A COMPARISON OF EXPLICIT RUNGE–KUTTA METHODS

STEPHEN J. WALTERS ⓘ ✉1, ROSS J. TURNER ⓘ1 and LAWRENCE K. FORBES ⓘ1

Abstract

Recent higher-order explicit Runge–Kutta methods are compared with the classic fourth-order (RK4) method in long-term integration of both energy-conserving and lossy systems. By comparing quantity of function evaluations against accuracy for systems with and without known solutions, optimal methods are proposed. For a conservative system, we consider positional accuracy for Newtonian systems of two or three bodies and total angular momentum for a simplified Solar System model, over moderate astronomical timescales (tens of millions of years). For a nonconservative system, we investigate a relativistic two-body problem with gravitational wave emission. We find that methods of tenth and twelfth order consistently outperform lower-order methods for the systems considered here.

1. Introduction

A popular technique for the numerical solution of many initial-value problems is the family of Runge–Kutta (RK) methods. An historical overview of this approach to solving differential equations is given by Butcher [7]. Explicit Runge–Kutta (ERK) methods are straightforward to code, because they give explicit formulae for the solution at the next time-step, in terms only of the solution at the current time. The most famous of these is the classical fourth-order, four-stage method in the text by Atkinson [3, p. 371]. An improved approach, which still retains the relative simplicity of ERK methods, is provided by the family of Runge–Kutta–Fehlberg methods [3, p. 376], in which pairs of RK methods are used together to create explicit formulae

---

[1]Mathematics Department, University of Tasmania, Tasmania 7005, Australia;
e-mail: stephen.walters@utas.edu.au, ross.turner@utas.edu.au and larry.forbes@utas.edu.au.

that control the growth of numerical error by adaptive time-stepping. A survey of these ideas is presented by Enright et al. [12], who also discuss concepts such as the detection of stiff zones in the solution and the development of ERK methods specifically for use in systems of second-order equations.

In this current work, we will directly compare the performance of some of the explicit RK methods developed in the last twenty years with the classic fourth-order Runge–Kutta (RK4) method. To make this comparison, we will consider conservative Newtonian orbits as well as a nonconservative relativistic binary system. The aim of this work is to identify which ERK methods are optimal for solving different systems of equations and to discuss any pertinent considerations.

The Newtonian equations for orbital motion of planets in the Solar System are straightforward to write down and lead to a system of conservative second-order differential equations for the positions (and velocities) of each planet. Since this n-body system is nonlinear, it is capable of exhibiting complex behaviours, including resonances, instability, quasi-periodicity and chaos. Long-term simulation of orbital motion therefore requires numerical integration techniques of extremely high accuracy and the question naturally arises as to how far forward in time it is possible to integrate these differential equations, using any quadrature technique, before accumulated numerical error renders the results meaningless.

Anastassi and Simos [2] discuss RK methods for such second-order systems for orbital motions. Because orbital motion is one in which overall energy is conserved, attention has also focussed on symplectic RK methods, and these, too, are surveyed by Enright et al. [12]. In these techniques, the aim is to preserve, in the numerical approximation, some quantity such as a discrete analogue of energy that should be constant in the exact system of differential equations. This should ensure that orbits remain elliptical, rather than spiralling into the Sun, due to a loss of (discrete) energy in the numerical approximation. An assessment of symplectic methods for Solar System dynamics is presented by Saha and Tremaine [32], and they conclude that these integrators are unquestionably the best available in this context; in this present paper, we will briefly consider the performance of symplectic methods as compared with ERK techniques.

Of course, techniques other than RK methods are also available for the integration of orbital equations. RK methods are single-step integrators, in that the solution at each time-step only depends on conditions at the previous step. Multi-step methods are also possible, however, and can have certain advantages in stability, although now values at each time-step depend on values at the previous k time-steps for a k-step method. Such techniques can be more onerous to code. A mathematical analysis of these methods is presented by Dahlquist [11], and a symmetric multi-step technique has been adapted for orbital systems by Quinlan and Tremaine [30]. More recently, Zeebe [36] has compared various methods, observing that a Bulirsch–Stoer extrapolative method required vastly more computational time to achieve the same accuracy as a symplectic technique. To improve numerical stability, implicit RK methods have also been developed, in particular, to allow the solution of stiff equations; such systems

can occur in orbital-type equations through resonances or relaxation oscillations. In general, however, implicit RK methods require a nonlinear system of algebraic equations to be solved at each time-step, and this could pose considerable difficulty. Consequently, diagonally implicit RK methods have been proposed to make their implementation easier, while nevertheless retaining some of the stability advantages. A review of these techniques has been given by Kennedy and Carpenter [24].

The long-term stability of the Solar System is a key question in understanding planetary dynamics and accurate, reliable large-time integrations of the governing equations are usually necessary. Laskar [26] reported numerical integrations of approximate, averaged equations of Solar System motion, from which he concluded that the Solar System is chaotic. Resonances in the motions of the inner planets made his numerical solutions unreliable at some time beyond 10 mega-years (Myr). Laskar et al. [27] confirmed these results by integrating the equations for the whole Solar System over a 6-Myr time interval, and showed the presence of resonances in the motions of the Earth and Mars. These findings are consistent with a subsequent integration of the Newtonian equations for the Sun and eight planets, carried out by Batygin and Laughlin [4]. Their results also showed an instability in the motion of Mercury, which eventually destabilizes the Solar System. Olsen et al. [29] likewise found that long-term numerical solutions were only reliable over an interval of 60 Myr, and that the system becomes chaotic. However, here we are simulating n-body problems to demonstrate accuracy of integrators and so are concerned only with numerical instabilities. We note that stellar system integrations normally use mixed-variable symplectic methods based on the work of Wisdom and Holman [35], which are more efficient in dealing with close encounters of two bodies. We will briefly consider a fourth-order symplectic method, but will not consider the more specialized methods developed for orbital mechanics.

Relativistic orbits may also be considered a type of n-body problem, requiring forward integration to simulate the orbits as they spiral inwards. The field of gravitational wave (GW) astronomy has progressed greatly since the detection of the first merging binary black hole (BBH) system in September 2015 [1]. The advanced Laser Interferometer Gravitational-wave Observatory (aLIGO) uses a matched-filtering detection algorithm to target a 4-dimensional parameter space describing the most common variants of compact binary sources, that is, those with spin-aligned components on quasi-circular orbits [17, 19, 31]. However, several studies have suggested this algorithm may miss compact binaries formed in dense stellar environments due to their more eccentric orbits [19–21, 25]. Constructing the template waveforms involves numerically solving the Einstein field equations to describe the inspiral, merger and ringdown. The large (9-dimensional) parameter space and computationally intensive nature of numerical relativity (NR) simulations has led to the development of accurate Keplerian-type parametric solutions including a post-Newtonian (PN) approximation [18]. The PN approximation invokes a gauge-invariant dimensionless parameter, x, related to the total mass of the binary system and the orbital frequency. The temporal evolution of this parameter is computed using a first-order differential equation that

considers the present state of the Keplerian-type orbit and the PN corrections [21, equation (5)]. The eccentricity, mean anomaly and phase of the orbit are typically computed with this gauge-invariant parameter as a coupled system of ordinary differential equations (ODEs). The waveform for the merger-ringdown, for example, derived using the IRS model introduced by Kelly et al. [23] and calibrated with NR simulations [21], can be appended to that of the inspiral phase.

Large systems of ODEs may also arise from the numerical solution of partial differential equations (PDEs) in fluid mechanics or combustion theory, for example. Any numerical technique that approximates the spatial derivatives in some manner results in a large system of time-dependent ODEs that must be integrated forward in time. RK methods are often employed for this purpose. There are three broad classes of techniques that are of this type. The first are methods such as finite-element or discontinuous Galerkin methods, in which spatial variations are approximated over collections of small elements of finite size. Examples of these approaches are given in the review by Cockburn and Shu [9]. A second type is related to the method of lines, in which finite differences are used to approximate only the spatial derivatives, so that the PDE is replaced by a system of ODEs at each spatial node point. Verwer [33] applies this approach to the solution of parabolic PDEs, which arise in diffusion processes, and a somewhat similar approach has been used recently by Zhao et al. [38] to solve systems of nonlinear wave equations in conservation form. The third group of techniques are spectral methods, in which the unknown function is expressed as a series of orthogonal functions in space, with time-dependent Fourier coefficients that solve systems of ODEs that must be integrated forward in time. Walters and Forbes [34] recently used methods of this type to study Rayleigh–Taylor instabilities in fluid mechanics, in three space variables and using RK methods for the time integration. We do not consider PDEs further in this paper, however, instead focus on ODE systems arising in orbital motions. Our primary interest is ERK methods, since these are the easiest to code, for large systems of ODEs, and so are the first choice in a wide variety of applications. We will consider two types of problems to be solved: n-body Newtonian systems and a two-body relativistic binary orbit. These are areas of current interest and we give a very brief introduction to these topics here.

This present paper is concerned with higher-order ERK methods, which will be examined in the solution of n-body systems, and their practical implementation. These methods are equally applicable to the solution of PDEs as described above, and Forbes et al. [15] employed a 10th-order, 16-stage ERK recently to study a problem in free-surface hydrodynamics.

In Section 2, we lay out the general structure of ERK methods and identify the particular methods used here. Section 3 presents the n-body equations and our numerical scheme to solve them. We provide modern Fortran code and show the results of tests using methods of various orders. The performance of these methods for Newtonian Solar System orbits is discussed, and in Section 3.4, they are also applied to a two-body post-Newtonian system radiating gravitational waves. A discussion concludes the paper in Section 4.

Throughout this study, we have compared results obtained using double-precision (64-bit floats) and quad-precision (128-bit floats). Many modern compilers allow this choice to be made by means of a single compiler switch, as was the case here, using the open-source compiler, GFORTRAN. We note, however, that calculations using quad-precision floats are implemented via an additional software layer and so perform slower by some orders of magnitude on standard desktop hardware.

2. Explicit Runge–Kutta methods

ERK methods have the property that function values at each new time-step depend only on values at the current time level. This makes these methods fast and particularly simple to code. For clarity, the ERK method is used to solve an initial value problem of the form

$$\frac{dy}{dt} = f(t, y(t)), \quad y(t_0) = y_0,$$

for some known function f, with initial value y_0 at known time t_0. The solution after a small time-step Δt is given by

$$t_{n+1} = t_n + \Delta t,$$

$$y_{n+1} = y_n + \Delta t \sum_{i=1}^{p} b_i k_i,$$

with

$$k_i = f\left(t_n + c_i \Delta t, y_n + \Delta t \sum_{j=1}^{i-1} a_{ij} k_j\right).$$

The coefficients a_{ij}, b_i, c_i are defined by the particular RK method chosen. The a_{ij} components are a set of coefficients in lower triangular matrix form. For a method of p stages, $i = j = p$, there are a total of $p(p-1)/2$ such coefficients, although some of these may be zero. There are p weights, b_j. The c_i components are simply the sum of the corresponding row of a_{ij} components, that is, $c_i = \sum_{j=1}^{i-1} a_{ij}$. Newtonian n-body problems may be classified as autonomous systems, that is, systems in which the derivative function, f, (in this case, acceleration) does not depend on the independent variable (time), but only on the dependent variables (position, velocity). For such systems, the intermediate function evaluations, k_i, take a slightly simpler form:

$$k_i = f\left(y_n + \Delta t \sum_{j=1}^{i-1} a_{ij} k_j\right),$$

and the c_i components are not required.

Each ERK method may be characterized by its "Butcher tableau", being a visual display of these coefficients used for each step of the ERK method. The number p of stages of an RK method is the number of function evaluations performed at every

$$
\begin{array}{c|ccccc}
0 \\
c_2 & a_{2,1} \\
\vdots & \vdots \\
c_i & a_{i,1} & a_{i,i-1} \\
\vdots & \vdots & & \vdots \\
c_p & a_{p,1} & \dots & & a_{p,p-1} \\
\hline
 & b_1 & \dots & & \dots & b_p
\end{array}
\qquad
\begin{array}{c|cccc}
0 \\
\frac{1}{2} & \frac{1}{2} \\
\frac{1}{2} & 0 & \frac{1}{2} \\
1 & 0 & 0 & 1 \\
\hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}
\end{array}
$$

$$
\begin{array}{c|ccccccc}
0 \\
\frac{1}{3} & \frac{1}{3} \\
\frac{2}{3} & 0 & \frac{2}{3} \\
\frac{1}{3} & \frac{1}{12} & \frac{1}{3} & -\frac{1}{12} \\
\frac{1}{2} & -\frac{1}{16} & \frac{9}{8} & -\frac{3}{16} & -\frac{3}{8} \\
\frac{1}{2} & 0 & \frac{9}{8} & -\frac{3}{8} & -\frac{3}{4} & \frac{1}{2} \\
1 & \frac{9}{44} & -\frac{9}{11} & \frac{63}{44} & \frac{18}{11} & 0 & -\frac{16}{11} \\
\hline
 & \frac{11}{120} & 0 & \frac{27}{40} & \frac{27}{40} & -\frac{4}{15} & -\frac{4}{15} & \frac{11}{120}
\end{array}
$$

FIGURE 1. The Butcher tableau is a standard way of displaying the coefficients for RK methods. For ERK methods, the a_{ij} coefficients are displayed in lower-triangular matrix form, with their row-sums, c_i, as a column on the left and the integration weights, b_j, as a row underneath. The general form for ERK methods is shown in the left panel, the "classic" RK4 method is in the centre panel and one of Butcher's RK6 methods is shown in the right panel.

time-step of the integration routine. The order of an RK method is a measure of the accuracy of the method, in terms of powers of the time-step, Δt. The "classic" fourth-order method has four stages and is fourth-order accurate in global truncation error (the numerical solution at the final time will vary from the true solution by a factor proportional to Δt^4). The number of stages required for a given order increases quickly for high-order methods, although the general formula defining the required number of stages for a given order is not known. A method is generally labelled by its order. Thus, the Butcher RK6 method is a sixth-order method, although it requires seven function evaluations (stages) per time-step. The Butcher tableau for a general ERK of p stages is presented in the left panel of Figure 1, and as specific examples, Runge and Kutta's "classic" RK4 method [3, p. 371] is presented in the middle panel and Butcher's RK6 method [6] (with his choices of $\lambda = 4, \mu = 1/3$, these being additional constraints that are chosen to complete the definition of a particular method) on the right.

In the current paper, we consider RK methods ranging from the classic RK4 method with four stages up to a 35 stage 14th-order method. With $p = 35$ stages, the highest-order method investigated here has $35 \times 34/2 = 595$ a_{ij} coefficients. Regardless of the number of stages and coefficients, the numerical algorithm is the same and will be described in Section 3, along with a small code sample.

Seven explicit RK methods are considered in this work, which will mostly be referenced by the number of stages they contain. These are listed by the number of stages and order of accuracy of each method in Table 1. We note that several of these methods can be expanded to embed methods of lower or higher order, which allow the use of adaptive stepsize control. We have not investigated the use of adaptive stepsize in the current work, although we note that such methods may be efficient, particularly

TABLE 1. The seven methods investigated in this work are listed below, showing number of stages and order of accuracy. The number of stages increases rapidly with increasing order of accuracy.

Stages	Order	Source
4	4	Atkinson [3]
7	6	Butcher [6]
11	8	Cooper and Verner [10]
16	10	Zhang [37]
17	10	Feagin [13]
25	12	Feagin [14]
35	14	Feagin [14]

for high ellipticity orbits where the accelerations and velocities vary greatly throughout the orbital path.

3. The n-body simulations

A system of three or more bodies interacting under Newtonian gravity has motion of the bodies which is not generally solvable analytically. However, given knowledge of the masses, positions and velocities of all of the bodies ("initial conditions"), these systems may be integrated forward in time to determine positions and velocities at a future time. Such a simulation of Newtonian n-body problems requires the solution of the gravitational equation

$$\ddot{\mathbf{r}}_{\mathbf{i}} = -\sum_{j=1}^{n} \frac{m_j}{r_{ij}^3} \mathbf{r}_{\mathbf{ij}} : \mathbf{i} \neq \mathbf{j},$$

where $\mathbf{r}_{\mathbf{ij}}$ is the position vector from the jth body to the ith body, r_{ij} is the corresponding distance, m_j is the mass of the jth body and $\ddot{\mathbf{r}}_{\mathbf{i}}$ is the acceleration of the ith body, and units have been chosen such that Newton's constant has value 1. The components in Cartesian co-ordinates have equations:

$$\ddot{x}_i = -\sum_{j=1}^{n} \frac{m_j}{r_{ij}^3}(x_i - x_j) : i \neq j,$$

$$\ddot{y}_i = -\sum_{j=1}^{n} \frac{m_j}{r_{ij}^3}(y_i - y_j) : i \neq j,$$

$$\ddot{z}_i = -\sum_{j=1}^{n} \frac{m_j}{r_{ij}^3}(z_i - z_j) : i \neq j,$$

$$r_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}.$$
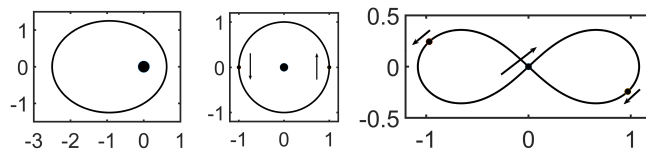
FIGURE 2. The orbital paths of the two- and three-body simulations, showing the initial locations of the bodies, along with an indication of their initial velocities, represented by the arrows. The left panel shows the simplified two-body orbit, where the mass is all located at the origin and a much lighter body orbits in an elliptical path. The ellipse shown here has eccentricity of 0.6. The central panel shows the orbits of two massive bodies orbiting a more massive central body. The right panel shows the path of three equal mass bodies chasing each other around a figure-eight path. Initially, the central body has twice the speed of the other two bodies, but in the opposite direction, so that the system has zero total momentum.

The RK methods integrate first-order equations, so the three second-order Newtonian equations are re-written as six first-order equations:

$$\dot{x}_i = v_i^x, \quad \dot{y}_i = v_i^y, \quad \dot{z}_i = v_i^z,$$

$$\dot{v}_i^x = -\sum_{j=1}^{n} \frac{m_j}{r_{ij}^3}(x_i - x_j) : i \neq j,$$

$$\dot{v}_i^y = -\sum_{j=1}^{n} \frac{m_j}{r_{ij}^3}(y_i - y_j) : i \neq j,$$

$$\dot{v}_i^z = -\sum_{j=1}^{n} \frac{m_j}{r_{ij}^3}(z_i - z_j) : i \neq j,$$

$$r_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2},$$

where v_i^x, v_i^y and v_i^z are the Cartesian velocity components of the ith body. For a system containing n bodies, this is now a coupled system of $6n$ first-order differential equations to be integrated forward in time. Here we first consider some simple two- and three-body systems. These include elliptical orbits and two simple three-body systems, with orbital paths shown in Figure 2, and as described below. These are followed by a nine-body Solar System model, and a relativistic binary black hole inspiral.

We found it expedient to load the a_{ij} and b_j values from text files at the start of the program. The explicit RK calculation is then written in a general way that works correctly with any number of stages. For illustration, the RK calculation to move the system forward by the small time-step Δt is shown in Figure 3, in modern Fortran, for an RK method with number of stages equal to *rksteps*. The positions and velocities of all of the bodies in the system are stored in vectors x, y, z, vx, vy, vz. For example, $x(1)$ is the x-value of the first body, and $x(n)$ is the x-value of the nth body. The "accel" code is a simple routine that calculates the Newtonian acceleration of each body due to all the other bodies in the system. This routine takes in the arrays containing the position data for all the bodies in the system (in this case, xi, yi, zi) and their corresponding masses

```
!call the steps of the rk routine
do j=1,rksteps
  xi=x
  yi=y
  zi=z
  xk(:,j)=vx
  yk(:,j)=vy
  zk(:,j)=vz
  !store the incremental updates
  do k=1,j-1
    xi=xi+xk(:,k)*dt*a(j,k)
    yi=yi+yk(:,k)*dt*a(j,k)
    zi=zi+zk(:,k)*dt*a(j,k)
    xk(:,j)=xk(:,j)+vxk(:,k)*dt*a(j,k)
    yk(:,j)=yk(:,j)+vyk(:,k)*dt*a(j,k)
    zk(:,j)=zk(:,j)+vzk(:,k)*dt*a(j,k)
  enddo
  !calculate the acceleration of each body due to the other bodies
  call accel(masses,xi,yi,zi,nmasses,vxk(:,j),vyk(:,j),vzk(:,j))
enddo
!update the values for time t+dt
vx=vx+Matmul(vxk,b)*dt
vy=vy+Matmul(vyk,b)*dt
vz=vz+Matmul(vzk,b)*dt
x=x+Matmul(xk,b)*dt
y=y+Matmul(yk,b)*dt
z=z+Matmul(zk,b)*dt
t=t+dt
```

FIGURE 3. The n-body 3D code to utilize any of the ERK methods, is shown here in modern Fortran. The initial values are loaded into the variables xi etc. Next, the intermediate integration point is determined, based on the a_{ij} values. The accelerations at that point are determined, based on the masses and positions of all the other bodies. Finally, the positions and velocities are updated based on all the integrated values at each stage, weighted by the b_j weights.

(*masses*) and the number of bodies in the system (*n masses*), and returns the component accelerations (*vxk, vyk, vzk*, that is, the derivatives of the velocity components). These acceleration data are stored in *vxk, vyk, vzk* for each of the stages of the RK routine, and for each body. "Matmul" is the intrinsic Fortran routine for matrix multiplication. The time values after each step of the integration process are stored in the variable t.

For clarity, the "accel" routine for the n-body simulations is shown in Figure 4, apart from the variable definitions. The term $r2$ is the square of the distance between two bodies, and $r3$ is the inverse of the cube of that distance.

The accuracy of any forward-integration scheme, such as the one shown in Figure 3, is dependent on the smallness of the time-step Δt. The dependence of accuracy on the stepsize is illustrated in Figure 5. For this illustration, a circular orbit has been numerically integrated using Zhang's 10th-order 16-stage method [37] for 1000 orbits. The position of the satellite in the final orbit is compared against its analytically determined position. The maximum discrepancy in position in the final orbit is plotted against the stepsize, using $\log_{10}$ scales. This line has a slope of 9.82, demonstrating that the method is close to 10th-order accurate (that is, $\Delta err \propto \Delta t^{9.82}$).

```fortran
subroutine accel(masses,x,y,z,nmasses,ax,ay,az)
!calculate the acceleration on each body, due to the other bodies in the system
  do m=1,nmasses
    ax(m)=0.
    ay(m)=0.
    az(m)=0.
    do j=1,nmasses
      if (m==j) then
        cycle
      else
        !avoid using power function for inverse of r cubed, as it is slow
        r2=(x(j)-x(m))**2+(y(j)-y(m))**2+(z(j)-z(m))**2
        r3=1./(sqrt(r2)*r2)
        ax(m)=ax(m)-masses(j)*(x(m)-x(j))*r3
        ay(m)=ay(m)-masses(j)*(y(m)-y(j))*r3
        az(m)=az(m)-masses(j)*(z(m)-z(j))*r3
      endif
    enddo
  enddo
```

FIGURE 4. The code snippet used to update the position of each body. The "cycle" statement skips the calculation where $m = j$. That is, the body's acceleration is not affected by its own position and mass.
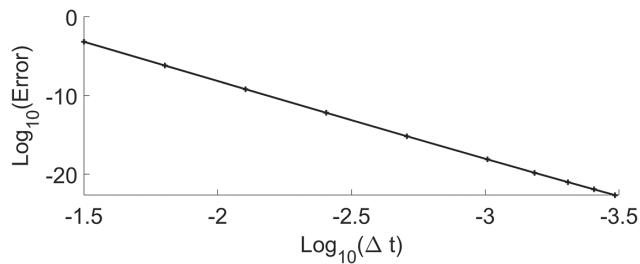
FIGURE 5. Display of error level due to different stepsize Δt. Both the error and Δt are displayed using log scales, which appears as a linear relation, demonstrating the power law dependence of accuracy on Δt. This graph compares results at different step-sizes for a simple circular orbit using quad-precision (128-bit floats) integrated by the 16-stage 10th-order method. The error is relative to the analytically determined position for the final orbit after 1000 orbits.

In comparing the various RK methods, the computational time is primarily dependent on the number of function evaluations, being the number of time-steps multiplied by the number of RK stages per time-step. For the remainder of this study, the error will be plotted against the number of function calls.

3.1. Kepler orbits   The best known orbital system is the two-body system, acting under Newtonian gravity. The two bodies follow paths that are conic sections. We consider here a bound orbit (an ellipse), and also further simplify the problem by considering a body at the origin, and an orbiting body of comparably small mass. In this case, the solution consists of no motion of the body at the origin, while the orbiting body follows an elliptical path, with one focus at the origin.

The test system is set up with a body with dimensionless mass 1 at the origin. A particle of negligible mass orbits in an elliptical path with specific angular momentum $L = 1$, eccentricity ϵ, semi-major axis $a = L^2/(1 - \epsilon^2)$. Such an orbit has period $T = 2\pi a^{3/2}$. At time $t = 0$, we set

$$(x, y, z) = (L^2/(1 + \epsilon), 0, 0) \quad \text{and} \quad (\dot{x}, \dot{y}, \dot{z}) = (0, (1 + \epsilon)/L, 0).$$

These initial conditions describe an ellipse in the x–y plane.

To compare the seven RK methods under consideration, the elliptical system is integrated forward in time for 100 orbits. The final orbit is compared with the analytical solution at each time-step in the final orbital period. Specifically, the relative error is defined as $Err = ((x_{rk} - x_{an})^2 + (y_{rk} - y_{an})^2)/r_{an}^2$, where x_{an} and x_{rk} are the x-values of position computed with the analytical method and the RK method, respectively, and similarly for y and r. The largest value of this error in the final orbit is plotted for each method for various numbers of evaluation steps per orbit, in Figure 6. The upper panels show the results using double precision (64-bit floats), and the lower panels show the results using quad-precision (128-bit floats).

For the high eccentricity cases, the stepsize has been significantly reduced to handle the rapid motion of the orbiting body whilst near the origin. It may be seen in Figure 6 that the runtime has been increased by a factor of approximately 30 in the $\epsilon = 0.9$ case over the $\epsilon = 0.3$ case. It is clear that the efficiency in high eccentricity cases would be increased by the use of an adaptive stepsize method, but this has not been pursued here.

Figure 6 shows the error, on a logarithmic scale, versus the number of function evaluations per orbital period, also on a logarithmic scale. The number of function evaluations is therefore the number of stages of the method multiplied by the number of time-steps per orbit. Thus, at the same number of function evaluations, Zhang's 10th-order 16 stage method has a time-step that is four times as large as that of the four-stage RK4 method. It may be observed that, given a sufficiently small time-step, the higher-order integrators consistently reach the same level of precision for lower computational cost. This is particularly so with the quad-precision models. However, caution should be taken when using large stepsizes, where the 12th- and 14th-order integrators, in particular, may perform very badly. It is also apparent that there is a minimum useful stepsize for each method, beyond which there are no further improvements in accuracy. This is seen in the "levelling off" in the double-precision graphs. Once the accuracy of the method is close to the limit of the machine-precision, reducing the stepsize will not help. In fact, it may be seen in Figure 6 that decreasing the stepsize beyond this point, resulting in more function evaluations, actually decreases accuracy slightly due to greater accumulation of "rounding errors".

These results suggest that high-accuracy double-precision calculations will be performed accurately with the 10th- and 12th-order integrators, and the classic RK4 is
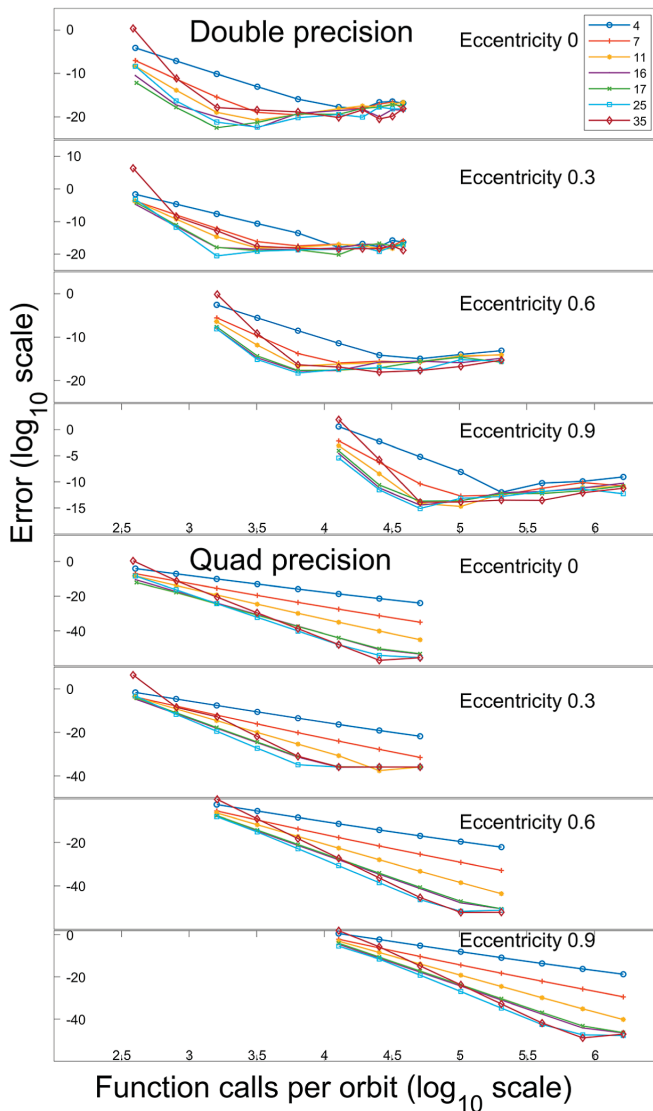
FIGURE 6. Maximum variation of position for each of the RK methods for elliptical orbits. The eccentricity varies from 0 to 0.9. The legend shows the different RK methods by their number of stages. The x-axis shows the number of function evaluations per orbit. The y-axis shows the variation relative to the analytic value for the final orbit. Both axes use $\log_{10}$ scaling. The upper figures show results using double precision (64-bit floats), the lower using quad-precision (128-bit floats).

much slower. In the case of quadruple-precision, the 12th-order method is excellent, and may be a safer method than the 14th-order, unless very small time-steps are used.

A note on symplectic methods: Symplectic methods are suited to conservative systems such as the Newtonian systems we are considering. While a detailed investigation
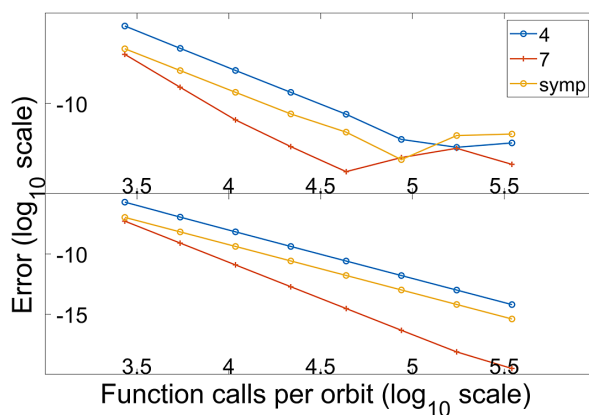
FIGURE 7. Comparison of the fourth-order symplectic method with the four- and seven-stage RK methods. The x-axis shows the total number of function evaluations for an elliptical system with eccentricity of 0.6. The y-axis shows the variation relative to the analytic value for the orbit. Both axes use log_{10} scaling. The upper panel shows results using 64-bit floats, the lower using 128-bit floats.

of these methods is beyond the scope of this paper, we did perform some simulations of the Kepler orbit with a fourth-order symplectic integrator (the optimized method of [28]) for an elliptical orbit with eccentricity 0.6. The symplectic method is compared with the RK4 and RK6 methods in Figure 7. These three methods were evaluated with respect to the analytically determined elliptical orbit. In every case, the fourth-order symplectic integrator was more efficient than the classic RK4 method, but less efficient than the RK6 method (which is itself usually less efficient than even higher-order methods, as seen in Figure 6). This suggests that at a given order, n-body problems are solved more efficiently with an optimized symplectic integrator, if available. However, for high positional accuracy, the order of the method is more important than whether the method is symplectic or not.

3.2. Three-body orbits While there is no solution to the general Newtonian three-body problem, many solutions to particular three-body systems are known. We first consider here a simple three-body system with central body at the origin with dimensionless mass 1, orbited by two satellites of mass 0.1 situated on opposite sides of the origin, at radius 1. These satellites have initial speed $\sqrt{1 + 0.1/4}$, as required for a circular orbit. The accuracies of the RK methods against the exact circular orbit are shown in Figure 8, in double- and quad-precision. Even in the double-precision case, it may be seen that the fourth-order method takes approximately 50 times as long to reach the 10^{-10} accuracy level as the 10th- and 12th-order methods, so the savings are not insignificant. For very high accuracy, the 12th- and 14th-order methods clearly perform better than the lower-order methods in this problem.

Another stable three-body system is the remarkable figure-eight system with three equal masses and zero total momentum [8]. The three bodies are initially placed
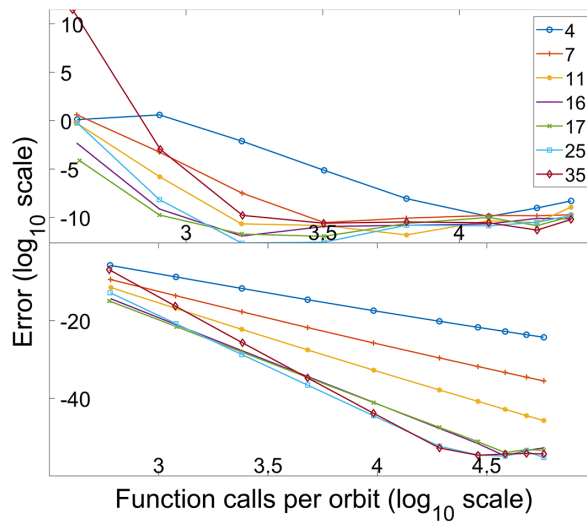
FIGURE 8. Maximum variation of position for each of the RK methods for the circular three-body problem. The legend shows the different RK methods by their number of stages. The x-axis shows the number of function evaluations per orbit. The y-axis shows the variation relative to the analytic value for the final orbit. Both axes use $\log_{10}$ scaling. The upper panel shows results using 64-bit floats, the lower using 128-bit floats.

co-linearly, as shown in the right panel in Figure 2. The body at the origin has twice the velocity of the other two bodies. This system has a region of stable orbits. The accuracy of the integration methods is tracked by variations in the total momentum of the system (which should remain zero). The accuracy of the RK methods is shown in Figure 9. As with the circular three-body system, the 12th- and 14th-order methods clearly out-perform the other methods in the high-accuracy region, but perform poorly when used with large stepsizes, particularly the 14th-order method. In the double-precision case, the fourth-order method is very slow, and even the sixth-order method is significantly slower than the higher-order methods.

3.3. Solar System orbits We now consider the orbits of the nine major bodies in our Solar System (the sun and the eight planets). Initial position and velocity data were obtained from the JPL web interface [22]. The co-ordinate system was then adjusted to place the origin at the barycentre, and the system integrated forward in time. To measure the accuracy of the RK integrators, the z-component of the initial angular momentum was compared with the angular momentum at each time-step in the final (Earth) year of the simulation, and the largest relative error displayed against each RK method for a various number of function evaluations per (Earth) year. For context, the variation of the z-component of momentum for each of the bodies is shown in Figure 10, smoothed for clarity using a 25-year moving average, to remove short-timescale, high-amplitude variabilities. It may be seen that while the
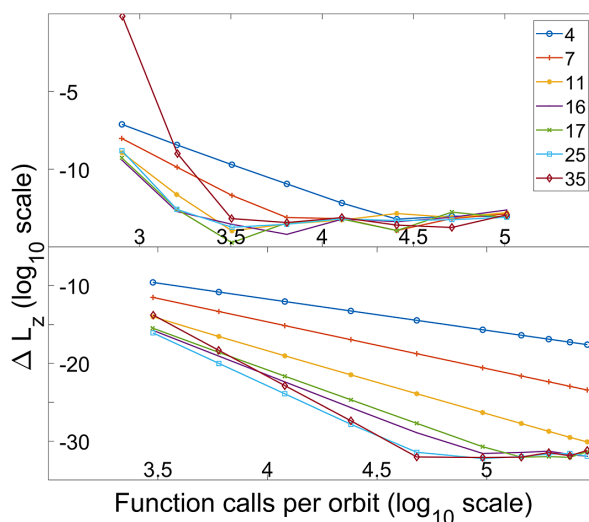
FIGURE 9. Final maximum absolute value of the variation of angular momentum for each of the RK methods for the figure-eight three-body problem. The legend shows the different RK methods by their number of stages. The x-axis shows the number of function evaluations per orbit. The y-axis shows the maximum absolute value of the total angular momentum for the final orbit. Both axes use log_{10} scaling. The upper panel shows results using 64-bit floats, the lower using 128-bit floats.
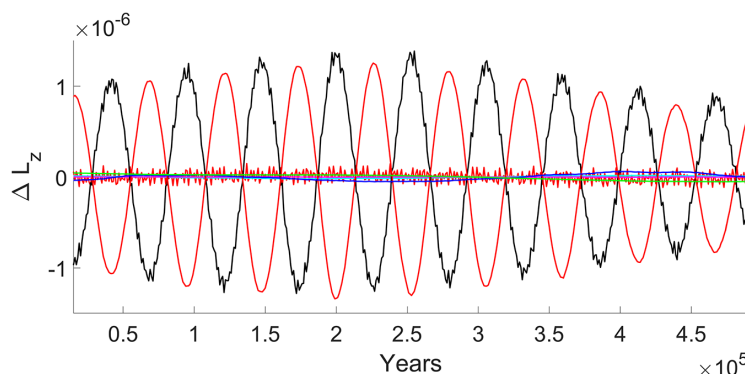
FIGURE 10. Variation in the z-component of angular momentum for each of the nine Solar System bodies over 500 000 years, smoothed with a 25-year moving average. The variation is dominated by Jupiter (black) and Saturn (red online), exchanging momenta over a 50 000-year period. The total momentum of the system should remain zero.

total momentum of the system should remain constant, there is significant transfer of momenta between the bodies. This transfer is dominated by Jupiter and Saturn, with a period of just over 50 000 years. There is also a longer envelope visible with a period of approximately 600 000 years. However, for the main purpose of this paper, the concern is with accuracy, which in this case may be assessed by considering the change in total momentum. Total momentum should be conserved, so the best integration
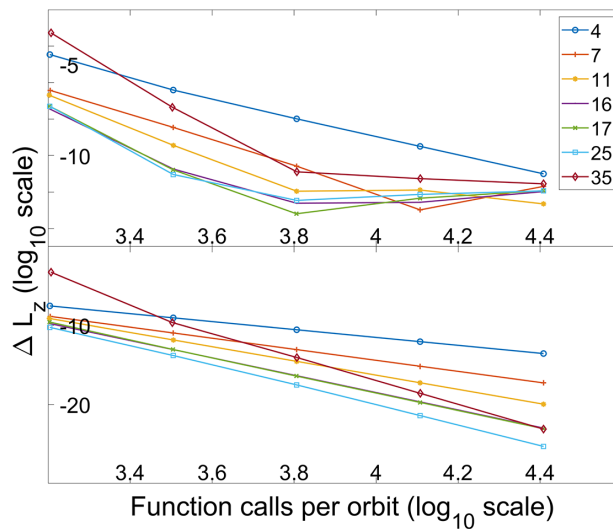
FIGURE 11. Final maximum absolute value of the variation in the total system z-component of angular momentum for each of the RK methods for the nine-body Solar System. The legend shows the different RK methods by their number of stages. The x-axis shows the number of function evaluations per orbit. The y-axis shows the maximum absolute value of the total angular momentum for the final orbit. Both axes utilize $\log_{10}$ scaling. The upper panel shows results using double precision (64-bit), the lower using quad-precision (128-bit).

methods have the lowest variation. A comparison of the seven RK methods is shown in Figure 11, considering the total change in the z-component of momentum after a period of time. The double-precision model (upper panel) was simulated over a period of 500 000 earth years. Calculations using quad-precision numbers are not directly supported in hardware, resulting in much longer runtimes. To perform comparisons in a reasonable time, the quad-precision runs were conducted over a period of 1000 simulated earth years.

The comparison of the RK methods here shows similarities with the two- and three-body systems discussed previously. The higher-order methods generally perform better, reaching higher accuracy for the same computational effort, given a suitably small time-step. Again, caution is advised when using the highest-order method as it may perform worse than the other methods for large time-steps. The 10th- and 12th-order methods perform well consistently, with the 12th-order method clearly outperforming lower-order methods in the quad-precision simulation. The 14th-order method does not appear to be a good choice in the double precision runs, but may outperform all lower-order methods in the quad-precision simulation, given a small enough time-step.

3.3.1. Long-time simulations Over a long period of time, it was found that the simulation fails at some point. Presumably, this is due to accumulation of errors
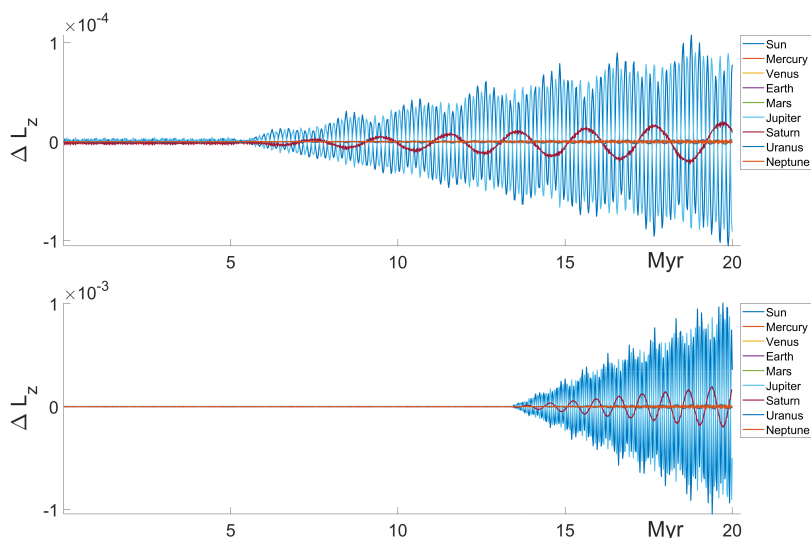
FIGURE 12. Variation in z-component of angular momentum for the nine-body Solar System. The upper panel show the results using 85 time-steps per earth year, which become unstable at 4.9 million years. The lower panel used 90 time-steps per year and became unstable at 13.4 million years.

associated with the Runge–Kutta approximation (also known as "truncation errors"). This is demonstrated in Figure 12 by the diverging of the extrema of the angular momenta of the orbiting bodies, occurring at approximately 4.9 million years for a simulation using 85 time-steps per earth year, and at 13.4 million years when using 90. Up until this point, the individual angular momenta show a pattern similar to that seen in Figure 10, oscillating back and forth regularly, but staying between extrema which do not vary. The point in time at which this divergence occurs is strongly dependent on the stepsize (Δt) used. Feagin's 17-step RK10 method was used for these long-term Solar System simulations. In Figure 13, we show a simulation using 95 time-steps per year (dashed lines), laid over the 90 time-steps simulation (dotted lines), and zoomed in to the time when the divergence occurs. We see that the two simulations are visually indistinguishable until approximately 13.38 million years, at which time the coarser simulation (larger Δt) diverges rapidly. In the lower panel, we have added the results from use of the fourth-order symplectic method, using 80 time-steps per year. This method maintains angular momenta terms as well as Feagin's RK10 method using 95 time-steps per year. As noted earlier, this is the design strength of symplectic methods, but they do not match the accuracy of the higher-order RK methods. That is, symplectic methods maintain the conserved quantities, such as energy and momentum, but this does not guarantee that the bodies will be in the correct position at each time, as was demonstrated by comparison to an analytical model in Figure 7.

The time to failure ($\log_{10}$ scale) is plotted against the number of points per year in Figure 14. This shows a strong power-law correlation between the number of time-steps
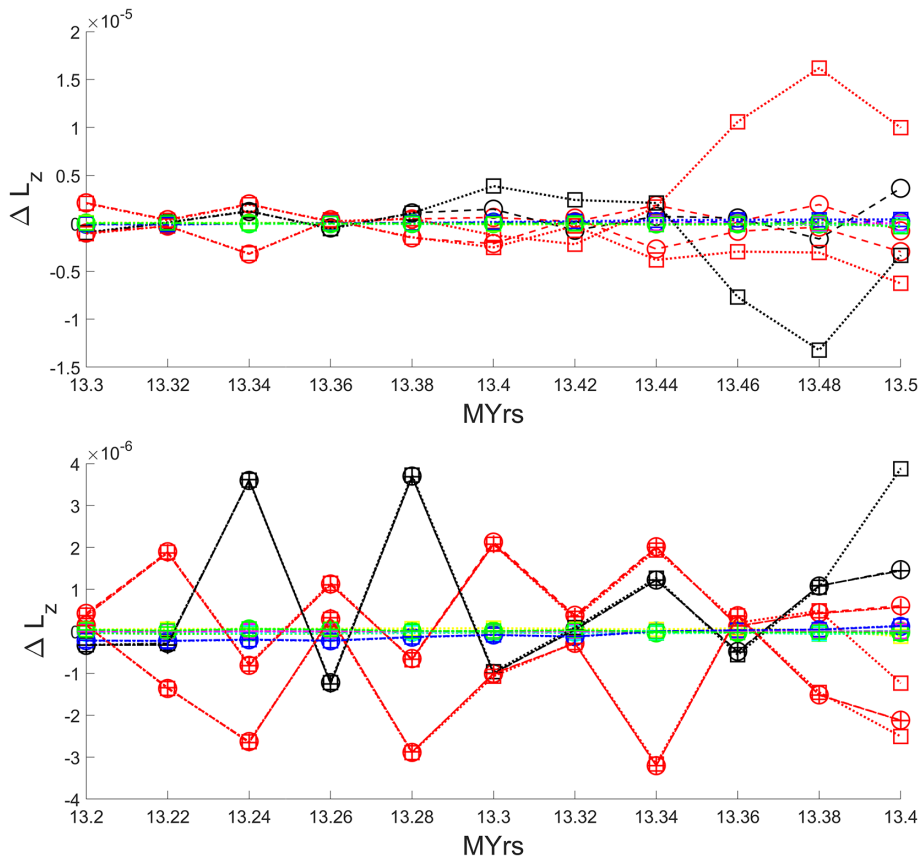
FIGURE 13. Comparison of the variation in the *z*-component of angular momentum for the nine-body Solar System using Feagin's RK10 method. In the upper panel, we have used dotted lines (with squares) and dashed lines (with circles) for 90 or 95 points per year, respectively. It may be seen that the two methods diverge at 1.338×10^7 years. A fourth-order symplectic method has been added in the lower panel, shown with a dash-dot line, with "+" markers. This symplectic simulation used 80 points per year, with four function calls per point. Additional magnification is used in the lower panel. By observing that the "+" and circle markers stay together, it is apparent that the symplectic method maintains the angular momenta terms better than a much slower (90 points per year) RK method.

used and the time at which numerical failure occurs. However several points did not fit this model well, with some failing significantly earlier than expected. We assume that this is due to a particular value of time-step causing a function evaluation at a time when there is higher than normal accelerations, such as a conjunction of planets. This causes a larger truncation error which slowly builds, eventually causing failure of the integration procedure. Perhaps this is an example of a slow-burning numerical instability such as that mentioned in relation to the DuFort–Frankel averaging method by Galloway and Ivers [16]. While caution is indicated, the simulation time is clearly significantly extended by increasing the number of time-steps. The graph in Figure 14
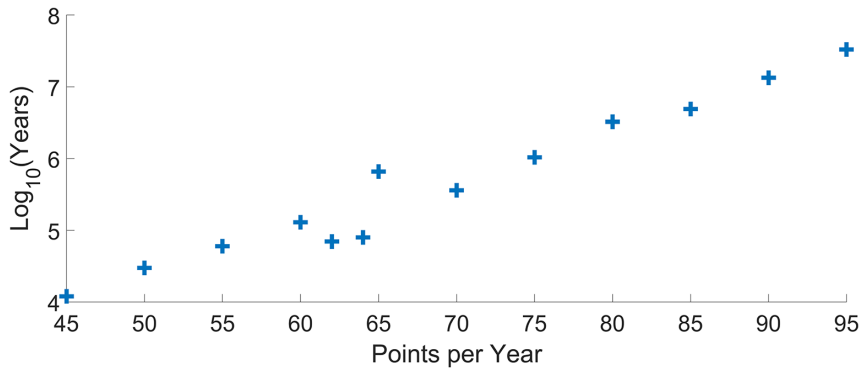
FIGURE 14. The number of years until numerical failure of the Solar System model, plotted using a $\log_{10}$ scale, against the number of points per year used in the simulation. While the graph has a strong linear correlation, there is significant variability, as seen in the data at 62, 64 and 65 points per year.

suggests that 120 points per year would be sufficient to allow a simulation up to one billion years. It is possible that adaptive time-stepping methods would avoid this instability, as they do not accept truncation errors larger than a pre-defined value.

3.4. Gravitational wave emissions A simple two-body relativistic system, a pseudo-circular double black hole binary, is considered in Figure 15. The positions of the black holes and the gravitational wave components are computed using the post-Newtonian equations given by Boyle et al. [5], which are a General Relativistic approximation, valid to 3.5 post-Newtonian order. These are equivalent to the more general elliptical orbit equations of Hinder et al. [18], but with equal masses and zero eccentricity. These equations are a good approximation for the inspiral until close to the merger.

Using the geometrized units $c = G = 1$, the quantities to be integrated are

$$
\dot{x} = \frac{16x^5}{5m}\left[1 - \frac{487}{168}x + 4\pi x^{3/2} + \frac{274229}{72576}x^2 - \frac{254\pi}{21}x^{5/2}\right.
$$
$$
\left. + \left(\frac{178384023737}{3353011200} + \frac{1475\pi^2}{192} - \frac{1712\gamma}{105} - \frac{856}{105}\ln(16x)\right)x^3 + \frac{3310\pi}{189}x^{7/2}\right],
$$
$$
\dot{\phi} = \frac{x^{3/2}}{m},
$$

where x is related to the orbital radius r by

$$
r = m\left[\frac{1}{x} - \frac{11}{12} + \frac{43}{36}x + \frac{38745\pi^2 - 1886084}{725760}x^2\right].
$$

The gravitational wave displacement for the h_+ polarisation at a distance R from the binary is calculated as
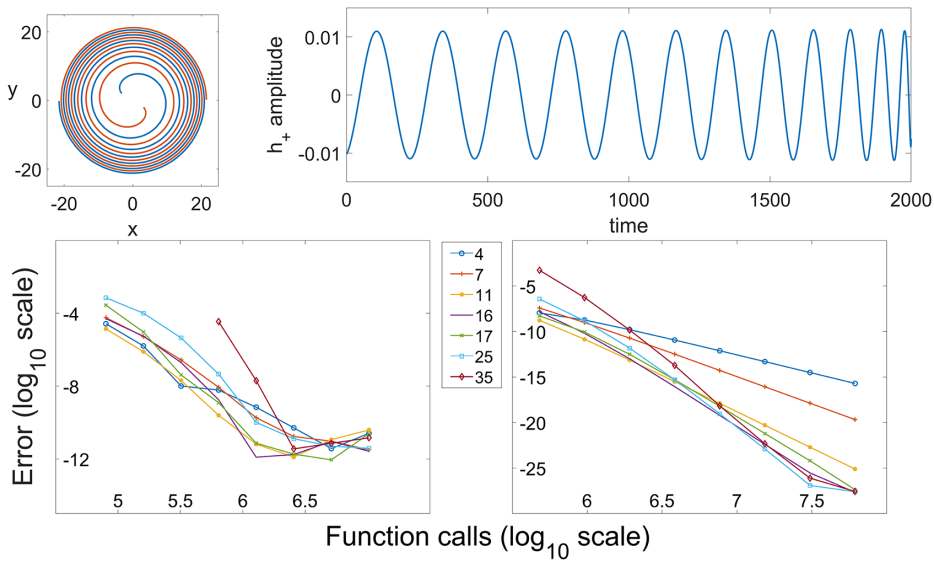
FIGURE 15. Accuracy of the Binary Black Hole simulation. The top left panel shows the orbital paths of the two equal mass black holes. The top right panel shows the value over time of the gravitational wave component h_+. The bottom panels compare the accuracy of the RK methods using double-precision (left) and quad-precision (right). The final value of h_+ in each simulation is compared against the value of h_+ calculated using high precision and very small time-steps. Both axes use $\log_{10}$ scaling.

$$h_+ = \frac{1}{2R}[\cos 2\phi(-2 + 17/4x - 4\pi x^{3/2} + 15917/2880x^2 + 9\pi x^{5/2})$$
$$+ \sin 2\phi(-x^{3/2}[12\ln(x) - 22/3 + 8\gamma] + x^{5/2}[27\ln(x) - 47/10 + 18\gamma])]].$$

The orbital phase of one of the bodies is ϕ, and of the other, $\phi + \pi$, while m is the mass of each of the two bodies and γ is Euler's constant. The constant R is the distance to the observer, which for the purposes of this simulation has been arbitrarily set to 90. In Figure 15, the inspiral of the two bodies is shown, along with the value of the gravitational wave component h_+ against time. These results are qualitatively consistent with those of Huerta et al. [21]. As we are interested only in how well the different RK methods perform, we have chosen arbitrary values for the mass and initial separation. The value of h_+ converges to a certain value as the accuracy of the integration procedure improves, so the simulation was initially run at very high accuracy, and that value of h_+ was used to evaluate the lower accuracy runs. The lower panels of Figure 15 show the performance of the RK methods against number of function evaluations. The required precision in gravitational wave detection events clearly warrants a high accuracy method. In achieving this high accuracy in the quad precision case, the higher-order methods are seen to be significantly faster than the lower-order methods. Optimized computational efficiency for the desired precision improves methods describing the inspiral phase of black hole binary

systems, which may aid in the detection of a greater breadth of GW events using LIGO.

4. Conclusions

While the particular choice of ERK method depends on the problem, these investigations suggest that the classic RK4 method is rarely a good choice when both efficiency and accuracy are important, at least for the types of systems considered here. The 10th- and 12th-order methods consistently out-perform lower-order methods, at both double and quadruple precision. The 14th-order method appears to be on course to exceed all lower methods if very high accuracy is desired, but caution should be used, as it may under-perform all of the methods used here for large stepsizes, as may be seen in most of the figures. We note that in a very different setting, fluid flow over a bump [15], we tested these ERK integrators, and found at least an order of magnitude decrease in run-time by choosing Zhang's 10th-order method [37] over the classic RK4 method, for the same level of accuracy. This suggests that modern higher-order integrators may be worth considering as an alternative to the classic RK4 method in a wide range of problems.

Together with hardware improvements and GPU acceleration, these new integrators have allowed a great improvement in accuracy of numerical integration over the last ten to twenty years. Problems that could not be simulated accurately in a reasonable amount of time, or without more sophisticated integration techniques, may now be solved by researchers with simple code running on modest hardware. Additionally, these explicit methods are easily implemented on graphics hardware for very parallel data applications, such as n-body systems with large numbers of bodies or for fluid flow problems.

The more general conclusion is that in many cases, it may be worth testing a range of integration methods to determine those well suited to the problem at hand, before committing to a large computational integrative task. The standard format of the coefficients of the ERK methods (the Butcher tableau) makes it particularly easy to write a routine to test ERK methods on a sample problem, similar to the large computational task, to determine an optimal method. Savings in computational time can be significant.

References

[1] B. P. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration), "Observation of gravitational waves from a binary black hole merger", *Phys. Rev. Lett.* **116** (2016) Article Id 061102; doi:10.1103/PhysRevLett.116.061102.

[2] Z. A. Anastassi and T. E. Simos, "An optimized Runge–Kutta method for the solution of orbital problems", *J. Comput. Appl. Math.* **175** (2005) 1–9; doi:10.1016/j.cam.2004.06.004.

[3] K. E. Atkinson, *An introduction to numerical analysis*, 1st edn (John Wiley & Sons, New York, 1978).

[4] K. Batygin and G. Laughlin, "On the dynamical stability of the Solar System", *Astrophys. J.* **683** (2008) 1207–1216; doi:10.1086/589232.

[5]   M. Boyle, D. A. Brown, L. E. Kidder, A. H. Mroué, H. P. Pfeiffer, M. A. Scheel, G. B. Cook and S. A. Teukolsky, "High-accuracy comparison of numerical relativity simulations with post-Newtonian expansions", *Phys. Rev. D* **76** (2007) Article Id 124038; doi:10.1103/PhysRevD.76.124038.

[6]   J. C. Butcher, "On Runge–Kutta processes of high order", *J. Aust. Math. Soc.* **4** (1963) 179–194; doi:10.1017/S1446788700023387.

[7]   J. C. Butcher, "A history of Runge–Kutta methods", *Appl. Numer. Math.* **20** (1996) 247–260; doi:10.1016/0168-9274(95)00108-5.

[8]   A. Chenciner and R. Montgomery, "A remarkable periodic solution of the three-body problem in the case of equal masses", *Ann. of Math. (2)* **152** (2000) 881–901; doi:10.2307/2661357.

[9]   B. Cockburn and C.-W. Shu, "Runge–Kutta discontinuous Galerkin methods for convection-dominated problems", *J. Sci. Comput.* **16** (2001) 173–261; doi:10.1023/A:1012873910884.

[10] G. J. Cooper and J. H. Verner, "Some explicit Runge–Kutta methods of high order", *SIAM J. Numer. Anal.* **9** (1972) 389–405; doi:10.1137/0709037.

[11] G. G. Dahlquist, "A special stability problem for linear multistep methods", *BIT* **3** (1963) 27–43; doi:10.1007/BF01963532.

[12] W. H. Enright, D. J. Higham, B. Owren and P. W. Sharp, "A survey of the explicit Runge–Kutta method", Technical Report TR 291/94, Department of Computer Science, University of Toronto, 1994. 1–35.

[13] T. Feagin, "A tenth-order Runge–Kutta method with error estimate", in: *Proceedings of the IAENG conference on scientific computing* (Hong Kong, March, 2007); https://hdl.handle.net/10657.1/2301.

[14] T. Feagin, "High-order explicit Runge–Kutta methods using *m*-symmetry", *Neural, Parallel Sci. Comput.* **20** (2012) 437–458; https://hdl.handle.net/10657.1/2296.

[15] L. K. Forbes, S. J. Walters and G. C. Hocking, "Ideal planar fluid flow over a submerged obstacle: review and extension", *ANZIAM J.* **63** (2021) 377–419; doi:10.1017/S1446181121000341.

[16] D. Galloway and D. Ivers, "Slow-burning instabilities of Dufort–Frankel finite differencing", *ANZIAM J.* **63** (2021) 23–28; doi:10.1017/S1446181121000043.

[17] D. Gerosa, M. Kesden, E. Berti, R. O'Shaughnessy and U. Sperhake, "Resonant-plane locking and spin alignment in stellar-mass black-hole binaries: a diagnostic of compact-binary formation", *Phys. Rev. D* **87** (2013) Article Id 104028; doi:10.1103/PhysRevD.87.104028.

[18] I. Hinder, F. Herrmann, P. Laguna and D. Shoemaker, "Comparisons of eccentric binary black hole simulations with post-Newtonian models", *Phys. Rev. D* **82** (2010) Article Id 024033; doi:10.1103/PhysRevD.82.024033.

[19] E. A. Huerta and D. A. Brown, "Effect of eccentricity on binary neutron star searches in advanced LIGO", *Phys. Rev. D* **87** (2013) Article Id 127501; doi:10.1103/PhysRevD.87.127501.

[20] E. A. Huerta, P. Kumar, S. T. McWilliams, R. O'Shaughnessy and N. Yunes, "Accurate and efficient waveforms for compact binaries on eccentric orbits", *Phys. Rev. D* **90** (2014) Article Id 084016; doi:10.1103/PhysRevD.90.084016.

[21] E. A. Huerta et al., "Complete waveform model for compact binaries on eccentric orbits", *Phys. Rev. D* **95** (2017) Article Id 024038; doi:10.1103/PhysRevD.95.024038.

[22] JPL Horizons web interface (Jet Propulsion Laboratory, California Institute of Technology, CA) accessed August 2021, https://ssd.jpl.nasa.gov/horizons.cgi.

[23] B. J. Kelly, J. G. Baker, W. D. Boggs, S. T. McWilliams and J. Centrella, "Mergers of black-hole binaries with aligned spins: waveform characteristics", *Phys. Rev. D* **84** (2011) Article Id 084009; doi:10.1103/PhysRevD.84.084009.

[24] C. A. Kennedy and M. H. Carpenter, "Diagonally implicit Runge–Kutta methods for ordinary differential equations. A review" (NASA Langley Research Center, Hampton, VA) NF1676L-19716 (2016) 1–156; https://ntrs.nasa.gov/citations/20160005923.

[25] S. Klimenko et al., "Method for detection and reconstruction of gravitational wave transients with networks of advanced detectors", *Phys. Rev. D* **93** (2016) Article Id 042004; doi:10.1103/PhysRevD.93.042004.

[26]   J. Laskar, "A numerical experiment on the chaotic behaviour of the Solar System", *Nature* **338** (1989) 237–238; doi:10.1038/338237a0.

[27]   J. Laskar, T. Quinn and S. Tremaine, "Confirmation of resonant structure in the Solar System", *Icarus* **95** (1992) 148–152; doi:10.1016/0019-1035(92)90196-E.

[28]   R. I. McLachlan and P. Atela, "The accuracy of symplectic integrators", *Nonlinearity* **5** (1992) Article Id 541; doi:10.1088/0951-7715/5/2/011.

[29]   P. E. Olsen, J. Laskar, D. V. Kent, S. T. Kinney, D. J. Reynolds, J. Sha and J. H. Whiteside, "Mapping Solar System chaos with the geological Orrery", *Proc. Natl. Acad. Sci. USA* **116** (2019) 10664–10673; doi:10.1073/pnas.1813901116.

[30]   G. D. Quinlan and S. Tremaine, "Symmetric multistep methods for the numerical integration of Planetary orbits", *Astron. J.* **100** (1990) 1694–1700; doi:10.1086/115629.

[31]   C. L. Rodriguez, S. Chatterjee and F. A. Rasio, "Binary black hole mergers from globular clusters: masses, merger rates, and the impact of stellar evolution", *Phys. Rev. D* **93** (2016) Article Id 084029; doi:10.1103/PhysRevD.93.084029.

[32]   P. Saha and S. Tremaine, "Symplectic integrators for Solar System dynamics", *Astron. J.* **104** (1992) 1633–1640; doi:10.1086/116347.

[33]   J. G. Verwer, "Explicit Runge–Kutta methods for parabolic partial differential equations", *Appl. Numer. Math.* **22** (1996) 359–379; doi:10.1016/S0168-9274(96)00022-0.

[34]   S. J. Walters and L. K. Forbes, "Fully 3D Rayleigh–Taylor instability in a Boussinesq fluid", *ANZIAM J.* **61** (2019) 286–304; doi:10.21914/anziamj.v61i0.13700.

[35]   J. Wisdom and M. Holman, "Symplectic maps for the *n*-body problem", *Astron. J.* **102** (1991) 1528–1538; doi:10.1086/115978.

[36]   R. E. Zeebe, "Numerical solutions for the orbital motion of the Solar System over the past 100 Myr: limits and new results", *Astron. J.* **154** (2017) Article Id 193; doi:10.3847/1538-3881/aa8cce.

[37]   D. K. Zhang, "Discovering new Runge–Kutta methods using unstructured numerical search", Preprint, 2019, arXiv:1911.00318.

[38]   W. Zhao, J. Huang and S. J. Ruuth, "Boundary treatment of high order Runge–Kutta methods for hyperbolic conservation laws", *J. Comput. Phys.* **421** (2020) Article Id 109697; doi:10.1016/j.jcp.2020.109697.