AAS 13-725

F AND G TAYLOR SERIES SOLUTIONS TO THE STARK PROBLEM WITH SUNDMAN TRANSFORMATIONS

Etienne Pellegrini[*], Ryan P. Russell[†], and Vivek Vittaldev[‡]

The classic F and G Taylor series of Keplerian motion are extended to solve the Stark problem and to use the generalized Sundman transformation. Exact recursion formulas for the series coefficients are derived, and the method is implemented to high order via a symbolic manipulator. The results lead to fast and accurate propagation models with efficient discretizations. The new F and G Stark series solutions are compared to the Modern Taylor Series (MTS) and 8th order Runge-Kutta-Fehlberg (RKF8) solutions. In terms of runtime, the F and G approach is shown to compare favorably to the MTS method up to order 18, and both Taylor series methods enjoy approximate order of magnitude speedups compared to RKF8 implementations. Actual runtime is shown to vary with eccentricity, perturbation size, prescribed accuracy, and the Sundman power law. The effects of the generalized Sundman transformation on the accuracy of the propagation are analyzed, and the results are valid for both the Stark and Kepler problems. The Taylor series solutions are shown to be exceptionally efficient when the unity power law from the classic Sundman transformation is applied. An example low-thrust trajectory propagation demonstrates the utility of the F and G Stark series solutions.

1 INTRODUCTION

The development of numerical methods for the integration of Ordinary Differential Equations (ODEs) is a fundamental problem in celestial mechanics, and has been studied extensively over the past decades. The typical goal when developing new integration schemes is to improve on the Pareto front of speed and accuracy. A number of different methods have been developed, including Runge-Kutta, multistep, symplectic, collocation, and Taylor series methods.[1,2,3] Montenbruck[4] demonstrated that the Taylor series methods have several advantages over other numerical schemes, which include high and variable order, and simple step control. Taylor series methods also allow for the interpolation of states between time steps with no loss of accuracy, and trivial computational efforts.

Different formulations of the ODEs have also been exploited to achieve better integration performance, in some cases avoiding integration altogether. In the well-known case of the 2-body problem, an efficient scheme consists in using the Lagrange f and g functions, coupled with a solution to Kepler's equation using Universal Functions.[5] The so-called F and G series are a Taylor series representation of these Lagrange f and g functions. The advantage of such a representation lies in the fact that simple recursive formulations exist for obtaining the coefficients of the F and G series.[6,7] This method has been used to solve a variety of problems. Steffensen[8] first developed a method using recursive power series for the restricted 3-body problem. Rabe[9] adapted it for the case of periodic Trojan orbits, and Broucke[10] extended the method to solve the N-body problem. Sconzo et. al[6] showed that the F and G Taylor series can be used efficiently as integrators, since their coefficients can be obtained to high order using a symbolic manipulator. Bond[7] developed a recursive formulation of the Lagrange coefficients for Keplerian motion. The generalized F and G series, extension of the classic F and G series to the N-body problem, were introduced by Papadakos,[11] who also demonstrated their convergence. The F and G series integration method has also been used for orbit determination,[12] for

[*]Graduate Student, Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX
[†]Assistant Professor, Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX
[‡]Graduate Student, Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX

the solution to Lambert's two point boundary value problem,[13] and for the propagation of certain cases of perturbed 2-body motion.[14,15]

In this study, the F and G series method is extended to the so-called Stark problem, which corresponds to Keplerian motion perturbed by an inertially constant acceleration. The German physicist Johannes Stark, the namesake of the problem, discovered the shifting and splitting of spectral lines of atoms and molecules in the presence of an external static electric field. This phenomenon is now commonly known as Stark effect.[16] Apart from its importance in nuclear physics, the Stark effect can also approximate the behavior of many astrodynamical systems. A historical survey of the extensive research conducted on solving the Stark problem's equations of motion can be found in Reference 17. Examples of the Stark effect in celestial mechanics include the propagation of any system with an inertially constant or approximately constant perturbation, such as a finite thrust arc, solar radiation pressure, or the direct influence of a distant third body. Time varying perturbations can be approximated by the Stark model with the use of sufficiently small time steps.[18,19]

In particular, low-thrust trajectories often are closely modeled by the Stark dynamics; however, most implementations of low-thrust optimization algorithms use general ODE solvers instead of exploiting the custom Stark solution. The computationally expensive nature of general ODE solutions has led some researchers to use reduced models, such as the Sims and Flanagan[20] model that employs a Kepler arc plus an impulsive ΔV to approximate a low-thrust arc. Such an approach avoids classic numerical integration and leads to significant runtime savings, but comes at the expense of accuracy. Alternatively, References 21, 22, 23 and 17 give analytic solutions to the Stark problem, leading to highly accurate solutions. However, analytic techniques suffer from complex formulations and implementations, may include various levels of approximation, and require transcendental functions such as elliptic integrals to solve. In the current work, a new technique to solve the Stark problem is sought that 1) preserves the accuracy of the analytic solutions, 2) is simple to implement, 3) avoids classic numeric integration, and 4) is computationally efficient. It is noted that Reference 18 makes a similar argument, and suggests the use of the Modern Taylor Series method to accomplish the aforementioned goals.

Additionally, both the present work and Reference 18 make use of the Sundman transformation in order to generate an efficient geometric discretization scheme. In this study, the classic transformation, employed in Reference 18, is extended to the generalized Sundman transformations. The original transformation was introduced by Karl Sundman as a way to regularize the equations of motion of the 3-body problem and avoid collision singularities.[24] The use of the transformation has proven to be a remarkably simple way to obtain efficient discretization schemes, especially for highly eccentric orbits.[25,18] The Sundman transformation provides a uniformization of the truncation error at each integration step[26] and often reduces the global truncation error by reducing dynamical instability.[26,27] Both of these effects result in an improved integration performance, because the lower global truncation error allows the integrator to take larger steps. Furthermore, the uniformization of the local truncation error makes the use of fixed-step integrators possible, noting that these are often more efficient than their variable-step counterpart.[28]

Unlike the F and G approach, which is the main focus of the current study, Taylor series coefficients for the state variables can also be computed recursively using Leibniz's product rule. The Taylor series formed in this manner is deemed the Modern Taylor Series (MTS) method and has identical coefficients to those found using the F and G approach. The MTS method was originally implemented for the 2-body problem in the astrodynamics field as the power series method.[29] The method has been shown to be valid for any ODE problem that is piecewise smooth,[30] and, as previously mentioned, has been used recently for the integration of a low-thrust trajectory using the classic Sundman transformation.[18] Other recent astrodynamics applications of the MTS method include Reference 31, where the efficiency of the MTS method is highlighted for near Keplerian orbits. Furthermore, in Reference 30, the method is shown to perform similarly to conventional integration methods on a restricted three body problem simulation. The MTS method is used as one of two benchmarks in the current study to evaluate the performance of the proposed F and G Stark series solutions.

The main contributions of this work are: 1) Recursion equations for the F and G Stark series are derived in the case of the 3D Stark problem with generalized Sundman transformations. The recursions are implemented to high order using a symbolic manipulator. 2) A historical survey is presented for the generalized

Sundman transformation, and its application is studied in detail for the F and G Stark series, the MTS, and two implementations of a Runge-Kutta-Fehlberg order 8 (RKF8) integrator. A variety of different power laws defining the generalized Sundman transformation are considered, and their effects on the integration accuracy are presented. The tests and their results apply to both the Stark problem and the unperturbed Kepler problem. 3) A comparative study of the computational times required by each of the integration schemes is performed, paying close attention to compare only solutions of similar accuracy.

The numerical results presented in this paper* show that: 1) The Taylor series solutions to the Stark problem have significantly better performance than a classic RKF8 integrator; 2) The Taylor series methods are exceptionally efficient for the Sundman transformation corresponding to a unity power law, a result which departs from the conventional wisdom that the power law of $3/2$ (corresponding to the so-called intermediate anomaly) is most efficient for astrodynamics; and 3) The impressive speedups of both Taylor series methods compared to the RKF8 are specific to the simple form of the Stark and 2-body equations of motion. These speedups can not be generally expected when applying the Taylor series methods to other more complicated dynamical systems.

The remaining agenda for the paper is organized as follows: Section 2 presents the details of the generalized Sundman transformation and its effects on the discretization of the problem. Section 3 shows the derivation of the recursion formulas needed to compute the coefficients of the F and G series for the Stark problem. Section 4 presents the comparison in performance between the F and G series, the MTS method, and the two implementations of the RKF8 method. Finally, the application of all methods to a low-thrust orbital transfer is presented.

2 THE GENERALIZED SUNDMAN TRANSFORMATION

In this section, the generalized Sundman transformation and its application to astrodynamics are briefly surveyed, and the effects of the transformation on the discretization of the orbit are demonstrated.

2.1 Presentation and historical survey

The classic Sundman transformation was introduced in 1912 by Karl F. Sundman, as a way to regularize the equations of motion and to avoid collision singularities in the 3-body problem.[24] It consists of a transformation of the independent variable from time t to τ, given by:

$$dt = r\, d\tau \tag{1}$$

where r is the magnitude of the radius vector and τ is the new independent variable.

It is well known that the use of the Sundman transformation significantly improves integration performance by reducing and uniformizing the local truncation error.[32] The numerical behavior of the regularized expressions becomes less dependent on the eccentricity when compared to the use of time as the independent variable.[33] The transformation was later extended to the so-called generalized Sundman transformation:

$$dt = cr^{\alpha}\, d\tau \tag{2}$$

where c is a constant and α is a positive constant.

Many authors have studied the generalized Sundman transformation in order to find the optimal value for α and to understand why it presents such numerical benefits. Baumgarte[27] showed that the transformation with $\alpha = 1$ reduces dynamical instability in the Lyapunov sense. Velez[26] tied this dynamical instability to error propagation, and further stated that the instability can be removed entirely by using $\alpha = 2$. Bond[34] showed that this $\alpha = 2$ case corresponds to transforming the equations of motion to oscillator form. However, Velez also explained that increasing α leads to higher numerical instability, and could therefore be detrimental to the overall accuracy of the propagation. Velez,[26] Feagin and Mikkileni,[35] and Nacozy[28] all show that in order to reduce the local truncation error, there is not a single optimal transformation for all problems, or even for

*Subject to the hardware and software specifications and the implementation described in Section 4

all points on one orbit. The ideal value for α depends on the problem, on the location on the orbit and on the order of the integrator. However, Merson,[36] Velez[26] and Nacozy[28] all agree that the case $\alpha = 3/2$ is the best choice for most satellite applications, because they found it most efficient in the case of a J_2 perturbation. It is well known that $\alpha = 1$ and $\alpha = 2$ make the independent variable τ proportional to eccentric and true anomalies, respectively. Because the $\alpha = 3/2$ case proved ideal for heavily used scenarios, Nacozy[37] introduced the *intermediate anomaly*, corresponding to $\alpha = 3/2$.

Authors have also proposed the idea of a varying α,[28] or more generally of a transformation of type:

$$dt = g(r)\, d\tau \tag{3}$$

where g is an arbitrary function of r.[38,39] However, this more complicated type of transformation will not be considered in this paper.

An important point, made by most of the cited authors, is that the Sundman transformation and its extensions introduce an extra differential equation in the problem: it is now necessary to integrate Eq. (2) in order to keep track of time. In fact, this new differential equation exhibits part or all of the dynamic instability that was removed from the equations of motion. However, if the equations of motion are autonomous, then there is no feedback of the instability into the coordinates, and the benefits of the transformation usually overcome this disadvantage.

2.2 Application of the Sundman transformation

In the present work, a time transformation of type $dt = cr^\alpha d\tau$, with $c = 1$ and α constant is studied. Different values for α are tested with different integrators applied specifically to the dynamics of the Stark problem. Note that the results of this section apply generally to both the Stark and the Kepler problems.

When using the transformation, the independent variable is changed from time to τ, so that the traditional ODE:

$$\dot{\mathbf{X}} = \frac{d\mathbf{X}}{dt} = \mathbf{f}(t, \mathbf{X}) \tag{4}$$

becomes:

$$\begin{cases} \mathbf{X}' = \dfrac{d\mathbf{X}}{d\tau} = \dfrac{d\mathbf{X}}{dt}\dfrac{dt}{d\tau} = cr^\alpha \mathbf{f}(t, \mathbf{X}) \\ \dfrac{dt}{d\tau} = cr^\alpha \end{cases} \tag{5}$$

where $(x)'$ and $(\dot{x})$ designate differentiations with respect to τ and time, respectively.

It is important to discuss the case of second-order ODEs, as they are heavily used in astrodynamics. The conventional way of handling a second-order ODE is to create a state vector containing the state and its first derivative, and effectively converting the second-order equation to a system of first-order equations. For astrodynamics applications, the typical state vector contains position and velocity:

$$\mathbf{X} = \begin{bmatrix} \mathbf{r} \\ \dot{\mathbf{r}} \end{bmatrix} \tag{6}$$

When using the Sundman transformation with a first order integrator, two formulations are possible. In the first formulation, the state vector is the one presented in Eq. (6), and the propagation essentially corresponds to a first-order integration of the position and velocity ($\mathbf{r}$ and $\dot{\mathbf{r}}$). For an arbitrary time-acceleration $\ddot{\mathbf{r}}$:

$$\mathbf{X}' = \begin{bmatrix} \mathbf{r}' \\ \dot{\mathbf{r}}' \end{bmatrix} = \begin{bmatrix} cr^\alpha \dot{\mathbf{r}} \\ cr^\alpha \ddot{\mathbf{r}} \end{bmatrix} \tag{7}$$

Equation (7) will be called "time-velocity" equations of motion in this paper.

In the second formulation, the following state vector is used:

$$\mathbf{X} = \begin{bmatrix} \mathbf{r} \\ \mathbf{r}' \end{bmatrix} \tag{8}$$

where the integration now corresponds to solving a second-order ODE in τ, yielding:

$$\begin{aligned}\mathbf{X}' = \begin{bmatrix}\mathbf{r}' \\ \mathbf{r}''\end{bmatrix} &= \begin{bmatrix}\mathbf{r}' \\ (cr^\alpha)^2\ddot{\mathbf{r}} + (cr^\alpha)'\dot{\mathbf{r}}\end{bmatrix} \\ &= \begin{bmatrix}\mathbf{r}' \\ (cr^\alpha)^2\ddot{\mathbf{r}} + \frac{\alpha}{r^2}(\mathbf{r}\cdot\mathbf{r}')\mathbf{r}'\end{bmatrix}\end{aligned} \tag{9}$$

since

$$\mathbf{r}'' = \frac{d}{d\tau}\left(\frac{d\mathbf{r}}{d\tau}\right) = \frac{d}{d\tau}(cr^\alpha\dot{\mathbf{r}}) = (cr^\alpha)^2\ddot{\mathbf{r}} + (cr^\alpha)'\dot{\mathbf{r}} \tag{10}$$

and

$$(cr^\alpha)'\dot{\mathbf{r}} = \alpha c r^{\alpha-1} r'\dot{\mathbf{r}} = \alpha r'\frac{r}{r}\frac{cr^\alpha}{r}\dot{\mathbf{r}} = \frac{\alpha}{r^2}rr'\mathbf{r}' \tag{11}$$

The true velocity of the particle is obtained from the τ-velocity:

$$\dot{\mathbf{r}} = \frac{1}{cr^\alpha}\mathbf{r}' \tag{12}$$

Equation (9) will be called "τ-velocity" equations of motion. It is shown in Section 3 that the Taylor series methods correspond to a τ-velocity method and associated state from Eq. (8). For the benchmark Runge-Kutta integrator, results from both formulations are presented.

Figure 1 presents the result of high fidelity integrations of Eq. (5) using 2-body dynamics only, for different values of α. The output nodes are equally spaced in τ, although $\Delta\tau$ is not the same from one trajectory to another: the τ-period (τ_p) depends on the value of α, and $\Delta\tau = \tau_p/50$. Table 1 gives the formulas for the computation of τ_p.

Table 1. **τ-period for different values of α**

α	τ proportional to	τ_p
0	Time	$\frac{2\pi}{nc}$
1	Eccentric Anomaly	$\frac{2\pi}{nca}$
3/2	Inter. Anomaly	$\frac{4\ K\left(\sqrt{\frac{2e}{1+e}}\right)}{c\sqrt{\mu(1+e)}}$
2	True Anomaly	$\frac{2\pi}{nc\sqrt{a(1-e^2)}}$

$n = \sqrt{\frac{\mu}{a^3}}$ is the mean orbital motion
$K(\,)$ is the complete elliptic integral of the first kind
μ is the standard gravitational parameter

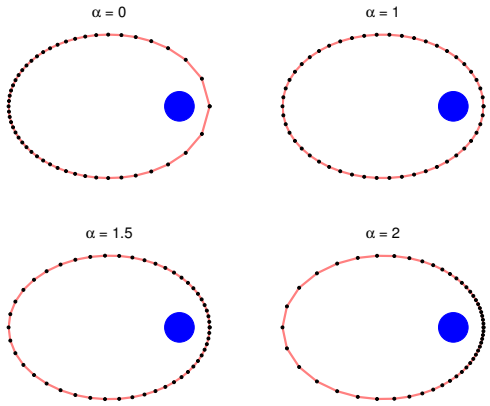
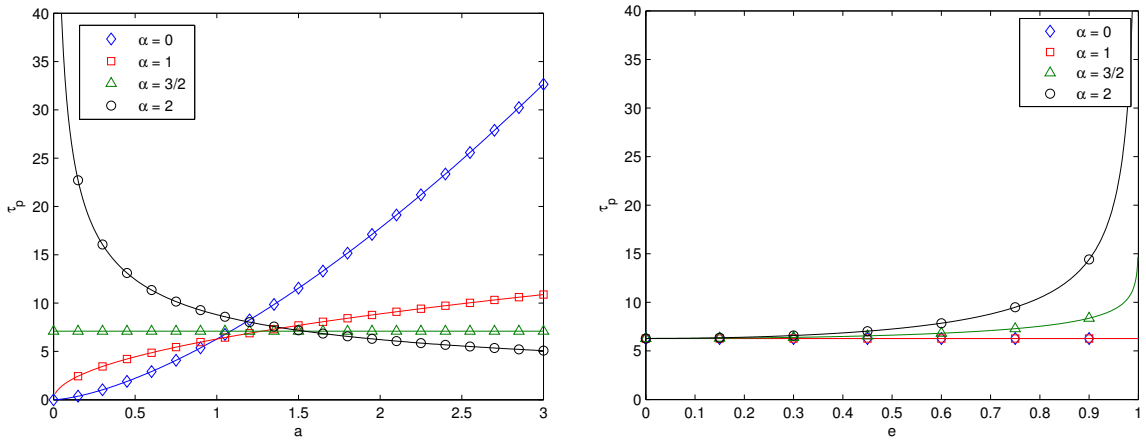


Figure 1. Effect of the Sundman power law. $a = 1$, $e = 0.7$

Figure 2 presents the τ-period as a function of a or e for those four cases. For $\alpha = 0$ or 1, the period depends solely on a, for $\alpha = 3/2$, it depends solely on e, and finally for $\alpha = 2$, it depends on both a and e. For visualization purposes, a is fixed at 1 when varying e, and e is fixed at 0.7 when varying a.

The typical clustering of points around apoapse is easily seen in Fig. 1 for the time case ($\alpha = 0$), and the gradual shifting of that clustering changes when increasing α. A value of $\alpha = 1$ corresponds to a regular geometric spacing, proportional to the eccentric anomaly, and independent of whether the particle is close to apoapse or periapse. The points start to cluster around periapse for $\alpha = 3/2$, and the effect is exaggerated further for $\alpha = 2$. Note that the discretization in the case of $\alpha = 3/2$ most closely mimics the behavior of variable-step conventional numerical integrators, when compared to the other discretizations of



(a) Varying a

(b) Varying e. The $\alpha = 0$ and $\alpha = 1$ overlap.

Figure 2. τ-period vs. orbital elements

Fig. 1. Therefore, the Sundman transformations, in particular the $\alpha = 3/2$ case, are often used in fixed step integrators. The results presented in Fig. 1 confirm the conclusions of many authors, in particular those of Berry and Healy.[25]

In the following sections, particular attention is given to the eccentric anomaly case ($\alpha = 1$) because the regular geometric discretization is ideally suited for modeling a perturbation or optimizing a control. In Subsection 4.3, the distribution of the local error along one revolution is presented for all the propagation methods and for different values of α. It is shown that the optimal α is strongly dependent on the integrator. In particular, the optimal α for Taylor Series methods is shown to be 1, both a peculiar and serendipitous result.

3 F AND G SERIES FORMULATION OF THE STARK PROBLEM

In this section, the principles of the classic F and G Taylor series method are reviewed, and the derivation is extended to include the generalized Sundman transformation and the 3D perturbation of the Stark problem.

3.1 Problem Statement

The Stark problem is a perturbed 2-body problem, where the perturbation is inertially constant in magnitude and direction. The equations of motion in time are:

$$\begin{aligned} \frac{d\mathbf{r}}{dt} &= \mathbf{v} \\ \frac{d\mathbf{v}}{dt} &= -\frac{\mu}{r^3}\mathbf{r} + \mathbf{p} \end{aligned} \tag{13}$$

where $\mathbf{p}$ is the perturbing acceleration.

The f and g functions, introduced by Lagrange, are a well-known formulation of the 2-body problem, which express position and velocity at a time t as a linear combination of position and velocity at time t_0:

$$\begin{aligned} \mathbf{r}(t) &= f\mathbf{r}_0 + g\mathbf{v}_0 \\ \mathbf{v}(t) &= \dot{f}\mathbf{r}_0 + \dot{g}\mathbf{v}_0 \end{aligned} \tag{14}$$

This formulation is valid for Keplerian orbits, as the motion is confined to a plane. However in the Stark problem, out-of-plane motion will occur when the perturbation $\mathbf{p}$ is out-of-plane. Therefore, a third basis

vector is needed for the general 3D problem. The choice of the third vector is arbitrary, and the perturbation vector itself will be used in this study. Note that the authors considered other options for this third basis vector, and found that the inertially fixed perturbation vector leads to the simplest and most elegant formulation. The three vectors $\mathbf{r}_0$, $\mathbf{v}_0$ and $\mathbf{p}$ form a basis to the 3D space. If the perturbation is in the plane of $\mathbf{r}_0$ and $\mathbf{v}_0$, $\mathbf{p}$ is a redundant basis vector, but the 2D space is still spanned. This new third basis vector necessitates the development of a third series to be computed alongside F and G. Moreover, the use of the Sundman transformation introduces an extra differential equation for time (see Eq. (5)), which requires the development of one more series. The method introduced in this paper will therefore use four series instead of the classic two F and G series. However, for legacy purposes the set of four series will be called "F and G Stark series".

3.2 Development of the series formulation

To solve the Stark integration problem using Taylor series, the development of the conventional[13] F and G series is extended. For order N, to obtain the position vector $\mathbf{r}$ at $\tau = \tau_0 + \Delta\tau$, the following truncated Taylor series is employed:

$$\mathbf{r} \simeq \sum_{n=0}^{N} \left.\frac{d^n\mathbf{r}}{d\tau^n}\right|_{\tau_0} \frac{\Delta\tau^n}{n!} \tag{15}$$

As mentioned previously, $\mathbf{r}_0$, $\mathbf{v}_0$ and $\mathbf{p}$ form a suitable basis for either a 2D or 3D space, and therefore the $\mathbf{r}$ derivative terms of Eq. (15) can be expressed as a linear combination of $\mathbf{r}_0$, $\mathbf{v}_0$ and $\mathbf{p}$:

$$\left.\frac{d^n\mathbf{r}}{d\tau^n}\right|_{\tau_0} = F_n|_{\tau_0}\mathbf{r}_0 + G_n|_{\tau_0}\mathbf{v}_0 + H_n|_{\tau_0}\mathbf{p} \tag{16}$$

which, when substituted into Eq. (15) yields:

$$\begin{aligned}\mathbf{r} &\simeq \sum_{n=0}^{N}\left[F_n|_{\tau_0}\frac{\Delta\tau^n}{n!}\right]\mathbf{r}_0 + \sum_{n=0}^{N}\left[G_n|_{\tau_0}\frac{\Delta\tau^n}{n!}\right]\mathbf{v}_0 + \sum_{n=0}^{N}\left[H_n|_{\tau_0}\frac{\Delta\tau^n}{n!}\right]\mathbf{p} \\ &\equiv F\mathbf{r}_0 + G\mathbf{v}_0 + H\mathbf{p}\end{aligned} \tag{17}$$

The "extended" Lagrange coefficients form of the solution is:

$$\mathbf{r}(\tau) = f\mathbf{r}_0 + g\mathbf{v}_0 + h\mathbf{p} \tag{18}$$

Comparing Eq. (17) taken to order infinity and Eq. (18) yields the relation between the f, g, and h Stark functions and the F, G, and H Stark series:

$$\begin{aligned} f &= \sum_{n=0}^{\infty}\left[F_n|_{\tau_0}\frac{\Delta\tau^n}{n!}\right] \\ g &= \sum_{n=0}^{\infty}\left[G_n|_{\tau_0}\frac{\Delta\tau^n}{n!}\right] \\ h &= \sum_{n=0}^{\infty}\left[H_n|_{\tau_0}\frac{\Delta\tau^n}{n!}\right] \end{aligned} \tag{19}$$

In order to find the coefficients F_n, G_n, and H_n for any n, Equation (16) is expressed at any τ:

$$\frac{d^n\mathbf{r}}{d\tau^n} = F_n\mathbf{r} + G_n\mathbf{v} + H_n\mathbf{p} \tag{20}$$

Equation (20) is differentiated with respect to τ:

$$\frac{d^{(n+1)}\mathbf{r}}{d\tau^{(n+1)}} = F_n'\mathbf{r} + F_n\frac{d\mathbf{r}}{d\tau} + G_n'\mathbf{v} + G_n\frac{d\mathbf{v}}{d\tau} + H_n'\mathbf{p} + H_n\frac{d\mathbf{p}}{d\tau} \tag{21}$$

$$= F_n'\mathbf{r} + cr^{\alpha}F_n\mathbf{v} + G_n'\mathbf{v} + cr^{\alpha}G_n(-\frac{\mu}{r^3}\mathbf{r} + \mathbf{p}) + H_n'\mathbf{p} \tag{22}$$

Alternatively, advancing the sequence in Eq. (20) from n to $n+1$ leads to:

$$\frac{d^{(n+1)}\mathbf{r}}{d\tau^{(n+1)}} = F_{n+1}\mathbf{r} + G_{n+1}\mathbf{v} + H_{n+1}\mathbf{p} \tag{23}$$

Equating terms of Eq. (22) and Eq. (23) gives the recursive coefficient definitions:

$$\begin{aligned} F_{n+1} &= F'_n + cr^\alpha G_n\Big(-\frac{\mu}{r^3}\Big) \\ G_{n+1} &= G'_n + cr^\alpha F_n \\ H_{n+1} &= H'_n + cr^\alpha G_n \end{aligned} \tag{24}$$

In order to integrate the time with respect to τ, a fourth series is added:

$$\Delta t \simeq \sum_{n=1}^{N} T_n \frac{\Delta\tau^n}{n!} \tag{25}$$

$$T_{n+1} = T'_n \tag{26}$$

The coefficients are initialized using $F_0 = 1$, $G_0 = 0$, $H_0 = 0$ and $T_1 = cr^\alpha$, since $dt = cr^\alpha d\tau$, and for $\Delta\tau = 0$, $\mathbf{r} = F_0\mathbf{r}_0 + G_0\mathbf{v}_0 + H_0\mathbf{p} = \mathbf{r}_0$.

To compute the primed quantities in Eq. (24) in terms of unprimed known values, the following intermediate variables and differentiation rules are adopted:

$$\frac{dr}{d\tau} = cr^\alpha \frac{dr}{dt} = cr^\alpha \frac{r\dot{r}}{r} = cr^\alpha \frac{\mathbf{r}\cdot\mathbf{v}}{r} = s\frac{k_1}{r} \tag{27}$$

$s \equiv cr^\alpha$

$$\frac{ds}{d\tau} = \alpha s^2 \frac{\mathbf{r}\cdot\mathbf{v}}{r^2} = \alpha s^2 \frac{k_1}{r^2} \tag{28}$$

$k_1 \equiv \mathbf{r}\cdot\mathbf{v}$

$$\frac{dk_1}{d\tau} = s\Big(v^2 - \frac{\mu}{r} + \mathbf{p}\cdot\mathbf{r}\Big) = s\Big(k_4 - \frac{\mu}{r} + k_2\Big) \tag{29}$$

$k_2 \equiv \mathbf{p}\cdot\mathbf{r}$

$$\frac{dk_2}{d\tau} = s\mathbf{p}\cdot\mathbf{v} = sk_3 \tag{30}$$

$k_3 \equiv \mathbf{p}\cdot\mathbf{v}$

$$\frac{dk_3}{d\tau} = s\Big(\frac{-\mu}{r^3}\mathbf{p}\cdot\mathbf{r} + p^2\Big) = s\Big(\frac{-\mu}{r^3}k_2 + p^2\Big) \tag{31}$$

$k_4 \equiv \mathbf{v}\cdot\mathbf{v}$

$$\frac{dk_4}{d\tau} = 2s\Big(\mathbf{p}\cdot\mathbf{v} - \frac{\mu}{r^3}\mathbf{r}\cdot\mathbf{v}\Big) = 2s\Big(k_3 - \frac{\mu}{r^3}k_1\Big) \tag{32}$$

$$\frac{d\mathbf{p}}{d\tau} = \mathbf{0},\ \frac{dc}{d\tau} = \frac{d\alpha}{d\tau} = \frac{d\mu}{d\tau} = 0 \tag{33}$$

Using Equations (24) to (33), the F, G, H, and T series, applied at τ_0, can be developed up to any order N. Equation (17) is used to obtain the position vector $\mathbf{r}$. To compute the velocity vector $\mathbf{v}$, Eq. (17) is differentiated with respect to time:

$$\begin{aligned} \mathbf{v} &= \frac{d\mathbf{r}}{dt} = \frac{d\mathbf{r}}{d\tau}\frac{d\tau}{dt} \\ &= \frac{1}{s}\left(\sum_{n=1}^{N}\left[F_n|_{\tau_0}\left(\frac{\Delta\tau^{(n-1)}}{(n-1)!}\right)\right]\mathbf{r}_0 + \sum_{n=1}^{N}\left[G_n|_{\tau_0}\left(\frac{\Delta\tau^{(n-1)}}{(n-1)!}\right)\right]\mathbf{v}_0 + \sum_{n=1}^{N}\left[H_n|_{\tau_0}\left(\frac{\Delta\tau^{(n-1)}}{(n-1)!}\right)\right]\mathbf{p}\right) \end{aligned} \tag{34}$$

Note that this form of the velocity computation corresponds to the τ-velocity implementation presented in Subsection 2.2.

The computation of the elapsed time is achieved using Eq. (25). Algorithm 1 gives the detailed steps for the computation of the F and G Stark series coefficients. In this study, the implementation of Algorithm 1 is performed with the symbolic manipulator Maple, and the software's code generation capabilities are employed to obtain optimized expressions for the coefficients. Details on the implementation are provided in

Appendix B, as well as the coefficients of all four series up to order six, which is the first order for which all the variables introduced in Equations (27) to (33) are used. Appendix B also provides a link to download the Fortran coefficient files.

An important drawback of the F and G Stark series is the complexity of the high order coefficients. This complexity, associated with taking repeated analytic derivatives of multivariate functions, leads to large coefficients files when high orders are required. The symbolic manipulator efficiently combats the problem at low orders using intermediate variables to minimize repeated computations (see Appendix B). However, the complexity burden begins to overwhelm the symbolic manipulator near order 25, and the files become too large to be efficiently compiled. However, as is shown in Section 4, this limitation only presents a small inconvenience in practice, as discretizations of 15 or more points per revolution typically require orders below 20.

Algorithm 1 Computation of the coefficients F and G Stark series

Input: $\mathbf{r}_0$, $\mathbf{v}_0$, $\mathbf{p}$, μ, α

Initialize $k_1 \leftarrow \mathbf{r}\cdot\mathbf{v}$, $k_2 \leftarrow \mathbf{p}\cdot\mathbf{r}$, $k_3 \leftarrow \mathbf{p}\cdot\mathbf{v}$, $k_4 \leftarrow \mathbf{v}\cdot\mathbf{v}$, $s \leftarrow cr^{\alpha}$
Initialize $F_0 \leftarrow 1$, $G_0 \leftarrow 0$, $H_0 \leftarrow 0$ and $T_1 \leftarrow s$
for $n \leftarrow 0, N-1$ **do**
    Compute F_n', G_n', H_n', T_n' (i.e. differentiate each coefficient with respect to τ)
    Substitute $\{r' = s\frac{k_2}{r}$, $s' = \alpha s^2\frac{k_1}{r^2}$, $k_1' = s(k_4 - \frac{\mu}{r} + k_2)$, $b' = sk_3$, $k_3' = s(-\frac{\mu}{r^3}k_2 + p^2)$, $k_4' = 2s(k_3 - \frac{\mu}{r^3}k_1)\}$
    $F_{n+1} \leftarrow F_n' + sG_n(-\frac{\mu}{r^3})$
    $G_{n+1} \leftarrow G_n' + sF_n$
    $H_{n+1} \leftarrow H_n' + sG_n$
    $T_{n+1} \leftarrow T_n'$
end for

Output: $\{F_n, G_n, H_n, T_n\}$ for $n = 0 \rightarrow N$

4 NUMERICAL VALIDATION AND COMPARISON TO OTHER INTEGRATORS

This section presents the results obtained when using the F and G Stark series propagator on a set of different test cases. The method is validated and its computational efficiency is compared to both the MTS method (see Subsection 4.1) and the RKF8 numerical integrator. The Sundman transformation uniformizes the local error and justifies the use of fixed step integrators. The RKF8 integrator is implemented using both the time-velocity and the τ-velocity equations of motion using Eq. (7) and Eq. (9), respectively. The Stark time-acceleration is used: $\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{p}$ and the gravitational parameter is fixed at $\mu = 1$.

All simulations in this study use the GNU Fortran compiler (gfortran) v4.7.0 on a Windows 7 workstation with a quad-core Intel Xeon W3550 CPU with 3.07GHz clock speed, and 6GB of RAM.

4.1 The Modern Taylor Series method

For the reasons stated in the introduction, the MTS method is used in this study as a benchmark for the new F and G Stark series method. The coefficients are computed for elementary operations using Leibniz's product rule. The main challenge in implementing the MTS is splitting the dynamics into the elementary operations such as additions, subtractions, multiplications, divisions, exponents, etc. This decomposition can be done manually[31] (not advisable as the dynamics become more complicated) or using Automatic Differentiation (AD) packages.[30,40,41,42,43]

For the dynamics of the problem addressed in this project, only multiplication, division, and raising to a power are needed. These operations have to be carried out on Taylor series and therefore, the general Leibniz

product rule can be used:

$$[f(t)g(t)]^{(n)} = \sum_{k=0}^{n} \left(\begin{array}{c} n \\ k \end{array} \right) f^{(k)}(t) g^{(n-k)}(t) \tag{35}$$

where $x^{(n)}$ represents the n-th derivative of x. Using the general Leibniz rule, relations for other common elementary operations can be derived.[30] All the recursion relations for the current study are implemented in Fortran. Splitting the dynamics into elementary operations is carried out manually for the relatively simple Stark dynamics, and checked with the output produced by *Taylor*.[30]

4.2 Integrals of motion

Integrals of motion play an important role, when available, in numerical simulations. The integration accuracy can be represented by the variation in any integral of motion. Since these integrals are theoretically constant, any change in an integral of motion is due to numerical error. In this study, the Hamiltonian, one of the known integrals of motion of the Stark problem, is used to assess the accuracy of the numerical methods. It is noted that any of the other integrals of motion could have been used for the same purpose. The detailed expressions of the Stark problem's integrals of motion can be found in Reference 17. The Hamiltonian is simply:

$$H = \frac{1}{2} v^2 - \frac{\mu}{r} - \mathbf{r} \cdot \mathbf{p} \tag{36}$$

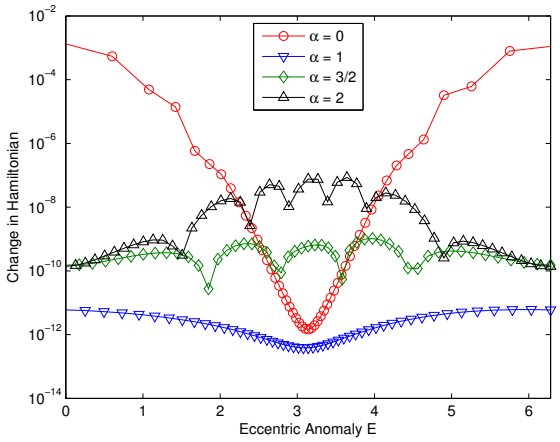
4.3 Study of the generalized Sundman transformation for the unperturbed 2-body problem

Many authors have studied the effects of the Sundman transformation on the performance of integrators. As mentioned previously, the benefits of the Sundman transformation include a reduction and a uniformization of the local error, allowing fixed-step integrators to perform similarly to their variable-step counterparts. In this section, the local truncation error is analyzed for the various power laws associated with the Sundman transformation. Unlike conventional integration methods, the Taylor series techniques are shown to be most efficient for $\alpha = 1$.

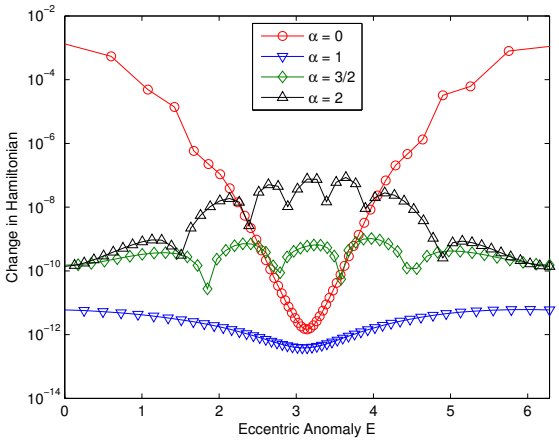
In this study, the local error is represented by the change in Hamiltonian over one integration step. The orbit used for the testing is an unperturbed Keplerian orbit with semi-major axis $a = 1$ LU and eccentricity $e = 0.6$, propagated for one full revolution with 50 equally spaced points in τ. The propagation starts and ends at periapse. The Taylor series methods are used with order 8 and 12, which result in fairly large changes in the Hamiltonians; indeed, it is important for this test that the round-off error can be neglected with respect to the local truncation error. Figure 3 shows the change in Hamiltonian across a single orbit for different α values. Figure 4 shows the maximum change in the Hamiltonian and the Root Mean Square (RMS) of this change over one period versus α for all integrators. The Taylor series solutions are presented with two different orders to show how the error is translated when the order changes. The plots of the total error and of the local error standard deviation are not shown, because they exhibit similar behavior to that of Fig. 4. The tests have also been realized using a Stark acceleration, but the behavior of the local error is similar to that of the unperturbed problem, so the non-zero perturbation plots are not included here.

For the case of the Runge-Kutta integrator, the results presented in Fig. 3(c) closely agree with past findings (see Subsection 2.1). The local error depends on the position of the particle along the orbit, and this behavior itself depends on α in the Sundman transformation. When $\alpha = 0$ (the independent variable is equivalent to time), the traditional rise of the error around periapse is observed, as well as a low error around apoapse. As α grows, the local error becomes more uniform, until the trend reverses near $\alpha = 2$. Figure 4 confirms this behavior, and shows that the ideal α for the presented case is between $3/2$ and 2, leading to minimized maximum numerical deviation of the Hamiltonian and minimized RMS of that deviation. When using the τ-velocity RKF8 integrator, the ideal α is considerably lower (see Fig. 4). Figure 3(d) shows that the local error is more uniform for $\alpha = 1$ when using the τ-velocity formulation than when using time-velocity.

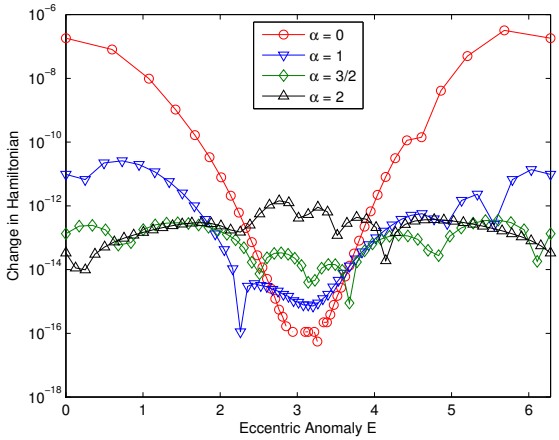
The results obtained when using either of the two Taylor series integrators demonstrate a peculiar behavior. As illustrated in Fig. 4, there is a narrow, but real, dip of the maximum and RMS errors at $\alpha = 1$. This



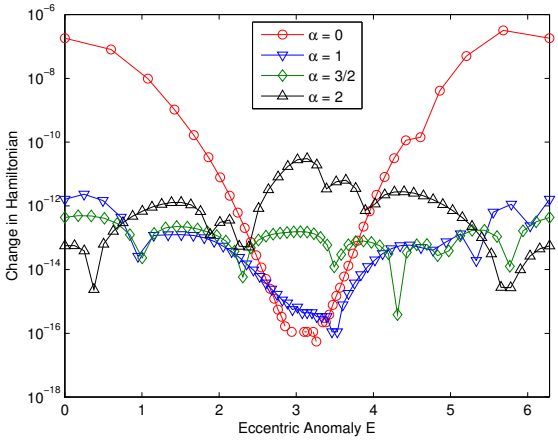
(a) F and G order 8



(b) MTS order 8

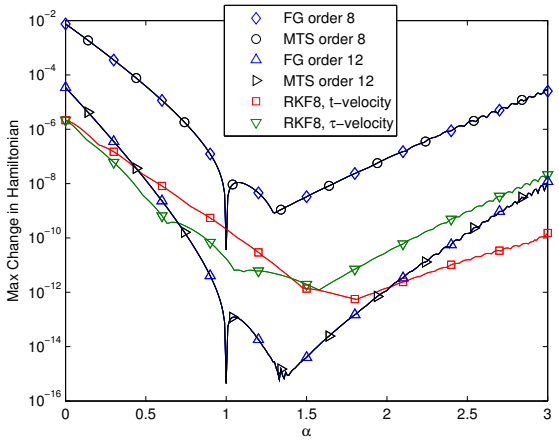


(c) RKF8, time-velocity

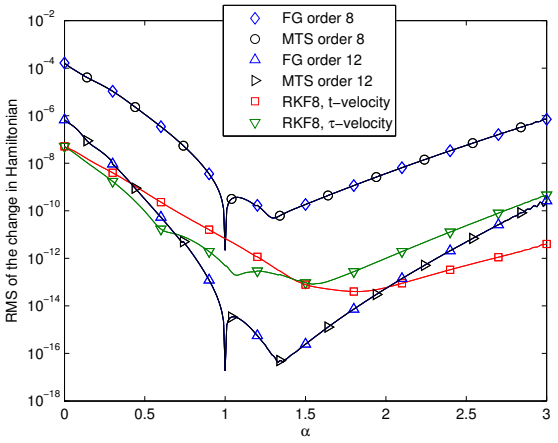


(d) RKF8, τ-velocity

Figure 3. Effect of orbit location on the local truncation error. $a = 1$**,** $e = 0.6$**,** $\mathbf{p} = \mathbf{0}$**,** $\mu = 1$



(a) Maximum change in Hamiltonian



(b) RMS of the change in Hamiltonian

Figure 4. Effect of α on the local truncation error. α equally spaced using $\Delta\alpha = 0.01$, $a = 1$**,** $e = 0.6$**,** $\mathbf{p} = \mathbf{0}$**,** $\mu = 1$

behavior is observed for both the F and G method and the MTS method, and is confirmed by Fig. 3(a) and Fig. 3(b). The local error when $\alpha = 1$ is lower and more uniform than for any other value of α. Aside from the sharp dip at $\alpha = 1$, the error statistics for other values of α behave similarly to their RKF8 counterparts. The optimal α for α not close to 1 is $\alpha \approx 1.25$, a further shift to the left of the RKF8 integrations. It is speculated that the narrow dip at $\alpha = 1$ is in part due to the recursion Equations (27) to (33) that are highly simplified for the exact case of $\alpha = 1$. This peculiar behavior presents a serendipitous advantage: the $\alpha = 1$ case corresponds to the regular geometric distribution of nodes along an ellipse, and is ideal for the low-thrust optimization problem and other problems benefiting from a regular discretization. Therefore, in the case of the Taylor series methods, the ideal α for discretization coincides with the ideal α for accuracy and efficiency.

4.4 Runtime comparisons between the Taylor series and the RKF8 integrator

Here, the F and G integrator is compared to the MTS integrator and to both the time- and τ-velocity implementations of the RKF8 integrator. The methods are compared on the basis of CPU time necessary for the propagation of approximately one revolution of a perturbed orbit.

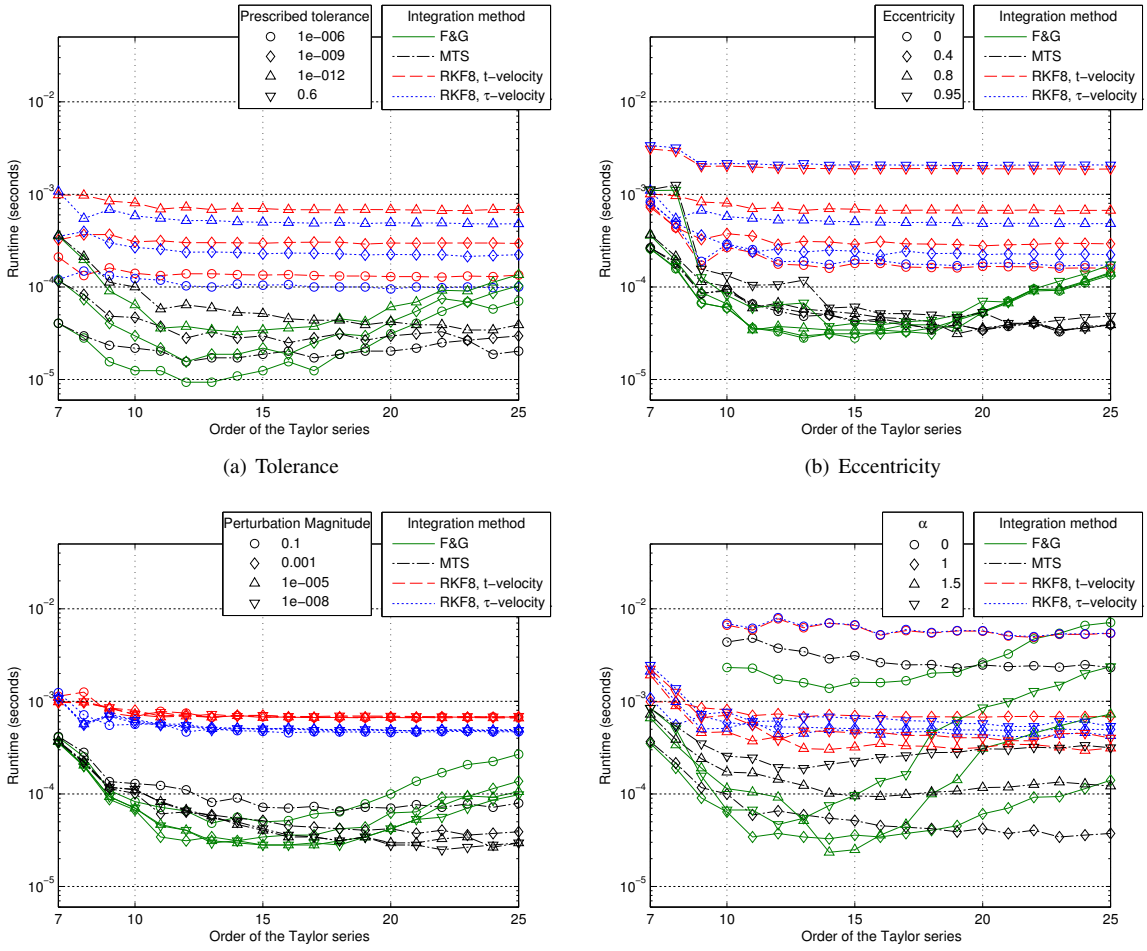
In order to compare the speed of different integrators in a fair manner, it is important to ensure that the accuracy obtained with those integrators is comparable. In this section, the accuracy ϵ is defined as the norm of the difference between the truth and the state vector at the final time of the integrator being tested, as shown in Eq. (37). The truth is obtained using the F and G series method at high order with quad precision arithmetic and a sufficiently small time step such that the max change in the local Hamiltonian of the truth trajectory is $< 10^{-25}$.

$$\epsilon = \|X(\tau_f)_{truth} - X(\tau_f)\| \tag{37}$$

The comparison scheme relies on the computation of n, the minimum number of steps necessary to achieve a prescribed accuracy. This dynamic selection of n ensures that the accuracy is reached, while minimizing the computational efforts for each integrator being tested. The comparisons, made according to Algorithm 2, allow for a fair evaluation of the timings: the accuracy and perturbation discretization achieved by each integration method are essentially the same.

Algorithm 2 Comparison of the timings for all integrators

1: Initialize the propagation (choose $a, e, \mathbf{p}, \alpha, \tau_f, N, \epsilon$)
2: Compute the truth using the F and G series method at high order with a small step size and using quad precision
3: Verify that the maximum change in Hamiltonian in truth never exceeds 10^{-25}
4: Compute the minimum number of steps n_{RK} necessary to achieve the tolerance ϵ with RKF8, using either the τ-velocity or time-velocity formulation.
5: **for** $k \leftarrow 1, N$ **do**
6: Compute the minimum number of steps n_k required to achieve ϵ using each of the TS methods and order k
7: **for** $i \leftarrow 1, 10000$ **do**
8: Propagate the orbit for n_k steps, using the corresponding TS method and time the execution using the `CPU_TIME` Fortran procedure
9: **end for**
10: Average the timings
11: Compute n_{st} =`CEILING(`n_{RK}/n_k`)`. The RKF8 propagation is done over n_k segments of n_{st} steps each, simulating a classic discretization of the orbit in n_k segments
12: **for** $i \leftarrow 1, 10000$ **do**
13: Propagate the orbit using RKF8 and time the execution using `CPU_TIME` procedure
14: **end for**
15: Average the timings
16: **end for**



(a) Tolerance

(b) Eccentricity

(c) Perturbation

(d) Sundman transformation α. For $\alpha = 0$ and order below 10, the Taylor series solutions did not reach the prescribed accuracy $\epsilon = 10^{-12}$

Figure 5. Timing results when varying integration parameters, for all integrators

In order to present a complete set of test cases, the procedure described in Algorithm 2 is repeated for a variety of parameters and initial conditions. The nominal trajectory is an ellipse specified by $a = 1\ LU$ and $e = 0.8$, and is in the plane of the first two coordinates. The nominal perturbation is $\mathbf{p} = 10^{-3}[1, 1, 1]^T\ LU/TU^2$ and the nominal tolerance is $\epsilon = 10^{-12}$. Figure 5 presents the timings obtained using all integration methods while varying the tolerance, the eccentricity, the magnitude of the perturbation and the Sundman transformation power law.

Figure 5 shows that the Taylor series solutions are significantly faster than the RKF8 integration in most cases. It also shows that the τ-velocity solution is faster than its time-velocity counterpart, for reasons previously described. The timings of the RKF8 are approximately constant over any TS order. The increase in runtime for the RKF8 integrator at the low TS orders occurs because the integrator cannot take less than 1 step per segment. Therefore in these regions, a lower order Runge-Kutta integrator would be necessary to reduce runtime without over-achieving the accuracy. The F and G series demonstrate optimal efficiency for orders between 10 and 15. The MTS method becomes more efficient than the F and G method beyond order 18.

The F and G series are increasingly inefficient at high orders due to the complexity mentioned in Subsec-

tion 3.2, making their coefficients difficult to obtain beyond order 25. Alternatively, it is noted that the MTS method easily achieves orders of 100 or more, although numerical issues have been observed to accumulate beginning around order 25.

Figure 5(a) shows that the prescribed tolerance, while having a significant influence on the timings, does not modify the relative performance of the methods. It is observed in Fig. 5(b) that the eccentricity, while significantly affecting the timings of the Runge-Kutta method, leaves the timings of the Taylor series methods essentially unchanged, and therefore has an important impact on the relative performances of the integrators. The speedups of the F and G Stark series solution over the RKF8 integrator go from $5\times$ for $e = 0$ up to $50\times$ for $e = 0.95$. Figure 5(c) shows that the magnitude of the perturbation does not alter the performance of the RKF8 integrations. However, it does have a minor impact on the timings of the Taylor series solutions, and therefore on the relative performances. A large perturbation tends to counteract the benefits of using the Taylor series. Finally, Fig. 5(d) demonstrates the influence of the Sundman transformation power law. The runtimes to achieve the prescribed accuracy for all methods are higher when $\alpha = 0$ (corresponding to no transformation). As expected from the results in Figures 3 and 4, the F and G series has best results for $\alpha = 1$ for the majority of orders.

It is emphasized that the favorable results of the TS methods are in part due to two beneficial behaviors. First, the formulation presented in Section 3 is simple enough to generate relatively small coefficient files. The simple nature of the 2-body equations of motion and the use of an inertially constant perturbation $\mathbf{p}$ as the third basis vector greatly influences the efficiency of the method. It is noted that the authors successfully adapted this F and G series method to other more complicated dynamical systems (e.g. the restricted 3-body problem, and the 2-body problem with a perturbation constant in a rotating frame). However, the timings obtained for those scenarios did not present as significant an improvement when compared to the RKF8 timings. Second, the behavior of the local error for the Taylor series solutions coupled with the Sundman transformation (see Subsection 4.3) is serendipitous, and highly beneficial to the results. Indeed, when the independent variable is proportional to the eccentric anomaly, the Taylor series methods abnormally require fewer steps to achieve equivalent accuracies when compared to all other techniques considered.

4.5 Application: low-thrust orbital transfer

In order to show the utility of the methods presented in this paper, an application to a low-thrust trajectory is presented. The problem consists of a transfer from a circular Low-Earth Orbit (LEO) to a highly-elliptical orbit. To simplify the implementation and to avoid Keplerian arcs, the selected strategy involves thrusting constantly. The thrust is directed along the velocity vector at the beginning of each segment, and its magnitude varies according to Eq. (38). In order to propagate this trajectory using the Stark formulation, a discretization in a finite number of segments with constant perturbation is necessary. For segment i, from t_i to t_{i+1}:

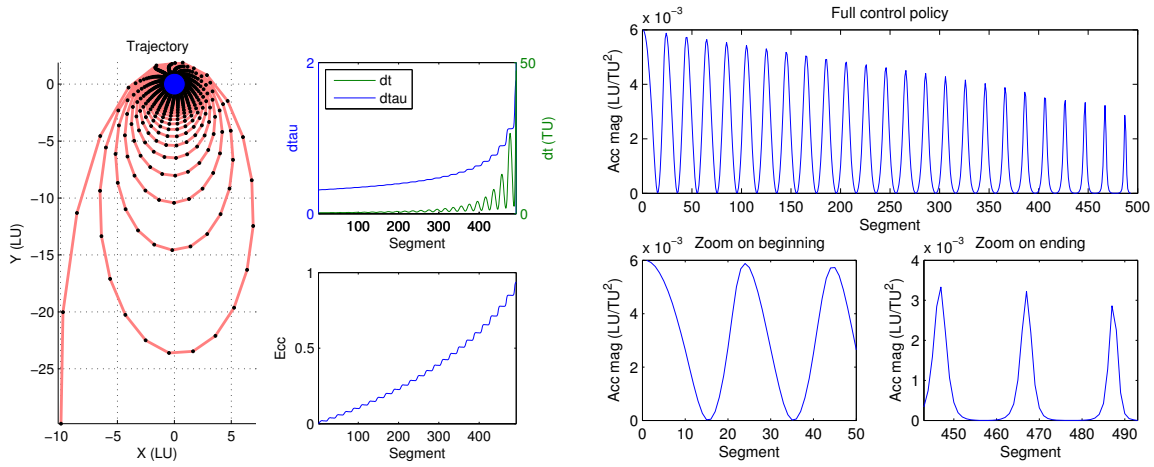
$$\mathbf{p}_i = \frac{q}{r(t_i)}[1 + \cos\nu(t_i)]\mathbf{v}(t_i) \tag{38}$$

where ν is the true anomaly and q is a constant ($q = 10^{-3}$ in the following).

The control policy and $\Delta\tau$ are computed at each step of the propagation, until a final condition is reached, as described in Algorithm 3 in Appendix A. Figure 6(a) presents the trajectory resulting from this control policy with a discretization of 20 Segments Per Revolution (SPR). The eccentricity increases from 0 to 0.94, with large changes when the particle is close to periapse (when the thrust is the largest). The size of the τ-steps has a similar behavior. Figure 6(b) shows the variations in the magnitude of the thrust.

In this application, the accuracy is represented by the RMS of the relative change in Hamiltonian across all K segments, as shown in Eq. (39). The accuracy is calculated in this manner to emulate a typical application where the truth is unknown. The change in Hamiltonian along the trajectory resembles the change presented in Fig. 3, with minor differences due to the presence of a perturbation.

$$\epsilon = \sqrt{\frac{1}{K}\sum_{i=0}^{K}\left(\frac{H_{i+1} - H_i}{H_i}\right)^2} \tag{39}$$



(a) Trajectory (b) Control Policy

Figure 6. Application: low-thrust orbital transfer

The method described in Algorithm 3 (see Appendix A) is used to compare the different integration methods. The test is designed to compare total compute speeds of the Taylor series solutions to the speeds of the RKF8 integrations, while preserving a similar accuracy for any two methods being compared.

Figures 7 and 8 present the results in terms of 1) achieved speedup and 2) order of magnitude of the common accuracy, as functions of the SPR discretization. The goal of displaying the data in this manner is to emulate realistic trajectory design applications. The desired accuracy and associated model discretization typically have large effects on the number of optimization variables, and therefore are of primary interest to a trajectory designer. Figures 7 and 8 provide the expected speedups with respect to these two important design variables. Each curve in Figures 7 and 8 stops when the speedup for a larger SPR is significantly smaller than that for a smaller SPR. As a matter of fact, a large decrease of the speedup means that the accuracy has reached machine precision for this order, and increasing the SPR will only increase the runtime, not the precision. The confirmation of this behavior can be observed on the accuracy plots of Figures 7 and 8.

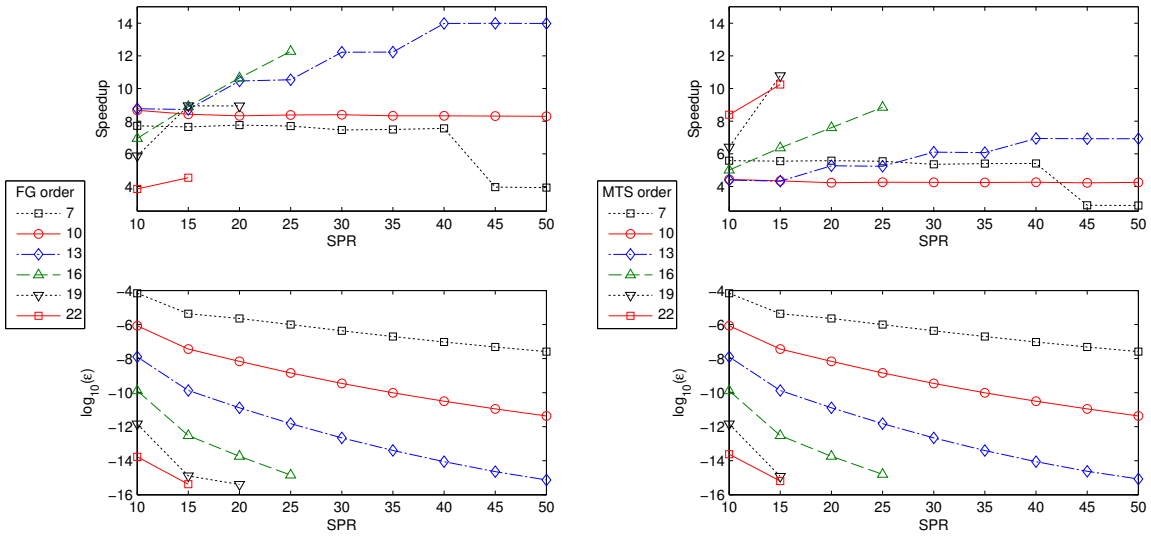
Figure 7(a) shows that the F and G Stark series method provides speedups between $4\times$ and $14\times$ when compared to the time-velocity RKF8. For orders 7 and 10, the speedups are relatively constant across all discretizations, and the accuracy is low, as expected. For high orders (> 15), the accuracy is high, but the speedups are lower, confirming the tendency observed in Subsection 4.4. The best results for the F and G method are obtained for intermediate orders (between 10 and 15), and for discretizations with SPR > 20. A low SPR results in lower accuracies for the TS methods, and since the RKF8 does not have to take many steps per segments to match these accuracies, the associated speedups are low.

Figure 7(b) shows the results obtained when comparing the MTS method to the RKF8 integrations. On a general level, the speedups are lower, confirming the findings of Subsection 4.4. The main difference comes from the higher orders: for low SPR, the MTS method leads to high speedups, confirming the previous results that the MTS method is faster than the F and G method for orders above 18.

Figure 8 presents the results obtained when comparing the Taylor series solutions to the τ-velocity RKF8. As expected, the behaviors are identical, with lower speedups due to the better efficiency of the τ-velocity RKF8.

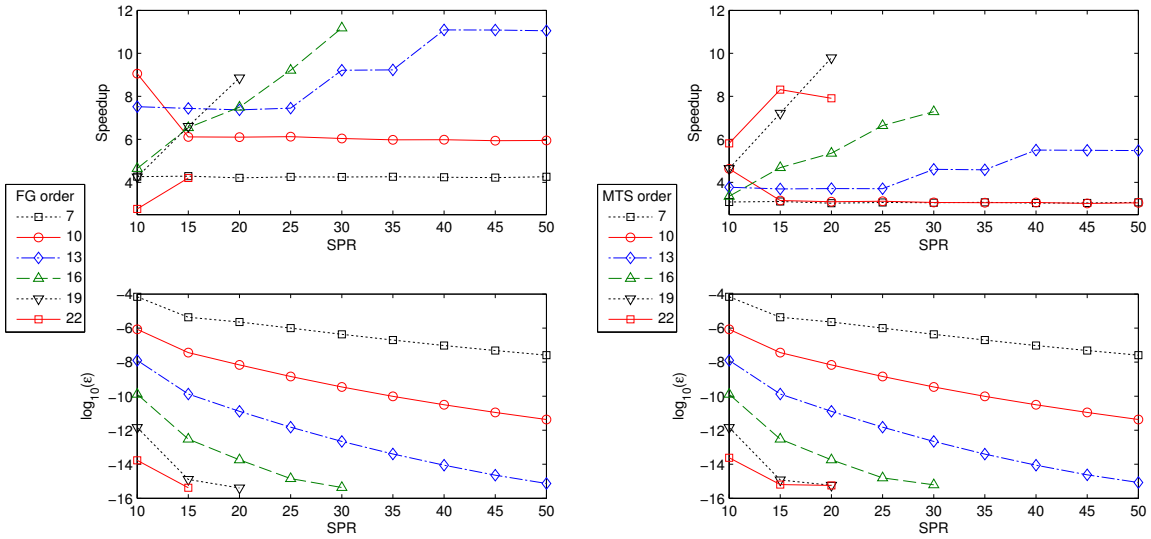
5 CONCLUSION

In this paper, the Lagrange coefficients and their associated F and G Taylor series are derived for the well-known Stark problem. The Stark effect can be used to approximate a variety of astrodynamics systems, such



(a) F and G (b) MTS

Figure 7. Results of the application: Speedup of Taylor series solutions compared to time-velocity RKF8



(a) F and G (b) MTS

Figure 8. Results of the application: Speedups of Taylor series solutions compared to τ-velocity RKF8

as low-thrust trajectories or slowly varying perturbations. The traditional F and G series are extended to 3D and to include an inertially constant perturbation. The generalized Sundman transformation is used to allow for powerful discretization schemes, decreasing the need for a variable-step method.

It is shown that the Sundman exponent has great influence on the discretization and the uniformization of the local error. The Taylor series methods exhibit a peculiar behavior for the unity power law, which corresponds to making the independent variable proportional to the eccentric anomaly, and results in a regular geometric discretization. This case is proven to greatly enhance the accuracy of both Taylor series methods,

and therefore their computational efficiencies. This result is a significant finding, since the Sundman transformation is conventionally assumed to perform best with the $3/2$ power law. This peculiar behavior of the Taylor series solutions make them ideal candidates for use in applications such as low-thrust optimization, where regular geometric discretizations are naturally preferred.

The F and G Stark series integration method is shown to perform between 3 and 50 times faster than a conventional RKF8 integration scheme, where performance is tied mostly to eccentricity, order of the Taylor series, and the exponent in the Sundman transformation. The new F and G Stark series method proves most efficient for orders between 10 and 15, corresponding to discretizations of approximately 20-40 points per revolution. High eccentricities are shown to increase the relative efficiency of the method, while large perturbations tend to diminish the efficiency. The Modern Taylor Series method is used as a second benchmark to compare computational efficiency. The MTS method is shown to be slower for low to intermediate orders, but becomes more efficient than the F and G Stark series method for orders above 18. Therefore, the MTS method could be used for high accuracy applications that afford large step sizes. It is noted that the particular cases of the 2-body and Stark problems are especially well suited for the Taylor series solutions. More complicated equations of motion have been implemented, but the improvement due to the use of Taylor series is much less significant than those reported in this study. A link to download the Fortran coefficient files for the F and G Stark series used in this study is provided in Appendix B.

ACKNOWLEDGEMENTS

This material is based upon work partially supported by the Air Force Research Laboratory under under Contract No. FA9453-13-C-0205.

REFERENCES

[1] K. Fox, "Numerical Integration of the Equations of Motion of Celestial Mechanics," *Celestial Mechanics*, Vol. 33, No. 2, 1984, pp. 127–142.

[2] O. Montenbruck, "Numerical Integration Methods for Orbital Motion," *Celestial Mechanics and Dynamical Astronomy*, Vol. 53, 1992, pp. 59 – 69.

[3] B. Jones and R. Anderson, "A Survey of Symplectic and Collocation Integration Methods for Orbit Propagation," *AAS/AIAA Spaceflight Mechanics Conference*, 2012, pp. 1–20.

[4] O. Montenbruck, "Numerical Integration of Orbital Motion using Taylor Series," *AAS/AIAA Spaceflight Mechanics Conference*, Colorado Springs, CO, AAS, 1992.

[5] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics*. Education Series, American Institute of Aeronautics and Astronautics, Inc, 1999.

[6] P. Sconzo, A. LeSchack, and R. Tobey, "Symbolic Computation of f and g Series by Computer," *The Astronomical Journal*, Vol. 70, No. 4, 1965, pp. 269 – 270.

[7] V. R. Bond, "A Recursive Formulation for Computing the Coefficients of the Time-Dependent f and g Series Solutions to the Two-Body Problem," *The Astronomical Journal*, Vol. 71, No. 1, 1966, pp. 7–8.

[8] J. F. Steffensen, "On the Restricted Problem of Three Bodies," *Kongelige Danske Videnskabernes Selskab Mat.-Fys. Medd.*, Vol. 30, No. 18, 1956.

[9] E. Rabe, "Determination and Survey of Periodic Trojan Orbits in the Restricted Problem of Three Bodies," *The Astronomical Journal*, Vol. 66, No. 9, 1961, pp. 500 – 513.

[10] R. Broucke, "Solution of the N-Body Problem with Recurrent Power Series," *Celestial Mechanics*, Vol. 4, No. 1, 1971, pp. 110–115.

[11] D. N. Papadakos, "Generalized F and G Series and Convergence of the Power Series Solution to the N-Body Problem," *Celestial Mechanics*, Vol. 30, No. 1, 1983, pp. 275–282.

[12] J. Bem and B. Szczodrowska-Kozar, "High Order f and g Power Series for Orbit Determination," *Astronomy and Astrophysics Supplement*, Vol. 110, 1995, pp. 411–417.

[13] R. R. Bate, D. D. Mueller, and J. E. White, *Fundamentals of Astrodynamics*. New-York, NY: Dover Publications, 1971.

[14] T. T. Soong and N. A. Paul, "A Second- and Higher Order Perturbation Analysis of Two-Body Trajectories," *AIAA Journal*, Vol. 9, No. 4, 1971, pp. 589–593.

[15] M. Sharifi and M. Seif, "Dynamic orbit propagation in a gravitational field of an inhomogeneous attractive body using the Lagrange coefficients," *Advances in Space Research*, Vol. 48, Sept. 2011, pp. 904–913, 10.1016/j.asr.2011.04.021.

[16] J. Stark, "Beobachtungen über den Effekt des elektrischen Feldes auf Spektrallinien.," *Annalen der Physik*, Vol. 348, No. 7, 1914.

[17] G. Lantoine and R. P. Russell, "Complete closed-form solutions of the Stark problem," *Celestial Mechanics and Dynamical Astronomy*, Vol. 109, Feb. 2011, pp. 333–366, 10.1007/s10569-010-9331-1.

[18] C. H. Yam, D. Izzo, and F. Biscani, "Towards a High Fidelity Direct Transcription Method for Optimisation of Low-Thrust Trajectories," *4th International Conference on Astrodynamics Tools and Techniques*, 2010, pp. 1–7.

[19] F. Zuiani, M. Vasile, A. Palmas, and G. Avanzini, "Direct transcription of low-thrust trajectories with finite trajectory elements," *Acta Astronautica*, Vol. 72, Mar. 2012, pp. 108–120, 10.1016/j.actaastro.2011.09.011.

[20] J. A. Sims and S. N. Flanagan, "Preliminary Design of Low-Thrust Interplanetary Missions," *AAS/AIAA Astrodynamic Specialist Conference*, Girdwood, AK, USA, 1999.

[21] U. R. S. Kirchgraber, "A problem of orbital dynamics, which is separable in KS-Variables," *Celestial Mechanics*, Vol. 4, No. 3, 1971, pp. 340–347.

[22] D. Rufer, "Trajectory Optimization by Making Use of the Closed Solution of Constant Thrust-Acceleration Motion," *Celestial Mechanics*, Vol. 14, No. 1, 1976, pp. 91–103.

[23] S. M. Poleshchikov, "One Integrable Case of the Perturbed Two-Body Problem," *Cosmic Research*, Vol. 42, July 2004, pp. 398–407, 10.1023/B:COSM.0000039740.22909.ee.

[24] K. F. Sundman, "Mémoire sur le problème des trois corps," *Acta Mathematica*, Vol. 36, Dec. 1912, pp. 105–179, 10.1007/BF02422379.

[25] M. M. Berry and L. M. Healy, "The generalized Sundman transformation for propagation of high-eccentricity elliptical orbits," *AAS/AIAA Space Flight Mechanics Meeting*, San Antonio, TX, Jan. 2002.

[26] C. E. Velez, "Notions of Analytic vs. Numerical Stability as applied to the Numerical Calculation of Orbits," *Celestial Mechanics*, Vol. 10, No. 4, 1974, pp. 405–422.

[27] J. Baumgarte, "Numerical Stabilization of the Differential Equations of Keplerian Motion," *Celestial Mechanics*, Vol. 5, No. 4, 1972, pp. 490–501.

[28] P. Nacozy, "A Discussion of Time Transformations and Local Truncation Errors," *Celestial Mechanics*, Vol. 13, No. 4, 1976, pp. 495–501.

[29] A. Deprit and R. V. M. Zahar, "Numerical Integration of an Orbit and Its Concomitant Variations by Recurrent Power Series," *Zeitschrift für angewandte Mathematik und Physik ZAMP*, Vol. 17, No. 3, 1966, pp. 425–430.

[30] A. Jorba and M. Zou, "A Software Package for the Numerical Integration of ODEs by Means of High-Order Taylor Methods," *Experimental Mathematics*, Vol. 14, No. 1, 2005, pp. 99–117.

[31] J. R. Scott and M. C. Martini, "High-Speed Solution of Spacecraft Trajectory Problems Using Taylor Series Integration," *Journal of Spacecraft and Rockets*, Vol. 47, No. 1, 2010, pp. 199–202.

[32] V. Szebehely and D. Pierce, "Advantages of regularization in space dynamics.," *AIAA Journal*, Vol. 5, No. 8, 1967, pp. 1520–1522.

[33] E. Stiefel, "Remarks on Numerical Integration of Keplerian Orbits," *Celestial Mechanics*, Vol. 2, No. 5, 1970, pp. 274–281.

[34] V. R. Bond, "A Transformation of the Two-Body Problem," *Celestial Mechanics*, Vol. 35, No. 1, 1985, pp. 1–7.

[35] T. Feagin and R. P. Mikkilineni, "The Effect of Time Transformations on Local Truncation Errors," *Celestial Mechanics*, Vol. 13, No. 4, 1976, pp. 491–493.

[36] R. H. Merson, "Numerical Integration of the Differential Equations of Celestial Mechanics," tech. rep., Royal Aircraft Establishment, Farnborough, Hants., England, 1973.

[37] P. Nacozy, "The Intermediate Anomaly," *Celestial Mechanics*, Vol. 16, No. 3, 1977, pp. 309–313.

[38] V. Szebehely and V. Bond, "Transformations of the Perturbed Two-Body Problem to Unperturbed Harmonic Oscillators," *Celestial Mechanics*, Vol. 30, No. 1, 1983, pp. 59–69.

[39] S. Ferrer and M. L. Sein-Echaluce, "On the Szebehely-Bond Equation. General Sundman's Transformation for the Perturbed Two-Body Problem," *Celestial Mechanics*, Vol. 32, No. 4, 1984, pp. 333–347.

[40] C. Bischof, A. Carle, G. Corliss, A. Griewank, and P. D. Hovland, "ADIFOR: Generating Derivative Codes from Fortran Programs," *Scientific Programming*, Vol. 1, No. 1, 1992, pp. 1–29.

[41] A. Walther, "Getting Started with ADOL-C," *Combinatorial Scientific Computing* (U. Naumann and O. Schenk, eds.), ch. 7, pp. 181–202, Chapman-Hall CRC Computational Science, 2012.

[42] C. Bendtsen and O. Stauning, "FADBAD, a Flexible C++ Package for Automatic Differentiation," tech. rep., Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1996, IMM–REP–1996–17.

[43] A. Gofen, "Interactive Environment for the Taylor Integration (in 3D Stereo)," *Proceedings of the 2005 International Conference on Scientific Computing (CSC 05)*, CSREA Press, 2005.

APPENDIX A: ALGORITHM USED FOR THE APPLICATION OF ALL INTEGRATORS TO AN EXAMPLE LOW-THRUST ORBITAL TRANSFER

Algorithm 3 Application of all integrators to a low-thrust orbital transfer

1: **for** SPR $\leftarrow 5, 50$ **do** ▷ SPR: Segments Per Revolution
2: For the Taylor series:
3: **for** order $\leftarrow 3, 25$ **do**
4: Call `propagateTrajectory` and store the control policy and $\Delta\tau$ histories and accuracy. The propagation stops when the final condition is reached, and the necessary number of segments K is stored.
5: **for** $i \leftarrow 1, 10000$ **do**
6: Propagate trajectory using the precomputed control and $\Delta\tau$ histories $\{\mathbf{p}_k, \Delta\tau_k\}$ for $k = 0, K$
7: **end for**
8: Average the timings
9: **end for**
10: For the RKF8: Repeat (3-10) replacing order from line 3 by SPS ▷ SPS: Steps Per Segment
11: **for** order $\leftarrow 3, 25$ **do**
12: Read accuracy of the Taylor series solution
13: Select SPS that produces closest accuracy for the RKF8 solution
14: Compute speedup
15: **end for**
16: **end for**

17: **procedure** PROPAGATETRAJECTORY
18: **Input**: $\mathbf{X}_0$, SPR, order N, r_{max}, μ ▷ $\mathbf{X}$ is the full state vector (position, velocity, time)
19: Initialize $k \leftarrow 0, \epsilon \leftarrow 0$
20: **repeat**
21: Compute τ_p using Table 1 and $\mathbf{X}_k$
22: $\Delta\tau_k \leftarrow \tau_p/\text{SPR}$
23: Compute $\mathbf{p}_k$ using Eq. (38) and $\mathbf{X}_k$
24: Compute Hamiltonian H_k for $\mathbf{X}_k$, $\mathbf{p}_k$ using Eq. (36)
25: Propagate $\mathbf{X}_k(\tau)$ to $\mathbf{X}_{k+1}(\tau + \Delta\tau)$ with $\mathbf{p}_k$ for order N
26: Compute Hamiltonian H_{k+1} for $\mathbf{X}_{k+1}$, $\mathbf{p}_k$
27: $\epsilon \leftarrow \epsilon + (H_{k+1} - H_k)^2$
28: $k \leftarrow k + 1$
29: **until** $\|\mathbf{r}\| \geq r_{max}$
30: $K \leftarrow k - 1$
31: $\epsilon \leftarrow \epsilon/K$
32: **Output**: $\mathbf{X}_f$, ϵ, K, $\{\mathbf{p}_k, \Delta\tau_k\}$ for $k = 0, K$
33: **end procedure**

APPENDIX B: IMPLEMENTING THE F AND G STARK SERIES WITH A SYMBOLIC MANIPULATOR

The coefficients of the F and G series are computed using the equations in Subsection 3.2 and implemented using the symbolic manipulator Maple 15.01. The `simplify(.,'size')` command and the `optimize` option of the code generation are used to minimize file sizes and the runtime of the resulting code. It is important to note that the order of the F and G method, in theory unbounded, is practically limited by the size of the corresponding coefficients file, and the ability for a compiler to handle those files. The Fortran source code of the coefficients computation can be downloaded from: `http://russell.ae.utexas.edu/index_files/fgstark.htm`. The size of the files ranges between 4 and 8MB. The coefficients and associated intermediate variables up to order 6 are shown below. The intermediate variables should be computed in ascending order.

$k_1 = \mathbf{r}\cdot\mathbf{v}$ $\quad k_2 = \mathbf{p}\cdot\mathbf{r}$ $\quad k_3 = \mathbf{p}\cdot\mathbf{v}$ $\quad k_4 = \mathbf{v}\cdot\mathbf{v}$ $\quad s = r^\alpha$

$t1 = r^\alpha$ $\quad t10 = \frac{k_1}{t3}$ $\quad t109 = k_3 t17$ $\quad t12 = 1/2\alpha t2 t10$ $\quad t120 = t16 r$

$t131 = t56 t1$ $\quad t133 = t17^2$ $\quad t139 = p^2$ $\quad t14 = t2 t1$ $\quad t140 = \alpha t82 t139$

$t142 = k_3 k_1$ $\quad t144 = t26^2$ $\quad t16 = \alpha - 1$ $\quad t162 = t24 t26$ $\quad t163 = t162 t62$

$t166 = \frac{7}{24} - \frac{47}{12} t24 t26$ $\quad t168 = 2t24 t26$ $\quad t17 = t3^2$ $\quad t178 = t24^2$ $\quad t179 = \alpha t178$

$t18 = t17 r$ $\quad t180 = \alpha - 4/3$ $\quad t182 = \alpha - 3/2$ $\quad t187 = t133^{-1}$ $\quad t2 = t1^2$

$t23 = \alpha^2$ $\quad t230 = t178 t16$ $\quad t231 = t180 t182$ $\quad t24 = {k_1}^2$ $\quad t25 = t23 t24$

$t254 = -\frac{52}{137} t26 t23$ $\quad t26 = k_4 + k_2$ $\quad t263 = \frac{421}{274} t163$ $\quad t27 = 1/2 t26 t3$ $\quad t28 = 1/2 r$

$t283 = t56 t2$ $\quad t289 = k_1 t139$ $\quad t291 = k_3 t26$ $\quad t3 = r^2$ $\quad t33 = t17^{-1}$

$t334 = t23^2$ $\quad t355 = t178 k_1$ $\quad t359 = t231(\alpha - 8/5)$ $\quad t364 = \frac{1}{t133 t3}$ $\quad t39 = t24 t16$

$t4 = t3 r$ $\quad t56 = t2^2$ $\quad t62 = t23\alpha$ $\quad t66 = t24 k_1$ $\quad t69 = -1 + t26 r$

$t73 = \alpha - 6/7$ $\quad t78 = \alpha k_3 t17$ $\quad t82 = t17 t3$ $\quad t83 = t82^{-1}$ $\quad t87 = \alpha t3$

$F_1 = 0$ $\quad G_1 = t1$ $\quad H_1 = 0$ $\quad T_1 = t1$

$F_2 = -1/2 \frac{t2}{t4}$ $\quad G_2 = t12$ $\quad H_2 = 1/2 t2$ $\quad T_2 = t12$

$F_3 = -1/2 \frac{t14 k_1 t16}{t18}$ $\quad G_3 = 1/3(t25 + (t27 - t28 - t24)\alpha - t28) t14 t33$

$H_3 = 1/2 t14 \alpha t10$ $\quad T_3 = 1/3\alpha(t27 - t28 + t39) t14 t33$

$F_4 = -\frac{11}{24}\big(4/11(\alpha - 3/4) t26 t3 + (2/11 - 4/11\alpha) r + t24\big(\alpha - \frac{15}{11}\big) t16\big) t56 t17^{-1} t4^{-1}$

$G_4 = 1/4 t56((t62 - 7/3 t23 + 4/3\alpha) t66 + 7/6 r(-1 + t69\alpha) t73 k_1 + 1/2 t78) t83$

$H_4 = 1/24 t56(11 t25 + 4 t87 k_4 - 4\alpha r + 4 t87 k_2 - 8\alpha t24 - r) t33$

$T_4 = 1/4\alpha((t23 - 7/3\alpha + 4/3) t66 + 7/6 r(t26 t73 r + 5/7 - \alpha) k_1 + 1/2 t109) t56 t83$

$F_5 = -\frac{5}{12}\big((t62 - \frac{39}{10} t23 + 5\alpha - \frac{21}{10}) t66 + \frac{9}{10} t120(t26 t16 r - \alpha + 2/3) k_1 + 3/10 t109(\alpha - 3/5)\big) t131 t133^{-1} r^{-1}$

$G_5 = 1/5\big(1/8 t140 + \frac{7}{24}\alpha\big(\big(\frac{33}{7} t142 + t144\big)\alpha - \frac{24}{7} t142 - 6/7 t144\big) t17 + \big(-\frac{7}{12} t26 t23 + (-1/12 k_2 + 1/24 k_4)\alpha + 3/8 k_2 + 3/8 k_4\big) t4 + \big(-1/3 + \frac{23}{12} t163 + t166 t23 + \big(\frac{5}{24} + t168\big)\alpha\big) t3 - \frac{23}{12} t39\big(t23 - 1/23\alpha - \frac{45}{46}\big) r + t179 t16 t180 t182\big) t131 t187$

$H_5 = \frac{5}{12} t131\big((t62 - 9/5 t23 + 4/5\alpha) t66 + \frac{9}{10} r(1/5 + t69 t23 + (1/3 - 2/3 t26 r)\alpha) k_1 + 3/10 t78\big) t83$

$T_5 = 1/5\alpha\big(1/8 t82 t139 + \big(\big(\frac{11}{8} t142 + \frac{7}{24} t144\big)\alpha - t142 - 1/4 t144\big) t17 + \big(-\frac{7}{12} t26\alpha + \frac{11}{24} k_4 + 1/3 k_2\big) t4 + \big(\frac{23}{12} t162 t23 + t166\alpha - \frac{5}{24} + t168\big) t3 - \frac{23}{12} t39\big(\alpha - \frac{37}{46}\big) r + t230 t231\big) t131 t187$

$F_6 = -\frac{137}{360}\big(\frac{9}{137} t139(\alpha - 1/2) t82 + \big(\big(\frac{243}{274} t142 + \frac{26}{137} t144\big) t23 + \big(-\frac{207}{137} t142 - \frac{48}{137} t144\big)\alpha + \frac{90}{137} t142 + \frac{45}{274} t144\big) t17 + \big(t254 + \big(\frac{83}{137} k_4 + \frac{74}{137} k_2\big)\alpha - \frac{57}{274} k_2 - \frac{33}{137} k_4\big) t4 + \big(t263 + \big(\frac{26}{137} - \frac{722}{137} t24 t26\big) t23 + \big(-\frac{35}{137} + \frac{1653}{274} t24 t26\big)\alpha + \frac{11}{137} - \frac{315}{137} t24 t26\big) t3 - \frac{421}{274} t39\big(t23 - \frac{857}{421}\alpha + \frac{420}{421}\big) r + t230\big(t23 - \frac{416}{137}\alpha + \frac{315}{137}\big) t182\big) t283 t133^{-1} t4^{-1}$

$G_6 = 1/6 t283\big(5/8\alpha\big(\big(\frac{16}{25} t289 + t291\big)\alpha - 2/5 t289 - 4/5 t291\big) t82 - 5/8 k_3\big(t23 - \frac{3}{25}\alpha - \frac{12}{25}\big) t18 + \frac{127}{120}\alpha k_1\big(\big(\frac{303}{127} t142 + t144\big) t23 + \big(-\frac{534}{127} t142 - \frac{246}{127} t144\big)\alpha + \frac{240}{127} t142 + \frac{120}{127} t144\big) t17 - \frac{127}{60}\big(t26 t62 + \big(-\frac{119}{127} k_2 - \frac{143}{127} k_4\big) t23 + \big(-\frac{72}{127} k_4 - \frac{183}{254} k_2\big)\alpha + \frac{90}{127} k_2 + \frac{90}{127} k_4\big) k_1 t4 + \frac{163}{60}\big(\frac{75}{163} + t162 t334 + \big(\frac{127}{326} - \frac{557}{163} t24 t26\big) t62 + \big(-\frac{20}{163} + \frac{634}{163} t24 t26\big) t23 + \big(-\frac{233}{326} - \frac{240}{163} t24 t26\big)\alpha\big) k_1 t3 - \frac{163}{60} t66\big(t62 - \frac{231}{163} t23 - \frac{130}{163}\alpha + \frac{210}{163}\big) t120 + \alpha t355 t16 t359\big) t364$

$H_6 = \frac{137}{360}\big(\frac{9}{137} t140 + \frac{26}{137}\alpha\big(\big(\frac{243}{52} t142 + t144\big)\alpha - \frac{36}{13} t142 - \frac{9}{13} t144\big) t17 + \big(t254 + \big(\frac{14}{137} k_2 + \frac{23}{137} k_4\big)\alpha + \frac{9}{137} k_2 + \frac{9}{137} k_4\big) t4 + \big(-\frac{17}{274} + t263 + \big(\frac{26}{137} - \frac{347}{137} t24 t26\big) t23 + \big(-\frac{5}{137} + \frac{144}{137} t24 t26\big)\alpha\big) t3 - \frac{421}{274} t24\big(\frac{47}{421}\alpha + t62 - \frac{528}{421} t23 + \frac{90}{421}\big) r + t179 t16\big(t23 - \frac{284}{137}\alpha + \frac{144}{137}\big)\big) t283 t187$

$T_6 = 1/6\alpha t283\big((2/5 t139(-5/8 + \alpha) k_1 + 5/8(\alpha - 4/5) t26 k_3) t82 - 5/8\big(\alpha - \frac{18}{25}\big) k_3 t18 + \frac{127}{120} k_1\big(\frac{303}{127}\big(t23 - \frac{178}{101}\alpha + \frac{80}{101}\big) k_3 k_1 + t144\big(\frac{120}{127} + t23 - \frac{246}{127}\alpha\big)\big) t17 - \frac{127}{60}\big(t26 t23 + \big(-\frac{403}{254} k_2 - \frac{451}{254} k_4\big)\alpha + \frac{201}{254} k_4 + \frac{81}{127} k_2\big) k_1 t4 + \frac{163}{60} k_1((t23 - \frac{394}{163}\alpha + \frac{240}{163}) t16 t162 + \frac{127}{326} t23 + \frac{41}{163} - \frac{205}{326}\alpha) t3 - \frac{163}{60} t66\big(t23 - \frac{687}{326}\alpha + \frac{355}{326}\big) t120 + t355 t16 t359\big) t364$