Celest Mech Dyn Astr (2010) 108:107–124
DOI 10.1007/s10569-010-9296-0

ORIGINAL ARTICLE

A fast, accurate, and smooth planetary ephemeris retrieval system

Nitin Arora · Ryan P. Russell

Received: 8 September 2009 / Revised: 18 April 2010 / Accepted: 11 June 2010 /
Published online: 17 August 2010

Abstract Precise trajectory simulations typically require an ephemeris retrieval system, i.e. some mechanism to identify planetary body states and orientations at given times. However, the ephemeris systems most commonly used throughout industry and academia are, by design, general in their capabilities and application. Here, we introduce a new system called FIRE (Fast Interpolated Runtime Ephemeris) that is designed for custom trajectory applications that favor speed and smooth derivatives. The new system minimizes the overhead associated with ephemeris calls through the use of archived splines, a runtime ephemeris (stored in random access memory of the computer), and batch processing routines. Further, our approach naturally provides first and second time derivatives for a small additional computational cost. The derivative capability is particularly attractive for optimization and targeting where smooth and accurate derivatives are important. Relative performance comparisons with the Jet Propulsion Laboratory's Spacecraft Planet Instrument C-matrix Events ephemeris system show typical speed improvements of approximately two orders of magnitude (250 times) for various state and orientation calls. Performance comparisons for high fidelity trajectory propagations are also considered and a factor of 70 in performance increase is achieved for typical cases. The new tool has potential value to any high precision application or software requiring fast, accurate, and smooth ephemeris data.

Keywords Ephemeris generation · FIRE · SPICE · Time derivatives

1 Introduction

Solar system ephemeris body states and orientations are consistently used in a variety of aerospace applications [for example Jacobson and French (2004), Giampieri and Dougherty

N. Arora (✉) · R. P. Russell
Guggenheim School of Aerospace Engineering, Georgia Institute of Technology,
270 Ferst Drive, Atlanta, GA, USA
e-mail: N.arora@gatech.edu

R. P. Russell
e-mail: ryan.russell@gatech.edu

(2004), Russell and Lam (2006), Russell and Lara (2007), Aiello (2005), Lo et al. (2004)]. The importance of the solar system data, robust reliability of the current ephemeris systems (like JPL's SPICE Acton 1996) and the need for precise trajectory computations have led to wide spread use of ephemerides amongst scientists, engineers, and their associated software. The current state-of-the-art ephemeris system in wide use today for high precision applications is the SPICE system provided by JPL. Similar SPICE related products from JPL such as the DE405 customized routines by Miles Standish (1997) are also commonly used. The SPICE system has not been designed for speed and fast ephemeris data retrieval but rather serves a much broader purpose. The SPICE system has a large collection of routines which can be used to read SPICE ephemeris files (for natural bodies and spacecrafts). The system also provides information regarding the derived observation geometry, such as altitude, latitude/longitude, and lighting angles of these bodies.

While SPICE and its derivative products are well maintained and provide invaluable capabilities to users around the world, it is well known that ephemeris calls are generally the bottleneck to speed improvements for precise applications. Typical trajectory simulation applications including optimization, differential correction, orbit determination, and Monte-Carlo analysis often require thousands or more trajectory propagations using common force models and time spans. Considering each single iterate may easily require millions of calls to an ephemeris system, an extraordinary amount of computational resources is wasted in the presence of any unnecessary overhead. While a broad spectrum of applications are indifferent to the computational burden of general ephemeris calls, many applications are simply bogged down. However, the extra cost has typically been considered an acceptable trade for the precision force model afforded by the accurate ephemerides. Therein lays the motivation of the current work: In this study we propose a new custom ephemeris system that maintains the heavily relied upon accuracy, yet eliminates or substantially reduces the typical computational burdens associated with ephemeris calls.

We propose FIRE, a fast and efficient ephemeris generation system. FIRE is particularly suited for problems that favor higher speeds and smooth derivatives, such as trajectory optimization (Russell 2007, 2004; Hull 1979; Sauer 1989), orbit determination (Campbell and Speckman 2004; Russell and Lara 2007; Koon et al. 1999; Lo et al. 2004) and Monte Carlo sensitivity analysis (Desai and Cheatwood 1999; Aiello 2005). The new system generally favors speed in sacrifice of relatively more memory. The basic structure consists of an internally formatted, archived ephemeris called ACE (created from an already established ephemeris such as SPICE) and a problem dependent, dynamically created runtime ephemeris called the RACE. Implementing a runtime ephemeris stored directly in RAM significantly reduces the overhead associated with the multiple runtime ephemeris calls.

Various numerical techniques such as a fast and efficient cubic spline interpolation for ACE generation, FFT for identifying base frequencies, batched runtime calls (batch calls) to reduce overheads, adaptive tree structure evaluation to eliminate redundancy, and numerically stable floating point algorithms have been implemented. FIRE also provides the user with continuous and analytic first and second time derivatives for all the ephemeris states (position, velocity) and orientation matrices. This valuable derivative feature is absent from SPICE routines (like "SPKEZ","SPKEZP" and "PXFORM"), where derivatives are available only through the expensive and less accurate numerical differencing method.

In this study, we use SPICE as the benchmark for our performance comparison as it is arguably the most widely accepted ephemeris retrieval architecture publicly available. Extensive performance comparisons for direct position and velocity calls show that FIRE is "70–250" (approximately) times faster (depending upon the number of bodies and type of call). If only the positions of bodies are evaluated, then FIRE performs "44–197" (approximately) times faster. Further, a performance increase of "150–250" times is observed if we also compute the orientation matrix along with the position and velocity vector.

When implemented in trajectory integration problems, the FIRE integration is demonstrated to be "50–70" (approximately) times faster than the same integration using SPICE. For similar applications using SPICE, there is little motivation for improving general algorithm efficiency because ephemeris calls dominate such a large fraction of the total computational burden. In the case of FIRE, users are free to experience the full benefit of improving the efficiency of their custom applications. With the times required for ephemeris calls reduced by up to two orders of magnitude, users can no longer blame the ephemeris for major speed bottleneck!

FIRE currently uses the SPICE ".bsp" and ".tpc" files (Acton 1996) to generate the archived cubic spline ephemeris (ACE), although FIRE could easily be tailored to use any established ephemeris as input. We emphasize that the FIRE system is not intended to be a replacement for the full functionality of SPICE or any other established ephemeris system. It acts as a wrapper over an already established ephemeris like SPICE and is intended to replace only the state and orientation ephemeris calls for custom applications that may benefit from FIRE's speed and smooth derivatives.

2 FIRE architecture and core numerical computation

This section gives an overview of FIRE's main architecture and its core routines.

The FIRE architecture is based upon the premise of increasing speed and maintaining accuracy while sacrificing memory as efficiently as possible. It is written in Fortran 90 using a modular approach to ease its implementation and expandability. The main architecture can be broadly divided into three sub-systems:

1. Archived Cubic spline Ephemeris (ACE)
2. Runtime Adaptive Custom Ephemeris (RACE)
3. RACE loading and Runtime batch processing routines

The above mentioned sub-systems are computationally separate from each other, each containing its own set of local core routines. A standard input method from a text file corresponding to a "namelist" local to that system has been followed, thereby imparting robustness and flexibility to the code.

Even though the SPICE naming convention has been adopted for consistency, we can create our own naming convention and our own custom bodies or spacecraft by changing the relevant inputs. Note, however, for the tree navigation algorithm, we do require that the "tree trunk' is the main solar system barycenter defined by a body number of zero (see Fig. 1). Details on the tree structure are discussed later.

2.1 Archived cubic spline ephemeris (ACE)

2.1.1 ACE computation

Cubic spline interpolation is actively being used in various fields like Robotics (Lin et al. 1983; Huang et al. 2001), Numerical integration (Spedding and Rignot 1993), Image Interpolation (Unser 1993) and Animation (Shoemake 1985). The ephemeris states of spacecraft and planetary bodies are well behaved functions. Hence, cubic splines can easily achieve the desired accuracy and efficiency required for interpolation. Further, cubic splines naturally

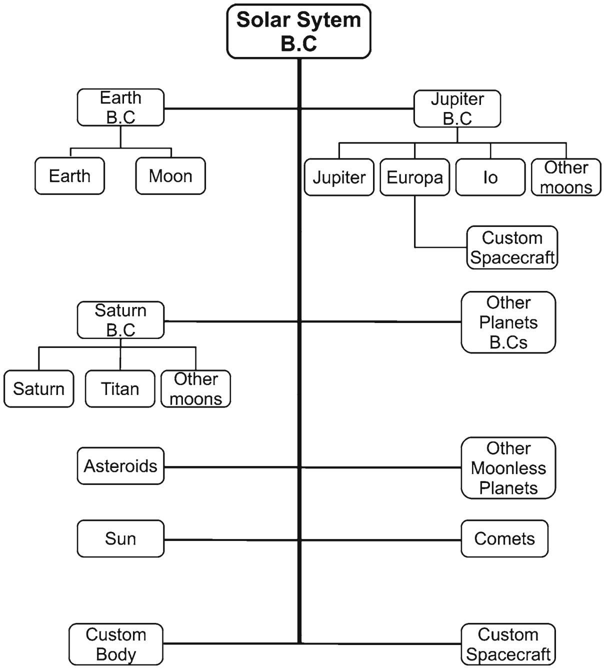

Fig. 1 Sample tree diagram

provide continuous time derivatives up to second order. The second derivative will have kinks at the knots but remain continuous while the first derivatives are smooth and continuous across the full domain. Note that we interpolate on velocities separately instead of using the first derivative of the position in order to maintain accuracy and continuous second derivatives. We choose the cubic splines primarily because they are remarkably fast to compute in comparison to other popular, higher order splines such as Chebyshev polynomials and B-splines.

Even though SPICE is already based on a Chebyshev polynomial, we interpolate on the interpolated ephemeris for ease of implementation. We note that errors in our interpolation of SPICE are on the same order or less than those interpolation errors published by SPICE. Nonetheless, future works include interpolating on the raw currently unpublished data that SPICE uses for its interpolation.

ACE is generated by implementing an efficient cubic spline algorithm to interpolate the data generated by an established ephemeris. We adopt a clamped cubic spline method well suited for accuracy and smoothness (deBoor 2001). The end conditions require the first derivative of the interpolation variable at the first and the last knot. These derivatives, for each state and orientation angle being interpolated, are calculated using a fourth order numerical difference scheme. For spline computation, the linear, tridiagonal system is given by Eq. 1.

$$A * \mathbf{M} = \mathbf{V} \tag{1}$$

where A is a tridiagonal matrix given by Eq. 2:

$$A = b \begin{bmatrix} 3.5 & 1 & & & & 0 \\ & 1 & 4 & 1 & & \\ & & 1 & 4 & 1 & \\ & & & \cdot & \cdot & \cdot \\ & & & 1 & 4 & 1 \\ 0 & & & & 1 & 3.5 \end{bmatrix} \tag{2}$$

The linear system given by Eq. 1 is solved for $M(1)$ to $M(N-1)$ by using a numerically stable tridiagonal matrix algorithm (Conte and deBoor 1972). Further, $M(0)$ and $M(N)$ are evaluated using the end conditions from Eqs. 3 and 4. Note: $M(1)$ to $M(N-1)$ values are solved first and then $M(0)$ and $M(N)$ are evaluated, subsequently.

$$M(0) = \frac{3}{h} * \left[\frac{(V(1)-V(0))}{h} - V\dot{(0)}\right] - \frac{M(1)}{2} \tag{3}$$

$$M(N) = \frac{3}{h} * \left[\frac{(V(N)-V(N-1))}{h} - V\dot{(N)}\right] - \frac{M(N-1)}{2} \tag{4}$$

After computing $M(0)$ to $M(N)$, the spline coefficients (a, b, c, d) are evaluated using the Eqs. 5–8 (where h is the knot to knot distance)

$$a = V(i) \tag{5}$$

$$b = \frac{(V(i+1)-V(i))}{h} - \frac{h}{6} * (2 * M(i) + M(i+1)) \tag{6}$$

$$c = \frac{M(i)}{2} \tag{7}$$

$$d = \frac{(M(i+1)-M(i))}{(6*h)} \tag{8}$$

And finally the spline is evaluated using a numerically efficient form given by Eq. 9 (where ξ is knot to current time distance and ζ is a dummy interpolation variable or state vector) and 1st and 2nd derivatives are calculated by Eqs. 10 and 11, respectively.

$$\zeta = a + \xi * (b + \xi * (c + \xi * d)) \tag{9}$$

$$\dot{\zeta} = b + \xi * (2 * c + 3 * \xi * d) \tag{10}$$

$$\ddot{\zeta} = 2 * (c + 3 * \xi * d) \tag{11}$$

The distance between knots, while performing interpolation, directly affects the accuracy and smoothness of the interpolation (deBoor 2001). In our case, the distance between knots is the minimum time step required for successful interpolation which is driven by the frequency of the body having the smallest orbital period for a particular system of related bodies. Take the Jupiter system as an example containing the Jupiter barycenter, Jupiter itself, and its major moons. Io has the smallest period of all the major satellites. In this case, the frequency associated with Io will of course be observed in the ephemerides of all bodies in the Jupiter system. Therefore, to capture the high resolution motion, we assign all bodies a time step commensurate with Io. In fact, to obtain the final time step for the whole Jupiter system, we divide Io's period by NPR (number of points per rev), a user defined accuracy parameter. In our experiments, we find that NPR values in the range of 30–60 gives good results.

This time step corresponding to the knot separation is given to all the moons of Jupiter according to the "*MINGM*" criteria. The "*MINGM*" criteria allows a body to be considered if its *GM* value is above a predefined value. Decreasing "*MINGM*" incorporates smaller bodies in the time step computation hence increasing the accuracy of interpolation. A body with *GM* less than the "*MINGM*" is assumed to have no effect on the states of the remaining bodies in the system. Therefore its frequency signature is deemed unimportant to the dynamics of

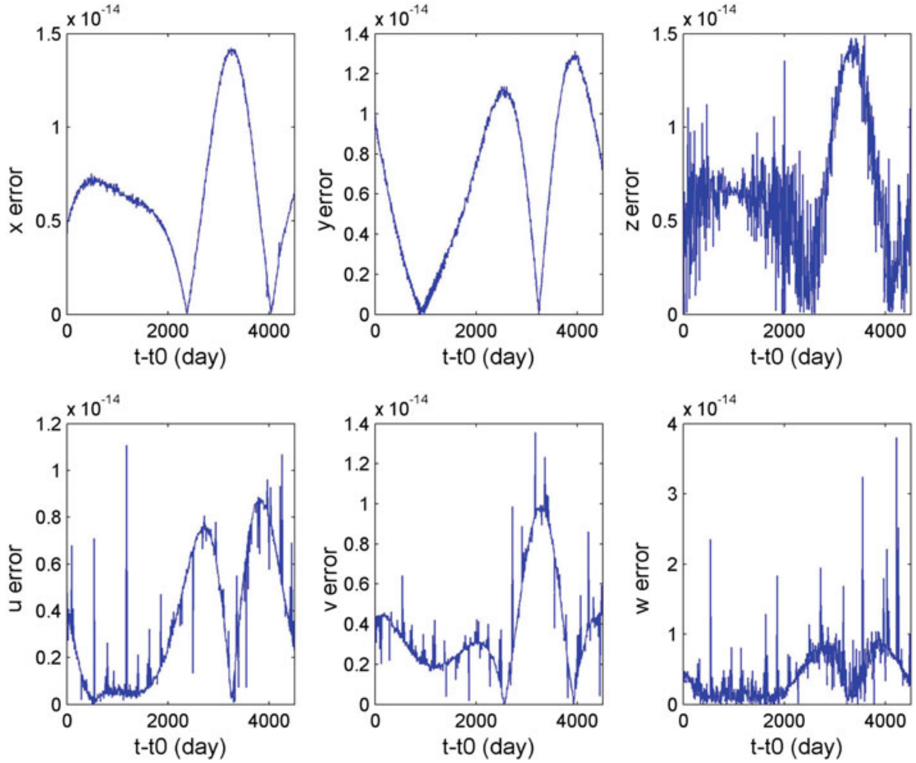

Fig. 2 Normalized error in Jupiter's barycenter states

the problem. As we will discuss later regarding the rotation states, the FFT method is an alternative approach to choosing the knot separation.

The ability to attain greater accuracy by either increasing the number of points or by decreasing the *MINGM* defined value can be misleading. For smooth data, if the knots are too densely populated, then the interpolating polynomial will use its high-degree coefficients, in combination with large and almost precisely canceling combinations. This may cause the interpolating polynomial values to oscillate between its constrained points, and hence may produce spikes which affect the well boundedness of errors.

The accuracy of the spline derivatives is typically less than the accuracy of their immediate integrals. Hence, the normalized accuracy for the zeroth, first and second derivative is generally on the order of 1E-8 (normalized dimensionless value), 1E-6 and 1E-4 respectively for bodies and is of the order of 1E-14 (normalized dimensionless value), 1E-11 and 1E-7 for barycenters. The normalizing factor (Nr) is calculated as per Eq. 12. For our purposes, it is very important that the zeroth derivative be accurate. Figure 2 shows a representative error plot for position and velocity components along with their first and second derivatives for Jupiter's barycenter.

$$Nr = \max(\zeta_i) \quad \text{for} \quad i = 1, \ldots, ns \tag{12}$$

Note that the SPICE data, to the authors knowledge, is not necessarily smooth to second order across the full time domain. Discontinuities in derivatives may exist at junctions in the SPICE interpolating functions. While our "error" plots presume that the SPICE numerical

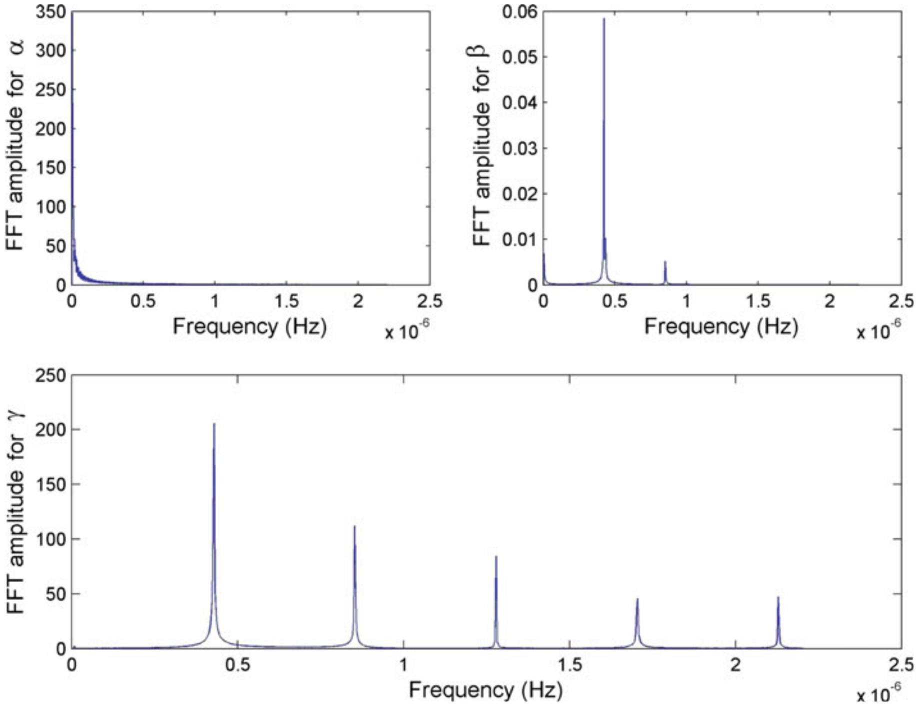

Fig. 3 Fast fourier transformation for Moon's three Euler angles

derivatives are truth, one could argue that the FIRE derivatives are closer to the truth in the absence of numerical differencing. In fact, for numerical purposes regarding targeting and optimization, it is more important for the derivatives to be self consistent rather than absolutely correct.

We choose Euler angles for orientation interpolation because they perfectly preserve orthogonality in the resulting rotation matrix. Other methods such as Quaternions (Shuster 1993), and Rodrigues Parameters (Tsiotras et al. 1997) were investigated extensively in order to avoid the computationally expensive trigonometric calls. However, it is well known that interpolating quaternions introduces complications including sign ambiguities and or loss of orthogonality (Shoemake 1985).

We note that the SPICE interface with orientation data is a common rotation matrix between user defined frames. In our custom ephemeris we are currently limited to obtaining rotation matrices between the IAU defined body fixed coordinate systems and the base ecliptic J2000 inertial frame (Seidelmann et al. 2007). A rotation matrix between two general frames is achievable with FIRE using two separate rotation calls and a matrix transpose and multiplication.

FFT's can be easily used to catch frequencies of a periodic system (Cooley et al. 1969). In this paper, an efficient and numerically stable FFT algorithm (Hou 1987; Burrus and Eschenbacher 1981) is used to calculate the minimum step size required for interpolation of Euler angles. For a given angle, the frequency corresponding to the largest spike in the Fourier transformed space is recorded and the corresponding time step is calculated. This provides us with a reliable, robust, and efficient interpolation method, even for complex and subtle body dynamics. A representative FFT plot for the three Euler angles of the Moon are shown in Fig. 3. Here, α and β represent the pointing direction of the north pole of the body

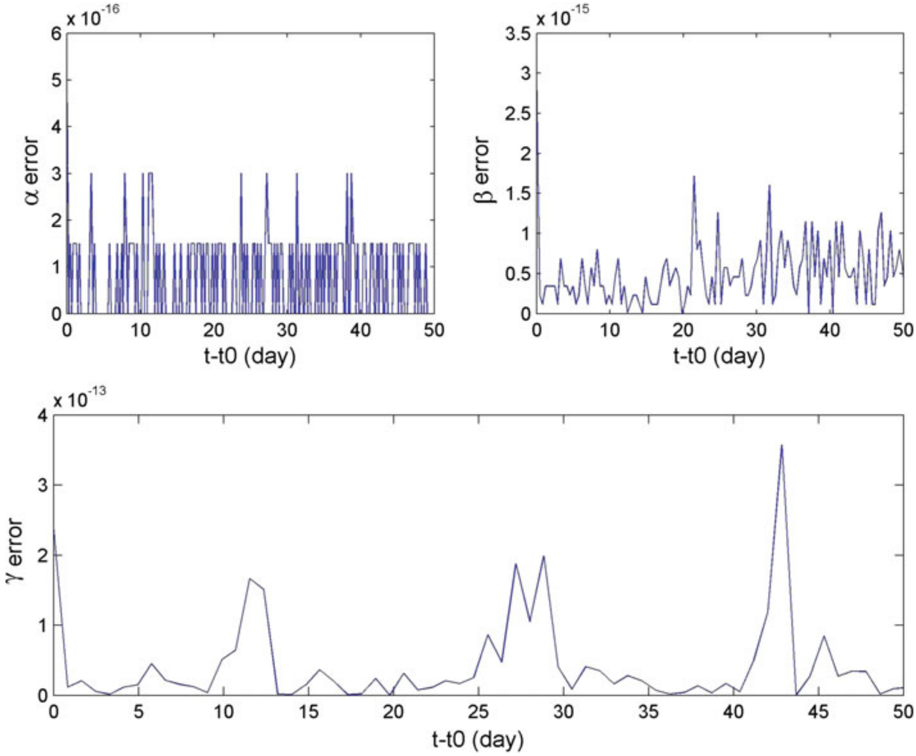

Fig. 4 Relative normalized error in Titan's Euler angles

in question, while γ represents the angular location of the prime meridian. The accuracy of the interpolation is of the order of 1E-15 (non-dimensional error) for both the slow period angles and 1E-13 (nondimensional error) for the fast moving angle. Relative normalized error plots for Euler angles of Titan (Neptune's largest moon) for 50 days are shown in Fig. 4. A 3-1-3 rotation matrix (R) has been adopted via IAU Standards (Seidelmann et al. 2007). The expression for the time derivatives for R in terms of the time derivatives of α, β, and γ are straightforward but left out of the text for brevity. We obtain the time derivatives of the states directly from the interpolation. The actual implementation for the time derivative of R is optimized for serial access, favoring speed and re-use of data. We note that currently the FIRE system is limited only to natural body orientation data retrieval.

2.1.2 ACE storage

ACE is stored in a structured format, consisting of numerous gravitational subsystems. Each body of a subsystem, has up to 3 associated binary files: one for position , one for velocity, and one for rotation. The binary files contain the dependent variables and their second derivatives at all the knot points for each interpolation variable. The spline coefficients are calculated at runtime using optimized implementations of Eqs. 5–8. The independent knot (time) values are stored in an efficient manner to avoid redundancy while preserving speed. An archived ephemeris (ACE) is typically generated when using FIRE for the first time or when there is major update in the underlying ephemeris system used to generate ACE.

2.2 Runtime adaptive custom ephemeris (RACE)

RACE is a binary, dense array, whose main purpose is to significantly increase the speed of run-time ephemeris state and orientation calls by eliminating the overhead arising from invoking the archived spline. RACE also provides the user with a portability option to store as a binary file the problem specific RACE for reuse later. Hence multiple RACEs can be generated and stored for different classes of problems. This strategy leads to the idea of parent (ACE) and child (RACE) ephemerides.

By default, each body within the RACE has its state relative to its own native barycenter. A user defined input can modify the RACE to change the interpolation center for the bodies present. Typically we choose the new interpolation center as the center of integration for a trajectory propagation and may lead to 50–150% performance improvement at runtime. The memory requirement of the runtime ephemeris typically varies from 5 to 100 MB (approximately) depending upon the accuracy of ACE and the time span and number of body states required for the problem.

The number of knots for all the bodies is equal to some 2^k value (where k is an integer). Hence, knowing the body and the corresponding particular interpolation variable having the largest value of k provides a mechanism to uniquely identify the appropriate knots for each state and orientation in a batch call without performing expensive searches in the RACE at runtime. This algorithm requires one "double" to "integer" conversion and just a few elementary floating point operations, instead of a linear or a binary search algorithm.

2.3 RACE loading and runtime batch processing routines

The third sub-system is invoked at runtime and performs the following two operations:

1. Dynamic allocation of the RACE in memory
2. Computing the state vectors, the rotation matrices and any derivatives by evaluating the tree navigation algorithm (see Fig. 1)

In contrast to the capabilities of SPICE, the new system employs a single batch call for interpolation of all of the necessary states and orientations at a particular time.[1] The term "batch call" refers to calling all the states of multiple bodies in one call rather than calling the routines over and over for each body state and orientation. These batch calls significantly reduce the overhead associated with the common initialization for each body.

In cases when an ephemeris is used in the force models for optimization and or targeting routines, continuous first and second time derivatives may be necessary for positions, velocities, and rotation matrices. Keeping this in mind, FIRE also provides the user with set of flagged routines which can be invoked to get the first or both the first and the second derivatives of position, velocity, rotation, or any combination of these via only one batch call. This derivative data can be extremely useful for problems which are very sensitive, like trajectories with multiple close flybys or multiple gravitating bodies (Lam and Hirani 2006; Whiffen 2006). The RACE is general and users are free to call for states and derivatives relative to any center.

Navigation of the tree structure (see Fig. 1) is required at runtime if the user requested reference center is different from the RACE interpolation center. In the case of trajectory integration applications, this reference center is typically the problem specific center of integration. Again for a general system like SPICE, there is overhead associated with evaluating

[1] We note that the Matlab version of SPICE, called MICE, includes batch calls although we suspect it is simply a wrapper around the un-batched SPICE.

Table 1 FIRE runtime routines and their function

Routine name	Function performed
FIRE_P	Computes xyz states
FIRE_PV	Computes xyz and uvw states
FIRE_PVR	Computes xyz, uvw states plus the rotation matrix
FIRE_R	Computes the rotation matrix
FIRE_PR	Computes xyz and the rotation matrix
FIRE_PD	Computes xyz states, and corresponding derivatives
FIRE_PVD	Computes xyz, uvw states, and corresponding derivatives
FIRE_PVRD	Computes xyz, uvw states, the rotation matrix, and corresponding derivatives
FIRE_DR	Computes the rotation matrix, and corresponding derivatives

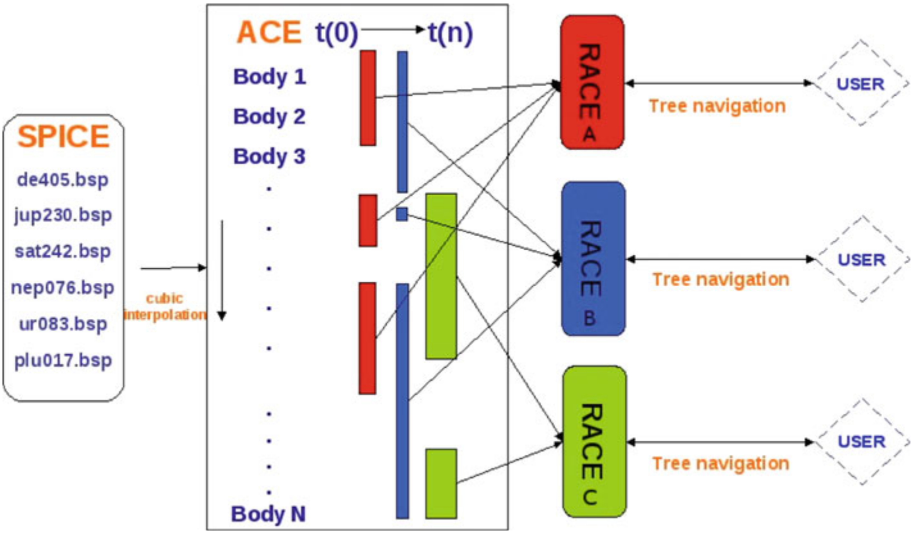

Fig. 5 Overview of the FIRE system

the tree at runtime for every call and every body. In the FIRE architecture, the batch calls provide all the necessary states with respect to one common center. An efficient tree navigation algorithm is implemented, which removes any redundant calls without sacrificing the numerical stability and accuracy of the state being calculated.

FIRE, has in total nine different routines (shown in Table 1) which can be called at runtime, five of these give states and orientations while the remaining four additionally provide derivatives. Various combinations of “P, V, R and D” are used to call for appropriate data. For example, if only position and rotation data is needed then the user would call the routine “FIRE_PR”. This provides a user friendly interface that is consistent with the existing functionality of SPICE. We emphasize again that the derivative calls are currently unavailable in SPICE.

The overview of the complete FIRE system is given in Fig. 5.

3 Performance

JPL's ephemeris generation system (SPICE), is used for benchmarking the performance of the FIRE system. The system configuration used for benchmarking is an Intel dual core processor (2.6 Ghz) with 4 GB of RAM. The benchmarking is done in two ways. First, we directly call the routine for a certain fixed number of calls and record the relative performance gain. Second, we perform two typical high fidelity trajectory simulations for the Earth-Moon-Sun (EMS) system and the Saturn system with Saturn and its nine moons along with Jupiter and the Sun. We take into account accelerations due to third body perturbation and gravity field of the central body (via spherical harmonics).

The code is compiled and linked on the Intel Fortran Compiler for Windows version 11.0 and has been subjected to default "-fast" compiler optimizations.

3.1 Performance with direct routine calls

In this comparison $3*2^{24}$ (approximately 50 million) calls with random time inputs are used to evaluate the performance of SPICE and FIRE. We use the SPICE routines "SPKEZP, SPKEZ" for computing position and velocity and "PXFORM" with integer inputs for generation of the rotation matrix.

Two gravitational systems are studied for this comparison: the Earth-Moon-Sun (EMS) system and the Jupiter-Moons-Saturn-Sun (JMSS) system. The EMS system has only three bodies passed at once, hence highlighting the performance gain due to the use of RACE, the runtime matrix. The JMSS system has eight bodies and thereby quantifies the performance of the full capability of the tree navigation algorithm, batch calls, and RACE.

A type defined bar graph system is used to illustrate the speed-up of FIRE over SPICE. As shown in Tables 2 and 3, there are a total of six types of ephemeris calls, each with different performance characteristics due to different path lengths required in evaluation of the tree algorithm (Fig. 1).

From Fig. 6a–d we demonstrate dramatic performance gains from FIRE in comparison to SPICE. We note that the most expensive call requires a full navigation of the tree algorithm. Even in this worst performing case, the FIRE routines are 45–65 times faster than SPICE. The typical FIRE call where the batch routines are fully exploited and most of the bodies share a common interpolation center is demonstrated to be as much as 250 times faster.

Often a typical routine call includes the necessary position and velocity data for all the bodies along with rotation data for some bodies. Hence, to demonstrate full performance

Table 2 Various cases for Earth-Moon-Sun performance comparison

System	Earth-Moon-Sun
Type1	Moon wrt Earth-Moon barycenter (native barycenter call)
Type2	Earth and Moon wrt Earth-Moon barycenter (native barycenter call)
Type3	Earth wrt Moon (native system call)
Type4	Earth and Moon wrt Solar-System barycenter (half tree evaluation)
Type5	Earth, Earth-Moon barycenter and Sun wrt Moon (typical call)
Type6	Earth wrt Sun (full tree evaluation)

Table 3 Various cases for Jupiter-Moons-Saturn-Sun performance comparison

System	Jupiter-Moons-Saturn-Sun
Type1	Jupiter wrt Jupiter barycenter
Type2	Io, Europa, Ganymede, Callisto wrt Jupiter barycenter
Type3	Io, Europa, Ganymede, Callisto wrt Io
Type4	Io, Europa, Ganymede, Callisto wrt Solar-System barycenter
Type5	Io, Europa, Ganymede, Callisto, Jupiter, Sun, Saturn barycenter wrt Europa
Type6	Jupiter wrt Saturn barycenter

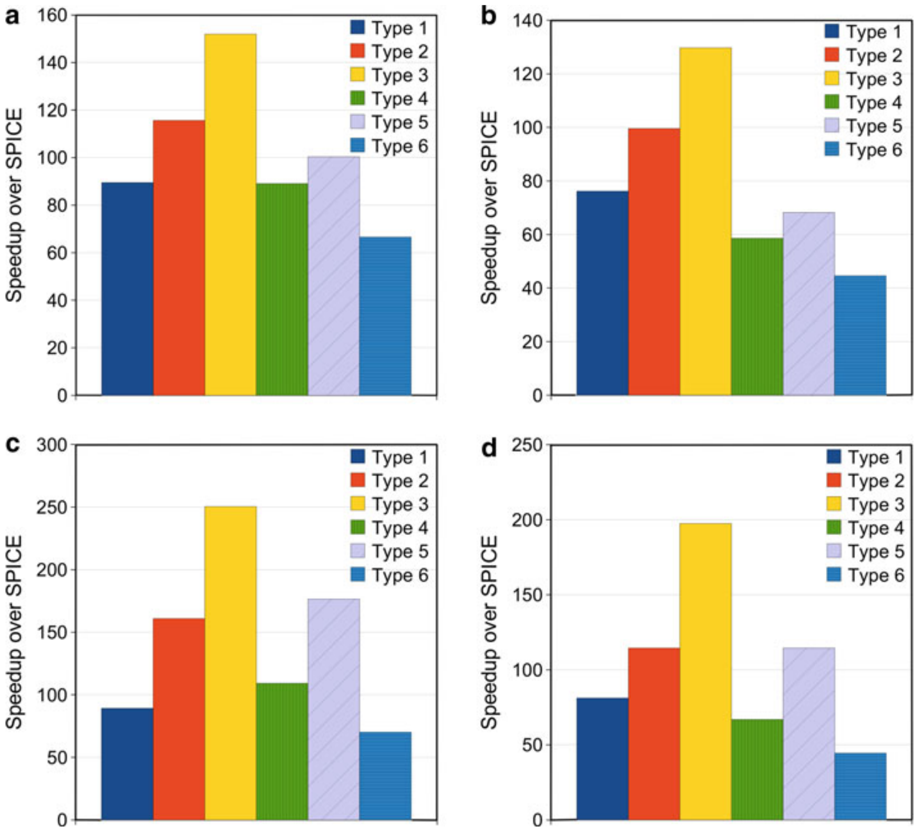

Fig. 6 **a** Speedup factor for EMS system (pos). **b** Speedup factor for EMS system (pos-vel). **c** Speedup factors for JMSS system (pos). **d** Speedup factors for JMSS system (pos-vel)

capability of FIRE, these calls have been again evaluated along with rotation matrices for Earth in the first system and for Io in the second system. Also, we include a test with only rotation calls to a couple of bodies in the Jupiter system for completeness. Accordingly, Tables 4 and 5 show the performance for both the tests respectively. Hence for the FIRE performance the typical call is around 150–200 times faster when rotation states are included. We note that the huge improvements when including rotation states may be over-shadowed by

Table 4 Typical performance comparison ratio table for Earth-Moon-Sun system

Type	Frame/center	SPICE time (s)	FIRE time (s)	Speedup
Typical call (pos and rot)	EclipJ2000/Earth	3066.63	20.17	152

Table 5 Typical performance comparison ratio table for Jupiter-Moons-Saturn-Sun system

Type	Frame/center	SPICE time (s)	FIRE time (s)	Speedup
Typical call (pos and rot)	EclipJ2000/Io	5385.97	26.87	200
Rotation call only	EclipJ2000	3443.91	14.11	244

the heavy computational burdens of spherical harmonics or polyhedral potential calculations that typically accompany orientation state needs.

A distinguishing feature of the FIRE system is its ability to provide the first and second time derivatives of all the ephemeris states. The states plus first derivative calls perform nearly equal to position-velocity calls only, and the combined states plus first and second derivative calls require approximately 1–5% more time than the first derivative only calls. We emphasize again that numerical derivative computation using SPICE requires additional state calls (typically 2 calls for central difference approximation), and the resulting derivatives are substantially less accurate due to round off errors.

We also compared SPICE with the DE405 customized routines by Miles Standish (Standish 1997), called PLEPH. It was found out that generally PLEPH (using typical max speed optimizations) is approximately 6 times faster than SPICE. It should be noted that PLEPH only works with DE405 ephemeris data, and thereby has a limited scope.

3.2 Performance during trajectory integration

Beyond the raw performance using direct calls only, we now proceed to a more realistic comparison like that of spacecraft trajectory propagation. The relative performance gains will of course not be as impressive as the previous cases because of the integration and force calculation being done alongside ephemeris calls.

Two trajectory propagations are considered. One for the Earth-Moon-Sun system and the other for the Saturn-Moons-Jupiter-Sun system. Spherical harmonics gravity fields of various resolutions about the central body are also considered.

For integration, a variable step 8th order with 7th order error control, Dormand and Prince integrator (Prince and Dormand 1981) set to a unitless tolerance of 1E-14 is used. Ephemeris rotation calls are only included during the propagation when the non-spherical gravity field acceleration is included.

3.2.1 Earth-Moon-Sun (EMS)

For this comparison, a trajectory propagation of a mid altitude Earth orbiter has been performed taking into account the third-body perturbing effects of Moon and the Sun. This integration is performed for a period of 100 days. The initial conditions for this propagation listed in Table 6. The size of RACE for this case was approximately 6 MB. Various Earth gravity fields evaluted in the propagation are given in Table 7.

Table 6 Initial condition (body-fixed frame at epoch) for EMS trajectory

Orbital parameter	Value
Semi-major axis (a)	7500 (km)
Eccentricity (e)	0.074
Inclination (i)	35 (deg)
Argument of periapsis (ω)	9 (deg)
Longitude of ascending node (Ω)	20 (deg)
True anomaly at epoch (ν)	0 (deg)
Epoch (t0)	2454477.50 (JD)

Table 7 Gravity Field at Earth

Case	Acceleration combination
1	Two body
2	2 by 0 (J2 only) + two body
3	10 by 0 (J2-J10) + two body
4	50 by 0 (J2-J50) + two body
5	10 by 10 (10 degree + order field) + two body
6	50 by 50 (50 degree + order field) + two body

Table 8 Performance comparison ratio table for EMS trajectory propogation

Cases =>	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
SPICE Time (s)	30.24	59.97	60.52	62.38	66.18	211.17
FIRE Time (s)	0.61	1.01	1.46	4.12	6.03	135.82
Speedup (factor)	50	60	41	15	11	2

Table 8 shows the comparison between FIRE and SPICE for different types of acceleration. Accuracy and performance of the numerical integrations are the two main criteria for comparison. Results of the trajectory integration show that FIRE performs 60 times faster than SPICE for the typical case (third body perturbations + J2 term). Also, as expected the speedup decreases as the resolution of the spherical harmonics field is increased. The final state vector of FIRE is off by 15 m (approximately) when compared to the final state vector obtained by using SPICE for a typical 100 day trajectory. This is well within acceptable limits consdering the spacecraft trajectory is highly perturbed and it makes hundreds of revolutions (approximately 1336) around the Earth during the integration.

Figures 7 and 8a show the evolution of the orbital elements (for 20 days) and the resulting trajectory for the fully perturbed plus 10 by 10 gravity field case. Figure 8b gives the instantaneous position differences between FIRE and SPICE propagations. We note that the orbital elements are reported in the body-fixed frame at epoch.

3.2.2 Saturn-Moons-Jupiter-Sun (SMJS)

Here we consider the case of Saturn, its nine moons, Jupiter and the Sun (SMJS). A highly elliptical orbit with an apoapsis of 1518900 km (near Titan) and periapsis of 141100 km (close

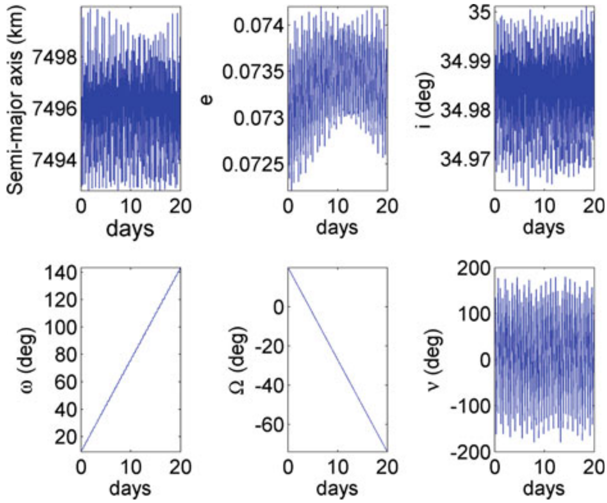

Fig. 7 Evolution of orbital elements for EMS system

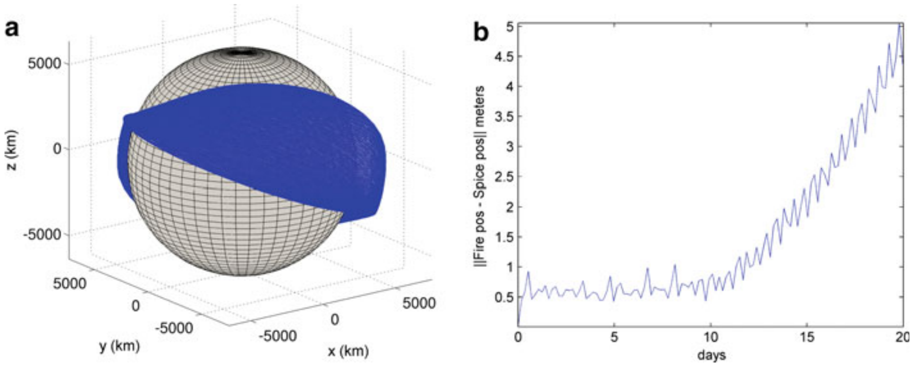

Fig. 8 **a** Propagated trajectory in EMS system. **b** SPICE and FIRE position difference (EMS)

Table 9 Initial condition (body-fixed frame at epoch) for SMJS trajectory

Orbital parameter	Value
Semi-major axis (a)	830000 (km)
Eccentricity (e)	0.83
Inclination (i)	9 (deg)
Argument of periapsis (ω)	2 (deg)
Longitude of ascending node (Ω)	11 (deg)
True anomaly at epoch (ν)	0 (deg)
Epoch (t0)	2454477.50 (JD)

to Mimas) around Saturn is selected for this propagation. This orbit enables us to capture the gravity signature of all the nine moons of Saturn, the Sun, and Jupiter. The trajectory propagation is carried out for 200 days. We only consider an 8 by 0 gravity field for Saturn. The initial conditions are given in Tables 9 and 10 and the results are given in Table 11.

Table 10 Gravity field at Saturn

Case	Acceleration combintion
1	Two body
2	2 by 0 (J2 only) + two body
3	8 by 0 (J2-J8) + two body

Table 11 Performance comparison ratio table for SMJS trajectory propogation

Cases =>	Case 1	Case 2	Case 3
SPICE Time (s)	61.79	81.07	81.79
FIRE Time (s)	1.06	1.16	1.36
Speedup (factor)	59	70	60

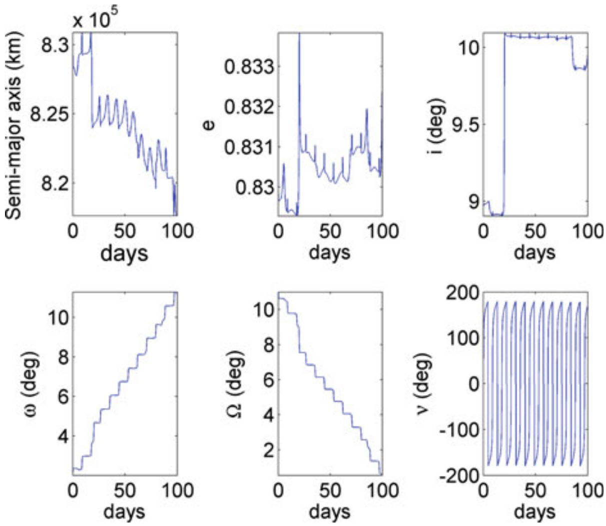

Fig. 9 Evolution of orbital elements for SMJS system

For the typical case, FIRE's trajectory integration ran 70 times faster. The final state vector is off by 21 m (approximately) which again is well within acceptable limits given the long flight time (approximately 22 revolutions) and highly perturbed and eccentric nature of the trajectory.

Figure 9 shows the evolution of orbital elements for the perturbations plus J2 acceleration (2 by 0) case. Amongst Saturn's moons, Titan is the largest and its gravity signature appears as jumps in the orbital elements.

The orbital elements are shown for the first 100 days only. Figure 10a shows the resulting highly eccentric trajectory propagation. Figure 10b gives the instantaneous position differences between the FIRE and SPICE propagations.

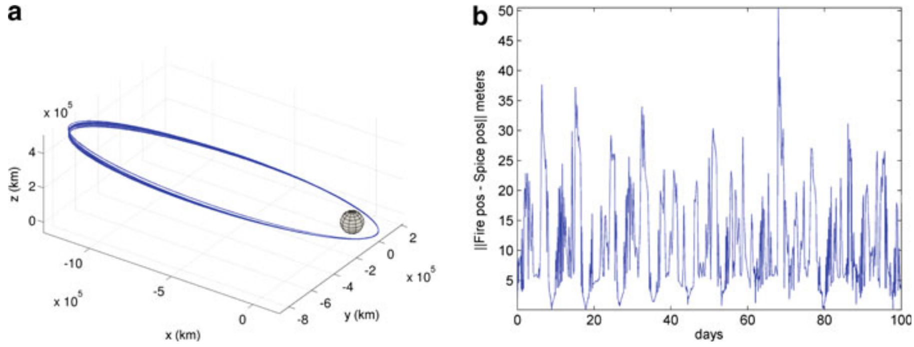

Fig. 10 **a** Propagated trajectory in SMJS system. **b** SPICE and FIRE position difference (SMJS)

4 Conclusion and future work

A new, fast, efficient, smooth, and accurate ephemeris interpolation system called FIRE is proposed for general use in precision trajectory and mission design. The FIRE system is custom built for applications traditionally bogged down by the heavy computational burden associated with typical ephemeris calls. FIRE relies on established ephemeris systems (like JPL's SPICE) to build a custom archived ephemeris, hence FIRE is intended as a supplemental capability for users that benefit from fast calls and smooth derivatives. The main disadvantage of FIRE system is the added layer of complexity and the additional memory requirements during runtime.

We demonstrate speed improvement of approximately two orders of magnitude for typical applications when compared to similar functionality in SPICE, one of the most widely used ephemeris system. The main performance gain is achieved through the modest use of random access memory to store a custom, portable runtime ephemeris. A major benefit of FIRE is that smooth, accurate, and self-consistent derivatives are natural artifacts of the interpolation method. These derivatives are often required for trajectory optimization and other classes of similar problems.

FIRE has been designed specifically to facilitate problems involving long or frequent ephemeris propagations for various classes of orbital mechanics problems. The new tool has potential value to any high precision application or software requiring fast, accurate, and smooth ephemeris data. Contact the authors for access to the FIRE system.

References

Acton, C.: Ancillary data services of NASA's navigation and ancillary information facility. Planet. Space Sci. **44**, 65–70 (1996)

Aiello, J.: Numerical investigation of mapping orbits about Jupiter's icy moons. In: AAS/AIAA Astrodynamics Specialist Conference. Lake Tahoe, California (2005)

Burrus, C., Eschenbacher, P.: An in-place, in-order prime factor FFT algorithm. IEEE Trans. Comput. **29**, 806–817 (1981)

Campbell, E.T., Speckman, L.E.: Preliminary design of feasible Athos intercept trajectories. In: Planetary Defense Conference: Protecting Earth from Asteroid. Orange County, California (2004)

Conte, S., deBoor, C.: In: Elementary Numerical Analysis (1972)

Cooley, J.W.A., Peter, W.D., Welch, P.: The fast Fourier transform and its applications. IEEE. Trans. Educ. **12**, 27–34 (1969)

deBoor, C.D.: In: A Practical Guide to Splines, vol. 27. Springer (2001)

Desai, P.N., Cheatwood, F.M.: Entry dispersion analysis for the genesis sample return capsule. In: AAS/AIAA Astrodynamics Specialist Conference. Girdwood, Alaska (1999)

Giampieri, G., Dougherty, M.K.: Rotation rate of Saturn's interior from magnetic field observations. Geophys. Res. Lett. **31**, L16701 (2004)

Hou, H.: The fast Hartley transform algorithm. IEEE Trans. Comput. **36**, 147–156 (1987)

Huang, Q., Yokoi, K., Kajita, S., Kaneko, K., Arai, H., Koyachi, N., Tanie, K.: Planning walking patterns for a biped robot. IEEE Trans. Rob. Autom. **17**, 280–289 (2001)

Hull, D.G.: Numerical derivatives for parameter optimization. J. Guid. Control Dyn. **2**, 158–160 (1979)

Jacobson, R.A., French, R.G.: Orbits and masses of Saturn's co-orbital and f-ring shepherding satellites. Icarus **172**, 382–387 (2004)

Koon, W.S., Lo, M.W., Marsden, J.E., Ross, S.D.: The genesis trajectory and heteroclinic connections. In: AAS/AIAA Astrodynamic Specialist Conference, p. 451 (1999)

Lam, T., Hirani, A.N.: Characteristics of transfers to and captures at Europa. In: AAS/AIAA Space Flight Mechanics Meeting. Tampa, Florida, Paper AAS 06-188 (2006)

Lin, C., Chang, P., Luh, J.: Formulation and optimization of cubic polynomial joint trajectories for industrial robots. IEEE Trans. Automat. Contr. **28**(12), 1066–1074 (1983)

Lo, M.W., Anderson, R., Whiffen, G., Romans, L.: The role of invariant manifolds in low thrust trajectory design. In: AAS/AIAA Astrodynamic Specialist Conference, vol. 288 (2004)

Prince, P.J., Dormand, J.R.: High order embedded Runge–Kutta formulae. J. Comput. Appl. Math. **7**(1), 67–75 (1981)

Russell, R.P.: Global search and optimization for free-return Earth-Mars cyclers. Ph.D. thesis, The University of Texas at Austin (2004)

Russell, R.P.: Primer vector theory applied to global low-thrust trade studies. J. Guid. Control Dyn. **30**, 460–472 (2007)

Russell, R.P., Lam, T.: Designing capture trajectories to unstable periodic orbits around Europa. J. Guid. Control Dyn. **30**, 482–491 (2006)

Russell, R.P., Lara, M.: Long-life lunar repeat ground track orbits. J. Guid. Control Dyn. **30**, 982–993 (2007)

Sauer, C.: Midas: mission design and analysis software for the optimization of ballistic interplanetary trajectories. J. Astronaut. Sci. **37**, 251–259 (1989)

Seidelmann, P.K., Archinal, B.A., A'hearn, M.F., Conard, A., Consolmagno, G.J., Hestroffer, D., Hilton, J.L., Krasinsky, G.A., Neumann, G., Oberst, J., Stooke, P., Tedesco, E.F., Tholen, D.J., Thomas, P.C., Williams, I.: Report of the IAU/IAG working group on cartographic coordinates and rotational elements: 2006. Celest. Mech. Dyn. Astron. **98**, 155–180 (2007)

Shoemake, K.: Animation rotation with quaternion curves. Comput. Graph. **19**, 245–254 (1985)

Shuster, M.: A survey of attitude representations. J. Astronaut. Sci. **41**, 439–517 (1993)

Spedding, G.R., Rignot, E.J.M.: Performance analysis and application of grid interpolation techniques for fluid flows. Exp. Fluids **15**, 417–430 (1993)

Standish, E.M.: JPL planetary and lunar ephemerides. In: CD-ROM. Willman-Bell Inc, Richmond, VA (1997)

Tsiotras, P., Schaub, H., Junkins, J.L.: Higher-order Cayley transforms with applications to attitude representations. J. Guid. Control Dyn. **20**(3), 528–536 (1997)

Unser, M.: Splines: a perfect fit for signal and image processing. IEEE Signal Process. Mag. **15**, 22–38 (1993)

Whiffen, G.J.: Jupiter icy moons orbiter reference trajectory. In: AAS/AIAA Astrodynamic Specialist Conference, vol. 186 (2006)