of development by the computer systems group) and thus was not dependable. Codes would compile one day but frequently not the next. Sometimes they would compile but would crash. The systems group was usually busy and often not able to help us debug. It was not unusual to have weeks go by without being able to move forward because of such problems. The only way at the time to debug a program was to dump (octal dump) the core of the computer and look at the machine language to determine exactly what the compiler had done with the FORTRAN. In self-defense we learned to "read" the octal dumps and understand exactly what the machine was doing. It

was common in a large FORTRAN "do loop" for the machine to "forget" the index, and it was necessary to add "wakeup" statements and dummy subroutines to the code to make sure the machine "remembered." But as a result, we learned a lot about how the compiler "translated". We also learned how to use FORTRAN in ways that were not anticipated by the compiler designers in order to save memory and perform tasks that we otherwise could not have done.

When the sections were formed in the spring of 1962 to conduct analysis in support of the Propulsion Research Division's research and the Atlas/Centaur program, there were few if any codes available commercially. Unlike the present, we had no choice but to develop our own codes in order to do our analysis. Numerical analysis, as such, did not exist or was in its infancy. We were unaware of any text books on the subject. We had to do the best we could with the codes we had developed as of the moment and we were constantly improving existing codes and developing new ones. When we were asked to tackle a new problem, we were never confident that we could formulate a code that would answer the questions being asked using the slow, small computer of that era. We soon found that the optimization of Atlas/Centaur trajectories was not feasible by way of the perturbation techniques we had used to up to that point. The only way to converge the two-point boundary problem was to obtain accurate analytical, partial derivatives. This meant numerically integrating many more differential equations. (The last version of the code numerically integrated almost 200 equations.) This was a real challenge to implement with the FORTRAN and mainframes of the time. For instance, numerical integration was done in double precision to obtain the desired accuracy. (With the arrival of the Cray supercomputers many years later, the need for double precision variables and operations was greatly diminished.)

When this code was initially developed, the discipline of numerical analysis was either undeveloped or in its infancy. The numerical problems encountered in the development of DUKSUP were solved by the developer by observing the lack of accuracy in the results of numerical calculations and solving the problems empirically and not theoretically. A course in numerical analysis as it later developed would have been quite useful. Another complication in the beginning was that FORTRAN was not standardized and in fact was in development also. We did know enough to be outraged with a routine would compile one day and might not again for several weeks. But learning, of necessity, how FORTRAN was translated into machine language gave us an advantage in saving storage and using FORTRAN in ways for which it was never intended. Later attempts to force good coding practices (no doubt well intended and in the end highly desirable) could only frustrate those of us who had learned to "trick" what is now a primitive language.

When we started a new code or modified an existing code, we were never sure that it would "fit" on the machine. Consequently, we were never sure that an immense amount of work might come to naught. Thanks to the initial problems with FORTRAN, we were aware of how it compiled and we designed codes to take advantage of any techniques that would save storage. We also designed to code such that core could be reused. This led to DUKSUP's structure and intimate reliance on FORTRAN's "Common Block" feature. Subroutines would pass data back and forth through the Common rather than arguments in the Calls. A major problem resulted from this type of operation in that identifying a variable's numeric value had to be done by querying specific cells in the Common rather than the variable itself. Other operations also used prodigious amounts of memory. Integrating as many as two hundred equations took a lot of core. Inverting a 20x20 matrix also used a lot of storage. The code was organized to use shared storage since the operations were independent. Almost all variables were assigned specific storage locations (i.e., the Common) such that where feasible, other variables were temporarily stored in the same location to save space. This made debugging the code difficult because the same cell would have a different variable depending on where the code crashed. And like all engineering and scientific codes of that era, DUKSUP's coding was unstructured (aka "spaghetti code," dominated by "GOTO statements") due to the lack of available memory and the limited capability of FORTRAN's early versions. DUKSUP was typically running with as few as two or three unused cells in memory. This was a standard procedure until the late 1970s, when larger, faster machines became available. One consequence of these operational constraints was that DUKSUP was virtually undocumentable because of the convoluted code design required to save memory space.

5.4 Overview of Algorithm Structure and General Operation

What follows is a general description of the code without detailing the internal subroutine interactions, nor the many versions that were created during the almost 30 years of its usage. Since DUKSUP was developed to analyze performance for the Atlas/Centaur, its structure was driven by the vehicle and how it operated. The rest of the code structure and operation was driven by the early computer's limitations (see Secs. 5.3. and 8.0). As a result, the code was structured to be as compact (in terms of memory) and fast as we could make it. Consequently, it was broken into two parts; first: the nonvariational zero angle-of-attack section (controlled by the MAIN procedure) and second: the remainder of the trajectory. This second part constituted the majority of DUKSUP, that is to say, the two point boundary value problem (2PBVP), which included the variational part of the problem, beginning with the jettison of

the booster engines. This second part was controlled by subroutine MAIN2, which was called by MAIN, but functioned as a second main procedure, and continued until the end of program execution. DUKSUP's algorithm structure is illustrated in Figure 5, but this is only representative of the discussion in Sections 3.0 and 4.0 (i.e., is not necessarily reflective of how the detailed, multiple subroutine calls actually took place). Any more than a top-level discussion of its operation would warrant a separate paper in and of itself (see Secs. 5.2. and 5.3).

The first part of the code operation was for flight of the Atlas booster at zero angle of attack, from liftoff to booster engine cut-off (BECO). The trajectory was integrated, but flown open loop and external to the majority of the code operation. During these early phases, the Main procedure called atmosphere-related subroutines (by way of the integration subroutine). To save computer time, when aerodynamic heating during launch was specified, the code satisfied the constraint by integrating only through the booster section, since almost all of the heating occurred up to that point. If the kick angle were to be optimized (which occurred in the booster phase of flight), and since payload was determined upon reaching the desired orbit, the payload to final injection had to be obtained first to optimize the kick angle (and maximize payload).

The second part of the code (and the majority of its operation) was driven by a generalized Newton-Raphson routine which iterated to solve the 2PBVP. However, the majority of the oper

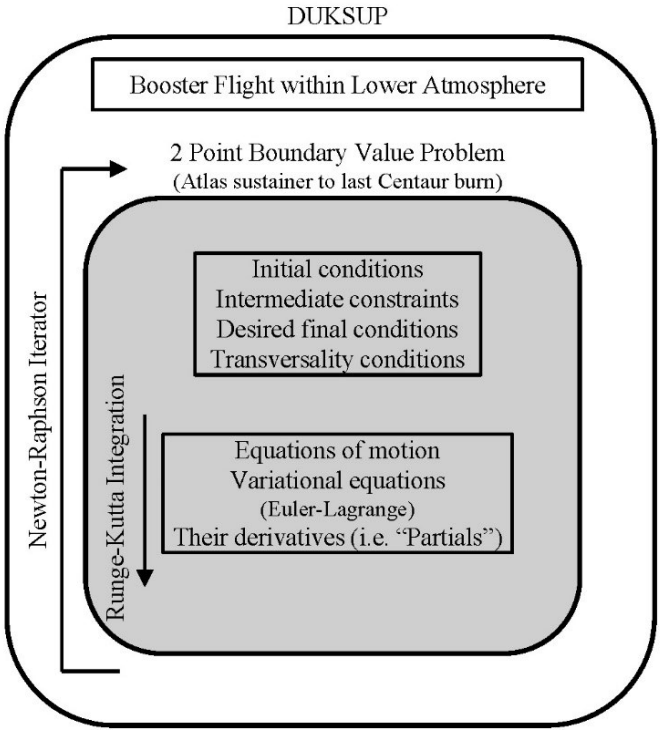

Figure 5.—DUKSUP Algorithm Structure

ation and complexity of the 2PBVP resided in the calculus of variations subroutines and the integration subroutine. Subroutine MAIN2 controlled the 2PBVP. The number of initial and final conditions was variable. In some simple cases, the number could be as small as two. It was usually as many as eight, but in one case it was twelve. The user could select various combinations of final conditions as well as intermediate constraints (Table 2). The variational equations and the partials related to these intermediate constraints/final conditions were analytically derived and hard-coded into the algorithm. They represent some of the most intellectually challenging aspects of the code, and at the same time some of the most powerful aspects of DUKSUP, speeding up its execution and even enabling convergence to an optimal solution at all. That is, compared to other codes (which were predominantly numerical in nature), the speed of DUKSUP with its built-in analytical theory and equations, was incredibly fast without sacrificing performance. This was independent of the hardware platform (computer). Within the 2PBVP were the operations which interpreted the specified conditions and which calculated the Transversality conditions for those variables which were left free to be optimized. These conditions defined the problem to be optimized. A 4^{th} Order Runge-Kutta subroutine then integrated the equations of motion, the variational (Euler-Lagrange) equations, and also their partial derivatives (which were exact, analytically derived equations, not numerically arrived at approximations). The differences between the end conditions and the desired final conditions (both those specified and those which were left to be optimized) were evaluated. These differences were then used with the exact partials to calculate values for the Newton-Raphson iteration scheme. The problem was then rerun with a new set of initial guesses. This would continue until the convergence criteria were satisfied, thus arriving at a solution. A simplified overview of this algorithm exists in a publication external to LeRC (Ref. 9).

TABLE 2.—USER SELECTABLE INTERMEDIATE CONSTRAINTS AND FINAL CONDITIONS

Intermediate constraints	Final (in-plane) conditions	Final (out-of-plane) conditions
Radius	Energy (C_3)	Inclination
Velocity	Apogee radius	Argument of perigee
Sin (flight path angle)	Perigee radius	Ascending node
Sin (elevation angle)	Radius	Declination of asymptote (hyperbolic)
Perigee radius	True anomaly	Right ascension of asymptote (hyperbolic)
Energy (C_3)	Flight path angle	

The code could accommodate up to four 2PBVPs simultaneously, with other nonvariational segments interspersed (though typically reserved for the atmospheric phase of flight). The flight could be split up into 100 time phases, the main unit of modeling the trajectory. Only five of these time phases however could be optimized, though they could be any combination of powered or coasting flight. Variational steering could be turned on or off throughout the trajectory, facilitating modeling of the small transients and other propulsive losses. Hardware could be dropped at the end of any phase. Phase lengths were specified if not designated for optimization. Thrusts and weight flows were specified for all powered phases, and various levels of fidelity were available modeling various sized engines. Various ways to constrain how the vehicle was allowed to fly (variational, constrained by planes, etc.) were available to the user.

Towards the end of its utilization in the mid-1990s, DUKSUP had grown to over 5,200 lines of (almost totally) uncommented code. There were 29 primary subroutines, six major utilities, and 31 smaller utilities/functions. With the exception of a few new subroutines and modest upgrades, the code remained written in FORTRAN IV throughout its life.

5.5 Primary Subroutines

The following is a brief summary of the 29 primary subroutines within DUKSUP. While most were written by the lead author, including all the major routines, some (ORBEL, others) were adopted either in whole or in part from other earlier LeRC codes. In addition, there were six major utilities, several block data, and over 30 small utilities /functions included in the compilation of the code. Figure 6 illustrates DUKSUP's primary subroutines and how they were called. The following section contains a brief explanation of each of the primary subroutines, arranged in the order of appearance in Figure 6 from left to right, then top to bottom.

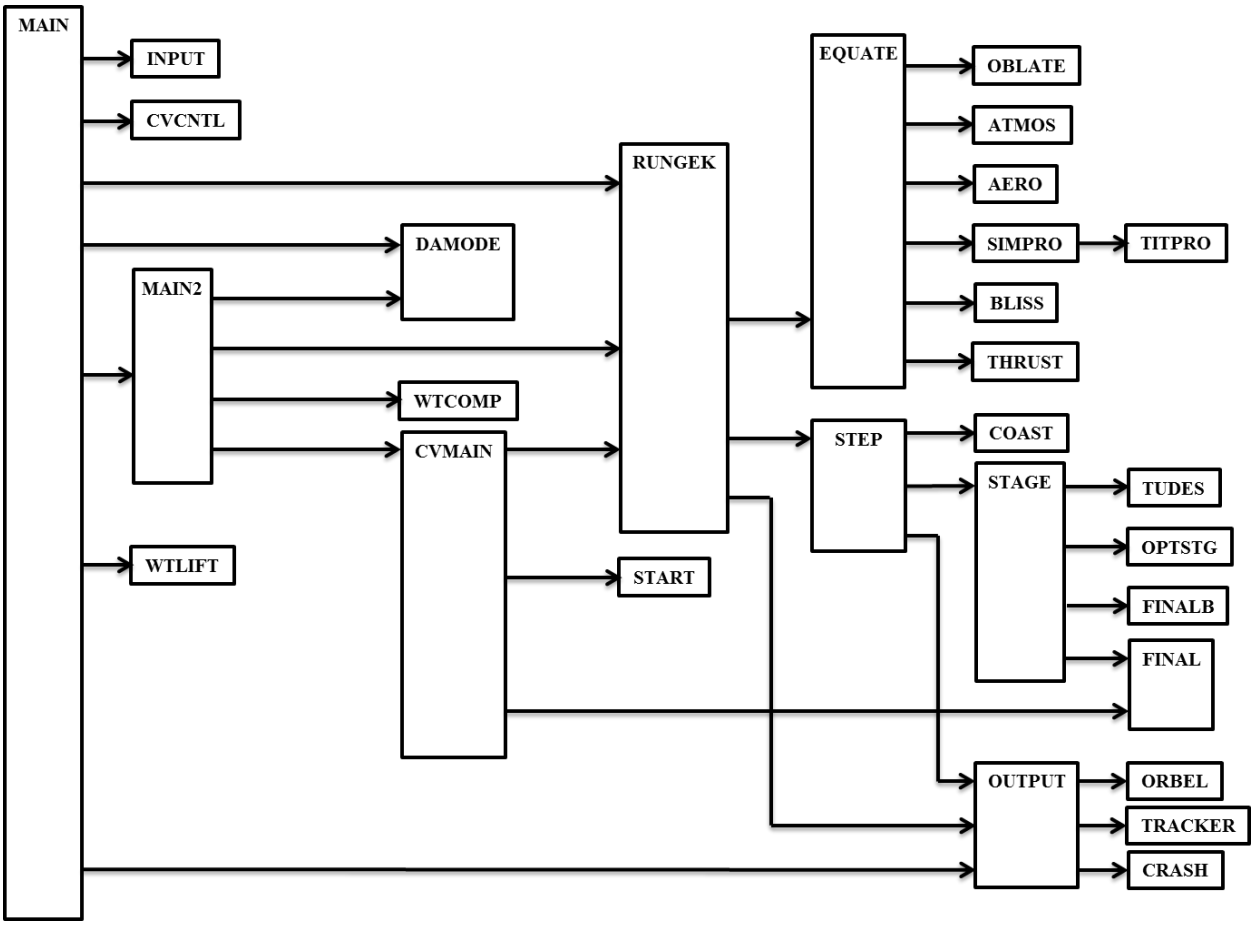

Figure 6.—Subroutine Calls in DUKSUP

5.5.1 MAIN

The main procedure of DUKSUP was actually broken up into two parts. MAIN initiated the program (and called INPUT and CVCNTL) and also performed the zero angle-of-attack (non-variational) booster part of the problem.

5.5.2 INPUT and Supporting Routines

This routine provided a convenient and very flexible way of inputting data into the code. It had existed for some time prior to its use by the performance analysis group. It is unknown whether it had a single developer or whether it represented the consecutive efforts of several developers. We believe that one of the primary developers was Vearl N. Huff, an early researcher at the NACA Lewis Research Center. It had been rewritten more than once to add features, making it more versatile. Arithmetic calculations as well as trigonometric and logarithmic functions were incorporated. As discussed, one of the greatest challenges to using a CoV optimization technique was solving the two point boundary problem. For other than simple problems, converging a problem usually involved moving from a solution of a related problem to the desired problem in small steps. Solving the problem was extremely dependent on acquiring initial guesses that were close to the final guesses which led to the solution. This routine simplified the tedious task of assembling the input for a series of problems leading to the solution. (Called from MAIN)

5.5.3 CVCNTL

The name of this subroutine was shorthand for "Calculus of Variations Control". The possible end conditions for the code are energy, radius, flight path angle, injection true anomaly, apogee radius, perigee radius, declination, right ascension, inclination, argument of perigee, and node. (If both right ascension and declination were specified, this defined an asymptote for a hyperbolic trajectory for an interplanetary mission). Any independent set of these could be entered as desired end conditions. This routine was very complicated, but set up the problem and designated the initial conditions and final conditions to be evaluated. These in turn were used to obtain the desired final orbit and identified the equations which I call variational end conditions. These end conditions, when satisfied, provided the optimum value of the unspecified element. This routine made the code usable for new users who were relieved from understanding the calculus of variations and facilitated converging the two point boundary value problem. (Called from MAIN)

5.5.4 MAIN2

This subroutine served as the main procedure for the remainder of the problem, beginning with the jettison of the booster engines. (Called from MAIN)

5.5.5 WTLIFT

This subroutine determines the time after SRM ignition when a thrust-to-weight ratio of 1.0 is achieved and the vehicle weight at that time. (Called from MAIN)

5.5.6 DAMODE

This subroutine set up the output mode (i.e., print out of results) for converged and nonconverged cases; and also boost and variational phases. (Called from MAIN, MAIN2)

5.5.7 WTCOMP

This subroutine was called at the end of a trajectory. It calculated the final payload after flight performance reserve (FPR) and any other reserves were subtracted from the weight at the end of the integration. There were many versions of this routine, depending on the problem. For an exploratory study to determine the payload potential for some mission (given a representative target), FPR was often calculated as a linear function of burnout weight. That linear function was derived from plotting FPR's from actual missions where FPR had been rigorously determined using dispersion data from previous flights. Dispersions in determining initial weight, engine performance, jettison weights, and other factors were used to determine an FPR for an actual mission. It was found that these rigorously determined FPR's fell close to a line that was a function of burnout weight. WTCOMP also provided a convenient place to make other calculations of interest to us or to the spacecraft, since DUKSUP users knew the program was at the end of its execution. (Called from MAIN2)

5.5.8 CVMAIN

This routine set up the calculus of variations problem and called RUNGEK. The partial derivatives for solving the two-point boundary value problem had to be analytical, not obtained by perturbing initial conditions and calculating "partials" from differences in final conditions from the nominal. (Called from MAIN2)

5.5.9 RUNGEK

A 4^{th} order Runge-Kutta algorithm resided in subroutine RUNGEK. The foundation of the code was this routine which numerically integrated the differential equations characteristic of problems of this sort. Adopted from the LeRC N-Body (Ref. 2) and the low-thrust Mars codes (Ref. 7), this variable step-size, fourth order, double precision Runge-Kutta routine used a single precision Simpson's rule parallel integration technique to vary step-size such that the difference between the two integration techniques could be specified. Step-size was varied to maintain that set difference. The Runge-Kutta was the more accurate integration technique and the results from that integration

were used to "boot strap" the integration. We rewrote the routine to meet our requirements. (Called from MAIN, MAIN2, CVMAIN)

5.5.10 START

This routine set up the steering angles and their rates of change for the start of the problem, along with other initial variables. (Called from CVMAIN)

5.5.11 EQUATE

This subroutine supplied the derivatives for the integration. It was a short routine which called several other routines, either directly or indirectly. (Called from RUNGEK)

5.5.12 STEP

This subroutine compared the results of the Runge-Kutta integration with a Simpson's rule integration. The integration step size was adjusted continuously to maintain a designated difference between the two methods. If the difference did not exceed an upper bound, the integration was accepted and integration proceeds. If the difference exceeds the bound, the step size was reduced based on a linear interpolation and the integration is "redone." (Called from RUNGEK)

5.5.13 OBLATE

Subroutine OBLATE provided three harmonics of the Earth's oblateness. We found that it was necessary to include Earth oblateness in the trajectory calculations when long coasts were included. The impact of the Earth's oblateness is significant for long missions, such as GSO missions. Orbit elements changed noticeably for these missions. We did not expect that the effects of oblateness would be required in the derivation of optimum steering and staging. However, parametric investigation of optimum staging revealed that neglecting this effect resulted in significant errors in optimum staging. Incorporation of the first harmonic of the oblateness in the CoV derivation eliminated the discrepancy. Eventually, using the first four harmonics became standard procedure. (Called from EQUATE)

5.5.14 ATMOS

Subroutine ATMOS provided atmospheric pressure as a function of altitude. Input to the routine was altitude and output was atmospheric pressure. The atmosphere was modeled by polynomials which were based on the standard atmosphere for the launch site at Cape Canaveral Air Force Station (CCAFS), Florida. (Called from EQUATE)

5.5.15 AERO

Subroutine AERO provided drag coefficients as a function of Mach number. There were different drag coefficients for the portion of flight where the booster engines were attached and operating. A different drag coefficient applied after the booster engines were jettisoned. Mach number was input and the drag coefficient was output. Drag on the vehicle was then calculated. Winds could be incorporated but this was rarely used. Lift coefficients could also be implemented, but were very rarely exercised since the vehicle was normally constrained to fly in a near zero-angle-of-attack mode. We were aware that some angle of attack would increase payload, but uncertainty in the calculation of bending loads and local heating rates dictated conservatism in inducing angles of attack. (Called from EQUATE)

5.5.16 SIMPRO

This subroutine calculated the thrusts of up to three engines, each with their own performance characteristics which could be quadratic functions of time. (Called from EQUATE)

5.5.17 BLISS

This routine initialized the calculus of variations problem, including initializing the equations for obtaining the analytical partials. It also supplied the derivatives of the calculus of variations adjoint variables during the integration of the differential equations. It would be very difficult to guess at the adjoint variables to begin the solution of the two point boundary value problem. Instead, the adjoint variables were calculated from initial guesses at pitch angle, the rate of change of pitch angle, yaw angle, and the rate of change of yaw angle. These variables had obvious physical meanings and reasonable guesses could be made. This subroutine was named after Gilbert Ames Bliss of the University of Chicago, one of the first leaders of the Calculus of Variations (particularly the Problem of Bolza) (Ref. 10). (Called from EQUATE)

5.5.18 THRUST

This subroutine calculated the vehicle thrust accelerations in Cartesian coordinates. (Called from EQUATE)

5.5.19 COAST

A subroutine pertained to the insulation panels and payload fairing. (Called from STEP)

5.5.20 STAGE

A weight (representing a spent stage, jettisoned hardware, etc.) could be dropped at the end of each phase. This subroutine decreased the current stack weight by the prescribed amount at end of that phase. (Called from STEP)

5.5.21 OUTPUT

This subroutine calculated various orbital elements not calculated by ORBEL, as well as various steering angles, and other quantities. It then printed out resultant variables at times controlled by DAMODE. (Called from MAIN, RUNGEK, STEP)

5.5.22 TITPRO

This subroutine computes the Titan SRM/thrust vector control (TVC) thrusts, weight-flows (wtflows), and wtflows of the inerts in a manner similar to subroutine SIMPRO. These values are then fed into SIMPRO to compute total thrusts and wtflows. (Called from SIMPRO)

5.5.23 TUDES

This subroutine calculated the initial state vector at lift-off (or at some later point) from initial latitude, longitude, and altitude. Usually this was the launch pad, but in some versions of DUKSUP, it could be a point after lift-off, such as in orbit, where latitude, longitude, radius, and elapsed time might have been input to calculate an initial state vector. This was used extensively for the Shuttle/Centaur studies. (Called from STAGE)

5.5.24 OPTSTG

This subroutine calculated the variational equations and their derivatives used to optimize the durations of burns and coasts. DUKSUP permitted the optimization of the duration of as many as five variable (time) "phases". These phases could have been powered or coasts. The five that were selected to perform the GSO missions were: the Centaur's two burns, separated by a coast (which had a very low thrust to keep propellants settled), a long coast up to geostationary altitude, followed by an injection burn into the final orbit. (Called from STAGE)

5.5.25 FINALB

This subroutine was called when any specified intermediate condition and its derivatives must be evaluated. These intermediate conditions included energy, radius, flight path angle, true anomaly, perigee radius, apogee radius, and view angle from a designated ground tracking site. Many of the equations used here were identical to those in FINAL, they were just evaluated at an intermediate point in the trajectory. There are conditions which were not included in FINALB, such as declination and right ascension. FINALB was called before the vehicle reached escape velocity, so those variables would not be defined at that point in the trajectory. Theoretically, they could be included if an intermediate hyperbolic target were desired, which was unlikely. Any time an intermediate constraint was imposed, there was a discontinuity in the adjoint variables. Sometimes the discontinuity could be calculated, but other times the discontinuity required that a single input constant be applied to calculated vectors for each of the discontinuities in the adjoint variables. As there were discontinuities in the adjoint variables, these discontinuities introduced associated discontinuities in the partial derivatives. Those discontinuities are calculated for each of the partials and the partials were then "corrected" to incorporate the constraint. (Called from STAGE)

5.5.26 FINAL

This subroutine calculated the designated end conditions and the analytic derivatives of those conditions from the final state vector and its derivatives. It also calculated the variational end conditions required to optimize the orbit elements that are not specified targets. CVCNTL set up the problem and provided FINAL with the appropriate list of conditions to be evaluated. (Called from CVMAIN, STAGE)

5.5.27 ORBEL

This subroutine calculated orbital elements and time to perigee from a state vector. Parts of this routine were originally written for the low thrust code, and were unchanged for perhaps 60 years. (Called from OUTPUT)

5.5.28 TRACKER

This subroutine calculates the elevation angle, range, and other visibility variables from various ground tracking stations and on-orbit assets to the launch vehicle. (Called from OUTPUT)

5.5.29 CRASH

This subroutine calculated where on the oblate surface of the Earth the vehicle would impact if thrust were terminated. The information from this routine is used to plot the instantaneous impact point trace for a trajectory and was a major contributor to the calculations of risk by Range Safety at the Cape. (Called from OUTPUT)

5.5.30 WISSEN, ITERAT, MINMAX, INVERT, RASCAL, ERRORZ (major utilities)

More than just utilities, these original or significantly upgraded subroutines found the maximums/minimums of functions, inverted nonsingular N X N matrixes, performed linear or parabolic interpolation schemes, etc. (Called from various locations)

5.5.31 Block Data (various)

There were several Block Data loads within DUKSUP pertaining to the INPUT routine, atmospheric model, initialized vehicle characteristics, tolerances, geo-physical data, etc.

5.5.32 Other Smaller Utilities and Functions

There were over 30 smaller utilities and functions which were either too small to mention or improvements over existing library functions. They were part of the code when submitted to the compiler, not merely routines linked to satisfy unresolved references in the object code.

5.6 Versions of the Code

Although DUKSUP had a single basic structure, because of computer limitations and the analytic environment, there was never a single version of the code. As new missions materialized, the code was changed and/or augmented to solve the new problem. Sometimes these changes were incorporated into a new version of the code if we thought that that particular problem would reoccur frequently enough to merit the work and time involved. For instance, when SKYLAB drag threatened and finally did result in re-entry, the possibility of using the Atlas/Centaur to raise its orbit to prolong its life was proposed. A new version of DUKSUP was created to do rendezvous missions. The study was completed (it was possible), but the changes in the code needed to do that analysis were never incorporated into the "standard" version since it was not anticipated that another rendezvous mission for Atlas/Centaur would be proposed.

The original version of the code utilized select sub-routines from other existing computer programs, such as the RUNKEK, ORBEL, and INPUT routines. The author's work with the Living Booster Table was also incorporated. But the lion-share of the code relied on new, major subroutines written by the lead author (originally 26, later expanded to 29). These were the Calculus of Variations routines just discussed in the preceding section. Together with the modeling of the Atlas/Centaur, all this combined constituted the first version of DUKSUP. Figure 7 illustrates an approximation of how DUKSUP originated and how its various versions evolved.

The earliest version of DUKSUP was written specifically to analyze Atlas/Centaur missions. Later, other versions ('decks') were created. The Titan deck was similar to the Atlas deck, but it did contain modifications to use the 'table look up' data for the monolithic solid rocket motors used as the first stage. The last version of the Titan deck was used solely for the Cassini mission on Titan IV/Centaur. It had significant modifications in subroutine STAGE, which incorporated optimal payload fairing jettison logic based on qV (see Sec. 7.2) into the variational problem. Another deck had been created which solved a rendezvous problem (see Sec. 9.6), which was then readily modified into the on-orbit deck for Shuttle/Centaur missions. Though this version (SHUTSUP) retained all the atmospheric routines of its predecessors, they were not called upon during the execution of the on-orbit deck for these missions. Another variant existed written in spherical coordinates but was not used for launch support. The most powerful and sophisticated version, SUPERDUK, could solve varying launch azimuth while also constraining the ascent to be within the line of sight of ground tracking stations. Since others in the launch organization could not easily use this result, SUPERDUK was never used again (Sec.11.10).

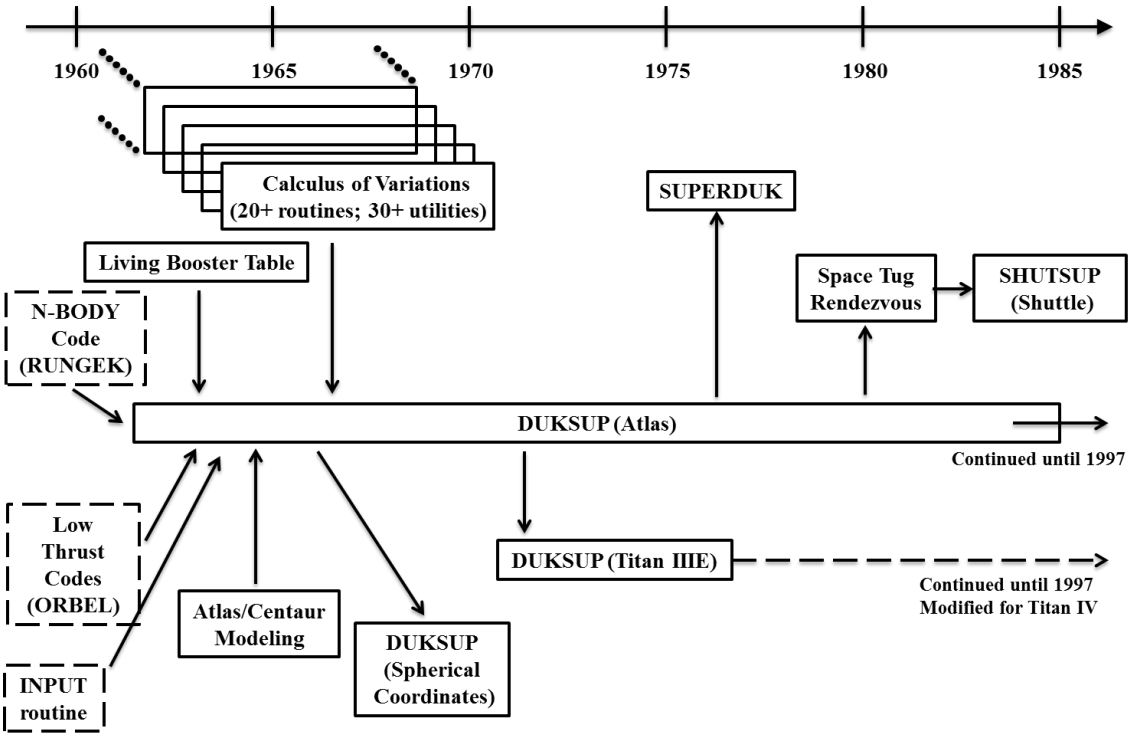

Figure 7.—DUKSUP's Origin and Evolution of Its Versions (dashed boxes indicate other authors)

5.7 Atlas/Centaur Steering Versus CoV and the Post-Flight Correlation to Actuals

In the late 1960s, these trajectory solutions were only obtainable by using a CoV formulation of the problem. The trajectories were 3D, which were sufficient to obtain the needed targets for the on-board guidance system. The 3D vehicle models used for the CoV were based on detailed, post-flight reconstruction of flight data. The pitch steering algorithm for the vehicle was a linear tangent law.[3] The CoV solution collapsed to this linear tangent steering law if the Earth was assumed flat and the gravity constant. The Centaur on-board computer actually recalculated a and b every 8 sec during flight, yielding a very nearly optimum trajectory. Optimum trajectories from lift-off to GSO injection obtained using DUKSUP's CoV algorithm were matched by the Centaur's linear tangent steering quite closely (within a couple of pounds). Accurate determination and prediction of performance allowed the Atlas/Centaurs to be flown with very small performance margins (near zero for some missions) above the flight performance reserve (FPR) (which was carried to protect against 3σ in-flight dispersions). This means that we obtained the most payload available from the vehicle.

Each segment (phase of flight) was analyzed independently to ensure that the performance characteristics of the segment matched the flight data. In most cases, the thrust level was determined by loading vacuum thrusts and exit areas for as many as three engine types. Three engine types were modeled because the Atlas at lift off had two booster engines, a sustainer engine, and (small) vernier engines. Each of these engine systems was modeled separately for simplicity. In atmospheric flight, total vehicle thrust was the vector sum of each vacuum engine thrust value reduced by the product of engine exit area and atmospheric pressure.

Post-flight reconstruction was a comprehensive analysis of how the vehicle flew. A system-by-system review of actual flight data was performed after each launch by the GD/LeRC team. A primary reason for this major activity was to correlate preflight predictions made by analytical tools to flight actuals. Thrust, specific impulse, and acceleration as functions of time, as well as telemetered and radar tracking state vector data, were key parts of the post-flight analysis. As a result of verification through many post-flight analyses, DUKSUP enabled us to predict Atlas/Centaur performance—a vehicle with a ~360,000 lb gross liftoff weight (GLOW)—to within less than 1 lb.

6.0 Centaur Configurations and Their Boosters

All Centaur upper stage configurations were manufactured by General Dynamics Convair Aerospace Division (later General Dynamics (GD) Space Systems Division), Kearny Mesa, California. Centaur's first stage boosters, however, were manufactured by various aerospace companies: General Dynamics (GD) for Atlas, Martin Marietta Corporation (MMC) for Titan, and Rockwell International for the Space Shuttle. The booster + upper stage analytical models resided partially in the input datasets (for variables likely to be changing) and partially within the code itself (for more invariant parts). How DUKSUP was structured and executed was fundamentally driven by how the launch vehicle itself operated. All LeRC-led Centaur missions were launched from pads at Cape Canaveral AFS.

6.1 Atlas/Centaur

The Atlas boosters were built by General Dynamics in Kearny Mesa (San Diego), California. During the tenure of the LeRC-led launch vehicle program, the Atlas/Centaur was a two and one half stage vehicle based on its heritage as the nation's earliest ICBM. At liftoff, three main MA-5 engines (two booster and a single sustainer engine) were burning using propellant from the same Atlas tanks. The booster engines were shut down and jettisoned upon a thrust-to-weight ratio of ~5.7 g's. After BECO, they were jettisoned based on measurements from an on-board accelerometer, while the sustainer engine was shut down based on low (propellant) level sensors. During the sustainer phase of flight, the insulation panels were jettisoned. Following sustainer engine cutoff (SECO) and Atlas jettison, Centaur phase of flight began. During the Centaur phase of flight, the payload fairing was jettisoned (in later years, the PLF was jettisoned earlier, prior to SECO). Centaur shutdowns were by guidance command, though missions in the 1990s accommodated "minimum residual shutdowns" and "in-flight retargeting" for the second burn. Figure 8 shows a drawing of the Atlas/Centaur vehicle and a photo of AC-45 on the pad prepared for launch of the HEAO-1 spacecraft (Ref. 11). A typical Atlas/Centaur flight profile is illustrated in Figure 9 (Ref. 11). Figure 10 is an artist's rendition of the Centaur with the Intelsat IV payload following fairing separation (Ref. 11).

To determine the performance capability of the Atlas/Centaur, an analytical model was constructed (partially within the input dataset) that was of sufficient detail to provide a 3D (not 6D) simulation of the vehicle. To fit on core, the model was

[3] A linear tangent law has the form: $\tan \theta = a + bt$, where θ is the angle between the thrust vector and the local horizontal, a and b are constants, and t is time.

only as complex as necessary to determine payload and trajectory characteristics. The model allowed us to predict the performance of the ~360,000 lb GLOW vehicle to less than 1 lb. This was verified by post-flight analyses. The vehicle model could be broken into as many as 100 segments. Base pressure/thrust/Isp/drag models for the Atlas were loaded as functions of Mach number. Vacuum thrust, flow-rate, and engine exit area could be constant or could vary linearly with time for each segment. If needed and with little effort, thrust and flow-rate could vary nonlinearly by employing a polynomial. Hardware weight could be dropped at the end of each segment. Constraints could be applied to each segment independently. Engine startups and shutdown times were typically very short and were found to be sufficient to model these with constant flow rates and thrusts. Flow rate was determined from data acquired from engine testing and thrust was determined by calculating the total impulse provided by the startup and shutdown flight data. It was verified that for such short segments, a more detailed approximation did not affect the payload determination.

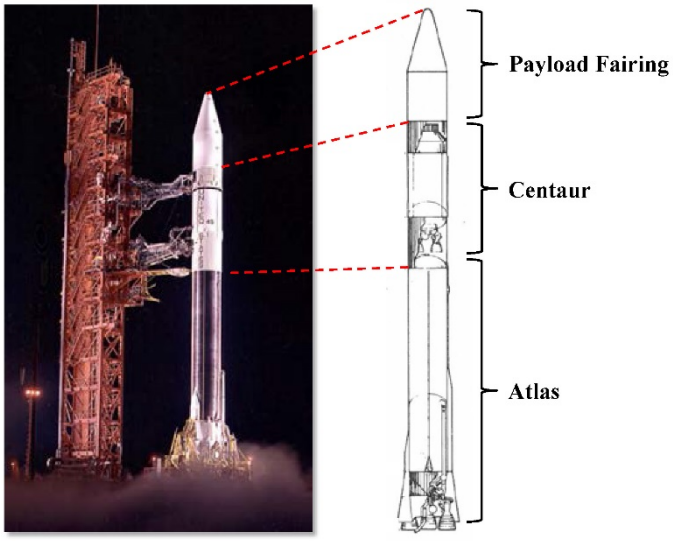

Figure 8.—AC-45 (HEAO-1) at Complex-36B with Vehicle Diagram

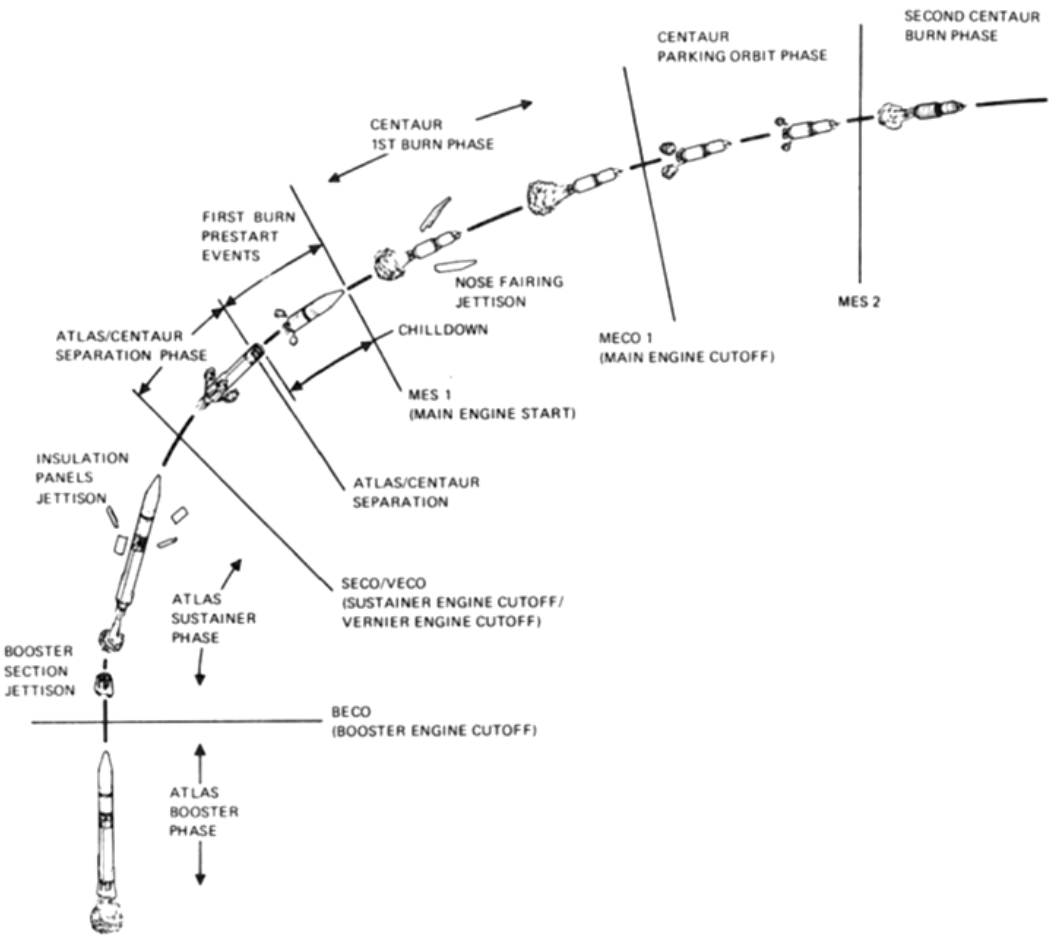

Figure 9.—Atlas/Centaur Ascent Profile

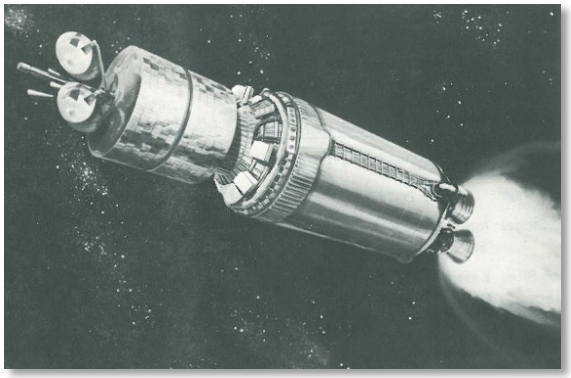

Figure 10.—Artist View of Centaur and Intelsat IV

6.2 Titan/Centaur

The Titan boosters were built by Martin Marietta in Denver, Colorado. There were two Titan configurations on which Centaur rode: the IIIE and IV. The Titan IIIE/Centaur was developed in the early 1970s and flew from 1974 to 1977. The Titan IV/Centaur flew for the U.S. Air Force (USAF) 16 times from 1994 to 2003, though only one was a NASA launch (and the last mission managed by LeRC in 1997—Cassini). (There was one other Titan variant; see Sec. 6.4.) The Titan IIIE/Centaur was a four stage vehicle based on its heritage as another of the nation's ICBMs. At liftoff, the two monolithic Solid Rocket Motors (SRMs) lifted the stack with a GLOW of ~1.4 M lb. The SRM's flew until burnout and were jettisoned. The first core stage engines were ignited in flight just before SRM tail-off and continued until exhaustion shutdown. The payload fairing covered the Centaur stage as well as the payload, and was jettisoned during stage one. Following a 'fire in the hole' separation maneuver to jettison core stage one, the single engine core stage two flew until either guidance or low level sensor commanded shut down. Following jettison of core stage two, Centaur flight operation began. Centaur shutdowns were by guidance command. For Titan IIIE/Centaur stack, Centaur avionics steered all stages. Years later, however, for Titan IV, this would not be the case, as Titan IV had its own avionics suite. Figure 11 shows the Titan IIIE/Centaur vehicle. Titan IIIE/Centaur had the capability to perform variable launch azimuth ascent, which its successor (Titan IV) years later lacked.

Most of the Titan IIIE/Centaur model was analogous to the Atlas input dataset and will not be repeated here. There was one major system which was quite different than Atlas: the two monolithic SRMs. These propulsive stages, ignited at liftoff, could not be modeled as a liquid propelled stage in vacuum. DUKSUP modeled their operation by utilizing a "table look-up" approach, where the SRM's performance was given as functions of Mach number. Two of the launches (Voyager) also included a large solid "kick" stage. Two drag models for the Titan had to be incorporated, with and without the solids.

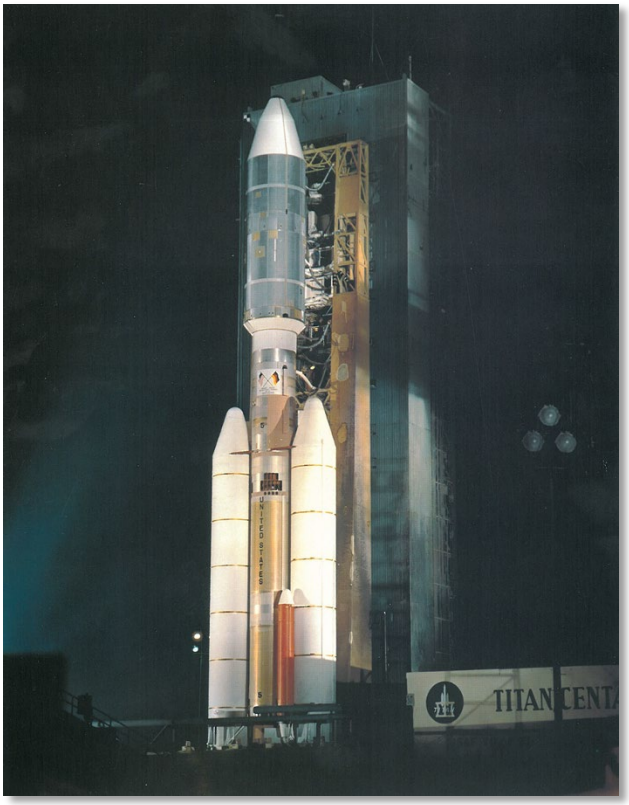

Figure 11.—Titan IIIE/Centaur at Complex-41 (CCAFS)

6.3 Shuttle/Centaur

The Space Shuttle was to serve as a booster for Centaur, the first time Centaur would be integrated with a manned vehicle. The Centaur diameter was widened from the 10 ft version on Atlas and Titan, to 15 ft to match the Shuttle cargo bay diameter. The Shuttle system was built by Rockwell International Space Systems Group, Downey, California, along with other major contractors Thiokol and Martin Marietta. However, since Centaur was integrated within the cargo bay and classified as a "payload" rather than a major "element", the NASA Johnson Spaceflight Center (JSC) served as the interface organization for LeRC.

In the early 1980s, LeRC was assigned to manage, develop, integrate, and launch Shuttle/Centaur, a joint NASA-USAF development program. There would be two new versions of the new Centaur to be launched from the Space Shuttle: the smaller "G" version primarily for national security missions and the larger "G-Prime" version designed primarily for NASA interplanetary missions. Shuttle/Centaur and its payload were to be cradled by the Centaur Integrated Support System (CISS) within the Shuttle cargo bay. The CISS and the forward attach points supported the upper stage and payload from Space Shuttle mating through cargo bay deployment on-orbit. Once in orbit, with cargo bay doors opened, the CISS rotated the Centaur/payload to the de-

ployment angle and the Centaur/payload were deployed. Shuttle/Centaur would then maneuver to a predetermined safe location, fire its engines, and inject the spacecraft into its final orbit. This was quite unlike any prior Centaur operation, and it represented challenges for most of the Centaur program, including the trajectory design by DUKSUP. Figure 12 shows an artist's rendition of the deployment of a spacecraft from the Shuttle cargo bay, integrated with a Centaur stage. Figure 13 illustrates a typical Shuttle/Centaur flight profile (Ref. 12).

Because Centaur would not be steering the Shuttle booster from launch to LEO, it had to rely on data from Shuttle, including accurate state vectors. Thus a new "on-orbit" version called SHUTSUP (much of which was scavenged from the earlier "space tug" version) was developed which could start the trajectory design and optimization from a Shuttle orbital state vector. None of the Shuttle phase flight from pad to LEO was performed by SHUTSUP, so it was able to easily determine the payload for these missions as well as the trajectory characteristics, since lacking the sub-orbital phases of flight greatly simplified the code and its execution. But there was one major complication unique to this booster which affected both Earth orbital and interplanetary missions, and had profound effect on how SHUTSUP was run: the particular orbit (revolution) on which the Centaur/payload could be deployed was allowed to vary. This not only meant that Centaur boiled off propellants, losing capability during each revolution, but also that the orbital node would regress for all deployment revolutions following the intended one. For Earth relative missions this meant a penalty even for an on-time launch should deployment be delayed. For interplanetary missions, not only was there a penalty for on-time launch with late deployment, but for a "second day deployment", this also held the possibility of significantly different interplanetary targets. (This was particularly true for the Galileo mission, where "first day deployment" targeted a main belt asteroid 29-Amphitrite for flyby, while a delay until "second day deployment" meant a significantly different hyperbolic target vector which was "non-Amphitrite" directed.) This was the impetus for a couple of significant modifications to SHUTSUP subroutines, the creation of stand-alone utilities to support missions on Shuttle/Centaur (such as multiple launch window analysis), and significant changes in how the mission was represented in the input dataset (see Sec. 9.4).

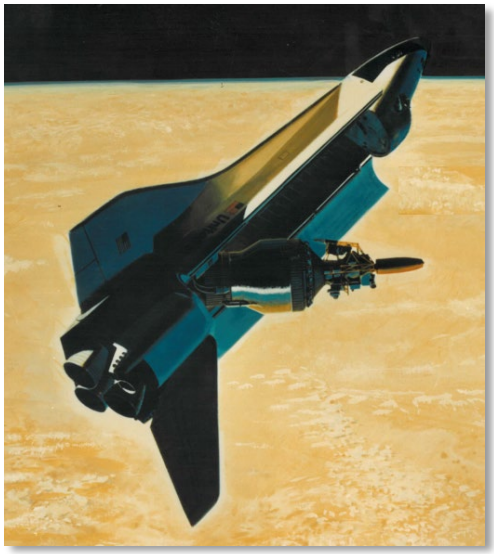

Figure 12.—Artist Rendition of Shuttle/Centaur (Deployment from Cargo Bay)

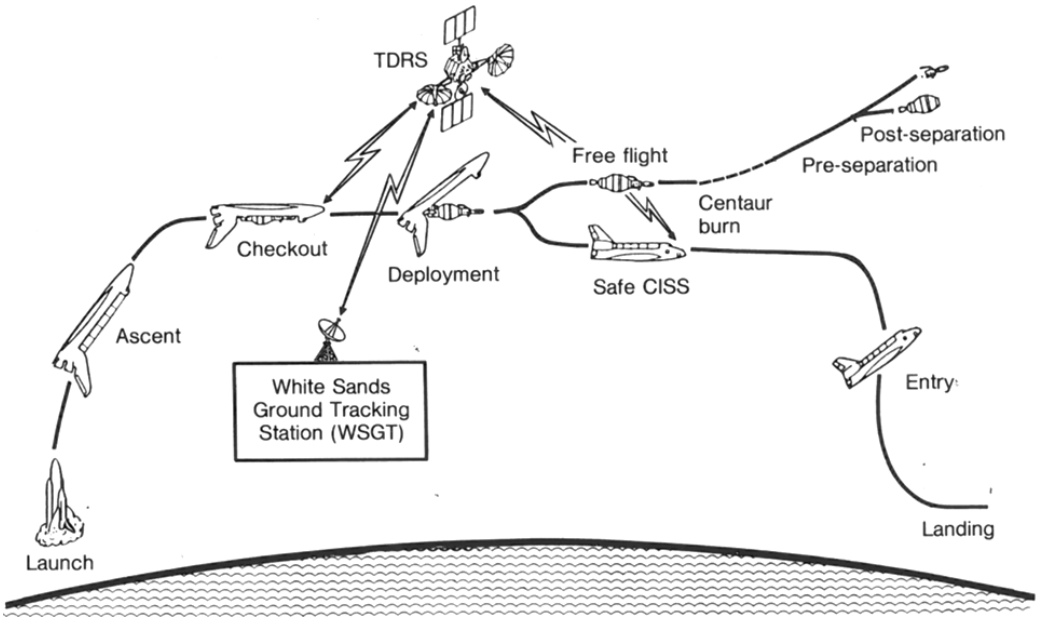

Figure 13.—Shuttle/Centaur Ascent and Mission Profile

Due to the Challenger failure of January 28, 1986, the Shuttle/Centaur project was cancelled on June 19, 1986. President Reagan decreed that there would be no more Shuttle launches for interplanetary or Earth orbit spacecraft. There proved to be exceptions—Galileo, Ulysses, and Magellan—all of which were re-assigned to the Shuttle-launched Inertial Upper Stage (IUS). The planned Shuttle missions for other spacecraft were cancelled. LeRC was asked by NASA Headquarters (HQ) to find "rides" for these spacecraft. Over the course of a year DUKSUP was used to match the requirements to available launch vehicles. Recovery from Challenger was difficult and the United States did not launch any nonmilitary spacecraft for almost 3 years.

6.4 Titan 3/TOS

The Titan 3/Transfer Orbit Stage (TOS) was the only LeRC-led launch vehicle without the Centaur for which DUKSUP was used. The Titan 3/TOS was a short-lived commercial Titan booster with a TOS solid upper stage (a derivative of the IUS, but with a less capable avionics suite and smaller propellant loading). It was used by LeRC only once in 1992 to launch the Mars Observer (MO) spacecraft.

7.0 The Flight Constraints

7.1 Tower Clearance and Initial Angle of Attack

The vehicle was constrained to fly vertically for approximately the first 15 sec of flight in order to clear the tower. We modeled this by constraining the thrust vector to be along the local vertical for 15 sec. After that initial phase, the vehicle inertial velocity vector was instantaneously "kicked over" by usually less than a tenth of a degree in the launch azimuth direction. Until the end of the "zero-angle-of-attack" phase, the thrust vector was constrained to be in the launch azimuth plane. The thrust vector was determined by projecting a unit relative velocity vector into the launch azimuth plane and defining that projection as the thrust direction until the vehicle could tolerate an angle-of-attack. The actual angles-of-attack were quite small and this was in fact the way the vehicle flew. We found that simulating this was necessary. If we constrained the thrust vector to be along the relative velocity vector, as kick angle changed, so did the inclination at the end of the first Centaur burn, which was not characteristic of the vehicle and was unacceptable. The vehicle steering forced the thrust vector to be in the launch azimuth plane through the first Centaur burn. We could simulate that by optimizing the thrust direction subject to that constraint. Since in most cases, final inclination was specified, at some point in the trajectory, yaw was necessary to satisfy final inclination.

In most cases where the first Centaur burn inserted itself into LEO (or for an interplanetary launch where there was a single burn), the optimum kick angle would result in a trajectory where aerodynamic heating was in excess of the capability of the vehicle, the kick angle was varied using a simple iteration to satisfy the heating limitation of the vehicle. In some cases where the injection altitude was high, the kick angle had to be optimized to maximize payload. This was accomplished by using a parametric scheme where the payload as a function of kick angle and the optimum kick angle was obtained by locating the top of a parabola. This was not an elegant method, but it provided an accurate determination of the optimum kick angle since payload did not vary greatly with kick angle due to the fact the calculus of variations solution to the steering algorithm resulted in small payload variations. DUKSUP also had the capability of optimizing kick angle, but if the heating for the optimum exceeded the limit, the code would revert to satisfying the heating limit.

7.2 Heating Constraints: Jettisoning Hardware on "qV"

The Atlas/Centaur, and later on Titan/Centaur, vehicles were designed such that insulation panels (on Atlas/Centaur only) and payload fairings were jettisoned in flight. In those projects, insulation panels protecting the Centaur hydrogen tank and a nose fairing protecting the payload from aerodynamic heating were jettisoned. At the beginning, they were jettisoned on fixed times from some prior event, such as when an earlier stage had been jettisoned. These times were selected to make certain that aero/thermal loads had decreased to acceptable levels before panels or fairings were jettisoned. They were chosen conservatively to make sure that the aero/thermal loads were "safe." Eventually, as the projects matured, it was realized that payload could be increased by jettisoning panels and fairings on instantaneous heating rate (sometimes called "free molecular heating rate") or $qV \equiv (\frac{1}{2}\rho V_r^2)V_r$, where q is the dynamic pressure, ρ is atmospheric density in mass per unit volume, and $\underline{V_r}$ is relative velocity, referred to as "V" (Ref. 8). The qV has units of energy/unit area/unit time; frequently quoted as lb/ft-sec (ft-lb/ft^2/sec), or Btu/ft^2/sec. The components of qV were calculated on board the vehicle in the flight program. Relative velocity and altitude are calculated from integrating the measured acceleration. Atmospheric density, or ρ, was a pre-stored polynomial function of altitude, accounting for 3-σ high dispersions in atmospheric density to be conservative. When jettisoning on qV, Atlas/Centaur panel jettison was enabled at 25 sec after BECO to ensure that the vehicle was stable. The guidance loop closed at BECO + 8 sec, and the guidance cycle was also 8 sec. Two cycles were required to ensure vehicle guidance stability. The qV limit was actually reached before 25 sec for some lofted trajectories, so jettison in those cases occurred at 25 sec rather

than on qV (Ref. 8). Panels were jettisoned on qV for some less lofted trajectories. Nose fairings were jettisoned on qV for Atlas/Centaur missions, as well as on the Titan IV/Centaur Cassini mission.

Clearly, the earlier any weight can be jettisoned from a vehicle, the greater the payload (the propellant required to raise the jettisonable weight to a higher energy level was now available to increase the final payload). A trajectory can be reshaped to climb more quickly, i.e., "lofted" above the optimum unconstrained by qV, such that the desired qV for jettison was reached at an earlier time. In this case, the panels were then jettisoned earlier, thus increasing payload. Lofting increases potential energy at the expense of kinetic energy, and jettison occurs at a higher altitude (lower ρ) and a lower relative velocity. It is important to note that if the same trajectory was flown as in the case with no qV constraint (i.e., no lofting), jettison on qV rather than on a later time would result in a very small payload gain. Flying a lofted trajectory was critical to allow the jettison to occur earlier, thus realizing the payload gain. To realize the full gain, the guidance equations might have had to be required to accommodate discontinuities. The trajectory optimization task was to determine the trajectory reshaping that will yield maximum payload. The optimum "trade" was between being able to jettison the hardware earlier, thus increasing payload, and reshaping the trajectory away from the optimum (without the constraint), which decreases payload. The qV constraint was included in the CoV formulation of the problem and solutions were obtained. As expected, the trajectory was lofted (the trajectory was higher at any time point in the trajectory) and the relative velocity was lower – and payload was increased. The characteristics of the thrust vector history were instructive. CoV theory "proved" that as long as there are no constraints, the thrust vector and the rate of change of the thrust vector are both continuous. In the qV constrained trajectories, at the point of hardware jettison, the thrust vector and the rate of change of the thrust vector were discontinuous. The theory also showed that if the constraint was only a function of the vehicle position vector, the thrust vector was continuous, but the rate of change was discontinuous. If the constraint was only a function of the inertial velocity, then the opposite was true. In the case of the qV constraint, the constraint was a function of both position and velocity, so both the thrust vector and the rate of change of the thrust vector are discontinuous, by theory. Examination of the trajectory demonstrated that this was indeed the case. In an environment dominated by senior test engineers who had spent much of their careers in the rigor of National Advisory Committee for Aeronautics (NACA) test cells, it was not sufficient to assert that the theory had provided the maximum payload meeting the qV constraint. It had to be demonstrated. First, it was shown that as the weight of the hardware to be jettisoned was increased, the jettison occurred earlier. It was also shown that as the jettison qV was reduced, the jettison occurred later. Furthermore, if the hardware weight jettisoned were split between hardware jettisoned at a fixed time and the remainder jettisoned on qV, the trajectory shape (and payload) would revert to the solution where jettison occurred at a fixed time. The trajectory would gradually approach the trajectory for jettisoning on time as the percentage of hardware jettisoned on time, not qV, approached 100 percent. The time for jettison on qV would move back toward the fixed time as the percentage of weight jettisoned on qV diminished, and the discontinuities would disappear, reproducing the unconstrained trajectory.

Obtaining the optimum constrained trajectory was just the beginning; it had to be implemented in the vehicle guidance system to obtain the payload benefit. If the entire benefit were to be realized, then a discontinuity in the thrust vector and rate of change of the thrust vector had to be introduced into the guidance equations at the jettison point. This meant that some kind of "optimized," intermediate state vector would probably be required at the point of hardware jettison. This was not easy to implement, so we were asked to assess the payload impact of eliminating these discontinuities. For Atlas/Centaur, the payload difference was small because the insulation panels weighed only about 1200 lb and the nose fairing only about 3000 lb (Ref. 8). So, the qV constraint was implemented on the vehicle without the discontinuity and the payload gain, mission dependent, was on the order of several tens of pounds. This was considered to be a very significant increase for Atlas/Centaur missions, justifying the effort.

With the development of the Titan IV program in the mid-1980s, payload fairings grew considerably in length and mass. The longest fairing configuration, built to accommodate the Centaur G-Prime vehicle which was adopted from the recently cancelled Shuttle/Centaur program, was 86 ft long and weighed in excess of 14,000 lb. The Titan deck of DUKSUP was modified in 1986 to enable the simulation of the new Titan IV vehicle, including jettisoning its new, large fairings on qV. This resulted in payload gains in the 100's of pounds (Ref. 8). However, this made the convergence of the two-point boundary value problem more sensitive because of the modeling of the atmosphere at those altitudes. A "smoother" atmospheric model would have solved the problem. While LeRC was not brought into the Titan IV development by the USAF, this upgrade to DUKSUP was not in vain. The CRAF and Cassini missions were green-lighting of pre-Phase-A study in the early 1990s. The analysis challenge was very similar but not identical to the analysis for the Voyager missions. Unlike Voyager's Titan IIIE/Centaur, Centaur guidance system did not control Titan IV, which complicated the implementation of the variable launch azimuth requirements for the interplanetary missions. LeRC

used the enhanced Titan deck to perform leading edge analyses of those missions. After LeRC was given Authority to Proceed to prepare to launch the Cassini mission, this significantly enhanced version of DUKSUP was used to create trajectories optimized for payload fairing jettison to a secure large performance improvement. The launch of Cassini would also be the last mission for DUKSUP.

By jettisoning on qV, the vehicle payload was maximized with no additional risk, as the trajectory was optimized to jettison the panels and the nose fairing when the aero/thermal environment met constraints. For example, a thermal constraint could result from an analysis of the heat loads that an avionics box can tolerate. The decision to implement qV as a jettison criterion for the Atlas/Centaur panels and fairing and the Titan/Centaur fairing reflected the general project policy to develop the capability of using optimum trajectories for all missions. Further, jettisoning on qV ensured that this hardware was jettisoned neither before nor after the actual, physical constraint was satisfied. In addition, having a totally optimized trajectory responsive to functional constraints, such as jettison on qV, ensured that the maximum energy was available to recover from in-flight dispersions or anomalies. This approach maximized the probability of mission success (see Sec. 11.7). Trajectories were optimally designed to be consistent with whatever constraints were applicable to a particular mission. This approach gave project management accurate data which allowed it to make informed decisions about the relative merits of various potential vehicle improvements, along with their associated costs and risks.

7.3 Range Safety Constraints (Early-in-Ascent and Downrange)

The Range Safety organization at the Eastern Space and Missile Center, Patrick AFB, Florida, has a mandate to fly any mission with minimum risk to people on the ground. They are fundamentally involved with evaluating the launch space from early in the mission planning process, and they make the final decision on the acceptability of planned launch trajectories. The launch space may be complex with mission advantages being weighed against varying risks. There are several areas of Range Safety's responsibilities which are directly affected by trajectory design. Two of these areas are the early-in-ascent overflight of land masses while jettisoning hardware during nominal flight and the instantaneous impact points (IIP) of debris in the event of a catastrophic failure.

In some cases, the thrust vector had to be constrained to the launch azimuth plane to satisfy Range Safety overflight of populated land mass constraints. Some missions had final inclinations which optimally required launch azimuths north of 35° or south of 108°. These limits were imposed because as seen in the vehicle description: booster engines, insulation panels, nose fairing, and the sustainer were jettisoned early-in-ascent. Without constraint, these jettisoned hardware posed threats to the islands in the Caribbean or even the South American mainland. The vehicle was only allowed to yaw north or south when these hardware items had been safely separated. DUKSUP accommodated this constraint through fixed launch azimuth control and specifying when in the CoV out of plane steering was permitted. Only in a latter version of the code was a variable launch azimuth capability incorporated (see Sec. 11.10). Figure 14 illustrates a DUKSUP-generated IIP for the Cassini launch (early in ascent) designed to skirt the Range Safety destruct lines protecting the Caribbean islands and the coastline of Brazil (Ref. 13).

Another Range Safety constraint was sometimes imposed on the IIP's. In maximizing payload, the ground track frequently passed over the budge of Africa (Liberia). By constraining the trace to pass south of that bulge, the probability of impacting Africa was reduced with little loss of payload. Consequently, the Range asked us to examine the payload penalty for moving a GSO IIP trace south to reduce its dwell time over Africa. By moving the launch azimuth south to ~ 98° (rather than ~ 92°), and constraining the thrust vector to be in the launch azimuth plane until sometime in the Centaur burn, the IIP was moved about 150 miles off the African "bulge" (see Figure 15) (Ref. 14). Dwell time over Africa was reduced from about 10 sec to 1 sec (i.e., the trans-Africa overflight only) (see Figure 16) (Ref. 8). The cost in payload penalty was only about 5 to 10 lb (Ref. 8). Though the penalty was noticeable, it was accepted and other remaining variables were re-optimized to accommodate this constraint. Project management decided that this loss was justified by the reduction in Range Safety risk. DUKSUP did not have IIP as part of the variational optimization, though it could have been incorporated in principle. The Combined Release and Radiation Effects Satellite (CRRES) mission of 1990 did use the ground tracking station constraint to indirectly "steer' the trajectory so that the IIP fell outside the 150 nmi limit off the African bulge (see Figure 15) (Ref. 14). Though it was a 'work-around' approach, it did match contractor simulations quite closely.

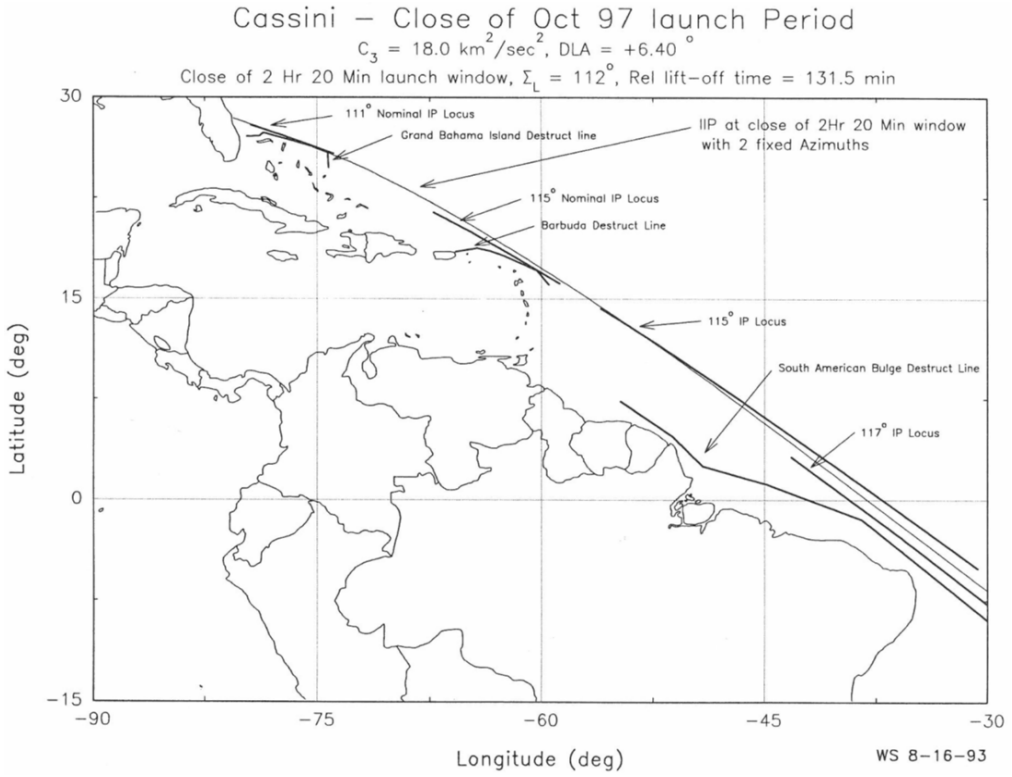

Figure 14.—DUKSUP IIP Data for Early Flight Phases of Cassini and Range Safety Destruct Lines

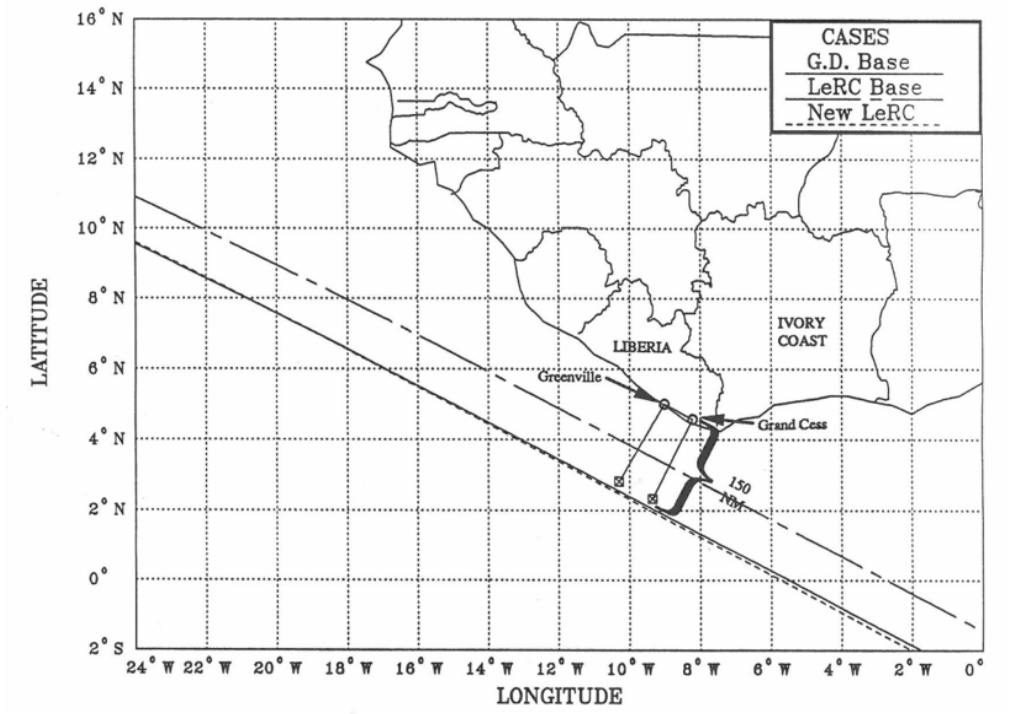

Figure 15.—DUKSUP IIP Data for CRRES Near African Bulge

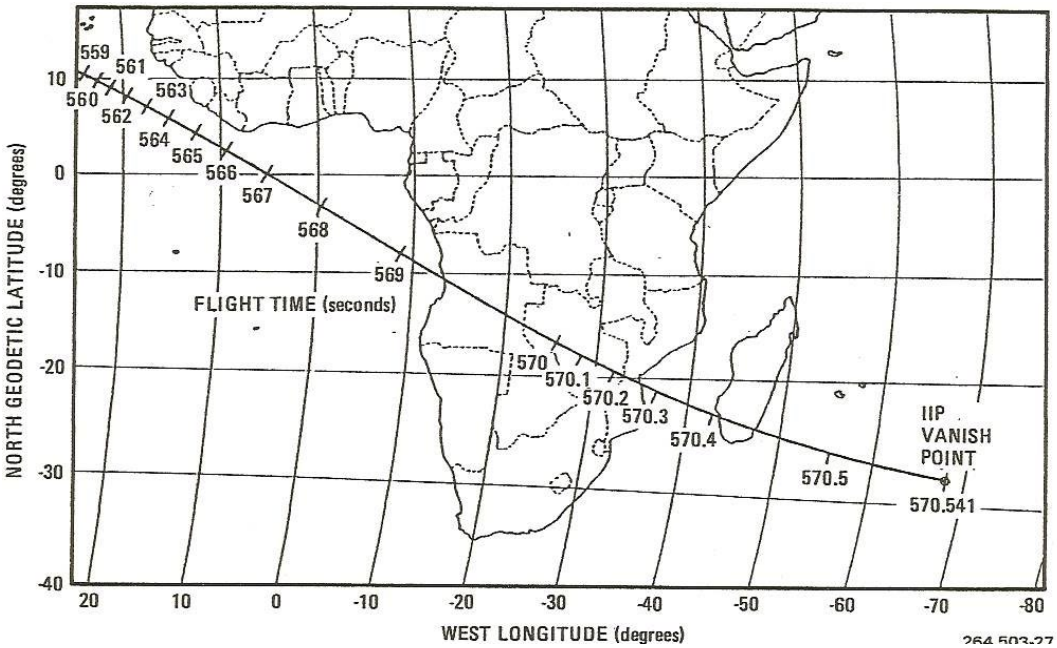

Figure 16.—Centaur IIP Data Near Africa (Flight Time Shown in Seconds)

7.4 Ascension Island Tracking

Before the Apollo program was terminated, tracking ships were available to the Atlas/Centaur program to obtain telemetry data for the Centaur second burns by positioning a vessel within line-of-sight. But the ship was expensive. Advanced Range Instrumentation Aircraft (ARIA) were not only expensive, but also unreliable, and the data was unavailable in real time. After Apollo was terminated, ARIA became the only option to obtain second burn data. (This was before the advent of Tracking Data Relay Satellite System (TDRSS)). We began to investigate whether it would be possible to reduce costs and/or risks by modifying trajectories. Ascension Island (in the south Atlantic) was available as a data recovery station, but for geostationary missions, it was not located such that the Centaur second burn, or indeed any part of the trajectory, was visible from the island. Obviously, if the Centaur and payload were placed in a high enough parking orbit, Ascension Island would be able to "see" it and receive the desired data, but the payload penalty would have been unacceptably high. Clearly, there was an optimum solution that would move the orbit nearer to Ascension Island, and would use a more elliptical parking orbit to raise the orbit altitude such that the vehicle could be seen. The only constraint on parking orbit would be the perigee altitude; the other orbit elements (apogee, argument of perigee, inclination, node, and injection true anomaly) would be free to be optimized to provide tracking from Ascension. The optimum solution would explicitly include a Centaur second burn tracking constraint. We implemented this change in the code and evaluated the resulting payload decrement.

The project required data from 40 sec prior to Centaur second burn until 135 sec following the burn to observe Centaur shutdown and spacecraft separation. These constraints would be explicitly imposed on a trajectory that began at liftoff and terminated at injection into the final GSO (i.e., the entire ascent was optimized from beginning to end). Imposing such a constraint was acceptable for GSO missions because the performance penalty was small. It was not practical for some other missions because unfavorable geometry resulted in large performance penalties. Geostationary orbit missions are circular at geostationary altitude and slightly inclined to the equator, either posigrade or retrograde, to minimize stationkeeping propellant usage over the spacecraft lifetime. The spacecraft solid motor burn could occur after more than one apogee passage (i.e., after several revolutions in the transfer orbit). All these factors were included in the optimum solution. Specifically, the tracking constraints were explicitly included in the CoV formulation of the problem. We constrained the trajectory such that the vehicle was 5° above the Ascension Island horizon from 40 sec before the second Centaur burn until after spacecraft separation. This required a slightly more southerly launch azimuth, a slightly more elliptical parking orbit, a slight increase in parking orbit inclination, and a slight change in parking orbit argument of perigee and node. Figure 17 illustrates several tracking-constrained and unconstrained trajectories, with the position of Ascension Island noted in the bottom left corner. All of these were optimized to maximize payload. The penalty for the constraint was relatively modest since there were so many orbit characteristics that could be "tweaked" to satisfy the constraint. A solution was obtained, and the result was that the tracking

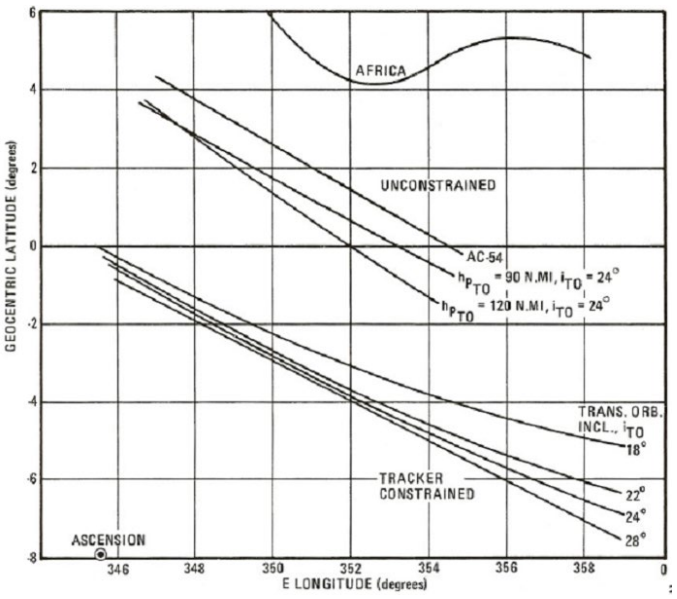

Figure 17.—Ascension Island Tracking Constrained Trajectories

requirements of the project could be met by constraining the trajectory with a reduction in payload of 29 lb (Ref. 8).

By implementing this approach, the Atlas/Centaur project had a more reliable tracking option. Typically, we supplied constrained and unconstrained trajectories to the spacecraft project. If it was advantageous to them to give up a little payload to avoid the cost of the ARIA, they could make the decision. If the ARIA could be avoided, the cost of planning for and the use of the ARIA could be avoided. The spacecraft project was usually willing to forego 29 lb of payload to avoid paying for the ARIA (Ref. 8). This trajectory design was implemented in the Centaur guidance equations by employing two targets, one at the end of the first Centaur burn and the second at the end of the second Centaur burn. These trajectories were flown many times, and the predicted payload penalty was verified by flight experience. In addition, verification of the tracking constraint was demonstrated by the availability of real-time tracking and telemetry data from Ascension at precisely the predicted times.

7.5 Elliptical Parking Orbits

The characteristics of the trajectory to circular parking orbit suggested that, perhaps, a circular parking orbit was not optimum even for GSO missions, or maybe, for any two-burn missions. The altitude plot versus time of the Atlas/Centaur had a "roller coaster" characteristic, i.e., the vehicle climbed above the desired final injection altitude and then dived, permitting the addition of ΔV at higher velocity, the most efficient way to increase energy. Further, forcing the flight path angle at burnout to be zero was not optimum. A trajectory with simply a perigee altitude constraint on the parking orbit was obtained and, surprisingly, the payload benefit to GSO was 30 lb (a significant increase for Centaur GSO missions). The payload increase was significantly more for two-burn interplanetary missions (for some missions (Mars Observer) 100's of pounds were gained). The parking orbits were slightly elliptical with apogee altitudes of ~ 130 nmi and perigee altitudes at 90 nmi (Ref. 8). The apogee varied somewhat with the missions and the length of the coast. Again, this was "free" payload. There was no cost to the project and no added risk. All that was required was to optimize the orbit elements of the parking orbit for each mission, which was a trivial effort using a CoV code like DUKSUP.

Directly related to this was the how the perigee heating constraint was incorporated. After inspecting the qV at perigee of the parking orbits and transfer orbits for GSO missions, where the perigee of the parking orbit was constrained to be 90 nmi, we noted that the qV at perigee of the parking orbit was lower than the qV at perigee of the transfer orbit. This was clearly the result of the great difference in the relative velocities at perigee of the two orbits. After further examination of qV, it was found that if the perigee constraint for the elliptical parking orbit were reduced to 80 nmi, the qV at perigee of the parking orbit was approximately equal to the qV at perigee of the transfer orbit (Ref. 8). The perigee of the transfer orbit had to be constrained to 90 nmi to maintain the qV. This was implemented and became standard for such missions, since it added no complexity to the mission planning, no changes to the vehicle guidance software, and added no risk to the mission. Approximately 30 lb was added to the payload in GSO, at essentially no cost.

7.6 Full Centaur Tanks or Fixed Initial Weight

DUKSUP could be run in two distinct ways with respect to Centaur propellant loading. To maximize the payload for any specified mission target (which was supplied by the spacecraft project), the lift-off weight was varied until all the usable Centaur propellants were consumed to achieve the target. Usable Centaur propellants were determined by adding the propellant "burned," the flight performance reserve (however that was calculated), and any additional reserves. Filling the Centaur tanks was performed by varying the lift-off weight until the desired Centaur propellant load (including reserves) was achieved. A simple iteration was employed to fill the tanks. DUKSUP could also be run where the spacecraft weight was fixed (as was in fact the entire liftoff mass of the Atlas/Centaur stack), and the amount of propellant excess was then calculated. The first technique was useful early in the study phase of a mission, while the latter was used during the year or so preceding launch where the mission design was mature, spacecraft design was finalized, and hardware was being manufactured.

8.0 The Hardware Platforms

The computers on which DUKSUP resided are an important part of the story. As was discussed in Sections 5.3 and 5.4, the structure of DUKSUP and how it operated was driven dramatically in the early years by the computer's limitations of that era. Table 3 contains some comparison data for the mainframes on which DUKSUP ran. Data is from various sources including recollections from individuals, the manufactures, and third parties (Refs. 15, 16, 17, 18, 19, and 20). After computers memory and speed surpassed the primary needs of the user running the code (i.e., there was not much practical difference to a User if a run took 1 or 0.1 sec), the ever-increasing capabilities of the machines mattered only to unique tasks (like the multiple revolution Shuttle/Centaur missions with the need for 1,000's of runs per mission). It was never judged to be worthwhile to restructure DUKSUP to take advantage of the immense capabilities on the new machines during the end of its use in the 1990s, though some preliminary planning did take place at that time for creating a new CoV trajectory optimization code not based on DUKSUP.

8.1 IBM 704 and 7094

The computer available to us in the spring of 1962 was an IBM 704, which had an 8,000 36 bit word memory. That was totally inadequate to the task at hand. Sometime after that LeRC got an IBM 7094, which had a 32,000 36 bit word memory (Figure 18) (Ref. 21). We thought that we had died and gone to heaven. Even so, this computer was small and very slow. If we had had a larger and faster computer, our coding would have been much simpler and "rational". Computer time was precious and our time almost worthless in comparison. We were fortunate to get one or two, 2-min "runs" during working hours. You might get a 5 min run during the day and one 20 min run at night. Punch cards were the only means of accessing the computer. A key-punch error might be hard to find and lose us a day or two of access. Again, access to the machine language dumps was invaluable for finding errors. An advantage of the situation was that we had a lot of time to think and to plan our machine submittals. The environment precluded "shot gunning" a problem. Computer time and access demanded that we make the best use of the time and limited availability of the machine.

8.2 UNIVAC 1100/42

Purchased in 1975, the UNIVAC 1100 was a marked improvement over its predecessor for DUKSUP. LeRC continued to run DUKSUP on this mainframe until 1982.

8.3 IBM 370/3033

Acquired in 1980, this mainframe had a TSS/370 operating system running on an IBM 3033 hardware platform, arguably the most capable machine available at the time. Figure 19 is a photo of the IBM 370 operator's console and tape drives at LeRC. It was the replacement for the IBM 360/67 which had been in use from 1966 to 1980. DUKSUP was brought up to run on this mainframe from the UNIVAC 1100. This was done after converting UNIVAC symbol codes to IBM symbol codes by way of manipulation of the INPUT subroutine to recognize the IBM machine language.[4]

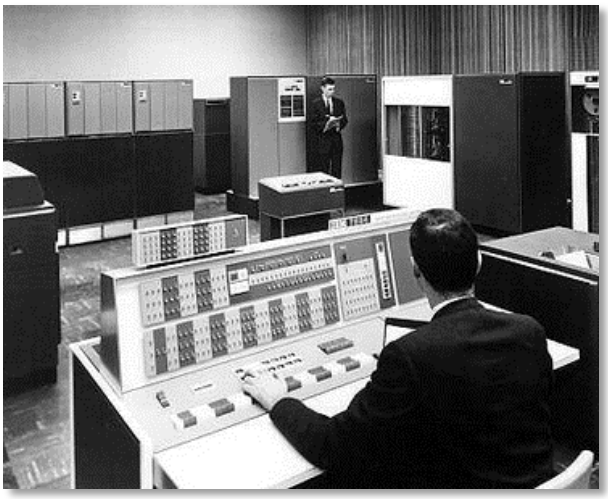

Figure 18.—IBM 7094

TABLE 3.—CHARACTERISTICS OF COMPUTER MAINFRAMES WHICH RAN DUKSUP

	IBM 704	IBM 7094	UNIVAC 1100/42	IBM 370/3033	Cray-1S	Cray-XMP
DUKSUP period (approximately)	1962 to 1964	~1964 to 1975	1975 to 1982	~1980 to 1983	1983 to 1988	1988 to 1998
Number of words (memory)	8,000	32,000		64,000	Up to 4,000,000	Up to 8,000,000
Bits per word (memory)	36	36	36	32	64	64
Typical number of runs/day		1 to 2			100's	~1,000
CPU time /DUKSUP run		~2 min			~ 2 sec	~0.25 sec

[4] Personal communication/email with Wickenheiser, T.J., NASA GRC (retired).

Figure 19.—IBM 370/3033 at LeRC

Figure 20.—Cray-1S at LeRC

8.4 The Crays (1S and XMP)

In the early 1980s, LeRC received its first supercomputer, the Cray-1S. Figure 20 is the Cray-1S at LeRC. Because of the Cray's tremendous capability relative to other machines and job requirements, it was paired with the existing IBM 370 which then served as the Cray's front end, prioritizing jobs and otherwise interacting with its human users. Run times shortened dramatically and quantity of runs increased exponentially. A typical Shuttle/Centaur run (which lacked an atmospheric portion) took ~ 2 sec of CPU time. Now 100's of runs could be done during the day; in the evening, 1,000's. The Cray-1S's 64 bit words challenged the need for double precision FORTRAN variables and operations.

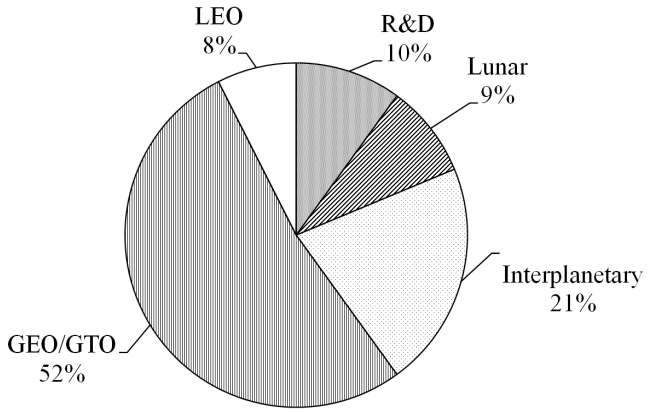

Figure 21.—Centaur Missions by Destination

In the late 1980s, another upgrade arrived. The Cray-XMP brought still more speed and memory. Now an Earth to orbit trajectory run was of the order of ¼ sec of CPU time. It was likely that 1,000's of trajectory runs could have been done with this mainframe during the work day. But unlike the multiple revolution problems associated with Shuttle/Centaur, the nature of the more-traditional ELV missions did not command that type of effort. Other still faster mainframes came afterwards to LeRC (Cray-YMP and Convex) and then the first of the workstations (Sun), where DUKSUK took longer to execute but the user wait queue was shorter. The workstations brought the opportunity to restructure DUKSUP into various versions which could be separately and more efficiently compiled and linked for the ever increasing variety of launch vehicles LeRC was supporting during the 1990s. Despite these changes, the last mission to rely on DUKSUP and which actually launched (Cassini to Saturn) used the Cray-XMP.

9.0 The Missions

9.1 Overview

LeRC launched 80 Centaurs from 1963 through 1997. Except for the earliest missions, NASA LeRC management relied on DUKSUP for trajectory design and optimization for 65 of those flights (>80 percent). Figure 21 illustrates the breakdown by mission destination for Centaur. Over half of these missions DUKSUP analyzed were geostationary, with interplanetary as the next largest destination. Figure 22 is a list of all missions which flew in which trajectories designed with DUKSUP were utilized. The following is a brief description of how these mission types influenced DUKSUP operation.

9.2 Early Mission Types Not Supported by DUKSUP (R&D Test Flights and Surveyor's Lunar Missions)

Before any operational missions, a series of seven test flights were flown by LeRC following MSFC's failure of F-1 (Ref. 5). These flights were Earth orbital with a single Centaur burn (except for the last two). General Dynamics, having done trajectory design for its Atlas ICBM program, continued in its role for the research and development test flights and the Surveyor missions, as LeRC capability was not yet in place. Most of these were single burn flights. These did not require the more complex trajectory planning for parking orbits, nor restart of the RL10 engines, and thus were inherently easier to perform. The test flights either lacked any payload (i.e., engineering instrumentation only) or carried a dummy payload mass simulator. Four of the seven test flights were successful.

When the Atlas/Centaur was assigned to LeRC in the fall of 1962, it had only the one operational mission assigned to it—the Surveyor series to soft land the first U.S. spacecraft on the Moon. Data gathered by Surveyor in situ was fundamental to the successful design of the Lunar Module used in the manned landings to follow. Surveyor launches took place between June 1966 and January 1968, and all seven were successful.

9.3 Geostationary Orbit (GSO) Missions

As was comprehensively discussed in the Section 5.2. but not repeated here, GSO missions are characterized by two coast periods—a relatively short one (~15 min) to the equator and the other longer one (~5¼ hr) from LEO to geostationary altitude—punctuated by two Centaur burns and a solid rocket motor kick stage to circularize at GSO (see Figure 3 and Figure 4). From an engineering perspective, these nonplanar, geocentric trajectories were much more complex that the preceding Surveyor lunar missions. It was for these missions DUKSUP was developed to perform the trajectory design and performance analysis.

In total, Atlas/Centaur launched 42 missions to geostationary/geo-transfer orbit (GTO) (overwhelmingly for the commercial Intelsat series, but also the other series, such as FLTSATCOM, Comstar, GOES, and ATS spacecraft).[6] Though representing a majority of the Centaur missions supported by DUKSUP (52 percent), the GSO missions had fairly routine trajectories which did not change much from launch-to-launch. The final targets used in DUKSUP were GSO energy and radius, a zero flight path angle, and zero inclination. The intermediate constraints imposed were sometimes radius, velocity, and sine of flight path angle; though with the increased usage of elliptical parking orbits, this was relaxed to a single qV constraint represented by a corresponding minimum radius of the parking orbit (see sections VII.B. and VII.E.). The duration of at least four (and sometimes five) phases of flight were optimized: the parking and transfer orbit coasts, and all Centaur burns (for Atlas/Centaur: 2, for Titan IV/Centaur and Shuttle/Centaur: 3).

One new major complexity to GSO and Molniya (somewhat similar to GTOs; highly elliptical, greatly inclined, and with specific arguments of perigee) missions (as well as interplanetary, see Sec. 9.4) occurred during the preparations for the Shuttle/Centaur G vehicle. Though never flown, these missions would have had to contend with multiple deployment revolution impacts to nodal regression, where even an on-time launch could still be penalized if deployment occurred on a later than planned revolution (see Sec. 6.3).

9.4 Interplanetary Missions

Centaur launched 17 interplanetary missions during LeRC's tenure. Every planet in the solar system was visited by a Centaur from a LeRC-led launch with the exception of Pluto.[7] Further, all first encounters with each planet were made by a Centaur-launched mission with the exception of Venus. In fact, with the exception of Mariners 2 and 4, every successful U.S. interplanetary mission until the late 1980s was launched on Centaur. Interplanetary missions were the second largest group (21 percent) of missions based on destination (see Figure 21). But unlike the GSO missions, they were anything but routine trajectories. Though their final targets in DUKSUP were always hyperbolic orbital energy (C_3) and declination, their values were significantly different depending on their planetary targets, and even between the launch opportunities for the same mission. Figure 23 illustrates the hyperbolic flight path of the spacecraft following Centaur injection burn (Ref. 22). Also shown are the declination (δ) and right ascension (α) of the outgoing asymptote of the hyperbola. Like the Surveyors, the interplanetary missions were high change in velocity (ΔV) and usually two burn (though some were single burn). The duration of three phases of flight were optimized: the Centaur burns and the intervening parking orbit coast. Relatively speaking, even though the hyperbolic targets changed due to the relative motion of the planets, these missions were simpler to model than GSO missions.

[5] Figure 21 "R&D" total includes one Titan IIIE/Centaur test flight which was analyzed by DUKSUP.

[6] The similar-to-GTO CRRES mission is included in this tally for convenience

[7] A Pluto flyby of a Centaur-launched (in January 2006) spacecraft will take place in July 2015, long after LeRC launch vehicle responsibilities ended.

Figure 22.—Missions Which Utilized DUKSUP-Generated Launch Trajectories

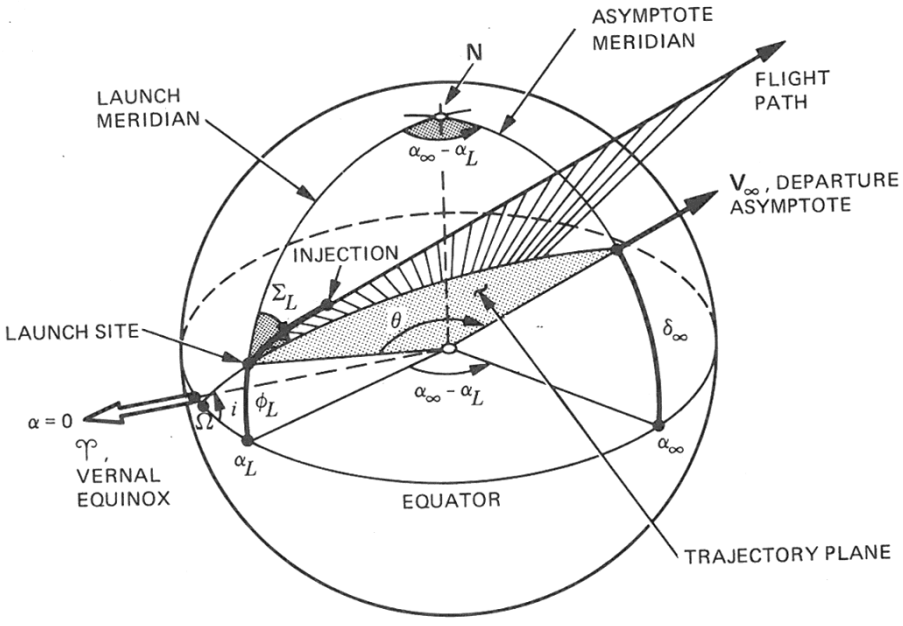

Figure 23.—Hyperbolic Geometry of an Interplanetary Escape Trajectory

Interplanetary missions in the late 1960s (Mariners 6 to 9 and Pioneers 10 and11) were launched by LeRC on Atlas/Centaur, but these were not initially analyzed with DUKSUP. Beginning in the 1970s, there were various interplanetary missions following NASA's decision to integrate the Centaur with the USAF's Titan IIIE booster and continue the Atlas/Centaur even though the Space Shuttle decision was looming in the near future. DUKSUP was then modified to support Titan. After the single test flight, all of the six Titan IIIE/Centaur missions were interplanetary: two launches of the Viking orbiter/lander mission to Mars (targeted for landing in 1976—200 years after the Declaration of Independence), two launches of Helios for the Germans into heliocentric orbit, and two Voyager missions to Jupiter and Saturn (and later on to Uranus and Neptune). Other Atlas/Centaur missions such as Mariner 10, the two Pioneer Venus, and SOHO launches followed and were supported by DUKSUP (see Figure 22). The last interplanetary mission for LeRC-led Centaur was the Cassini mission to Saturn—the largest launch vehicle and largest spacecraft ever launched by LeRC. Of the 17 missions, 16 were successful.[8]

There were other interplanetary missions which were within only four months of launch, yet never flew on their intended launch vehicle. They were manifested on the new Shuttle/Centaur vehicle. The Galileo mission to Jupiter and solar polar Ulysses mission were to use Shuttle/Centaur to place these high C_3/heavy payloads on "fast" trajectories to Jupiter (Ulysses used it for gravity assist out of the ecliptic plane). In addition, the Mars mission Magellan was also slated to fly on the smaller version of the Shuttle/Centaur (see Sec. 6.3).

Unlike most Earth-orbital missions, the use of the Shuttle introduced a new variable: which orbit the Centaur and payload could be separated from the Orbiter (multi-revolution deployment). Since the Shuttle was in a LEO, the node was constantly regressing, significantly altering the initial state vector for the analysis. The optimum right ascension of the ascending node had to be first calculated, then fixed for the targeted deployment revolution. For the escape missions, Shuttle launch time and azimuth took on added importance. This was accommodated by numerically integrating the Shuttle coast orbit and optimizing the burn time of the Centaur stage. To obtain a launch window, the injection right ascension was varied around the optimum to assess the penalties for launch windows. Delayed deployment meant a misalignment of the right ascension of the ascending node (RAAN) of the outbound hyperbolic asymptote. This drove the DUKSUP analysis for the first time to multiple launch window analysis, where inertial targets had to be updated for each run and RAAN had to be modeled to generate the launch windows.

After the Challenger accident of January 28, 1986 (about four months prior to the planned launch date for the first missions: Galileo and Ulysses), it was clear that any reduction in Shuttle lift capability due to the accident would preclude doing any of the missions planned for the Shuttle/Centaur combination. On June 19, 1986, the Shuttle/Centaur project was cancelled. All missions were re-assigned to other vehicles.

Finally, one aspect of the interplanetary missions (and some of the Earth orbital missions as well) was that LeRC management usually provided a guarantee of performance to the spacecraft project office at other Centers (JPL in the case of the Mariner missions and Ames in case of the Pioneer missions). The LeRC Center Director signed a letter to the spacecraft project guaranteeing the payload weight to a specified target such that the spacecraft project could proceed with spacecraft development. The implication of these letters could be immense: they fundamentally drove major decisions on spacecraft valued in the multi-$100's M range (and a launch vehicle of comparable cost). DUKSUP was the analytic tool used to calculate that guarantee. Around this way of doing business evolved a culture at LeRC of how performance analysis should be done, how various margins should be accounted for, that analysis ground rules be explicitly specified, the way nominal vs. dispersed values should be handled, and that performance guarantees should be made with "3σ" confidence. These engineering design philosophies were consistent with similar USAF methods, and as the commercial launch world developed they were continued.

9.5 LEO Missions

Because Atlas/Centaur was designed for demanding missions (high energy and/or high mass), there were very few LEO missions which it performed unless unusual circumstances warranted it. Thus, there were only six LEO missions which Centaur flew and DUKSUP modeled. These missions were the three Orbiting Astronomical Observatories (OAO A, B, and C) and the three High Energy Astronomical Observatories (HEAO A, B, and C). OAO required a relatively straight forward mission design using a single Centaur burn with a very high altitude requirement, between 400 to 500 nmi. This required a long, steep climb by Centaur. Trajectory design for the HEAO missions, however, was a much more complex (see Sec. 11.6).

9.6 Studies: Rendezvous Missions

In the years leading up to Space Shuttle operations, NASA decided to turn over the Atlas/Centaur vehicle to its manufacturer General Dynamics and to abandon the Titan IIIE/Centaur. All NASA flights were then planned for the Shuttle, regardless of technical or cost trades. DUKSUP had no utility for the Shuttle ascent phase, which was viewed as the replacement for expendable launch vehicles by decree from HQ. Our mission design staff was once again assigned to support the Propulsion Research Division. However, DUKSUP was modified to perform unique studies starting from LEOs to support that organization.

[8] The failure of the sole test flight of the Titan IIIE/Centaur is tallied under "R&D".

One area of interest was assessing the propulsion options for raising the altitude of a very flexible space platform. Since such a flexible platform requires very low thrust for orbit raising, evaluating the performance of low thrust orbit raising propulsion systems was required. DUKSUP lent itself to that, especially in the regime between high thrust chemical systems and very low thrust electric propulsion solutions. DUKSUP with its analytical partial derivatives was able to cover most of this gap—the remainder was assessed with a low thrust orbit raising code.

The other area of interest was the development of reusable space tugs. DUKSUP was again modified, this time to solve the rendezvous problem: from Shuttle orbit, to a higher orbit (typically geostationary) to deliver a payload, and return to the Shuttle. This would have been a trivial problem except that the node of the Shuttle's LEO was regressing. (The tug's node was changing little since its transfer orbits had significantly greater semi-major axes.) For rendezvous, the final node of the tug orbit had to coincide with that of the Shuttle's. This could only reasonably be accomplished by introducing a burn near the antinode of the returning orbit. DUKSUP was able to optimally introduce that fifth burn and determine optimum payload for the propulsion system being evaluated. Further, the experience gained with the effects of higher order harmonics of the gravity model on such trajectories was crucial to obtaining solutions to complex, multiple-burn optimum trajectories for the analysis of round-trip missions by space-tugs. This rendezvous version of DUKSUP was the basis for another version of the code years later (see Sec. 6.3).

9.7 Studies: LEO Missions Supporting Space Station

Beginning in 1992, LeRC was asked to support planning for International Space Station assembly and support. In addition to the assumed Shuttle and Russian Soyuz usage, there were other options to be considered: Titan IV/Centaur and Russia's Proton. LeRC had already modeled the Titan IV/Centaur in DUKSUP as a candidate for launching some of the spacecraft lacking a ride after Challenger's last flight. New DUKSUP runs were made to assess the suitability of domestic vehicles for Space Station development and support. They could have been used to launch station modules—or at least as a backup for the Shuttle in case of another Shuttle failure. The Proton was used to launch the Russian modules which were the first parts of the station, and could have also been used as a Shuttle backup. However, following the LeRC analysis, these domestic vehicles were discarded for two reasons. First, they would have had to be human-rated to send astronauts to station in case of another Shuttle failure (as was the case with Columbia in February 2003). But the United States could have avoided being dependent on the Russians for years had a realistic assessment of Shuttle reliability been made and prudently provision for domestic backups. Second, the full cost of the commercial, domestic vehicles always appeared unattractive compared to the U.S. government-run Space Shuttle and the government-subsidized foreign launch vehicles.

10.0 How DUKSUP Was Used

DUKSUP was used for several purposes by NASA's expendable launch vehicle program leadership and LeRC management. DUKSUP was an analytic tool independent of the contractor's at the most fundamental level—the nature of the algorithm. Because of this, and its validation from the actuals from post-flight analysis, several types of analyses could be confidently performed by NASA independent of the contractor. It also provided a means to measure both generic and mission peculiar changes. Because if its pedigree, it was regularly called upon to assist contractors in mission peculiar flight software design and the user community in preliminary mission studies.

10.1 Independent Check of Contractor High Fidelity Analysis

DUKSUP was a high fidelity 3D performance and trajectory optimization code whose primary purpose was to access various technical matters surrounding how the launch vehicle flew. Although it could be used for broad, low fidelity mission studies for potential missions, DUKSUP was not built for (it nor was it easy to use for) such applications. The code produced data which was generally used in the following analysis areas: Performance, Trajectory Design, Launch Window, Range Safety, and Ground Tracking Station Coverage.

10.1.1 Performance Analysis

Performance analysis was the primary product of DUKSUP. Performance analysis meant quantifying how much spacecraft mass could be delivered to a particular orbit. Because DUKSUP could calculate vehicle performance to the same accuracy and precision (and arguably better) as the contractor, and doing so through a trajectory optimization algorithm which was fundamentally different from the contractors', it provided NASA engineers and management confidence and certainty in this critical area of launch vehicle development, integration, and launch. This meant LeRC was not dependent on accepting the contractors' analyses at face value. Further, government engineers frequently did analysis in advance of the contractor (sometimes before the business contract was in place), supporting NASA planning and guiding the subsequent contractor efforts to support launch.

The steps in performance analysis consisted of:

- Modeling the launch vehicle operation through an input file
- Modifying DUKSUP if necessary to accommodate the vehicle definition or the problem to be solved

- Performing the 3D run
- Analyzing the results
- Providing them to the contractor and payload user, deliberating on how to use the data
- Determining the next steps

The last two parts were done in the venue of the "Performance, Trajectory, and Guidance" (P, T, & G) working group, sometimes chaired by LeRC, other times co-chaired with the spacecraft center. It was this close collaboration and sharing of job responsibilities which was a real strength of the LeRC-approach to launch vehicle program management. It meant, among other things, that civil servants—frequently those chairing and managing the technical work—had firsthand experience in what they were managing.

Typical performance analyses applications included:

- Verification of current mission nominal performance (generic and mission peculiar)
- Maintenance and verification of benchmark performance (see Sec. 10.2), including interface control documents (ICD)
- Performance impacts of vehicle hardware modifications, flight operations changes, or anticipated hits
- Performance partials (sensitivities to changes)
- Calculation of performance margins (Flight Performance Reserve (FPR), Launch Vehicle Contingency (LVC), and Launch Time Reserve (LTR)
- Assessment of contractor performance documents (i.e., the Horses, Ponies, and Colts)
- Performance assessments of major reviews
- Launch support: Firing Tables, updates prior to day-of-launch updates, ADDJUST support
- Post-flight assessments, including mission success satisfaction
- Maintenance of NASA's official repository for all ELV data, including performance comparison models for launch vehicles, ELV reference library, DUKSUP, and ELV reference guide
- Feasibility assessments for future missions
- Data for the official performance commitment letter from LeRC Center Director to Mission Program Manager

Deserving of elaboration is the performance commitment letter of the last bulleted statement above. This letter was issued by the LeRC Center Director to the Program Manager of a new mission who had received Authority to Proceed. In it, a benchmark mission was specified (orbital elements, spacecraft weight, mission peculiar chargeable, launch date (if interplanetary), etc.). This was a guaranteed performance which LeRC provided the spacecraft project in the form of a signed letter by the LeRC Center Director. These DUKSUP-generated data were trusted by HQ. Should the launch vehicle loose capability to the point where it could no longer perform to the value specified in the letter, it would be LeRC's responsibility to spend whatever funding, accommodate whatever schedule (but still make the launch date), and provide the necessary people to satisfy what was defined in the letter. Further, it was LeRC's policy to guarantee "3σ" (i.e., ~99.87 percent) performance to the spacecraft organization, unlike the increasingly common specification of "confidence levels" of programs today. This meant that a flight performance reserve (FPR) had to be calculated statistically from all known dispersion sources. Thus the performance commitment agreement LeRC entered into had major implications for the spacecraft mission, the launch vehicle contractor, and for the reputation of LeRC. A lot of effort was always placed on the DUKSUP analysis, which was at the center of this performance agreement, and the analysis was never wrong.

10.1.2 Trajectory Design

Intimately tied to performance analysis was trajectory design. Indeed, the trajectory optimization produced the performance data. As was discussed in Section 7.0, the trajectory DUKSUP designed satisfied all intermediate constraints, achieved the final orbit conditions, and maximized performance. Section 10.3. discusses how DUKSUP was called upon to design trajectories which sometimes exceeded the contractors' abilities. "Shooting matches" were sometimes performed where LeRC and contractors attempted to reconcile differing results, comparing trajectories (sometime phase by phase), in order to understand why different results were occurring. This helped find errors that one party might have made or potential advantages of the optimization which one party might not have noticed. Sometimes, overlooked constraints or misunderstandings between the user, contractor, and/or LeRC were discovered during the trajectory design analysis. Figure 24 is an excerpt from a DUKSUP trajectory output file for a Titan IV/Centaur trajectory to GSO.

10.1.3 Launch Window Analysis

The launch window can be a function of many things: sun lighting constraints, cooling constraints (eclipse), Earth relative orbit element satisfaction of experiment criteria, performance limitation (interplanetary target), and others. For a launch window constrained by performance, as was usually the case for interplanetary missions, numerous DUKSUP runs were used to generate the definition of the daily launch windows. Figure 25 is an example of an interplanetary launch window generated from DUKSUP output for the Cassini mission to Saturn via Titan IV/Centaur (Ref. 13) This meant running DUKSUP with varying hyperbolic targets (C_3 and declination) for each day, and varying the right ascension to generate the window. Performance penalties were then generated. Launch window analysis typically specified fixed window time lengths (such as the amount of propellant excess (PE) needed for a "1 hr window") early in a missions' design life. The process was then inverted closer to launch (how long each daily window could be based on whatever fixed PE was available).

01 TIME	ALTITUDE	TRUE ANOM.	RADIUS	LATITUDE	DYNAM.PRES.	WEIGHT	AXIAL ACCEL.
2 PSI	INR. VELOCITY	C-3	GRD. RANGE	LONGITUDE	ATM. PRES.	FLOW (T)	THRUST (T)
3 PSID	INR. GAMMA	ECCENTRICITY	TRAVEL ANGLE	INR. AZIMUTH	MACH. NO.	FLOW (S)	THRUST (S)
4 VAR. BETA	REL. VELOCITY	SEM. LAT. REC.	ALTAPO	ALTPER	DRAG COEFF.	FLOW (V)	THRUST (V)
5 VAR. BETAD	REL. GAMMA	PERIOD	TIME P. PER.	INR. INCLIN.	DRAG	FLOW (B)	THRUST (B)
6 ZLDOT	YAW ANGLE	ARG OF PERIGEE	TIME OF PER.	NODE	LONG OF AEC	HT. PARAM. QV	HEAT INTEGRAL
7 CRASH LAT	CRASH LONG	TIME OF IMPACT	LENGTH OF FALL	CRASH TRVL ANG	ALPHA	BETA	STAGE #
0ELEV= 87.48609							
01 0.2831633	-0.1192093E-06	180.0000	0.2090977E+08	28.42214	0.0000000E+00	2070506.	1.000000
2 90.00000	1340.976	-124.9177	0.0000000E+00	279.4177	14.75055	8543.376	2070506.
3 -1.371941	0.3885990E-13	0.9973289	0.0000000E+00	90.00000	0.0000000E+00	0.0000000E+00	0.0000000E+00
4 0.0000000E+00	0.0000000E+00	55853.03	-2.626128	-3439.329	2.410000	0.0000000E+00	0.0000000E+00
5 0.0000000E+00	90.00000	1793.835	-0.5338320E-02	28.42214	0.0000000E+00	8542.939	2070506.
6 0.0000000E+00	0.0000000E+00	-90.00000	0.2885017	189.4177	189.4173	0.0000000E+00	0.0000000E+00
7 28.42214	279.4165	0.2831633	-0.2983530E-11	-0.2864790E-05	0.0000000E+00	0.0000000E+00	1
01 11.00000	816.2440	179.9812	0.2091058E+08	28.42213	30.16748	1945190.	1.576762
2 90.03938	1350.727	-124.9104	21622.35	279.4177	14.33598	12068.88	3072497.
3 -1.351813	6.969824	0.9973297	0.3937599E-01	90.04134	0.1444859	0.0000000E+00	0.0000000E+00
4 0.0000000E+00	163.9092	55838.19	-2.422933	-3439.330	2.278879	0.0000000E+00	0.0000000E+00
5 0.0000000E+00	89.64763	1793.992	891.8910	28.42216	5396.720	12068.45	3072497.
6 0.0000000E+00	-0.6869377E-05	-89.90486	-880.8910	189.3756	185.5615	4944.728	11796.77
7 28.42208	279.4165	24.87491	13.87491	0.5097125E-01	0.3413172	-0.3779366E-01	1
01 11.00000	816.2440	179.9812	0.2091058E+08	28.42213	30.16748	1945190.	1.576405
2 87.48965	1358.116	-124.9085	21622.35	279.4177	14.33598	12068.88	3071803.
3 -1.344546	6.925144	0.9972999	0.3937599E-01	90.01599	0.1444859	0.0000000E+00	0.0000000E+00
4 0.0000000E+00	163.9092	56461.51	-2.423061	-3439.278	2.278879	0.0000000E+00	0.0000000E+00
5 0.0000000E+00	87.48609	1794.032	891.9157	28.42213	5396.720	12068.45	3071803.
6 0.0000000E+00	-0.1338079	-89.95161	-880.9157	189.4289	185.6147	4944.728	11796.77
7 28.42210	279.4168	24.86762	13.86762	0.5122803E-01	0.3205062E-13	-0.2986592E-02	2
01 15.00000	1627.051	179.9717	0.2091139E+08	28.42212	65.55057	1896301.	1.665752
2 85.26665	1382.830	-124.8974	0.8467864E-02	279.4178	13.93347	12367.78	3170161.
3 -1.309198	10.14290	0.9972475	0.5421554E-01	90.06063	0.2160371	0.0000000E+00	0.0000000E+00
4 0.0000000E+00	244.3571	57560.94	-2.206324	-3439.188	2.213946	0.0000000E+00	0.0000000E+00
5 0.0000000E+00	85.26093	1794.272	889.5499	28.42218	11392.35	12367.34	3170161.
6 0.0000000E+00	-0.2480133	-89.85968	-874.5499	189.3519	185.5101	16017.75	50429.60
7 28.42201	279.4179	35.19289	20.19289	0.7530994E-01	-0.4541357E-13	0.8495131E-01	2
01 15.00000	1627.051	179.9717	0.2091139E+08	28.42212	65.55057	1896301.	1.665695
2 85.26665	1382.830	-124.8974	0.8467864E-02	279.4178	13.93347	12367.78	3170053.
3 -1.309198	10.14290	0.9972475	0.5421554E-01	90.06063	0.2160371	0.0000000E+00	0.0000000E+00
4 0.0000000E+00	244.3571	57560.94	-2.206324	-3439.188	2.213946	0.0000000E+00	0.0000000E+00
5 0.0000000E+00	85.26093	1794.272	889.5499	28.42218	11392.35	12367.34	3170053.
6 0.0000000E+00	-0.2480133	-89.85968	-874.5499	189.3519	185.5101	16017.75	50429.60
7 28.42201	279.4179	35.19289	20.19289	0.7530994E-01	0.0000000E+00	0.8495131E-01	3
01 50.00000	25678.51	179.8071	0.2093545E+08	28.42058	883.8802	1465022.	1.852382
2 59.77607	2293.620	-124.4426	1.496885	279.4460	5.574162	10845.17	2939794.
3 -0.6934848	29.40162	0.9940673	0.2075634	91.13726	1.254228	0.0000000E+00	0.0000000E+00

Figure 24.—Example DUKSUP Trajectory Output File (Initial Phases Only)

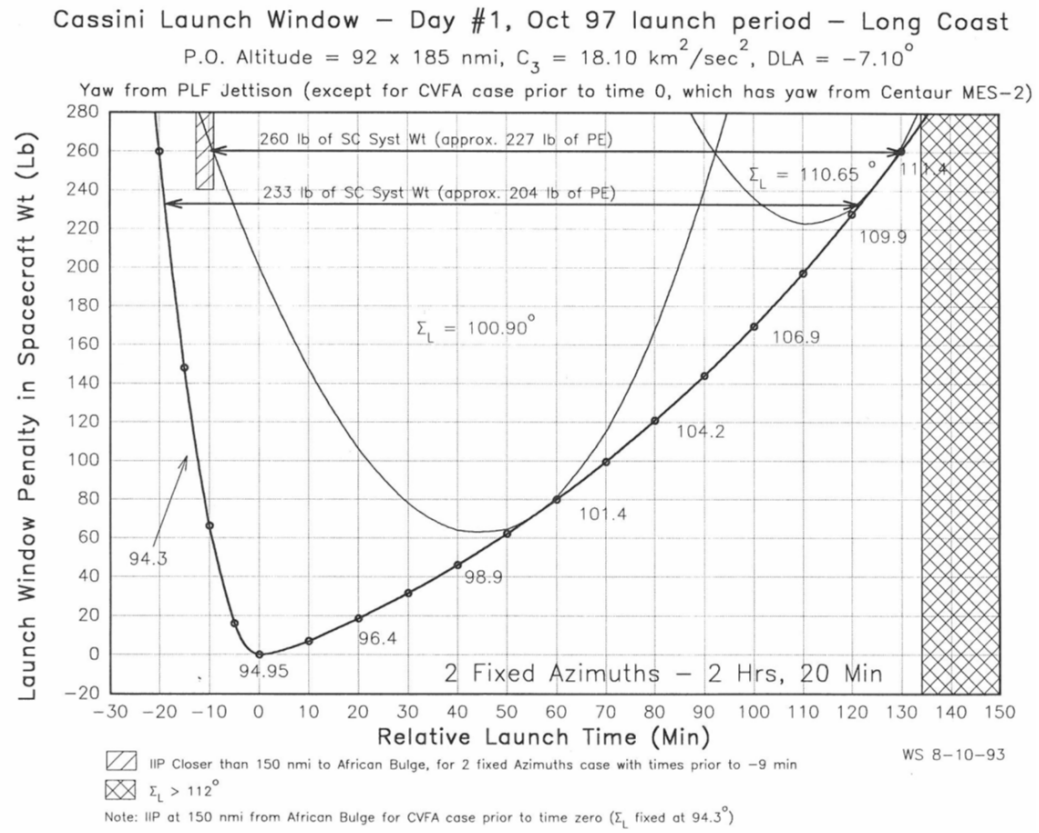

Figure 25.—DUKSUP Launch Window Penalty Data for Cassini

10.1.4 Range Safety Analysis

The Range required trajectories (nominal and dispersed) and the impact of hardware locations (both nominally and due to failure) from launch through orbit insertion. DUKSUP could calculate some, though not all, of these trajectory-related data which were eventually incorporated into the Range Safety Data Package generated by the contractor. Ground tracks were calculated for the nominal ascent. Dispersed trajectories were not calculated, though they could be done if the modeling data were available. The instantaneous impact points (IIP) (the landing locations should thrust be terminated) of the vehicle were also calculated, though the error ellipses for jettisoned hardware were not. Figure 14 to Figure 16 are examples of IIP traces generated from DUKSUP output for early ascent, African bulge flyby, and continental African overflight. DUKSUP was not used for the related trajectory analysis (steepest lateral, steepest ascent, etc., and the tumbling turns). All of this analysis was used to support requests for flight plan approval, hazard analysis, and day-of-launch Range Safety support.

10.1.5 Ground Tracking Station Coverage Analysis

Visibility by ground tracking stations worldwide during ascent and orbit was vitally important, particularly during mark events. DUKSUP calculated elevation and view angles from ground stations to the vehicle throughout the trajectory. Figure 26 is an example of ground tracking station elevation angles for the CRRES mission on Atlas/Centaur (Ref. 14). Section 7.4 also discusses how the code could optimally constrain the trajectory to be within a specified angle to a specified tacking station. This data was used by the LeRC launch vehicle program to ensure that telemetric data could be available for capture. It was also available to user community should health assessment or uplink capability be required.

10.2 Benchmarking

Closely related to Section 10.1.1 is benchmarking of performance analysis. Although we used the capabilities of DUKSUP to provide management with the payload implications of imposing constraints and trajectory modifications to improve performance for specific missions, we also maintained what we called “benchmark” trajectories for each mission. These were used to readily track the payload impacts of launch vehicle changes. The benchmark trajectory was typically chosen to be representative of the mission at the time the benchmark was established. Inevitably, the real mission evolved over time, but the benchmark ground rules did not change. The stability of this benchmark trajectory allowed the engineers and management to track the impact of vehicle changes from a constant baseline. Otherwise, it would not have been possible to have a clear idea of the implications of the on-going vehicle changes, such as hardware weights, operational decisions, and vehicle improvements. Such changes were constantly a part of the life of an on-going launch vehicle project. In addition, we were always able to relate the implications of vehicle changes on the benchmark trajectory and payload to those on the current mission.

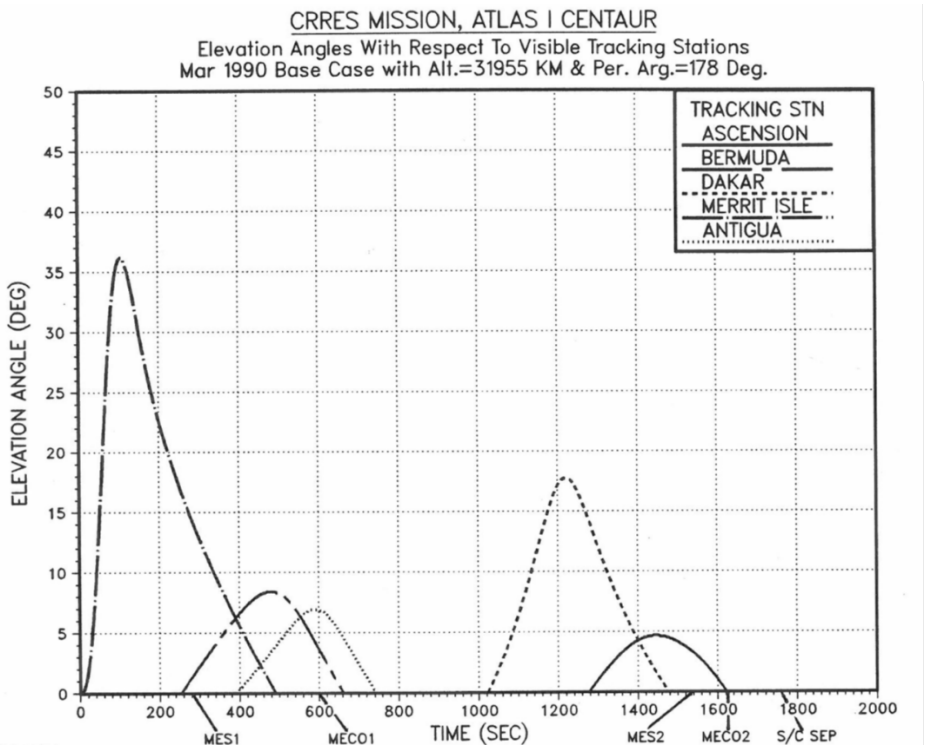

Figure 26.—DUKSUP Tracking Station Coverage Data for CRRES

10.3 Assisting and Leading Contractors

DUKSUP analysis was frequently used to assist launch vehicle contractors in their trajectory design and flight software development. Because of the close relationship LeRC staff had established and maintained with the contractors, particularly in the P,T, & G working group (in later years referred to as the Mission Design Panel), and because the software tools were of comparable pedigree and fidelity, trajectory optimization results were regularly shared and used during the preparation for launch. Sometimes, because CoV solutions guaranteed at least a local optimum, solutions could be found, which were difficult for (or sometimes even eluded) contractors to find with their codes (which were not CoV-based). Elliptical parking orbits (see Sec. 7.5), sub-orbital yaw for HEAO (see Sec. 7.6), and Galileo launch windows with combined Amphitrite/non-Amphitrite targets (see Sec. 11.8), were examples of this type of assistance provided to the contractors. The latter case actually resulted in a role reversal: where LeRC became the lead in calculating and defining the daily launch windows while the contractor checked the results.

Still another type of assistance was applying lessons learned in support of one user to another, as was the case with geostationary missions and Pioneer Venus. The optimal parking orbit design first worked on and applied to geostationary missions (see Sec. 11.4) was also applied to other non-GSO missions. The Ascension Island tracking constrained trajectory was also made available to other users (see Sec. 11.10).

10.4 Leading Edge Analysis for Yet to be Approved Missions

DUKSUP was frequently called upon to determine the performance capability and other mission trajectory related information for proposed missions. Many of the missions in these pre-Phase A studies were never given Authority to Proceed. Some of the last leading edge mission studies the on-orbit deck SHUTSUP evaluated included the "Mariner Mark II" class of missions: Comet Rendezvous Asteroid Flyby ((CRAF), Saturn Orbiter Titan Probe (SOTP), Main Belt Asteroid Rendezvous (MBAR), and Uranus Flyby Uranus Probe (UFUP). The Space Station assembly/resupply on domestic ELV's (see Sec. 11.7) were another set of leading edge analyses performed on the then-new DUKSUP Titan IV/Centaur deck. These studies leveraged the existing strong ties between LeRC and the user community in order to perform quality analysis required to enable HQ to make thoughtful and prudent decisions on how (or whether) to proceed with these conceptual missions which were usually poorly defined. DUKSUP was used to provide the best match of launch vehicle configuration to the mission, options on launch trajectory, and high quality performance data. Other assessments accompanied DUKSUP data, such as payload fairing dimensional accommodation.

Another variant of these activities were the post-Challenger studies of 1986 to 1988. Following the disaster, there were Shuttle/Centaur interplanetary missions which needed analysis for launch in later years (such as Galileo, Ulysses, Magellan), as well as other missions which were to become Shuttle refugees in need of finding alternate rides (such as CRRES). LeRC performed these trajectory studies with DUKSUP in order to provide NASA HQ with a recovery plan during the 3 year down period following the Challenger disaster.

10.5 Vehicle Performance Improvement Analysis for NASA

DUKSUP was also called upon to assess the payload improvements to be derived from various improvements the launch vehicle. These improvements were in both hardware (discussed here) and software/analysis (Sec. 7.2). One user, Intelsat—a profit making venture, was frequently interested in increasing the payload capability of Atlas/Centaur to enable an increase in the weight of their spacecraft. We did studies to determine the payload improvements associated with lengthening the Atlas tank to increase the propellant load. As the hardware group analyzed the structural and dynamic impact of Atlas tank lengthening, DUKSUP was able to quantify the payload improvements associated with the various options back to Intelsat. As a result, the Atlas was lengthened twice during LeRC's tenure leading launch vehicles.

Other examples of these upgrades which were studied by DUKSUP and were implemented on the flight vehicle include: upgrading the Atlas booster MA-5 engine thrust, developing Centaur guidance of Titan IIIE booster, and improving the I_{SP} of the RL-10 with a silver throat insert. There were other upgrades which were also studied, but not implemented, such as adding fluorine to the oxygen in the Atlas tank. Though this study was done numerous times because of its initially-perceived simplicity, the daunting operational problems could never be rationalized despite the significant performance improvement.

11.0 Notable Impacts

The following mini-case studies are examples of the significant impact DUKSUP data had on either the launch it was supporting, the launch vehicle program as a whole, or how the user community was able to improve the use of their spacecraft.

11.1 Maximized Performance Obtained From the Variational Solution Improved On-Orbit Lifetime

The ability to obtain all the performance available from the vehicle through the use of trajectories generated by DUKSUP's

variational calculus algorithm proved to be extremely valuable to the Centaur program and to the users. At the time, GSO satellites were primarily limited in lifetime by the available stationkeeping propellants. Adding a few more pounds of propellants to the spacecraft tanks could add months, if not years, to the lifetimes of the spacecraft. Consequently, as the schedule moved from the initial planning to launch, the trajectories were easily re-optimized several times to yield targets that would allow the spacecraft tanks to be loaded to the maximum payload available. Trajectories would be re-optimized just prior to flight to gain as few as five more pounds in orbit. In addition, even for interplanetary missions, increasing payload by only a few pounds could add lifetime to the spacecraft at the target planet. The cost to do this was trivial, and the benefits to the spacecraft could be financial, or could provide additional margin for spacecraft operational problems, or maneuvering flexibility, i.e., moving the spacecraft longitude in the case of the geostationary missions.

The success achieved with these early mission designs led to a LeRC philosophy to make certain that trajectories were flown that would obtain all the payload available from trajectory optimization. The Atlas/Centaur and Titan/Centaur would go on to fly a wide variety of missions—and every mission was provided with the maximum payload available from the launch vehicle. In the absence of the maximum payload, unnecessary compromises in the spacecraft would have meant potentially higher cost and increased risk. Similarly, in the case of the launch vehicle, the performance improvements available from hardware changes, such as lightening the vehicle or improving the propulsion, were always expensive and introduced risk. In many cases, the changes did not increase payload any more than some of the trajectory changes that we ultimately made. If hardware changes were ultimately required, the project knew that there was no alternative.

11.2 Communication Technology Satellite (CTS)

In 1975 the Canadian Space Agency and NASA collaborated on launching an experimental high powered communications satellite to GSO. Shortly before launch, it was discovered that the Delta launch vehicle was short 10 lb of being able to launch the spacecraft. The options were to either increase the launch capability of the Delta by 10 lb or decrease the weight of the spacecraft by 10 lb. Increasing the performance of a launch vehicle would have been expensive and would incur risk. Decreasing hardware weight of the spacecraft was also expensive and risky so close to launch. Decreasing the spacecraft propellant to remain on station would compromise the mission and was also undesirable. Someone in the CTS project at LeRC was aware of the capabilities of DUKSUP and requested that we analyze the proposed trajectory and determine whether trajectory optimization could pick up the needed payload. We went to Goddard Space Flight Center (GSFC) (which managed the Delta) to obtain a trajectory so that we could model and match the performance of the Delta, then use DUKSUP to optimize the trajectory to determine whether any payload improvement was possible. We started with the Delta state vector at solid separation. Using the injection targets from the GSFC trajectory and matching performance segment by segment, we obtained a trajectory that yielded the same performance as the GSFC trajectory. We then freed DUKSUP to optimize the trajectory. Payload increased slightly more than the 10 lb that was needed by optimizing the parking orbit elements, injection targets, and transfer orbit and final injection strategy. The optimized trajectory was provided to GSFC and we understand that it was used for launch.

11.3 Titan IIIE/Centaur Variable Launch Azimuth Decision

For each of the Titan IIIE/Centaur missions in the 1970's, the Titan phases of flight were steered by Centaur's guidance rather than Titan's. Centaur was capable of varying the launch azimuth as a function of time and date. While this made for a complex and substantive amount of preflight trajectory analysis to be performed for Range Safety, it allowed for maximizing vehicle performance for the six challenging missions Viking A and B, Voyager I and II, and Helios A and B. The Titan contractor at that time could not perform a continuously variable launch azimuth analysis nor could they accommodate this steering within the existing flight software. But because Centaur flight software could, and DUKSUP performed the initial trajectory design, the Titan IIIE/Centaur missions were able to successfully fly these complex trajectories. (Surprisingly, 20 years later, this feat was shied away from by Titan IV/Centaur (where the Titan steered itself) despite LeRC's successful analysis of viability and advocacy.)

11.4 Optimal Parking Orbits

When we began analyzing GSO missions, we assumed that the optimum parking orbit to be circular. The parking orbit altitude was selected to be 90 nmi above the equator. This altitude was selected to ensure that the instantaneous heating rate would be low enough such that no component of the Centaur or the spacecraft would be damaged by heating.

By observing the characteristics of the Atlas/Centaur trajectory into the parking orbit, we wondered why a circular orbit was optimum. The Atlas/Centaur tended to "rollercoaster" into the circular orbit—that is, the vehicle went above the circular orbit altitude and then dove slightly below the circular orbit before going into the circular orbit. This suggested to us that perhaps a circular orbit was not optimum. DUKSUP was modified to put a single constraint on the parking orbit injection—perigee altitude. All the other orbit elements were optimized. This tra-

jectory was obtained, checked for optimality, and the improvement in payload was 29 lb. This improvement was certain, risk free, cost nothing, and was easy to implement. No hardware changes were required—the only change was to the parking orbit injection target.

The spacecraft project sometimes wanted to wait until the second apogee before injecting into the final orbit. DUKSUP could maximize the payload for this mission design ground rule with no problem. The payload weight would vary very little—the only difference would be the effect of oblateness for the longer coast which would have a slight effect on the optimization.

In comparing the two trajectories—burn at first apogee and at second apogee—it was observed that the instantaneous heating rate at perigee of the 90 nmi altitude parking orbit was significantly lower than the heating rate for the near 90 nmi perigee of the transfer orbit (Ref. 8). Further study indicated that the heating rate at perigee for an 80 nmi altitude parking orbit was very near the heating rate for perigee of the 90 nmi altitude transfer orbit. The trajectory generated by DUKSUP yielded another approximately 30 lb of payload to GSO. The decision to use this trajectory was left to the spacecraft project. At this point, we were providing General Dynamics with the trajectory designs since they lacked the capability of optimizing these trajectories. We provided the spacecraft project with trajectory options allowing the office to size the solid motor on the spacecraft to maximize payload. Just before launch, we incorporated all the final weights and the spacecraft motor characteristics and re-optimized the trajectory. If the payload improved as much as 5 lb, we provided a final trajectory design such that the spacecraft project could take advantage of the additional payload. Post flight analysis verified the payload improvement.

Another example of how DUKSUP was used to introduce optimal parking orbits was for the MO mission. A post 1986 Challenger-refugee, MO was manifested on Titan 3/TOS which had sufficient capability from Shuttle's circular parking orbit. But DUKSUP determined that by using the full Titan III capability to inject into an elliptical parking orbit, performance could be improved by several 100's lb (Ref. 8). However, the optimum parking orbit shape varied significantly from day to day, and during each daily window. So LeRC management elected to use a fixed, near-circular orbit because it provided acceptable payload, was simpler, and could be accommodated by the TOS's avionics suite.

11.5 Propellant Settling During Coast

Normally during Centaur coast periods, small settling motors were fired along the velocity vector for a short period of time after Centaur cutoff to settle the propellants, and then again for a period prior to Centaur second burn to position the propellants at the rear of the tank for engine start. These firing periods added a very small amount of energy to the orbit. However, DUKSUP's CoV analysis revealed that the optimal direction for this propellant settling thrust used during the parking orbit coast periods (just after the first Centaur burn) was actually an angle of ~180° with respect to the local horizontal (i.e., essentially a small retro burn). Then, at the end of the parking orbit coast, the optimal thrust angle would reverse until it was at a ~5° angle below the local horizontal at the beginning of the Centaur second burn. Analysis determined that if the thrust angle were allowed to be optimal, the payload gain was trivial (~ 1 to 2 lb) (Ref. 8).

This result seemed counter-intuitive at first. The explanation was that the optimum solution lowered the altitude at initiation of the second burn, thus increasing the velocity. Adding ΔV at a higher velocity is a more efficient way to add orbital energy, thus increasing payload. This change was never implemented because fighting for a more sophisticated steering law during the coast was not worth the trouble for such a small payload gain. However, had it been worth even as little as 10 lb, we would have pushed for implementation. Although this approach was never pursued, we feel confident that only a CoV-based code like DUKSUP could have provided this insight.

11.6 High Energy Astronomical Observatory (HEAO) Missions

During 1977 to 1979, three unusual missions called the High Energy Astronomical Observatories (HEAO) were launched by Atlas/Centaur. Two of these missions (HEAO A and B) required circular orbits with altitudes between 230 and 260 nmi, at inclinations of ~22.75° to 23.5°. These low inclinations were desired to avoid the bulge in the Van Allen belts in the Southern Hemisphere ('South Atlantic Anomaly'). There were several trajectory options to reach such a relatively low altitude and low inclination. Since CCAFS is at a latitude of 28.5°, it was necessary to reduce the orbit inclination at some point in the trajectory. It was apparent that the trajectory yielding the most payload was probably a three or more Centaur burn mission, where inclination reduction could occur near the equator and at a higher altitude. Although Centaur had demonstrated on an earlier mission that it could perform seven burns, the HEAO payload requirements were low enough that taking the risks of multiple restarts was deemed unnecessary by LeRC management. Trajectories were then produced by DUKSUP for the two-burn Centaur missions, where the inclination reduction was done near the equator and the perigee of the parking orbit was significantly suborbital. These trajectories had the disadvantage that if the Centaur failed to ignite for the second burn, the spacecraft would immediately re-enter, a risk to be avoided. So direct ascent (i.e., one-burn) trajectories were investigated. These were selected since they had sufficient performance, although they were by far the most complex trajectories we ever designed and flew. To maximize payload, the optimum combination of four concerns had to be obtained: lofting to acquire the desired final altitude, reaching a lower latitude such that inclination could be reduced, minimizing the magnitude of the velocity vector that had to be "turned" to reduce inclination, and avoiding dropping hardware on Venezuela and the islands of the Caribbean (see Figure 27).

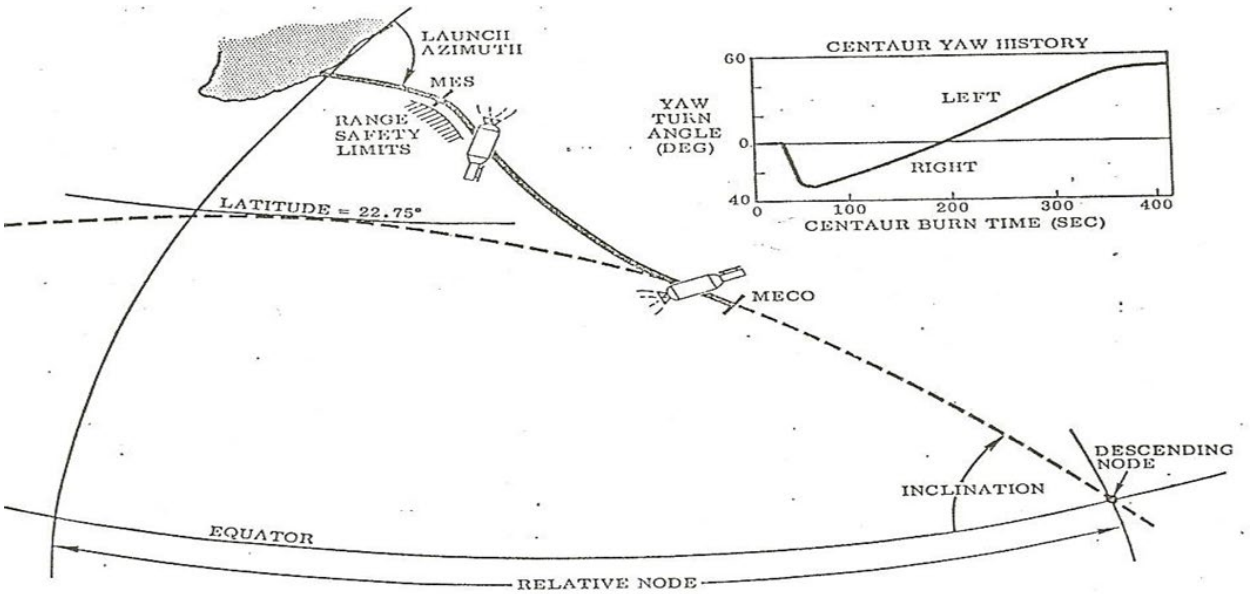

Figure 27.—DUKSUP-Designed Unique Direct Ascent with Sub-Orbital Yaw (HEAO Mission)

This requirement was an usual trajectory design challenge. Trying to obtain such a trajectory parametrically would have been daunting, if not impossible. To achieve a final inclination of 22.75° (i.e., less than the latitude of the Cape), the Centaur had to be at most at 22.75° latitude for a planar ascent. To reach that latitude limited by a relatively short burn arc, the vehicle had to get south as quickly as possible. This implied a very southerly launch azimuth. However, because of jettisoning hardware (booster engines, insulation panels, nose fairing, and sustainer stage) had to avoid the Caribbean islands and Venezuela, the launch azimuth was limited, usually at most to 115° (a value ultimately determined by the Range). The trajectory was constrained to the launch azimuth plane (i.e., planar ascent) until the hardware was dropped. As soon as yaw was allowed, the vehicle had to yaw as quickly as possible. Absent these hardware footprint problems, the optimum launch azimuth would have been far south of an acceptable value. With the launch azimuth constraint, when the vehicle was finally allowed to yaw south, a large discontinuity in the thrust vector was required (over 30°). The Centaur had a limit of 2° per second on the rate of change of the thrust vector. In previous missions, discontinuities in the thrust vector had been very small and, for performance purposes, they were ignored. In this case, they could not be. Consequently, the constraint on the rate of change of the thrust vector was included in a new version of DUKSUP and optimum solutions were obtained. The resulting trajectory was unusual in that it demonstrated the optimum trade-off between getting south as soon as allowed, while minimizing the horizontal velocity vector at the end of the trajectory that must ultimately lie in the orbit plane. There was some payload penalty, although not large. As soon as it was allowed, the vehicle yawed to the right (south) to reduce latitude. The thrust angle in yaw then moved smoothly back to the left until, at the end of the burn, the thrust was about 55° out of the plane of the orbit. To fly this trajectory, the Centaur guidance equations had to be significantly reformulated in yaw to mimic DUKSUP's CoV solution. The yaw equations had to incorporate a bi-linear tangent steering law, similar to the pitch equations. This was done, and the result was a flight trajectory for HEAO that matched DUKSUP's solution, and yielded the maximum payload from the vehicle. This demonstrated again that, with appropriately sophisticated guidance and steering laws, the vehicle could fly a near CoV trajectory.

Examination of the trajectory was fascinating as the instantaneous inclination of the orbit, vehicle latitude, the horizontal velocity, and altitude varied optimally to yield the best payload. To maximize payload, the trade-off between these contributions to the optimum solution are continuously changing and not linearly. For instance, it was advantageous to reach the orbit altitude as soon as possible because the circular orbit velocity diminished as orbit altitude increased. However, to reach the required latitude, higher horizontal velocities were required. As the trajectory proceeded, the inclination increased as latitude was reduced. Finally it had to be decreased to 22.75°; however, as inclination was reduced, the velocity had to increase to circular orbit velocity at the altitude (Ref. 8).

Intuition suggested that a two-burn solution would have yielded higher payload. To wit, putting the Centaur and payload in an 28.5° elliptical orbit with an optimum apogee altitude, perform the plane change to 22.75° and adjust the perigee to 240 nmi and the argument of perigee to 0°, (that is the perigee of the elliptical orbit was over the equator), coasting to the equator and then circularizing. Almost certainly, this trajectory to the desired final orbit would have a higher payload than the one

burn solution. But we were discouraged from examining this potential improvement in payload since the one burn payload was sufficient.

Finally, the low altitude and low inclination target benefited from the incorporation of a new technique called "in-flight retargeting." This capability allowed an in-flight compromise in the target if an on-board calculation of performance indicated that the nominal target could not be achieved, and the vehicle would burn out suborbital as a result. This capability was a direct result of the failure of AC-21, OAO-B, which burned out suborbitally trying to achieve the fixed target orbit. The users pre-agreed to a strategy for defining less demanding targets which were determined by the level of predicted performance deficiency. For example, achieving an altitude of 230 nmi, when the nominal was 240 nmi, could still result in a useful mission. An achieved, compromised target would have been better than a failed mission. Without triggering retargeting, the HEAO A and B missions were flown successfully to the desired final orbits.

11.7 Voyager

Two Voyager spacecraft were launched to Jupiter in 1977 on Titan IIIE/Centaur (TC). On the second Voyager launch (i.e., Voyager I; they were numbered based on Jupiter arrival dates, not Earth launch dates), TC-6, on 9/5/1977, the Titan core stage II experienced a hardware failure and shut down 544 ft/sec slower and 3,000 ft lower than nominal. Mission success depended on the Centaur's ability to make up for this huge energy shortfall. The failure occurred very early in the Titan core stage II burn, and soon after this the launch team was aware of the failure, based on real-time telemetry. To compensate for this energy deficiency, Centaur was required to burn longer than nominal for its first burn to parking orbit. There was great concern that the Centaur would have sufficient remaining propellants for its second burn to inject the Voyager spacecraft on the trajectory to Jupiter. However, with relatively simple calculations on slide rules, several members of the mission design team determined that Centaur would have enough propellants to make up this deficit (although barely). It did, and injected the Voyager I spacecraft successfully. Had the identical Titan failure occurred on the first launch (TC-7 with Voyager 2) the "Grand Tour" mission (Jupiter, Saturn, Uranus, and Neptune) would have been lost due to its more demanding launch geometry (Ref. 8). Voyager II would likely have been rerouted to perform Voyager I's mission (sacrificing flyby's of the third and fourth planets). Figure 28 illustrates the Voyager 1 and 2 heliocentric trajectories as a result of Centaur's injection burns (also shown are illustrations of Voyager and its launch on Titan IIIE/Centaur).

Three critical mission design points emerge from this experience. First, it proved essential to have obtained an optimum trajectory design from lift-off through spacecraft solid motor burn, similar to the geostationary missions. Second, all the propellants

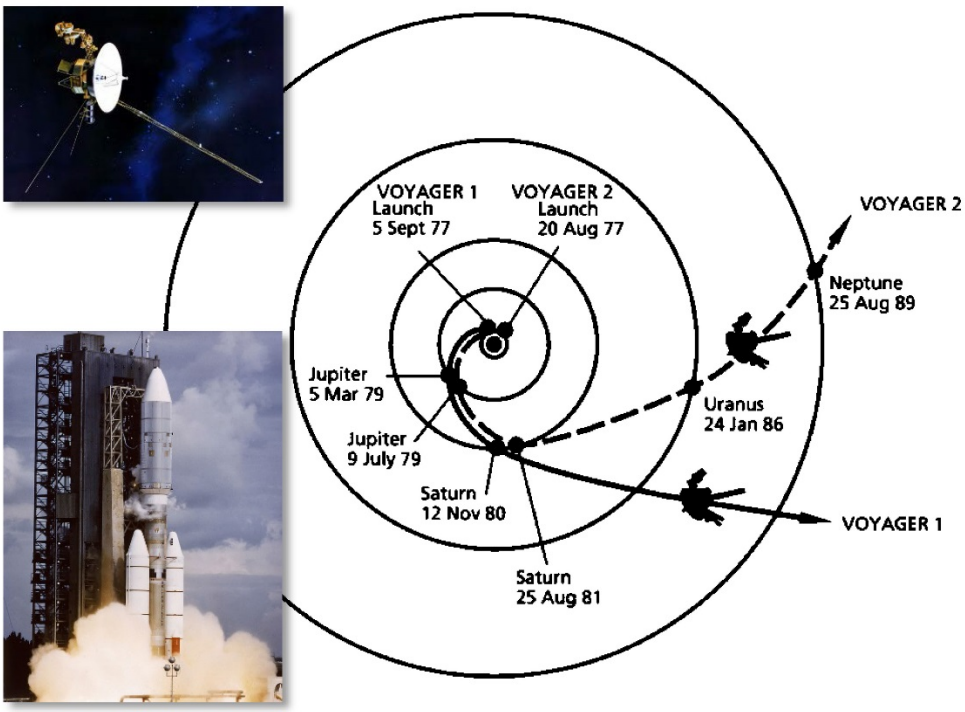

Figure 28.—Heliocentric Trajectories of Voyager 1 and 2; Voyager and Titan IIIE/Centaur

held in reserve to compensate for 3σ in-flight performance dispersions were carried in the Centaur upper stage, with none carried in any Titan stage. Third, the Centaur guidance and steering algorithms were sufficiently sophisticated to "redesign" the ascent trajectory after sensing the energy-deficient initial Centaur state vector. Voyager I is an example of where the optimum trajectory design philosophy and flexible guidance software saved a "flagship" NASA mission, and provided a wealth of scientific knowledge that mankind would have had to wait decades to obtain. DUKSUP was the essential part of the first point above.

11.8 The 109 Percent SSME NASA HQ Decision for Galileo's Launch Windows

The Galileo mission to Jupiter, manifested for launch in May 1986 on Shuttle/Centaur, was short on performance. Nine months before launch, the baselined 65,400 lb to LEO Shuttle lift commitment allowed only a 14 day launch opportunity with daily windows less than 1 hr long, while simultaneously exhibiting a significant (and still unresolved) negative total system performance margin. This was because of the demanding C_3 targets and a Shuttle lift constraint which required a ~2400 lb propellant offload on Centaur. A comprehensive analysis was performed using DUKSUP to illustrate how the opportunity could be enhanced if the Shuttle's SSME's were to be run at 109 percent throttle (a capability claimed as possible) enabling Centaur to have full propellant tanks. In the course of three days, DUKSUP, running on the then-state of the art Cray-1S supercomputer, calculated over 15,000 optimized 3D trajectories based on a 67,750 lb to LEO capability a 109 percent SSME could enable (i.e., full Centaur tanks). The launch opportunities expanded from 14 days to 22 days. The daily launch windows expanded from the marginal lengths of 30 to 50 min to the sufficient lengths of 90 to 130 min (see Figure 29). Just as importantly, the total system performance margin was resolved; changing from –168 lb of PE to +352 lb PE (Ref. 23). The analysis was presented to a HQ-level gathering, with the Associate Administrator (AA) for Space Flight presiding. Also present were senior executives from LeRC, JPL, and the Code M Centers. Over the objections of the Code M Centers and Space Shuttle senior management, the AA decided to confirm the baseline of 109 percent SSME for Galileo (Ref. 24) because of the analysis generated from DUKSUP.[9] However, the loss of the Challenger Space Shuttle, and the subsequent cancellation of the Shuttle/Centaur program, rendered this decision moot.

The other directly related accomplishment pertained to the analysis of the alternate Galileo targets for "second day deployment". DUKSUP, along with a new standalone utility, was able to calculate launch windows of varying lengths and/or penalties, both with and without alternate targets to the flyby of asteroid 29-Amphitrite (Ref. 23). Each set of daily performance curves had windows which frequently opened on one revolution/one target, and closed on a different revolution/different target (see Figure 30 for June 2, 1986). Because of how the first day Amphitrite targets were changing differently from the second day non-Amphitrite targets, the day-by-day resultant window analyses turned out to be irregular data and difficult to analyze. But because of DUKSUP's superior optimization capability and speed of the computer platform, these solutions were able to be discovered

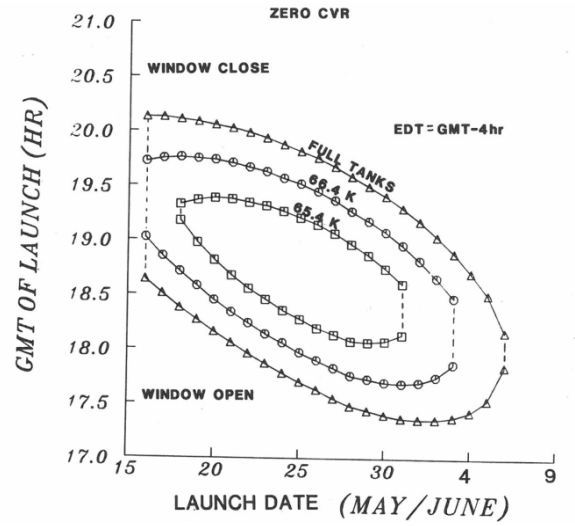

Figure 29.—DUKSUP Launch Opportunity for Galileo

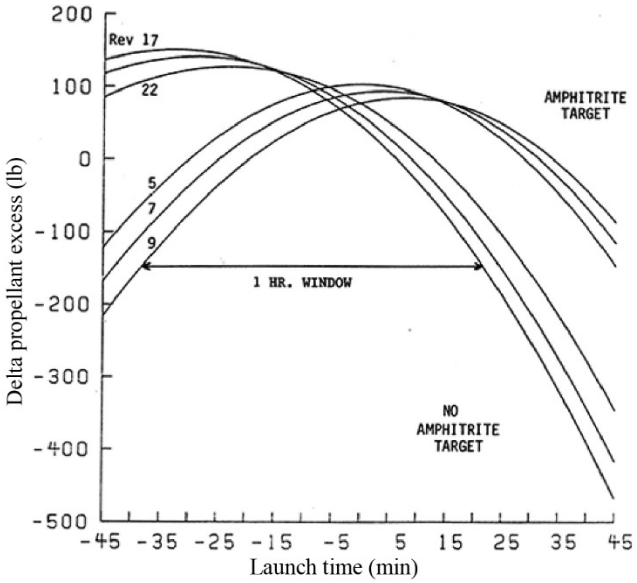

Figure 30.—DUKSUP Daily Launch Window Performance Penalties for Galileo

[9] Nieberding, J.J., Personal communication, NASA LeRC (ret.) June 2014; and prior discussions with lead author Spurlock, O.F.

11.9 Generation of Data for AIAA International Reference Guide to Space Launch Systems Textbook

In the 1990s, LeRC generated a prodigious amount of performance data for the American Institute of Aeronautics and Astronautics (AIAA) for the comprehensive textbook comparing launch vehicles (national and international) on a common basis. This reference guide is now in its fourth edition and has grown to be a 500+ page tome. It is considered the go-to reference for launch vehicles, particularly performance data, short of commitments from the vehicle providers. Though not generally acknowledged at the time, the second edition of this guide published LeRC-generated data which was thoroughly documented, meticulously ground-ruled, and comprehensive in parametric form with data run by DUKSUP (Ref. 25).

11.10 Pioneer Venus Mission and the Ascension Island Tracking Constraint

Arguably the most challenging optimization problem ever faced by LeRC staff was the genesis of the most capable version of the code: SUPERDUK. This version was written for the Pioneer Venus missions launched by Atlas/Centaur in 1978. There were two launches, an Orbiter in May and a multi-Probe in August. The launch of AC-50 of Pioneer Venus A is shown in Figure 31. Both of these launches were two burn, which meant that ARIA were used to provide Centaur second burn telemetry coverage. It occurred to us that it might be possible to acquire the second burn telemetry coverage from Ascension Island in the middle of the Atlantic Ocean, as was possible with the GSO missions.

SUPERDUK could optimize launch azimuth and trajectory lofting in order to (for the first time) produce launch day/launch time-varying interplanetary trajectories (Ref. 8). It also could constrain the Centaur second burn so it could be observed from Ascension Island (see Figure 17). This latter capability was similar to the launch time independent Intelsat mission trajectories described earlier. If such trajectories could not be obtained, it would be necessary to observe the second burn with ARIA, as the tracking ships were no longer available. In this formulation, the launch azimuth and the lofting of the trajectory would be optimized using CoV, not parametrically as had been done before. In addition to the Ascension constraint, the other constraints imposed were the qV's for the panel and fairing jettisons and the perigee of the parking orbit (Ref. 8). Filling the Centaur tanks was also included in the iteration, rather than performing that as a separate part of the problem. The target for each trajectory was injection energy (C_3) and the declination which were provided by the spacecraft project. The right ascension was optimized. The penalty for supplying a launch window was evaluated by varying the targeted right ascension about the optimum right ascension. Each day in the launch opportunity was evaluated separately since the target changed day-by-day. Optimum solutions were obtained, characterized by time dependent launch azimuth, lofting, qV jettison times, and parking orbit elements. For every day in the opportunity, and for each launch time in the window of any day, the optimum trajectory was different (Ref. 8). Vehicle performance met the requirements of the spacecraft project, i.e., the vehicle could lift the spacecraft and place it on its path to Venus. However, this optimum mission design required a very complex Range Safety data package because the entire mission geometry was varying and doing so rapidly. In addition, the verification process would have been tedious, and SUPERDUK would have had to provide time dependent trajectory characteristics for every minute of every launch window for the entire opportunity (Ref. 8).

Figure 31.—Launch of Pioneer Venus on AC-50

When these mission design results were presented to LeRC management, the decision was made to forgo the SUPERDUK analysis (Ref. 8). Rather, LeRC would use the aircraft to provide tracking, allowing much simpler trajectory profiles to be flown, although at the significant cost of the ARIA. Management rightly decided to go with the simpler solution, but at least they were shown what was possible in order to enable an informed decision. Use of the Ascension constrained trajectory would have put a considerable burden on General Dynamics to generate the trajectories. Revisiting the question now, with currently available computer resources, the tracking constrained trajectories would probably have been utilized. The results of the study showed that the Pioneer Venus missions could be launched with the Ascension constraint and meet the payload requirements. The study demonstrated that very complex, constrained trajectories could be produced with very little payload penalty, but operational considerations precluded their use (Ref. 8).

12.0 Other Contemporary 3D Trajectory Optimization Codes

12.1 Observations

The following are a few observations regarding optimization techniques. With the availability of large, fast computers over the last two decades, steepest descent optimization techniques with their flexibilities provide powerful tools for the studies described. However, during the period of 1960s to 1980s, these machines were not available. One of DUKSUP's standout traits was that because it was so tightly coded (because of the hardware limitations), combined with CoV's capability, it was able to produce such great answers so quickly (solutions that to this day may, in some cases, not be easily reproducible). Further, the use of such codes alone increases the possibility of obtaining sub-optimal solutions because of large number of independent variables and the difficulty of "guessing" initial values. This is more of a problem with a steepest descent technique than it is with CoV formulations like that in DUKSUP (although such can happen in CoV codes as well). In addition, with any code, there is the danger that the analyst may not understand the limitations of the codes used. We would suggest that a modern CoV trajectory optimization code be added to the NASA arsenal. It would provide a unique insight and understanding which are not readily apparent with most of the codes in use today. It may not provide the final precise answer, but it may reveal solutions not readily available to other techniques, such as the HEAO solutions and the nonintuitive low thrust coast example discussed earlier. Finally, there is no substitute for careful examination of trajectories to make certain that there are no unintended constraints being imposed. The analyst must look for results that are nonintuitive (but could be important), and to the best of their abilities ensure that the optimum has been obtained.

None of DUKSUP's contemporaries appeared to have the capability to call upon a CoV optimization technique. Though some had more than one optimization algorithm which could be selected, they were largely based on steepest descent or similar techniques. Anecdotal discussions with one of the other launch vehicle organizations regarding sensitivity of initial guesses suggested that almost any reasonable value supplied to the code would produce a converged solution.[10] How close that solution was to an optimum, even a local optimum, appeared to be difficult to gage necessitating multiple runs around the solution to ensure that an optimum had indeed been found. There was a corollary to this observation: that an "expert operator" (or at least an experienced one) seemed to be necessary for a CoV type optimization code like DUKSUP. Whereas a steepest descent or comparable optimization technique which was more forgiving might be more conducive to faster acquisition of the requisite skills necessary to master its operation. The former might be attractive to a federal agency with a longer retention period staff, while the latter might be more attractive to a private sector organization where a nimble, market-driven staffing capability is held in higher regard.

The following is a brief acknowledgement of the contemporary 3D trajectory simulation and optimization codes which were contemporaries of DUKSUP, with which "shooting matches" and interaction at the P, T, & G working groups were regular features of launch planning activities. No attempt is made here to compare performance (speed, accuracy, ease of use, etc.) of these codes since that would be a function of the problem, starting guesses, consistent maturities of the codes, and capabilities of their mainframe platforms (at the very least). What could be asserted, however, is that the following three organizations' codes (together with LeRC's DUKSUP) represented the primary, high fidelity, expendable launch vehicle trajectory design and optimization computer programs in the U.S. from the late 1960s through the mid-1990s.

12.2 TRAJEX

TRAJEX was the 3D trajectory optimization code developed and used by General Dynamics (now Lockheed Martin) corporation, and the code LeRC staff had the most interaction with comparing results. However, no published sources could be found in the open literature documenting this code. It is believed to be based on the early Atlas booster ballistic (constant gravity) code[11] (thus must date back to at least the mid-1950s). It was generally acknowledged during the period of GD's working with the LeRC (1963 to 1998) that TRAJEX used some type of Steepest Descent optimization technique. Good agreement with DUKSUP and TRAJEX output was the norm. TRAJEX was continually correlated to and corrected based on post-flight actuals, which in turn was used to calibrate DUKSUP. A brief reference to the code can be found in GD-internal documentation (Ref. 11).

[10] Williams, C.H., personal recollection, NASA LeRC, San Diego, California, c. 1989.

[11] Spurlock, O.F., personal recollection, NASA LeRC (ret.), 2014.

12.3 POST

Program to Optimize Space Trajectories (POST) was the 3D trajectory optimization code developed and used by Martin Marietta (now Lockheed Martin) Corporation. Built in the 1970 time frame, it was originally intended for Space Shuttle trajectory simulation and optimization. It was adopted and used for expendable launch vehicle applications, including Titan IIIE. Optimization techniques available: primarily the Accelerated Projected Gradient method (a combination of the Rosen's Projected Gradient Method and Davidson's Variable Metric Method for Unconstrained Optimization). It could also call individually: Steepest Descent, Conjugate Gradients, and Davidson's Method of Optimization (Ref. 26).

12.4 GTS

Generalized Trajectory Simulation (GTS) was the 3D trajectory optimization code developed and used by The Aerospace Corporation. Built in 1970 timeframe as well, The Aerospace Corporation used GTS to perform their mission assurance and technical advisement duties (i.e., they were the 'corporate memory' for the USAF) to their client USAF/Space Division (Los Angeles AFS). GTS used a Reduced Gradient Algorithm, where finite difference gradients were computed, and a quasi-Newton update was used to satisfy the imposed constraints (Ref. 27). This code was their primary trajectory design and optimization tool in support of the USAF vehicles such as Titan II/III/IV, Atlas, Delta, and their various upper stages (Ref. 28) In a shooting match performed with DUKSUP in 1987, results were in reasonable agreement (within ~30 lb to GSO for Titan IV/Centaur.)[12] The combined USAF/Space Division-The Aerospace Corporation organizations were the military analogue to NASA LeRC (where launch vehicle management and technical roles were combined).

13.0 Conclusions

This paper was intended to be a history of the development and use of a Calculus of Variations trajectory optimization code we called DUKSUP. It is not an attempt to document the code. Because of the constraints existent at the time of development, documenting it would be a formidable and perhaps impossible task, and probably be intelligible only to the developer. Since the developer was not hired to write a code, but to do analysis, and the problems became more and more complex with time, it was probably not feasible to provide a code developer with a set of requirements that would allow the developer to efficiently formulate an architecture that was flexible enough to attack the problems as they developed over the 30 years of use of the code. The coder would also have to develop an understanding of the CoV derivations with an appreciation of what future problems might require. The developer/user also had the advantage of having developed simpler but similar codes that provided experience with solving the two point boundary value problem.

Though it is tempting to focus largely on the technical accomplishments of this code, the larger picture must be kept in mind. Consider the aggregate of unprecedented national accomplishments that DUKSUP enabled. A great many of the most outstanding interplanetary missions were launched with the critical assistance of DUKSUP. Indeed, NASA Headquarters' program management, not just the Lewis Research Center's leadership, trusted the results of the code and the people's judgment who ran it enough to make critical payload performance commitments to mission programs typically valued in the $100's M. Not once did NASA ever have to reduce a DUKSUP-generated/LeRC-determined payload commitment. These guarantees by the LeRC Center Director, usually years before the launches, were used by the payloads to design and build their spacecraft. As was shown in Figure 22, these missions included Viking, Voyager, Pioneer Venus, and Cassini. The commercial viability of providers such as Intelsat rested on the certainty that DUKSUP's predictions were correct and that the judgment of LeRC executives with regard to margins was sound. Recalling one such episode by a retired LeRC senior executive, "*I especially remember how concerned Frank and I were on making the commitment to Viking, the highest national priority robotic mission ever at the time. We put a lot of sweat into that commitment number. In the end, Frank was never wrong; we always were able to deliver what he promised. Millions upon millions of dollars rode on what he said, as well as the Lewis reputation.*"[13] Only one correction should be made to this quote; where the last sentence should read, '…100's upon 100's of millions of dollars…'

The mission design philosophy at the NASA Lewis Research Center during the years of its leadership of launch vehicle projects (1963 to 1998) recognized the efficacy of extracting all of the payload capability available through trajectory optimization (Ref. 8). All this analysis was done in support of that project philosophy to always know the best possible performance. The critical value of this philosophy was repeatedly demonstrated over more than 35 years and 119 launches during LeRC's tenure leading NASA's unmanned launch vehicle programs for the Intermediate class (Atlas/Centaur) and Large class (Titan IIIE/IV). DUKSUP was ***the*** major LeRC computer code which enabled this modus operandi—and many missions have reaped the rewards of it.

[12] Williams, C.H., personal recollection, USAF/Space Division/CLVD, Los Angeles AFS, 1987.

[13] Nieberding, J.J., personal recollection, NASA LeRC (ret.), 2014.

Appendix

The question sometimes arose, “How did the program get its name?” The lead author had a professor in college who was notorious for transcribing long mathematical derivations during lectures, only to stop just before the critical point in the derivation. He would then proclaim to the students, ‘***The rest is as easy as duck soup.’*** He would end the class without providing the last parts of the solution. This expression was adopted ironically as the name for the code, shortened to six characters, which was the limit for variable and subroutine names in FORTRAN at that time. For everyone except Frank, DUKSUP’s operation and the solutions it discovered were rarely easy to understand.

References

1. Dawson, V.P., and Bowles, M.D., "Taming Liquid Hydrogen: The Centaur Upper Stage Rocket 1958-2002," NASA/SP—2004-4230, Washington DC, 2004.
2. Strack, W.C., Huff, V.N., "The N-Body Code – A General FORTRAN Code for the Numerical Solution of Space Mechanics Problems on an IBM 7090 Computer," NASA Lewis Research Center, NASA TN D-1730, November 1963.
3. Spurlock, O.F., and Teren, F., "Optimum Launch Trajectories for the ATS-E Mission," AIAA Journal of Spacecraft, Vol. 8, No. 12, pp. 1202 – 1208, December 1971.
4. Teren, F. and Spurlock, O.F., "Payload Optimization of Multistage Launch Vehicles," NASA TN D-3191, 1966.
5. Pontryagin, L.S., et al., "The Mathematical Theory of Optimal Processes," Interscience, New York, 1962.
6. Teren, F., Spurlock, O.F., "Optimal Three Dimensional Launch Vehicle Trajectories with Attitude and Attitude Rate Constraints," NASA TN D-5117, 1965.
7. Zimmerman, A.V., MacKay, J.S., and Rossa, L.G., "Optimum Low-Acceleration Trajectories for Interplanetary Transfers," NASA TN D-1456, 1963.
8. Spurlock, O.F., and Nieberding, J.J., "Optimum Constrained Mission Design Considerations for Ares I," Alphaport Inc., Cleveland, OH, September 2007.
9. Zimmerman, A.V., "Application of the Calculus of Variations to the Trajectory Optimization of a Single Stage Space Vehicle," unpublished document, NASA Lewis Research Center, Cleveland, OH, February 1991.
10. McShane, E.J., "Gilbert Ames Bliss 1876 – 1951 A Biographical Memoir," National Academy of Science, Washington DC, 1958.
11. "Centaur D-1A Systems Summary," General Dynamics Convair Aerospace Division, Report No. GDCA B NZ 72-020, Contract NAS3-13514, San Diego, CA, Sept 1972.
12. "Shuttle/Centaur Program," General Dynamics Convair Division, San Diego, CA, 1983.
13. Sjauw, W., "Cassini Variable Flight Azimuth Launch Windows," Technical Report, Analex Corporation, Brook Park, OH, 31 January 1994.
14. Burdick, R.A., "Alternative DUKSUP Procedure for Controlling Vehicle Instantaneous Impact Point (IIP)," Technical Report, Analex Corporation, Brook Park, OH, 6 May 1991.
15. Spurlock, O.F., NASA Glenn Research Center, Cleveland, OH, personal notes, 2014.
16. http://en.wikipedia.org/wiki/UNIVAC_1100/2200_series
17. "IBM 370/3033," International Business Machines, New York, NY, undated brochure.
18. http://www-03.ibm.com/ibm/history/exhibits/3033/3033_TR02.html
19. Kolodzey, J.S., "Cray-1 Computer Technology," IEEE Transactions on Components, Hybrids, and Manufacturing Technology, Vol. CHMT-4, No. 2, June 1981.
20. "The Cray-XMP Series of Computer Systems," Cray Research Corp., Minneapolis, MN, 1986.
21. http://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframe_PP7094.html
22. Sergeyevsky, A.B., Snyder, G.C., "Interplanetary Mission Design Handbook," Volume 1, Part 3, JPL publication 82-43, NASA Jet Propulsion Laboratory, Pasadena, CA, 1 December 1982.
23. Nieberding, J.J., "Ulysses/Galileo Performance Status," Level I Meeting, NASA Lewis Research Center, 16-17 January 1986.
24. Robbins, W.H., "Shuttle/Centaur Project Office Weekly Status Report," NASA Lewis Research Center, Cleveland, OH, 3 October 1985.
25. Isokowitz, S.J., "AIAA International Reference Guide to Space Launch Systems," 2nd Edition, American Institute of Aeronautics and Astronautics, Reston VA, 1995.
26. Brauer, G.L., Cornick, D.E., Stevenson, R., "Capabilities and Applications of the Program to Optimize Simulated Trajectories (POST)," NASA Contractor Report CR-2770, prepared by Martin Marietta Corp (for NASA LaRC), February 1977.
27. Betts, J.T., "A Survey of Numerical Methods for Trajectory Optimization Mathematics and Engineering Analysis," Boeing Information and Support Services, Seattle, WA, August 1998.
28. "Generalized Trajectory Simulation," Volume 1: Overview, Final Report, Air Force Rocket Propulsion Laboratory, Report SAMSO-TR-75-255, Vol. 1, The Aerospace Corporation, El Segundo, CA, 21 November 1975.