MODELING PLANETARY MOTIONS: WHY WE CARE AND HOW WE DO IT

MARC A. MURISON
Astronomical Applications Dept., U.S. Naval Observatory

1. INTRODUCTION

1.1. Why do we need precise planetary positions?

There are several good reasons why we need precise planetary positions. I will mention just a few that impact the U.S. Naval Observatory and its various scientific (and military) programs and objectives. There are two broad requirement categories: Department of Defense (DoD) requirements and astronomical requirements, which overlap significantly. First, for ordnance guiding and targeting purposes, the DoD requires stellar positions to better than 20 milliarcseconds (mas)[1]. Placing accuracy requirements upon a stellar reference frame has implications for how accurately we must then know, among other things, the positions of the planets and other solar system objects. The sequence of connections that joins the two seemingly disparate accuracies (that of the stars and that of the planets) requires a discussion of dynamical and astronomical reference frames, which I will get to in a moment.

The second broad requirement category is the astronomical need for accurate planetary positions. Besides the intrinsic interest of astronomers in planetary, asteroidal, and cometary positions, knowledge of these positions over time — called an *ephemeris* — fundamentally affects many areas of solar system and even stellar astronomy:

1. To the general public, perhaps the most apparent astronomical need for precise planetary positions is in spacecraft navigation.
2. Solar system celestial mechanics depends greatly on accurate positions. Theories of planetary and satellite motions live or die according to how well their predictions agree with observational knowledge of positions. These theories are the means by which we develop our most fundamental understanding of the many complicated dynamical processes and interactions in the solar system.

3. Another area where accurate knowledge of planetary positions is crucial is stellar occulations. If a planet is passing in front of a star, and we can predict where on the Earth's surface this event is visible, then we may learn several things, including density and composition of the planetary atmosphere, certain facts about the atmosphere of the occulted star, and, of course, better knowledge of either the star's position, the planet's position, or both. Further, if the occulting body is an asteroid, we can even determine the projected shape of the asteroid.

4. Finally, General Relativity has been tested by observing starlight that grazes a body — the Sun, Jupiter, and Earth have all been used thus. Again, accurate knowledge of planetary positions is essential.

This is not an exhaustive list of the astronomical benefits of accurate planetary positions, but it gives a flavor of the value of such positional knowledge and prediction.

1.2. Dependencies

Theory, observation, and application are interdependent, as illustrated in Figure 1. Observations can be interpreted only in the context of our understanding — represented by theoretical models — of the solar system and its dynamics. The observations can then be used to correct and update our theoretical models. The fusion of the two results in planetary (and satellite) ephemerides as well as a better solar system reference frame. The combination of planetary system model — as determined by

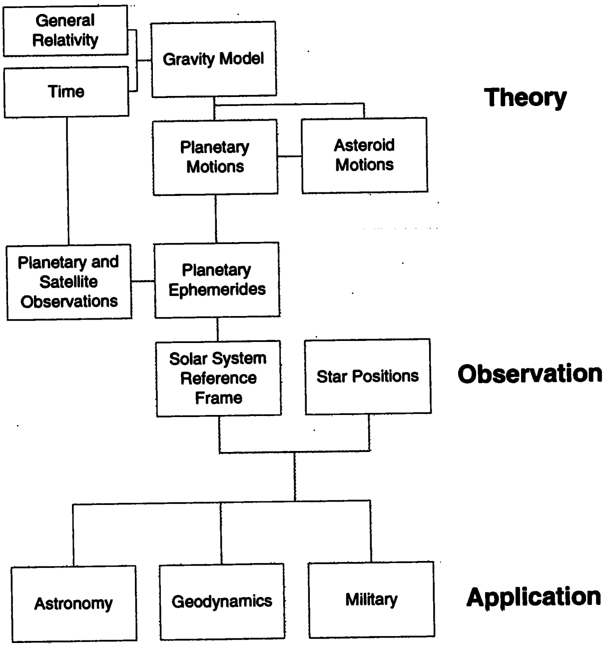


Figure 1 — Interplay between theory, observation, and application.

theory and refined by observations — plus stellar position catalogs, gives rise to a multitude of practical applications in many areas, including astronomy, geophysics, and military.

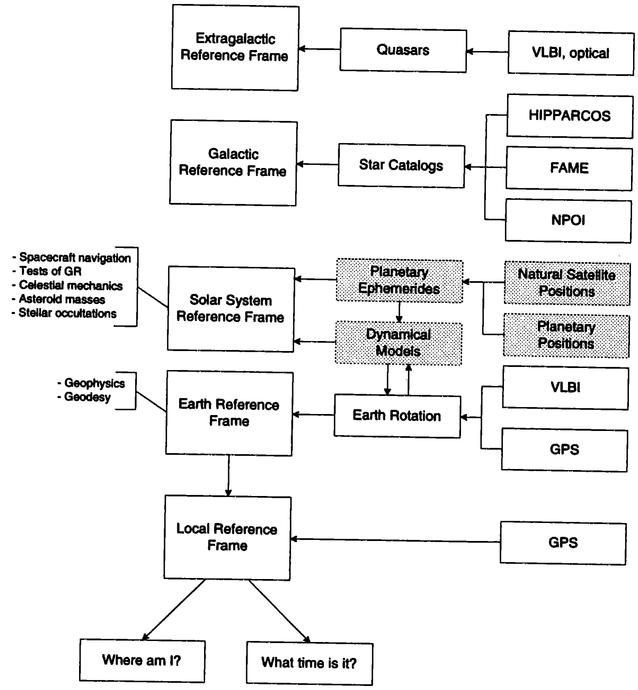


Figure 2 — A reference frame hierarchy, showing the context in which modeling planetary motions (shaded boxes) resides.

1.3. Reference Frames

Although we here on the Earth's surface might often prefer to work with a nearly inertial frame of reference tied to the distant stars, we must instead put up with the various noninertial reference frames to which we find ourselves affixed. Defining and/or connecting them is both observationally and theoretically a complex undertaking. We must tie together a local reference frame, attached to a specific location (and a specific time) on the Earth's surface, to a reference frame that takes into account the spinning and wobbling motions of the Earth. This spinning and wobbling frame can be connected via lunar laser ranging to a frame that encompasses the dynamic solar system, with all its complicated planetary and satellite motions, each body affecting to various degrees the motions of every other body in accordance with Newton's and Einstein's theories of motion in gravitational fields. We then try to join the solar system dynamical frame to a Galactic frame, which takes the form of standard catalog frames such as FK5[2] or HIPPARCOS[3], which are stellar catalogs, or the new ICRF[4], based on extragalactic objects. The Naval Observatory has been and continues to be among the world's foremost contributors to and creators of these kinds of fundamental position catalogs. Figure 2 is an illustration showing various reference frames, from the largest scale to the smallest, and the kinds of processes or observational methods

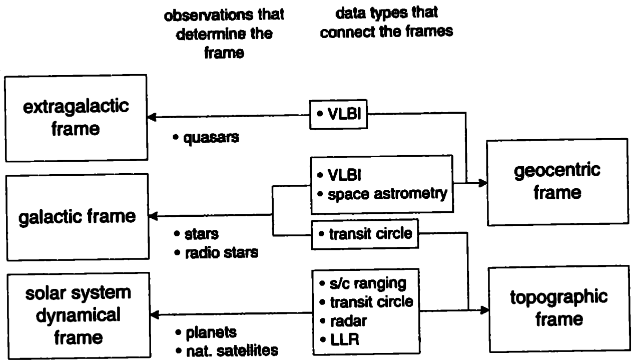


Figure 3 — Connections between reference frames.

that they rely on. The first column is a hierarchy of frames, from the size scale of the universe down to the local and very practical question of "Where am I right now?". We must attach the notion of time to that of position, since in any dynamical frame the two are inextricably linked. The second column contains the major input category or dynamics type that corresponds to the associated reference frame. The third column lists the most important observation types that determine the reference frame. The activities associated with planetary ephemeris generation correspond to the shaded boxes, and these are the areas we will concentrate on here. Figure 3 shows these same reference frames, but organized to show how they are related to each other observationally.

1.4. Generating Precise Predictions of Planetary Positions

We have established the need for accurate planetary positions. Therefore, we need to be able to *generate* accurate *predictions* of planetary positions. These tabulated predictions we call *ephemerides*.[5] How does one generate an ephemeris? This is a three-stage process, as illustrated in Figure 4.

First and foremost, we must obtain accurate observations. Historically, observations consisted mainly of ground-based optical positions of planets and their satellites. Satellite positions are the more valuable since satellites generally have no atmospheres (and hence no limb-darkening) to contend with. Since they orbit their parent planets in a manner predictable by Newton's law of gravity, their positions can form the basis for determining the parent planet positions. This is complicated considerably, however, by the difficulty in constructing highly accurate theories of satellite motions, caused by the complex interactions of the satellites with each other, with the planets, and with the nonspherical parent planet whose mass distribution we don't always know as well as we'd like. The modern era has seen the advent of other types of observations, including ground-based radar, lunar laser ranging, spacecraft telemetry, and space-based astrometry. The European HIPPARCOS mission[6] is a successful example of a space astrometry mission. We hope other missions, such as FAME (USNO)[7], and SIM (JPL)[8] or GAIA (ESA)[9], will follow.

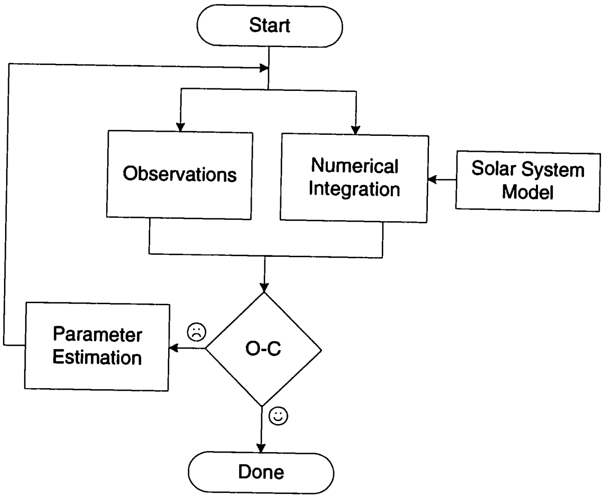


Figure 4 — Generating a planetary ephemeris. Numerical integrations of a solar system model are compared with observational data, resulting in O-C residuals. Based on these residuals, model parameters are refined and the entire process repeated until convergence.

The second step in generating an ephemeris is to develop a comprehensive solar system model that we then integrate numerically. We must include complications, such as planetary (especially Earth) rotation dynamics and lunar motion, as well as more subtle effects, including general relativity, tidal interactions between Earth and Moon, and planetary topography models (for better resolution of radar data). The state of the art has advanced to the point that it is becoming necessary to include the masses of individual asteroids[10] as well as a mass model for the asteroid belt. Both of these kinds of masses are in general very poorly known, yet asteroidal mass uncertainties are now the largest source of error in high-precision ephemerides of the inner planets. Currently, the JPL ephemerides (specifically, DE405) include mass estimates for 300 asteroids. These masses are based on IRAS magnitudes, albedo estimates, and mean density estimates.

The solar system model contains many adjustable parameters, such as masses, orbital elements, initial positions and velocities, gravity model parameters, and so on. The third step in generating an ephemeris is to simultaneously fit all of these model parameters to the available observations. This requires performing a nonlinear least squares analysis of a comparison between a numerical integration of the solar system model and the observational data. This analysis results in (hopefully minor) adjustments to the model parameters. We then integrate the model again, using the adjusted parameter values, then compare again to the observations. We iterate this process until the parameters stop changing appreciably. At that point, we have the best fit of the solar system model to the available observations.

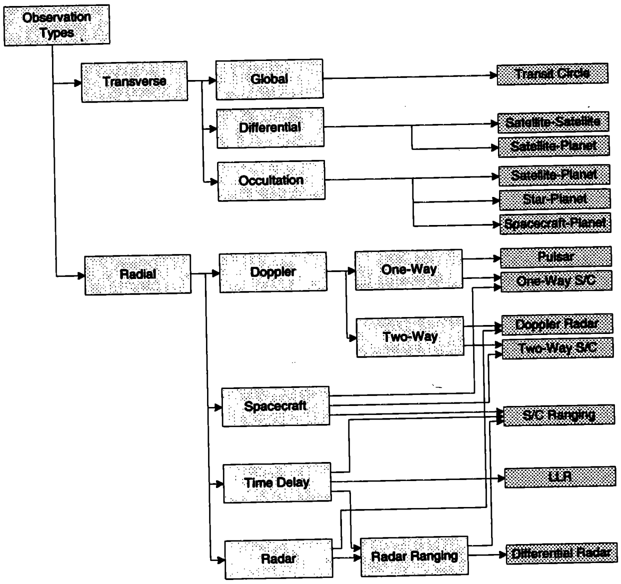


Figure 5 — Observation types.

1.5. Observation Types

For observing planetary positions, the various observational data types fall naturally into the two broad categories: timing (in a sense, the radial coordinate from the observer) and positions on the sky (i.e., transverse to the radial direction). The hierarchy of types is illustrated in Figure 5.

1.6. Example: Space-Based Asteroid and Natural Satellite Observations

Figure 6 shows space-based astrometric observations by HIPPARCOS of the 48 asteroids and 3 natural satellites it was able to reach. For the brightest asteroids, the

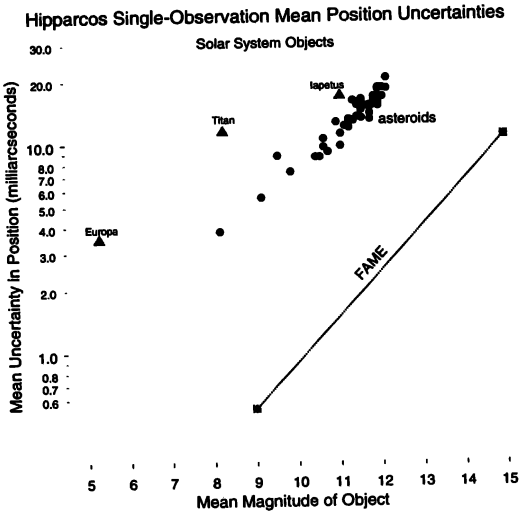


Figure 6 — HIPPARCOS single-observation accuracy of solar system objects, with a comparison to projected FAME single-observation accuracy.

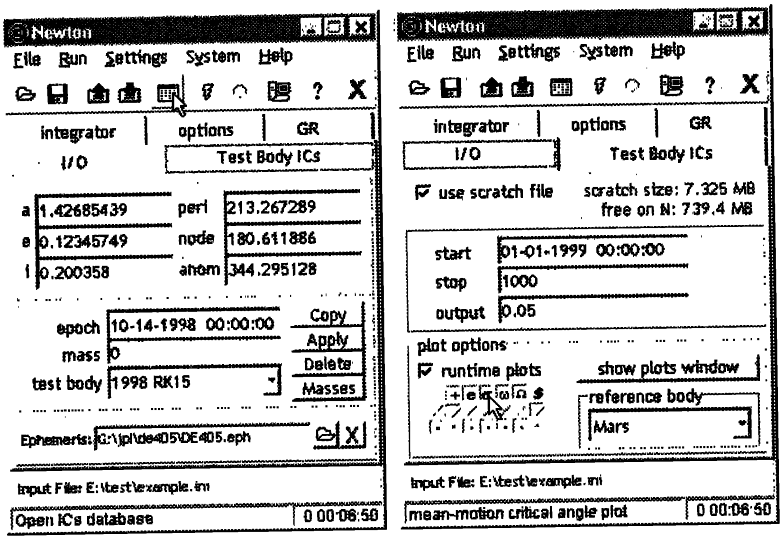
single-measurement accuracy is less than 10 milliarcseconds. The accuracy of the satellites is degraded by the fact that at the resolution of the HIPPARCOS telescope these objects are not point sources but extended bodies, introducing centroiding difficulties. This figure also shows the projected single-measurement accuracy of the FAME satellite[11]. (The USNO is hoping to launch FAME in 2003 or 2004 as a NASA MIDEX mission.) FAME will be able to do an order of magnitude better than HIPPARCOS in positional measurements of solar system objects. FAME will also go substantially fainter, allowing observations of many more asteroids and natural satellites than HIPPARCOS. Natural satellite observations — especially in the outer solar system — are important because, combined with integrations of their motion, they can be used to obtain the positions of the parent planets much more accurately than observations of the planets themselves. FAME will potentially be able to reach over 20 natural satellites and upwards of 2100 asteroids.

2. NEWCOMB: A SOLAR SYSTEM EPHEMERIS PROGRAM

Newcomb[12] is a new Solar System Ephemeris program currently under development at the U.S. Naval Observatory. In terms of use at the USNO, Newcomb will be the successor of PEP, the Planetary Ephemeris Program maintained at the Smithsonian Astrophysical Observatory[13], and of the DE series of programs from the Jet Propulsion Laboratory. DE and PEP are the only currently existing high-precision solar system ephemeris programs.

2.1. Motivation: Why a New Program?

The developmental origins of both PEP and the DE programs dates from the early to mid 1960s. Both computer program design and language capabilities, as well as the precision of both observational data and the practical needs for that data, have advanced far beyond the anticipations of three and a half decades ago when PEP and the JPL DE programs were originally developed. Program technology that is several generations out of date, combined with the practical inability to add further significant capabilities or modifications to PEP, has been deemed sufficient cause for development of a new ephemeris program. Additional motivations are that it is to the USNO's great advantage to have a comprehensive ephemeris capability in-house (especially since the NAO publishes the *Astro-*



a 1.42685439
e 0.12345749
i 0.200358
peri 213.267289
node 180.611886
anom 344.295128
epoch 10-14-1998 00:00:00
mass 0
test body 1998 RK15
Ephemeris: G:\jpl\de405\DE405.eph
Input File: E:\test\example.ini
Open ICs database 0 00:06:50

☑ use scratch file
scratch size: 7.325 MB
free on N: 739.4 MB
start 01-01-1999 00:00:00
stop 1000
output 0.05
☑ runtime plots
reference body Mars
Input File: E:\test\example.ini
mean-motion critical angle plot 0 00:06:50

Figure 7 — Examples of a graphical user interface (from Newton)

nomical Almanac), and that Newcomb will provide a check against PEP and the JPL DE programs.

Finally, the creation of a modern ephemeris program provides an opportunity to simultaneously develop a flexible research tool for investigation of solar system dynamics. The integration module of Newcomb is in large part already completed and exists as a standalone program called Newton.[14] This program is currently being used in the Astronomical Applications Department[15] for investigations of the dynamics of inner solar system asteroids, the asteroidal "noise" in the motions of Earth and Mars, and the dynamics of trans-Neptunian objects.

To highlight the difference between "old" and "new" programming, consider the task of setting program input parameters. Appendix A contains a typical input file used by PEP. Figure 7 shows how it is done using a GUI. The intuitiveness of the GUI approach leads to substantial time savings in coming up to speed in program usage, as well as actual use of the program day to day. Even more valuable is that it allows a much more sophisticated interface and a much more sophisticated set of program capabilities.

2.2. *Advantages of Modern Programming and Design*

Chief among the advantages of writing a new program is the opportunity to make use of both modern programming and modern design technologies, namely object-oriented programming (OOP) and object-oriented design (OOD), as well as graphical user interfaces (GUIs). Recently, the highly productive "components" programming associated with rapid application development (RAD) environments has greatly enhanced the efficiency, sophistication, and dependability of GUI programming. Additionally, modern integrated development environments (IDEs) have matured into a powerful and reliable means of rapidly developing, testing, and debugging complex and sophisticated programs. None of these powerful technologies was available until the 1990s. Hence, design and construction of modern programs is faster, safer, and more intuitive. Also very important is the fact that all of the numerical algorithms used in a high-precision ephemeris program — e.g., numerical integrators, nonlinear estimation, etc. — are now mature technologies, which was certainly not the case thirty-five years ago.

Consequently, the Newcomb computational back end is written entirely in ANSI C++, and development and testing are done entirely within the best C++ RAD environment currently available.[16] Throughout the program, we take full advantage of standard OOP/OOD concepts and techniques, including full data encapsulation, template and nested template classes, polymorphism, and, where necessary, multiple inheritance.

The benefits of a completely object-oriented approach are many, including faster prototyping and development, fewer and more easily locatable coding errors, vastly simpler and more intuitive design, more sophisticated functionality, easily extensible architecture, and (most importantly) drastically reduced long-term maintenance costs. Another major benefit is that the program can be brought up and running with minimal functionality, allowing further capability to be easily and relatively painlessly incorporated as need arises.

Ease of extensibility is largely a result of object-oriented design, but it is also directly related to how good that design is. Hence, considerable effort has gone and is still going into the design of Newcomb. Experience in the software industry over the last one to two decades abundantly shows that the payoff later on in terms of maintenance and extensibility is far out of proportion to the effort expended early on — in the design stages — of the program life cycle.

The benefits of a RAD environment for development and testing are also very attractive. Chief among the attractions is the ease by which it is possible to create highly sophisticated graphical user interfaces. During design, graphical interface components — such as buttons, edit fields, toolbars and so on — are "dropped" onto a window form or dialog box. Useful properties of the components are settable at design time, in addition to being available during runtime. It is easy to create custom components as well. For example, for Newcomb we designed a custom component that is in fact a fully functional and self-contained power spectral density (PSD) analysis package, including plots and file output. All that is needed to add a PSD module to a program is to drop the PSD component onto a form or dialog. Hence, building, changing, and extending the graphical user interface of a program is astoundingly easy once a good overall design has been created. This of course spills over and makes changing or extending major program structural elements correspondingly painless.

2.3. Newcomb Project Outline

In these beginning stages of the Newcomb project, tasks naturally fall into three main categories: program design, documentation, and science applications. A rough outline of the most obvious subjects that must be addressed is:

I. Design Issues
  A. numerical integration scheme
    1. object-oriented design
    2. Integrable objects have knowledge of dynamical environment as well as the ability to dynamically evolve in that environment.
  B. exception handling
    1. all exceptions fully recoverable
    2. procedure stack traceback
  C. robust parameter estimation
    1. Singular Value Decomposition (SVD)
    2. use a mature package from elsewhere
  D. graphical user interface
  E. reduction of observations
  F. individual class design and testing

II. Science Issues and Projects to Consider
  A. asteroids
    1. masses from orbital interactions
    2. provide ephemerides (services to the community)
    3. cumulative effects on planetary motions
      a. Asteroids are the largest source of "noise" in the orbits of Mars and the Earth-Moon system.
  B. lunar motion
    1. chaotic dynamics
      a. predictions from numerical models
      b. comparisons with LLR data
    2. radiation pressure
    3. resonant interaction between tidal and GR terms
    4. lunar librations
  C. Nordtvedt *h* parameter (anomalous gravitational field energy effects — i.e., a difference between gravitational and inertial mass proportional to the gravitational binding energy of a body)
  D. GR precession

    1. lunar orbit
    2. Earth's spin
- E. bounds on time variation of the gravitational constant
- F. millisecond pulsars
    1. derive Earth orbit
- G. bounds on dark matter in the solar system?
- H. planetary satellites?
    1. centroiding vs. satellite-derived center of mass
- I. other science?

III. Documentation
- A. code
    1. source documentation model
    2. interface (user manual)
- B. algorithms
- C. physics
    1. GR and partial derivatives
    2. Earth-Moon tidal interactions
- D. parameter estimation and error and correlation analysis
- E. numerical integration design
- F. reduction of observations

2.4. An Overview of the Newcomb Program Structure

The top level process structure of Newcomb is shown in Figure 8. Basic operation is as follows.

The observations module is responsible for reading input astrometric observations and reducing ("massaging") them as necessary. The observations will be of various types (Figure 5), taken at various observing locations (Figure 11), including spacecraft. The reduction process corrects for various instrumental and other effects (e.g. from the atmosphere) that are specific to a particular set of observations.

The integration module is responsible for numerically integrating a sophisticated dynamical model of the solar system — including general relativistic terms, a detailed Earth-Moon system, planetary spin vectors including preces-

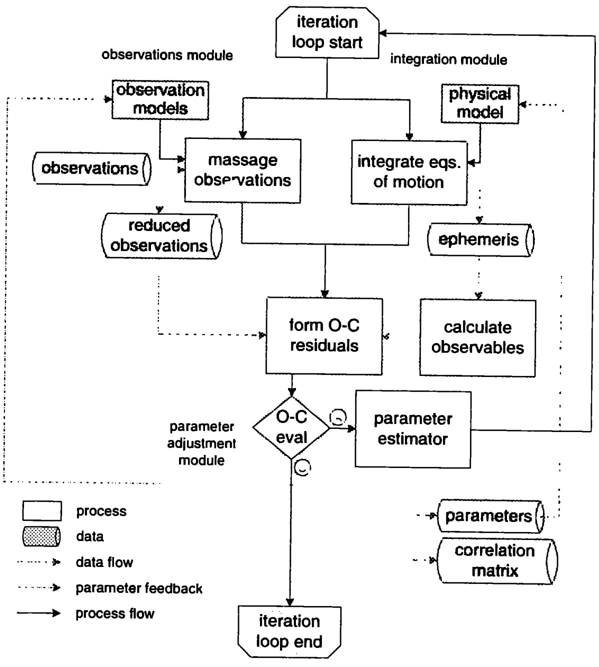


Figure 8 — Major program processes.

sion and nutation, and an unlimited number of asteroids — to produce an ephemeris.

The model ephemeris is then compared with the observations in the O-C section of the parameter adjustment module to produce a set of residuals. The parameter estimator uses the partial derivatives of the model equations with respect to the model parameters (including initial conditions) to solve the associated nonlinear least squares problem for the most probable set of model parameter values that minimizes the O-C residuals.

The adjusted model parameters are then fed back into both the ephemeris generator and the observation transformation methods. The data are rereduced as necessary, and a new ephemeris is generated by the integration module, using the updated parameter values. These are again combined to produce a new set of residuals. This process is iterated until the parameters satisfy predetermined success criteria.

At the end of the iterative process, we will have produced an ephemeris that best fits the observations, given the model used, as well as the best-fit model parameters, formal error estimates of those parameters, and the parameter cross correlations. The parameter error estimates and parameter correlations are derived from the partial derivatives and the correlation matrix from the least squares analysis. Experience with PEP has shown that, normally, at most only a couple or a few iterations are needed

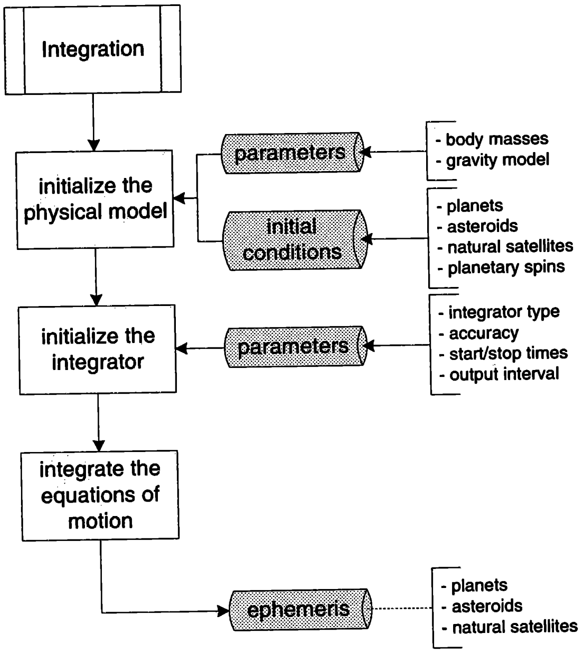


Figure 9 — The Integration Module.

2.5. *The Integration Module*

The integration module of Newcomb is relatively straightforward, as shown in Figure 9. After choosing which bodies to integrate, one sets all the initial conditions for all the integrated bodies, as well as both the physical model parameters (G, masses, etc.) and the integrator parameters (accuracy limits, step size limits, etc.). The integrator then integrates the equations of motion, providing intermediate output along the way. The intermediate output varies in complexity, from simple diagnostics to runtime graphics of orbital elements, close approaches, mean-motion resonance angles, and so on. As previously mentioned, the integration module is such an intrinsically useful tool that it has been broken out as a standalone solar system dynamics application, called Newton.

2.6. The Parameter Adjustment Module

The parameter adjustment module is relatively straightforward. The processed observations from the Observations Module and the calculated ephemeris data from the Integration Module are compared, thus forming the O-C residuals. First, coordinate frame compatibility between the observations and the synthetic ephemeris is reconciled. The calculated ephemeris must be transformed to apparent positions in order to match the observations. The residuals are characterized, with statistical and descriptive output going to disk as well as to an output window on-screen. At this point, outlying data points can be automatically — or manually — detected and removed.

The core of the module follows with the determination of parameters via a nonlinear

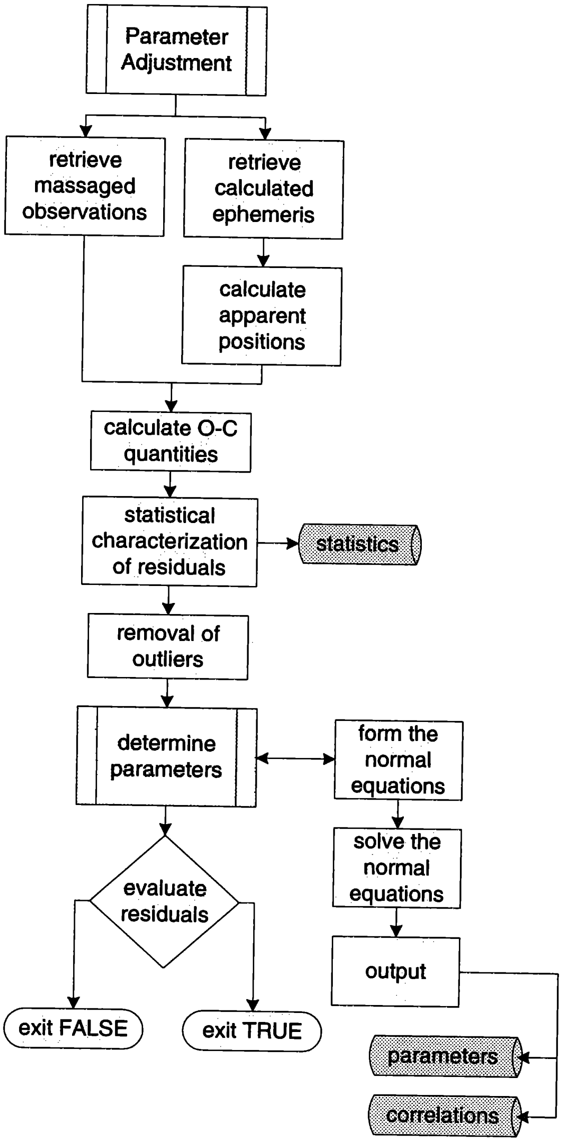


Figure 10 — The Parameter Adjustment Module.

maximum likelihood estimator (e.g., Levenberg-Marquardt). The normal equations are formed and solved, and the parameters and associated formal error estimates are saved. Finally, the residuals are evaluated, and the module exits with a solution "acceptability" code. Figure 10 illustrates the process.

Matrix inversion is accomplished via singular value decomposition (SVD), which is very robust and offers useful diagnostics for ill-conditioned matrices. Singularities are automatically detected and corrected, and the problem parameters are identified. In essence, if the algorithm encounters an ill-conditioned matrix, it safely steps around the problem point(s) and proceeds in such a way as to mine the matrix for the maximum amount of information. When a singularity (rare in practice) or degenerate column (not rare!) is encountered, the combination of parameters that led to the fault is easily extracted. Thus, not only are singularities safely handled, but — more importantly — parameter combinations to which the data are insensitive are automatically identified.

It is unusual to encounter a computational method that is this reliable and blowup-proof. I have already developed and tested matrix inversion using SVD and incorporated it into the Matrix utility class. With regard to Newcomb, SVD is a "plug'n'play" capability.

2.7. *The Observations Module*

Perhaps the most difficult section of the program is the module that processes input observations and reduces them to a form suitable for passage to the O-C section of the parameter adjustment module (see Figure 10). In essence, the observations are sent to the O-C section in the form of apparent positions, corrected for various biases, including (but not limited to):

- ▸ catalog corrections
- ▸ delay/doppler bias corrections
- ▸ coordinate frame fiducialization
- ▸ aberration corrections
- ▸ nutation and precession

Integral to this section are the specific types of observational datasets and the specific types of observational platforms. The data and platform types vary widely.

2.7.1. Observing Platforms

One must consider the various observing platforms presently available in the solar system. They are

I. Planet
  A. Earth
    1. Earth-based observatories
    2. Earth orbiters
  B. Planetary landers
  C. Planetary orbiters

II. Deep space probes (i.e., gravitationally unbound from all planets and satellites)

Figure 11 shows the object hierarchy of observing platforms.[17] The C++ code classes reflect this hierarchy. Each input data stream will contain relevant observing platform information. An appropriate observing platform object will encapsulate this information. Each type of platform object also encapsulates the necessary functionality (referred to as *methods*) to provide information needed to manipulate or transform data of the corresponding type (see Figures 5 and 11).

For example, planetary observing platform objects know how to precess and nutate coordinates to a specified epoch. Each base class contains parameters and functionality common to all subclasses derived from it. The derived classes contain only the additional or specialized parameters and functionality required to handle platforms of a specific kind. For example, since all planetary platforms have a basic precession and nutation capability, these methods reside in the base class PlanetPlatform. An EarthPlatform object automatically inherits all the functionality and data of PlanetPlatform. The EarthPlatform object therefore contains only additional abilities, data, or refinements, for ex-

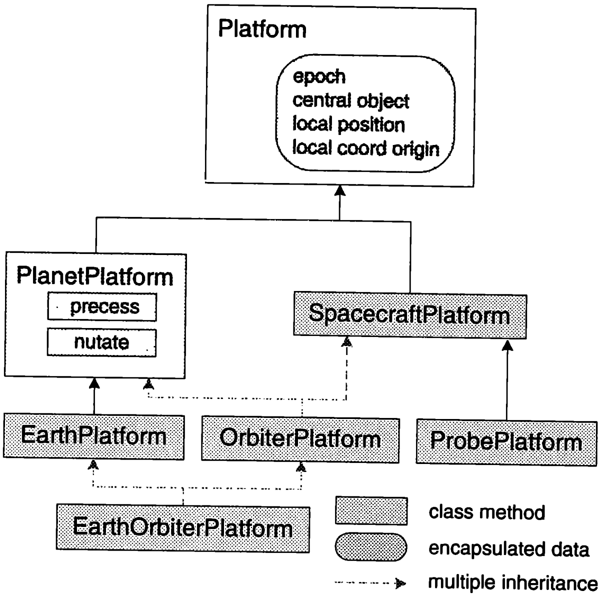


Figure 11 — Observing platform class hierarchy.

ample precession parameters specific to the Earth. Proper use of inheritance eliminates code duplication for common tasks in a natural and intuitive way. The inheritance mechanism is built into the C++ language and therefore requires no enforcement by or special discipline from the programmer.

Figure 11 intentionally shows only the major class types, in accord with the introductory nature appropriate to this Chapter. It is a simple matter to derive further specialized classes from the base classes shown. For example, one would derive a VikingOrbiter from OrbiterPlatform.

2.7.2. Observation Types

As previously mentioned, the various observation data types fall naturally into the two broad categories: timing and position. For reasons having mainly to do with datasets that are currently insufficiently large or insufficiently accurate to have a substantial effect on ephemeris accuracy, early versions of Newcomb will not include some of the observation types. Newcomb will include the following subset types:

I. Transverse (position)
    A. Optical observations
        1. Global positions
            a. Transit circle
        2. Differential positions

II. Radial (timing)
    A. Doppler observations
        1. Oneway
            a. Pulsars
            b. Spacecraft
        2. Twoway
            a. Radar
            b. Spacecraft
    B. Time delay observations
        1. LLR
        2. Radar
            a. Differential radar
        3. Spacecraft
            a. Single

Because extensibility is built into the design of Newcomb, adding further capabilities as they become necessary will involve minimal effort — there is no need, from a maintenance standpoint, to include capabilities that are anticipated to go unused for a long time. That is, with a good object-oriented design we do not have to worry so much about "making room" for anticipated future capabilities. Figure 5 shows the observation types hierarchy. Figure 12 shows the proposed corresponding object class hierarchy used in Newcomb.

Each type of input data stream will contain embedded type information, and instantiations of the appropriate data objects will handle the data. The

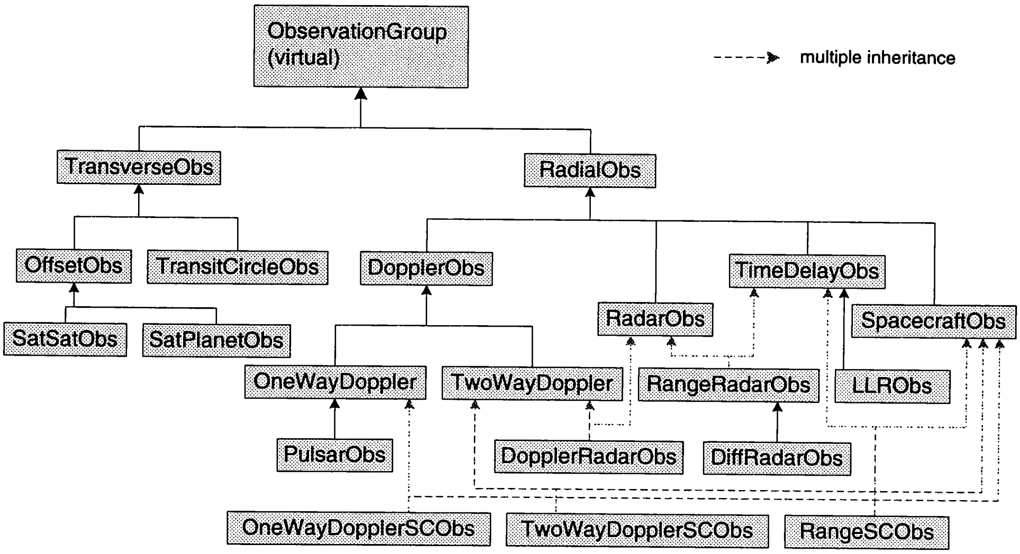


Figure 12 — Observational class hierarchy.

specific objects shown in Figure 12 encapsulate not only the corresponding observational data but also the functionality required to reduce that data type. For example, notice that all datatype objects have, via inheritance from the base class Observation, platform information and the ability to handle (say) aberration.

As with Figure 11, Figure 12 is intentionally not complete, especially regarding encapsulated data and method details. However, all the important base classes, and their inheritance dependencies, are shown.

3. SUMMARY

We have given a brief description of the field of high-precision modeling of solar system planetary and natural satellite motions. Motivations for high-precision ephemerides stem from — perhaps surprisingly to many — military as well as astronomical requirements. The latter category includes such areas as spacecraft navigation, celestial mechanics, occultation predictions, tests of General Relativity, etc. Several kinds of observations go into determining high-precision ephemerides — essentially, we use anything we can get our hands on. We have also discussed in broad terms the method of generating high-precision ephemerides, making use of both observational data sets and comprehensive models to solve for the "best" model parameter values.

Given that the extant first generation of high-precision ephemeris programs is antiquated, the U.S. Naval Observatory has begun development of a new, highly flexible ephemeris program called Newton. This modern program takes full advantage of design and programming techniques developed in the 1980s and early 1990s and now available as a mature set of technologies. An overview of the program design has been presented, which serves also to provide further insight into how a modern, high-precision solar system ephemeris can be generated, as well an indication as to some of the complexity of such an undertaking. It is relatively simple and straightforward to write a program that makes low-precision predictions. However, generation of a high-precision ephemeris is another matter altogether.

APPENDIX: TYPICAL PEP INPUT

As an example of the old-style program interface, following is a small excerpt taken from a typical PEP input file. Compare to Figure 7. This particular file (courtesy James Hilton) was used in the generation of ephemerides for the four largest asteroids for use in the Astronomical Almanac for the year 2000.

```
Ephemerides improvement for the first 4 asteroids.
 &NMLST1
 EXTPRC=   0,          $ Use hardware extended precision
 ICT(1)=  10,
 ICT(3)=   1,
 ICT(4)=   1,          $ Compute partial derivatives
 ICT(5)=  -1,          $ Do not used saved normal equations
 ICT(9)=   0,
 ICT(10)=  -2, ICT(11)= -2,
 ICT(12)=  2,          $  prediction or harmonic analysis
 ICT(34)=  3,
 ICT(39)=  1,
 ICT(50)=  1,          $ USE BROWN MEAN MOON IF NO IPERT
 ICT(80)=  0,
 JCT(13)=  1,          $ Use J2000.0 coordinates.
 JCT(33)= -1,          $ Use USNO UT1 and wobble
 JCT(27)=  1,          $ Use * commands
 JCT(28)=  7,
 MASS(1)= 6023600.D0, $ Use DE200 masses for the planets
 MASS(2)=  408523.5D0,
 MASS(3)=  328900.550000000047D0,
 MASS(4)= 3098710.D0,
 MASS(5)=    1047.35001090551827D0,
 MASS(6)=    3497.99999984177066D0,
 MASS(7)=   22960.000007059389D0,
 MASS(8)=   19314.0002382557432D0,
 MASS(9)= 130000000.238686755D0,
 MASS(10)=       0.012150581D0,
 MASS(11)=       2.239D9,
 MASS(12)=       9.247D9,
 MASS(13)=       8.7D10,
 MASS(14)=       7.253D9,
 MASS(17)=       1.849D11,
 AULTSC=      499.0047837D0,   $ AU in light seconds
 ECINC=        23.439281083D0, $ Use DE118 Obliquity
 PRMTER(47)= 0.0D0,            $ RA OF ASC. NODE OF BELT
 PRMTER(48)= 23.4433D0,        $ INCLINATION OF BELT
 PRMTER(49)= 2.9D0,            $ DISTANCE OF BELT FROM SUN
 PRMTER(50)= 8.773725302941010D-10,  $ MASS OF BELT
 PRMTER(81)= 0.0D0,
 MDSTSC=  0.,              $ MOON TAPE DISTANCE UNIT IN AU
 NBODY= 0, IPERT= 10,
 NUMOBT= 1,
 IOBS = 30,
 IOBS1= 14, IOBS2= 15,
$ EPS(3)= 100,
$ EPS(4)= 100,
 LPRM(1)= 11,  LPRM(2)= 12,  LPRM(3)= 14,
*OBJECT EARTH-ROTATION
```

```
 CON(22)= 5029.0966,
 CON(23)= 84381.4119,
*OBJECT 11
 NAME=  '  CERES ',
 INCND= 0, ITAPE= 31, NCENTR= 0, JTYPE=6,
 A=2.767121817D0,   E=0.07749262D0, INC=27.116375D0,
 ASC= 23.471566D0,   PER= 133.40890D0,   ANOM=2.08129D0,
  JD1=2378801, JD2=2450001, JD0=2444801,
 K(31)=1, K(32)=1, K(33)=1, K(34)=1, K(35)=1, K(36)=1, K(37)=1,
  K(38)=    1, K(39)=   1,   K(40)= 1,   K(41)= 1,   K(42)= 1,
K(43)= 1, K(44)= 1,
 K(61)=  1                         $ Include GR
 K(87)=   2, INT=    2,            $ INTERVALS
 K(88)=   2, K(89)=  6,            $ ADAMS-MOULTON, 7 TERMS
 K(91)=  -3, K(92)= -6, EPS(3)=1E-9  $ STARTING INTERVALS
  K(98)=  -500, K(99)=  0, K(100)= -1,   $ PRINT + TAPE; ORDI-
NARY EQNS OF MOTION
 KI=        1, 1, 1, 1, 1, 1, 1, 12, 13, 14,
 L=         1, 1, 1, 1, 1, 1,
*OBJECT 12
 NAME=  ' PALLAS ',
 INCND= 0, ITAPE= 32, NCENTR= 0, JTYPE=6,
 A=      2.771672932D0, E=     0.23398027D0, INC=   11.809637D0,
 ASC= 161.02570D0,      PER= 322.78775D0,    ANOM= 298.543057D0,
  JD1= 2379251, JD2= 2450001, JD0= 2449601,
  K(31)=    1, K(32)=   1,   K(33)= 1,   K(34)= 1,   K(35)= 1,
K(36)= 1, K(37)= 1,
  K(38)=    1, K(39)=   1,   K(40)= 1,   K(41)= 1,   K(42)= 1,
K(43)= 1, K(44)= 1,
 K(61)=   1,                      $ Include GR
 K(87)=   2, INT=    2,           $ INTERVALS
 K(88)=   2, K(89)=  6,           $ ADAMS-MOULTON, 7 TERMS
 K(91)=  -3, K(92)= -6, EPS(3)= 1E-9  $ STARTING INTERVALS
  K(98)=  -500, K(99)=  0, K(100)= -1,   $ PRINT + TAPE; ORDI-
NARY EQNS OF MOTION
 KI=        1, 1, 1, 1, 1, 1, 1, 11,
 L=         1, 1, 1, 1, 1, 1,
*OBJECT 13
 NAME=  '  JUNO  ',
 INCND= 0, ITAPE= 33, NCENTR= 0, JTYPE= 6,
 A=      2.670660949D0, E=      .25626106D0, ANOM= 156.782239D0,
 INC=  10.814499D0,   PER=  46.75209D0,   ASC=   11.27760D0,
 JD1=   2380151, JD2= 2450001, JD0= 2444801,

 K(31)=  1, K(32)= 1, K(33)= 1, K(34)= 1, K(35)= 1, K(36)= 1,
K(37)= 1,
 K(38)=  1, K(39)= 1, K(40)= 1, K(41)= 1, K(42)= 1, K(43)= 1,
K(44)= 1,
 K(61)= -1,
```

1 One arc second is equal to one 3600th of a degree. One milliarcsecond equals one one-thousandth of an arc second. To provide some context, consider that 1 arcsecond corresponds to 1/16 of an inch around the perimeter of the 1000-foot radius Observatory Circle here at USNO. How much is 50 *micro*arcseconds [the nomina accuracy of the proposed space astrometry mission FAME (http://aa.usno.navy.mil/FAME/)]? That is the angle subtended by the width of a typical strand of human hair as seen from a distance of *65 miles*.

2 http://adc.gsfc.nasa.gov/adc-cgi/cat.pl?/catalogs/1/1149A/

3 http://astro.estec.esa.nl/Hipparcos/hipparcos.html

4 For an informative introduction to the International Celestial Reference System (ICRS), see http://aa.usno.navy.mil/AA/faq/docs/ICRS_doc.html. The ICRF catalog and related information is available at
http://hpiers.obspm.fr/webiers/results/icrf/README.html.

5 From the Greek *ephemeros*, meaning *daily*. That is, a table of coordinates of a celestial body at specific times.

6 http://astro.estec.esa.nl/Hipparcos/hipparcos.html

7 http://aa.usno.navy.mil/FAME/

8 http://sim.jpl.nasa.gov/

9 http://astro.estec.esa.nl/SA-general/Projects/GAIA/gaia.html

10 James Hilton of USNO is the world's foremost expert in determination of asteroid masses. See http://aa.usno.navy.mil/hilton/asteroid_masses.htm

11 See the FAME homepage at http://aa.usno.navy.mil/FAME/

12 See the official Newcomb program web site at http://aa.usno.navy.mil/Newcomb/. As reviewed elsewhere in these Proceedings, Simon Newcomb (1835-1909) was a remarkable force in 19th century American mathematics and astronomy. He was Superintendent of the Nautical Almanac Office from 1877 to 1897, and he devoted much of his prolific career to (in his words)

> ...a systematic determination of the constants of astronomy from the best existing data, a reinvestigation of the theories of the celestial motions, and the preparation of tables, formulae, and precepts for the construction of ephemerides, and for other applications of the same results.

See also the biography page located at
http://www-history.mcs.st-and.ac.uk/~history/Mathematicians/Newcomb.html

13 See http://cfa-www.harvard.edu/~reasen/ssd.html for information about PEP.

14 http://aa.usno.navy.mil/Newton/

15 http://aa.usno.navy.mil/AA/

16 http://www.borland.com/bcppbuilder/

17 Arrows in Figures 11 and 12 point *from* derived classes *to* parent (also called *base*) classes. This is the standard notation.