Rendezvous Simulation of the Automated Transfer Vehicle with the International Space Station

By
Sidharth Saraf, B.Eng.

A thesis submitted to
the Faculty of Graduate Studies and Research
in partial fulfillment of
the requirements for the degree of
Master of Engineering

Ottawa-Carleton Institute for
Mechanical and Aerospace Engineering

Department of
Mechanical and Aerospace Engineering
Carleton University
Ottawa, Ontario
April 14, 1998

0-612-32407-9

Abstract

Simulation of an automated space rendezvous is a necessary step in designing the rendezvous algorithms for a space vehicle. This thesis describes a simulation for a particular rendezvous mission involving the International Space Station (ISS). The ISS program will require automated rendezvous technology for the station's supply vehicle, the Automated Transfer Vehicle (ATV). The European Space Agency (ESA) is responsible for the design and construction of the ATV. ESA contracted CAE Electronics Ltd., Montreal, to deliver a space vehicle simulator including a simulation to demonstrate an ATV specific automated rendezvous mission. This thesis describes the rendezvous algorithms that were developed by the author for implementation within the simulation software package, CAE ROSE™. The algorithms determine the 'burns' or Delta-V required by the ATV for the rendezvous phase. This thesis identifies part of the rendezvous phases that were used in the real-time simulation. The rendezvous phases include the 'homing phase' which brings the ATV to the ISS orbit and the 'closing phase' which brings the ATV closer to the ISS in the same orbit. The correction burns required in case of drift from predicted trajectories are also covered. The solution to the 'Kepler problem' and 'the Gauss problem', solved using universal variables, were part of the algorithm. The thesis concludes with a brief introduction to perturbations and their effects on such a mission. The simulator including the ATV specific simulation was accepted by ESA for the ATV program. ESA comments on the differences between the actual rendezvous algorithms and the simulation are discussed.

Acknowledgments

A teacher does not teach us anything new, he only helps us remember that which we always knew. Thank you Dr. Douglas A. Staley.

I would like to thank Dr. Peter Frise, Dr. Xavier Cyril, Mr. Jean St-Pierre, and the entire team in the Space Department at CAE Electronics, Ltd.; without them this thesis project would not have even begun.

My special thanks to Mr. Chris Cull, Mr. Marc-Élian Bégin, and Mr. Siamak Tafazoli from CAE Electronics, Ltd., for their never-ending support and excellent problem solving skills.

Finally, my thanks to Mr. Gilbert Pitre, Mr. David Prodger, Mr. Henry Saari, and Mr. Gerry Warner for their reviews.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ATV	Automated Transfer Vehicle
ISS	International Space Station
ESA	European Space Agency
ROSE™	Real-time Object-oriented Software Environment
MSVS	Multi-purpose Space Vehicle Simulator
COM	Centre Of Mass

Nomenclature

a	semi-major axis
e	orbit eccentricity
i	orbit inclination
Ω	right-ascension of the ascending node
ω	argument of periapsis
ν	true anomaly
E	eccentric anomaly
P	orbit period
n	mean motion
ξ	specific mechanical energy an orbiting spacecraft
$\vec{h}$	specific angular momentum of an orbit
$J2$	Earth oblateness coefficient
$\vec{r}$	spacecraft position vector in geocentric inertial frame
$\vec{v}$	spacecraft velocity vector in geocentric inertial frame
μ	Earth's gravitational constant
x, z	universal variables
t	time-of-flight

Chapter 1

1.0 INTRODUCTION

The construction, in space, of the International Space Station (ISS) is scheduled to begin in June of 1998. During construction and after completion, an Automated Transfer Vehicle (ATV) will be required to rendezvous with the ISS for support. The key feature of the ATV is its onboard guidance, navigation, and control capability. Generally, spacecraft trajectories are computed at ground stations and control commands are uplinked to the spacecraft. The ATV will be designed to carry out these processes onboard while being monitored by the ground stations. The task of designing and constructing the ATV has been assigned to the European Space Agency (ESA). Various spacecraft simulation tools are used by ESA in the design and development of its spacecraft. CAE Electronics Ltd. (CAE), in Montreal, was contracted by ESA to develop and deliver a new simulation tool that could be used for its ATV and future spacecraft development programs. CAE was required to demonstrate the software tool by delivering a simulation of the ISS and ATV rendezvous scenario. The project at CAE was called the Multi-purpose Space Vehicle Simulator (MSVS). The software tool developed by CAE, and used for the ISS/ATV simulation, is called CAE ROSE™ (ROSE™), an acronym for Real-time Object-oriented Software Environment.

This thesis describes the algorithms of some of the ATV's rendezvous phases implemented in ROSE™ as part of the MSVS project. Specifically, the algorithms for the homing and closing rendezvous phases and the corrections required during these two phases were developed, written in the C programming language, and implemented in ROSE™ by the author. The algorithms determine the Delta-V burns required by the ATV thrusters to allow it to complete the homing and closing phases successfully. These algorithms are part of the ATV's onboard software in the simulation.

A literature search on the state-of-the-art methods for an autonomous rendezvous was made by the author, however, no literature on the subject was found. Space vehicles that use this technology include the Progress vehicles that service the Salyut and Mir space stations. There are no published papers on the algorithms used by this vehicle. The algorithms developed for the ISS/ATV simulation are therefore unique and are governed by the ISS/ATV mission and the ROSE™ software tool. The orbit control algorithms are generally processed at ground stations and uplinked to the spacecraft. The mission operations group that monitor and control the spacecraft develop their own algorithms which are unique to their application. The Space Shuttle Atlantis tested the new ESA rendezvous and docking technology on September, 1997. Thus, automated rendezvous is a fairly new and unique technology. The ISS/ATV rendezvous mission is described in Chapter 2 as a definition for the simulation. The ROSE™ software package and the overall ISS/ATV simulation architecture is covered in Chapter 3. The ROSE™ software is described to the reader since it is fundamental in the development of the algorithms. The algorithms were developed specifically to interface with the entire simulation of the ISS, the ATV, and the space environment. A background of orbital mechanics and space rendezvous is covered in Chapter 4 mainly for the reader who is unfamiliar with the theory of orbits. The algorithms that solve the Kepler problem and the Gauss problem are used in the determination of the Delta-V's required by the ATV for the homing and closing phases. The homing and closing rendezvous phases are described in Chapter 5 and the Kepler and Gauss problems are covered in Chapter 6. The phenomenon of perturbed spacecraft orbits was also investigated for this rendezvous mission and is described in Chapter 7. Finally, the differences between the rendezvous algorithms implemented in the ROSE™ simulation and the actual ATV's onboard software will be covered. Some conclusions reached by the author and recommendations for future study are found in Chapter 8.

1.1 Multi-purpose space vehicle simulator

The MSVS project was awarded to CAE in August 1994 and delivered in its entirety at ESA's European Space and Technology Centre (ESTEC) in Noordwijk NL on April 1997. A portion of this project was subcontracted to a company called TRASYS in Belgium and another company called VEGA GmbH in Germany. MSVS is a research and development simulator implemented in CAE's ROSE™ software package. MSVS consists of generic spacecraft simulation models, such as attitude control, thermal subsystem, orbit dynamics, etc., and provides the capability of simulating a full mission. It also consists of a specific simulation of the ATV on a mission to rendezvous and dock with the ISS. The MSVS project was developed to be used as a template by ESA for the actual ISS/ATV simulation. The ISS/ATV simulation delivered by CAE was a demonstration of MSVS's capability.

The MSVS allows users to:

1) Design prototype space vehicles in a simulation environment

2) Perform spacecraft dynamics analysis

3) Create prototypes and verify existing spacecraft design concepts

Some technical achievements include the ISS/ATV mission simulation and the capability to design prototype spacecraft in a real-time simulation environment. The future potential of MSVS includes flight hardware-in-the-loop (HITL), software-in-the-loop (SITL) simulation and man-in-the-loop simulation capability [1].

This thesis will include an overview of the development and use of a space vehicle simulator, focusing on the rendezvous algorithms that were used as part of the ATV onboard software models.

1.1.1 Defining spacecraft simulation requirements

The specific mission of a spacecraft is fundamental in determining the type of models that will need to be created and re-used in a spacecraft simulator. The spacecraft itself must be carefully analyzed to determine the type of models required and the architecture of the simulation. The analysis should provide information about the different mission phases and vehicle subsystems that need to be simulated. Figure 1 generally describes this process:

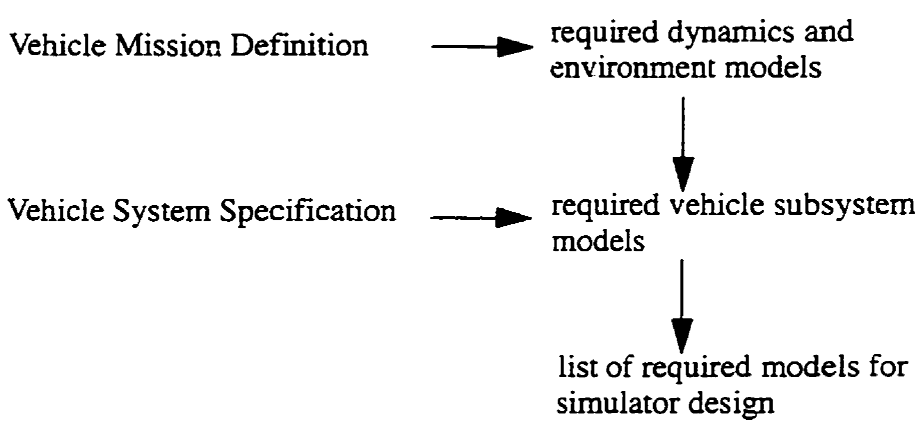


FIGURE 1. Process for Defining the Simulation Requirements [2]. The figure describes a method that can be used to develop a list of individual models required for a specific spacecraft simulator.

1.1.2 Simulation design

The ability to utilize existing simulation models is the primary consideration in the design of a multi-purpose spacecraft simulator. A template project called *baseline*, containing generic spacecraft objects, was developed. Other projects may be used as a template, but only the *baseline* project is officially validated in ROSE™. The validated project was copied into the MSVS project. The MSVS project was designed to be a template for other spacecraft simulations. Figure 2 describes the re-usability process of ROSE™ projects.

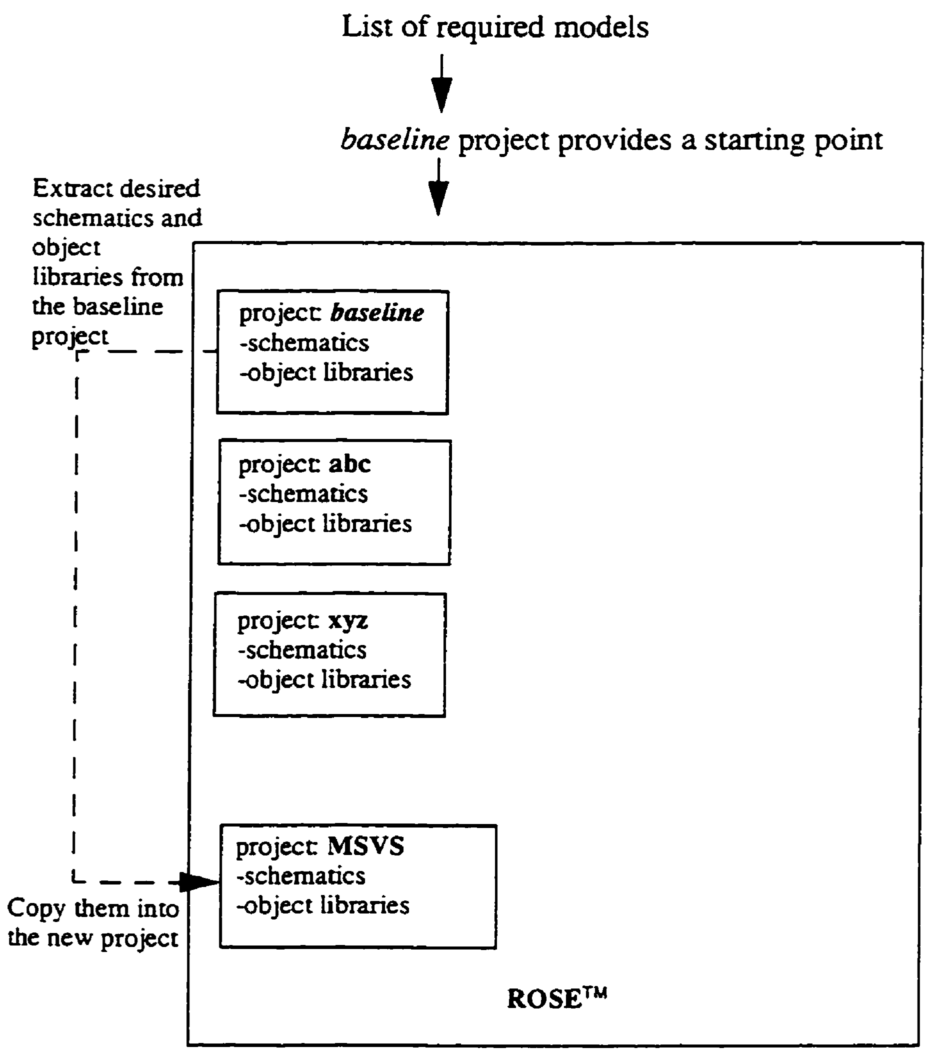


FIGURE 2. Building a Simulator from the Baseline Project [2].

The extracted schematics can be modified to conform to the requirements of the space vehicle of the new project. The modifications can be in the form of changing the copied schematics, creating new schematics using existing objects from the object libraries, or creating new objects in the object libraries [2]. In this case, the MSVS was the new project. The ISS/ATV simulation was developed by using generic objects created in libraries that belong to the MSVS project. These libraries include propulsion, thermal, onboard software, and others that were required for a complete space mission simulation. The simulation itself is developed on various schematics. Chapter 3 describes how objects, schematics, and libraries are related in a spacecraft simulation. The next chapter will outline the ISS/ATV mission.

Chapter 2

2.0 ISS/ATV MISSION OVERVIEW

The mission for the ATV primarily involves servicing the International Space Station. This includes the re-supply and de-supply of the station and the delivery and disposal of the station infrastructure elements. The following sections briefly outline some of the parameters of the ATV, the ISS, and the scenario in the generic ROSE™ simulation [3].

2.1 ATV performance specifications

The purpose of this section is to provide the reader with an overview of the ATV specifications used for the simulation. The following ATV specifications briefly describe some of the characteristics of the ATV and its required performance parameters.

The performance requirements for the ATV include:

- achieving a phasing orbit after injection
- rendezvous with ISS
- departure from the ISS and de-orbit

The ATV's target orbit parameters include:

- altitude: 350 km - 460 km

• inclination: 51.6 degrees

Cargo mass with injection into a 70 x 300 km orbit by the Ariane 5 launch vehicle:

• reference case: 11,000 kg

Maximum mass at injection for the ATV and cargo:

• reference case: 16,300 kg

The reference case is used for the simulation and may be different from the actual sizing case. One of the missions for the ATV may be to re-boost the ISS to a higher altitude, using its own thrusters and the cargo propellant. The reboost mission includes the ISS with a mass of 353,800 kg and refuelling capability includes 1000 kg of propellants.

2.1.1 ATV operational requirements

The following includes some of the scenarios during the ATV's operational life.

- docking and berthing
- retreating to safe hold points on request from the ground station, the space station or automatically in case of a communication loss
- remaining attached to the space station for up to 6 months
- after mission completion the ATV can deorbit and perform a destructive re-entry with debris impacting authorized zones

2.1.2 ATV orbit navigation

The ATV uses the Global Positional System (GPS) to determine its absolute position and velocity. Communication during ascent and descent occurs with the ground station and through Data Relay Satellites.

2.1.3 ATV failure tolerance requirements

The failure tolerance of the ATV includes a single failure or operator error that will have no critical consequences. Two failures and/or operator errors shall have no catastrophic hazardous consequences.

2.1.4 ATV interface to ISS

Figure 3 illustrates the ISS frame of reference. The ISS velocity direction is called the V-bar direction. The R-bar direction points to the centre of the Earth. The H-bar direction is opposite to the angular momentum vector of the ISS orbit. Approach along the negative V-bar (tangential or roll axis) direction to the Russian segment includes direct docking (Figure 4). Approach along the R-bar (radial) direction to the American segment includes a Common Berthing Adaptor (Figure 5).

2.1.5 ATV reference mission

The ATV reference rendezvous mission includes:

- Mixed cargo / ISS reboost (negative V-bar approach)
- Un-pressurized supply mission (positive R-bar approach)

Figure 4 and Figure 5 illustrate the two reference missions and the ATV's approach directions to the ISS:

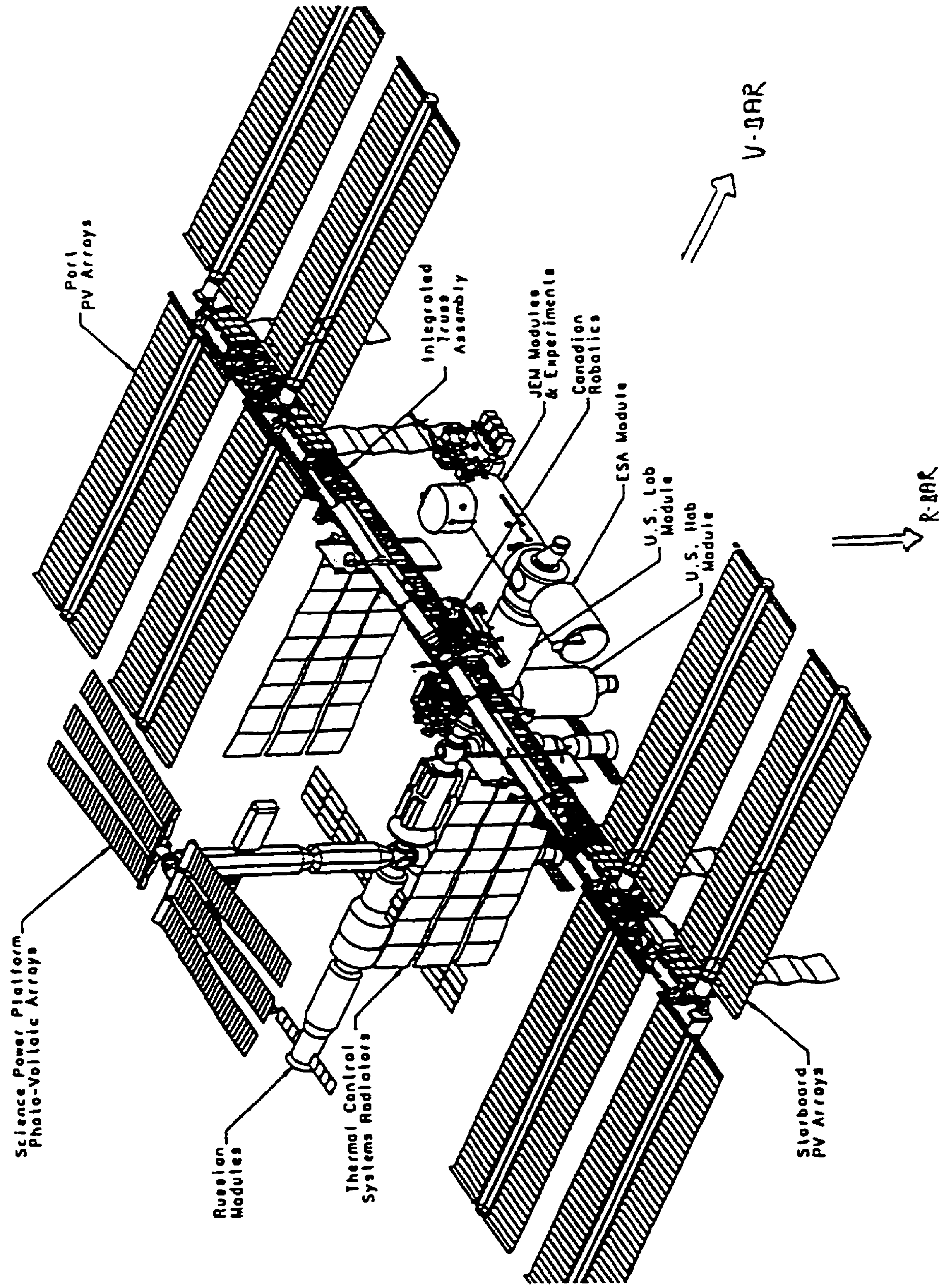


FIGURE 3. The ISS Frame of Reference [4, 14]. The ISS velocity direction is called the V-bar direction. The R-bar direction points to the centre of the Earth. The H-bar direction is opposite to the angular momentum vector of the ISS orbit.

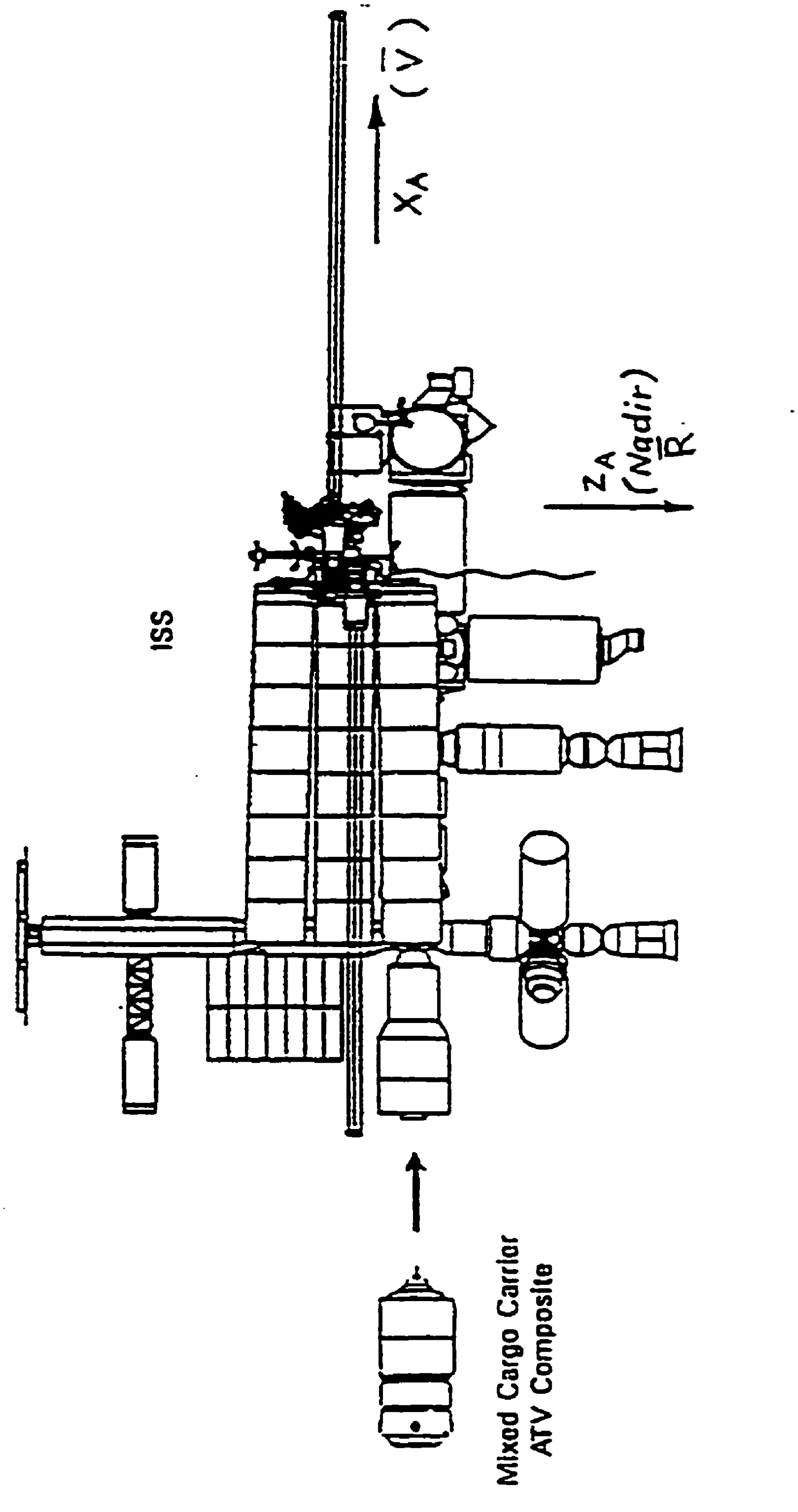


FIGURE 4. Negative V-bar Approach to Docking for Mixed Cargo/Re-boost Mission [3, p. 9]. Approach along the negative V-bar (tangential or roll axis) direction to the Russian segment includes direct docking.

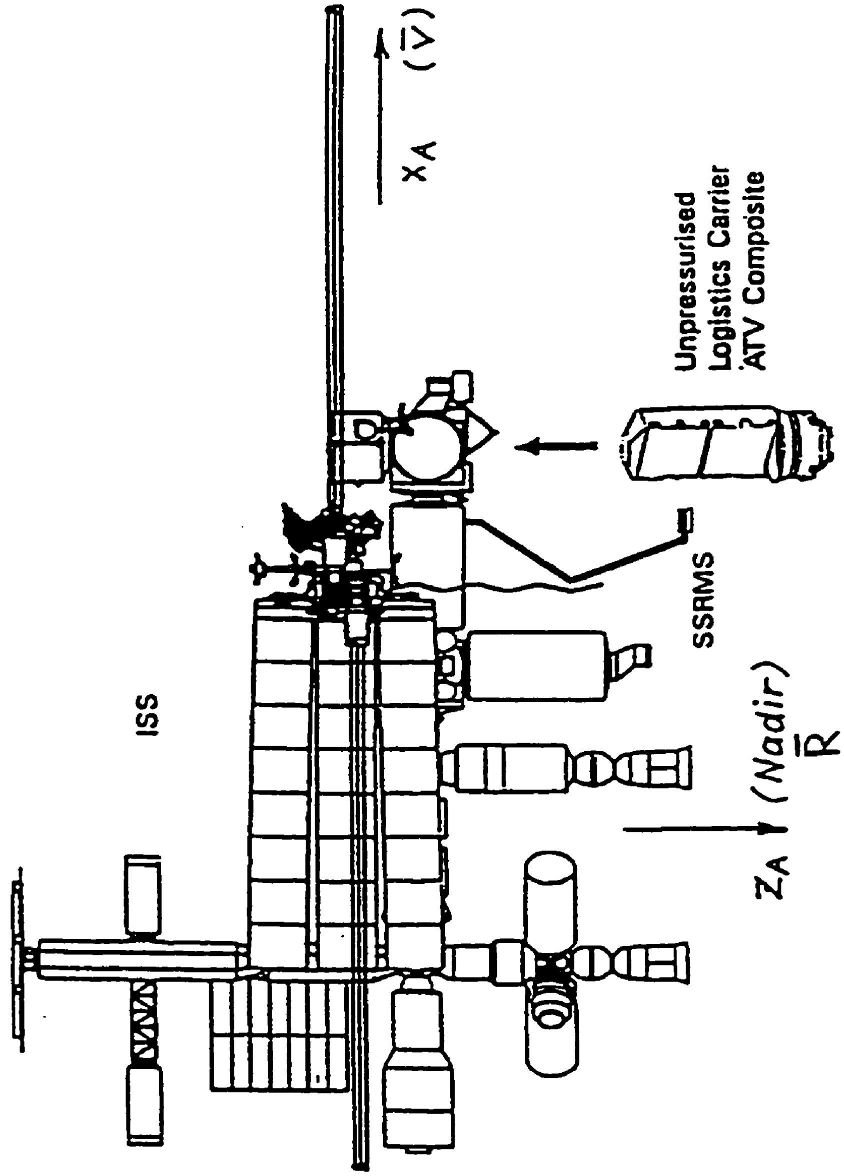


FIGURE 5. R-bar Approach for Unpressurized Supply Mission [3, p. 10]. Approach along the R-bar (radial) direction to the American segment includes a Common Berthing Adaptor.

2.2 International Space Station

This section describes the space station orbit used as a reference for the simulation. The altitude of the space station, once complete, shall range from 350 km to 460 km. This range depends on the eleven year solar activity cycle; the higher the solar activity, the higher the density of the Earth's atmosphere. This requires the altitude of the space station to be higher due to the increased atmospheric drag in low-Earth orbit (LEO). The inclination of the space station is chosen to be 51.6 degrees. At this inclination the space station will last up to 90 days before it falls to 278 km due to aerodynamic drag from its minimum design altitude of 350 km. The minimum design altitude is where the ISS requires a reboost. Therefore, 350 km is the altitude used in the simulation. The maximum altitude of 460 km was chosen for ISS budgeting and sizing purposes [3].

2.2.1 ISS command and control authority

The rendezvous algorithms developed for the simulation are used for the ATV when it enters the ISS communication range. There are various virtual boundaries defined around the ISS COM. These zones define the phase of the mission and the control authority between the ISS astronauts and the ground station. The Command and Control Zone (CCZ) is distinguished between the following two concepts:

1) Space Station Communication Range (SSCR)

The UHF transmitter onboard the space station determines the size of the SSCR. The reference size is found in Figure 7.

2) Space Station Approach Ellipsoid (SSAE)

The size of the SSAE is 4 x 2 x 2 km along the V-R-H- bar directions respectively as shown in the figure below:

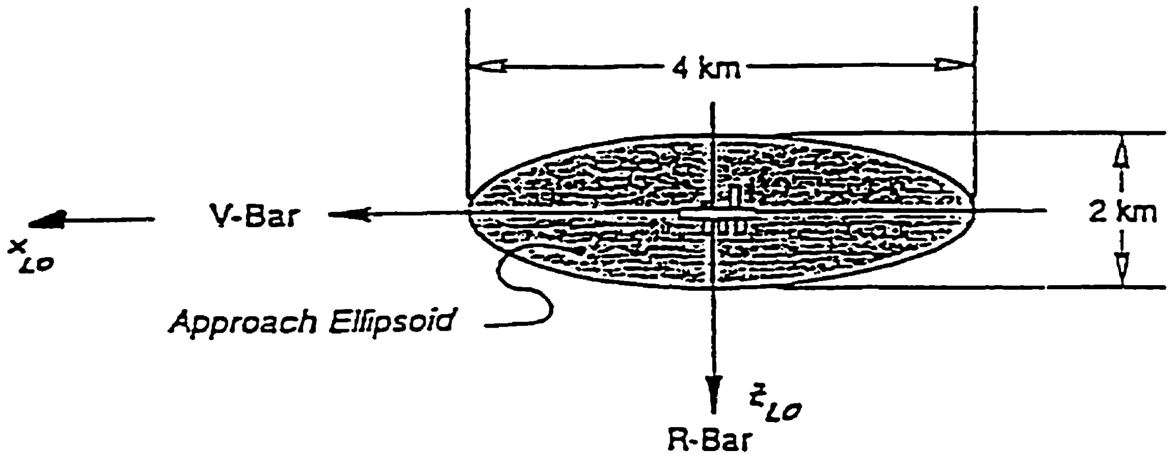


FIGURE 6. Size of the Space Station Approach Ellipsoid [3, p.12]. There are various fictitious boundaries defined around the ISS COM. These zones define the phase of the mission and the control authority between the ISS astronauts and the ground station. The size of the SSAE is 4 x 2 x 2 km along the V-R-H- bar directions respectively.

2.3 ISS/ATV simulation mission phases

The following summarizes the ATV and ISS simulation mission phases:

- Injection into a 70 x 300 km orbit by the Ariane 5 launch vehicle
- Transfer into a phasing orbit
- Phasing orbit for a maximum of 46 hrs
- Transfer to the ISS target Orbit
- Rendezvous (RVD) phases include:

  34 hrs (maximum for hold points)

  10 hrs (maximum for RVD) + 10 hrs (maximum for contigency RVD)
- Final docking phase

The total mission times adds up to 100 hrs (approximately four days) if the maximum allowable times for each phase of the mission are used [3].

2.3.1 Rendezvous phases

The following phases are included in the rendezvous mission:

1] Homing phase

This phase begins when the ATV enters the SSCR and receives the 'go-ahead' from the ISS after the diagnostics are complete. At the end of this phase the ATV reaches the target space station orbit, 2500 m behind the space station centre of mass.

2] Closing phase

This phase ends when the ATV is located approximately 300 m behind the space station at the same target orbit as the space station. This phase can be performed in more than one step. The ATV can be brought closer behind the space station in short 'jumps'. The current simulation brings the ATV from 2500 m to 750 m behind the space station in the first 'jump'. The second 'jump' brings the ATV to approximately 300 m behind the ISS.

There is also a correction algorithm which ensures that the ATV remains on a desired trajectory during the homing and closing phases of the rendezvous. This algorithm calculates an additional burn if the ATV strays off course by a pre-determined position magnitude between the actual and predicted trajectories. Figure 7 illustrates the homing and closing phases described above.

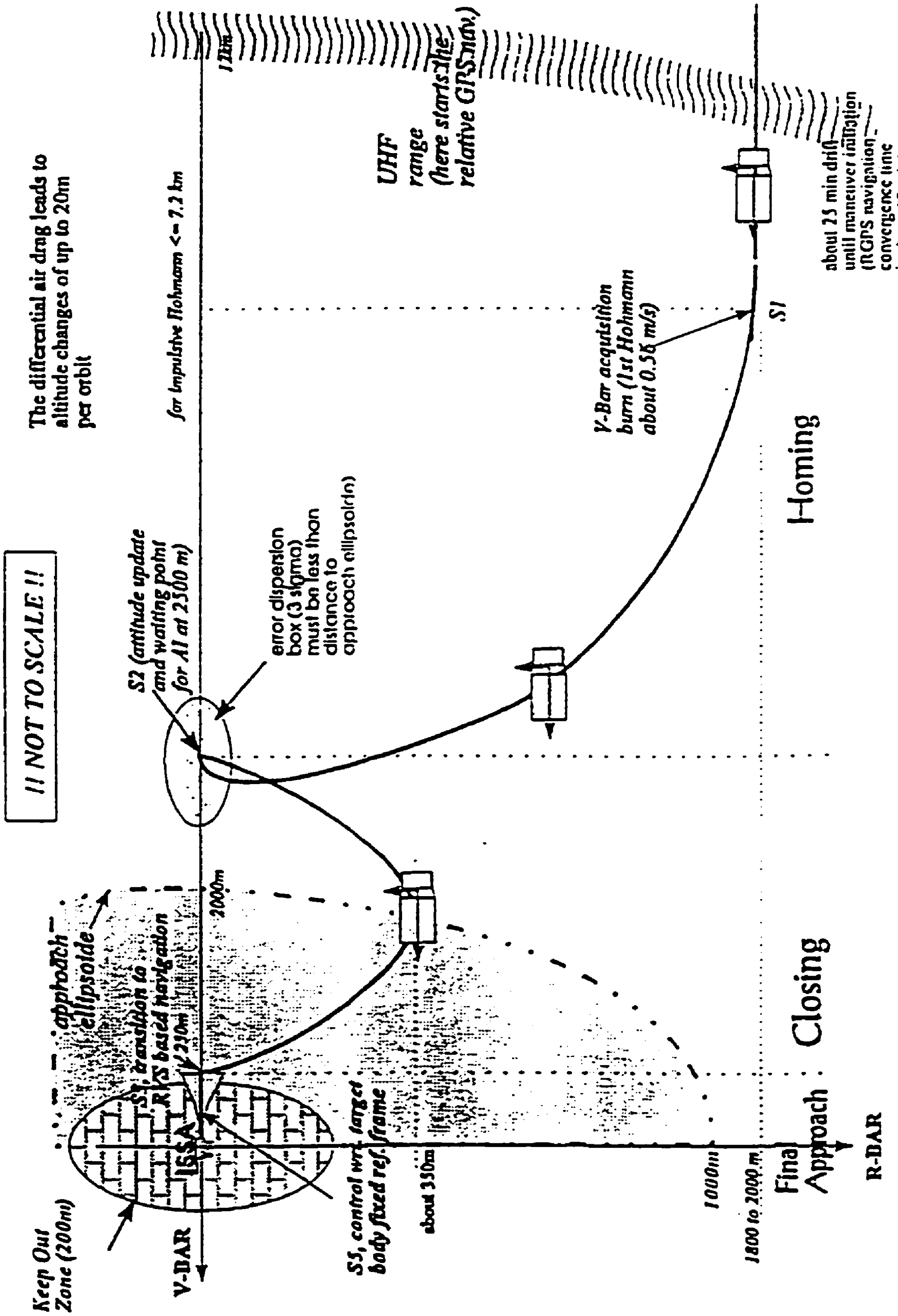


FIGURE 7. Homing and Closing Rendezvous Phases for -V-bar Approach [3, p.21]. The homing phase begins when the ATV enters the SSCR and receives the 'go-ahead' from the ISS after the diagnostics are complete. At the end of this phase the ATV reaches the target space station orbit, 2500 m behind the space station centre of mass. The closing phase ends when the ATV is located approximately 300 m behind the space station at the same target orbit as the space station.

The algorithms were developed for the homing and closing phases of the rendezvous mission. These algorithms were implemented on schematics as part of the entire simulation. The homing phase of the rendezvous begins when the ATV enters the SSCR, which is 12 km behind and 2 km below the space station centre of mass. The relative position of the ATV once this phase begins is (-12km, 0km, 2km) in the V-H-R bar frame of reference (Figure 3). The closing phase ends when the relative position is approximately (-300m, 0km, 0km) in the V-H-R bar frame of reference (Figure 3). There will be dispersion in the relative positions mentioned above due to perturbations in the actual thrusts applied to the ATV. The ATV does not follow the predicted trajectory because of the errors involved in the ATV thruster burns. These dispersions required the development of a schematic that computes the correction burns that return the ATV to a desirable rendezvous trajectory.

The development of the algorithms was governed by the mission parameters and the software tool itself. An overview of the ROSE™ software and ISS/ATV simulation architecture is provided in the next chapter.

Chapter 3

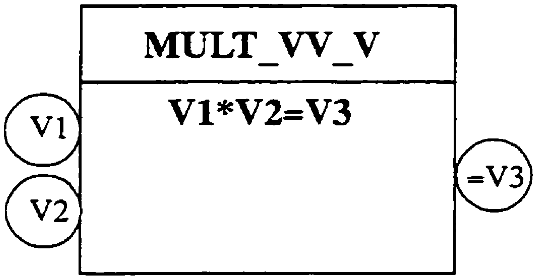
3.0 ROSE™ OVERVIEW

Models for spacecraft simulation in ROSE™ are made up of objects from generic object libraries (e.g. Propulsion subsystem library containing a valve object) and schematics related to a specific project (e.g. ATV subsystem specific schematics). A ROSE™ object consists of an unconnected component without data flow from other objects. A schematic is a collection of these objects that are connected through data flows. The same object can be used as many times as required in the same schematic. Data can flow from one schematic to another via objects. Therefore, one subsystem can be modeled in a schematic and can interact with another subsystem modeled in another schematic. This allows for a completely integrated spacecraft simulation environment.

3.1 Objects

An object can perform a simple algebraic calculation (addition, multiplication) or a complex algorithm extracted from an existing non-ROSE™ simulation (e.g. C or Fortran subroutines). A ROSE™ object is defined in the following ways:

1) Its graphical representation:



2) Variables

3) Connect points (for data flow)

4) Pseudocode (includes calls to existing subroutines).

The code generator creates high-level code which calls the individual component models. Code generators are used to translate the pseudocode into executable code at build time. The subroutines that are called by the pseudocode can be in Fortran, C or Ada. This external code is linked during build time. The same method is used to launch external applications during the simulation such as pre-processors, animation packages, etc.

The data flows through connection points from one object to another. The objects are connected through connect points that are defined within an individual object. The variables are defined within an object and are associated to each connect point on the object. This allows for graphical connectivity from one object to another. The data from the variable of one object can flow to another object through their respective connect points [5].

3.2 Schematics

A schematic contains objects that are connected together to allow the appropriate data manipulation and flow. An example of how objects are 'dragged and dropped' from object libraries into schematics is illustrated in the following figure:

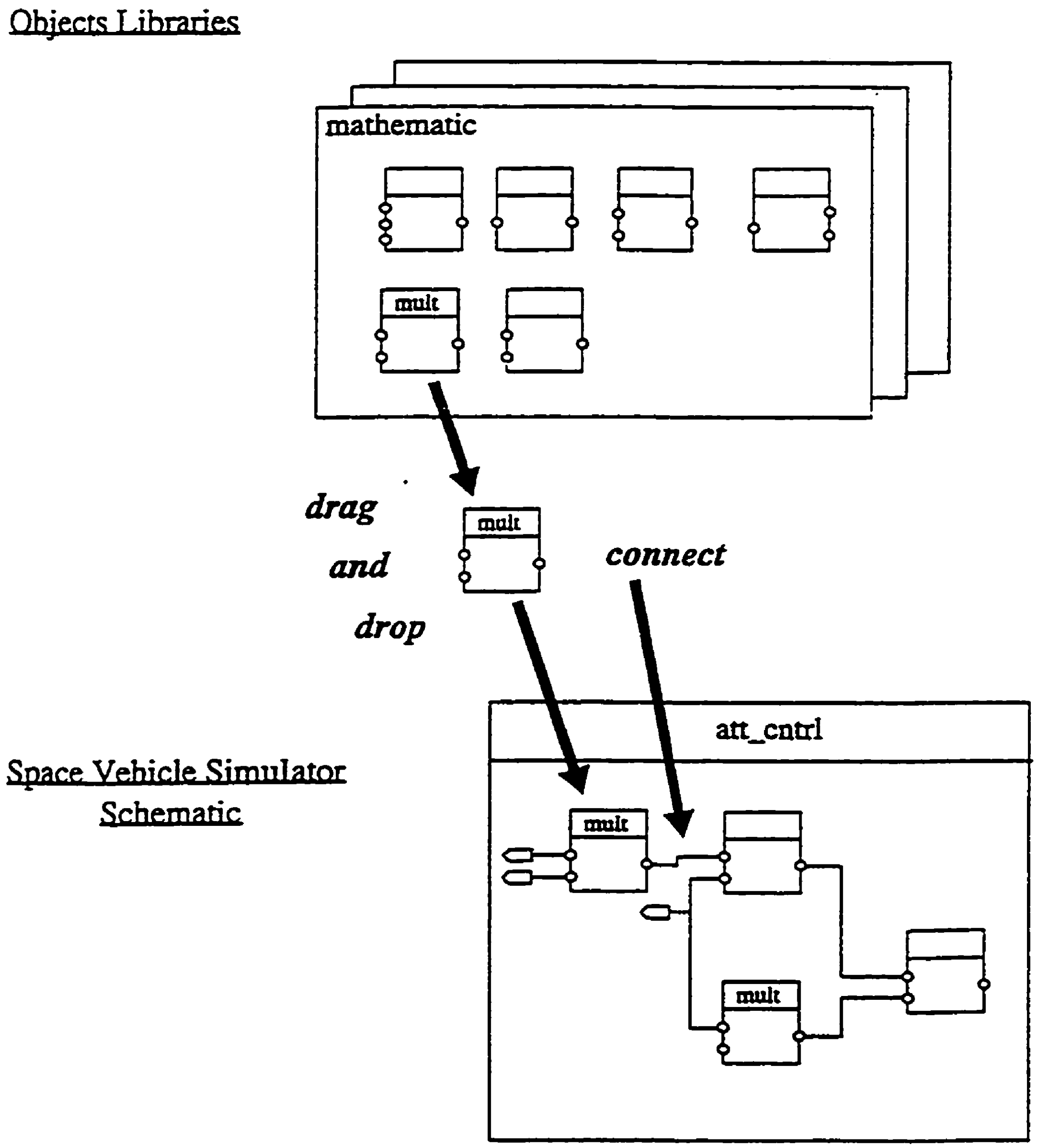


FIGURE 8. Object Libraries and Schematics [2, p.9]. The figure illustrates the use of objects from a library in a simulation schematic. The example shows a mathematical library containing a multiplication object called 'mult' being 'dragged and dropped' into an attitude control schematic called 'att_cntrl'. The schematics is part of the simulation and uses various objects from existing libraries.

The above figure shows a multiplication object called 'mult' being 'dragged and dropped' from the mathematic library into an attitude control schematic called att_cntrl. The mathematic object library contains various math objects while the attitude control schematic represents a model of a spacecraft attitude control subsystem. The schematics may perform the function of a model, part of a model, or several models. (i.e. The propulsion subsystem or simply one thruster could be modeled in a schematic). The data may flow within a schematic or between schematics through object connections. This allows for complex subsystems, such as the electronics, to be modeled in more than one schematic to make the simulation more user friendly.

3.3 Build time

The run-time executable must be 'built' once the models have been completed on various schematics. This is where the code is compiled and linked to create the simulation executable. The following figure describes the ROSE™ model development process.

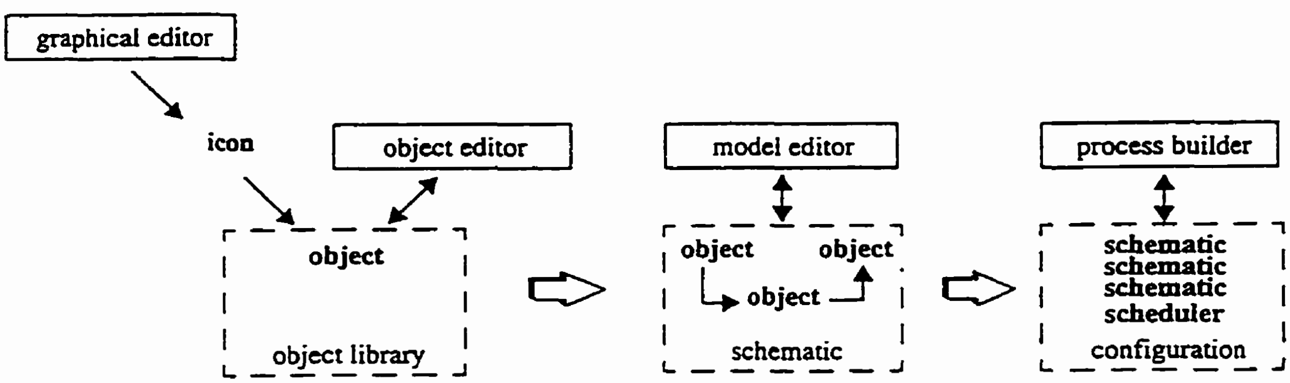


FIGURE 9. Model Development in ROSE™ [2, p.10]. The figure illustrates the steps in creating a simulation from creating objects in the object libraries to building all the schematics required for the simulation. The object editor is used to create objects in the libraries. The model editor is used to develop the schematics which are used as the simulation models. This is where the objects and schematics are connected together to allow for the flow of data. The process builder is used to create the simulation executable containing the code from all the schematics.

3.3.1 Simulation scheduler

To decrease the processing in the computer, the models are scheduled at different update rates depending on the accuracy they require. If the simulation contains a large number of models, running all the models at the same rate could overload the computer's processing capability. This could result in a non-real-time simulation. Scheduling the various models ensures that the computer will be able to run more complex simulations since the rate at which different models are run is optimized for the desired accuracy. For example, the attitude control models should be updated every iteration while the thermal models can be scheduled to be updated every four iterations to maintain their respective accuracies. This is because attitude control models require more frequent updates compared to the thermal models. The developer of the simulation can schedule the schematics in the order and at the rate required by specifying them in a scheduling tree. The schematics are placed in the proper order on the tree band scheduler. The tree can be modified using the ROSE™ process builder [2]. Figure 10 describes the process of scheduling a set of schematics.

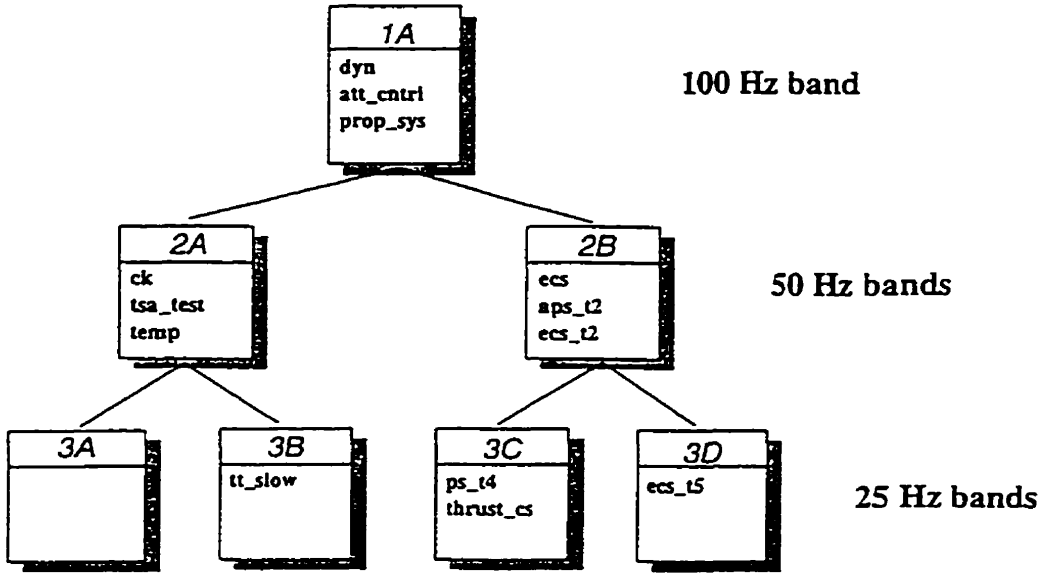


FIGURE 10. Scheduling Tree [2, p.11]. The figure illustrates the capability of scheduling various schematics at different update rates. The schematics can be placed in the desired frequency bands by the user through the process builder. This tool allows the simulation to remain in real-time by decreasing the processing required.

Figure 10 is an example of how schematics such as **dyn**, **att_cntrl**, and **prop_sys** are called in the fast 1A band, while the schematics in the 2A band; **ck**, **tsa_test**, and **temp** are called on alternate iterations and all the schematics in the 3A band are called on every fourth iteration. This would be similar to having the attitude control subsystem, the propulsion subsystem, or the spacecraft dynamics schematic being updated at a higher frequency for accuracy while updating the thermal subsystem at a lower frequency. The fastest rate shown is 100 Hz. This rate may be set by the developer. The other bands follow and are executed accordingly at the corresponding fraction of the primary rate as shown in the figure above.

3.3.2 Code generators

The purpose of the code generator is to produce a high-level executable code which can include calls to external subroutines included in an object. The rendezvous algorithms are modeled on schematics with objects that are called by the sequential code generator. The executable code is generated during build time using the process builder tool. ROSE™ supports both special and sequential code generators. The order of the objects in the schematics is determined by the developer. This determines the order in which the code is generated. The order of the objects is used by the sequential code generator and is the only process that the user controls in code generation. The objects in a schematic may be processed by more than one code generator.

The sequential code generator is used by default in ROSE™. The object's pseudocode is placed in the order the object is assigned on the schematic. The complete code is then generated in the order it appears on the schematic. The code that is called by each object (i.e. non-pseudocode) is treated the same way. The schematic is parsed to call the referenced code in the sequence the object is assigned. The generated code is then translated into the user-defined target code language.

The special code generator is used for special applications. An example is a network solution as in the case of the electrical, thermal, or the propulsion subsystem. The special code generator seeks special variables for the input and output. The network solution is solved numerically during run-time using the connection between the special variables [2].

The schematics developed for the homing and closing phases use the sequential code generator. The corrections schematic also uses the sequential code generator. In general, the objects are ordered from the left side of a schematic to the right side. The inputs to the homing, closing, and corrections schematics, from other subsystems, flow into objects on the left side of the schematics. The outputs from the rendezvous schematics flow out of objects on the right side of the schematics. This follows a user friendly logical order.

ROSE™ supports the use of additional code generators that may be more suitable for the particular application. The schematics can be found in Appendix B.

3.4 ISS/ATV simulation models overview

This section will describe the overall ROSE™ simulation of the ATV and the ISS. The models required to simulate the ISS/ATV rendezvous mission from launch to docking are briefly described. The space vehicle simulator is divided into four main categories which are then divided further into more detailed simulation models. Figure 11 illustrates the simulation architecture high-level breakdown. Figure 12 illustrates the four main categories of the ISS/ATV simulation, which include the dynamics and environment, the external elements, the simulation control, and the space vehicle subsystems. The remainder of this section describes the models contained in the four main categories.

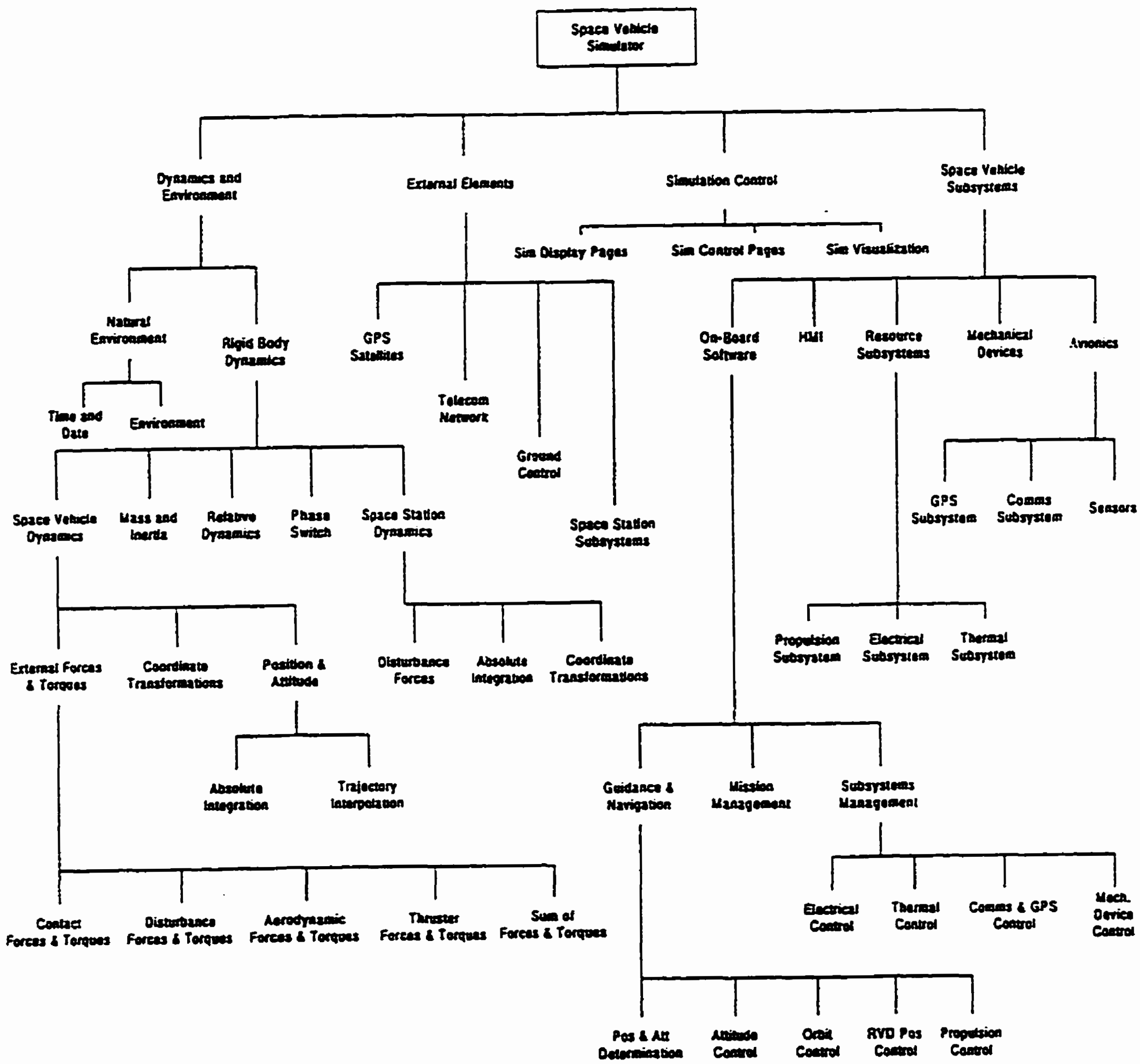


FIGURE 11. ISS/ATV Simulation High-Level Breakdown [2, p.13]. The space vehicle simulator is divided into four main categories which are then divided further into more detailed simulation models. The rendezvous algorithms were developed for the Guidance and Navigation category shown in the figure.

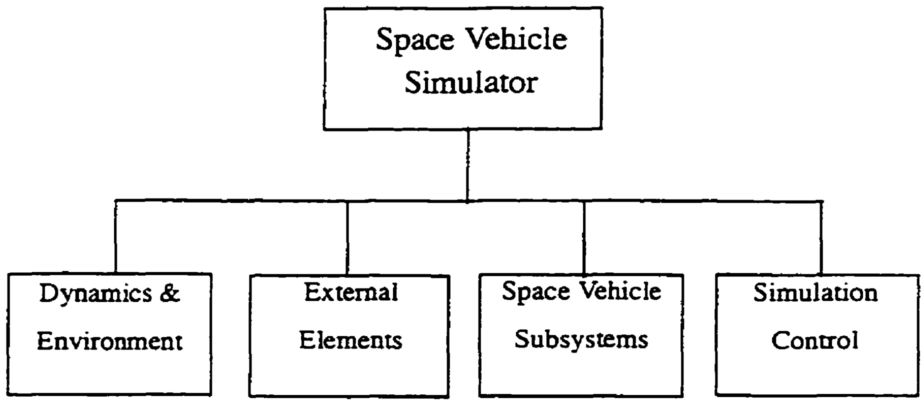


FIGURE 12. Hierarchy of the Space Vehicle Simulator [6, p.29]. The space vehicle simulator is divided into four main categories which are then divided further into more detailed simulation models.

3.4.1 Dynamics and environment models

The dynamics and environment models include part of the simulation that closes the loop between the spacecraft sensors and the actuators. This is the simulation of the environment that the ATV and the ISS experience. The following sample models will give the reader a general idea about the environment simulated [6].

<u>The Natural Environment</u>

- Time and Date - Set by the user.
- Atmospheric properties - Uses the U.S. standard atmosphere 1976 model.
- Gravitational acceleration - The Earth's gravitation and oblateness can be modeled.

- Magnetic field - The dipole model calculates the Earth's magnetic field vector at a given position.
- Solar flux - The mean solar flux integrated over all wavelengths at a given position varies with the day of the year.
- Natural object position - The Earth, sun, moon, and star positions relative to the space vehicle based on the time and date.

Rigid Body Dynamics

- Satellite orbit propagation - The communication satellites' orbits are propagated. These satellites are used as relay and link spacecraft.
- Space station orbit propagation - During the ATV injection, atmospheric and far rendezvous phases the space station's true anomaly is integrated using the Euler method. During the rendezvous and docked phases, the space station's orbit is propagated as part of the space vehicle dynamics described later.
- Space vehicle dynamics - The equations of motion for a single rigid body are integrated. The position and attitude equations are processed independently.
- External forces and torques - These could be due to air drag or solar pressure.
- Position and attitude computation - These involve solving the equations of motion (position and attitude) in an inertial frame. The fourth-order Adams-Moulton integration method with Runge-Kutta initialization is used. This computation is used during the rendezvous phase of the mission.
- Relative Integration - This is the same as above except the equations of motion are solved relative to a body in orbit, in this case the ISS.
- Phase switch - The flight phase of the ATV is determined by this model. The switch changes global variables which sends signals to the models indicating the flight phase.

3.4.2 External elements models

External elements are modeled as part of the simulation that is external to the ATV but is required by the ATV's subsystems to perform. These models include the following:

- The control center - The communications are relayed via an active communication channel. Errors can also be included.
- Relay and link satellites - Communications from satellites such as the TDRSS (US). DRS (European), and the LUCH (Russian) are modeled.
- Ground Stations - Communications from ground stations are modeled.
- Communication signals - This models the communication between two devices. Distances, orientation of the antennas, and Earth shadowing are accounted for.
- Space station - The communications of the space station are modeled. The docking mechanisms are also included in these models.
- GPS satellites - the GPS NAVSTAR constellation of satellites is modeled. The model includes the position of the GPS satellites visible for the appropriate signal transmission.

3.4.3 ATV subsystem models

Subsystem models simulate the function of the subsystems on-board the ATV. A sample of the subsystems modeled is as follows:

- ATV on-board software
- Mission management - This is the onboard software that drives the subsystems based on the mission phase of the ATV
- Guidance, navigation, and control - This models the onboard algorithms that determine the thrusts for the desired ATV trajectories. The following lower level models are part of this high level model:

1] Position and attitude determination

2] Attitude control

3] Orbit inclination control

4] Hohmann transfer control

5] Homing control

6] Closing control

7] Forced motion control

8] Aerodynamic control

- Subsystems management - This includes the onboard software that manages the propulsion, electrical power, thermal, communications, and mechanical devices subsystems.
- ATV High-Level Equipment - This includes the following models:

- Avionics subsystems - This includes the data management subsystem, the communications subsystem, the GPS receiver, and other ATV sensors.

- Mechanical devices - Aerodynamic control devices, parachute deployment, docking mechanism, antenna deployment, solar panel deployment, etc.

- Human-machine interface - This provides the user with graphical representation of the panels to simulate the controls from the ground station or the space station.

- ATV resources equipment - This includes the following subsystems:
  - propulsion subsystem
  - electrical power subsystem
  - thermal subsystem

3.4.4 Simulation control models

These models allow the user to monitor and control the simulation. The user can control the various elements of the simulation through the simulation control schematics without having to interfere with the element schematics themselves. The user can control the time acceleration factor that controls the speed of the simulation and can visualize the state of the simulation through three visualization packages. The following visualization packages were used for the ISS/ATV simulation:

- Orbit 2D - The ground track displayed on a mercator map
- Orbit 3D - Spacecraft orbit displayed on the spherical Earth with background of catalogued stars
- Animator - Space vehicle rendering during flight displaying its attitude, orbit, and mechanisms deployment

These visualization packages are independent packages that are interfaced with the ROSE™ simulation. The interface is made through specially designed ROSE™ objects.

3.5 Space simulation object libraries

The simulation schematics developed for the ISS/ATV simulation consisted of generic objects created for spacecraft simulation purposes. Figure 13 illustrates some of the object libraries created for the simulation.

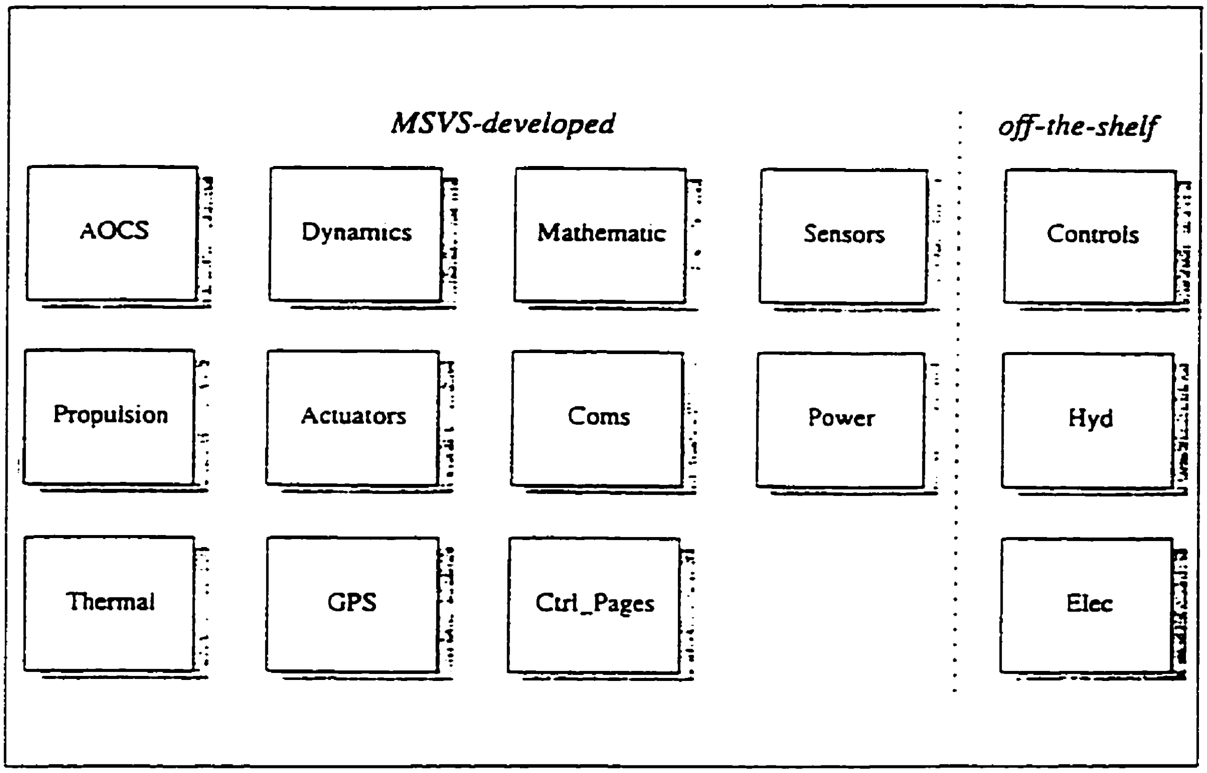


FIGURE 13. Object Libraries for Simulation of ISS/ATV Mission [6, p.40]. The simulation schematics developed for the ISS/ATV simulation consisted of generic objects created for spacecraft simulation purposes. These objects are placed in object libraries and are 'dragged' out of the libraries onto the schematics.

The object libraries above were created for generic objects that could be used as a template for any spacecraft simulation. The objects and libraries can be modified or new ones can be created to suit the needs of a specific spacecraft and/or mission. These new libraries and objects can further be re-used for other projects. The usefulness of the generic libraries is demonstrated through the creation of the ISS/ATV rendezvous simulation. Specifically, the dynamics and onboard software libraries contain the main objects created for the rendezvous algorithms. Objects from other libraries such as the mathematic library were also used. The objects were put together in the homing phase schematic, the closing phase schematic, and the corrections schematic. These schematics will be described in Chapter 6. The next chapter covers some of the fundamentals of orbital mechanics and rendezvous that were used to develop various objects.

Chapter 4

4.0 FUNDAMENTALS OF ORBITAL MECHANICS

The fundamentals of orbital mechanics are the foundation of rendezvous algorithms, orbit propagation, and orbit predictions. These are well known and may be found in many texts [11..14] such as Fundamentals of Astrodynamics [8], Modern Astrodynamics [12], and Methods of Orbit Determination [10]. This chapter will capture the fundamentals and refer to the texts that explain the concepts in more detail. This will provide the reader with an overview of the important ideas and equations.

4.1 Basic theory

Newtonian and Keplerian laws form the basis of orbital mechanics. Newton's laws of motion and universal gravitation describe the dynamics of orbiting spacecraft. Newton's laws of motion are stated, from Book 1 of the *Principia*, as [8]:

First Law

Every body continues in its state of rest or of uniform motion in a straight line unless it is compelled to change that state by forces impressed upon it.

Second Law

The rate of change of momentum is proportional to the force impressed and is in the same direction as that force.

Third Law

To every action there is always opposed an equal reaction.

Kepler's laws of orbits are also the basic building blocks upon which orbital mechanics is built and are stated as follows:

First Law

The orbit of each planet is an ellipse, with the sun at a focus.

Second Law

The line joining the planet to the sun sweeps out equal areas in equal times.

Third Law

The square of the period of a planet is proportional to the cube of its mean distance from the sun.

Newton's and Kepler's laws of motion and orbits allow for a description of how spacecraft will move in their orbits and how their trajectories can be controlled for any particular mission.

For a system that has n-bodies that exert gravitational forces on each other, the motion of the *i*th body in an inertial reference frame is governed by the following equation [7]:

$$m_i \frac{d^2 r_i}{dt^2} = - \sum_{j = 1, j \neq i}^{n} G \frac{m_i m_j}{r_{ij}^3} (r_i - r_j) \qquad \text{(EQ 1)}$$

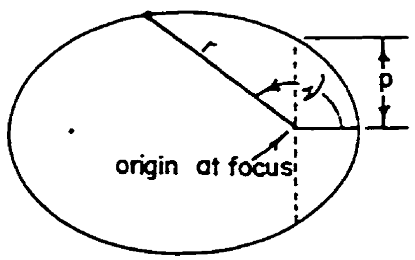
The constant of proportionality, G, is the universal gravitational constant; m_i is the mass of the *i*th body; r_i is the position of the ith body; and r_{ij} is the distance between the ith and the *j*th body, $r_{ij} = |r_i - r_j|$. (EQ 1) indicates that the motion of the ith body depends on the relative positions of all the other bodies in the system. Thus, a system of equations including all the bodies would have to be solved simultaneously in order to determine the motion of one of the bodies. A complete analytical solution for this problem is only possible for the two-body problem. This is sufficient in the case of a spacecraft orbiting the Earth. Using (EQ 1) and $\vec{\mathbf{r}} = \vec{\mathbf{r}}_m - \vec{\mathbf{r}}_M$, where m refers to the mass of the spacecraft and M refers to the mass of the Earth, the two-body problem can be stated as:

$$\frac{d^2}{dt^2}\vec{\mathbf{r}} + \frac{\mu}{r^3}\vec{\mathbf{r}} = 0 \qquad \text{(EQ 2)}$$

where $\mu = G(M + m)$. The two assumptions that are satisfied in the two-body problem are: Firstly the two bodies are spherically symmetric thereby becoming point masses, and secondly the gravitational forces are the only forces acting on this system. (EQ 2) describes the motion of a spacecraft with respect to the Earth. Taking the first integral of (EQ 2) and taking the vector product with $\vec{\mathbf{r}}$ yields the constant specific angular momentum, $\vec{\mathbf{h}}$. Since the specific angular momentum is constant, the motion is planar. Another equation can be developed by taking the vector product between the angular momentum vector, $\vec{\mathbf{h}}$, and (EQ 2) and integrating. This results in the trajectory equation which, in polar co-ordinates, is:

$$r = \frac{\frac{h^2}{\mu}}{1 + e\cos\nu} \qquad \text{(EQ 3)}$$

where e is the eccentricity which determines the type of conic that (EQ 3) represents and ν is the angle between the eccentricity vector, $\vec{e}$, and the radius vector, $\vec{r}$, called the true anomaly of the orbiting spacecraft. The value of $\frac{h^2}{\mu}$ is called the *semilatus rectum* or the parameter p. It is clear from (EQ 3), that r is at a minimum when $\nu = 0$, thus the eccentricity vector points to the periapsis, the closest point to the Earth in an orbit. Figure 14 describes the conic section determined by (EQ 3) [7].



$$r = \frac{p}{1 + e\cos\nu}$$

$e = 0$	Circle
$0 < e < 1$	Ellipse
$e = 1$	Parabola
$e > 1$	Hyperbola

FIGURE 14. General Equation of a Conic Section [8, p21]

There are six constants of integration for the two-body problem equation. The six constants can be stated as the components of $\vec{r}(t_o)$ and $\frac{d}{dt}\left(\vec{r}(t_o)\right)$, where t_o is the reference time. However, constants that geometrically represent an orbit and are related to $\vec{r}(t_o)$ and $\frac{d}{dt}\left(\vec{r}(t_o)\right)$ are used instead. These are called the classical orbital elements. Figure 15 describes these constants [7].

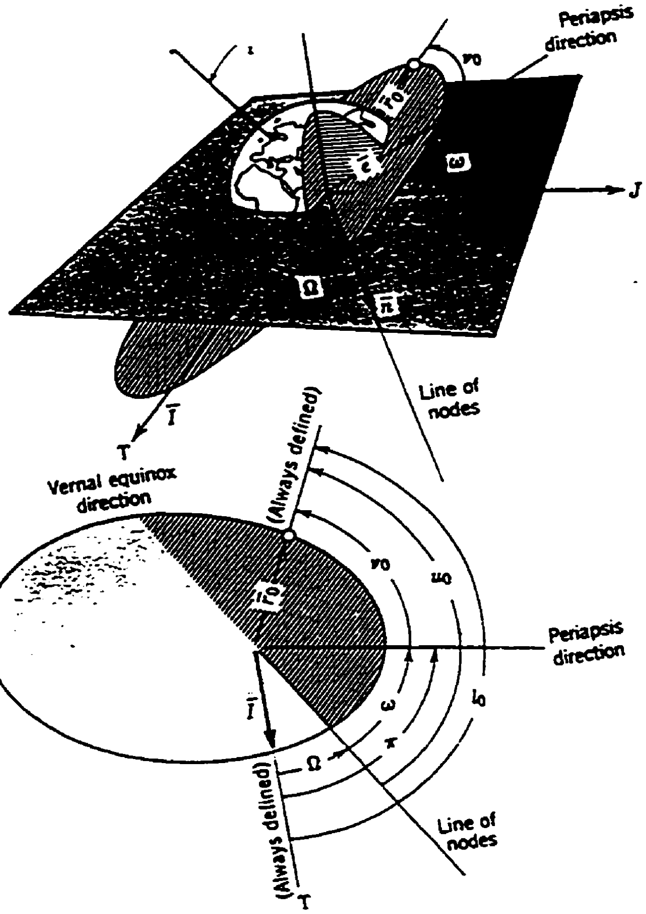


FIGURE 15. Graphical Representation of the Classical Orbital Elements [7, 7.9]. There are six constants of integration for the two-body problem equation. The constants that geometrically represent an orbit and are related to the position and velocity vectors are used instead. These are called the classical orbital elements.

The orbital elements are:

- i - inclination angle
- Ω - longitude or the right ascension of the ascending node
- ω - argument of the periapsis
- ν - true anomaly
- a - semi-major axis
- e - eccentricity

The orientation of the orbit can be described by i, Ω, and ω. The size and the shape of an orbit are fixed by the semi-major axis, a, and either the semilatus rectum, p, or the eccentricity, e. The position of the body in the orbit can be described by ν. Thus, there are six orbital elements that can locate a spacecraft in an orbit and allow for the prediction of its location at any time. The prediction of the orbit is described in Chapter 5 where the solution to the 'Kepler problem' is discussed. The orbital elements are related to the position and velocity vectors of a spacecraft. Thus, the orbital elements can be converted to the position and velocity vectors and vice-versa. The detailed derivation of these relations can be obtained in a number of references that deal with the fundamentals of orbital mechanics [7] [8].

Another consequence of the two-body equation is the specific mechanical energy of a spacecraft. The detailed derivation can be found in reference [8]. The following equation describes the specific kinetic energy and the specific potential energy of the satellite.

$$\xi = \frac{v^2}{2} - \frac{\mu}{r} \qquad \text{(EQ 4)}$$

The time rate of change of the above equation is zero, therefore ξ is a constant. The above equation shows that the total energy, ξ, of a spacecraft is constant along an orbit with a velocity, v, at a radius, r.

4.2 Rendezvous techniques

The problem of rendezvous involves one spacecraft achieving the same orbit as another. The spacecraft that is performing the rendezvous is usually called the chaser or the interceptor vehicle and the spacecraft being chased is called the target vehicle. The chaser vehicle has to arrive at the same time and place as the target vehicle or a location around the target vehicle for a successful rendezvous. There are many constraints to the trajectory required to achieve a rendezvous. The following will describe some of the general techniques for a chaser vehicle to rendezvous with the target vehicle.

If a minimum fuel transfer orbit is required then the Hohmann transfer can be used [9]. The constraint in this case is the propellant. The chaser has to be in the proper position with respect to the target in order to use a Hohmann transfer orbit successfully. This requires the chaser to wait in its orbit until the correct relative positions have been achieved. The orbit that places the chaser in the desired geometry for a rendezvous is called a phasing orbit. If the two orbits are coplanar, circular, and of different sizes, the phasing orbit is simply the chaser's initial orbit. Assuming the chaser is at a lower altitude, it can fire its thrusters to achieve the transfer orbit and rendezvous with the target at the apoapsis (Figure 16). The chaser can fire the second burn at the apoapsis to achieve the same orbit as the target vehicle. Figure 16 describes the geometry of the chaser and target to initiate a Hohmann transfer [9].

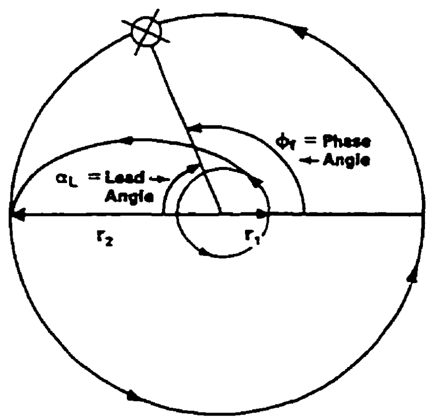


FIGURE 16. Geometry of Chaser and Target Vehicles for Rendezvous [9, p.150]. Assuming the chaser is at a lower altitude, it can fire its thrusters to achieve the transfer orbit and rendezvous with the target at the apoapsis.

The following is a derivation of the wait time required by the chaser vehicle to commence its rendezvous. The mean motion of spacecraft is given by the following equation:

$$n = \sqrt{\frac{\mu}{a^3}}$$

The correct geometry of the two vehicles is determined by the position of the chaser relative to the target such that if a Hohmann transfer were initiated, the chaser would meet the target 180 degrees into the transfer orbit. The lead angle, α_L, is the angular distance the target travels during the chaser's transfer orbit and is given by:

$$\alpha_L = (n_{target})\,TOF$$

Where, TOF is the time-of-flight of the Hohmann transfer and the final phase angle, ϕ_f, is the angle between the chaser and the target when the rendezvous transfer orbit is possible given by:

$$\phi_f = \pi - \alpha_L$$

The wait time, τ, is given by dividing the relative angular positions by the difference in the chaser and target mean motions.

$$\tau = \frac{\phi_i - \phi_f + 2\pi\,(k)}{n_{chaser} - n_{target}}$$

The constant, k, is the rendezvous opportunity integer. The first opportunity is when k=0. The total time for rendezvous would be the sum of the wait time plus half the period of the transfer orbit. Note that as the two orbits get closer the denominator approaches zero and thus the wait time approaches infinity.

If the chaser is already at the target orbit, it must enter another phasing orbit in order to rendezvous with the target vehicle [9]. This is depicted in Figure 17 as follows:

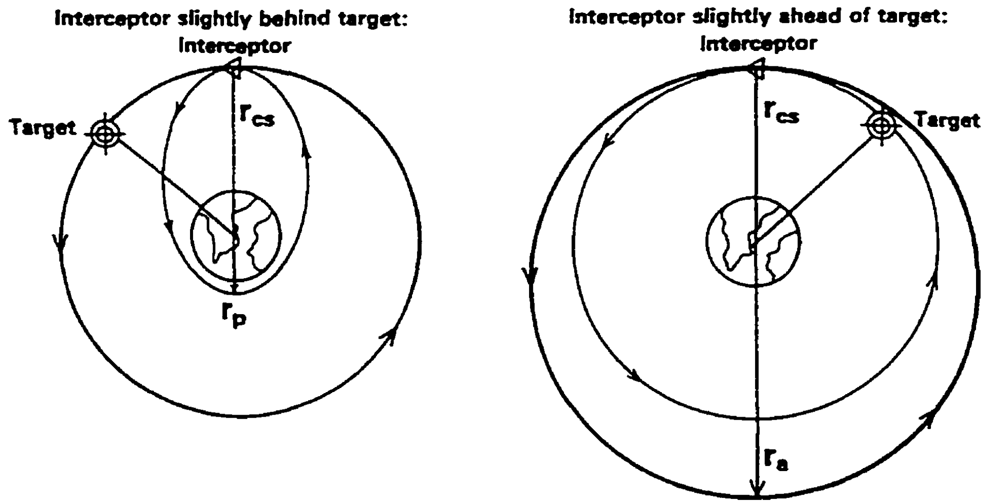


FIGURE 17. Phasing Orbits when Target and Chaser are on the Same Initial Orbit [9, p.151]. If the chaser is already at the target orbit, it must enter another phasing orbit in order to rendezvous with the target vehicle. The phasing orbit has a new semi-major axis that will have a longer or shorter period to allow for a rendezvous with the target vehicle. The chaser will return to the point where the phasing orbit starts.

The period, P, of an orbit is determined by its semi-major axis as the following equation shows:

$$P = 2\pi\sqrt{\frac{a^3}{\mu}}$$

Thus, the chaser can be put into a smaller orbit so that its period becomes smaller. The rendezvous will occur at the point the chaser initiates its phasing orbit. The period of the phasing orbit should be such that the target reaches the rendezvous point at the same time. Thus the period of the phasing orbit, which is equal to the time-of-flight for the target to get to the rendezvous point, determines the semi-major axis of the phasing orbit. The above figure depicts the cases when the chaser or interceptor is slightly behind and slightly ahead of the target vehicle.

The techniques described above involve Hohmann transfers that require minimum fuel for transfer orbits and rendezvous when the chaser and target are on the same orbit. The problem of rendezvous becomes even more complex when the orbits are not circular and non-coplanar. The trajectories for rendezvous have to be carefully analyzed when safety or collision avoidance becomes the main constraint. In the case of the ISS/ATV mission, circular coplanar orbits were considered, but safety, rather than fuel consumption was the main constraint. Thus the phasing orbits described above were not used. Instead, orbits that ensure the ATV will not collide with the ISS and which allow the ISS a reasonable amount of time to perform evasive manoeuvers, such as changing its altitude, were determined. This method will be covered in Chapter 5.

4.3 Guidance, navigation, and control

Navigation of a spacecraft involves determining the orbit of the spacecraft. This essentially means determining the position and velocity of the spacecraft or equivalently its orbital elements as a function of time. Guidance is referred to as determining the additional velocity or Delta-V that the spacecraft needs to change its orbit in order to get it to where it is required. Control is referred to as the execution of the Delta-V command by firing the thrusters onboard the spacecraft. Therefore, in a rendezvous mission, first a target orbit and the chaser's orbit is determined. The required change in the chaser's orbit is determined in order for it to reach its target orbit. Finally, the commanded Delta-V is executed by firing thrusters onboard the spacecraft. During the rendezvous trajectory, the chaser's position and velocity is continually monitored in case there is a perturbation from the desired trajectory. This perturbation is usually due to an inaccurate initial burn that sets the chaser on its intercept trajectory. This process continues as follows; the target and chaser orbits are determined (navigation), the required change in the chaser's orbit is calculated (guidance), and the burn for intercept is carried out (control) [9].

Figure 18 illustrates the Automatic Control Loop for the ATV's guidance, navigation, and control subsystem.

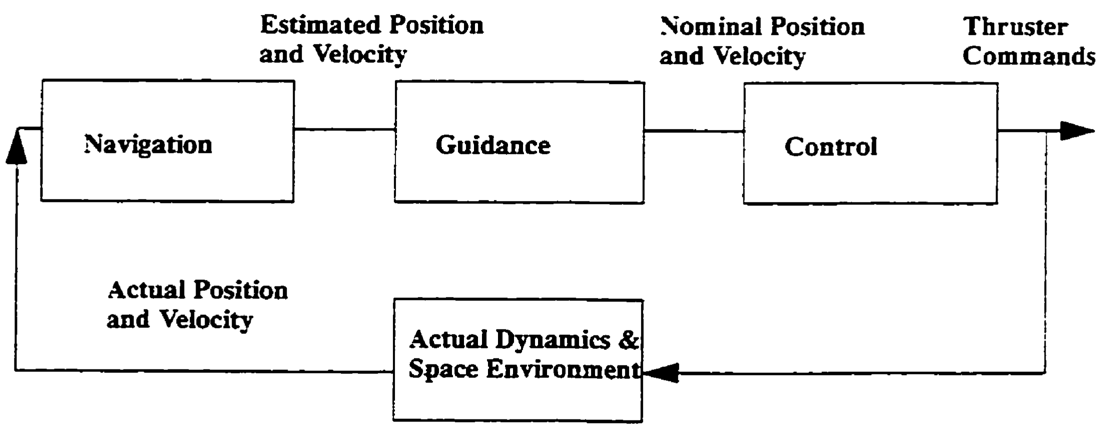


FIGURE 18. Guidance, Navigation, and Control Automatic Control Loop [3, p.282]. Navigation of a spacecraft involves determining a spacecraft's orbit. Guidance is referred to as determining the additional velocity or Delta-V that the spacecraft needs to change its orbit in order to get it to where it is required. Control is referred to as the execution of the Delta-V command by firing the thrusters onboard the spacecraft.

4.4 Orbit determination

The spacecraft's position and velocity as a function of time is called the ephemeris. The spacecraft's orbit can be predicted by two different methods of timing. The real-time determination of the spacecraft's orbit provides its current position. The second method is called definitive orbit determination and involves the estimation of the spacecraft ephemeris at an earlier time. The determined orbit can then be propagated by integrating the equations of motion. This will provide the spacecraft's orbit at a later time. Orbit propagation is critical in missions where the chaser vehicle has to intercept the target vehicle. Orbits can also be propagated backwards in time so that the prediction can be compared to actual observations and the error in the predictions can be reduced [9]. Traditionally, orbit determination has been a ground-station activity. Recent technological advances such as the Global Positioning System and more advanced onboard computers allow for autonomous control of spacecraft. The orbit of the spacecraft can be determined in real-time from onboard the spacecraft and autonomous control can be executed to place the spacecraft in a desired orbit. The spacecraft requires onboard processing to perform an autonomous rendezvous mission, namely the ATV with the ISS.

4.5 Orbit determination systems

The first factor to consider in orbit determination is the source and type of data used. This may be in the form of radar tracking data or GPS data obtained onboard the spacecraft. It should be understood that these observations will have limited accuracy which will affect the propagation results. The second consideration is the algorithms that are used to propagate the orbit of the spacecraft. This becomes even more complex when perturbations are added to the simple two-body motion problem. Two-body motion would include the spacecraft and the Earth and the integration of the equations of motion which were described earlier. Orbit perturbations are discussed in Chapter 7. The third important factor in orbit determination is the processor that is used to compute the propagated orbit. This thesis focuses on the second factor of orbit propagation, the rendezvous algorithms.

The most frequently used orbit propagation algorithm is called the Goddard Trajectory Determination System, GTDS. This system is used by NASA for all of their low-Earth orbit satellites. Another well known system is the Jet Propulsion Laboratory's Deep Space Network (DSN) used for interplanetary spacecraft [9].

Since definitive systems work on old data that can be corrected for errors and propagated forwards to meet real-time needs they are more accurate than real-time data that is propagated forward in time without any error corrections [9].

Orbit propagation, simply put, is the determination of the spacecraft's position and velocity some time in the future, knowing its present position and velocity. Johannes Kepler was one of the first persons to tackle this problem theoretically. Kepler developed an empirical expression that determined the time-of-flight of a planet from one point to another. This development is the basis for what is called the 'Kepler problem' of propagating the position and velocity of an orbiting object given the time-of-flight. The Kepler problem is discussed in detail in Chapter 6.

Orbit determination from two positions and a time-of-flight is a problem that has significance in the intercept and rendezvous mission. This problem was explored by Carl Friedrich Gauss. It is therefore referred to as the 'Gauss problem' and discussed in Chapter 6.

The solution to the Kepler and Gauss problems are the primary algorithms used for the rendezvous phases discussed in this thesis. The two algorithms are incorporated in the ROSE™ schematics to provide the Delta-V commands necessary for the ATV to rendezvous with the ISS. The position and velocity of the ATV and ISS are provided by GPS receivers onboard the spacecraft. The Delta-V commanded is sent to the propulsion subsystem. The Delta-V burns provide the necessary thrust to the ATV to place it in the required rendezvous trajectory. Thus, there is a navigation, guidance, and control function that allows the ATV to reach the space station. These algorithms were developed for the guidance simulation implemented in ROSE™. The guidance simulation schematics interface with both the navigation and control ROSE™ schematics that were developed by the author concurrently with CAE Electronics Ltd.

Two other software packages were used to propagate orbits (Appendix C), but these were not part of the ROSE™ simulation. They were used for verification and comparison purposes. The two software packages are the Satellite Tool Kit (STK) developed by Analytical Graphics and the Numerical Prediction of Orbital Elements (NPOE) downloaded from the internet. NPOE was developed by Mr. David Eagle of Science Software.

4.6 Chapter summary

This chapter has covered the following concepts:

- fundamental equation of motion related to two bodies, namely the Earth and a spacecraft, and the related assumptions.
- the orbital elements used to describe a spacecraft's position and velocity vectors and the geometry of an orbit.
- the techniques used in the rendezvous problem
- orbit determination and orbit determination systems

The next chapter will define the ISS/ATV simulation rendezvous phases and constraints. Chapter 6 will describe the rendezvous algorithms that were implemented in the simulation to complete the phases successfully and satisfy the constraints.

Chapter 5

5.0 ISS/ATV RENDEZVOUS PHASES

This chapter will describe the two rendezvous phases for which the algorithms were developed. These phases were simulated on ROSE™ schematics as part of the ISS/ATV simulation. The two phases are the homing phase, the closing phase, and a corrections schematic for the homing and closing phases. The corrections are Delta-V burns applied to the ATV if it strays off the planned trajectory by a pre-determined position magnitude. The constraints to the trajectories in both phases are also discussed in this chapter.

5.1 Constraints

There are many constraints that can be applied to the rendezvous trajectories of the ATV. The following constraints could apply:

- Propellant - constrains the amount of Delta-V available for rendezvous.
- Time - constrains the trajectory to the fastest one.
- Collision avoidance (safety) - This constraint applies the concept of 'safe trajectories'. It requires all burns to be performed in a manner such that the ATV's free-flight trajectory never intersects that of the space station.

Collision avoidance was deemed to be the most critical constraint. The ROSE™ simulation was based on a 'safe trajectory' rendezvous. This ensures that all of the ATV's orbits avoid collision with the ISS if there is a failure in the propulsion subsystem. The safe trajectories ensure the safety of the astronauts onboard the space station and will be described in detail in this chapter, where the ROSE™ schematics are described. The propellant and time constraints were also met using this rendezvous strategy but were not optimized. A brief description of the rendezvous phases which were considered is given in the next sections.

5.2 Homing phase

The homing phase is modeled on one ROSE™ schematic called 'rvd_homing'. The homing phase begins when the ATV enters the Space Station Communication Range (SSCR). This point is 12 km behind and 2 km below the space station COM. The space station and the ATV are in a coplanar orbit and are traveling in the same direction. The homing phase brings the ATV from this point to the space station orbit approximately 2500 m behind its COM. Figure 19 illustrates the trajectory of the ATV during this phase.

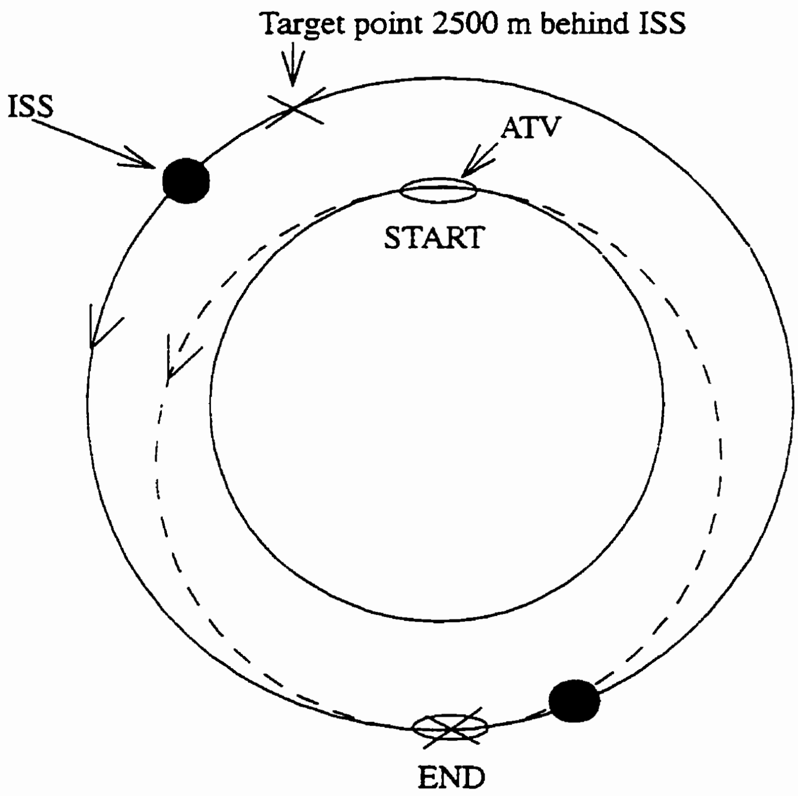


FIGURE 19. Homing phase of the ATV. The start of the homing phase is at a point 12 km behind and 2 km below the space station centre of mass. The homing phase brings the ATV from this point to the space station orbit approximately 2500 m behind its centre of mass.

5.3 Closing phase

The closing phase has been modeled on one ROSE™ schematic called rvd_closing. This phase begins at the space station orbit, 2500 m behind its COM. The closing phase ends at approximately 300 m behind the space station COM on the same orbit. The closing phase can be performed in more than one step. In the ROSE™ simulation, for example, the first step brings the ATV to 750 m behind the ISS COM and the second step brings it to approximately 300 m behind the ISS COM. The simulation can be modified to model one step or many steps in bringing the ATV closer to the ISS. The actual distance behind the space station is approximate because the ATV thrusters will not provide the exact Delta-V commanded. If the error is less than a specified value, no corrections to the trajectory will be made and the ATV will reach the desired position behind the space station within an acceptable range. If, however, the ATV's actual trajectory is different from the desired trajectory by a pre-determined position magnitude, there will be a correction burn to eliminate this error. Figure 20 illustrates a closing transfer from 2500 m behind the space station COM to 300 m behind its COM (one step transfer).

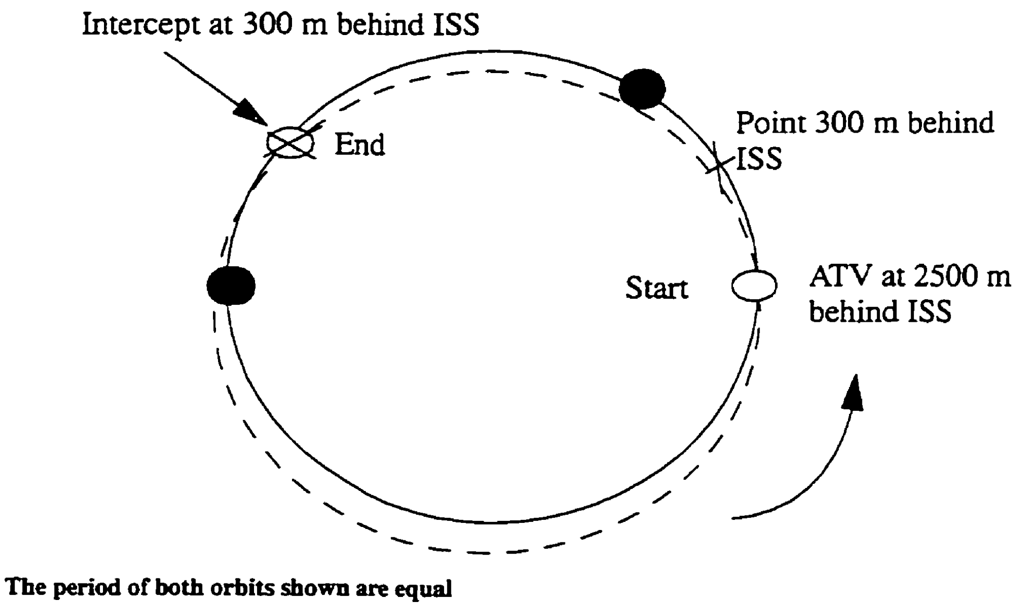


FIGURE 20. Closing Phase from 2500 m to 300 m behind Space Station (ISS) COM. The closing transfer brings the ATV from 2500 m behind the ISS to 300 m behind its centre of mass.

5.4 Corrections

There is a third ROSE™ schematic which models corrections to the ATV that bring it back to the desired trajectory when perturbations take it away from the predicted trajectory.

The predicted trajectory of the ATV, after its first transfer burn, is continually compared to the actual ATV trajectory. If the inertial position magnitude of the actual trajectory is greater than the predicted inertial position magnitude, a correction burn is determined and the second circularizing burn is re-calculated. This magnitude has been set to 100 m in the current ISS/ATV simulation. The corrections can be used during both the homing and the closing phases of the rendezvous mission. Figure 21 illustrates the correction burn.

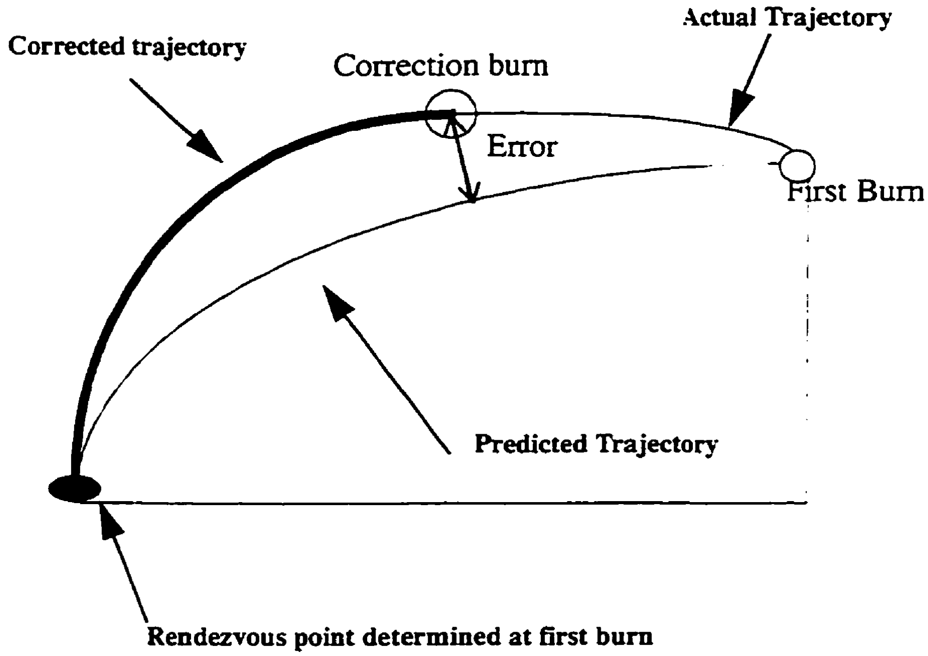


FIGURE 21. Correction Burn of the ATV to reach the Rendezvous Point. The correction burn brings the ATV back to the desired trajectory when perturbations take it away from the predicted trajectory. The predicted trajectory position is compared to the actual position. If the difference is greater than a user defined value, a correction burn is made.

A new Delta-V is required to ensure the ATV reaches the rendezvous point at the original time-of-flight. The new Delta-V is determined by using the remaining time-of-flight after the first burn, the rendezvous point position, and the current ATV position.

The chapter has described the two rendezvous phases and the corrections applied to the two phases.

Chapter 6

6.0 RENDEZVOUS ALGORITHMS

The two main problems solved for this phase of the mission are called 'the Kepler problem' and 'the Gauss problem'. The Kepler and Gauss problems are used in the schematics to predict the trajectories and calculate the Delta-V burns required by the ATV. The solution to these problems were required by the ATV from the homing phase to the completion of the closing phase. while using correction burns. The Kepler and Gauss algorithms were only part of the complete rendezvous phase algorithms and are described below. The ROSE™ schematics that were developed for the rendezvous phase consist of the complete algorithm for the rendezvous phase. The Kepler and Gauss problems were solved using the Universal Variable method as suggested by Bate [8]. The C code. test files, and the results of the Kepler and Gauss algorithms can be found in Appendix B.

6.1 Universal variable

The detailed development of the following equations is covered in reference [8]. The following is a brief overview of the Universal Variable, x.

The specific angular momentum, h, and the energy, ξ, of an orbit are given by the following equations:

$$h = r^2 \cdot \dot{\nu} = \sqrt{\mu \cdot p} \qquad \text{(EQ 5)}$$

$$\xi = \frac{1}{2} \cdot v^2 - \frac{\mu}{r} = -\frac{\mu}{2a} \qquad \text{(EQ 6)}$$

The velocity vector, $\vec{v}$, is resolved into its radial component, $\dot{r}$, and transverse component, $r\dot{\nu}$, to get:

$$\frac{1}{2}\dot{r}^2 + \frac{1}{2}(r\dot{\nu})^2 - \frac{\mu}{r} = -\frac{\mu}{2a} \qquad \text{(EQ 7)}$$

Solving for $\dot{r}$ and setting, $(r\dot{\nu})^2 = \frac{\mu p}{r^2}$ from (EQ 5) yields the following:

$$\dot{r}^2 = -\frac{\mu p}{r^2} + 2\frac{\mu}{r} - \frac{\mu}{a} \qquad \text{(EQ 8)}$$

(EQ 8) needs to be solved for r. The solution to this equation is found by introducing a Universal Variable, x, where

$$\dot{x} = \frac{\sqrt{\mu}}{r}. \qquad \text{(EQ 9)}$$

To develop an equation for r in terms of x, (EQ 8) is divided by the square of (EQ 9) to get:

$$\left(\frac{dr}{dx}\right)^2 = -p + 2r - \frac{r^2}{a}$$

Separating the variables yields:

$$dx = \frac{dr}{\sqrt{-p + 2r - \frac{r^2}{a}}}$$

The indefinite integral is (for $e \neq 1$):

$$x + c_o = \sqrt{a}\,\mathrm{asin}\left(\frac{\frac{r}{a} - 1}{\sqrt{1 - \frac{p}{a}}}\right)$$

where c_o is the constant of integration. Since $e = \sqrt{1 - \frac{p}{a}}$, the equation can be rearranged to give:

$$x + c_o = \sqrt{a}\,\mathrm{asin}\left(\frac{\frac{r}{a} - 1}{e}\right)$$

Thus, we can solve for r to get

$$r = a\left(1 + e\sin\left(\frac{x + c_o}{\sqrt{a}}\right)\right) \qquad \text{(EQ 10)}$$

Substituting (EQ 10) into the definition of the universal variable and integrating yields:

$$\sqrt{\mu}t = ax - ae\sqrt{a}\left(\cos\left(\frac{x + c_o}{\sqrt{a}}\right) - \cos\left(\frac{c_o}{\sqrt{a}}\right)\right) \qquad \text{(EQ 11)}$$

Where $t = 0$ at $x = 0$.

Now (EQ 10) and (EQ 11) can be used in a specific problem such as the prediction problem described in the next section. The constant of integration, c_o, is evaluated in the prediction problem as well. Note that the equations above give the relations for r and t in terms of the universal variable, x.

6.2 Kepler problem

Finding a spacecraft's position and velocity as a function of the time-of-flight is the basis for Kepler's problem. The problem can be stated as follows:

Given: $\vec{r}_o$, $\vec{v}_o$, and $t_o = 0$

Find: $\vec{r}$ and $\vec{v}$ at time t

Figure 22 illustrates Kepler's problem.

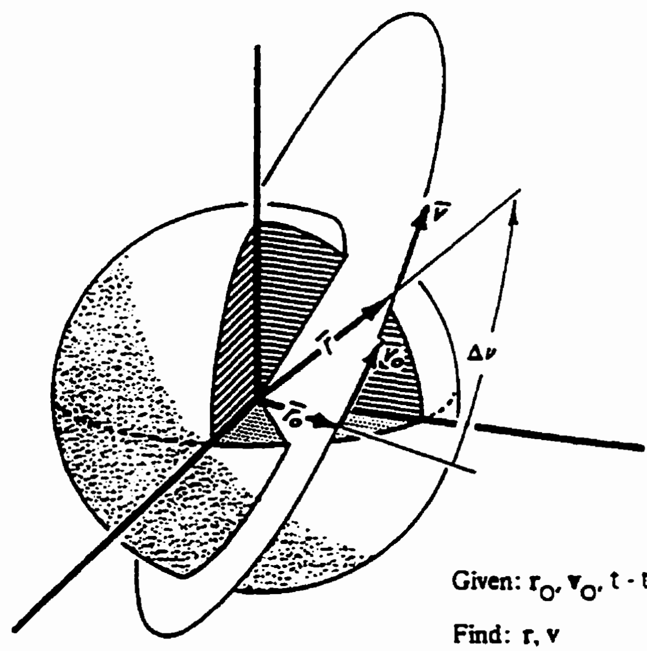


FIGURE 22. Kepler's Problem [8, p.195]. Finding a spacecraft's position and velocity after a given time-of-flight and its initial position and velocity is the basis of Kepler's problem.

The following section describes the problem of finding v when a,e, v_0. and the time-of-flight are given. The final position and velocity of a spacecraft are predicted knowing some initial position, velocity, and a time-of-flight. First the ROSE™ object that solves the Kepler problem is described. The input and output variables are described and the variables that may be tuned by the user are also described.

Description of the ROSE™ object

This ROSE™ object solves Kepler's prediction problem using the Universal Variable formulation. The inputs to the object are the current position and velocity vectors of the spacecraft in orbit around the Earth, and a time-of-flight. The object calculates, as the output, the final position and velocity vectors after the given time-of-flight. The vectors are in the geocentric inertial frame of reference. The Newton method is used as a convergence scheme. The maximum number of iterations for convergence and the tolerance for convergence are specified by the user.

If a singularity occurs or a non-real number is calculated, the flag SINGULAR is set to TRUE, and the output velocity and position vectors will have a value of zero. If the singularity condition no longer holds true, the correct position and velocity will be output, but the SINGULAR flag will remain TRUE indicating a singularity did occur at some point. The color of the object turns red on the schematic during run-time indicating that a singularity has occurred.

The detailed mathematical derivations used in this object can be found in reference [8].

Input Data

The external inputs required by this object are listed in the following table:

TABLE 1. The Input Parameters of the Kepler Object

Name	Description	Units
$\vec{r}_o$	Initial position vector	m
$\vec{v}_o$	Initial velocity vector	m/s
t	Time-of-flight	s

Output data

The external outputs generated by this component are listed in the following table:

TABLE 2. The Output Parameters of the Kepler Object

Name	Description	Units
$\vec{r}$	Predicted position vector	m
$\vec{v}$	Predicted velocity vector	m/s

Processing

The following is a summary of the main equations used to create the ROSE™ object.

From the given initial position vector, $\vec{r}_o$, and initial velocity vector, $\vec{v}_o$, the scalars $|\vec{r}_o|$, $|\vec{v}_o|$ are determined. The semi-major axis, a, is then found using the following equation:

$$\frac{1}{2}v_o^2 - \frac{\mu}{r_o} = -\frac{\mu}{2a}$$

Given the time-of-flight, t, the universal variable, x, is solved for using Newton's iteration scheme. The number of iterations performed is limited by a specified value, *imax*.

$$x_{n+1} = x_n + \frac{(t - t_n)}{\frac{dt}{dx}} \tag{EQ 12}$$

Where from (EQ 11) t_n is obtained as:

$$\sqrt{\mu}t_n = \frac{\vec{r}_o \cdot \vec{v}_o}{\sqrt{\mu}} x_n^2 C + \left(1 - \frac{r_o}{a}\right) x_n^3 S + r_o x_n$$

and from the definition of $\dot{x}$, (EQ 10), and (EQ 11), $\frac{dt}{dx}$ is:

$$\sqrt{\mu}\frac{dt}{dx} = x^2 C + \frac{\vec{r}_o \cdot \vec{v}_o}{\sqrt{\mu}} x(1 - zS) + r_o(1 - zC)$$

A first guess for x_n is:

$$x_n = \sqrt{\mu}\frac{(t_n)}{a}$$

Note that C and S are functions of another universal variable z, defined as:

$$z = \frac{x^2}{a}$$

If z is positive then

$$C = \frac{1 - \cos\sqrt{z}}{z}$$

$$S = \frac{\sqrt{z} - \sin\sqrt{z}}{\sqrt{z^3}}$$

If z is negative then

$$C = \frac{1 - \cosh\sqrt{-z}}{z}$$

$$S = \frac{\sinh(\sqrt{-z}) - \sqrt{-z}}{\sqrt{(-z)^3}}$$

If z is near zero then a truncated power series expansion of cos and sin can be used to evaluate C and S.

$$C = \frac{1}{2!} - \frac{z}{4!} + \frac{z^2}{6!}$$

$$S = \frac{1}{3!} - \frac{z}{5!} + \frac{z^2}{7!}$$

When the solution converges so that $(t\text{-}t_n)$ in (EQ 12) is below a specified *tolerance*, the value of the universal variable x, has been found. Therefore, z is also defined.

The final position vector, $\vec{\mathbf{r}}$, and the final velocity vector, $\vec{\mathbf{v}}$, are expressed as:

$$\vec{\mathbf{r}} = f\vec{\mathbf{r}}_o + g\vec{\mathbf{v}}_o$$

$$\vec{\mathbf{v}} = \dot{f}\vec{\mathbf{r}}_o + \dot{g}\vec{\mathbf{v}}_o$$

where,

$$f = 1 - \frac{x^2}{r_o}C$$

$$g = t - \frac{x^3}{\sqrt{\mu}} S$$

$$\dot{f} = \frac{\sqrt{\mu}}{r_o r} x (zS - 1)$$

$$\dot{g} = 1 - \frac{x^2}{r} C$$

Internal data

The internal data of the object that can be tuned by the user is listed in the following table:

TABLE 3. The Internal Data of the Kepler Object Specified by the User

Name	Description	Units
imax	Maximum number of iterations for Newton's method	-
tolerance	Tolerance for convergence of the time-of-flight	-

The number of iterations have to be limited to maintain a real-time simulation. Tests were made to determine a value for *imax* and *tolerance* that would yield an accurate result while maintaining a real-time simulation.

Constants

The internal constants needed by this component are listed in the following table:

TABLE 4. The Constants used in the Kepler Object

Symbol	Description	Units
μ	Earth's gravitational constant	m^3/s^2

6.3 Gauss problem

The solution of the Gauss problem yields the velocity vectors of a spacecraft at the given initial and final positions, when the angle between the two position vectors and the time-of-flight is known. The problem can be stated as follows:

Given: $\vec{\mathbf{r}}_1$, $\vec{\mathbf{r}}_2$, the time-of-flight and the angle between the position vectors

Find: $\vec{\mathbf{v}}_1$ and $\vec{\mathbf{v}}_2$

The angle between the two position vectors indicates which way the spacecraft is travelling (Figure 23). The angle is determined by taking the dot product of the target's orbit normal with the cross product of the chaser's position and velocity vector. The sign of the result indicates whether the target and chaser are traveling in the same direction or not. There is only one orbit that will pass through two position vectors in a given direction for a given time-of-flight. The figure below describes the orbit through two position vectors, in both possible directions but one time-of-flight.

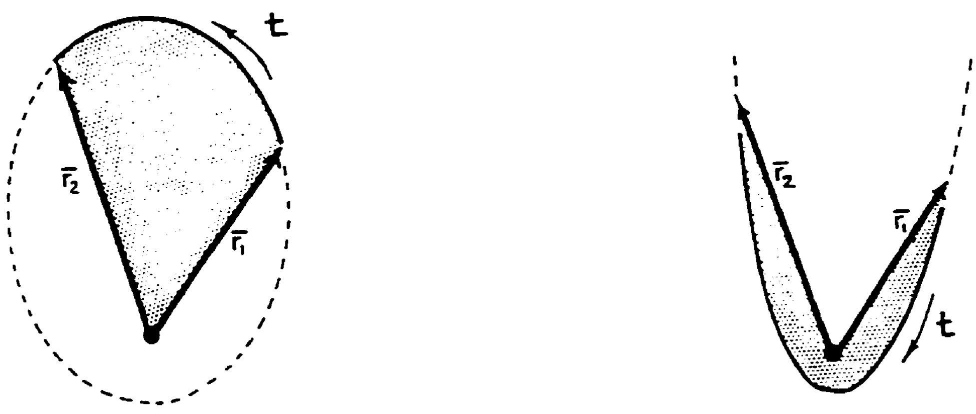


FIGURE 23. The Gauss problem Illustrating Two Orbits with the same Time-of-flight [8, p.229]. The solution of the Gauss problem yields the velocity vectors of a spacecraft at the given initial and final positions, when the angle between the two position vectors and the time-of-flight is known.

Description of the ROSE™ object

The ROSE™ object computes the initial and final velocity vectors of an orbiting body given its initial and final position vectors, the time-of-flight, and the angle between the two vectors.

Input data

The external inputs required by this object are listed in the following table:

TABLE 5. Input Parameters of the Gauss Object

Name	Description	Units
$\vec{r}_1$	Initial position of body in geocentric inertial frame	m
$\vec{r}_2$	Final position of body in geocentric inertial frame	m
t	Time-of-flight between the two position vectors	s
$\Delta\nu$	Angle between $\vec{r}_1$ and $\vec{r}_2$	rad

Output data

The external outputs generated by this object are listed in the following table:

TABLE 6. Output Parameters of the Gauss Problem

Name	Description	Units
$\vec{v}_1$	Velocity at initial position in geocentric inertial frame	m/s
$\vec{v}_2$	Velocity at final position in geocentric inertial frame	m/s

Processing

The detailed derivation of the Gauss problem can be found in reference [8]. The following is a summary of the main equations used to create the ROSE™ object.

Given two position vectors in the geocentric inertial frame, $\vec{r}_1$ and $\vec{r}_2$, and the angle between the two vectors, Δv, the magnitudes, $|\vec{r}_1|$ and $|\vec{r}_2|$, are calculated and the constant A is defined as:

$$A = \pm\frac{\sqrt{|\vec{r}_1||\vec{r}_2|}\sin\Delta v}{\sqrt{1-\cos\Delta v}}$$

If Δv is less than π this is the 'short-way' trajectory and A is taken as positive. If Δv is greater than π then the 'long-way' trajectory is considered and A is taken as negative. Note: if $\Delta v = \pi$, the two vectors are colinear and the plane of the orbit can not be defined. A unique solution is not possible in this case.

At this point a trial value is chosen for the universal variable z, where $z = \Delta E^2$. E is the eccentric anomaly. The initial trial value chosen depends on the value of the time-of-flight. The plot of time-of-flight vs. z can be used to determine the initial value for fastest convergence. The C and S functions are then evaluated for the trial value for z.

Note that C and S are functions of the universal variable z, where

If z is positive then

$$C = \frac{1 - \cos\sqrt{z}}{z}$$

$$S = \frac{\sqrt{z} - \sin\sqrt{z}}{\sqrt{z^3}}$$

If z is negative then

$$C = \frac{1 - \cosh\sqrt{-z}}{z}$$

$$S = \frac{\sinh(\sqrt{-z}) - \sqrt{-z}}{\sqrt{(-z)^3}}$$

If z is near zero then the truncated power series expansion of cos and sin is used to evaluate C and S.

$$C = \frac{1}{2!} - \frac{z}{4!} + \frac{z^2}{6!}$$

$$S = \frac{1}{3!} - \frac{z}{5!} + \frac{z^2}{7!}$$

Next the auxiliary variable y is evaluated as

$$y = r_1 + r_2 - A\frac{(1 - zS)}{\sqrt{C}}$$

Then the universal variable x is calculated as

$$x = \sqrt{\frac{y}{C}}$$

Next the trial value of the time-of-flight is calculated using the following equation:

$$\sqrt{\mu}t_n = x^3 S + A\sqrt{y}$$

The time-of-flight is compared with the actual time-of-flight and a Newton's iteration scheme, described below, is used to adjust the value of z until the trial value of time-of-flight, t_n, and the actual time-of-flight, t, converge to within a given *tolerance*. The following derivative of the time-of-flight with respect to z, is required for the iteration scheme.

$$\sqrt{\mu}\frac{dt}{dz} = x^3\left(S' - \frac{3SC'}{2C}\right) + \frac{A}{8}\left(\frac{3S\sqrt{y}}{C} + \frac{A}{x}\right)$$

Where,

$$C' = \frac{dC}{dz} = \frac{1}{2z}(1 - zS - 2C)$$

$$S' = \frac{dS}{dz} = \frac{1}{2z}(C - 3S)$$

However, if z is nearly zero (near-parabolic orbit) then the power series expansion of the C' and S' functions are used.

$$C' = \frac{1}{4!} + \frac{2z}{6!} - \frac{3z^2}{8!} + \frac{4z^3}{10!}$$

$$S' = \frac{1}{5!} + \frac{2z}{7!} - \frac{3z^2}{9!} + \frac{4z^3}{11!}$$

The following Newton's iteration method is used to adjust the value of z. A maximum number of iterations is specified by *imax*.

$$z_{n+1} = z_n + \frac{(t - t_n)}{\frac{dt}{dz}}$$

The iterations are carried out until *imax* is reached or $t - t_n$ becomes negligible (i.e. meets the specified *tolerance*).

Once the universal variable z has been evaluated, the f, g, and $\dot{g}$ functions, defined below, can be used to evaluate $\vec{v}_1$ and $\vec{v}_2$. The values of y and A are calculated using the value of z determined.

$$f = 1 - \frac{y}{r_1}$$

$$g = A\sqrt{\frac{y}{\mu}}$$

$$\dot{g} = 1 - \frac{y}{r_2}$$

Thus the two velocity vectors can be determined as follows:

$$\vec{v}_1 = \frac{\vec{r}_2 - f\vec{r}_1}{g}$$

$$\vec{v}_2 = \frac{\dot{g}\vec{r}_2 - \vec{r}_1}{g}$$

Internal data

The internal computation data needed by this object are listed in the following table:

TABLE 7. User Specified Internal Data of the Gauss Object

Name	Description	Units
imax	Maximum number of iterations for Newton's method	-
tolerance	Tolerance for convergence of the time-of-flight	-

The number of iterations have to be limited to maintain a real-time simulation. Tests were made to determine a value for imax and tolerance that would yield an accurate result while maintaining a real-time simulation.

Constants

The internal constants needed by this object are listed in the following table:

TABLE 8. The Constants used in the Gauss Problem

Name	Description	Units
μ	Earth's gravitational constant	m^3/s^2

6.4 ROSE™ homing phase schematic

This section describes the ROSE™ schematic that calculates the Delta-V burns required by the ATV during the homing phase of the rendezvous mission. The Kepler and Gauss objects described above were used as part of the overall homing algorithm. The ROSE™ schematics can be seen in Appendix B. The ROSE™ schematic is described using a flow chart which indicates the inputs, the processing, and the outputs. The name given to the ROSE™ schematic is 'rvd_homing'.

Inputs to schematic

- Control Flag - Mission management controls when the more complex objects are run so that unnecessary computer processing and memory is not used.
- ATV position and velocity vectors in geocentric inertial frame of reference
- ISS position and velocity vectors in geocentric inertial frame of reference

Outputs of schematic

- Delta-V for the first burn - places ATV in the transfer orbit
- Delta-V for the second burn - places ATV in the space station orbit
- Time-of-flight between the two burns

A certain period of time is required for the ATV to orient itself before performing any Delta-V burns. This is because the thrusters are fixed on the ATV. During this time, the ATV's position and velocity will have changed and so the calculated burns will have an error associated with them. To overcome this problem, the initial position and velocity vectors of the ATV and the ISS are propagated four minutes into the future. This allows the ATV to re-orient itself within the four minutes and the actual burns are applied when four minutes have passed. Due to the propulsion system not being able to provide the exact Delta-V required, errors will still remain in the actual orbit of the ATV. These errors are considered as perturbations to the ATV.

The time-of-flight used is that of a Hohmann transfer orbit. This would provide the most fuel efficient transfer orbit and would also be a safe orbit since the transfer orbit and the ISS orbit have approximately the same period. The apogee of the transfer orbit is at the same altitude as the space station. Thus, in the case of a complete loss of ATV control after the first burn, the ISS will have sufficient time to increase its altitude and avoid a collision with the ATV. Therefore, this trajectory is considered both safe and fuel efficient.

Processing

Some major elements used in this algorithm are described below.

The processing that occurs on the ROSE™ schematic is associated with the rendezvous algorithms that would be used in the onboard software. Figure 24. generally describes the algorithm used on the homing schematic (Appendix B).

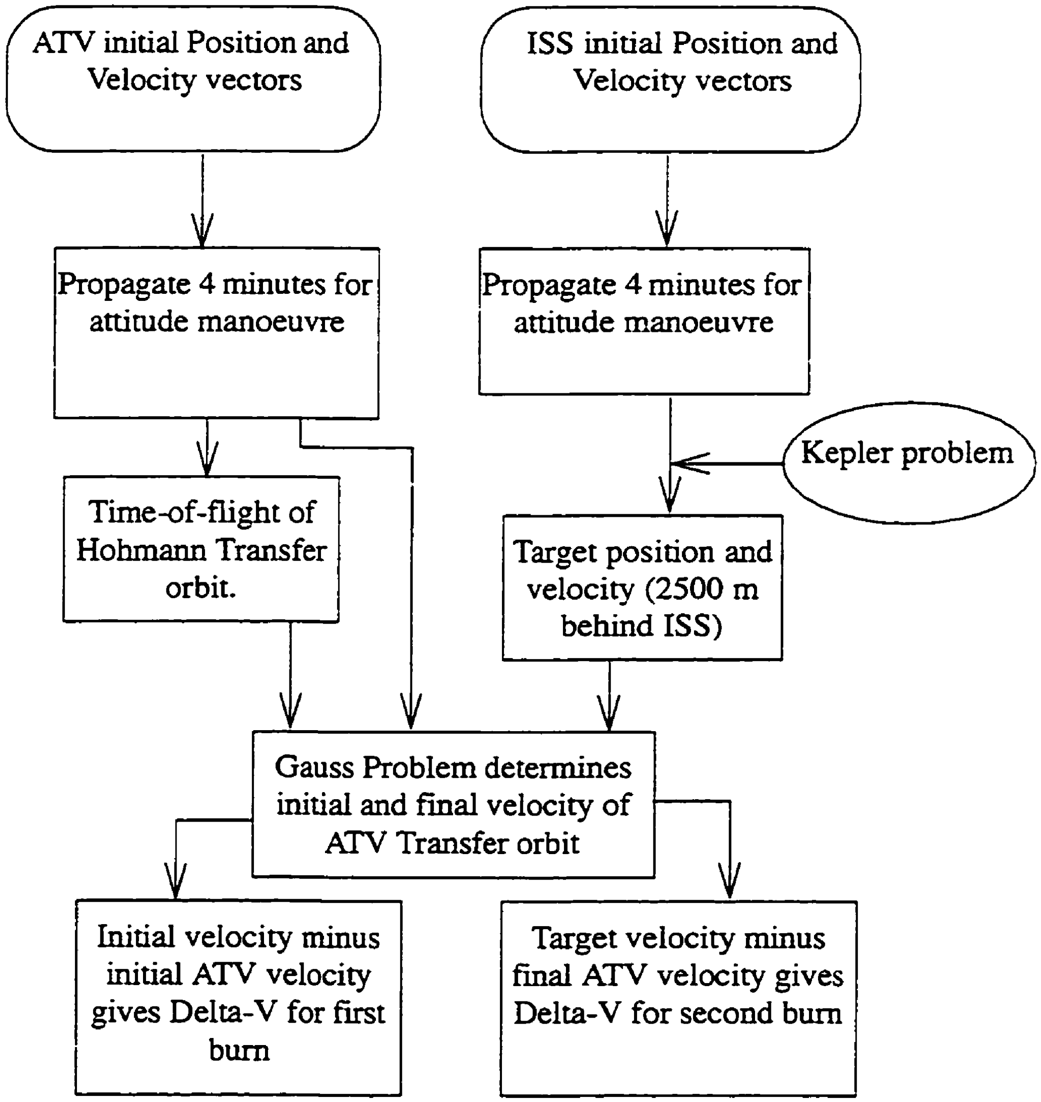


FIGURE 24. Flowchart Illustrating the ROSE™ Homing Phase Schematic. The flowchart describes the logic of the homing schematic. The ATV and ISS position and velocity vectors are inputs to the schematic and the Delta-V burns required to initiate and circularize the transfer orbit are the outputs.

Some of the more detailed calculations are not shown on the flow chart above to avoid clutter. The important calculations used in the homing schematic are described below.

Calculation of target point behind the ISS

The target point for the ATV has to be calculated before the Delta-V burns are determined. The rendezvous point is 2500 m, arc-length, behind the space station. The Kepler object is used to determine the position and velocity vectors of the target point. The ISS is simply propagated to the point that is 2500 m, arc-length, behind the ISS COM. Since the initial position and velocity of the ISS is known, the time-of-flight is required by the Kepler object to calculate the position and velocity of the target point. This is determined as follows:

$$\phi = \frac{\text{ArcLength}}{|r_{ISS}|}$$

$$t = \left(\frac{2\pi}{2\pi - \phi}\right) \cdot \text{Period}$$

Where,

ϕ - The angle between the target point and the ISS measured from the Earth's centre

ArcLength - The arc-length of 2500 m used as the target point behind the ISS on the same orbit

Period - The period of the ISS orbit

r_{ISS} - Magnitude of the ISS geocentric inertial position vector

t - Time-of-flight

Knowing the time-of-flight, the position and velocity of the target point is obtained. This is used to calculate the ATV's rendezvous burns.

6.5 ROSE™ closing phase schematic

The closing phase of the rendezvous brings the ATV from 2500 m behind the ISS COM to approximately 300 m behind its COM. This manoeuvre can be performed in shorter 'jumps'. The current simulation 'jumps' the ATV from 2500 m to 750 m and then to 300 m behind the ISS COM. All target orbits are the same as the ISS orbit. The actual position achieved by the ATV is not exactly the desired target position due to the perturbations in the Delta-V firings. However, the ATV is brought to a position that is within a certain tolerance around the desired position. If the ATV is perturbed outside a specified position tolerance during a transfer trajectory, a correction burn is made. The correction scheme is modeled on the ROSE™ corrections schematic described in the next section. Figure 25 is a flow chart describing the algorithm of the closing phase.

Inputs to schematic

- Control Flag - Mission management controls when the more complex objects are run so that unnecessary computer processing and memory is not used.
- ATV position and velocity vectors in geocentric inertial frame of reference
- ISS position and velocity vectors in geocentric inertial frame of reference
- Arc-length behind ISS for ATV rendezvous (i.e. 750 m and 300 m)

Outputs of schematic

- Two Delta-V burns to initiate and circularize the ATV transfer orbit
- Time-of-flight between the two burns
- Output control flag indicating that the solution is correct

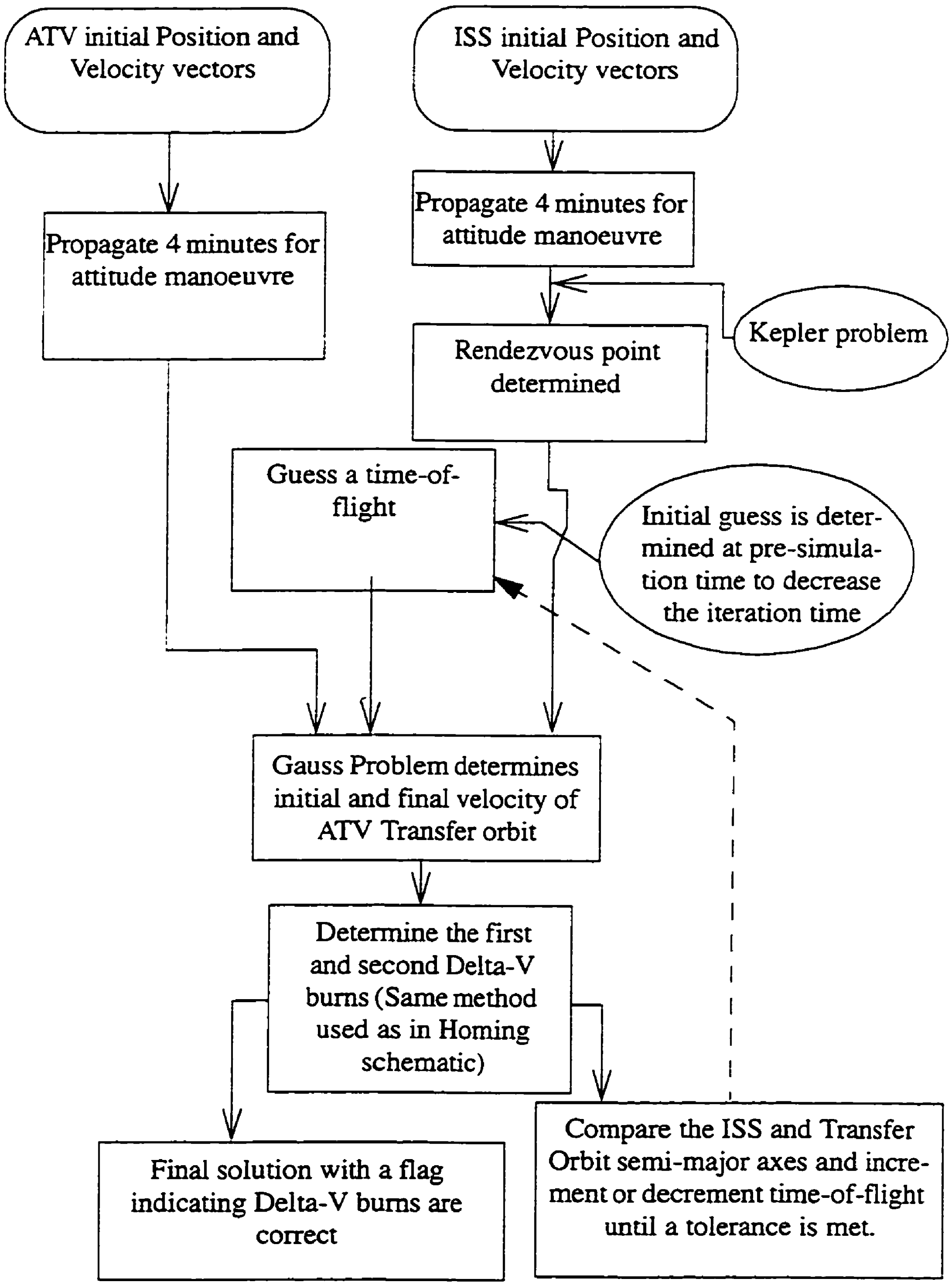


FIGURE 25. Flowchart Illustrating the ROSE™ Closing Phase Schematic. The flowchart describes the logic of the closing schematic. The ATV and ISS position and velocity vectors are the inputs to the schematic and the Delta-V burns required to initiate and circularize the transfer orbits are the outputs.

Processing

Some major aspects of this algorithm are described below.

The rendezvous point behind the space station is an input to this schematic in terms of the arc-length behind the ISS COM. The rendezvous point's position and velocity vectors are calculated in the same way as those described in the homing phase. The difference in this schematic is primarily the period of the transfer orbit calculated. To keep the ATV trajectories safe, the period of the transfer orbit and the space station are kept almost the same. This is done by comparing the semi-major axes of both the ISS orbit and the ATV transfer orbit. This will ensure that should the ATV lose its ability to fire the thrusters after the first burn it will not be in an orbit that will allow it to collide with ISS. Since the periods are kept the same, should the ATV not circularize at the intersection with its rendezvous point, it will return to the same initial position after one orbit as will the ISS. This will allow the ISS to obtain a higher orbit and safely remain away from the ATV. This is the 'safe-trajectory' condition that was employed in the ISS/ATV simulation.

The safe-trajectory strategy is implemented by using an initial guessed time-of-flight and iterating the solution until the difference between the space station semi-major axis and the transfer orbit semi-major axis is within a specified tolerance. The time-of-flight is then increased or decreased to meet this condition. The comparison is made to ensure that the difference is decreasing. If the difference is increasing and the time-of-flight is being increased then the time is decreased so that the difference decreases. If the difference is increasing while the time is being decreased then the time-of-flight is simply decreased so that the difference decreases. The initial time-of-flight used can be determined at pre-simulation time to ensure minimum iterations and a solution within the four minutes allotted for the attitude control manoeuvre. When the correct time-of-flight for rendezvous has been determined, the Delta-V burns calculated are output from the schematic with a control flag indicating that the solution is satisfactory.

6.6 ROSE™ corrections schematic

Corrections to both the homing phase and the closing phase are required due to perturbations to the desired or predicted transfer orbits (Chapter 7). Since the ATV thrusters will not provide the exact Delta-V required for rendezvous, the actual transfer orbit trajectory will be slightly different from the required trajectory. The criteria used is the magnitude of the actual position compared to the desired or predicted position in the transfer orbit. If the difference becomes greater than a specified value, a correction burn will be calculated. The new first burn is calculated so that the ATV will reach the rendezvous point at the same time as originally predicted. The time-of-flight used is the remaining time of the predicted transfer orbit. The second burn is calculated to circularize the ATV's orbit to match the ISS orbit.

The inputs and outputs of the schematic are described below. The general processing of the schematic is described using a flowchart (Figure 26).

Inputs

- ATV position and velocity vectors
- Control flag that indicates the first burn in homing or closing phase has been made
- Elapsed time since first burn
- Position and velocity vectors at first burn
- Position and velocity of the rendezvous point

Outputs

- First Delta-V for correction
- Final Delta-V to circularize the orbit

- Control flag to indicate burns are required (i.e. position comparison between actual and predicted trajectory is outside tolerance)
- Time before first burn (allows ATV to re-orient itself for the burn)

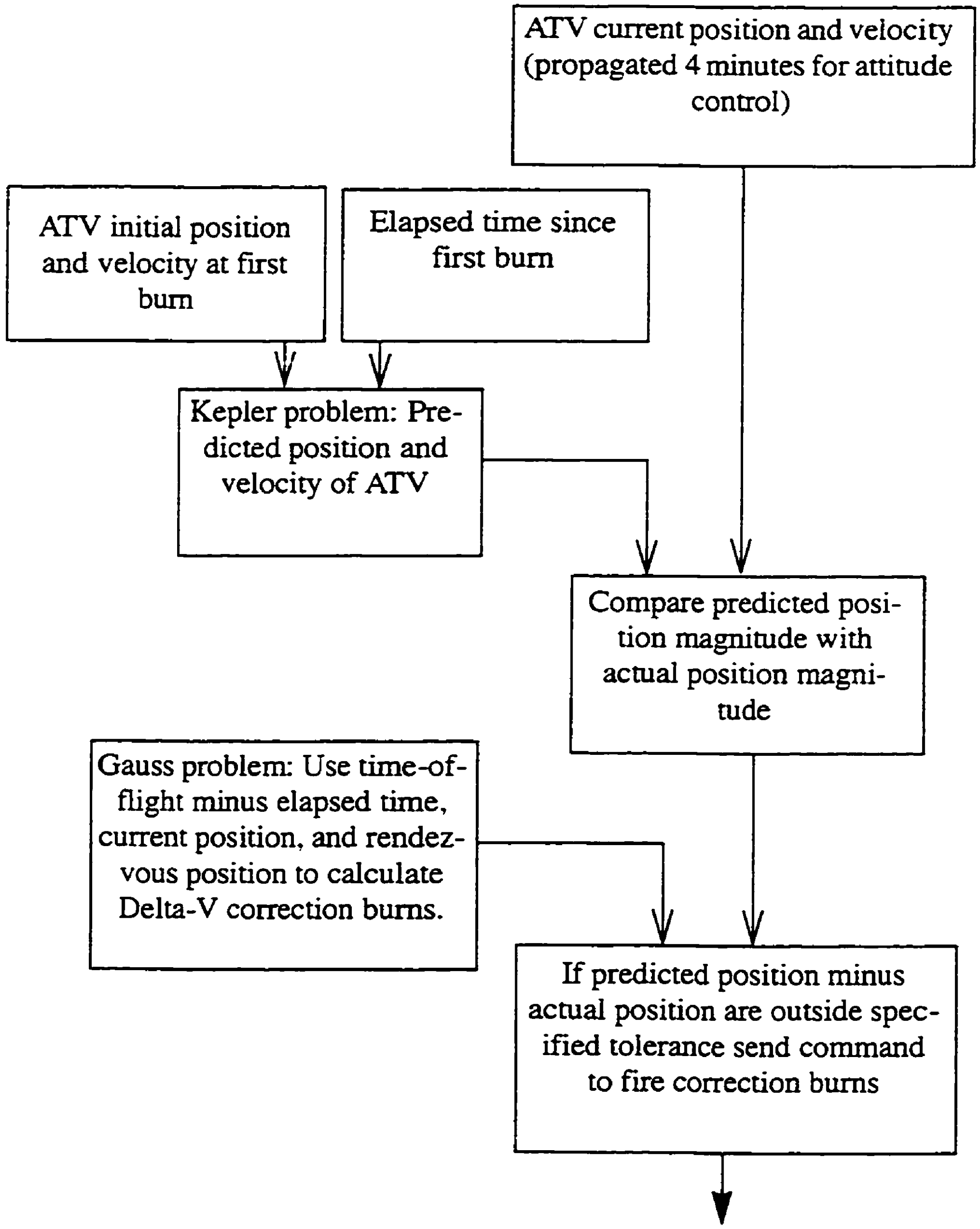


FIGURE 26. Flowchart Illustrating the ROSE™ Corrections Schematic.

6.7 Chapter Summary

The following ideas have been covered in this chapter:

- The algorithms that solve the Kepler and Gauss problems using the universal variable were described
- The logic of the homing and closing phases, which provide the Delta-V required, were covered using flowcharts
- The algorithm of the corrections schematic, which determines the correction burns required, was also described
- The use of the Kepler and Gauss algorithms in the homing and closing phases and the corrections schematic was covered

The results of the relative position of the ATV during the homing and closing phases can be found in Appendix A.

The next chapter will cover perturbations and their affects on this rendezvous simulation and the actual mission.

Chapter 7

7.0 PERTURBATIONS

The trajectories discussed above involve two-body motion only. The trajectories do not include the perturbations to the ATV or the ISS. In reality, there are accelerations other than those governed by the assumptions of the two-body problem. The Earth is not spherically symmetrical and forces other than gravity are involved in the motion of a spacecraft. This means that the ISS and ATV will not take the path that two-body motion predicts. There are many different kinds of perturbations which affect the orbit in various ways. The perturbations that concerns this thesis affect the rendezvous phases only. Therefore, trajectories that are in the order of half an orbit are investigated. The perturbations that affect spacecraft motion are described in the section below. The perturbations that will affect the rendezvous mission are discussed and the methods for predicting the orbits with these perturbations are also covered.

7.1 Perturbations that affect spacecraft orbits

Two major perturbation types affect a spacecraft orbit. Those which cause a secular variation in the Keplerian orbital elements and those which cause periodic variations in the orbital elements. The periodic variations are further broken down into short-term variations, which occur in less than one orbit, and long-term variations that occur in more than one orbit [9]. Since all transfer orbits will be occurring in less than one orbit, this thesis concerns short-term periodic variations in the orbital elements. Figure 27 illustrates the perturbations to the orbital elements.

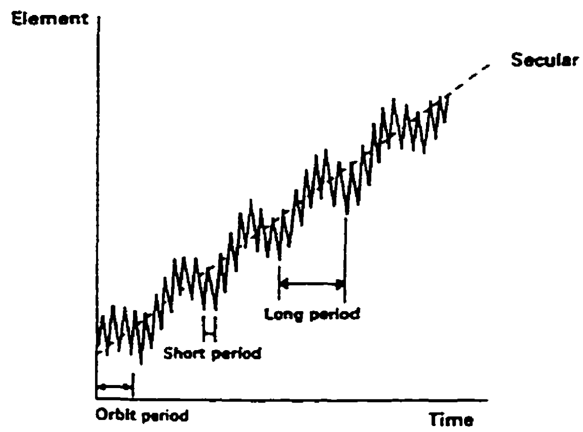


FIGURE 27. Perturbations on an Orbital Element [9, p.139]. Two major perturbation types affect a spacecraft orbit. Those which cause a secular variation in the Keplerian orbital elements and those which cause periodic variations in the orbital elements. The periodic variations are further broken down into short-term variations, which occur in less than one orbit, and long-term variations that occur in more than one orbit.

The following major perturbations affect a spacecraft's nominal orbit:

- Third-body perturbations - These are gravitational accelerations due to the Sun and the Moon.
- Perturbations due to a non-spherical Earth - The two-body equations of motion assume a spherical Earth with a homogeneous mass. In reality this is not true. The most dominant perturbation is caused by the oblateness of the Earth (The Earth bulges at the equator and is flat at the poles).
- Atmospheric drag - The ISS is in low-Earth orbit where there is still a slight atmosphere that will oppose the motion of any spacecraft. The drag causes the orbit to decay by removing its energy.

- Solar radiation - Solar radiation causes periodic variations in all the orbital elements.

The above are some of the natural perturbations that will affect the trajectory of a spacecraft. The other type of perturbations include an erroneous thruster firing which results in a Delta-V burn that is slightly different from the required burn. This type of unnatural perturbation is usually the largest type affecting a spacecraft since the burns are usually large over a short period of time. The two-body problem with perturbations becomes [8]:

$$\frac{d^2}{dt^2}\vec{\mathbf{r}} + \frac{\mu}{r^3}\vec{\mathbf{r}} = a_p$$

where a_p is the acceleration due to perturbations.

This thesis investigates the natural perturbations to the ATV that affect its rendezvous mission the most, namely the oblate Earth perturbation. The final ISS/ATV simulation does not include the effects of any perturbations. This discussion of perturbations illustrates the effects they would have on a real mission. The ATV's actual rendezvous algorithms do not take into account natural perturbations. The actual scenario will be discussed at the end of this chapter.

7.1.1 Method of perturbations

When perturbations are applied to the two-body motion other techniques are required to predict the orbit of a spacecraft. These techniques fall into two main categories, Special perturbations and General perturbations. General perturbation involves solving the equations of motion analytically while Special perturbation involves the numerical integration of the equations of motion. The General perturbation method involves series expansions and approximations to give the position and velocity of a spacecraft. The Special perturbation method includes the perturbation accelerations which are then integrated to obtain the velocity and then integrated again to obtain the position. The methods of perturbation and their effects on the orbital elements are covered in detail by Escobal [10] and Bate. Mueller, and White [8].

The ROSE™ simulation integrates the equations of motion for both the ATV and the ISS. This is the direct numerical integration technique. The result of this is the actual orbit of the ATV and the ISS. However, all rendezvous algorithms require a prediction or propagation of the orbit to a future time. Therefore, numerical integration can not be used for the rendezvous algorithms, otherwise an accurate solution would take as long as the time-of-flight that is being predicted. An analytical solution is required to accurately predict an orbit. The onboard software should be able to predict orbits including perturbations in a short period of time and without complexity. This is where a very interesting problem in simulation occurs. If the actual orbit of a spacecraft is being integrated using a certain method, the only way to exactly match or predict that orbit is to use the same integration method. This defeats the purpose of predicting the actual orbit. Since we can not use the integration method to predict the orbit for rendezvous there are errors in the analytically predicted orbit compared to the actual or integrated orbit. In the ISS/ATV mission, the position of the ISS with the J2 perturbation will have a certain value at any given mission elapsed time. However, using the Kepler object to predict that position will give an error since the Kepler object does not include the J2 oblateness perturbation. If the rendezvous mission is run including the J2 perturbation to propagate the orbits, the rendezvous algorithms will not be able to provide an accurate solution for the ATV. The ATV will transfer to a point far away from the desired target. If the simulation is integrated using two-body motion only then the ATV comes close to its desired target point.

The results in Appendix A show that the Kepler object almost exactly predicts the orbit that is numerically integrated using the Adams-Moulton fourth order integration scheme without the J2 perturbation. However, when the J2 perturbation is included in this integration, the Kepler object does not predict the orbit very well. Therefore, the current simulation does not include the J2 perturbations. The effects of the J2 perturbations were explored outside the ISS/ATV ROSE™ simulation. Two other, commercially available, orbit propagation software packages were used to compare the results of orbit propagation with and without the J2 perturbation. The results from these packages are described below. The problem becomes even more complex when other perturbations are accounted.

To study the differences that the J2 perturbation causes to the prediction of an orbit, one orbit of the ISS was predicted and compared using two-body motion, J2 perturbations, and different software, namely ROSE™, Satellite Tool Kit (STK), and Numerical Propagation of Orbital Elements (NPOE). The results were in the form of the geocentric inertial position vector (Figure 28) and magnitude of the ISS. The comparison shows how much any orbit will vary using perturbations and the different prediction schemes. The goal is to find a prediction scheme that includes the J2 perturbations and can accurately predict an orbit that is numerically integrated. This means replacing the Kepler object with the new prediction scheme which includes perturbations while integrating the equations of motion including the J2 perturbation. This will give a more accurate simulation of the real orbits.

7.2 Results

Appendix C contains the results of orbit propagation using two-body motion and the J2 perturbations, with different software. It was found that the Kepler object accurately predicts the orbit compared to the numerical integration scheme used by ROSE™, without the J2 perturbation. The prediction fails when the J2 perturbations were included. The other software packages such as STK and NPOE do not predict the orbit very well compared to the numerical integration scheme used in ROSE™, with or without perturbations. It is not possible to determine the reasons for the differences since the source code or the methods used by STK and NPOE are not available. The differences may be because different integration schemes are being used.

Table 9 gives the results of orbit propagation using different software packages and prediction schemes (Appendix C). The table shows a comparison of the position vector of an orbit after forty five minutes from an initial position. Forty five minutes were used because it is in the order of half the ISS orbit, therefore, comparable to the time for rendezvous transfer orbits.

TABLE 9. Orbit propagation using different software/schemes. The initial orbital elements are the same for all cases. The results are the X,Y,Z co-ordinates of the position vector in the geocentric inertial frame of reference (Figure 28) after 45 minutes. (Appendix C)

Tool	X (m)	Y (m)	Z (m)	Magnitude (m)
ROSE Adams-Moulton Integration, two-body motion	-5405082	3996699	278310	6728000
Kepler, two-body motion	-5405082	3996699	278310	6728000
ROSE Adams-Moulton Integration, J2	-5401717	3984012	243385	6716408
STK two-body motion	-5405382	3996336	277719	6728000
STK J2	-5400694	4003543	264821	6728000
NPOE two-body motion (osculating elements)	-5405387	3996330	277709	6728000
NPOE two-body motion (mean elements)	-5404007	4016124	299692	6739611
NPOE J2 (osculating elements)	-5402020	3983641	2427832	6716409
NPOE J2 (mean elements)	-5400698	4003539	264799	6728000

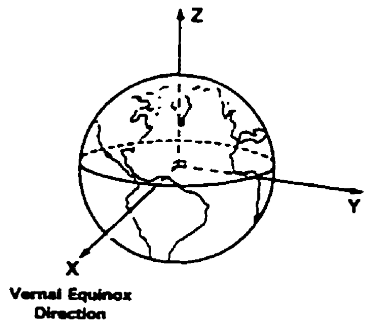


FIGURE 28. Geocentric Inertial Frame of Reference [9, p.95].

To predict the perturbation in a simulation environment the method of orbit propagation simply has to be matched by the prediction algorithm. The numerical integration used by ROSE™ may be matched by some existing analytical techniques including perturbations. In reality all the perturbations would have to be modeled in order to account for the exact motion of the spacecraft. This is very complex and difficult to do using the onboard computer available for autonomous guidance. The European Space Agency commented on this problem of perturbations. CAE Electronics Ltd., were informed by ESA, that the actual ATV will use two-body motion for rendezvous. The ATV will fire Delta-V burns every mid-point (in time) of the transfer orbit to reach the target point accurately. This will account for the errors introduced by the thrusters and all the perturbations that affect the ATV and the ISS. The method allows for simple and fast algorithms to be used. Complex calculations will take longer computing time and memory and will still require burns by the ATV since all the perturbations can not be accounted for by the mathematical models accurately. The results obtained by STK and NPOE only emphasize the fact that the algorithms used for orbit propagation or prediction have to be considered carefully. The numerical methods used have assumptions associated with them that need to be properly understood in order to apply them correctly. The results are accurate for certain orbits and may be inaccurate for others. An example would be the prediction of the lifetime of a satellite compared to the prediction of less than half an orbit for a rendezvous mission.

Chapter 8

8.0 CONCLUSIONS AND RECOMMENDATIONS

Simulation is used as a concurrent design tool from the very early stages of a spacecraft development program. The simulation of a spacecraft and its mission can be used from the conceptual design stage up to the integration, testing, and flight phases. CAE has developed a Real-time Object-oriented Software Environment simulation tool, to meet the needs of today's spacecraft development programs. The tool was used to develop a rendezvous simulation of the Automated Transfer Vehicle with the International Space Station. The algorithms for the homing phase and the closing phases, including a corrections algorithm, were developed and implemented in ROSE™, by the author. The rendezvous algorithms were designed to interface with the entire ISS/ATV simulation, concurrently designed at CAE. The algorithms determine the Delta-V burns required by the ATV to successfully complete the homing and closing phases of the rendezvous simulation. The mission definition and requirements for the homing and closing phases are covered in Chapter 3. The results of the simulation are illustrated in the form of the relative position of the ATV and the ISS (Appendix A). The simulation code, test files, and ROSE™ schematics can be found in Appendix B. The effects of perturbations on the rendezvous mission are described in Chapter 7. The project was awarded to CAE in August 1994 and was delivered and accepted in its entirety at ESA on April 1997 [1]. The simulation will be used as a template for the final ATV algorithms and specifications. The purpose of the simulator, that was delivered, is to demonstrate the ROSE™ tool and provide ESA with generic simulation including libraries for its future spacecraft development programs.

8.1 Conclusions

Four important conclusions have been arrived at in completing this thesis.

The first is that the two-body problem is sufficient to model rendezvous algorithms on an ATV type spacecraft. Generally, spacecraft are monitored and controlled by ground stations. The Delta-V commands are uplinked after complex simulations are run on the ground. The ATV requires the technology that allows it to compute the Delta-V burns autonomously using its onboard computer. Even though the ATV will be monitored by the ground station, its onboard software can determine the Delta-V burns required to bring it to the ISS orbit at approximately 300 m behind its COM. Figure 29 shows the V-bar and R-bar position of the ATV relative to the ISS during the homing and closing phases. The positions where the Delta-V burns are made by the ATV are also shown in this figure. The relative position (Figure 29) indicates that the simulation is valid and the Delta-V burns determined by the homing phase and closing phase algorithms result in a successful rendezvous.

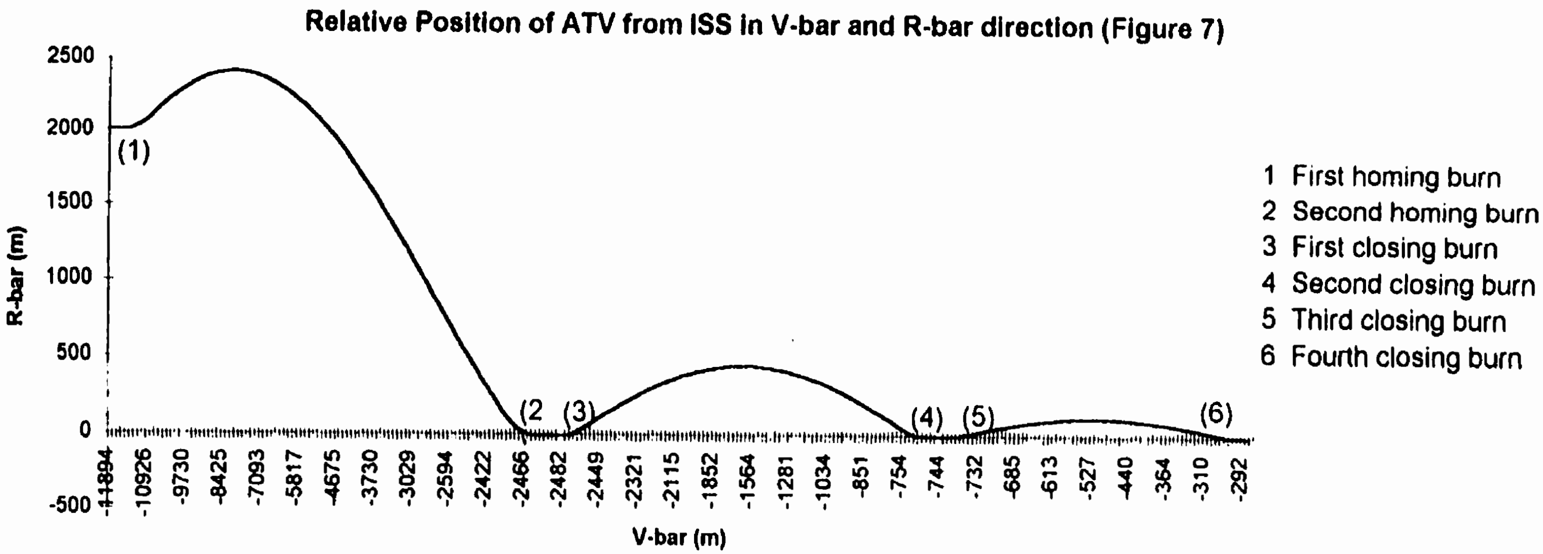


FIGURE 29. Relative Position of the ATV from the ISS in the V-bar and R-bar Frame of Reference as shown in Figure 3. V-bar and R-bar position of the ATV relative to the ISS during the homing and closing phases. The positions where the Delta-V burns are made by the ATV are also shown in this figure. The relative position indicates that the simulation is valid and the Delta-V burns determined by the homing phase and closing phase algorithms result in a successful rendezvous.

The second conclusion is that the ROSE™ simulation developed can be used to optimize various parameters such as fuel or time limits. The simulation can also be used to initiate failures in various subsystems to determine the best contingency procedures. The use of such a simulator can be used for optimization of the real spacecraft, determining mission contingency plans, training personnel, and verifying spacecraft subsystems during flight.

The third conclusion is that more accurate perturbation methods are required to determine the exact trajectories of space vehicles. In the rendezvous mission this would mean less fuel consumption and therefore better performance and lower cost. The incorporation of more complex algorithms will require better processors and more memory onboard the ATV. The current ATV onboard software uses two-body motion so the ROSE™ simulation is sufficient to meet its needs. However, the actual space station and ATV trajectories are perturbed whereas the simulated trajectories of the ISS and ATV are not perturbed in order for the simulation rendezvous algorithms to be successful. The simulation can be modified to account for the perturbations using extra correction burns as the real ATV does. The European Space Agency has accepted the space vehicle simulator as is, including the ISS/ATV simulation. ESA will modify the algorithms as required for the actual mission as the development of the ATV program continues.

One final conclusion was noted regarding the use of commercial software packages. As shown in Table 9, several different packages were used to propagate an orbit and all gave different, and incorrect results. This occurred even in the simplest two-body case. This suggests that one must be aware of software limitations and fully aware of the assumptions and equations used. The dangers of using software packages without careful analysis and comparison is clearly displayed in Table 9 where some results would be dangerously wrong for a real spacecraft mission. For any mission critical computations, it is probably best to ensure that detailed documentation is available with basic assumptions, equations, and techniques fully described along with source code.

8.2 Recommendations

The following will enhance the effectiveness of the rendezvous simulation:

- Integrate the actual orbit of the ISS and the ATV using the J2 perturbation and find an analytical method to predict the orbit including this perturbation. This will simulate the real trajectories more accurately.
- Modify the corrections schematic to determine extra correction Delta-V burns to account for the J2 perturbations as well as the errors in the thruster firings.
- Include other perturbation effects in the simulation such as aerodynamic drag and luni-solar gravity.

The above recommendations can be implemented in the second phase of the ATV program which will include Hardware-In-The-Loop simulation (HITL). This will allow various teams such as the ground station mission operations team or the space station astronauts to run various scenarios with the ISS/ATV specific simulator.

References

[1] "MSVS - Final Report to European Space Agency, ESTEC", CAE Electronics, Ltd., Montreal: 1997.

[2] "Multi-Purpose Space Vehicle Simulator (MSVS) - MSVS models. Architecture Design Document", CAE Electronics Ltd. and TRASYS Space, Montreal: 1996.

[3] "ATV System Design Baseline Report", ATV Team, Daimler-Benz Aerospace, 1995.

[4] "ATV Mission Analysis - Proximity operations for ISS Rendezvous", ATV Team, Daimler-Benz Aerospace, 1995.

[5] "ROSE™ Reference Manual", CAE Electronics, Ltd., Montreal: 1991

[6] "Multi-Purpose Space Vehicle Simulator (MSVS) for ESA", CAE Electronics Ltd. and TRASYS space: Interim Delivery, MSVS version3.0, 1996.

[7] Tapley, Byron D. (editor), Eshbach's Handbook of Engineering Fundamentals, John Wiley & Sons, Inc., 1990, 7.1-7.32.

[8] Bate, Roger R., Mueller, Donald D., White, Jerry E., Fundamentals of Astrodynamics, Dover Publications, Inc., New York, 1971.

[9] Larson, Wiley J., Wertz, James R., (editors) Space Mission Analysis and Design, Published jointly by Microcosm, Inc., Kluwer Academic Publishers, 1992.

[10] Escobal, Pedro Ramon, Methods of Orbit Determination, John Wiley & Sons, 1965.

[11] Rimrott, F.P.J., Introductory Orbit Dynamics, Friedr. Vieweg & Sohn, 1989.

[12] Bond, Victor R., Allman, Mark C., Modern Astrodynamics - Fundamentals and Perturbation Methods, Princeton University Press, 1996.

[13] Thomson, William Tyrell, Introduction to Space Dynamics, Dover Publications, Inc., New York, 1986.

[14] Staley, D.A., Orbital Mechanics and Spacecraft Control - Graduate Course Notes, Carleton University, 1996-97.

Appendix A

Results from MSVS simulator

Relative position of ATV from ISS centre of mass

Relative Position between ATV and ISS (centre of mass)
Data is extracted from ROSE simulation
Start : Homing phase
End : Closing phase
V-H-R - bar frame of reference

Time (minutes)	V-bar (m)	H-bar (m)	R-bar (m)	Magnitude (m)
0.5	-11894.37392	0	2005.072661	12062.19082
1.0	-11791.40306	0	2004.938489	11960.64231
1.5	-11688.43219	0	2004.805893	11859.11859
2.0	-11585.46131	0	2004.674873	11757.6203
2.5	-11482.0632	0	2005.66359	11655.91961
3.0	-11376.70027	0	2011.103446	11553.08816
3.5	-11268.81154	0	2021.033173	11448.61077
4.0	-11157.93597	0	2035.228038	11342.03193
4.5	-11043.63122	0	2053.441884	11232.91655
5.0	-10925.47517	0	2075.407861	11120.85094
5.5	-10803.06767	0	2100.838748	11005.44386
6.0	-10676.09467	-0.000001	2129.337211	10886.37103
6.5	-10546.24174	-0.000001	2157.856419	10764.73683
7.0	-10414.46937	-0.000001	2185.011159	10641.2145
7.5	-10280.87239	-0.000001	2210.769442	10515.88506
8.0	-10145.54774	-0.000001	2235.100925	10388.83127
8.5	-10008.59444	-0.000001	2257.976942	10260.13754
9.0	-9870.113383	-0.000002	2279.370541	10129.88985
9.5	-9730.207307	-0.000002	2299.256514	9998.175571
10.0	-9588.980604	-0.000002	2317.611426	9865.083464
10.5	-9446.53923	-0.000002	2334.413644	9730.703494
11.0	-9302.990572	-0.000002	2349.643363	9595.126759
11.5	-9158.443323	-0.000002	2363.282624	9458.445372
12.0	-9013.007352	-0.000003	2375.315343	9320.752357
12.5	-8866.793576	-0.000003	2385.727323	9182.141535
13.0	-8719.913829	-0.000003	2394.506275	9042.70742
13.5	-8572.480731	-0.000003	2401.641829	8902.545106
14.0	-8424.607553	-0.000003	2407.12555	8761.750158
14.5	-8276.408086	-0.000003	2410.950944	8620.418508
15.0	-8127.996503	-0.000003	2413.113468	8478.64634
15.5	-7979.48723	-0.000004	2413.610534	8336.52999
16.0	-7830.994808	-0.000004	2412.441515	8194.165836
16.5	-7682.633756	-0.000004	2409.607741	8051.650197
17.0	-7534.51844	-0.000004	2405.112501	7909.07923
17.5	-7386.762935	-0.000004	2398.961038	7766.548829
18.0	-7239.480893	-0.000004	2391.160542	7624.154533
18.5	-7092.785408	-0.000004	2381.720144	7481.991425
19.0	-6946.788881	-0.000004	2370.650902	7340.154048
19.5	-6801.60289	-0.000004	2357.965791	7198.736316
20.0	-6657.338059	-0.000005	2343.679688	7057.831431
20.5	-6514.103923	-0.000005	2327.809349	6917.531806
21.0	-6372.008805	-0.000005	2310.373399	6777.928994

Time (minutes)	V-bar (m)	H-bar (m)	R-bar (m)	Magnitude (m)
21.5	-6231.159683	-0.000005	2291.392299	6639.11362
22.0	-6091.662068	-0.000005	2270.888333	6501.175322
22.5	-5953.619877	-0.000005	2248.885574	6364.202697
23.0	-5817.135312	-0.000005	2225.409858	6228.283253
23.5	-5682.308738	-0.000005	2200.488756	6093.503373
24.0	-5549.238567	-0.000005	2174.151537	5959.948287
24.5	-5418.02114	-0.000005	2146.42914	5827.702045
25.0	-5288.750613	-0.000005	2117.354133	5696.847511
25.5	-5161.518849	-0.000005	2086.960674	5567.466361
26.0	-5036.415308	-0.000005	2055.284476	5439.63909
26.5	-4913.52694	-0.000005	2022.36276	5313.445033
27.0	-4792.938085	-0.000005	1988.234216	5188.962399
27.5	-4674.730373	-0.000005	1952.938952	5066.268312
28.0	-4558.982627	-0.000005	1916.518452	4945.438865
28.5	-4445.770771	-0.000005	1879.015525	4826.549191
29.0	-4335.167741	-0.000005	1840.474256	4709.673538
29.5	-4227.243397	-0.000005	1800.939953	4594.885358
30.0	-4122.064443	-0.000005	1760.459092	4482.25741
30.5	-4019.694347	-0.000005	1719.079269	4371.861866
31.0	-3920.193267	-0.000005	1676.849135	4263.770429
31.5	-3823.61798	-0.000005	1633.818349	4158.054455
32.0	-3730.021815	-0.000005	1590.03751	4054.785077
32.5	-3639.454592	-0.000005	1545.558103	3954.03333
33.0	-3551.962558	-0.000005	1500.43244	3855.870267
33.5	-3467.58834	-0.000005	1454.713593	3760.367074
34.0	-3386.370888	-0.000005	1408.455336	3667.595156
34.5	-3308.345436	-0.000004	1361.71208	3577.626212
35.0	-3233.543454	-0.000004	1314.53881	3490.532274
35.5	-3161.992614	-0.000004	1266.99102	3406.3857
36.0	-3093.716758	-0.000004	1219.124647	3325.259131
36.5	-3028.73587	-0.000004	1170.996006	3247.225372
37.0	-2967.066051	-0.000004	1122.661724	3172.357215
37.5	-2908.719501	-0.000004	1074.178672	3100.727165
38.0	-2853.704504	-0.000004	1025.6039	3032.40709
38.5	-2802.025421	-0.000004	976.994569	2967.467751
39.0	-2753.682682	-0.000003	928.407883	2905.978236
39.5	-2708.672785	-0.000003	879.901022	2848.005278
40.0	-2666.988304	-0.000003	831.531077	2793.612454
40.5	-2628.617894	-0.000003	783.354977	2742.859284
41.0	-2593.546307	-0.000003	735.429429	2695.800232
41.5	-2561.754409	-0.000003	687.810848	2652.483631
42.0	-2533.219203	-0.000003	640.555287	2612.950555
42.5	-2507.913854	-0.000003	593.718377	2577.233674
43.0	-2485.807724	-0.000002	547.355258	2545.356128
43.5	-2466.866408	-0.000002	501.520516	2517.330472
44.0	-2451.051771	-0.000002	456.268115	2493.157712
44.5	-2438.321996	-0.000002	411.651339	2472.826517
45.0	-2428.631634	-0.000002	367.722721	2456.312605

Time (minutes)	V-bar (m)	H-bar (m)	R-bar (m)	Magnitude (m)
45.5	-2421.931654	-0.000002	324.533992	2443.578369
46.0	-2418.169505	-0.000001	282.136008	2434.572751
46.5	-2417.289174	-0.000001	240.5787	2429.231373
47.0	-2419.231257	-0.000001	199.91101	2427.476939
47.5	-2423.933024	-0.000001	160.180834	2429.219876
48.0	-2431.299427	-0.000001	121.51332	2434.334075
48.5	-2440.018854	0	87.028093	2441.570375
49.0	-2449.009123	0.000001	58.213367	2449.700896
49.5	-2457.724952	0.000001	34.898996	2457.972718
50.0	-2465.635932	0.000001	16.891904	2465.693794
50.5	-2472.22827	0	3.976532	2472.231468
51.0	-2477.006476	0	-4.084584	2477.009843
51.5	-2479.495133	-0.000001	-7.550426	2479.506629
52.0	-2479.985397	-0.000001	-7.742151	2479.997481
52.5	-2480.355162	-0.000002	-7.751254	2480.367273
53.0	-2480.725536	-0.000002	-7.759951	2480.737673
53.5	-2481.096491	-0.000003	-7.76823	2481.108652
54.0	-2481.467997	-0.000003	-7.776083	2481.480181
54.5	-2481.840025	-0.000003	-7.783501	2481.85223
55.0	-2482.203839	-0.000004	-7.608845	2482.215501
55.5	-2482.305767	-0.000004	-3.702565	2482.308529
56.0	-2481.636014	-0.000004	5.281842	2481.641635
56.5	-2479.67283	-0.000005	19.290868	2479.747866
57.0	-2476.081785	-0.000005	37.168007	2476.36073
57.5	-2471.207565	-0.000005	55.190039	2471.823774
58.0	-2465.102332	-0.000005	73.032546	2466.183947
58.5	-2457.779136	-0.000005	90.674514	2459.451189
59.0	-2449.252465	-0.000005	108.095164	2451.636637
59.5	-2439.538221	-0.000005	125.273979	2442.752608
60.0	-2428.653707	-0.000006	142.190728	2432.812577
60.5	-2416.617605	-0.000006	158.825487	2421.831163
61.0	-2403.449951	-0.000006	175.158665	2409.824107
61.5	-2389.172116	-0.000006	191.171027	2396.808245
62.0	-2373.806778	-0.000006	206.843716	2382.80149
62.5	-2357.377896	-0.000006	222.158275	2367.822807
63.0	-2339.910682	-0.000006	237.096668	2351.892181
63.5	-2321.431571	-0.000006	251.641305	2335.030596
64.0	-2301.968189	-0.000006	265.775057	2317.260004
64.5	-2281.549324	-0.000006	279.481281	2298.603295
65.0	-2260.204888	-0.000006	292.743835	2279.084265
65.5	-2237.965882	-0.000006	305.547103	2258.727589
66.0	-2214.864363	-0.000006	317.876008	2237.558782
66.5	-2190.933403	-0.000006	329.716031	2215.604169
67.0	-2166.207051	-0.000006	341.053231	2192.890853
67.5	-2140.720293	-0.000006	351.874258	2169.446673
68.0	-2114.509012	-0.000006	362.166369	2145.300175
68.5	-2087.609943	-0.000006	371.917446	2120.480573
69.0	-2060.060632	-0.000006	381.116006	2095.017712

Time (minutes)	V-bar (m)	H-bar (m)	R-bar (m)	Magnitude (m)
69.5	-2031.89939	-0.000006	389.751217	2068.942034
70.0	-2003.16525	-0.000006	397.812912	2042.284538
70.5	-1973.897921	-0.000006	405.291597	2015.076743
71.0	-1944.137737	-0.000006	412.178467	1987.350656
71.5	-1913.925614	-0.000006	418.465412	1959.138729
72.0	-1883.303003	-0.000006	424.145028	1930.473829
72.5	-1852.311834	-0.000006	429.210628	1901.389201
73.0	-1820.994475	-0.000006	433.656246	1871.918432
73.5	-1789.393675	-0.000006	437.476647	1842.095422
74.0	-1757.552521	-0.000006	440.667333	1811.954349
74.5	-1725.514379	-0.000006	443.224545	1781.529643
75.0	-1693.32285	-0.000005	445.145271	1750.855959
75.5	-1661.021715	-0.000005	446.427249	1719.968147
76.0	-1628.654882	-0.000005	447.068968	1688.901236
76.5	-1596.266339	-0.000005	447.069671	1657.690417
77.0	-1563.900099	-0.000005	446.429357	1626.371019
77.5	-1531.600149	-0.000005	445.148778	1594.978511
78.0	-1499.410396	-0.000005	443.22944	1563.548487
78.5	-1467.374619	-0.000005	440.673602	1532.116672
79.0	-1435.536416	-0.000004	437.484271	1500.718924
79.5	-1403.93915	-0.000004	433.665202	1469.391249
80.0	-1372.625902	-0.000004	429.220888	1438.169822
80.5	-1341.639419	-0.000004	424.156561	1407.091013
81.0	-1311.02206	-0.000004	418.478182	1376.191423
81.5	-1280.815753	-0.000004	412.192436	1345.507933
82.0	-1251.061939	-0.000003	405.306721	1315.077759
82.5	-1221.801527	-0.000003	397.829143	1284.938519
83.0	-1193.074844	-0.000003	389.768505	1255.128308
83.5	-1164.921589	-0.000003	381.134295	1225.685792
84.0	-1137.380786	-0.000003	371.936678	1196.650302
84.5	-1110.490736	-0.000002	362.186482	1168.061952
85.0	-1084.288975	-0.000002	351.895186	1139.961755
85.5	-1058.812226	-0.000002	341.074904	1112.391757
86.0	-1034.096361	-0.000002	329.738376	1085.395172
86.5	-1010.176352	-0.000002	317.898949	1059.016526
87.0	-987.086236	-0.000001	305.57056	1033.301798
87.5	-964.859073	-0.000001	292.767726	1008.298553
88.0	-943.526903	-0.000001	279.50552	984.056072
88.5	-923.120715	-0.000001	265.799556	960.625452
89.0	-903.670407	-0.000001	251.665972	938.059681
89.5	-885.204748	0	237.121408	916.413667
90.0	-867.751351	0	222.182991	895.744209
90.5	-851.336634	0	206.86831	876.109902
91.0	-835.985791	0	191.195396	857.570943
91.5	-821.722766	0.000001	175.182705	840.188839
92.0	-808.570219	0.000001	158.849092	824.02599
92.5	-796.549502	0.000001	142.213789	809.145148
93.0	-785.680634	0.000001	125.296387	795.608725

Time (minutes)	V-bar (m)	H-bar (m)	R-bar (m)	Magnitude (m)
93.5	-775.982279	0.000001	108.116805	783.477977
94.0	-767.471719	0.000002	90.695276	772.812055
94.5	-760.16484	0.000002	73.052313	763.666959
95.0	-754.076109	0.000002	55.208694	756.094424
95.5	-749.218557	0.000002	37.185433	750.140788
96.0	-745.718571	0.000002	19.882948	745.983591
96.5	-743.857449	0.000003	7.11409	743.891467
97.0	-743.18827	0.000003	-0.619782	743.188528
97.5	-743.188486	0.000003	-3.268371	743.195672
98.0	-743.350914	0.000003	-3.266172	743.35809
98.5	-743.513126	0.000003	-3.262369	743.520283
99.0	-743.675051	0.000003	-3.257912	743.682187
99.5	-743.836644	0.000003	-3.252807	743.843756
100.0	-743.997861	0.000003	-3.24706	744.004946
100.5	-744.158658	0.000004	-3.240676	744.165714
101.0	-744.318991	0.000004	-3.233665	744.326015
101.5	-744.352361	0.000004	-2.666324	744.357137
102.0	-743.210741	0.000004	2.10404	743.213719
102.5	-741.1792	0.000004	8.625639	741.22939
103.0	-738.704199	0.000004	15.02402	738.856965
103.5	-735.794462	0.000004	21.291644	736.102456
104.0	-732.459228	0.000004	27.421127	732.972332
104.5	-728.708236	0.000004	33.405249	729.473511
105.0	-724.551712	0.000004	39.236958	725.613342
105.5	-720.000364	0.000004	44.909385	721.399596
106.0	-715.065362	0.000004	50.415847	716.84045
106.5	-709.75833	0.000004	55.749856	711.944474
107.0	-704.091329	0.000004	60.905127	706.72062
107.5	-698.076844	0.000004	65.875588	701.178203
108.0	-691.727769	0.000004	70.655382	695.326894
108.5	-685.057393	0.000004	75.238878	689.176697
109.0	-678.079383	0.000003	79.620675	682.737945
109.5	-670.807768	0.000003	83.795611	676.021277
110.0	-663.256924	0.000003	87.758766	669.037629
110.5	-655.441555	0.000003	91.505472	661.79822
111.0	-647.376676	0.000003	95.031313	654.314535
111.5	-639.077598	0.000003	98.332135	646.598318
112.0	-630.559904	0.000003	101.404048	638.661549
112.5	-621.83944	0.000003	104.243432	630.516441
113.0	-612.932286	0.000003	106.846942	622.175422
113.5	-603.854744	0.000003	109.211508	613.651127
114.0	-594.623316	0.000003	111.334345	604.956382
114.5	-585.254687	0.000003	113.212949	596.104203
115.0	-575.765701	0.000003	114.845106	587.107776
115.5	-566.173346	0.000002	116.228892	577.980461
116.0	-556.49473	0.000002	117.362676	568.735776
116.5	-546.747065	0.000002	118.245118	559.387398
117.0	-536.94764	0.000002	118.875179	549.949158

Time (minutes)	V-bar (m)	H-bar (m)	R-bar (m)	Magnitude (m)
117.5	-527.113811	0.000002	119.252114	540.435043
118.0	-517.262968	0.000002	119.375477	530.859193
118.5	-507.412527	0.000002	119.245119	521.235907
119.0	-497.5799	0.000002	118.861194	511.579652
119.5	-487.782478	0.000002	118.22415	501.905066
120.0	-478.037612	0.000002	117.334735	492.226979
120.5	-468.362591	0.000001	116.193996	482.560422
121.0	-458.77462	0.000001	114.803272	472.920652
121.5	-449.290804	0.000001	113.1642	463.323173
122.0	-439.928123	0.000001	111.278707	453.783764
122.5	-430.703416	0.000001	109.149012	444.318511
123.0	-421.633358	0.000001	106.777621	434.943846
123.5	-412.734443	0.000001	104.167324	425.676582
124.0	-404.022964	0.000001	101.321194	416.53396
124.5	-395.51499	0	98.242579	407.533694
125.0	-387.226355	0	94.935103	398.694022
125.5	-379.17263	0	91.40266	390.033754
126.0	-371.369113	0	87.649407	381.572321
126.5	-363.830805	0	83.679761	373.329823
127.0	-356.572396	0	79.498397	365.32707
127.5	-349.608245	0	75.110236	357.585615
128.0	-342.952365	-0.000001	70.520443	350.127773
128.5	-336.618406	-0.000001	65.734423	342.976626
129.0	-330.61964	-0.000001	60.757809	336.156002
129.5	-324.968942	-0.000001	55.596461	329.69043
130.0	-319.678778	-0.000001	50.256454	323.605056
130.5	-314.761189	-0.000001	44.744076	317.925523
131.0	-310.227778	-0.000001	39.065817	312.677809
131.5	-306.089695	-0.000001	33.228361	307.888007
132.0	-302.357623	-0.000002	27.238583	303.582069
132.5	-299.04177	-0.000002	21.103533	299.785489
133.0	-296.151851	-0.000002	14.830437	296.522952
133.5	-293.69708	-0.000002	8.426679	293.817944
134.0	-291.81728	-0.000002	2.425903	291.827363
134.5	-291.278239	0.008437	0.551118	291.27876
135.0	-291.348312	1.755203	1.12972	291.355789
135.5	-292.468069	5.233011	1.045714	292.51675
136.0	-293.014546	6.635275	-0.389268	293.089922
136.5	-291.627205	5.469053	-0.917281	291.679925

Appendix B

Kepler & Gauss C Code, test files, output of test files. ROSE™ schematics.

```
/*

*Title          gauss
*Module_ID      gauss.c
*Entry_Point    gauss_
*Author         CAE

*Revision_History

        6-MAR-97        ssaraf          First release

*Purpose

        Orbit determination from two positions and the time-of-flight

*Description

        This subroutine computes the the velocity vectors at two positions,
        given the position vectors, the angle between the position vectors and
        the time-of-flight. All the vectors are in the geocentric inertial frame
        of reference. The method used to solve the 'Gauss problem' is called
        the universal variable formulation. A Newton's iteration scheme is used
        as a method of convergence for the universal variable.

*Documentation

        MSVS Generic Models Detailed Design Document (HT-DD-4300-CAE)

*Include_Files

        none

*Libraries

        math.h          Contains standard mathematical functions
        libmath.h       Contains special mathematical functions
        libdyn.h        Conatains special dynamics functions

*Subroutines_Called

        angle_vv_       Angle between two vectors
        mag_vec_        Magnitude of a vector
        C_              C function of cos used in solving Gauss' problem
        S_              S function of sin used in solving Gauss' problem

*Arguments_Inputs

        radius1[3]      double          First radius vector of orbiting object [m
]
        radius2[3]      double          Second radius vector of orbiting object [
m]
        t               double          Time-of-flight [s]
        angle_diff      double          The angular separation between the first
and second radius vectors
        imax            int             Max number of iterations for Newton's Met
hod
        tolerance       double          Tolerance for convergence of Newton's Met
hod

*Arguments_Outputs

        singular        unsigned char   Flag to indicate non-real number or singu
larity
        velocity1[3]    double          Velocity vector of object at the first po
int in orbit [m/s]
        velocity2[3]    double          Velocity vector of object after time-of-f
light (second point)[m/s]
        i               int             Iteration Counter for Newton's Method

*Local_Variables

        mag_radius1     double          Magnitude of the first radius vector
        angle           double          Angle between the two given radius vector
s
        A               double          Auxiliary constant A
        mag_radius2     double          Magnitude of the second radius vector
        x               double          Universal variable x
        y               double          Universal variable z
        z               double          Auxiliary variable y
        C               double          C function for z
        S               double          S function for z
        C_deriv         double          Derivative of C function for z
        S_deriv         double          Derivative of S function for z
        t_n             double          Time-of-flight corresponding to given tri
al z
        dt_dz           double          Slope of t vs. z curve
        z_new           double          New guess for Universal variable, z
        f               double          f function
        g               double          g function
        g deriv         double          Derivative of g function

*/


#include <math.h>
#include "libmath.h"
#include "libdyn.h"

static char *RCS_ID = "@(#) $Id: gauss.c,v 2.0 1997/05/01 18:56:28 meb Exp $";

/*********************************************************************
 *********************************************************************

   The following are the C and S functions

 *********************************************************************
 *********************************************************************/

double C_(double z)
{
   double C ;
   if (z > ZERO)
      C = (1.0-cos(sqrt(z)))/z ;
   else if (z<-ZERO)
      C = (1.0-cosh(sqrt(-z)))/z ;
   else
      C = 1./2. - z/24. + (z*z)/720. ;
   return C ;
}

double S_(double z)
{
   double S, sqrt_z ;
   if (z > ZERO) {
      sqrt_z = sqrt(z) ;
      S = (sqrt_z-sin(sqrt_z))/(sqrt_z*sqrt_z*sqrt_z) ;
   }
   else if (z<-ZERO) {
      sqrt_z = sqrt(-z) ;
      S = (sinh(sqrt_z)-sqrt_z)/(sqrt_z*sqrt_z*sqrt_z) ;
   }
   else
```

```
    S = 1.0/6.0 - z/120. + (z*z)/5040. ;
  return S ;
}
#include <math.h>
#include "libmath.h"
#include "libdyn.h"
#define  sqrt_MHU 19964981.8432
#define PI 3.14159265359

/*****************************************************************
 *****************************************************************

      This functions solves the Gauss problem - Orbit determination from
      two positions and time-of-flight. Given two position vectors at two
      points in an orbit, the angular seperation between the position
      vectors, and the time-of-flight between the two points, determine
      the two velocity vectors at the two points in the orbit.

 *****************************************************************
 *****************************************************************/

void gauss_(

      /* Inputs */

            double radius1[3],   /* First radius vector of orbiting object [m
] */
            double radius2[3],   /* Second radius vector of orbiting object [
m] */
            double *t,           /* Time-of-flight [s] */
            double *angle_diff,  /* The angular separation between the first
and second radius vectors */
            int *imax,           /* Max number of iterations for Newton's Met
hod */
            double *tolerance,   /* Tolerance for convergence of Newton's Met
hod */

      /* Outputs */

            unsigned char *singular, /* Flag to indicate non-real number or s
ingularity */
            double velocity1[3],     /* Velocity vector of object at the firs
t point in orbit[m/s] */
            double velocity2[3],     /* Velocity vector of object after time-
of-flight (second point)[m/s] */
            int *i                   /* Iteration Counter for Newton's Method
 */

)



{
  /* Local variables */

  double mag_radius1 ;          /* Magnitude of the first radius vector */
  double angle;                 /* Angle between the two given radius vectors */
  double A;                     /* Auxiliary constant A */
  double mag_radius2 ;          /* Magnitude of the second radius vector */
  double x, y, z;               /* Universal variable x and z and auxiliary varia
ble y*/
  double C, S;                  /* C and S functions for z */
  double C_deriv, S_deriv;      /* Derivatives of S and C functions for z */
  double t_n;                   /* Time-of-flight corresponding to given trial z
*/
  double dt_dz;                 /* Slope of t vs. z curve */
  double z_new;                 /* New guess for Universal variable, z */
  double f, g;                  /* f and g functions */
  double g_deriv;               /* Derivative of g function */


/* Initialize singularity flag */

        (*singular) = 0;

/* Determine the magnitude of the two position vectors */

  mag_vec_(radius1, &mag_radius1);
  mag_vec_(radius2, &mag_radius2);

/* Determine the angle between the two radius vectors */

  angle_vv_(radius1, radius2, &angle);

/* Determine auxiliary constant A */

  if ((*angle_diff) <= PI)
{
/* The short-way solution */

    if ((mag_radius1>ZERO) && (mag_radius2>ZERO))
     {
       A = sqrt((mag_radius1*mag_radius2)*(1+cos(angle)));
     }
    else
     {
       A = 0.0;
       (*singular) = 1;
     }
}
  else
{
/* The long-way solution */

    if ((mag_radius1>ZERO) && (mag_radius2>ZERO))
     {
       A = -1*sqrt((mag_radius1*mag_radius2)*(1+cos(angle)));
     }
    else
     {
       A = 0.0;
       (*singular) = 1;
     }
}

/* Pick a trial value for z*/

 if ((*angle_diff) == PI)
  {
    z = 0.0;
    (*singular) = 1;
  }
 if ((*angle_diff) < PI)
  {
    z = 15.0;
  }
 if ((((*angle_diff) < (3./2.)*PI) && ((*angle diff) > PI))
  {
    z = 18.0;
  }
 if ((*angle_diff) > (3./2.)*PI)
  {
```

```
    z = 30.0;
  }

  (*i) = 0;               /* initialize counter i */
  t_n = 0.;               /* Initialize first trial time-of-flight value */

/* Iterate using the Newton's iteration scheme to determine the actual value for
z */

while ((fabs((*t)-t_n) > (*tolerance) && (*i)<(*imax)))

{

  C = C_(z);
  S = S_(z);


  (*i) = (*i) + 1;                         /* Increase counter by 1 */


  if  (fabs(z) < (ZERO))
        {
         C_deriv = (1/24 + 2*z/720 - 3*z*z/40320 + 4*z*z*z/3628800);
         S_deriv = (1/120 + 2*z/5040 - 3*z*z/362880 + 4*z*z*z/39916800);
        }
  else
        {
        S_deriv = (C-3*S)/(2*z);
        C_deriv = (1-z*S-2*C)/(2*z);
        }

/* Determine auxillary variable y */

  if (C>ZERO)
   {
     y = mag_radius1 + mag_radius2 - ((A)*(1-z*S))/(sqrt(C));
   }
  else
   {
     y = 0.0;
     (*singular) = 1;
   }

/* Ensure that y is not negative */

  if (y<ZERO)
        {
          y = fabs(y);
        }

/* Determine Universal Variable x */

  if ((y>ZERO) && (C>ZERO))
   {
    x = sqrt(y/C);
   }
  else
   {
    x = 0.0;
    (*singular) = 1;
   }

/* Determine t and dt_dz for trial value of z */

  if ((y>ZERO) && (fabs(C)>ZERO))
   {
     t_n = x*x*x*S/sqrt_MHU + (A)*sqrt(y)/sqrt_MHU;
     dt_dz = (x*x*x)*(S_deriv - (3*S*C_deriv)/(2*C))/sqrt_MHU + (A/8 * (3*S*sqrt(
y)/C + A/x))/(sqrt_MHU);
   }
  else
   {
     t_n = 0.0;
     dt_dz = 0.0;
     (*singular) = 1;
   }

/* Apply Newton's Iteration Scheme */


  if (dt_dz>ZERO)
   {
     z_new = z + ((*t) - t_n)/(dt_dz);
     z = z_new;
   }
  else
   {
    z = 0.0;
    (*singular) = 1;
   }
}

/* Evaluate  f, g, and g_deriv */

  if ((mag_radius1>ZERO) && (mag_radius2>ZERO) && (y>ZERO))
   {
     f = 1 - y/mag_radius1;
     g = A*sqrt(y/MHU);
     g_deriv = 1 - y/mag_radius2;
   }
  else
   {
     f = 0.0;
     g = 0.0;
     g_deriv = 0.0;
     (*singular) = 1;
   }

/* Compute the two velocity vectors at the two given points in the orbit */

if (fabs(g) > ZERO)
 {
  velocity1[0] = (radius2[0] - f*radius1[0])/g;
  velocity1[1] = (radius2[1] - f*radius1[1])/g;
  velocity1[2] = (radius2[2] - f*radius1[2])/g;

  velocity2[0] = (g_deriv*radius2[0] - radius1[0])/g;
  velocity2[1] = (g_deriv*radius2[1] - radius1[1])/g;
  velocity2[2] = (g_deriv*radius2[2] - radius1[2])/g;
 }
else
 {
  velocity1[0] = 0.0;
  velocity1[1] = 0.0;
  velocity1[2] = 0.0;

  velocity2[0] = 0.0;
  velocity2[1] = 0.0;
  velocity2[2] = 0.0;

  (*singular) = 1;
 }
```

```
  return;
}
```

```
/*

'Title          kepler
'Module_ID      kepler.c
'Entry_Point    kepler_
'Author         CAE

'Revision_History

        6-MAR-97        ssaraf          First release

'Purpose

        Position and velocity determination as a function of the time-of-flight

'Description

        This subroutine computes the position and velocity vectors of an
        orbiting body, given the intial position and velocity vectors, and
        the time-of-flight between the initial and final positions. All the
        vectors are in the geocentric inertial frame of reference.
        The method used to solve the 'Kepler problem' is called the
        universal variable formulation. It overcomes the problem of losing
        numerical accuracy and a slow convergence to a solution using other
        known methods, when the orbit is nearly parabolic. A Newton's iteration
        scheme is used as a method of convergence for the universal variable.

'Documentation

        MSVS Generic Models Detailed Design Document (HT-DD-4300-CAE)

'Include_Files

        none

'Libraries

        math.h          Contains standard mathematical functions
        libmath.h       Contains special mathematical functions
        libdyn.h        Conatains special dynamics functions

'Subroutines_Called

        mag_vec_        Magnitude of a vector
        mult_vv_s_      Product of two vectors resulting in a scalar
        C_              C function of cos used in solving the Kepler's problem
        S_              S function of sin used in solving the Kepler's problem

'Arguments_Inputs

        radius[3]       double          Radius vector of orbiting object (m)
        velocity[3]     double          Velocity vector of orbiting object (m/s)
        t               double          Time-of-flight (s)
        imax            int             Max number of iterations for Newton's Met
hod
        tolerance       double          Tolerance for convergence of Newton's Met
hod

'Arguments_Outputs

        singular        unsigned char   Flag to indicate non-real number or singu
larity
        fradius[3]      double          Radius vector of object after time-of-fli
ght (m)
        fvelocity[3]    double          Velocity vector of object after time-of-f
light (m/s)
        i               int             Iteration Counter for Newton's Method

'Local_Variables

        mag_radius      double          Magnitude of the radius vector
        mag_velocity    double          Magnitude of the velocity vector
        mag_fradius     double          Magnitude of the final velocity vector
        alpha           double          Inverse of the Semi-major axis
        x               double          Universal variable x
        z               double          Universal variablse z
        C               double          C function for z
        S               double          S function for z
        t_n             double          Time-of-flight corresponding to given tri
al x
        dt_dx           double          Slope of t vs. x curve
        x_new           double          New guess for Universal variable, x
        dot_prod        double          Dot product of initial radius and velocit
y vectors
        f               double          f function
        g               double          g function
        f_deriv         double          Derivative of f function
        g_deriv         double          Derivative of g function

*/


#include <math.h>
#include "libmath.h"
#include "libdyn.h"

static char *RCS_ID - "@(#) $Id: kepler.c,v 2.0 1997/05/01 18:56:28 meb Exp ccull
 $";

/*************************************************************************
 *************************************************************************

   The following are the C and S functions

 *************************************************************************
 *************************************************************************/

double C_(double z)
{
   double C ;
   if (z > ZERO)
      C = (1.0-cos(sqrt(z)))/z ;
   else if (z<-ZERO)
      C = (1.0-cosh(sqrt(-z)))/z ;
   else
      C = 1./2. - z/24. + (z*z)/720.;
   return C ;
}

double S_(double z)
{
   double S, sqrt_z ;
   if (z > ZERO) {
      sqrt_z = sqrt(z) ;
      S = (sqrt_z-sin(sqrt_z))/(sqrt_z*sqrt_z*sqrt_z) ;
   }
   else if (z<-ZERO) {
      sqrt_z = sqrt(-z) ;
      S = (sinh(sqrt_z)-sqrt_z)/(sqrt_z*sqrt_z*sqrt_z) ;
   }
   else
      S = 1.0/6.0 - z/120. + (z*z)/5040. ;
```

```c
   return S ;
}
#include <math.h>
#include "libmath.h"
#include "libdyn.h"
#define  sqrt_MHU 19964981.8432

/*******************************************************************
 *******************************************************************

        This functions solves the Kepler problem, - predicting the future
        position and velocity of an orbiting object as a function of some known
        initial position and the time-of-flight (using the universal variable
        formulation of the time-of-flight).

 *******************************************************************
 *******************************************************************/

void kepler_(

        /* Inputs */

                double radius[3],    /* Radius vector of orbiting object [m] */
                double velocity[3],  /* Velocity vector of orbiting object [m/s]
*/
                double *t,           /* Time-of-flight [s] */
                int *imax,           /* Max number of iterations for Newton's Met
hod */
                double *tolerance,   /* Tolerance for convergence of Newton's Met
hod */


        /* Outputs */
                unsigned char *singular, /* Flag to indicate non-real number or s
ingularity */
                double fradius[3],       /* Radius vector of object after time-of
-flight [m] */
                double fvelocity[3],     /* Velocity vector of object after time-
of-flight [m/s] */
                int *i                   /* Iteration Counter for Newton's Method
 */

)



(
  /* Local variables */

  double mag_radius ;            /* Magnitude of the radius vector */
  double mag_velocity;           /* Magnitude of the velocity vector */
  double mag_fradius;            /* Magnitude of the final velocity vector */
  double alpha;                  /* Inverse of the Semi-major axis */
  double x, z;                   /* Universal variables x and z */
  double C, S;                   /* C and S functions for z */
  double t_n;                    /* Time-of-flight corresponding to given trial x
*/
  double dt_dx;                  /* Slope of t vs. x curve */
  double x_new;                  /* New guess for Universal variable, x */
  double dot_prod;               /* Dot product of initial radius and velocity vec
tors */
  double f, g;                   /* f and g functions */
  double f_deriv, g_deriv;       /* Derivative of f and g functions */


  /* Determine magnitude of Radius vector and the semi-major axis from
     the radius vector and the velocity vector */

  mag_vec_(radius, &mag_radius);
  mag_vec_(velocity, &mag_velocity);

  /* Dot Product of the initial Radius and Velocity vectors */
 mult_vv_s_(radius, velocity, &dot_prod);

  if (mag_radius > ZERO)
      {
       alpha = (2.*MHU/mag_radius - (mag_velocity*mag_velocity))/MHU;
      }
  else {
       alpha = 0.;
       (*singular) = 1;
      }

  /* First guess at Universal Variable x */;

  x = sqrt_MHU*(*t)*alpha;


  /* Initialize t_n and i */

  t_n = 0.;
  (*i) =0.;

/* Start Loop Here */
/* Newton's Iteration method to determine Universal Variable x */

while ((fabs((*t)-t_n) > (*tolerance)  && (*i) <( *imax)))

{
  z = (x*x)*alpha;

/* Set limit to loop using imax and counter i */

  (*i) = (*i) + 1;                /* counter of iterations */

/* The C(z) and S(z) functions */

  C = C_(z);
  S = S_(z);

  t_n = (dot_prod*(x*x)*C/MHU + (1-mag_radius*alpha)/sqrt_MHU*(x*x*x)*S+mag_radiu
s*x/sqrt_MHU);
  dt_dx = (x*x)*C/sqrt_MHU + dot_prod/MHU*x*(1-z*S)+mag_radius*(1-z*C)/sqrt_MHU;

 if (fabs(dt_dx) > ZERO)
  {
    x_new = x + ((*t)-t_n)/(dt_dx);
    x = x_new;
  }
 else
  {
    x_new = 0.;
    x = 0.;
    (*singular) = 1;
  }
}

/* Close Loop Here */


/* When x has been calculated, evaluate the f and g functions */
```

```
 if (mag_radius > ZERO)
  {
   f = 1-(x*x)/mag_radius*C;
   g = (*t) - (x*x*x)/sqrt_MHU*S;
  }
 else
  {
   f = 0.;
   g = 0.;
   (*singular) = 1;
  }

/* Compute the final radius vector and magnitude */

  fradius[0] = f*radius[0] + g*velocity[0];
  fradius[1] = f*radius[1] + g*velocity[1];
  fradius[2] = f*radius[2] + g*velocity[2];

  mag_vec_(fradius, &mag_fradius);

/* Calculate the derivative of the f and g functions */

 if ((mag_radius > ZERO) && (mag_fradius > ZERO))
  {
   f_deriv = sqrt_MHU/(mag_radius*mag_fradius) * x*(z*S-1);
   g_deriv = 1 - (x*x)/mag_fradius*C;
  }
 else
  {
   f_deriv = 0.;
   g_deriv = 0.;
   (*singular) = 1;
  }
/* Compute the final velocity vector */

  fvelocity[0] = f_deriv*radius[0] + g_deriv*velocity[0];
  fvelocity[1] = f_deriv*radius[1] + g_deriv*velocity[1];
  fvelocity[2] = f_deriv*radius[2] + g_deriv*velocity[2];

  return;

}
```

gauss_test.Out


```
Gauss - Computes final velocity vectors at two given points in an orbit

           given the two initial radii at the two points and
           time-of-flight. All vectors are in the geocentric
           inertial frame of reference

Test Case no 1: This is a circular orbit with an initial radius in the
           positive y direction and the final radius at 30
           degrees past the z axis. The orbit is in the y-z
           plane only. The velocities determined are for the
           'short-way' trajectory.

  Input(s):

    radius1:  0.000000000000
              6700000.000000
              0.000000000000
    radius2:  0.000000000000
             -3350000.000000
              5802370.205356
    Time-of-Flight:  1819.289856579
    Angular Difference:  2.094395102390
    Tolerance:  1.000000000000e-10
    Max iterations: 10

  Actual Output(s):

    Singular: 0
    Velocity1:  0.000000000000
               -6.379495807581e-09
                7713.145398625
    Velocity2:  0.000000000000
               -6679.779858295
               -3856.572699307
    Iteration Counter: 7

  Expected results, tolerance and validation status:

    Singular: 0 (Passed)
    Velocity1:  0.000000000000 (Tol: 1e-07 %, Passed)
                2.808810418535e-06
                7713.145397812
    Velocity2:  0.000000000000 (Tol: 1e-07 %, Passed)
               -6679.779856184
               -3856.572701339

Test Case no 2: This is a circular orbit with an initial radius in the
           positive y direction. The final position vector is
           270 degrees past the initial position. The orbit
           is in the y-z plane only.
           The velocities determined are for the 'long-way'
           trajectory.

  Input(s):

    radius1:  0.000000000000
              6700000.000000
              0.000000000000
    radius2:  0.000000000000
              0.000000000000
             -6700000.000000
    Time-of-Flight:  4093.402177303
    Angular Difference:  4.712388980380
    Tolerance:  1.000000000000e-09
    Max iterations: 10

  Actual Output(s):

    Singular: 0
    Velocity1:  0.000000000000
               -2.894399325475e-09
                7713.145398622
    Velocity2:  0.000000000000
                7713.145398622
               -2.894399325475e-09
    Iteration Counter: 7

  Expected results, tolerance and validation status:

    Singular: 0 (Passed)
    Velocity1:  0.000000000000 (Tol: 1e-05 %, Passed)
               -2.226461019597e-11
                7713.145398623
    Velocity2:  0.000000000000 (Tol: 1e-05 %, Passed)
                7713.145398623
               -2.264610195965e-11

Test Case no 3: This is a circular orbit with the initial radius in
           the positive y direction and the final radius vector
           at 30 degrees past the negative y direction. The orbit
           is in the y-z pane. The velocities determined are for
           the 'long-way' trajectory.

  Input(s):

    radius1:  0.000000000000
              6700000.000000
              0.000000000000
    radius2:  0.000000000000
             -5802370.200000
             -3350000.000000
    Time-of-Flight:  3183.757249013
    Angular Difference:  3.665191000000
    Tolerance:  1.000000000000e-09
    Max iterations: 10

  Actual Output(s):

    Singular: 0
    Velocity1:  0.000000000000
                2.829427301026e-06
                7713.145397571
    Velocity2:  0.000000000000
                3856.572705337
               -6679.779859763
    Iteration Counter: 6

  Expected results, tolerance and validation status:

    Singular: 0 (Passed)
    Velocity1:  0.000000000000 (Tol: 0.0001 %, Passed)
               -0.002705232655400
                7713.151217790
    Velocity2:  0.000000000000 (Tol: 0.0001 %, Passed)
                3856.573270195
               -6679.786254251

Test Case no 4: This is a circular orbit with the initial radius
           in the positive y direction and the final radius
           in the positive z direction. The orbit is in the
           y-z plane. The velocities determined are for the
           'short-way' trajectory.
```

```
  Input(s):

    radius1:  0.000000000000
              6678000.000000
              0.000000000000
    radius2:  0.000000000000
              0.000000000000
              6678000.000000
    Time-of-Flight:  1357.744737150
    Angular Difference:  1.570796326790
    Tolerance:  1.000000000000e-09
    Max iterations: 10

  Actual Output(s):

    Singular: 0
    Velocity1:  0.000000000000
               -0.04167334228342
                7725.860879864
    Velocity2:  0.000000000000
               -7725.860879864
                0.04167334228342
    Iteration Counter: 7

  Expected results, tolerance and validation status:

    Singular: 0 (Passed)
    Velocity1:  0.000000000000 (Tol: 0.001 %, Passed)
                1.989969308898e-10
                7725.883653474
    Velocity2:  0.000000000000 (Tol: 0.001 %, Passed)
               -7725.883653474
               -1.989969308898e-10

Test Case no 5: All elements are zero

  Input(s):

    radius1:  0.000000000000
              0.000000000000
              0.000000000000
    radius2:  0.000000000000
              0.000000000000
              0.000000000000
    Time-of-Flight:  0.000000000000
    Angular Difference:  0.000000000000
    Tolerance:  1.000000000000e-09
    Max iterations: 0

  Actual Output(s):

    Singular: 1
    Velocity1:  0.000000000000
                0.000000000000
                0.000000000000
    Velocity2:  0.000000000000
                0.000000000000
                0.000000000000
    Iteration Counter: 0

  Expected results, tolerance and validation status:

    Singular: 1 (Passed)
    Velocity1:  0.000000000000 (Tol: 0.001 %, Passed)
                0.000000000000
                0.000000000000
    Velocity2:  0.000000000000 (Tol: 0.001 %, Passed)
                0.000000000000
                0.000000000000

Test Case no 6: This is a circular orbit with the initial radius in
                the positive y direction and the final position vector
                315 degrees from the initial position. The orbit is
                in the y-z plane. The velocities determined are
                for the 'long-way' trajectory.

  Input(s):

    radius1:  0.000000000000
              6700000.000000
              0.000000000000
    radius2:  0.000000000000
              4737615.433950
             -4737615.433950
    Time-of-Flight:  4775.635873520
    Angular Difference:  5.497787143780
    Tolerance:  1.000000000000e-09
    Max iterations: 10

  Actual Output(s):

    Singular: 0
    Velocity1:  0.000000000000
               -1.648167657208e-09
                7713.145398621
    Velocity2:  0.000000000000
                5454.017415644
                5454.017415642
    Iteration Counter: 5

  Expected results, tolerance and validation status:

    Singular: 0 (Passed)
    Velocity1:  0.000000000000 (Tol: 1e-05 %, Passed)
                2.202964847439e-08
                7713.145398650
    Velocity2:  0.000000000000 (Tol: 1e-05 %, Passed)
                5454.017415647
                5454.017415679

Test Case no 7: The initial and final vectors are colinear. There is
                no solution for this case and the singular flag
                is set.

  Input(s):

    radius1:  0.000000000000
              6700000.000000
              0.000000000000
    radius2:  0.000000000000
             -6700000.000000
              0.000000000000
    Time-of-Flight:  2728.934784860
    Angular Difference:  3.141592653590
    Tolerance:  1.000000000000e-09
    Max iterations: 10

  Actual Output(s):

    Singular: 1
    Velocity1:  0.000000000000
```

```
              0.000000000000
              0.000000000000
  Velocity2:  0.000000000000
              0.000000000000
              0.000000000000
  Iteration Counter: 10

Expected results, tolerance and validation status:

  Singular: 1 (Passed)
  Velocity1:  0.000000000000 (Tol: 0.001 %, Passed)
              0.000000000000
              0.000000000000
  Velocity2:  0.000000000000 (Tol: 0.001 %, Passed)
              0.000000000000
              0.000000000000
```

kepler_test.Out


```
      Kepler - Computes final radius and velocity vectors given intial
      radius, velocity and time-of-flight

   Test Case no 1: A general case that has been computed in a textbook.

     Input(s):

       radius:  6378145.000000
                0.000000000000
                0.000000000000
       velocity:  0.000000000000
                  0.000000000000
                  8695.905108000
       Time-of-Flight:  1613.623748800
       Max iterations: 5
       Tolerance:  1.000000000000e-09

     Actual Output(s):

       Singular Flag: 0
       Final Radius: -2045250.640642
                      0.000000000000
                      7886177.585374
       Final Velocity: -6956.542531254
                        0.000000000000
                       -294.9339759730
       Iteration Counter: 4

     Expected results, tolerance and validation status:

       Singular Flag: 0 (Passed)
       Final Radius: -2045250.640658 (Tol: 1e-09 %, Passed)
                      0.000000000000
                      7886177.585366
       Final  Velocity: -6956.542531266 (Tol: 1e-09 %, Passed)
                         0.000000000000
                        -294.9339759940

   Test Case no 2: A circular orbit with intial values of radius in the positive
 y direction and the velocity in the positive z direction, with a time-of-flight
 of a quarter of the period. The expected value is calculated using the same
 algorithm and an accuracy of thirteen decimal places, not the expected value of
 an actual orbit.

     Input(s):

       radius:  0.000000000000
                6678000.000000
                0.000000000000
       velocity:  0.000000000000
                  0.000000000000
                  7725.840043170
       Time-of-Flight:  1357.752401257
       Max iterations: 5
       Tolerance:  1.000000000000e-09

     Actual Output(s):

       Singular Flag: 0
       Final Radius:  0.000000000000
                      1.210791666928e-05
                      6678000.000015
       Final Velocity:  0.000000000000
                       -7725.840043146
                        3.140188153732e-08
       Iteration Counter: 2

     Expected results, tolerance and validation status:

       Singular Flag: 0 (Passed)
       Final Radius:  0.000000000000 (Tol: 1e-09 %, Passed)
                      1.219169565303e-05
                      6678000.000009
       Final  Velocity:  0.000000000000 (Tol: 1e-09 %, Passed)
                        -7725.840043159
                         2.476897386379e-08

   Test Case no 3: A circular orbit with intial values of radius in the positive
 y direction and the velocity in the positive z direction, with a time-of-flight
 of half of the period. The expected value is calculated using the same
 algorithm and an accuracy of thirteen decimal places, not the expected value of
 an actual orbit.

     Input(s):

       radius:  0.000000000000
                6678000.000000
                0.000000000000
       velocity:  0.000000000000
                  0.000000000000
                  7725.840043170
       Time-of-Flight:  2715.504802515
       Max iterations: 5
       Tolerance:  1.000000000000e-09

     Actual Output(s):

       Singular Flag: 0
       Final Radius:  0.000000000000
                     -6678000.000018
                      6.186227959922e-05
       Final Velocity:  0.000000000000
                       -7.156508332550e-08
                       -7725.840043149
       Iteration Counter: 1

     Expected results, tolerance and validation status:

       Singular Flag: 0 (Passed)
       Final Radius:  0.000000000000 (Tol: 1e-09 %, Passed)
                     -6678000.000018
                      4.359309207559e-05
       Final  Velocity:  0.000000000000 (Tol: 1e-09 %, Passed)
                        -5.043077541430e-08
                        -7725.840043149

   Test Case no 4: A circular orbit with intial values of radius in the positive
 y direction and the velocity in the positive z direction, with a time-of-flight
 of a three and a quarter of the period. The expected value is calculated using th
 e same
 algorithm and an accuracy of thirteen decimal places, not the expected value of
 an actual orbit.

     Input(s):

       radius:  0.000000000000
                6678000.000000
                0.000000000000
       velocity:  0.000000000000
                  0.000000000000
                  7725.840043170
       Time-of-Flight:  4073.257203772
```

```
      Max iterations: 5
      Tolerance:  1.000000000000e-09

    Actual Output(s):

      Singular Flag: 0
      Final Radius:  0.000000000000
                    -0.0001108521994198
                    -6678000.000015
      Final Velocity:  0.000000000000
                       7725.840043146
                      -1.108338029152e-07
      Iteration Counter: 2

    Expected results, tolerance and validation status:

      Singular Flag: 0 (Passed)
      Final Radius:  0.000000000000 (Tol: 1e-09 %, Passed)
                    -7.421335146418e-05
                    -6678000.000009
      Final  Velocity:  0.000000000000 (Tol: 1e-09 %, Passed)
                        7725.840043159
                       -7.519296757957e-08
```

Test Case no 5: A circular orbit with intial values of radius in the positive y direction and the velocity in the positive z direction, with a time-of-flight of one period. The expected value is calculated using the samealgorithm and an accuracy of thirteen decimal places, not the expected value of an actual orbit.

```
    Input(s):

      radius:  0.000000000000
               6678000.000000
               0.000000000000
      velocity:  0.000000000000
                 0.000000000000
                 7725.840043170
      Time-of-Flight:  5431.009605029
      Max iterations: 5
      Tolerance:  1.000000000000e-09

    Actual Output(s):

      Singular Flag: 0
      Final Radius:  0.000000000000
                     6678000.000000
                    -0.0001237175325878
      Final Velocity:  0.000000000000
                       1.431409453370e-07
                       7725.840043170
      Iteration Counter: 1

    Expected results, tolerance and validation status:

      Singular Flag: 0 (Passed)
      Final Radius:  0.000000000000 (Tol: 1e-09 %, Passed)
                     6678000.000000
                    -8.716510431943e-05
      Final  Velocity:  0.000000000000 (Tol: 1e-09 %, Passed)
                        1.008561615603e-07
                        7725.840043170
```

Test Case no 6: All values are zero. This test should set off the singularity flag

```
    Input(s):

      radius:  0.000000000000
               0.000000000000
               0.000000000000
      velocity:  0.000000000000
                 0.000000000000
                 0.000000000000
      Time-of-Flight:  0.000000000000
      Max iterations: 0
      Tolerance:  0.000000000000

    Actual Output(s):

      Singular Flag: 1
      Final Radius:  0.000000000000
                     0.000000000000
                     0.000000000000
      Final Velocity:  0.000000000000
                       0.000000000000
                       0.000000000000
      Iteration Counter: 0

    Expected results, tolerance and validation status:

      Singular Flag: 1 (Passed)
      Final Radius:  0.000000000000 (Tol: 1e-09 %, Passed)
                     0.000000000000
                     0.000000000000
      Final  Velocity:  0.000000000000 (Tol: 1e-09 %, Passed)
                        0.000000000000
                        0.000000000000
```

```
#include<math.h>
#include<stdio.h>
#include<string.h>
#include "libtest.h"
#include "libmath.h"


/**************************************************/
/*                                                */
/*                 GAUSS Test File                */
/*                                                */
/**************************************************/
static char *RCS_ID = "@(#) $Id: gauss_test.c,v 2.0 1997/05/01 18:56:28 meb Exp c
cull $";


/* Handler object testing */

main() {

  /* Variable declaration */

  int k, N;
  char *Title;
  char *Desc[Ntest];

  double I_radius1[3][Ntest];
  double I_radius2[3][Ntest];
  double I_t[Ntest];
  double I_angle_diff[Ntest];
  int    I_imax[Ntest];
  double I_tolerance[Ntest];
  unsigned char O_singular[Ntest];
  double O_velocity1[3][Ntest], C_velocity1[Ntest];
  double O_velocity2[3][Ntest], C_velocity2[Ntest];

  double radius1[3];
  double radius2[3];
  double t;
  double angle_diff;
  int    imax;
  double tolerance;
  unsigned char singular;
  double velocity1[3];
  double velocity2[3];
  int    i;

  /* Initialize test case data tables */

  N = 7;

  Title = "\tGauss - Computes final velocity vectors at two given points in an or
bit \
          \n\t\tgiven the two initial radii at the two points and\n\
          \ttime-of-flight. All vectors are in the geocentric\n\
          \tinertial frame of reference";

  Desc[0] = "This is a circular orbit with an initial radius in the\n\
            \tpositive y direction and the final radius at 30 \n\
            \tdegrees past the z axis. The orbit is in the y-z\n\
            \tplane only. The velocities determined are for the \n\
            \t'short-way' trajectory. ";
  I_radius1[0][0] = 0.0;
  I_radius1[1][0] = 6700000.0;
  I_radius1[2][0] = 0.0;
  I_radius2[0][0] = 0.0;
  I_radius2[1][0] = -3.35000E6;
  I_radius2[2][0] = 5.8023702053557E6;
  I_t[0] = 1.8192898565791E3;
  I_angle_diff[0] = 2.09439510239;
  I_imax[0] = 10;
  I_tolerance[0] = 0.0000000001;
  O_singular[0] = 0;
  O_velocity1[0][0] = 0.0;
  O_velocity1[1][0] = 2.8088104185350E-6;
  O_velocity1[2][0] = 7.7131453978121E3;
  O_velocity2[0][0] = 0.0;
  O_velocity2[1][0] = -6.6797798561839E3;
  O_velocity2[2][0] = -3.8565727013386E3;
  C_velocity1[0] = 1.0e-9;
  C_velocity2[0] = 1.0e-9;

  Desc[1] = "This is a circular orbit with an initial radius in the\n\
            \tpositive y direction. The final position vector is\n\
            \t270 degrees past the initial position. The orbit\n\
            \tis in the y-z plane only.\n\
            \tThe velocities determined are for the 'long-way'\n\
            \ttrajectory. ";
  I_radius1[0][1] = 0.0;
  I_radius1[1][1] = 6700000.0;
  I_radius1[2][1] = 0.0;
  I_radius2[0][1] = 0.0;
  I_radius2[1][1] = 0.0;
  I_radius2[2][1] = -6700000.0;
  I_t[1] = 4.0934021773029E3;
  I_angle_diff[1] = 4.71238898038;
  I_imax[1] = 10;
  I_tolerance[1] = 0.000000001;
  O_singular[2] = 0;
  O_velocity1[0][1] = 0.0;
  O_velocity1[1][1] = -2.2264610195965E-11;
  O_velocity1[2][1] = 7.7131453986229E3;
  O_velocity2[0][1] = 0.0;
  O_velocity2[1][1] = 7.7131453986229E3;
  O_velocity2[2][1] = -2.264610195965E-11;
  C_velocity1[1] = 1.0e-7;
  C_velocity2[1] = 1.0e-7;

  Desc[2] = "This is a circular orbit with the initial radius in\n\
            \tthe positive y direction and the final radius vector\n\
            \tat 30 degrees past the negative y direction. The orbit\n\
            \tis in the y-z pane. The velocities determined are for\n\
            \tthe 'long-way' trajectory.";
  I_radius1[0][2] = 0.0;
  I_radius1[1][2] = 6700000.0;
  I_radius1[2][2] = 0.0;
  I_radius2[0][2] = 0.0;
  I_radius2[1][2] = -5802370.2;
  I_radius2[2][2] = -3350000.0;
  I_t[2] = 3.1837572490134E3;
  I_angle_diff[2] = 3.665191;
  I_imax[2] = 10;
  I_tolerance[2] = 0.000000001;
  O_singular[2] = 0;
  O_velocity1[0][2] = 0.0;
  O_velocity1[1][2] = -0.0027052326554;
  O_velocity1[2][2] = 7.7131512177896E3;
  O_velocity2[0][2] = 0.0;
  O_velocity2[1][2] = 3.8565732701951E3;
  O_velocity2[2][2] = -6.6797862542512E3;
  C_velocity1[2] = 1.0E-6;
```

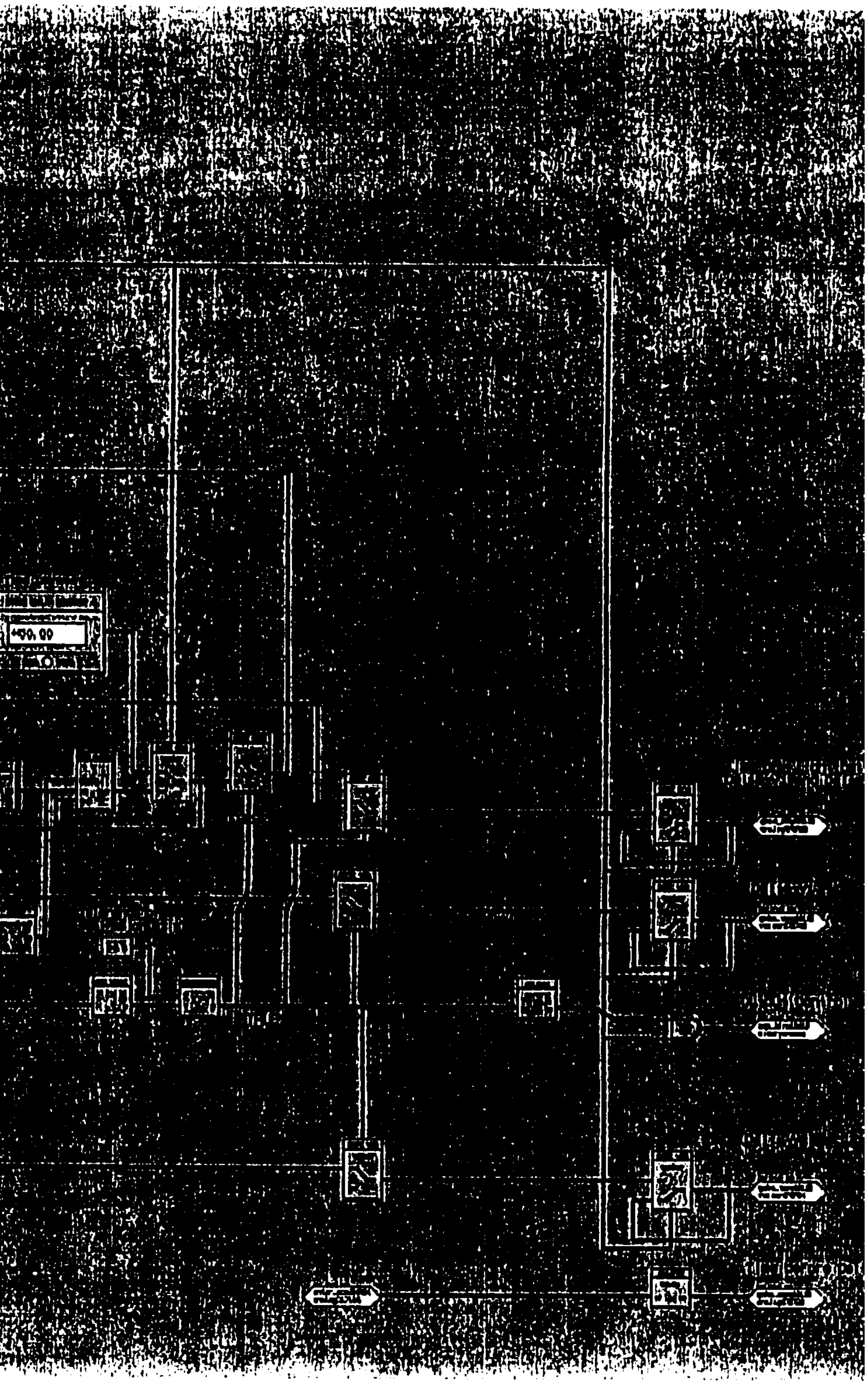
```
C_velocity2[2] = 1.0E-6;

Desc[3] = "This is a circular orbit with the initial radius\n\
          \tin the positive y direction and the final radius\n\
          \tin the positive z direction. The orbit is in the\n\
          \ty-z plane. The velocities determined are for the\n\
          \t'short-way' trajectory.";
I_radius1[0][3] = 0.0;
I_radius1[1][3] = 6678000.0;
I_radius1[2][3] = 0.0;
I_radius2[0][3] = 0.0;
I_radius2[1][3] = 0.0;
I_radius2[2][3] = 6678000.0;
I_t[3] = 1.3577447371499E3;
I_angle_diff[3] = 1.57079632679;
I_imax[3] = 10;
I_tolerance[3] = 0.000000001;
O_singular[3] = 0;
O_velocity1[0][3] = 0.0;
O_velocity1[1][3] = 1.9899693088975E-10;
O_velocity1[2][3] = 7.7258836534737E3;
O_velocity2[0][3] = 0.0;
O_velocity2[1][3] = -7.7258836534737E3;
O_velocity2[2][3] = -1.9899693088975E-10;
C_velocity1[3] = 1.0E-5;
C_velocity2[3] = 1.0E-5;

Desc[4] = "All elements are zero";
I_radius1[0][4] = 0.0;
I_radius1[1][4] = 0.0;
I_radius1[2][4] = 0.0;
I_radius2[0][4] = 0.0;
I_radius2[1][4] = 0.0;
I_radius2[2][4] = 0.0;
I_t[4] = 0.0;
I_angle_diff[4] = 0.0;
I_imax[4] = 0;
I_tolerance[4] = 0.000000001;
O_singular[4] = 1;
O_velocity1[0][4] = 0.0;
O_velocity1[1][4] = 0.0;
O_velocity1[2][4] = 0.0;
O_velocity2[0][4] = 0.0;
O_velocity2[1][4] = 0.0;
O_velocity2[2][4] = 0.0;
C_velocity1[4] = 1.0E-5;
C_velocity2[4] = 1.0E-5;

Desc[5] = "This is a circular orbit with the initial radius in\n\
          \tthe positive y direction and the final position vector\n\
          \t315 degrees from the initial position. The orbit is\n\
          \tin the y-z plane. The velocities determined are\n\
          \tfor the 'long-way' trajectory.";
I_radius1[0][5] = 0.0;
I_radius1[1][5] = 6700000.0;
I_radius1[2][5] = 0.0;
I_radius2[0][5] = 0.0;
I_radius2[1][5] = 4.7376154339499E6;
I_radius2[2][5] = -4.7376154339499E6;
I_t[5] = 4.7756358735201E3;
I_angle_diff[5] = 5.49778714378;
I_imax[5] = 10;
I_tolerance[5] = 0.000000001;
O_singular[5] = 0;
O_velocity1[0][5] = 0.0;
O_velocity1[1][5] = 2.2029648474395E-8;
O_velocity1[2][5] = 7.7131453986496E3;
O_velocity2[0][5] = 0.0;
O_velocity2[1][5] = 5.4540174156473E3;
O_velocity2[2][5] = 5.4540174156786E3;
C_velocity1[5] = 1.0E-7;
C_velocity2[5] = 1.0E-7;

Desc[6] = "The initial and final vectors are colinear. There is \n\
          \tno solution for this case and the singular flag\n\
          \tis set.";
I_radius1[0][6] = 0.0;
I_radius1[1][6] = 6700000.0;
I_radius1[2][6] = 0.0;
I_radius2[0][6] = 0.0;
I_radius2[1][6] = -6700000.0;
I_radius2[2][6] = 0.0;
I_t[6] = 2728.93478486;
I_angle_diff[6] = 3.14159265359;
I_imax[6] = 10;
I_tolerance[6] = 0.000000001;
O_singular[6] = 1;
O_velocity1[0][6] = 0.0;
O_velocity1[1][6] = 0.0;
O_velocity1[2][6] = 0.0;
O_velocity2[0][6] = 0.0;
O_velocity2[1][6] = 0.0;
O_velocity2[2][6] = 0.0;
C_velocity1[6] = 1.0E-5;
C_velocity2[6] = 1.0E-5;


/* Loop over test cases */

printf("%s \n", Title);
printf("\n");

for (k = 0; k < N; k++) {

      printf("  Test Case no %d: %s \n", k+1, Desc[k]);
      printf("\n");

      /* Copy table elements into object inputs */

      copy_vec(k, I_radius1, radius1);
      copy_vec(k, I_radius2, radius2);
      copy_sca(k, I_t, &t);
      copy_sca(k, I_angle_diff, &angle_diff);
      copy_idx(k, I_imax, &imax);
      copy_sca(k, I_tolerance, &tolerance);

      /* Print inputs */

      printf("    Input(s): \n");
      printf("\n");
      print_vec("      radius1", radius1);
      print_vec("      radius2", radius2);
      print_sca("      Time-of-Flight", t);
      print_sca("      Angular Difference", angle_diff);
      print_sca("      Tolerance", tolerance);
      print_idx("      Max iterations", imax);

      printf("\n");

      /* Call tested object */

      gauss_(radius1, radius2, &t, &angle_diff, &imax, &tolerance, &singular, v
```

```
elocity1, velocity2, &i);

        /* Print actual outputs */

        printf("    Actual Output(s): \n");
        printf("\n");
        print_bin("      Singular", singular);
        print_vec("      Velocity1", velocity1);
        print_vec("      Velocity2", velocity2);
        print_idx("      Iteration Counter", i);
        printf("\n");

    /* Verify and print results */

        printf("    Expected results, tolerance and validation status: \n");
        printf("\n");
        check_bin("      Singular", singular, k, O_singular);
        check_vec("      Velocity1", velocity1, k, O_velocity1, C_velocity1);
        check_vec("      Velocity2", velocity2, k, O_velocity2, C_velocity2);
    printf("\n");

  )
)
```

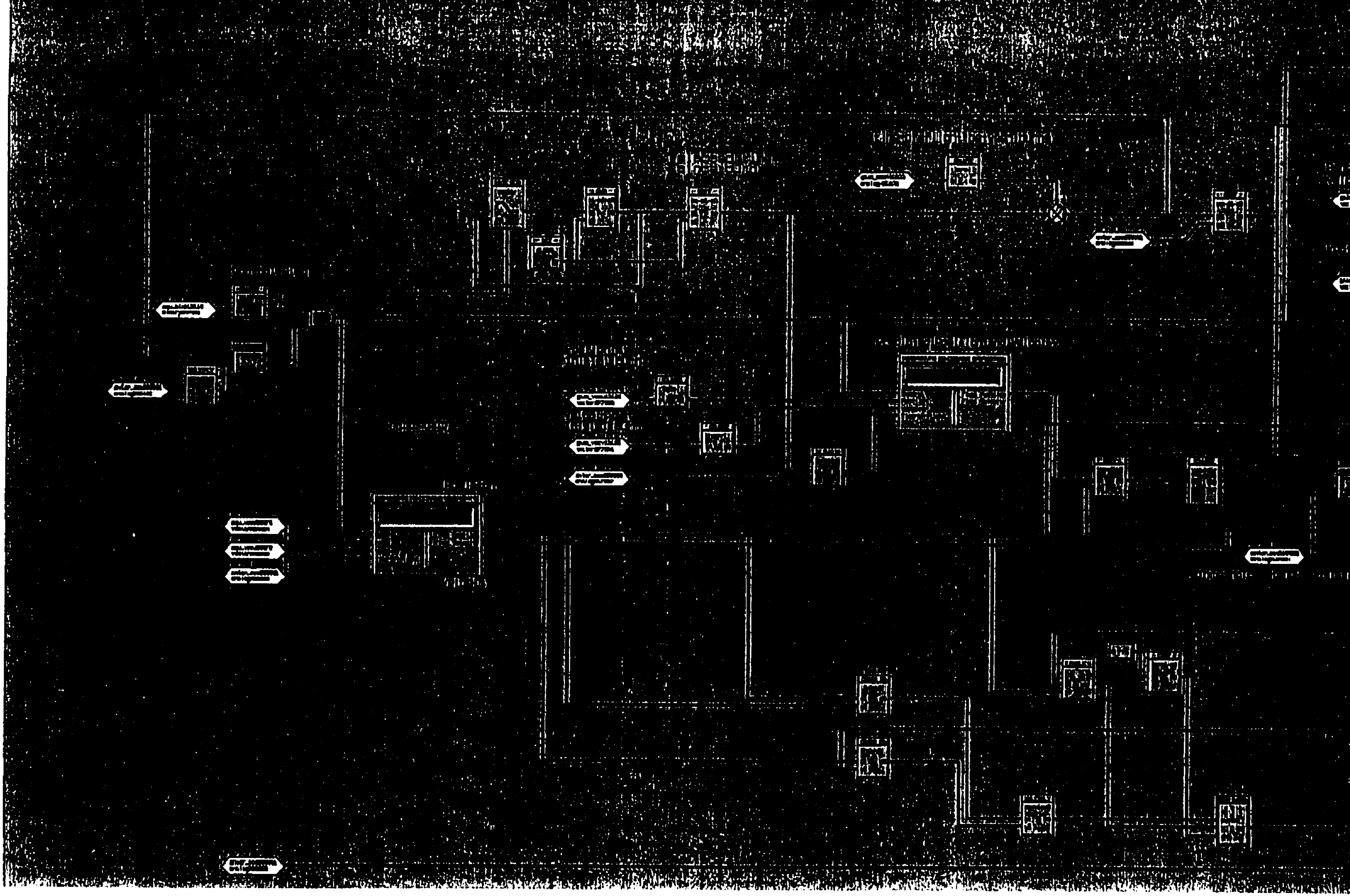
```c
#include<math.h>
#include<stdio.h>
#include<string.h>
#include "libtest.h"
#include "libmath.h"


/*****************************************************/
/*                                                   */
/*                  KEPLER Test File                 */
/*                                                   */
/*****************************************************/
static char *RCS_ID = "@(#) $Id: kepler_test.c,v 2.1 1997/05/01 19:14:42 meb Exp
ccull $";


/* Handler object testing */

main() {

  /* Variable declaration */

  int k, N;
  char *Title;
  char *Desc[Ntest];

  double I_radius[3][Ntest];
  double I_velocity[3][Ntest];
  double I_t[Ntest];
  int    I_imax[Ntest];
  double I_tolerance[Ntest];
  unsigned char O_singular[Ntest];
  double O_fradius[3][Ntest], C_fradius[Ntest];
  double O_fvelocity[3][Ntest], C_fvelocity[Ntest];

  double radius[3];
  double velocity[3];
  double t;
  int    imax;
  double tolerance;
  unsigned char singular = 0;
  double fradius[3];
  double fvelocity[3];
  int    i;

  /* Initialize test case data tables */

  N = 6;

  Title = "\tKepler - Computes final radius and velocity vectors given intial\n\t
\
radius, velocity and time-of-flight";

  Desc[0] = "A general case that has been computed in a textbook.";
  I_radius[0][0] = 6378145.0;
  I_radius[1][0] = 0.0;
  I_radius[2][0] = 0.0;
  I_velocity[0][0] = 0.0;
  I_velocity[1][0] = 0.0;
  I_velocity[2][0] = 8695.905108;
  I_t[0] = 1613.6237488;
  I_imax[0] = 5;
  I_tolerance[0] = 0.000000001;
  O_singular[0] = 0;
  O_fradius[0][0] = -2.0452506406577E6;
  O_fradius[1][0] = 0.0;
  O_fradius[2][0] = 7.8861775853657E6;
  O_fvelocity[0][0] = -6.9565425312658E3;
  O_fvelocity[1][0] = 0.0;
  O_fvelocity[2][0] = -294.9339759940204;
  C_fradius[0] = 1.0e-11;
  C_fvelocity[0] = 1.0e-11;

  Desc[1] = "A circular orbit with intial values of radius in the positive\n\
y direction and the velocity in the positive z direction, with a time-of-flight\n
\
of a quarter of the period. The expected value is calculated using the same\n\
algorithm and an accuracy of thirteen decimal places, not the expected value of\n
\
an actual orbit.";

  I_radius[0][1] = 0.0;
  I_radius[1][1] = 6678000.0;
  I_radius[2][1] = 0.0;
  I_velocity[0][1] = 0.0;
  I_velocity[1][1] = 0.0;
  I_velocity[2][1] = 7725.84004317;
  I_t[1] = 1.3577524012573E3;
  I_imax[1] = 5;
  I_tolerance[1] = 0.000000001;
  O_singular[1] = 0;
  O_fradius[0][1] = 0.0;
  O_fradius[1][1] = 1.219169565303E-5;
  O_fradius[2][1] = 6.6780000000092E6;
  O_fvelocity[0][1] = 0.0;
  O_fvelocity[1][1] = -7.7258400431593E3;
  O_fvelocity[2][1] = 2.4768973863789E-8;
  C_fradius[1] = 1.0e-11;
  C_fvelocity[1] = 1.0e-11;

  Desc[2] = "A circular orbit with intial values of radius in the positive\n\
y direction and the velocity in the positive z direction, with a time-of-flight\n
\
of half of the period. The expected value is calculated using the same\n\
algorithm and an accuracy of thirteen decimal places, not the expected value of\n
\
an actual orbit.";

  I_radius[0][2] = 0.0;
  I_radius[1][2] = 6678000.0;
  I_radius[2][2] = 0.0;
  I_velocity[0][2] = 0.0;
  I_velocity[1][2] = 0.0;
  I_velocity[2][2] = 7725.84004317;
  I_t[2] = 2.7155048025145E3;
  I_imax[2] = 5;
  I_tolerance[2] = 0.000000001;
  O_singular[2] = 0;
  O_fradius[0][2] = 0.0;
  O_fradius[1][2] = -6.6780000000184E6;
  O_fradius[2][2] = 4.3593092075595E-5;
  O_fvelocity[0][2] = 0.0;
  O_fvelocity[1][2] = -5.04307754143E-8;
  O_fvelocity[2][2] = -7.7258400431487E3;
  C_fradius[2] = 1.0e-11;
  C_fvelocity[2] = 1.0e-11;

  Desc[3] = "A circular orbit with intial values of radius in the positive\n\
y direction and the velocity in the positive z direction, with a time-of-flight\n
\
```

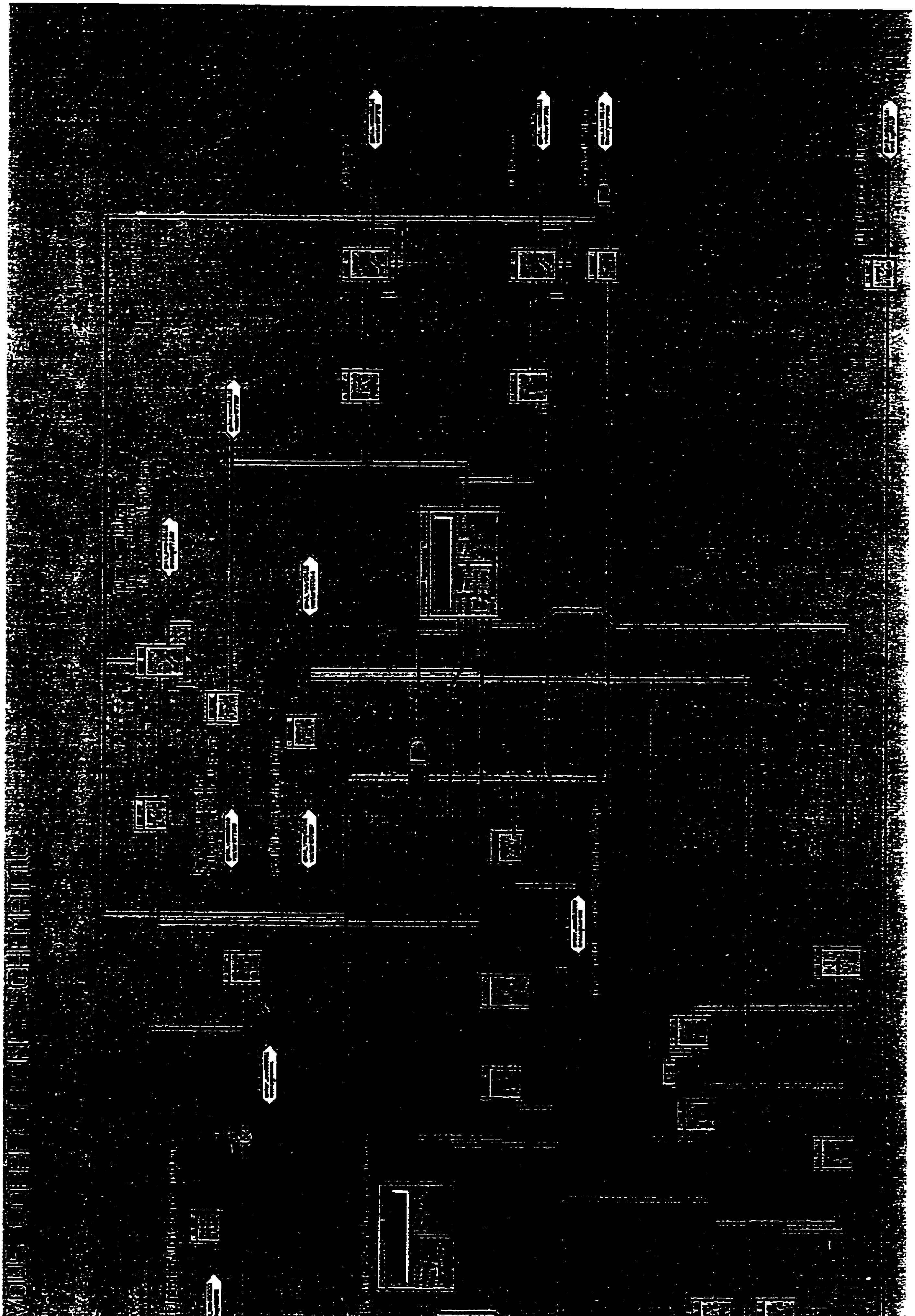
```c
of a three and a quarter of the period. The expected value is calculated using th
e same\n\
algorithm and an accuracy of thirteen decimal places, not the expected value of\n
\
an actual orbit.";

  I_radius[0][3] = 0.0;
  I_radius[1][3] = 6678000.0;
  I_radius[2][3] = 0.0;
  I_velocity[0][3] = 0.0;
  I_velocity[1][3] = 0.0;
  I_velocity[2][3] = 7725.84004317;
  I_t[3] = 4.0732572037718E3;
  I_imax[3] = 5;
  I_tolerance[3] = 0.000000001;
  O_singular[3] = 0;
  O_fradius[0][3] = 0.0;
  O_fradius[1][3] = -7.4213351464181E-5;
  O_fradius[2][3] = -6.6780000000092E6;
  O_fvelocity[0][3] = 0.0;
  O_fvelocity[1][3] = 7.7258400431593E3;
  O_fvelocity[2][3] = -7.5192967579569E-8;
  C_fradius[3] = 1.0e-11;
  C_fvelocity[3] = 1.0e-11;

  Desc[4] = "A circular orbit with intial values of radius in the positive\n\
y direction and the velocity in the positive z direction, with a time-of-flight\n
\
of one period. The expected value is calculated using the samealgorithm\n\
and an accuracy of thirteen decimal places, not the expected value of\n\
an actual orbit.";

  I_radius[0][4] = 0.0;
  I_radius[1][4] = 6678000.0;
  I_radius[2][4] = 0.0;
  I_velocity[0][4] = 0.0;
  I_velocity[1][4] = 0.0;
  I_velocity[2][4] = 7725.84004317;
  I_t[4] = 5.431009605029E3;
  I_imax[4] = 5;
  I_tolerance[4] = 0.000000001;
  O_singular[4] = 0;
  O_fradius[0][4] = 0.0;
  O_fradius[1][4] = 6.678E6;
  O_fradius[2][4] = -8.7165104319432E-5;
  O_fvelocity[0][4] = 0.0;
  O_fvelocity[1][4] = 1.0085616156032E-7;
  O_fvelocity[2][4] = 7.72584004317E3;
  C_fradius[4] = 1.0e-11;
  C_fvelocity[4] = 1.0e-11;

Desc[5] = "All values are zero. This test should set off the\n\
singularity flag";

  I_radius[0][5] = 0.0;
  I_radius[1][5] = 0.0;
  I_radius[2][5] = 0.0;
  I_velocity[0][5] = 0.0;
  I_velocity[1][5] = 0.0;
  I_velocity[2][5] = 0.0;
  I_t[5] = 0.0;
  I_imax[5] = 0;
  I_tolerance[5] = 0.0;
  O_singular[5] = 1;
  O_fradius[0][5] = 0.0;
  O_fradius[1][5] = 0.0;
  O_fradius[2][5] = 0.0;
  O_fvelocity[0][5] = 0.0;
  O_fvelocity[1][5] = 0.0;
  O_fvelocity[2][5] = 0.0;
  C_fradius[5] = 1.0e-11;
  C_fvelocity[5] = 1.0e-11;


  /* Loop over test cases */

  printf("%s \n", Title);
  printf("\n");

  for (k = 0; k < N; k++) {

        printf("  Test Case no %d: %s \n", k+1, Desc[k]);
        printf("\n");

        /* Copy table elements into object inputs */

        copy_vec(k, I_radius, radius);
        copy_vec(k, I_velocity, velocity);
        copy_sca(k, I_t, &t);
        copy_idx(k, I_imax, &imax);
        copy_sca(k, I_tolerance, &tolerance);

        /* Print inputs */

        printf("    Input(s): \n");
        printf("\n");
        print_vec("      radius", radius);
        print_vec("      velocity", velocity);
        print_sca("      Time-of-Flight", t);
        print_idx("      Max iterations", imax);
        print_sca("      Tolerance", tolerance);
        printf("\n");

        /* Call tested object */

        kepler_(radius, velocity, &t, &imax, &tolerance, &singular, fradius, fve
locity, &i);

        /* Print actual outputs */

        printf("    Actual Output(s): \n");
        printf("\n");
        print_bin("      Singular Flag", singular);
        print_vec("      Final Radius", fradius);
        print_vec("      Final Velocity", fvelocity);
        print_idx("      Iteration Counter", i);
        printf("\n");

   /* Verify and print results */

        printf("    Expected results, tolerance and validation status: \n");
        printf("\n");
        check_bin("      Singular Flag", singular, k, O_singular);
        check_vec("      Final Radius", fradius, k, O_fradius, C_fradius);
        check_vec("      Final  Velocity", fvelocity, k, O_fvelocity, C_fveloc
y);
   printf("\n");

  }
}
```

OLOSTING SCHEMATIC


+90.00

RENDEZVOUS CORRECTIONS SCHEMATIC





Appendix C

Results of ROSE™ numerical integration, without J2 perturbation,

compared to Kepler prediction (one orbit)

Results of ROSE™ numerical integration with J2 perturbations.

Results from STK and NPOE software.

Orbit Propagation - Adams-Moulton fourth order numerical integration (two-body motion)
Initial orbital elements
semi-major axis = 6728 km
eccentricity = 0
argument of perigee = 0
inclination = 51.6 degrees
True-anomaly = 0 degrees
Right-ascension = 325.4 degrees
Geocentric inertial frame of reference

Time	X	Y	Z	Magnitude
(minutes)	(m)	(m)	(m)	(m)
1	5687556.762	-3575937.637	361043.0695	6728000
2	5810535.724	-3314186.724	720988.8765	6728000
3	5906147.633	-3036826.315	1077538.897	6728000
4	5973942.164	-2745162.752	1429013.814	6728000
5	6013600.014	-2440569.742	1773758.216	6728000
6	6024934.396	-2124481.887	2110148.391	6728000
7	6007891.928	-1798387.93	2436599.974	6728000
8	5962552.877	-1463823.743	2751575.41	6728000
9	5889130.787	-1122365.089	3053591.195	6728000
10	5787971.467	-775620.2053	3341224.864	6728000
11	5659551.369	-425222.2266	3613121.692	6728000
12	5504475.338	-72821.49287	3868001.071	6728000
13	5323473.768	279922.223	4104662.543	6728000
14	5117399.158	631347.5325	4321991.457	6728000
15	4887222.099	979799.2571	4518964.215	6728000
16	4634026.702	1323636.223	4694653.094	6728000
17	4359005.494	1661238.992	4848230.616	6728000
18	4063453.798	1991017.488	4978973.448	6728000
19	3748763.631	2311418.487	5086265.802	6728000
20	3416417.156	2620932.932	5169602.343	6728000
21	3067979.69	2918103.038	5228590.563	6728000
22	2705092.341	3201529.166	5262952.634	6728000
23	2329464.271	3469876.405	5272526.714	6728000
24	1942864.653	3721880.865	5257267.709	6728000
25	1547114.333	3956355.632	5217247.488	6728000
26	1144077.256	4172196.351	5152654.543	6728000
27	735651.6873	4368386.434	5063793.1	6728000
28	323761.2711	4544001.845	4951081.687	6728000
29	-89654.02897	4698215.452	4815051.165	6728000
30	-502647.0674	4830300.925	4656342.223	6728000
31	-913272.6873	4939636.154	4475702.366	6728000
32	-1319596.882	5025706.18	4273982.388	6728000
33	-1719705.905	5088105.623	4052132.373	6728000
34	-2111715.282	5126540.586	3811197.211	6728000
35	-2493778.687	5140830.046	3552311.684	6728000
36	-2864096.638	5130906.7	3276695.116	6728000
37	-3220924.975	5096817.286	2985645.636	6728000
38	-3562583.07	5038722.362	2680534.057	6728000
39	-3887461.749	4956895.549	2362797.425	6728000
40	-4194030.863	4851722.243	2033932.249	6728000

Time (minutes)	X (m)	Y (m)	Z (m)	Magnitude (m)
41	-4480846.503	4723697.801	1695487.452	6728000
42	-4746557.795	4573425.206	1349057.074	6728000
43	-4989913.264	4401612.225	996272.7701	6728000
44	-5209766.731	4209068.083	638796.1188	6728000
45	-5405082.707	3996699.642	278310.8004	6728000
46	-5574941.273	3765507.137	-83485.33465	6728000
47	-5718542.412	3516579.462	-444888.2619	6728000
48	-5835209.776	3251089.043	-804195.8091	6728000
49	-5924393.874	2970286.313	-1159715.673	6728000
50	-5985674.657	2675493.825	-1509773.389	6728000
51	-6018763.499	2368100.024	-1852720.221	6728000
52	-6023504.555	2049552.703	-2186940.922	6728000
53	-5999875.494	1721352.19	-2510861.345	6728000
54	-5947987.607	1385044.276	-2822955.858	6728000
55	-5868085.282	1042212.94	-3121754.525	6728000
56	-5760544.849	694472.8834	-3405850.034	6728000
57	-5625872.815	343461.9272	-3673904.323	6728000
58	-5464703.471	-9166.701063	-3924654.883	6728000
59	-5277795.909	-361752.155	-4156920.702	6728000
60	-5066030.445	-712633.7918	-4369607.832	6728000
61	-4830404.474	-1060158.993	-4561714.538	6728000
62	-4572027.771	-1402690.95	-4732336.015	6728000
63	-4292117.266	-1738616.369	-4880668.653	6728000
64	-3991991.308	-2066353.076	-5006013.819	6728000
65	-3673063.463	-2384357.462	-5107781.151	6728000
66	-3336835.848	-2691131.758	-5185491.335	6728000
67	-2984892.064	-2985231.087	-5238778.362	6728000
68	-2618889.73	-3265270.27	-5267391.256	6728000
69	-2240552.683	-3529930.351	-5271195.253	6728000
70	-1851662.851	-3777964.804	-5250172.437	6728000
71	-1454051.869	-4008205.414	-5204421.824	6728000
72	-1049592.445	-4219567.769	-5134158.893	6728000
73	-640189.544	-4411056.374	-5039714.577	6728000
74	-227771.4125	-4581769.334	-4921533.699	6728000
75	185719.5	-4730902.611	-4780172.879	6728000
76	598335.6918	-4857753.799	-4616297.914	6728000
77	1008133.781	-4961725.443	-4430680.637	6728000
78	1413183.659	-5042327.846	-4224195.289	6728000
79	1811577.581	-5099181.379	-3997814.395	6728000
80	2201439.149	-5132018.267	-3752604.187	6728000
81	2580932.155	-5140683.852	-3489719.58	6728000
82	2948269.223	-5125137.319	-3210398.737	6728000
83	3301720.232	-5085451.892	-2915957.229	6728000
84	3639620.462	-5021814.485	-2607781.849	6728000
85	3960378.436	-4934524.823	-2287324.072	6728000
86	4262483.416	-4823994.033	-1956093.223	6728000
87	4544512.517	-4690742.704	-1615649.366	6728000
88	4805137.409	-4535398.436	-1267595.958	6728000
89	5043130.574	-4358692.885	-913572.2985	6728000
90	5257371.088	-4161458.319	-555245.8025	6728000

Time	X	Y	Z	Magnitude
(minutes)	(m)	(m)	(m)	(m)
91	5446849.9	-3944623.691	-194304.1532	6728000
92	5610674.582	-3709210.272	167552.6496	6728000
93	5748073.537	-3456326.836	528620.2959	6728000
94	5858399.628	-3187164.439	887198.1924	6728000
95	5941133.231	-2902990.81	1241597.472	6728000
96	5995884.679	-2605144.379	1590148.95	6728000
97	6022396.097	-2295027.973	1931210.981	6728000
98	6020542.619	-1974102.21	2263177.198	6728000
99	5990332.976	-1643878.619	2584484.073	6728000
100	5931909.451	-1305912.521	2893618.28	6728000
101	5845547.214	-961795.7024	3189123.829	6728000
102	5731653.022	-613148.9205	3469608.917	6728000

Orbit Propagation - Kepler's prediction problem
<u>Initial orbital elements</u>
semi-major axis = 6728 km
eccentricity = 0
argument of perigee = 0 degrees
inclination = 51.6 degrees
True-anomaly = 0 degrees
Right-ascension = 325.4 degrees
Geocentric inertial frame of reference

Time	X	Y	Z	Magnitude
(minutes)	(m)	(m)	(m)	(m)
1	5687556.762	-3575937.637	361043.0695	6728000
2	5810535.724	-3314186.724	720988.8765	6728000
3	5906147.633	-3036826.315	1077538.897	6728000
4	5973942.164	-2745162.752	1429013.814	6728000
5	6013600.014	-2440569.742	1773758.216	6728000
6	6024934.396	-2124481.887	2110148.391	6728000
7	6007891.928	-1798387.93	2436599.974	6728000
8	5962552.877	-1463823.743	2751575.41	6728000
9	5889130.787	-1122365.089	3053591.195	6728000
10	5787971.467	-775620.2053	3341224.864	6728000
11	5659551.369	-425222.2266	3613121.692	6728000
12	5504475.338	-72821.49287	3868001.071	6728000
13	5323473.768	279922.223	4104662.543	6728000
14	5117399.158	631347.5325	4321991.457	6728000
15	4887222.099	979799.2571	4518964.215	6728000
16	4634026.702	1323636.223	4694653.094	6728000
17	4359005.494	1661238.992	4848230.616	6728000
18	4063453.798	1991017.488	4978973.448	6728000
19	3748763.631	2311418.487	5086265.802	6728000
20	3416417.156	2620932.932	5169602.343	6728000
21	3067979.69	2918103.038	5228590.563	6728000
22	2705092.341	3201529.166	5262952.634	6728000
23	2329464.271	3469876.405	5272526.714	6728000
24	1942864.653	3721880.865	5257267.709	6728000
25	1547114.333	3956355.632	5217247.488	6728000
26	1144077.256	4172196.351	5152654.543	6728000
27	735651.6873	4368386.434	5063793.1	6728000
28	323761.2711	4544001.845	4951081.687	6728000
29	-89654.02896	4698215.452	4815051.165	6728000
30	-502647.0673	4830300.925	4656342.223	6728000
31	-913272.6873	4939636.154	4475702.366	6728000
32	-1319596.882	5025706.18	4273982.388	6728000
33	-1719705.905	5088105.623	4052132.373	6728000
34	-2111715.282	5126540.586	3811197.211	6728000
35	-2493778.687	5140830.046	3552311.684	6728000
36	-2864096.638	5130906.7	3276695.116	6728000
37	-3220924.975	5096817.286	2985645.636	6728000
38	-3562583.07	5038722.362	2680534.057	6728000
39	-3887461.749	4956895.549	2362797.425	6728000
40	-4194030.863	4851722.243	2033932.249	6728000

Time (minutes)	X (m)	Y (m)	Z (m)	Magnitude (m)
41	-4480846.503	4723697.801	1695487.452	6728000
42	-4746557.795	4573425.206	1349057.074	6728000
43	-4989913.264	4401612.225	996272.7701	6728000
44	-5209766.731	4209068.083	638796.1188	6728000
45	-5405082.707	3996699.642	278310.8004	6728000
46	-5574941.273	3765507.137	-83485.33463	6728000
47	-5718542.412	3516579.462	-444888.2619	6728000
48	-5835209.776	3251089.043	-804195.8091	6728000
49	-5924393.874	2970286.313	-1159715.673	6728000
50	-5985674.657	2675493.825	-1509773.389	6728000
51	-6018763.499	2368100.024	-1852720.221	6728000
52	-6023504.555	2049552.703	-2186940.922	6728000
53	-5999875.494	1721352.19	-2510861.345	6728000
54	-5947987.607	1385044.277	-2822955.858	6728000
55	-5868085.282	1042212.94	-3121754.525	6728000
56	-5760544.849	694472.8834	-3405850.034	6728000
57	-5625872.815	343461.9272	-3673904.323	6728000
58	-5464703.471	-9166.701048	-3924654.883	6728000
59	-5277795.909	-361752.155	-4156920.702	6728000
60	-5066030.445	-712633.7918	-4369607.832	6728000
61	-4830404.474	-1060158.993	-4561714.538	6728000
62	-4572027.771	-1402690.95	-4732336.015	6728000
63	-4292117.266	-1738616.369	-4880668.653	6728000
64	-3991991.308	-2066353.076	-5006013.819	6728000
65	-3673063.463	-2384357.462	-5107781.151	6728000
66	-3336835.848	-2691131.758	-5185491.335	6728000
67	-2984892.064	-2985231.087	-5238778.362	6728000
68	-2618889.73	-3265270.27	-5267391.256	6728000
69	-2240552.683	-3529930.351	-5271195.253	6728000
70	-1851662.851	-3777964.804	-5250172.437	6728000
71	-1454051.869	-4008205.414	-5204421.824	6728000
72	-1049592.445	-4219567.769	-5134158.893	6728000
73	-640189.544	-4411056.374	-5039714.577	6728000
74	-227771.4126	-4581769.334	-4921533.699	6728000
75	185719.4999	-4730902.611	-4780172.879	6728000
76	598335.6918	-4857753.799	-4616297.914	6728000
77	1008133.781	-4961725.443	-4430680.637	6728000
78	1413183.659	-5042327.846	-4224195.289	6728000
79	1811577.581	-5099181.379	-3997814.395	6728000
80	2201439.149	-5132018.267	-3752604.187	6728000
81	2580932.155	-5140683.852	-3489719.58	6728000
82	2948269.223	-5125137.319	-3210398.737	6728000
83	3301720.232	-5085451.892	-2915957.229	6728000
84	3639620.462	-5021814.485	-2607781.849	6728000
85	3960378.436	-4934524.823	-2287324.072	6728000
86	4262483.416	-4823994.033	-1956093.223	6728000
87	4544512.517	-4690742.704	-1615649.366	6728000
88	4805137.409	-4535398.436	-1267595.958	6728000
89	5043130.574	-4358692.885	-913572.2985	6728000
90	5257371.088	-4161458.319	-555245.8026	6728000

Time (minutes)	X (m)	Y (m)	Z (m)	Magnitude (m)
91	5446849.9	-3944623.691	-194304.1533	6728000
92	5610674.582	-3709210.272	167552.6496	6728000
93	5748073.537	-3456326.836	528620.2959	6728000
94	5858399.628	-3187164.439	887198.1923	6728000
95	5941133.231	-2902990.81	1241597.472	6728000
96	5995884.679	-2605144.379	1590148.95	6728000
97	6022396.097	-2295027.973	1931210.981	6728000
98	6020542.619	-1974102.21	2263177.198	6728000
99	5990332.976	-1643878.619	2584484.073	6728000
100	5931909.451	-1305912.521	2893618.28	6728000
101	5845547.214	-961795.7024	3189123.829	6728000
102	5731653.022	-613148.9206	3469608.917	6728000

Orbit Propagation - Adams-Moulton fourth order numerical integration (J2 perturbation)
<u>Initial orbital elements</u>
semi-major axis = 6728 km
eccentricity = 0
argument of perigee = 0
inclination = 51.6 degrees
True-anomaly = 0 degrees
Right-ascension = 325.4 degrees
Geocentric inertial frame of reference

Time	X	Y	Z	Magnitude
(minutes)	(m)	(m)	(m)	(m)
1	5687537.636	-3575924.84	361041.8328	6727976.964
2	5810458.821	-3314136.825	720978.9938	6727907.945
3	5905974.606	-3036717.461	1077505.714	6727793.662
4	5973635.938	-2744975.957	1428935.782	6727635.305
5	6013125.703	-2440289.172	1773607.392	6727434.514
6	6024260.162	-2124095.012	2109891.076	6727193.358
7	6006989.723	-1797885.536	2436197.46	6726914.3
8	5961399.054	-1463199.816	2750984.805	6726600.165
9	5887706.546	-1121616.588	3052766.286	6726254.095
10	5786263.134	-774746.7392	3340116.997	6725879.505
11	5657550.478	-424225.6649	3611680.615	6725480.034
12	5502178.545	-71705.53856	3866175.728	6725059.493
13	5320882.579	281152.467	4102401.777	6724621.809
14	5114519.51	632685.97	4319244.595	6724170.97
15	4884063.809	981239.1461	4515681.531	6723710.973
16	4630602.819	1325170.435	4690786.127	6723245.767
17	4355331.592	1662860.157	4843732.346	6722779.203
18	4059547.254	1992718.005	4973798.325	6722314.983
19	3744642.932	2313190.382	5080369.661	6721856.612
20	3412101.267	2622767.569	5162942.2	6721407.362
21	3063487.545	2919990.67	5221124.341	6720970.231
22	2700442.472	3203458.342	5254638.839	6720547.911
23	2324674.618	3471833.258	5263324.099	6720142.768
24	1937952.575	3723848.296	5247134.97	6719756.819
25	1542096.836	3958312.426	5206143.027	6719391.723
26	1138971.436	4174116.272	5140536.339	6719048.774
27	730475.3858	4370237.328	5050618.725	6718728.905
28	318533.9286	4545744.808	4936808.494	6718432.691
29	-94910.35765	4699804.103	4799636.675	6718160.373
30	-507906.565	4831680.832	4639744.729	6717911.87
31	-918504.543	4940744.451	4457881.754	6717686.809
32	-1324764.028	5026471.42	4254901.187	6717484.557
33	-1724763.774	5088447.881	4031757	6717304.253
34	-2116610.655	5126371.86	3789499.415	6717144.849
35	-2498448.679	5140054.934	3529270.132	6717005.151
36	-2868467.884	5129423.392	3252297.101	6716883.859
37	-3224913.067	5094518.834	2959888.839	6716779.614
38	-3566092.298	5035498.22	2653428.329	6716691.038

Time	X	Y	Z	Magnitude
(minutes)	(m)	(m)	(m)	(m)
39	-3890385.165	4952633.359	2334366.518	6716616.779
40	-4196250.718	4846309.816	2004215.437	6716555.549
41	-4482235.051	4717025.255	1664540.991	6716506.162
42	-4746978.474	4565387.204	1316955.436	6716467.567
43	-4989222.244	4392110.257	963109.5978	6716438.879
44	-5207814.801	4198012.72	604684.8651	6716419.402
45	-5401717.465	3984012.715	243385.0048	6716408.649
46	-5570009.573	3751123.768	-119072.1538	6716406.356
47	-5711893.008	3500449.891	-480963.1182	6716412.487
48	-5826696.098	3233180.206	-840567.1647	6716427.236
49	-5913876.864	2950583.12	-1196174.755	6716451.024
50	-5973025.598	2654000.102	-1546095.885	6716484.484
51	-6003866.749	2344839.102	-1888668.321	6716528.447
52	-6006260.128	2024567.628	-2222265.674	6716583.918
53	-5980201.41	1694705.561	-2545305.261	6716652.047
54	-5925821.957	1356817.719	-2856255.723	6716734.098
55	-5843387.947	1012506.226	-3153644.348	6716831.413
56	-5733298.848	663402.7396	-3436064.067	6716945.373
57	-5596085.228	311160.5667	-3702180.087	6717077.354
58	-5432405.95	-42553.28427	-3950736.139	6717228.687
59	-5243044.746	-396066.0688	-4180560.303	6717400.612
60	-5028906.225	-747707.2545	-4390570.395	6717594.239
61	-4791011.328	-1095816.411	-4579778.899	6717810.5
62	-4530492.259	-1438751.006	-4747297.421	6718050.117
63	-4248586.937	-1774894.075	-4892340.659	6718313.558
64	-3946632.991	-2102661.729	-5014229.873	6718601.01
65	-3626061.329	-2420510.477	-5112395.861	6718912.35
66	-3288389.326	-2726944.327	-5186381.412	6719247.121
67	-2935213.646	-3020521.647	-5235843.262	6719604.514
68	-2568202.747	-3299861.764	-5260553.523	6719983.362
69	-2189089.091	-3563651.266	-5260400.612	6720382.132
70	-1799661.088	-3810649.998	-5235389.649	6720798.927
71	-1401754.819	-4039696.729	-5185642.356	6721231.5
72	-997245.5473	-4249714.462	-5111396.442	6721677.267
73	-588039.0745	-4439715.383	-5013004.473	6722133.328
74	-176062.9514	-4608805.413	-4890932.244	6722596.501
75	236742.4119	-4756188.356	-4745756.653	6723063.353
76	648432.7064	-4881169.62	-4578163.067	6723530.241
77	1057068.722	-4983159.501	-4388942.215	6723993.357
78	1460725.347	-5061675.996	-4178986.586	6724448.774
79	1857500.527	-5116347.149	-3949286.368	6724892.502
80	2245524.156	-5146912.897	-3700924.917	6725320.538
81	2622966.851	-5153226.415	-3435073.79	6725728.922
82	2988048.566	-5135254.941	-3152987.344	6726113.791
83	3339047.02	-5093080.064	-2855996.926	6726471.436
84	3674305.876	-5026897.487	-2545504.677	6726798.352
85	3992242.642	-4937016.232	-2222976.98	6727091.291
86	4291356.245	-4823857.312	-1889937.567	6727347.307

Time (minutes)	X (m)	Y (m)	Z (m)	Magnitude (m)
87	4570234.24	-4687951.847	-1547960.33	6727563.802
88	4827559.609	-4529938.641	-1198661.863	6727738.56
89	5062117.105	-4350561.233	-843693.7808	6727869.783
90	5272799.122	-4150664.414	-484734.8354	6727956.118
91	5458611.035	-3931190.249	-123482.8977	6727996.672
92	5618675.987	-3693173.603	238353.1742	6727991.033
93	5752239.105	-3437737.212	599061.7248	6727939.269
94	5858671.098	-3166086.313	956936.5059	6727841.931
95	5937471.24	-2879502.862	1310284.852	6727700.042
96	5988269.714	-2579339.388	1657435.782	6727515.085
97	6010829.297	-2267012.497	1996747.979	6727288.977
98	6005046.399	-1943996.079	2326617.61	6727024.046
99	5970951.44	-1611814.251	2645485.945	6726722.988
100	5908708.573	-1272034.072	2951846.729	6726388.838
101	5818614.765	-926258.0744	3244253.282	6726024.915
102	5701098.235	-576116.6538	3521325.284	6725634.783

Orbit Propagation - Satellite Tool Kit (two-body motion)

<u>Initial orbital elements</u>

semi-major axis = 6728 km
eccentricity = 0
argument of perigee = 0 degrees
inclination = 51.6 degrees
True-anomaly = 0 degrees
Right-ascension = 325.4 degrees
Geocentric inertial frame of reference

Time	X	Y	Z	Magnitude
(minutes)	(km)	(km)	(km)	(km)
0	5538.06139	-3820.452858	-0.000041	6728
1	5687.784018	-3575.51537	361.644589	6728
2	5810.717774	-3313.737576	721.58591	6728
3	5906.283655	-3036.352423	1078.128636	6728
4	5974.031553	-2744.666366	1429.593489	6728
5	6013.642384	-2440.053215	1774.325105	6728
6	6024.929586	-2123.947668	2110.699834	6728
7	6007.839995	-1797.838549	2437.133387	6728
8	5962.454103	-1463.261796	2752.088295	6728
9	5888.985673	-1121.793233	3054.081153	6728
10	5787.780732	-775.041141	3341.689607	6728
11	5659.315946	-424.638685	3613.55905	6728
12	5504.196371	-72.236225	3868.409006	6728
13	5323.152603	280.50646	4105.039158	6728
14	5117.03734	631.927988	4322.335003	6728
15	4886.821364	980.373202	4519.273099	6728
16	4633.588967	1324.20096	4694.925889	6728
17	4358.532848	1661.791869	4848.466066	6728
18	4062.948493	1991.555911	4979.170473	6728
19	3748.228073	2311.939932	5086.423504	6728
20	3415.853889	2621.434956	5169.72001	6728
21	3067.391388	2918.583294	5228.667672	6728
22	2704.481793	3201.985409	5262.988854	6728
23	2328.834369	3470.306505	5272.521905	6728
24	1942.218379	3722.282819	5257.221927	6728
25	1546.454741	3956.727568	5217.16098	6728
26	1143.407463	4172.536541	5152.527747	6728
27	734.974854	4368.693301	5063.626645	6728
28	323.08059	4544.273971	4950.876388	6728
29	-90.335349	4698.451582	4814.808018	6728
30	-503.325819	4830.499976	4656.062404	6728
31	-913.945676	4939.797216	4475.387219	6728
32	-1320.260945	5025.828526	4273.633426	6728
33	-1720.357921	5088.188706	4051.751265	6728
34	-2112.352191	5126.584047	3810.785776	6728
35	-2494.397501	5140.83371	3551.871883	6728
36	-2864.694459	5130.870581	3276.229042	6728
37	-3221.499004	5096.741585	2985.155504	6728
38	-3563.130624	5038.607466	2680.022193	6728
39	-3887.980269	4956.742031	2362.266257	6728

Time	X	Y	Z	Magnitude
(minutes)	(km)	(km)	(km)	(km)
40	-4194.517931	4851.530856	2033.384293	6728
41	-4481.29985	4723.469475	1694.9253	6728
42	-4746.975313	4573.161044	1348.483385	6728
43	-4990.293016	4401.313501	995.690253	6728
44	-5210.106958	4208.736228	638.207524	6728
45	-5405.381838	3996.336246	277.718904	6728
46	-5575.197932	3765.113936	-84.077743	6728
47	-5718.755424	3516.158331	-445.478392	6728
48	-5835.378172	3250.641985	-804.780885	6728
49	-5924.516896	2969.815454	-1160.292944	6728
50	-5985.751762	2675.001401	-1510.340143	6728
51	-6018.794359	2367.58837	-1853.273798	6728
52	-6023.489061	2049.024244	-2187.478725	6728
53	-5999.813757	1720.809427	-2511.380856	6728
54	-5947.879953	1384.489776	-2823.454644	6728
55	-5867.932253	1041.649323	-3122.230254	6728
56	-5760.347203	693.902809	-3406.300484	6728
57	-5625.631517	342.888086	-3674.327393	6728
58	-5464.419692	-9.741605	-3925.048603	6728
59	-5277.470945	-362.325501	-4157.283327	6728
60	-5065.665919	-713.202786	-4369.937595	6728
61	-4830.002134	-1060.720961	-4562.009911	6728
62	-4571.589541	-1403.243251	-4732.595636	6728
63	-4291.645237	-1739.156411	-4880.891327	6728
64	-3991.48773	-2066.878325	-5006.198527	6728
65	-3672.530731	-2384.865457	-5107.927053	6728
66	-3336.276494	-2691.620121	-5185.597775	6728
67	-2984.308742	-2985.697534	-5238.84487	6728
68	-2618.285206	-3265.712622	-5267.41755	6728
69	-2239.92982	-3530.346543	-5271.181242	6728
70	-1851.024597	-3778.3529	-5250.118218	6728
71	-1453.40124	-4008.563607	-5204.327684	6728
72	-1048.932515	-4219.894397	-5134.025308	6728
73	-639.523427	-4411.349923	-5039.542206	6728
74	-227.10225	-4582.028449	-4921.323385	6728
75	186.388555	-4731.126099	-4779.925642	6728
76	599.001489	-4857.940636	-4616.014947	6728
77	1008.793189	-4961.874778	-4430.363302	6728
78	1413.833578	-5042.439006	-4223.845106	6728
79	1812.214957	-5099.253871	-3997.433039	6728
80	2202.060993	-5132.051781	-3752.193479	6728
81	2581.53555	-5140.67826	-3489.281477	6728
82	2948.851343	-5125.09268	-3209.935323	6728
83	3302.278354	-5085.368446	-2915.470707	6728
84	3640.151976	-5021.692656	-2607.274527	6728
85	3960.880862	-4934.365216	-2286.798355	6728
86	4262.95441	-4823.797428	-1955.551601	6728
87	4544.949887	-4690.510057	-1615.094403	6728
88	4805.539123	-4535.130871	-1267.030278	6728
89	5043.494769	-4358.39169	-912.998572	6728

Time	X	Y	Z	Magnitude
(minutes)	(km)	(km)	(km)	(km)
90	5257.696079	-4161.124938	-554.666737	6728
91	5447.134188	-3944.25972	-193.72248	6728

Orbit Propagation - Satellite Tool Kit (J2 Perturbation)

Initial orbital elements

semi-major axis = 6728 km

eccentricity = 0

argument of perigee = 0 degrees

inclination = 51.6 degrees

True-anomaly = 0 degrees

Right-ascension = 325.4 degrees

Geocentric inertial frame of reference

Time	X	Y	Z	Magnitude
(minutes)	(km)	(km)	(km)	(km)
0	5538.06139	-3820.452858	-0.000041	6728
1	5687.669687	-3575.668266	361.930918	6728
2	5810.478703	-3314.032905	722.154517	6728
3	5905.911125	-3036.777634	1078.971448	6728
4	5973.518626	-2745.20693	1430.698477	6728
5	6012.983975	-2440.692755	1775.676377	6728
6	6024.122519	-2124.668098	2112.277763	6728
7	6006.883056	-1798.620233	2438.914761	6728
8	5961.348061	-1464.083719	2754.046506	6728
9	5887.733299	-1122.633176	3056.186407	6728
10	5786.386797	-775.875878	3343.909158	6728
11	5657.787203	-425.444183	3615.857467	6728
12	5502.541521	-72.987849	3870.748453	6728
13	5321.382251	279.833731	4107.379702	6728
14	5115.163929	631.359342	4324.634939	6728
15	4884.859097	979.933769	4521.489291	6728
16	4631.553716	1323.91559	4697.014124	6728
17	4356.442048	1661.684907	4850.381423	6728
18	4060.821021	1991.650973	4980.867699	6728
19	3746.084119	2312.259686	5087.857399	6728
20	3413.714811	2622.000904	5170.845815	6728
21	3065.27956	2919.415563	5229.44146	6728
22	2702.42044	3203.102549	5263.367916	6728
23	2326.847396	3471.725296	5272.465141	6728
24	1940.33019	3724.018086	5256.690218	6728
25	1544.690055	3958.79201	5216.117565	6728
26	1141.791121	4174.940575	5150.938578	6728
27	733.531625	4371.444916	5061.460728	6728
28	321.834973	4547.378597	4948.106117	6728
29	-91.359325	4701.911983	4811.409479	6728
30	-504.10479	4834.31615	4652.015663	6728
31	-914.457148	4943.966321	4470.676586	6728
32	-1320.48348	5030.34482	4268.247693	6728
33	-1720.271336	5093.04351	4045.683915	6728
34	-2111.937738	5131.765725	3804.035165	6728
35	-2493.638049	5146.327673	3544.44139	6728
36	-2863.574662	5136.659307	3268.127188	6728
37	-3220.005466	5102.804665	2976.396034	6728

Time (minutes)	X (km)	Y (km)	Z (km)	Magnitude (km)
38	-3561.252048	5044.921664	2670.624132	6728
39	-3885.707597	4963.281371	2352.253919	6728
40	-4191.844474	4858.266727	2022.787263	6728
41	-4478.221398	4730.370755	1683.77838	6728
42	-4743.490236	4580.194242	1336.826499	6728
43	-4986.402348	4408.442921	983.568318	6728
44	-5205.814465	4215.924151	625.670285	6728
45	-5400.69407	4003.543124	264.820736	6728
46	-5570.124254	3772.298607	-97.278068	6728
47	-5713.308034	3523.278246	-458.917977	6728
48	-5829.5721	3257.653454	-818.393001	6728
49	-5918.369982	2976.673893	-1174.007366	6728
50	-5979.284616	2681.6616	-1524.083509	6728
51	-6012.030307	2374.004768	-1866.969993	6728
52	-6016.454064	2055.15121	-2201.049296	6728
53	-5992.53632	1726.601546	-2524.745445	6728
54	-5940.391013	1389.902144	-2836.531445	6728
55	-5860.265049	1046.637838	-3134.936489	6728
56	-5752.537126	698.424473	-3418.552891	6728
57	-5617.71595	346.901292	-3686.042729	6728
58	-5456.437831	-6.276775	-3936.144154	6728
59	-5269.463605	-359.446992	-4167.677432	6728
60	-5057.675326	-710.94629	-4379.550168	6728
61	-4822.071704	-1059.119639	-4570.762963	6728
62	-4563.763653	-1402.327476	-4740.413798	6728
63	-4283.968984	-1738.953514	-4887.702366	6728
64	-3984.006674	-2067.412359	-5011.933855	6728
65	-3665.290639	-2386.156974	-5112.522218	6728
66	-3329.323077	-2693.685966	-5188.992943	6728
67	-2977.687382	-2988.550658	-5240.98529	6728
68	-2612.040685	-3269.361914	-5268.253993	6728
69	-2234.10604	-3534.796682	-5270.670414	6728
70	-1845.664312	-3783.604229	-5248.223155	6728
71	-1448.545777	-4014.612033	-5201.018108	6728
72	-1044.621505	-4226.73131	-5129.277955	6728
73	-635.79454	-4418.962147	-5033.341122	6728
74	-223.990933	-4590.39821	-4913.660177	6728
75	188.849326	-4740.231023	-4770.7997	6728
76	600.781454	-4867.753779	-4605.433616	6728
77	1009.865031	-4972.364673	-4418.342016	6728
78	1414.173146	-5053.569744	-4210.407481	6728
79	1811.801468	-5110.985204	-3982.610914	6728
80	2200.877217	-5144.339255	-3736.026914	6728
81	2579.567988	-5153.473378	-3471.818708	6728
82	2946.090376	-5138.343078	-3191.232663	6728
83	3298.718378	-5099.018109	-2895.592407	6728
84	3635.791521	-5035.682148	-2586.292582	6728
85	3955.722678	-4948.63194	-2264.792269	6728

Time (minutes)	X (km)	Y (km)	Z (km)	Magnitude (km)
86	4257.005544	-4838.275904	-1932.608102	6728
87	4538.221729	-4705.13222	-1591.307115	6728
88	4798.04743	-4549.826395	-1242.499351	6728
89	5035.259669	-4373.088324	-887.830263	6728
90	5248.742045	-4175.748861	-528.972955	6728
91	5437.489993	-3958.735912	-167.620288	6728

Orbit Propagation - NPOE, Osculating Elements (two-body motion)

Initial orbital elements

semi-major axis = 6728 km

eccentricity = 0

argument of perigee = 0

inclination = 51.6 degrees

True-anomaly = 0 degrees

Right-ascension = 325.4 degrees

Geocentric inertial frame of reference

Time (minutes)	X (km)	Y (km)	Z (km)	Magnitude (km)
0	5.538061D+03	-3.820453D+03	0.000000D+00	6.728000D+03
1	5.687784D+03	-3.575515D+03	3.616449D+02	6.728000D+03
2	5.810718D+03	-3.313737D+03	7.215864D+02	6.728000D+03
3	5.906284D+03	-3.036352D+03	1.078129D+03	6.728000D+03
4	5.974032D+03	-2.744665D+03	1.429594D+03	6.728000D+03
5	6.013642D+03	-2.440052D+03	1.774326D+03	6.728000D+03
6	6.024930D+03	-2.123946D+03	2.110701D+03	6.728000D+03
7	6.007840D+03	-1.797837D+03	2.437135D+03	6.728000D+03
8	5.962454D+03	-1.463260D+03	2.752090D+03	6.728000D+03
9	5.888985D+03	-1.121791D+03	3.054083D+03	6.728000D+03
10	5.787780D+03	-7.750388D+02	3.341691D+03	6.728000D+03
11	5.659315D+03	-4.246362D+02	3.613561D+03	6.728000D+03
12	5.504195D+03	-7.223347D+01	3.868411D+03	6.728000D+03
13	5.323151D+03	2.805094D+02	4.105041D+03	6.728000D+03
14	5.117035D+03	6.319311D+02	4.322337D+03	6.728000D+03
15	4.886819D+03	9.803765D+02	4.519275D+03	6.728000D+03
16	4.633586D+03	1.324204D+03	4.694928D+03	6.728000D+03
17	4.358530D+03	1.661795D+03	4.848468D+03	6.728000D+03
18	4.062945D+03	1.991560D+03	4.979172D+03	6.728000D+03
19	3.748224D+03	2.311944D+03	5.086425D+03	6.728000D+03
20	3.415849D+03	2.621439D+03	5.169721D+03	6.728000D+03
21	3.067387D+03	2.918587D+03	5.228668D+03	6.728000D+03
22	2.704477D+03	3.201989D+03	5.262989D+03	6.728000D+03
23	2.328829D+03	3.470310D+03	5.272522D+03	6.728000D+03
24	1.942212D+03	3.722286D+03	5.257222D+03	6.728000D+03
25	1.546448D+03	3.956731D+03	5.217160D+03	6.728000D+03
26	1.143401D+03	4.172540D+03	5.152527D+03	6.728000D+03
27	7.349677D+02	4.368696D+03	5.063625D+03	6.728000D+03
28	3.230731D+02	4.544277D+03	4.950874D+03	6.728000D+03
29	-9.034306D+01	4.698454D+03	4.814805D+03	6.728000D+03
30	-5.033338D+02	4.830502D+03	4.656059D+03	6.728000D+03
31	-9.139538D+02	4.939799D+03	4.475384D+03	6.728000D+03
32	-1.320269D+03	5.025830D+03	4.273629D+03	6.728000D+03
33	-1.720366D+03	5.088190D+03	4.051747D+03	6.728000D+03
34	-2.112361D+03	5.126585D+03	3.810780D+03	6.728000D+03
35	-2.494406D+03	5.140834D+03	3.551866D+03	6.728000D+03
36	-2.864703D+03	5.130870D+03	3.276223D+03	6.728000D+03
37	-3.221507D+03	5.096740D+03	2.985149D+03	6.728000D+03

Time (minutes)	X (km)	Y (km)	Z (km)	Magnitude (km)
38	-3.563139D+03	5.038606D+03	2.680015D+03	6.728000D+03
39	-3.887988D+03	4.956740D+03	2.362258D+03	6.728000D+03
40	-4.194526D+03	4.851528D+03	2.033376D+03	6.728000D+03
41	-4.481307D+03	4.723466D+03	1.694916D+03	6.728000D+03
42	-4.746982D+03	4.573157D+03	1.348474D+03	6.728000D+03
43	-4.990299D+03	4.401308D+03	9.956806D+02	6.728000D+03
44	-5.210113D+03	4.208730D+03	6.381976D+02	6.728000D+03
45	-5.405387D+03	3.996330D+03	2.777087D+02	6.728000D+03
46	-5.575203D+03	3.765107D+03	-8.408823D+01	6.728000D+03
47	-5.718759D+03	3.516151D+03	-4.454891D+02	6.728000D+03
48	-5.835381D+03	3.250634D+03	-8.047917D+02	6.728000D+03
49	-5.924519D+03	2.969806D+03	-1.160304D+03	6.728000D+03
50	-5.985753D+03	2.674992D+03	-1.510351D+03	6.728000D+03
51	-6.018795D+03	2.367578D+03	-1.853285D+03	6.728000D+03
52	-6.023489D+03	2.049014D+03	-2.187490D+03	6.728000D+03
53	-5.999813D+03	1.720798D+03	-2.511392D+03	6.728000D+03
54	-5.947878D+03	1.384478D+03	-2.823465D+03	6.728000D+03
55	-5.867929D+03	1.041637D+03	-3.122240D+03	6.728000D+03
56	-5.760343D+03	6.938904D+02	-3.406310D+03	6.728000D+03
57	-5.625626D+03	3.428754D+02	-3.674337D+03	6.728000D+03
58	-5.464413D+03	-9.754536D+00	-3.925057D+03	6.728000D+03
59	-5.277464D+03	-3.623385D+02	-4.157292D+03	6.728000D+03
60	-5.065657D+03	-7.132159D+02	-4.369945D+03	6.728000D+03
61	-4.829993D+03	-1.060734D+03	-4.562017D+03	6.728000D+03
62	-4.571579D+03	-1.403256D+03	-4.732602D+03	6.728000D+03
63	-4.291634D+03	-1.739170D+03	-4.880897D+03	6.728000D+03
64	-3.991475D+03	-2.066891D+03	-5.006203D+03	6.728000D+03
65	-3.672517D+03	-2.384878D+03	-5.107931D+03	6.728000D+03
66	-3.336262D+03	-2.691633D+03	-5.185600D+03	6.728000D+03
67	-2.984294D+03	-2.985710D+03	-5.238847D+03	6.728000D+03
68	-2.618269D+03	-3.265724D+03	-5.267418D+03	6.728000D+03
69	-2.239913D+03	-3.530358D+03	-5.271181D+03	6.728000D+03
70	-1.851007D+03	-3.778363D+03	-5.250117D+03	6.728000D+03
71	-1.453383D+03	-4.008573D+03	-5.204325D+03	6.728000D+03
72	-1.048914D+03	-4.219903D+03	-5.134022D+03	6.728000D+03
73	-6.395046D+02	-4.411358D+03	-5.039537D+03	6.728000D+03
74	-2.270831D+02	-4.582036D+03	-4.921317D+03	6.728000D+03
75	1.864080D+02	-4.731133D+03	-4.779918D+03	6.728000D+03
76	5.990211D+02	-4.857946D+03	-4.616007D+03	6.728000D+03
77	1.008813D+03	-4.961879D+03	-4.430354D+03	6.728000D+03
78	1.413853D+03	-5.042442D+03	-4.223835D+03	6.728000D+03
79	1.812235D+03	-5.099256D+03	-3.997421D+03	6.728000D+03
80	2.202080D+03	-5.132053D+03	-3.752181D+03	6.728000D+03
81	2.581555D+03	-5.140678D+03	-3.489268D+03	6.728000D+03
82	2.948870D+03	-5.125091D+03	-3.209921D+03	6.728000D+03
83	3.302296D+03	-5.085366D+03	-2.915455D+03	6.728000D+03
84	3.640169D+03	-5.021689D+03	-2.607258D+03	6.728000D+03
85	3.960897D+03	-4.934360D+03	-2.286781D+03	6.728000D+03

Time	X	Y	Z	Magnitude
(minutes)	(km)	(km)	(km)	(km)
86	4.262970D+03	-4.823791D+03	-1.955533D+03	6.728000D+03
87	4.544965D+03	-4.690502D+03	-1.615076D+03	6.728000D+03
88	4.805553D+03	-4.535122D+03	-1.267011D+03	6.728000D+03
89	5.043507D+03	-4.358381D+03	-9.129787D+02	6.728000D+03
90	5.257707D+03	-4.161113D+03	-5.546464D+02	6.728000D+03
91	5.447144D+03	-3.944247D+03	-1.937018D+02	6.728000D+03
92	5.538061D+03	-3.820453D+03	3.025276D-08	6.728000D+03

Orbit Propagation - NPOE, Mean Elements (two-body motion)
Initial orbital elements
semi-major axis = 6728 km
eccentricity = 0
argument of perigee = 0 degrees
inclination = 51.6 degrees
True-anomaly = 0 degrees
Right-ascension = 325.4 degrees
Geocentric inertial frame of reference

Time (minutes)	X (km)	Y (km)	Z (km)	Magnitude (km)
0	5.538066D+03	-3.820456D+03	-6.685526D-11	6.728005D+03
1	5.687694D+03	-3.575683D+03	3.619316D+02	6.728028D+03
2	5.810561D+03	-3.314083D+03	7.221632D+02	6.728097D+03
3	5.906090D+03	-3.036886D+03	1.079003D+03	6.728211D+03
4	5.973830D+03	-2.745392D+03	1.430774D+03	6.728368D+03
5	6.013463D+03	-2.440970D+03	1.775824D+03	6.728568D+03
6	6.024801D+03	-2.125051D+03	2.112531D+03	6.728808D+03
7	6.007789D+03	-1.799117D+03	2.439312D+03	6.729085D+03
8	5.962504D+03	-1.464702D+03	2.754631D+03	6.729398D+03
9	5.889159D+03	-1.123375D+03	3.057004D+03	6.729743D+03
10	5.788095D+03	-7.767427D+02	3.345009D+03	6.730116D+03
11	5.659787D+03	-4.264342D+02	3.617290D+03	6.730514D+03
12	5.504837D+03	-7.409753D+01	3.872564D+03	6.730934D+03
13	5.323971D+03	2.786092D+02	4.109630D+03	6.731370D+03
14	5.118040D+03	6.300259D+02	4.327370D+03	6.731820D+03
15	4.888013D+03	9.784979D+02	4.524760D+03	6.732279D+03
16	4.634973D+03	1.322384D+03	4.700869D+03	6.732744D+03
17	4.360110D+03	1.660066D+03	4.854867D+03	6.733210D+03
18	4.064722D+03	1.989951D+03	4.986029D+03	6.733673D+03
19	3.750198D+03	2.310487D+03	5.093740D+03	6.734131D+03
20	3.418024D+03	2.620164D+03	5.177492D+03	6.734580D+03
21	3.069765D+03	2.917525D+03	5.236893D+03	6.735017D+03
22	2.707063D+03	3.201169D+03	5.271667D+03	6.735439D+03
23	2.331629D+03	3.469763D+03	5.281652D+03	6.735844D+03
24	1.945234D+03	3.722044D+03	5.266806D+03	6.736230D+03
25	1.549699D+03	3.956828D+03	5.227204D+03	6.736595D+03
26	1.146888D+03	4.173012D+03	5.163038D+03	6.736939D+03
27	7.386984D+02	4.369584D+03	5.074615D+03	6.737259D+03
28	3.270523D+02	4.545625D+03	4.962357D+03	6.737556D+03
29	-8.611357D+01	4.700312D+03	4.826800D+03	6.737829D+03
30	-4.988564D+02	4.832923D+03	4.668586D+03	6.738079D+03
31	-9.092369D+02	4.942843D+03	4.488468D+03	6.738305D+03
32	-1.315328D+03	5.029563D+03	4.287296D+03	6.738509D+03
33	-1.715226D+03	5.092682D+03	4.066024D+03	6.738691D+03
34	-2.107054D+03	5.131912D+03	3.825694D+03	6.738852D+03
35	-2.488980D+03	5.147077D+03	3.567440D+03	6.738994D+03
36	-2.859214D+03	5.138112D+03	3.292478D+03	6.739118D+03
37	-3.216026D+03	5.105067D+03	3.002102D+03	6.739225D+03

Time	X	Y	Z	Magnitude
(minutes)	(km)	(km)	(km)	(km)
38	-3.557749D+03	5.048104D+03	2.697675D+03	6.739316D+03
39	-3.882786D+03	4.967494D+03	2.380627D+03	6.739393D+03
40	-4.189622D+03	4.863622D+03	2.052444D+03	6.739456D+03
41	-4.476823D+03	4.736977D+03	1.714663D+03	6.739508D+03
42	-4.743052D+03	4.588157D+03	1.368865D+03	6.739548D+03
43	-4.987066D+03	4.417859D+03	1.016669D+03	6.739579D+03
44	-5.207727D+03	4.226884D+03	6.597221D+02	6.739599D+03
45	-5.404007D+03	4.016124D+03	2.996918D+02	6.739611D+03
46	-5.574989D+03	3.786566D+03	-6.173885D+01	6.739614D+03
47	-5.719874D+03	3.539283D+03	-4.228810D+02	6.739608D+03
48	-5.837986D+03	3.275430D+03	-7.820472D+02	6.739593D+03
49	-5.928770D+03	2.996238D+03	-1.137559D+03	6.739569D+03
50	-5.991799D+03	2.703012D+03	-1.487753D+03	6.739536D+03
51	-6.026776D+03	2.397120D+03	-1.830993D+03	6.739492D+03
52	-6.033531D+03	2.079991D+03	-2.165673D+03	6.739436D+03
53	-6.012029D+03	1.753105D+03	-2.490224D+03	6.739368D+03
54	-5.962363D+03	1.417991D+03	-2.803126D+03	6.739287D+03
55	-5.884758D+03	1.076215D+03	-3.102913D+03	6.739190D+03
56	-5.779572D+03	7.293761D+02	-3.388175D+03	6.739078D+03
57	-5.647289D+03	3.790974D+02	-3.657572D+03	6.738948D+03
58	-5.488520D+03	2.702000D+01	-3.909838D+03	6.738800D+03
59	-5.304003D+03	-3.252054D+02	-4.143783D+03	6.738631D+03
60	-5.094595D+03	-6.759262D+02	-4.358305D+03	6.738442D+03
61	-4.861271D+03	-1.023495D+03	-4.552389D+03	6.738230D+03
62	-4.605120D+03	-1.366280D+03	-4.725118D+03	6.737996D+03
63	-4.327338D+03	-1.702667D+03	-4.875673D+03	6.737738D+03
64	-4.029224D+03	-2.031073D+03	-5.003341D+03	6.737457D+03
65	-3.712175D+03	-2.349953D+03	-5.107514D+03	6.737152D+03
66	-3.377677D+03	-2.657802D+03	-5.187695D+03	6.736824D+03
67	-3.027300D+03	-2.953169D+03	-5.243502D+03	6.736473D+03
68	-2.662690D+03	-3.234661D+03	-5.274666D+03	6.736101D+03
69	-2.285561D+03	-3.500950D+03	-5.281034D+03	6.735708D+03
70	-1.897687D+03	-3.750777D+03	-5.262573D+03	6.735297D+03
71	-1.500894D+03	-3.982965D+03	-5.219366D+03	6.734869D+03
72	-1.097048D+03	-4.196415D+03	-5.151612D+03	6.734428D+03
73	-6.880538D+02	-4.390122D+03	-5.059628D+03	6.733976D+03
74	-2.758362D+02	-4.563170D+03	-4.943847D+03	6.733516D+03
75	1.376625D+02	-4.714744D+03	-4.804811D+03	6.733051D+03
76	5.504943D+02	-4.844128D+03	-4.643176D+03	6.732585D+03
77	9.607149D+02	-4.950712D+03	-4.459701D+03	6.732122D+03
78	1.366392D+03	-5.033996D+03	-4.255253D+03	6.731665D+03
79	1.765616D+03	-5.093587D+03	-4.030793D+03	6.731219D+03
80	2.156508D+03	-5.129205D+03	-3.787379D+03	6.730788D+03
81	2.537228D+03	-5.140685D+03	-3.526158D+03	6.730376D+03
82	2.905985D+03	-5.127974D+03	-3.248359D+03	6.729986D+03
83	3.261046D+03	-5.091133D+03	-2.955292D+03	6.729622D+03
84	3.600742D+03	-5.030337D+03	-2.648335D+03	6.729288D+03
85	3.923476D+03	-4.945874D+03	-2.328932D+03	6.728987D+03

Time	X	Y	Z	Magnitude
(minutes)	(km)	(km)	(km)	(km)
86	4.227733D+03	-4.838142D+03	-1.998586D+03	6.728721D+03
87	4.512084D+03	-4.707649D+03	-1.658850D+03	6.728495D+03
88	4.775193D+03	-4.555010D+03	-1.311321D+03	6.728309D+03
89	5.015827D+03	-4.380942D+03	-9.576314D+02	6.728167D+03
90	5.232854D+03	-4.186265D+03	-5.994441D+02	6.728068D+03
91	5.425256D+03	-3.971893D+03	-2.384411D+02	6.728015D+03
92	5.517814D+03	-3.849389D+03	-4.462297D+01	6.728006D+03

Orbit Propagation - NPOE, Osculating Elements (J2 Perturbation)
Initial orbital elements
semi-major axis = 6728 km
eccentricity = 0
argument of perigee = 0 degrees
inclination = 51.6 degrees
True-anomaly = 0 degrees
Right-ascension = 325.4 degrees
Geocentric inertial frame of reference

Time	X	Y	Z	Magnitude
(minutes)	(km)	(km)	(km)	(km)
0	5.538061D+03	-3.820453D+03	0.000000D+00	6.728000D+03
1	5.687765D+03	-3.575502D+03	3.616436D+02	6.727977D+03
2	5.810641D+03	-3.313687D+03	7.215765D+02	6.727908D+03
3	5.906111D+03	-3.036243D+03	1.078096D+03	6.727793D+03
4	5.973725D+03	-2.744479D+03	1.429516D+03	6.727635D+03
5	6.013168D+03	-2.439771D+03	1.774175D+03	6.727434D+03
6	6.024255D+03	-2.123559D+03	2.110444D+03	6.727193D+03
7	6.006937D+03	-1.797334D+03	2.436732D+03	6.726914D+03
8	5.961300D+03	-1.462636D+03	2.751499D+03	6.726600D+03
9	5.887561D+03	-1.121042D+03	3.053258D+03	6.726254D+03
10	5.786071D+03	-7.741652D+02	3.340583D+03	6.725879D+03
11	5.657314D+03	-4.236394D+02	3.612119D+03	6.725479D+03
12	5.501898D+03	-7.111737D+01	3.866585D+03	6.725059D+03
13	5.320559D+03	2.817398D+02	4.102780D+03	6.724621D+03
14	5.114155D+03	6.332697D+02	4.319589D+03	6.724170D+03
15	4.883660D+03	9.818165D+02	4.515991D+03	6.723710D+03
16	4.630162D+03	1.325739D+03	4.691060D+03	6.723245D+03
17	4.354856D+03	1.663417D+03	4.843969D+03	6.722779D+03
18	4.059038D+03	1.993260D+03	4.973996D+03	6.722314D+03
19	3.744103D+03	2.313716D+03	5.080528D+03	6.721856D+03
20	3.411533D+03	2.623273D+03	5.163060D+03	6.721407D+03
21	3.062894D+03	2.920475D+03	5.221201D+03	6.720970D+03
22	2.699827D+03	3.203918D+03	5.254674D+03	6.720548D+03
23	2.324039D+03	3.472267D+03	5.263318D+03	6.720142D+03
24	1.937300D+03	3.724254D+03	5.247088D+03	6.719757D+03
25	1.541431D+03	3.958688D+03	5.206055D+03	6.719392D+03
26	1.138295D+03	4.174460D+03	5.140407D+03	6.719049D+03
27	7.297915D+02	4.370547D+03	5.050449D+03	6.718729D+03
28	3.178460D+02	4.546019D+03	4.936600D+03	6.718433D+03
29	-9.559918D+01	4.700042D+03	4.799390D+03	6.718160D+03
30	-5.085930D+02	4.831882D+03	4.639461D+03	6.717912D+03
31	-9.191854D+02	4.940907D+03	4.457562D+03	6.717687D+03
32	-1.325436D+03	5.026595D+03	4.254547D+03	6.717485D+03
33	-1.725424D+03	5.088531D+03	4.031370D+03	6.717304D+03
34	-2.117255D+03	5.126415D+03	3.789081D+03	6.717145D+03
35	-2.499075D+03	5.140057D+03	3.528823D+03	6.717005D+03
36	-2.869073D+03	5.129385D+03	3.251823D+03	6.716884D+03
37	-3.225494D+03	5.094441D+03	2.959391D+03	6.716780D+03

Time	X	Y	Z	Magnitude
(minutes)	(km)	(km)	(km)	(km)
38	-3.566647D+03	5.035380D+03	2.652908D+03	6.716691D+03
39	-3.890910D+03	4.952476D+03	2.333827D+03	6.716617D+03
40	-4.196744D+03	4.846114D+03	2.003658D+03	6.716556D+03
41	-4.482694D+03	4.716791D+03	1.663969D+03	6.716507D+03
42	-4.747401D+03	4.565117D+03	1.316372D+03	6.716468D+03
43	-4.989606D+03	4.391804D+03	9.625173D+02	6.716439D+03
44	-5.208159D+03	4.197673D+03	6.040864D+02	6.716420D+03
45	-5.402020D+03	3.983641D+03	2.427832D+02	6.716409D+03
46	-5.570268D+03	3.750721D+03	-1.196744D+02	6.716407D+03
47	-5.712107D+03	3.500019D+03	-4.815630D+02	6.716413D+03
48	-5.826865D+03	3.232723D+03	-8.411617D+02	6.716428D+03
49	-5.913999D+03	2.950101D+03	-1.196761D+03	6.716452D+03
50	-5.973101D+03	2.653496D+03	-1.546672D+03	6.716485D+03
51	-6.003895D+03	2.344315D+03	-1.889230D+03	6.716529D+03
52	-6.006241D+03	2.024027D+03	-2.222811D+03	6.716585D+03
53	-5.980135D+03	1.694150D+03	-2.545832D+03	6.716653D+03
54	-5.925709D+03	1.356250D+03	-2.856761D+03	6.716735D+03
55	-5.843229D+03	1.011930D+03	-3.154126D+03	6.716832D+03
56	-5.733094D+03	6.628195D+02	-3.436520D+03	6.716946D+03
57	-5.595836D+03	3.105736D+02	-3.702608D+03	6.717078D+03
58	-5.432113D+03	-4.314129D+01	-3.951134D+03	6.717229D+03
59	-5.242710D+03	-3.966523D+02	-4.180926D+03	6.717401D+03
60	-5.028531D+03	-7.482889D+02	-4.390902D+03	6.717595D+03
61	-4.790598D+03	-1.096391D+03	-4.580075D+03	6.717811D+03
62	-4.530042D+03	-1.439315D+03	-4.747557D+03	6.718051D+03
63	-4.248102D+03	-1.775446D+03	-4.892563D+03	6.718314D+03
64	-3.946116D+03	-2.103198D+03	-5.014413D+03	6.718602D+03
65	-3.625515D+03	-2.421029D+03	-5.112539D+03	6.718913D+03
66	-3.287816D+03	-2.727443D+03	-5.186484D+03	6.719248D+03
67	-2.934616D+03	-3.020998D+03	-5.235905D+03	6.719606D+03
68	-2.567583D+03	-3.300313D+03	-5.260574D+03	6.719984D+03
69	-2.188451D+03	-3.564075D+03	-5.260380D+03	6.720383D+03
70	-1.799007D+03	-3.811045D+03	-5.235328D+03	6.720800D+03
71	-1.401088D+03	-4.040061D+03	-5.185540D+03	6.721233D+03
72	-9.965697D+02	-4.250046D+03	-5.111254D+03	6.721678D+03
73	-5.873570D+02	-4.440013D+03	-5.012822D+03	6.722134D+03
74	-1.753779D+02	-4.609068D+03	-4.890711D+03	6.722598D+03
75	2.374273D+02	-4.756414D+03	-4.745498D+03	6.723064D+03
76	6.491141D+02	-4.881358D+03	-4.577867D+03	6.723531D+03
77	1.057743D+03	-4.983309D+03	-4.388611D+03	6.723994D+03
78	1.461390D+03	-5.061786D+03	-4.178622D+03	6.724450D+03
79	1.858153D+03	-5.116417D+03	-3.948890D+03	6.724893D+03
80	2.246160D+03	-5.146943D+03	-3.700499D+03	6.725321D+03
81	2.623584D+03	-5.153216D+03	-3.434620D+03	6.725730D+03
82	2.988644D+03	-5.135204D+03	-3.152507D+03	6.726114D+03
83	3.339617D+03	-5.092989D+03	-2.855493D+03	6.726472D+03
84	3.674849D+03	-5.026767D+03	-2.544980D+03	6.726799D+03
85	3.992755D+03	-4.936847D+03	-2.222434D+03	6.727092D+03

Time (minutes)	X (km)	Y (km)	Z (km)	Magnitude (km)
86	4.291837D+03	-4.823650D+03	-1.889378D+03	6.727348D+03
87	4.570680D+03	-4.687707D+03	-1.547387D+03	6.727564D+03
88	4.827968D+03	-4.529658D+03	-1.198078D+03	6.727739D+03
89	5.062487D+03	-4.350246D+03	-8.431020D+02	6.727870D+03
90	5.273129D+03	-4.150315D+03	-4.841379D+02	6.727956D+03
91	5.458899D+03	-3.930810D+03	-1.228835D+02	6.727997D+03
92	5.547814D+03	-3.805617D+03	7.084858D+01	6.727999D+03

Orbit Propagation - NPOE, Mean Elements (J2 Perturbation)

Initial orbital elements

semi-major axis = 6728 km

eccentricity = 0

argument of perigee = 0 degrees

inclination = 51.6 degrees

True-anomaly = 0 degrees

Right-ascension = 325.4 degrees

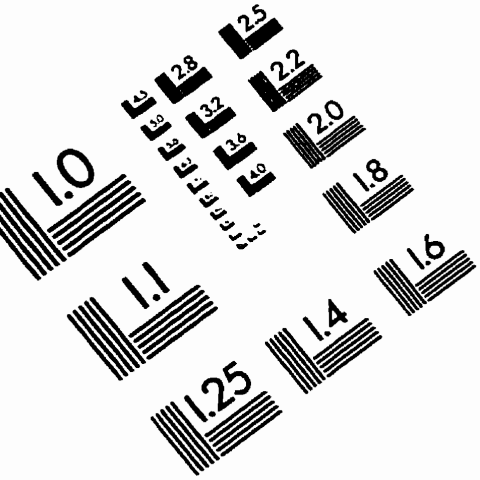
Geocentric inertial frame of reference

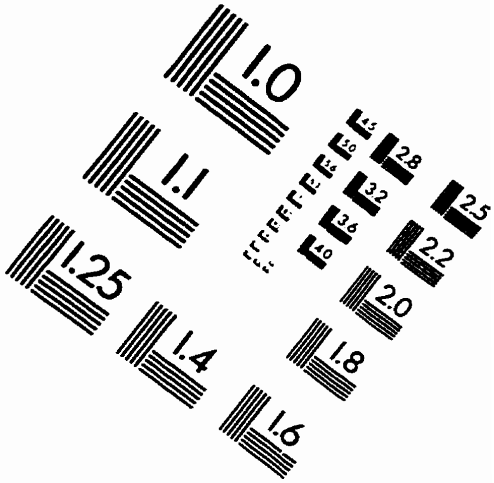
Time	X	Y	Z	Magnitude
(minutes)	(km)	(km)	(km)	(km)
0	5.538066D+03	-3.820456D+03	-6.685526D-11	6.728005D+03
1	5.687674D+03	-3.575671D+03	3.619315D+02	6.728005D+03
2	5.810483D+03	-3.314035D+03	7.221556D+02	6.728005D+03
3	5.905915D+03	-3.036780D+03	1.078973D+03	6.728005D+03
4	5.973522D+03	-2.745209D+03	1.430700D+03	6.728004D+03
5	6.012987D+03	-2.440694D+03	1.775679D+03	6.728004D+03
6	6.024125D+03	-2.124669D+03	2.112280D+03	6.728003D+03
7	6.006885D+03	-1.798621D+03	2.438917D+03	6.728003D+03
8	5.961349D+03	-1.464084D+03	2.754049D+03	6.728002D+03
9	5.887734D+03	-1.122634D+03	3.056189D+03	6.728002D+03
10	5.786387D+03	-7.758761D+02	3.343912D+03	6.728002D+03
11	5.657787D+03	-4.254443D+02	3.615861D+03	6.728002D+03
12	5.502541D+03	-7.298786D+01	3.870752D+03	6.728002D+03
13	5.321382D+03	2.798339D+02	4.107384D+03	6.728002D+03
14	5.115163D+03	6.313597D+02	4.324639D+03	6.728002D+03
15	4.884858D+03	9.799344D+02	4.521494D+03	6.728003D+03
16	4.631553D+03	1.323917D+03	4.697019D+03	6.728003D+03
17	4.356441D+03	1.661686D+03	4.850387D+03	6.728004D+03
18	4.060820D+03	1.991653D+03	4.980874D+03	6.728004D+03
19	3.746083D+03	2.312262D+03	5.087864D+03	6.728005D+03
20	3.413713D+03	2.622004D+03	5.170852D+03	6.728005D+03
21	3.065277D+03	2.919419D+03	5.229448D+03	6.728005D+03
22	2.702418D+03	3.203107D+03	5.263374D+03	6.728006D+03
23	2.326844D+03	3.471730D+03	5.272471D+03	6.728005D+03
24	1.940326D+03	3.724023D+03	5.256695D+03	6.728005D+03
25	1.544685D+03	3.958797D+03	5.216121D+03	6.728005D+03
26	1.141785D+03	4.174946D+03	5.150941D+03	6.728004D+03
27	7.335252D+02	4.371450D+03	5.061462D+03	6.728003D+03
28	3.218280D+02	4.547383D+03	4.948106D+03	6.728002D+03
29	-9.136663D+01	4.701916D+03	4.811407D+03	6.728002D+03
30	-5.041123D+02	4.834320D+03	4.652012D+03	6.728001D+03
31	-9.144648D+02	4.943970D+03	4.470671D+03	6.728000D+03
32	-1.320491D+03	5.030348D+03	4.268240D+03	6.727999D+03
33	-1.720279D+03	5.093046D+03	4.045675D+03	6.727998D+03
34	-2.111945D+03	5.131768D+03	3.804025D+03	6.727998D+03
35	-2.493645D+03	5.146329D+03	3.544430D+03	6.727998D+03
36	-2.863582D+03	5.136660D+03	3.268114D+03	6.727998D+03
37	-3.220012D+03	5.102805D+03	2.976382D+03	6.727998D+03

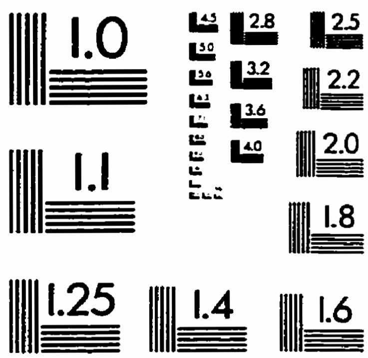
Time (minutes)	X (km)	Y (km)	Z (km)	Magnitude (km)
38	-3.561259D+03	5.044922D+03	2.670609D+03	6.727998D+03
39	-3.885714D+03	4.963281D+03	2.352238D+03	6.727998D+03
40	-4.191851D+03	4.858266D+03	2.022770D+03	6.727998D+03
41	-4.478227D+03	4.730370D+03	1.683760D+03	6.727999D+03
42	-4.743496D+03	4.580193D+03	1.336807D+03	6.727999D+03
43	-4.986407D+03	4.408441D+03	9.835481D+02	6.727999D+03
44	-5.205819D+03	4.215921D+03	6.256491D+02	6.728000D+03
45	-5.400698D+03	4.003539D+03	2.647986D+02	6.728000D+03
46	-5.570127D+03	3.772294D+03	-9.730100D+01	6.728000D+03
47	-5.713310D+03	3.523272D+03	-4.589416D+02	6.728000D+03
48	-5.829572D+03	3.257646D+03	-8.184172D+02	6.728000D+03
49	-5.918369D+03	2.976665D+03	-1.174032D+03	6.727999D+03
50	-5.979282D+03	2.681651D+03	-1.524108D+03	6.727999D+03
51	-6.012026D+03	2.373993D+03	-1.866995D+03	6.727999D+03
52	-6.016447D+03	2.055138D+03	-2.201074D+03	6.727998D+03
53	-5.992528D+03	1.726588D+03	-2.524770D+03	6.727998D+03
54	-5.940381D+03	1.389887D+03	-2.836555D+03	6.727998D+03
55	-5.860253D+03	1.046622D+03	-3.134960D+03	6.727998D+03
56	-5.752523D+03	6.984077D+02	-3.418575D+03	6.727998D+03
57	-5.617700D+03	3.468838D+02	-3.686064D+03	6.727998D+03
58	-5.456421D+03	-6.294981D+00	-3.936165D+03	6.727998D+03
59	-5.269445D+03	-3.594657D+02	-4.167697D+03	6.727999D+03
60	-5.057656D+03	-7.109656D+02	-4.379569D+03	6.727999D+03
61	-4.822051D+03	-1.059140D+03	-4.570781D+03	6.728000D+03
62	-4.563742D+03	-1.402348D+03	-4.740431D+03	6.728001D+03
63	-4.283946D+03	-1.738975D+03	-4.887718D+03	6.728002D+03
64	-3.983982D+03	-2.067434D+03	-5.011948D+03	6.728003D+03
65	-3.665265D+03	-2.386179D+03	-5.112535D+03	6.728004D+03
66	-3.329296D+03	-2.693708D+03	-5.189004D+03	6.728004D+03
67	-2.977660D+03	-2.988573D+03	-5.240994D+03	6.728005D+03
68	-2.612012D+03	-3.269385D+03	-5.268261D+03	6.728005D+03
69	-2.234076D+03	-3.534819D+03	-5.270675D+03	6.728005D+03
70	-1.845633D+03	-3.783627D+03	-5.248225D+03	6.728005D+03
71	-1.448514D+03	-4.014634D+03	-5.201017D+03	6.728005D+03
72	-1.044589D+03	-4.226752D+03	-5.129274D+03	6.728005D+03
73	-6.357617D+02	-4.418982D+03	-5.033334D+03	6.728005D+03
74	-2.239579D+02	-4.590417D+03	-4.913649D+03	6.728004D+03
75	1.888823D+02	-4.740249D+03	-4.770786D+03	6.728004D+03
76	6.008142D+02	-4.867770D+03	-4.605416D+03	6.728003D+03
77	1.009897D+03	-4.972380D+03	-4.418322D+03	6.728003D+03
78	1.414205D+03	-5.053583D+03	-4.210384D+03	6.728002D+03
79	1.811832D+03	-5.110997D+03	-3.982585D+03	6.728002D+03
80	2.200906D+03	-5.144350D+03	-3.735998D+03	6.728002D+03
81	2.579596D+03	-5.153483D+03	-3.471788D+03	6.728002D+03
82	2.946117D+03	-5.138351D+03	-3.191199D+03	6.728002D+03
83	3.298744D+03	-5.099025D+03	-2.895557D+03	6.728002D+03
84	3.635815D+03	-5.035687D+03	-2.586256D+03	6.728002D+03
85	3.955745D+03	-4.948636D+03	-2.264754D+03	6.728003D+03

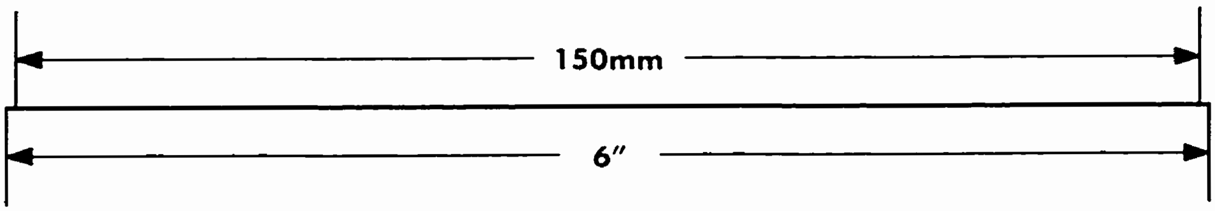
Time (minutes)	X (km)	Y (km)	Z (km)	Magnitude (km)
86	4.257026D+03	-4.838278D+03	-1.932568D+03	6.728003D+03
87	4.538241D+03	-4.705133D+03	-1.591266D+03	6.728004D+03
88	4.798065D+03	-4.549826D+03	-1.242457D+03	6.728004D+03
89	5.035275D+03	-4.373086D+03	-8.877867D+02	6.728005D+03
90	5.248756D+03	-4.175745D+03	-5.289285D+02	6.728005D+03
91	5.437502D+03	-3.958731D+03	-1.675752D+02	6.728005D+03
92	5.528062D+03	-3.834826D+03	2.629573D+01	6.728005D+03

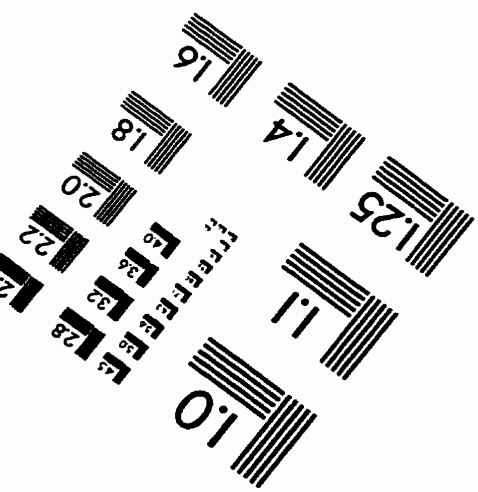
IMAGE EVALUATION TEST TARGET (QA-3)

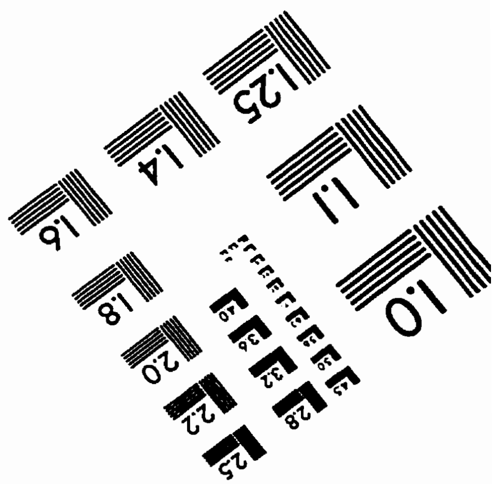












APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989