Copyright Statement

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

This thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of this thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from their thesis.

To request permissions please use the Feedback form on our webpage. [http://researchspace.auckland.ac.nz/feedback](http://researchspace.auckland.ac.nz/feedback)

General copyright and disclaimer

In addition to the above conditions, authors give their consent for the digital copy of their work to be used subject to the conditions specified on the Library Thesis Consent Form and Deposit Licence.
Efficient algorithms for modeling close-encounters of the Solar System

Shafiq Ur Rehman
May 3, 2013

Supervisor: Dr. Philip W. Sharp

A thesis submitted in fulfillment of the requirements of Doctor of Philosophy in Applied Mathematics
Abstract

$N$-body simulations are used extensively to model the dynamical evolution of the Solar System. Many of these simulations have a small number of massive bodies representing the Sun and planets, and a large number of test particles representing massless bodies, such as asteroids and comets. Massive bodies act upon one another and the massless bodies, but the massless bodies do not interact with each other. The small bodies can come very close to a planet, an event known as a close-encounter, and even hit the planet. A small body is usually removed from the simulation if it makes a close-encounter with a massive body or is ejected from the Solar System.

A numerical scheme for performing simulations of massive and massless bodies consists of at least four essential parts: the integration of the massive bodies, the evaluation of the positions of the massive bodies using an interpolating polynomial, the integration of the massless bodies, and the detection of close-encounters. We present and illustrate a general multirate algorithm. The performance of the algorithm is illustrated by numerical tests that involve comparing combinations of different integrators and interpolation schemes on a collection of test problems for short and long intervals of integration. We also investigate the growth of the global error in the positions and velocities of the bodies, and the relative error in the energy and angular momentum of the system. In addition, we investigate the close-encounter errors in time, positions and velocities of the bodies.
First and foremost, I wish to acknowledge my indebtedness to Allah Almighty, the Omnipotent and the Benevolent, for providing me with this opportunity and endowing upon me the potential to carry out this study.

I owe special debt of thanks to my esteemed supervisor, Dr. P. W. Sharp. His comprehensive knowledge of the subject, wonderful programming skills, critical comments, and guidance throughout the process of this PhD project, deserve my special gratitude. My special thanks and appreciation to my co-supervisor Prof. H. M. Osinga, for her valuable suggestions, discussions, and help in editing this thesis. I also wish to put on record my special thanks to Prof. J. C. Butcher and other members of the numerical analysis group. The numerical analysis workshops are always fruitful for me in learning more about the subject.

I am grateful to the Higher Education Commission (HEC) of Pakistan for providing the funding to carry out this research. Special thanks go to Dr. Atta Ur Rehman (former chairman of HEC), who is arguably credited for reviving the higher education and research practices in Pakistan. I thank the University of Auckland for the financial support to participate in conferences.

Finally, I offer my humble regards and heartfelt thanks to my beloved parents, who deserve the utmost acknowledgement for their guidance, prayers, sacrifices, and enabling
me to attain the sense of self-worth that I have achieved today. Special thanks also to my sisters, brothers, my wife, and children, Muneeb Ur Rehman and Haseeb Ur Rehman, who always have supported and encouraged me throughout this work. May Allah Almighty bless them with Eeman, health, happiness, and peace, Ameen.

Last but not the least, of course, I thank all the HEC scholars graduated and/graduating at the University of Auckland, who have been contributing to this research directly or indirectly. Needless to say, however, any errors and mistakes remain the researcher’s responsibility.

Shafiq Ur Rehman
Dedication

To my deceased mother, whose sweet memories will remain with me life-long.
Contents

Abstract iii

Acknowledgments iv

Symbols and abbreviations xx

1 Introduction 1

1.1 Background ........................................ 2

1.2 Overview ........................................ 3

1.3 Imaginary sphere .................................. 8

1.3.1 Activity sphere ................................ 8

1.3.2 Hill’s sphere ................................ 10

1.4 Test problems .................................. 11
1.4.1 Kepler’s two-body problem ............................................. 12
1.4.2 Jovian problem ....................................................... 13
1.4.3 Helin-Roman-Crockett problem ....................................... 13
1.4.4 Asteroidal problem ................................................... 16
1.5 Structure of the Thesis .................................................. 17

2 Numerical integrators ....................................................... 18

2.1 Reference solution and errors ........................................... 19
2.2 Numerical methods and integrators ..................................... 26

2.2.1 One step methods for first-order ODEs ................................. 26

2.2.1.1 Explicit Runge-Kutta-Nyström methods .......................... 26
2.2.1.2 Round-off error control for ERKN methods .................... 30

2.2.2 Störmer methods ...................................................... 32
2.2.3 Extrapolation integrators ............................................. 34

2.2.3.1 ODEX2 integrator ............................................... 34

2.3 Step-size variation ...................................................... 35
2.4 Numerical experiments for long-term simulation ....................... 36

3 Accuracy and efficiency of interpolation schemes ....................... 46

3.1 Continuous approximations ............................................. 47
3.1.1 Hermite interpolation schemes
  3.1.1.1 Newton divided difference approach
  3.1.1.2 Cubic Hermite interpolation
  3.1.1.3 Quintic Hermite interpolation
  3.1.1.4 Two-step Hermite interpolation polynomial
  3.1.1.5 Three-step Hermite interpolation polynomial

3.1.2 Continuous approximation with embedded Runge-Kutta-Nyström methods

3.1.3 Continuous approximation with ODEX2

3.1.4 Computational cost of interpolation schemes

3.2 Numerical experiments for long-term simulations
  3.2.1 Kepler problem with different eccentricities

4 Algorithm and initial conditions
  4.1 Core Algorithm
    4.1.1 The routine Massive
    4.1.2 The routine TestParticle
    4.1.3 CloseEncounter
    4.1.4 Removal
  4.2 Other integration details
4.3 Initial conditions of asteroids ............................................. 87

5  N-Body simulations ..................................................... 90

5.1 Selection of quality asteroids ........................................ 91

5.2 Close-encounter error .................................................. 95

5.2.1 Detecting close-encounter using NLES ............................ 97

5.2.1.1 Linear inverse interpolation scheme ............................ 98

5.2.2 Admissible close-encounter error ................................. 100

5.3 Close-encounters at a given numerical accuracy ................. 103

5.4 Best observed numerical accuracy ................................... 111

6 Summary ................................................................. 116

A Appendix ....................................................................... 120

A.1 Jovian problem ............................................................ 121

A.2 HRC problem .............................................................. 122

A.3 Coefficients - ERKN689 ................................................. 122

References ...................................................................... 126
List of Figures

1.1 The close-encounters of an asteroid with a planet, where the planet is at the centre of the sphere with radius $R_{act}$. .................................................. 4

1.2 System of three bodies: the Sun, planet and a test particle. ................................. 9

1.3 The distance of the comet from Jupiter in the HRC problem. ............................. 14

1.4 The two-dimensional phase portrait for the $x$- and $y$-components of the position of the comet relative to Jupiter (blue dot) in the HRC problem. Plotted is the orbit segment from $t = 2000$ (A) to $t = 6000$ days (B). .......................... 15

1.5 The step-size sequence versus time using the integrators ERKN689, ERKN101217 and ODEX2 on the HRC problem. ....................................................... 16

2.1 The maximum global error in the position for the variable-step-size integrators $ODEX2$, ERKN689, and ERKN101217 applied to the Jovian problem over one million years for local error tolerances ranging from $10^{-16}$ to $10^{-08}$. 37
2.2 The error growth in position for the integrators $S-13$, \textit{ODEX}2, \textit{ERKN}689, and \textit{ERKN}101217 (the graphs are smoothed using a filter command with $WS = 50$; filtering is explained later in the section) applied to the Jovian problem over one million years. The selection of local error tolerances is subject to a prescribed accuracy (maximum global error $\approx 10^{-4}$), labeled by \textit{ERKN}689-G and \textit{ERKN}101217-G, and best observed (smallest maximum global error) accuracy, labeled by \textit{ERKN}689-M, S-13-M, and \textit{ERKN}101217-M.

2.3 Experiments with the window size for the Matlab function filter. The window size is set to 1, 10, 20 and 50 in plots (a) to (d), respectively.

2.4 The error growth in the energy for the four integrators $S-13$, \textit{ODEX}2, \textit{ERKN}689, and \textit{ERKN}101217 (filter command used with $WS = 100$) applied to the Jovian problem over one million years. The local tolerance was chosen to give the smallest maximum global error (the ‘-M’ graphs), and a maximum global error $\approx 10^{-4}$ (the ‘-G’ graphs).

2.5 Efficiency plots showing the number $N_{fev}$ of function evaluations against the $L_2$-norm of the maximum global error in position, obtained for the variable-step-size integrators \textit{ERKN}689, \textit{ERKN}101217, and \textit{ODEX}2, applied to the Jovian problem over one million years with $TOL$ ranging from $10^{-16}$ to $10^{-08}$.

3.1 Classification of the errors when using a numerical integrator together with an interpolation polynomial.

3.2 The maximum global error in position for the cubic, quintic, 2-step, and 3-step Hermite interpolation schemes against different eccentricities applied to the two-body problem over a period of $2\pi$. 
3.3 The ratio of the maximum relative error in energy at the interior points to the endpoint (of a step) relative error in energy for the $\tilde{S}$-13 integrator using cubic and quintic Hermite interpolations (the Matlab filter command with WS = 100 was used) applied to the Jovian problem over one million years. 

3.4 The ratio of the maximum relative error in energy at the interior points to the endpoint relative error in energy for $ERKN101217$ using cubic, quintic, 2-step, 3-step Hermite interpolations, and 23-stage, 26-stage, 29-stage interpolants (the Matlab filter command with WS = 100 was used) applied to the Jovian problem over one million years for a local error tolerance of $10^{-11}$. 

3.5 The maximum relative error in the energy for the cubic, quintic, 2-step, and 3-step Hermite interpolation schemes against different step-size sequences applied to the Jovian problem over one period of Jupiter. 

3.6 The relative error in the energy for $ERKN101217$ using cubic, quintic, 2-step, 3-step Hermite interpolations, and 23-stage, 26-stage, 29-stage interpolants (the Matlab filter command with WS = 100 was used) applied to the Jovian problem over one million years for a local error tolerance of $10^{-11}$. 

3.7 The relative error in the energy for $ERKN689$ using cubic, quintic, 2-step Hermite interpolations, and the 12-stage interpolant (the Matlab filter command with WS = 100 was used) applied to the Jovian problem over one million years for a local error tolerance of $10^{-10}$. 

3.8 The relative error in the energy for $ODEX2$ using cubic, quintic, 2-step, 3-step Hermite interpolations, and its built-in interpolant (the Matlab filter command with WS = 100 was used) applied to the Jovian problem over one million years for a local error tolerance of $10^{-16}$. 

64 65 67 69 70 71
4.1 A schematic that presents different combinations in the main algorithm of the integration schemes of the planets and asteroids with interpolation schemes. ................................. 79

4.2 A schematic with interpolants that presents different combinations in the main algorithm. ................................................................. 80

4.3 The Keplerian orbital elements for a body whose orbital plane intersects a reference plane. ................................................................. 88

4.4 The orbits of Jovian planets and the initial positions for a set of 1000 asteroids, all projected on the x-y plane. ........................................ 89

5.1 (Top) The norm of the difference of the position at the end of the integration calculated using double and quadruple precision of a set of 1000 asteroids against the minimum distance to the Jovian planets using the ERKN101217 integrator with its 29-stage interpolant over one million days. The dashed lines bound four categories of asteroids as described in the bottom part of the figure. (Bottom) The classification of stable and unstable asteroids including those that do and do not make close-encounters (CE). ....... 92

5.2 The growth of the global error for a set of 5 quality asteroids using the ERKN101217 integrator with its 29-stage interpolant over approximately one million days, with $TOL = 10^{-16}$. Shown are the error growth (The Matlab filter command with WS = 100 was used for both position and velocity) in position (top) and the error growth in velocity (bottom). .... 93

5.3 An illustration of error in a single close-encounter detected during a simulation. The integration time-steps of three different combinations of the main algorithm are denoted by symbols: small-circle, triangle, and cross, respectively. ................................................................. 96
5.4 The error for all close-encounters in a simulation. ........................................ 100

5.5 An illustration of an admissible close-encounter error between double and quadruple-precision orbits, where the positions of the planet and asteroid are represented by blue and red dots, respectively and circles represent the activity spheres. .......................................................... 101

5.6 The close-encounter error as a function of time for a set of 1000 quality asteroids. .......................................................... 103

5.7 Close-encounter errors in time for a given integration accuracy of $10^{-04}$. Shown are the combinations of $ERKN689$ with quintic Hermite interpolation (top) and $ERKN689$ with its 12-stage interpolant (bottom). ............... 106
## List of Tables

1.1 The radii of the close-encounter spheres of the Jovian planets. ........... 11

2.1 The values of estimated maximum global error using ERKN101217 with 23-stage interpolant obtained with different values of $k$ applied to the Jovian problem over one million years for the local error tolerances $10^{-08}$, $10^{-10}$, $10^{-12}$, $10^{-14}$, and $10^{-16}$. The row labelled $k = 10$ shows errors, whereas the rows labelled $k = 50$ and $k = 100$ list the percentage changes. ......... 23

2.2 Our algorithm for controlling the step-size sequence, where without loss of generality we have assumed $h > 0$. ................................................................. 29

2.3 The values of $E_r,_{max}$ and $E_v,_{max}$ for ERKN689 and ERKN101217 obtained with and without round-off error control applied to the Jovian problem over one million years for the local error tolerances $10^{-08}$, $10^{-10}$, $10^{-12}$, $10^{-14}$, $10^{-16}$. ................................................................. 31

2.4 The step-size variation for the variable-step-size integrators ERKN689, ERKN101217, and ODEX2 applied to the Jovian problem over one million years, with the local error tolerance $TOL$ as specified in the first column. . 35
2.5 The maximum global error as a function of the step-size for the fixed-step-size integrator $\bar{S}$-13, applied to the Jovian problem over one million years. 40

3.1 An illustrative Newton divided difference table. ............................ 50

3.2 The CPU-time in seconds for evaluating the stage derivatives (without the cost of additional function evaluations) for ERKN689 and ERKN101217 applied to the Jovian problem. .................................................. 57

3.3 The CPU-time in seconds for finding the coefficients of the interpolation schemes and evaluating all the stage derivatives of the interpolants applied to the Jovian problem. .................................................. 57

3.4 The CPU-time in seconds for evaluating the position and velocity polynomials using different interpolation polynomials applied to the Jovian problem. 58

3.5 The CPU-time in seconds for evaluating the acceleration for one asteroid using different interpolation polynomials. ................................. 58

3.6 The maximum global error in position for eccentricity 0.05 attained by different interpolation schemes applied to the Kepler problem over the interval $[0, 2\pi]$ with four choices of numbers of evenly spaced sub-intervals. The dash means the combination is not used. ................................. 62

3.7 The exponent $b$ of the power law for the global error in position and velocity, and relative error in the energy and angular momentum for the Jovian problem over one million years. We used local error tolerances $10^{-11}$, $10^{-10}$, and $10^{-16}$ with different interpolation schemes, for the integrators ERKN689, ERKN101217, ODEX2, and $\bar{S}$-13, respectively. The step-size for $\bar{S}$-13 is four days. ................................................................. 72

4.1 The core algorithm for our N-body simulations. ............................... 77
4.2 Description of the six orbital elements. ............... 89

5.1 An analysis of detecting close-encounters using plausible combinations of the main algorithm for a maximum global error of $10^{-04}$. Only those combinations that detect at least 99% true close-encounters are listed. .......... 104

5.2 The close-encounter errors in time using those combinations of the main algorithm that detect 100% true close-encounters. The local error tolerance is chosen so the maximum global error is approximately $10^{-04}$. .............. 107

5.3 The close-encounter errors in the positions and velocities of the planets and asteroids. Integrations are done for a maximum global error of $10^{-04}$. Shown are close-encounter errors in positions (top) and velocities (bottom). 108

5.4 Close-encounter errors in time using those combinations of the main algorithm that detect 100% true close-encounters. Integrations are done for a maximum global error of $10^{-03}$. .................................................. 109

5.5 The close-encounter errors in the positions and velocities of the planets and asteroids. The integrations are done for a maximum global error of $10^{-03}$. Shown are close-encounter errors in positions (top) and velocities (bottom). 110

5.6 The close-encounter errors in time at best observed integration accuracy using those combinations of the main algorithm that detect 100% true close-encounters. .................................................. 111

5.7 The close-encounter errors in the positions and velocities of the planets and asteroids integrated at best observed accuracy. Shown are close-encounter errors in positions (top) and velocities (bottom). ................................. 113

5.8 Study of additional accuracy measures using the same combinations and integration tolerances as in Table 5.6. .......................................................... 114
A.1 The values $\mu_j, j = 1, \ldots, 5$, for the five bodies ordered from Sun, Jupiter, Saturn, Uranus and Neptune. .................. 121

A.2 Rows 1 to 5 are the initial positions and Rows 6 to 10 are the initial velocities for the Jovian problem. .................. 121

A.3 Rows 1 to 6 are the initial positions and Rows 7 to 12 are the initial velocities for the HRC problem. .................. 122

A.4 Coefficients used for the interpolants of ERKN689. .................. 125
Symbols and abbreviations

′ - differential operator
· - dot product
|| · ||_2 - $L_2$-norm of a vector

$\beta$ - safety factor for step-size selection
$\delta$ - estimated round-off error on the previous step (described in Equation (2.2.5))
$\Delta t$ - time-step larger than the typical step-size, used especially for the integration of planets
$\eta$ - the eccentric anomaly of Kepler’s equation
$\mu$ - $Gm$, where $m$ is the mass of the body and $G$ the gravitational constant
$\nu$ - the true anomaly in Figure 2.2
$\omega$ - the ascending node in Figure 2.2
$\Omega$ - argument of periapsis in Figure 2.2

$a$ - the instantaneous semi-major axis of a planet’s orbit
$a_{ij}$ - the interior weights of RK or RKN method
$\alpha_{dw}$ - the minimum allowable ratio of consecutive step-sizes
$\alpha_{up}$ - the maximum allowable ratio of consecutive step-sizes
$b_j$ - exterior weights of the solution formulae used for RK or RKN method
$b'_j$ - exterior weights of the derivative formulae used for RK or RKN method
- exterior weights of the higher-order derivative formulae used for RK or RKN method
\( \hat{b}_j \) - exterior weights of the lower-order solution formulae used for RK or RKN method
\( c_j \) - the abscissas of RK or RKN method
\( d_{ij} \) - distance between the \( i^{th} \) and \( j^{th} \) body
\( e \) - the eccentricity of Kepler’s two-body problem
\( E_r \) - the global error in the position of \( N \)-body problem
\( E_v \) - the global error in the velocity of \( N \)-body problem
\( \vec{F}_1 \) - the Solar attraction
\( \vec{F}_2 \) - the planetary attraction
\( \vec{F}_1' \) - the disturbing force of the heliocentric orbits
\( \vec{F}_2' \) - the disturbing force of the planetocentric orbits
\( G \) - the gravitational constant
\( H \) - the total energy
\( H_0 \) - the initial energy
\( H_{rel} \) - the relative error in energy
\( h \) - the time-step of the integration
\( h_{\text{max}} \) - the maximum allowable step-size
\( h_{\text{mean}} \) - the mean step-size (described in Table 2.4)
\( h_{\text{min}} \) - the minimum allowable step-size
\( h_{\text{min}} \) - percentage variation in the minimum step-size (described in Table 2.4)
\( h_{\text{max}} \) - percentage variation in the maximum step-size (described in Table 2.4)
\( h_{p,i} \) - the initial step-size of the \( i^{th} \) test particle
\( K_j \) - the stage values for ERK or ERKN method
\( L \) - the total angular momentum
\( L_0 \) - the initial angular momentum
\( L_e \) - local error of the approximation to \( y \)
\( L_{e_{\text{max}}} \) - maximum of local error of the approximations to \( y \) and \( y' \)
\( L_{rel} \) - relative error in angular momentum
\( m \) - mass of a body (excluding the Sun)
\( M \) - mass of the Sun
\( N \) - total number of bodies
\( N_a \) - number of asteroids
\( N_{\text{fcn}} \) - number of function evaluations
\( N_m \) - number of massive bodies
\( N_{\text{sub}} \) - number of sub-intervals
\( p \) - lower-order approximation of variable-step-size method
\( P_m(t) \) - polynomial of degree \( m \)
\( q \) - higher-order approximation of variable-step-size method
\( r \) - position vector
\( r'' \) - acceleration vector
\( \bar{r}_6 \) - position vector of HRC comet
\( \bar{r}_6'' \) - acceleration vector of HRC comet
\( R_{\text{act}} \) - radius of the activity sphere
\( R_{\text{err}} \) - the ratio of the relative error (described in Equation (3.1.1))
\( R_H \) - radius of the Hill sphere
\( r_{\text{num}} \) - the position vector of a body calculated numerically
\( r_{\text{true}} \) - the position vector of a body in the reference or true solution
\( s \) - the number of stages for the method
\( t \) - time
\( t_0 \) - initial time
\( t_f \) - pre-specified final time
\( t_{n-1} \) - time at the previous step
\( t_{\text{num}} \) - the numerical time of a close-encounter
\( TOL \) - local error tolerance for an integration method
\( TOL_a \) - the absolute tolerance for NLES
\( TOL_f \) - the \( f \)-tolerance for NLES
\( TOL_r \) - the relative tolerance for NLES
\( t_{\text{ref}} \) - the reference time of close-encounter
\( y_{\text{true}} \) - true solution at any time
\( y_{\text{ref}} \) - reference solution at any time
\( y_{\text{num}} \) - numerical approximation at initial time
\( y_0 \) - numerical approximation at initial time
CE - close-encounter
CPU-time - computational time
ERKN689 - explicit Runge-Kutta-Nyström, 9 stage, 6-8 pair
ERKN101217 - explicit Runge-Kutta-Nyström, 17 stage, 10-12 pair
FSAL - first same as last
i - the inclination in Figure 2.2
NLES - non-linear equation solver
NR - Newton-Raphson method
ODEX2 - variable-step-size and order, extrapolation integrator
QDF - quadratic formula
QRF - quartic formula
RK - Runge-Kutta method
RKN - Runge-Kutta-Nyström method
SEC - Secant method
WS - window size used for the filter command in Matlab
$N$-body simulations of the Sun, the planets, and small celestial bodies are frequently used to model the evolution of the Solar System. Large numbers of numerical integrators and associated interpolation schemes for performing such simulations have been developed and used; see, for example, [30, 33, 41, 46, 48, 60]. One of the objectives of this thesis is to analyse and compare the efficiency and error growth for different numerical integrators and interpolation schemes. Throughout the thesis the error growth is examined in terms of the global error, that is, the errors in the positions and velocities of the $N$ bodies, the relative errors in the energy and angular momentum of the system, and the phase error. We also investigate the errors in time, positions and velocities when two bodies come non-negligibly close together, at so-called close-encounters. We begin with an overview of
the relevant literature and discuss some notation and background for $N$-body simulations of the Solar System. We also introduce some interesting test problems.

1.1 Background

The history of astronomy is arguably the oldest of the natural sciences and is also the history of awareness of mankind in this universe. We can trace the origins of astronomy in the religious practices of some early cultures who associated celestial objects with gods and spirits. Our Solar System consists of the Sun; eight relatively solitary planets that are classified as Terrestrials (inner planets), namely, Mercury, Venus, Earth, Mars, and the Gas giants (outer planets) Jupiter, Saturn, Uranus, Neptune; and a large number of smaller bodies, such as moons, asteroids and comets. The inner planets are primarily composed of silicate rocks and within the Solar System they are closest to the Sun. The Gas giants are largely made-up of gaseous material and liquids, for example, Hydrogen, Helium, and they are significantly more massive than Terrestrials. Gas giants collectively drive much of the dynamics of the Solar System. Therefore, in this thesis, we focus on the Gas giants.

Key mathematical discoveries in astronomy were Kepler’s laws of planetary motion and Newton’s formulation of the universal law of gravitation (Newton’s law of gravity). These laws describe the orbits of planets, asteroids, comets and satellites, and their possible future motion. Sometimes, these motions are very systematic and essentially repeating, as in the case of a planet orbiting about the Sun, or the Moon about the Earth; in contrast, sometimes there is seemingly no repetition, as when an asteroid is ejected from the Solar System.

Diverse applications, for example, the explanation for the five mass extinctions [51], the Chicxulub crater [1], and the evolution of the Kuiper belt [58], have made this field of astronomy increasingly attractive for the researchers.
1.2 Overview

Computational astronomers make extensive use of accurate $N$-body simulations when studying the dynamics of the planets, asteroids and other small celestial bodies in the Solar System. These simulations are performed by first deriving a set of ordinary differential equations for the acceleration of the $N$ bodies in the simulation, and specifying the initial positions and velocities of the bodies at time $t = t_0$. Generally, the initial value problems (IVPs) that occur for $N$-body simulations are a mixture of first- and second-order differential equations, but the sort of problems we are considering are of the form,

$$y''(t) = f(t, y(t)), \quad y(t_0) = y_0, \quad y'(t_0) = y_0', \quad (1.2.1)$$

where $y_0 \in \mathbb{R}^k$ and $y_0' \in \mathbb{R}^k$ denote the initial positions and velocities, the operator $'$ denotes differentiation with respect to time $t$, and $f : \mathbb{R} \times \mathbb{R}^k \to \mathbb{R}^k$ is a sufficiently smooth function. Here, $k$ is the dimension of the IVP, which in some cases may change over time, as bodies are added or removed in the simulations. In some cases, (1.2.1) can be solved analytically, but mostly the differential equations are too complicated to find analytical solutions, necessitating the use of approximation techniques to find the numerical approximate solution.

Most models of $N$-body simulations use more than one type of particle. The Sun and planets are represented as massive bodies and the small bodies as massless bodies. The massive bodies act upon one another and the massless bodies, but the massless bodies do not interact with each other (theoretically they do interact but their interaction is negligible compared to that with the massive bodies). The gravitational attraction between bodies depends on the mass and the distance between the bodies. When the bodies in a simulation consist of the Sun and planets only, no two bodies come close to one another. This is quite different from the case when small bodies such as asteroids are included in the simulation, because these bodies can come close to a planet, an event known as a close-encounter, and even hit the planet. Therefore, an important property of
numerical schemes for such simulations is to detect and handle close-encounters accurately. A simple detection scheme is to calculate the distance between the small body and a planet: if the distance is less than the radius of an imaginary sphere (discussed in Section 1.3) then the small body is said to have made a close-encounter. There are several definitions for the imaginary sphere, the most commonly used being the activity and Hill’s spheres. We refer to these spheres as close-encounter spheres. The details about close-encounter spheres are discussed in Section 1.3.

Figure 1.1 depicts an asteroid making close-encounters to a planet, where the planet is placed at the centre of a sphere having radius $R_{act}$. Figure 1.1 illustrates three cases of close-encounters. The first, illustrated by Asteroid$^1$, is the most typical case, where the trajectory of the asteroid lies well inside the activity sphere. The second possibility is
illustrated by Asteroid², which has collided with the planet. The third case illustrated by Asteroid³, is potentially difficult to detect as a close-encounter, because the trajectory of the asteroid is just touching the activity sphere. For comprehensive details on detecting close-encounters using interpolation schemes and non-linear equation solvers, we refer to Section 5.2 in Chapter 5.

The acceleration of the small body undergoing a close-encounter is dominated by that due to the planet. Since the acceleration depends inversely on the square of the distance, the magnitude of the acceleration varies considerably during a close-encounter, necessitating a large variation in the integration time-step \( h \). This variation can lead to an inefficient simulation unless care is taken. For example, in a simulation of the Sun, Jupiter and an asteroid, suppose that the asteroid had a close-encounter with Jupiter and \( h \) was reduced to \( h/100 \) for both the asteroid and for Jupiter. This wastes computer time, because to a very good approximation, and one that is often used in accurate \( N \)-body simulations, asteroids and other small bodies are too small to affect the motion of the planets. Hence, computational time could potentially be saved if the original time-step \( h \) was retained for Jupiter.

Several computational time-saving schemes have appeared in the literature; see, for example, [9, 27, 31, 40]. These schemes depend strongly on whether the integrator used to advance the position of the Sun, planets and small bodies is symplectic or non-symplectic (symplectic integrators are special numerical methods that inherit the property of symplecticness when applied to Hamiltonian systems; see, for example, [6]). Symplectic integrators have the advantage of ensuring that the error in the energy remains bounded. Except for very large \( t \), non-symplectic integrators have the advantage of producing numerical solutions with a smaller error in the position and velocity than for symplectic integrators, provided \( h \) is sufficiently small. In this thesis we will concentrate on non-symplectic integrators, because we are interested in performing accurate simulations.

A general scheme for handling close-encounters is the following. Suppose an asteroid
begins a close-encounter at \( t = t_a \) and finishes the close-encounter at \( t = t_b \). Let \( y_1 \) denote the position of the Sun and the planets, \( y_2 \) the position of the small body undergoing the close-encounter, and let \( z \) be the position of the Sun and planets that is obtained by continuous approximation; for details on continuous approximation, see Chapters 3 and 4. The original system (1.2.1) restricted to just the Sun, the planets, and the asteroid is then written as

\[
\begin{align*}
y_1'' &= f_1(t, y_1), \\
y_2'' &= f_2(t, y_2, z).
\end{align*}
\] (1.2.2)

A numerical solution to the \( y_1 \)-system at \( t = t_a + h \) is found by taking a time-step of size \( h \). The \( y_2 \)-system is then integrated from \( t_a \) to \( t_a + h \) using time-steps no larger than \( h \). This integration will require the position of the Sun and planets for \( t \in (t_a, t_a + h) \). The position, in general, is not available from the integration of the \( y_1 \)-system and is found by interpolation (continuous approximation). This involves fitting a polynomial or piecewise polynomial to the position, and optionally the velocity and acceleration, and evaluating the polynomial at the required time \( t \), with \( t_a \leq t \leq t_a + h \).

Specific schemes of the above type have been used; see, for example, [9, 27, 31, 40]. One of the goals of this thesis is to investigate two important properties, namely, accuracy and efficiency of the \( N \)-body simulations when integration schemes combined with continuous approximations are used. This can be done using numerical testing for different types of non-symplectic integrators and interpolation methods. The main sources of errors involved in these kinds of simulations are from the integration, the continuous approximations, and from possibly round-off error.

The order of the continuous approximation should, with one notable exception in this thesis, be compatible with the order of the integrator. For example, for an integrator of order \( p \), there should be an interpolation polynomial of the same order for a sufficiently accurate approximation to the positions, velocities, and possibly accelerations of the massive bodies and the massless bodies. The notable exception occurs for high order Störmer
methods when used with small step-sizes. The order of the continuous approximation can be significantly less than the order of the Störmer method.

The accurate approximation of the orbits of the massive bodies at any time \( t \) is important, because the accuracy influences the accuracy of the orbits of the massless bodies. Similarly, it is equally important to use an appropriate interpolation scheme to obtain accurate approximations of the times of the close-encounters as well as to the positions and velocities of both the massive and massless bodies at the times of close-encounters. The interpolation error for a polynomial of degree \( p \) can be written as

\[
\alpha h^{p+1} \frac{y_{\text{true}}^{(p+1)}(\xi)}{(p+1)!},
\]

(1.2.3)

where \( y_{\text{true}} \) is the function being approximated, \( \alpha \) is a constant, and \( \xi \) lies in the interval over which the polynomial is applied. This expression, as is, cannot be used for our analysis because the true solution \( y_{\text{true}} \) is unknown for general \( N \). However the Solar acceleration dominates the acceleration between planets. This means that the motion of the planet is described, to a good approximation for our analysis, by the two-body problem consisting of the Sun and the planet. This problem can be solved analytically and provides a solution that can be used to approximate the derivative in expression (1.2.3).

Another difficulty with expression (1.2.3) is that it assumes the data used to form the polynomial is exact. This assumption does not hold for \( N \)-body simulations because, for example, the position of a planet at the end of several consecutive steps, will contain errors from the integration. The integration error grows as \( t^{3/2} \) for some integrators and as \( t^2 \) for others [4, 28, 29, 34]. In contrast, the interpolation error does not grow with \( t \), so that the interpolation error will become insignificant for sufficiently large \( t \). As soon as this happens, the interpolation method is inefficient because it is too accurate. For planetary orbits, the interpolation error typically decreases with increasing \( p \). This means that the efficiency of the interpolation can be improved by decreasing \( p \) with \( t \). The dependence of \( p \) on \( t \) will both complicate and simplify our analysis. The complication comes from having to analyse different polynomials. The simplification comes from low-degree polynomials.
being easier to analyse than high-degree polynomials.

Our numerical testing will involve comparing combinations of different non-symplectic integrators and interpolation methods on a collection of test problems. These problems will range from those involving one small body to those with 100,000 bodies. We have chosen the number 100,000 because more small bodies are rarely used in simulations. We use the results of the numerical experiments to eliminate combinations of integrators and interpolation methods. For example, using linear polynomials \((p = 1\) in (1.2.3)) is unlikely to produce sufficiently accurate results except possibly at extremely large \(t\). One challenge for the numerical testing will be to assess the influence of round-off error.

1.3 Imaginary sphere

All the interactions between the massive and the massless bodies are through the Newtonian gravitational forces. This gravitational influence can be approximated as an imaginary sphere centred on the particular planet. We use the most common definitions for the imaginary spheres and refer to them, as noted previously, as close-encounter spheres. Here, we present an overview of these close-encounter spheres.

1.3.1 Activity sphere

One definition of the imaginary sphere is the activity sphere, which is based on a balance of perturbing forces. Consider a system of three bodies: the Sun with mass \(M\) at the origin, a planet with mass \(m\) at position \((x_p, y_p, z_p)\), and a test particle with negligible mass at position \((x_t, y_t, z_t)\), as shown in Figure 1.2. Here, the Sun is at the centre of this coordinate system, which is known as the heliocentric frame (for simplicity we have taken the heliocentric coordinate system but this idea also applies to other origins, for example, the centre of mass). The test particle is said to be in the heliocentric region if it is far from the planet, so that the main gravitational force will be that due to the Sun and the planet may be considered as a perturbing body. Otherwise, the test particle is said to be
Figure 1.2: System of three bodies: the Sun, planet and a test particle.

in the planetocentric region, where the planetary attraction will be greater than the Solar attraction, so that the Sun should be considered as the perturbing body.

Whether the test particle is in the heliocentric or planetocentric region depends upon the ratio between the total disturbing force and the appropriate central attraction. The boundary of the imaginary sphere is, therefore, the surface on which the ratio of the total disturbing force with the planet’s central attraction is equal to that of the Sun. Hence, we must have

\[
\frac{||\vec{F}_1'||2}{||\vec{F}_1||2} = \frac{||\vec{F}_2'||2}{||\vec{F}_2||2}.
\]

(1.3.1)

where \(||.||_2\) denotes the \(L_2\)-norm and \(\vec{F}_1'\) and \(\vec{F}_2'\) are the disturbing forces of the heliocentric and planetocentric orbits, respectively, and \(\vec{F}_1\) and \(\vec{F}_2\) are the Solar and planetary attractions, respectively. The solution of equation (1.3.1) gives the radius \(R_{act}\) of the activity sphere. An approximation of \(R_{act}\) can be obtained using

\[
R_{act} = a \left(\frac{m}{M}\right)^{2/5},
\]
where $a$ is the semi-major axis of the planet’s orbit and $\frac{m}{M}$ is the ratio of planetary and Solar masses (for detailed calculations see [26]).

### 1.3.2 Hill’s sphere

A second definition of the imaginary sphere is the Hill’s sphere, which is a region around the planet in which the planetary attraction prevents the test particle from moving into the heliocentric orbit. The region is calculated using the equations of motion of the circular restricted three-body problem, a special problem in which two of the bodies move in a circular orbit and the mass of the third body is assumed negligible. The three-body problem is solved in rotating Cartesian coordinates with the $(x, y)$-plane in the plane of motion of the two massive bodies and the $x$-axis along their line of centres. The equations of motion of the massless body in $(x, y, z)$-space are given as

$$
\begin{align*}
\dddot{x} - 2\dot{y} - x &= -\mu_1 \frac{x + \mu_2}{r_1^3} - \mu_2 \frac{x - \mu_1}{r_2^3}, \\
\dddot{y} + 2\dot{x} - y &= -\mu_1 \frac{y}{r_1^3} - \mu_2 \frac{y}{r_2^3}, \\
\ddot{z} &= -\mu_1 \frac{z}{r_1^3} - \mu_2 \frac{z}{r_2^3},
\end{align*}
$$

(1.3.2)

where $\mu_1$ and $\mu_2$ are the masses of the two main bodies, scaled so that $\mu_1 + \mu_2 = 1$, and $r_1^2 = (x + \mu_2)^2 + y^2 + z^2$, $r_2^2 = (x - \mu_1)^2 + y^2 + z^2$.

By making various assumptions, the system of equations of motion of the test particle in relation to the planet reduces to what are known as Hill’s equations [49],

$$
\begin{align*}
\dddot{x} - 2\dot{y} &= (3 - \frac{\mu_2}{r_2^3})x, \\
\dddot{y} + 2\dot{x} &= -\frac{\mu_2}{r_2^3}y.
\end{align*}
$$

(1.3.3)

The Hill’s sphere radius $R_H$ is the distance at which the radial forces vanishes, i.e., at this distance, the Solar tide and mutual attraction are in equilibrium in Hill’s equations [49]. Different approximations are used to obtain the radius of Hill’s sphere. To give
consistency with the definition for the radius of the activity sphere, we use

\[ R_H = a \left( \frac{m}{M} \right)^{1/3} \]

Table 1.1 lists the radii of close-encounter spheres of the Jovian planets, calculated using the data obtained from Williams [63] and expressed in astronomical units (1 A.U = 149597870 km). From Table 1.1 we observe that \( R_{\text{act}} < R_H \) for all four Jovian planets and the radius of Jupiter’s activity sphere is the smallest, while that of Neptune’s Hill’s sphere is the largest. We also observe from Table 1.1 that \( R_H \) monotonically increases with distance of the Gas giants to the Sun but \( R_{\text{act}} \) is non-monotonic.

<table>
<thead>
<tr>
<th>Sphere</th>
<th>Jupiter</th>
<th>Saturn</th>
<th>Uranus</th>
<th>Neptune</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity : ( R_{\text{act}} )</td>
<td>0.322258</td>
<td>0.364620</td>
<td>0.346036</td>
<td>0.579209</td>
</tr>
<tr>
<td>Hill’s : ( R_H )</td>
<td>0.512331</td>
<td>0.628208</td>
<td>0.675749</td>
<td>1.118706</td>
</tr>
</tbody>
</table>

Table 1.1: The radii of the close-encounter spheres of the Jovian planets.

Throughout the thesis, we use the smallest radius, that is, the radius \( R_{\text{act}} \) of Jupiter’s activity sphere, for detecting close-encounters.

### 1.4 Test problems

As is the case for the system of ODEs (1.2.1), many problems are independent of the first derivative of their solution and are also independent of time. These types of IVPs are known as autonomous second-order systems of ODEs. We now consider test problems of this type, namely, Kepler’s two-body problem, the Jovian problem, the Helin-Roman-Crockett (HRC) problem, and the Asteroidal problem consisting of the Sun, the Jovian planets and two or more test particles.
1.4.1 Kepler’s two-body problem

Kepler’s two-body problem (see, for example, [25]) is often considered as the simplest, integrable problem in Solar System dynamics, mainly, because an analytical solution exists. It models two point masses interacting through the effect of Newtonian gravitational forces. The variation of masses in the Solar System (the Sun is significantly more massive than the planets) and the large distance between the planets allows the orbits of most planets and test particles to be approximated by two-body motion, consisting of a test particle orbiting about a much larger central body. This orbiting motion is planar and the equations of motion of Kepler’s problem can be written as

\[ y''_1 = \frac{-y_1}{(y_1^2 + y_2^2)^{3/2}}, \quad y''_2 = \frac{-y_2}{(y_1^2 + y_2^2)^{3/2}}, \]  

(1.4.1)

where \( y_1 \) and \( y_2 \) are the \( x \)- and \( y \)-coordinates of one body relative to the other. This use of \( y_1 \) and \( y_2 \) is distinct from the use in equations (1.2.2).

Without loss of generality the standard initial conditions can be chosen as

\[ y(0) = [1 - e, 0]^T, \quad y'(0) = [0, \sqrt{(1 + e)(1 - e)^{-1}}]^T, \]

where the parameter \( e \) is the orbital eccentricity and \( 0 \leq e < 1 \). The analytical solution to the above two-body problem is

\[ y_1 = \cos(\eta) - e, \quad y_2 = \sin(\eta)\sqrt{1 - e^2}, \]

\[ y'_1 = \frac{-\sin(\eta)}{1 - e \cos(\eta)}, \quad y'_2 = \frac{\cos(\eta)\sqrt{1 - e^2}}{1 - e \cos(\eta)}, \]

where the eccentric anomaly \( \eta \) satisfies Kepler’s equation \( \eta - e \sin(\eta) - t = 0 \), an implicit equation in \( \eta \). Except in the special cases when \( e \) and \( t \) are equal to zero or \( \pi \), the numerical solution to Kepler’s equation is found using a non-linear equation solver, such as the Newton-Raphson method.
1.4.2 Jovian problem

The Jovian problem (see, for example, [56]) models the orbital motion of the Sun and the four Gas giants, Jupiter, Saturn, Uranus and Neptune, interacting through Newtonian gravitational forces. The Jovian problem is often used in numerical experiments, because the Gas giants collectively drive much of the dynamics of the Solar System.

Let \( r_i = [x_i, y_i, z_i]^T, i = 1, ..., 5, \) be the position vector of the \( i^{th} \) body of the Jovian problem, where the bodies are ordered from Sun to Neptune and the coordinate system is the three-dimensional Cartesian system with the origin at the barycentre (centre of mass) of the bodies. Then the equations of motion for the \( i^{th} \) body can be written as

\[
\ddot{r}_i(t) = \sum_{j=1, j \neq i}^{5} \frac{\mu_j(r_j(t) - r_i(t))}{||r_j(t) - r_i(t)||_2^3}, \quad i = 1, ..., 5, \tag{1.4.2}
\]

where as before \(|.||_2\) denotes the \( L_2 \)-norm and \( \mu_j \) is the gravitational constant \( G \) times the mass \( m_j \) of the \( j^{th} \) body, i.e., \( \mu_j = Gm_j \). For each body we have a second-order differential equation for the \( x-, y-, \) and \( z- \)components, giving us fifteen second-order differential equations in total. We express distance in astronomical units, the independent variable \( t \) in Earth days and the mass \( m_j \) in Solar mass. The values of \( \mu_j \) are given in Appendix A.

1.4.3 Helin-Roman-Crockett problem

The Helin-Roman-Crockett (HRC) problem models the HRC comet having multiple close-encounters with Jupiter, during which the comet actually orbits Jupiter. The last such close-encounter was observed in 1976 and the comet is expected to make another close-encounter in the year 2075, during which it will be temporarily captured by Jupiter. This temporary capture has been modeled by researchers; see for example, Levison and Duncan [45]. The equations of motion for the HRC problem are the same as those for the Jovian
problem with the addition of the following equations for the comet at position $\vec{r}_6$

$$\dddot{\vec{r}}_6(t) = \sum_{i=1}^{5} \frac{\mu_i (r_i(t) - \vec{r}_6(t))}{||r_i(t) - \vec{r}_6(t)||_2^3}. \quad (1.4.3)$$

Here, the symbol $\vec{r}$ is used instead of $r$ to highlight that the comet is treated as a massless body. The initial conditions of the comet are given in Appendix A.

Figure 1.3 shows the distance of the comet to Jupiter over a time interval from $t = 0$ to $t = 10,000$ days. The graph clearly shows five close-encounters where the comet comes to within $10^{-2}$ distance from Jupiter. The figure indicates a possible sixth close-encounter at $t \approx 7000$ days, but the minimum distance to Jupiter is significantly larger than the other five local minima. Hence, this sixth local minimum is often not regarded as a close-encounter.

Figure 1.4 shows the two-dimensional phase portrait for the components of the position of the comet from $t = 2000$ to $t = 6000$ days. Here, Jupiter lies at the origin. The plotted trajectory is reminiscent of the petals of a flower; due to this similarity, such
plots are often referred to as petal plots or tulip diagrams.

The time-step plays a vital role in close-encounters. For example, the close-encounter Asteroid in Figure 1.1 could easily be missed if the time-step were too large and the distance between planet and test particle were calculated only at the end of the time-steps. The time-step must also be reduced because the magnitude of acceleration increases as the asteroid approaches the planet.

Figure 1.5 shows the step-size sequence for \( t \in [0, 10000] \) for different variable-step-size integrators, namely, ERKN689 (Explicit Runge-Kutta-Nyström, nine stage, 6-8 pair [12]), ERKN101217 (Explicit Runge-Kutta-Nyström, seventeen stage, 10-12 pair [12]) and the ODEX2 integrator [36]; these integrators are discussed in detail in Chapter 2. The maximum step-sizes taken by ERKN689, ERKN101217, and ODEX2 are approximately
Figure 1.5: The step-size sequence versus time using the integrators ERKN689, ERKN101217 and ODEX2 on the HRC problem.

19.98, 157.99 and 408.02 days, respectively. The integrations are performed in double precision using FORTRAN with a local error tolerance of $10^{-14}$. We have taken a time interval of 10,000 days to illustrate that the step-size decreases significantly as the comet makes close-encounters with the Jupiter. The biggest step-size variation has been observed with ODEX2, where it reduces from approximately 408.02 days to 7.6 days during one close-encounter.

1.4.4 Asteroidal problem

The Asteroidal problem is the Jovian problem with the addition of massless bodies representing asteroids. The equations of motion are the same as those for the Jovian problem with the addition of the equations for the asteroids. If $N$ is the total number of bodies in the simulation and $r_i$, $i = 6, \ldots, N$, the position of the $(i - 5)^{th}$ asteroid at any time $t$,
the equations of motion for the \((i - 5)\)th asteroid are

\[
\ddot{r}_i(t) = \sum_{j=1}^{5} \frac{\mu_j(r_j(t) - \bar{r}_i(t))}{||r_j(t) - \bar{r}_i(t)||^2}, \quad i = 6, ..., N. \tag{1.4.4}
\]

In this thesis we used up to 100,000 asteroids.

1.5 Structure of the Thesis

In the next chapter, we survey and measure the accuracy and the efficiency of different types of \(N\)-body integrators. We perform numerical experiments for the different integrators applied to the Jovian problem over a long interval, as long as 100 million years, for the local error tolerance ranging from \(10^{-16}\) to \(10^{-08}\). In Chapter 3, we investigate the accuracy and the efficiency of different types of interpolation schemes for a variety of gravitational problems. In Chapter 4, we present and illustrate a general algorithm. We begin with a description of the core part of our algorithm and develop further details in subsequent sections. This is followed by a description of the enhancements to the core algorithm in order to reduce CPU-time and a schematic that presents the different combinations for the algorithm. Numerical testing is done in Chapter 5 involving comparisons of combinations of different non-symplectic integrators, interpolation schemes and non-linear equation solvers on a collection of test problems. Overall conclusions are presented in Chapter 6, where we also provide directions for future research.

All computer programs are written in FORTRAN and we have used Matlab to analyse the results. Throughout the thesis we have performed simulations using a single compiler combination, namely, the ifort compiler with the -01 optimisation option.
There are two general ways to solve system (1.2.1) of second-order ordinary differential equations. One way is to solve the second-order ordinary differential equations directly by using numerical integrators such as Runge-Kutta-Nyström, Störmer, or extrapolation integrators. The other way is to convert system (1.2.1) into a system of first-order ordinary differential equations, and then use numerical integrators such as Runge-Kutta, Adams, or extrapolation integrators. In this chapter, in particular, we will survey $N$-body integrators that use the first approach. We will also measure the accuracy and the efficiency of different types of integrators. We perform numerical experiments for the different integrators applied to the Jovian problem over a long interval, as long as 100 million years, with the local error tolerance ranging from $10^{-16}$ to $10^{-08}$. 
2.1 Reference solution and errors

Unlike the Kepler problem, an analytical solution for the Jovian problem is not available. Therefore, numerical experiments using the Jovian problem require a reference solution \( y_{\text{ref}} \) in order to obtain an estimate of the error in the position and velocity. The reference solution has to be more accurate than the numerical solution. Since we plan to test the numerical integrators near the limit of double-precision arithmetic (\( 2 \times 10^{-16} \)), it is essential to use quadruple-precision arithmetic for the reference solution.

Different types of errors are discussed throughout this chapter. The global error is of major importance in the measurement of the quality of the numerical solution. We measure this global error in position and velocity. In addition we measure the relative error in energy and angular momentum, and the phase error.

For the total error in the system there are two main sources of error when an interpolation scheme is used: the integration error, which consists of truncation and round-off error, and the interpolation error. When performing accurate simulations, the round-off error can contribute significantly to the global error. For fixed-step-size schemes, Brouwer [4] showed that, if the step-size is smaller than a prescribed value, the round-off error for conserved quantities, such as total energy and angular momentum, grows as \( t^{\frac{1}{2}} \) and for other dynamical variables, such as the coordinates of particles, as \( t^2 \). This error growth is known as Brouwer’s law in the literature; see, for example, [28, 34]. In contrast, when the round-off error is systematic, the power laws become \( t \) and \( t^2 \), respectively.

First we define the types of errors used in this thesis. We use the notation \( y_{\text{true}}(t) \) to denote the true solution of (1.2.1) and \( y_{\text{num}}(t) \) to denote its approximate solution. The difference is monitored time wise by considering the global error in \( y \) as

\[
y_{\text{num}}(t) - y_{\text{true}}(t). \tag{2.1.1}
\]
The norm of the global error in $y$ at time $t$ is then

$$||y_{\text{num}}(t) - y_{\text{true}}(t)||_2.$$  \hspace{1cm} (2.1.2)

The calculation of the global error is discussed later. A wide range of integrators, for example, Runge-Kutta [37, 43], linear multistep [42], Runge-Kutta-Nyström [50], and Störmer [59] can be used to find $y_{\text{num}}(t)$ for $t \geq 0$. This leads to numerical solutions $y_n = y_{\text{num}}(t_n)$ and $y'_n = y'_{\text{num}}(t_n)$ at times $t_n = t_0 + nh$, $n = 1, 2, ...$, where the time-step $h$ can depend on $n$. The approximation over the continuous time interval can then be computed using (local) interpolation.

From one time-step to the next, the local problem is solved, which is defined as

$$u''_n = f(t, u_n), \quad u_n(t_{n-1}) = y_{n-1}, \quad u'_n(t_{n-1}) = y'_{n-1}, \quad t \in [t_{n-1}, t_n],$$  \hspace{1cm} (2.1.3)

where the function $u_n(t)$ is the true local solution on the $n^{th}$ interval. The initial conditions $y_{n-1}$ and $y'_{n-1}$ are the values of the approximate solution at the end of the $(n - 1)^{st}$ step. Hence, for the local problem the differential equations are the same as for the original problem, but the initial conditions depend upon the numerical solution. Since, in general, the approximation $y_{n-1}$ is not equal to the true solution $y_{\text{true}}(t_{n-1})$, there is a difference between the error made in this local step and the global error. The norm of the local error is defined as

$$||y_{\text{num}}(t) - u_n(t)||_2,$$  \hspace{1cm} (2.1.4)

where $t \in [t_{n-1}, t_n]$. In general, the global error cannot be calculated because the true solution is not known.

The estimated global error in the position over each time-step, $n$, is defined in terms of an accurate numerical solution (computed in quadruple precision), $y_{\text{ref}}(t)$, and an underlying polynomial approximation introduced on each step, $P_n(t)$. We then estimate global error
for $t \in [t_{n-1}, t_n]$ by evaluating $P_n(t) - y_{ref}(t)$ at $k$ evenly spaced sample points over the step. A more detailed description of $y_{ref}(t)$, $k$, $P_n(t)$, and the definition of the estimated global error in the position is presented later in this section.

The global error in the position for all $t \in [t_0, t_f]$, where $t_f$ is the pre-specified final value of $t$ in the integration, is more difficult to define and estimate, because we need a continuous approximation to $y(t)$, and not just the discrete values $y_1, y_2, ...$. We obtain the continuous approximation by first forming a polynomial $P_n(t)$ that approximates the local solution on the interval $[t_{n-1}, t_n]$. The numerical approximation is then the piecewise-defined function

$$
y_{num}(t) = \begin{cases} 
P_1(t), & t_0 \leq t \leq t_1, \\
P_2(t), & t_1 \leq t \leq t_2, \\ \vdots \\
P_n(t), & t_{n-1} \leq t \leq t_n, \\
\end{cases}$$

where we have assumed $P_{n-1}(t_{n-1}) = P_n(t_{n-1})$, an assumption that holds throughout the thesis. The corresponding definition of $y'_{num}(t)$ is

$$
y'_{num}(t) = \begin{cases} 
\hat{P}_1(t), & t_0 \leq t \leq t_1, \\
\hat{P}_2(t), & t_1 \leq t \leq t_2, \\ \vdots \\
\hat{P}_n(t), & t_{n-1} \leq t \leq t_n, \\
\end{cases}$$

where $\hat{P}_n(t)$ not necessarily the derivative of $P_n(t)$.

The polynomials $P_n(t)$ and $\hat{P}_n(t)$ are commonly called local interpolants and can be of many types. The main requirements of the local interpolants are that they be sufficiently
accurate and have sufficient continuity. These requirements along with the types of local interpolants we used in this thesis are discussed in detail in Chapter 3.

We write the norm of the estimated global error in the position at time $t$ as

$$E_r(t) = \| y_{num}(t) - y_{ref}(t) \|_2,$$  

and the norm of the estimated end-point global error as

$$E_{r,end}(t_f) = \| y_{num}(t_f) - y_{ref}(t_f) \|_2,$$  

where $t_f$ is the time at the end-point. The maximum of the norm of the estimated global error in position, loosely referred to as the maximum global error is defined as the maximum norm of $E_{r,max}(t)$ evaluated at $k$ evenly spaced sample points over each time-step. That is, we estimate the maximum global error by introducing an accurate approximation $y_{ref}(t)$ to $y(t)$, sampling $\| y_{num}(t) - y_{ref}(t) \|_2$ over each time-step, and using the maximum over all sampled values to be the estimate of the maximum of $\| y_{num}(t) - y(t) \|_2$.

In a similar way to that for the position, the norm of the estimated global error in the velocity at time $t$ and the maximum of this norm over the interval $[t_0, t_f]$ are defined as

$$E_v(t) = \| y'_{num}(t) - y'_{ref}(t) \|_2,$$

$$E_{v,max} = \max_{t \in [t_0, t_f]} \| y'_{num}(t) - y'_{ref}(t) \|_2,$$

where $y'_{num}(t)$ and $y'_{ref}(t)$ at any time $t$ are the vectors of the derivatives to the numerical and reference solutions, respectively.

We store the information, such as positions and times, in separate files. In a post-processing program, we estimate the maximum global error by sampling the norm at $k$ evenly spaced data points on each sub-interval $[t_{n-1}, t_n]$ with respect to the reference solution, which we obtain at these stored values of time by forcing the integrator to hit the
sample points, and then taking the maximum of these norms. To see the effect of different values of $k$ on the estimated maximum global error in position we performed experiments using the combination of $ERKN101217$ and the 23-stage interpolant, which are described in Sections 2.2.1.1 and 3.1.2. The integration was performed using the Jovian problem over one million years for the local error tolerances $10^{-08}, 10^{-10}, 10^{-12}, 10^{-14},$ and $10^{-16}$. We found that $k = 10$ led to an estimated maximum global error that was sufficiently accurate for our work; this is illustrated by the results in Table 2.1. The rows labelled $k = 50$ and $k = 100$ list the percentage changes in the estimated maximum global error when compared with the values in row labelled $k = 10$.

<table>
<thead>
<tr>
<th>$TOL$</th>
<th>$10^{-08}$</th>
<th>$10^{-10}$</th>
<th>$10^{-12}$</th>
<th>$10^{-14}$</th>
<th>$10^{-16}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 10$</td>
<td>$4.70 \times 10^{-1}$</td>
<td>$1.63 \times 10^{-3}$</td>
<td>$4.82 \times 10^{-5}$</td>
<td>$2.42 \times 10^{-5}$</td>
<td>$8.80 \times 10^{-6}$</td>
</tr>
<tr>
<td>$k = 50$</td>
<td>$0 %$</td>
<td>$3.4 \times 10^{-2} %$</td>
<td>$3.9 \times 10^{-3} %$</td>
<td>$1.0 \times 10^{-2} %$</td>
<td>$5.2 \times 10^{-5} %$</td>
</tr>
<tr>
<td>$k = 100$</td>
<td>$0 %$</td>
<td>$3.5 \times 10^{-2} %$</td>
<td>$3.9 \times 10^{-3} %$</td>
<td>$1.0 \times 10^{-2} %$</td>
<td>$4.2 \times 10^{-5} %$</td>
</tr>
</tbody>
</table>

Table 2.1: The values of estimated maximum global error using $ERKN101217$ with 23-stage interpolant obtained with different values of $k$ applied to the Jovian problem over one million years for the local error tolerances $10^{-08}, 10^{-10}, 10^{-12}, 10^{-14},$ and $10^{-16}$. The row labelled $k = 10$ shows errors, whereas the rows labelled $k = 50$ and $k = 100$ list the percentage changes.

We observe from Table 2.1 that the data for $TOL = 10^{-08}$ shows a percentage change of zero. In this case, the estimated maximum global error occurs at $t_f$ and this is possible with $N$-body simulations; if this occurs, the percentage difference between different values of $k$ would be zero. We repeated the same set of experiments with other combinations. For example, when using the integrator $ODEX2$ with its interpolant, which are described in Sections 2.2.3.1 and 3.1.3, an increase of $k$ from 10 to 100 changed the estimated maximum global error by not more than 1%. Throughout the remainder of this thesis, all maximum global errors are estimated by sampling at 10 evenly spaced data points on every time-step.
Appropriate combinations of integrators and interpolation schemes are essential to maintain accuracy; for more details on appropriate combinations, see Chapter 3. The experiments, for example, as reported in Table 2.3 as well as figures 2.1, 2.2, 2.4, and 2.5 are performed with local interpolation schemes based on definition of the maximum global error. Here, we have used the 12-stage, 23-stage, built-in interpolants, and quintic Hermite interpolation with integrators ERKN689, ERKN101217, ODEX2, and Störmer, respectively.

Physical systems often have conserved quantities, for example, the total energy $H$ or the total angular momentum $L$. Usually, these quantities will not be conserved exactly by the numerical solution and this deviation provides assessment about the accuracy of the solution. The total energy for a system of $N$ bodies interacting with one another through Newtonian forces is defined as

$$H(t) = \frac{1}{2} \sum_{i=1}^{N} m_i (v_{i,\text{num}}(t) \cdot v_{i,\text{num}}(t)) - \sum_{j=1}^{N-1} \sum_{i=j+1}^{N} G \frac{m_i m_j}{d_{ij}(t)},$$

where $G$ is the gravitational constant, $m_i$ the mass of the $i^{th}$ body, $v_{i,\text{num}}(t)$ the numerical approximation to velocity of the $i^{th}$ body (components $3i - 2$ to $3i$ of $y_{\text{num}}(t)$), and $d_{ij}(t) = ||r_{i,\text{num}}(t) - r_{j,\text{num}}(t)||_2$ is the distance between the $i^{th}$ and $j^{th}$ bodies. Here, $r_{i,\text{num}}(t)$ is formed by the components $3i-2$ to $3i$ of $y_{\text{num}}(t)$ and $r_{j,\text{num}}(t)$ by the components $3j-2$ to $3j$ of $y_{\text{num}}(t)$, which represent the numerical approximations to the positions of the $i^{th}$ and $j^{th}$ bodies, respectively.

The relative error in the energy can be defined as

$$H_{\text{rel}}(t) = \left| \frac{H_0 - H(t)}{H_0} \right|,$$

where $H_0$ is the total energy at the initial time $t = t_0$. We use

$$H_{\text{rel}}(t) = \left| \frac{GH_0 - GH(t)}{GH_0} \right|,$$
to calculate $H_{\text{rel}}(t)$, because the value of $\mu = Gm$ is usually known more accurately than $m$.

The total angular momentum $L(t)$ is defined as

$$L(t) = \sum_{i=1}^{N} m_i r_{i,\text{num}}(t) \times v_{i,\text{num}}(t).$$

We define the relative error of the angular momentum as

$$L_{\text{rel}}(t) = \frac{||L_0 - L(t)||_2}{||L_0||_2},$$

where $L_0$ is the angular momentum at the initial time $t = t_0$. A reference solution is not required to calculate $H_{\text{rel}}(t)$ and $L_{\text{rel}}(t)$, unlike the global errors in the position and velocity. Hence, fewer computing resources are needed to measure the performance of the integrators here. However, $H_{\text{rel}}(t)$ and $L_{\text{rel}}(t)$, being scalar quantities, impose only one constraint each on the error; if we look at the error in position or velocity then every component of $E_r(t)$ or $E_v(t)$ has to be small if we want a small error.

The phase error is the difference between the phase angle of the numerical solution and the reference solution. The cosine of the phase error for the $i^{th}$ body is defined by

$$\cos(\theta) = \frac{r_{i,\text{ref}} \cdot r_{i,\text{num}}}{||r_{i,\text{ref}}||_2 ||r_{i,\text{num}}||_2},$$

where $\theta$ is the angle between the numerical solution $r_{i,\text{num}}$ and the reference solution $r_{i,\text{ref}}$.

The maximum errors in the energy, angular momentum, and phase were estimated using the same sampling technique we used for the maximum global errors in the position and velocity.
2.2 Numerical methods and integrators

2.2.1 One step methods for first-order ODEs

One of the earliest one-step numerical methods is the famous Euler method, introduced by Euler in 1768. During the late 19th and early 20th century, German mathematicians Runge [53] and his successors Heun [37] and Kutta [43] developed the fundamentals of what are known as explicit Runge-Kutta methods; Kutta characterised the famous 4th-order classical Runge-Kutta method [43]. All these methods are easy to implement because of their explicit nature. Explicit Runge-Kutta methods evaluate the function $f$ at points on an interval $[t_{n-1}, t_n]$ and use additional information at those intermediate steps to obtain better approximations at the end of the time-step. All information at $t = t_{n-1}$ is then discarded before taking the next time-step. We now review integration methods for second-order ODEs.

2.2.1.1 Explicit Runge-Kutta-Nyström methods

Explicit Runge-Kutta-Nyström methods (ERKN) for solving (1.2.1) were introduced by E. J. Nyström in 1925 [50]. These methods calculate order-$q$ approximations $y_n$ and $y'_n$ to $y(t_n)$ and $y'(t_n)$ respectively, using the formulae

$$
 y_n = y_{n-1} + h y'_{n-1} + h^2 \sum_{j=1}^{s} b_j K_j,
$$

$$
 y'_n = y'_{n-1} + h \sum_{j=1}^{s} b'_j K_j,
$$

(2.2.1)

where

$$
 K_j = f(t_{n-1} + c_j h, y_{n-1} + c_j h y'_{n-1} + h^2 \sum_{l=1}^{j-1} a_{jl} K_l).
$$
The integer \( s \) is the number of stages and \( K_j, j = 1, \ldots, s \), are called the stages. The coefficients \( c_j, j = 1, \ldots, s \) are called the abscissas and the coefficients \( a_{jl}, j = 2, \ldots, s, l = 1, \ldots, j - 1 \) of the \( s \times s \) matrix \( A \) are called the interior weights. The coefficients \( b_j \) and \( \hat{b}_j \) (the prime in \( \hat{b}_j \) is used for notational convenience and does not stand for the derivative operator in this case) are called the exterior weights of the solution and derivative formulae, respectively. Explicit Runge-Kutta-Nyström methods reduce the computational cost compared to explicit Runge-Kutta methods applied to the equivalent system of first-order differential equations. For example, an order-five ERKN method requires only four function evaluations per time-step, whereas an explicit Runge-Kutta method of the same order requires at least six function evaluations [36].

The efficiency of an ERKN method depends upon the approach for controlling the error in the numerical approximations. One way of controlling the error is to use an adaptive step-size technique that permits control of the estimated local error. A pair of formulae of different orders is used in such a way that the function evaluations of the two methods are identical. This idea was first introduced by Merson [47] and further developments were made by England [18] and Fehlberg [22]. A variable-step-size method can be formed from (2.2.1) by calculating order-\( p \) approximations (\( p < q \)) \( \hat{y}_n \) and \( \hat{y}'_n \) to \( y(t_n) \) and \( y'(t_n) \), respectively, using the formulae

\[
\begin{align*}
\hat{y}_n &= y_{n-1} + h y'_{n-1} + h^2 \sum_{j=1}^{s} \hat{b}_j K_j, \\
\hat{y}'_n &= y'_{n-1} + h \sum_{j=1}^{s} \hat{b}'_j K_j.
\end{align*}
\]

(2.2.2)

The estimated local error of the approximations to \( y_n \) and \( y'_n \) are then

\[
\begin{align*}
y_n - \hat{y}_n &= h^2 \sum_{j=1}^{s} (b_j - \hat{b}_j) K_j \quad \text{and} \quad y'_n - \hat{y}'_n = h \sum_{j=1}^{s} (b'_j - \hat{b}'_j) K_j.
\end{align*}
\]

(2.2.3)

Here, \( K_j, c_j, s, \) and \( a_{jl} \) are the same as in system (2.2.1) and \( \hat{b}_j \) and \( \hat{b}'_j \) are called the exterior weights of the lower-order solution and derivative formulae, respectively.
The step-size is then adjusted so that \( \max\{||y_n - \hat{y}_n||_2, ||y'_n - \hat{y}'_n||_2\} \leq TOL \) on each step, where \( TOL \) is the local error tolerance.

The combination of \( p^{th} \)- and \( q^{th} \)-order formulae, where \( q > p \), is called a \( p-q \) pair. Fehlberg [21, 23] was the first who developed RKN pairs, and then other researchers added their algorithms to the RKN family; see, for example, [3, 15, 24, 38]. Many classes of explicit Runge-Kutta-Nyström pairs have been developed, see, for example, [11, 12, 17]. If the solution \( y_n \) is obtained using the \( p^{th} \)-order formula, then the pair is said to be implemented in lower-order mode. However, it is recommended for efficiency reasons and as we have assumed in (2.2.1) and (2.2.2) that the solution \( y_n \) be obtained using the \( q^{th} \)-order formula for the next step [19]. The pair operated in this fashion is said to be implemented in higher-order mode or local extrapolation. We assumed this when writing formulae (2.2.1) and (2.2.2). In this thesis, we have assumed the local extrapolation mode, that is, the numerical solution is advanced using the higher-order (\( q^{th} \)-order) formula.

We update the step-size according to the algorithm in Table 2.2, where we have assumed without loss of generality that the integration duration is in the direction of positive \( t \). Here, \( L_{e\text{max}} \) is \( \max\{||y_n - \hat{y}_n||_2, ||y'_n - \hat{y}'_n||_2\} \), \( h_{\text{max}} \) and \( h_{\text{min}} \) are the maximum and minimum allowable step-sizes, respectively; \( \alpha_{up} \) and \( \alpha_{dw} \) are the maximum and minimum allowable ratios of consecutive step-sizes, respectively; and \( \beta < 1 \) is a safety-factor. The algorithm keeps trying smaller step-sizes until \( L_{e\text{max}} \leq TOL \) or \( h = h_{\text{min}} \). When the inequality is satisfied, the integrator moves on to the next step. If \( h = h_{\text{min}} \) and the estimated local error cannot be made small enough, the integration exits. We observed that for long-term simulations (as in our case), the choice of initial step-size has little impact on the results and we used \( (TOL)^{1+1} \) as the initial step-size for all our simulations.

Fehlberg [23] and then Dormand and Prince [15] popularized the idea of ERK methods with the property FSAL (first same as last), where the vector \( b^T = [b_1, ..., b_s]^T \), \( c_s = 1 \), and the last row of the matrix \( A \) are equal. This means that the last stage can be used as the first stage of the next step. Dormand and Prince developed an ERK pair of orders
if \((Le_{\text{max}} \leq TOL)\) then

% Accept the step;

if \((Le_{\text{max}} = 0)\) then
  \[ h_{\text{new}} = \alpha_f h; \]
else
  % The step-size for the next step is calculated as:
  \[ h_{\text{new}} = \min \left( \alpha_f h, \beta h \left( \frac{TOL}{Le_{\text{max}}} \right)^{1/p} \right); \]
end if
\[ h = \min(h_{\text{max}}, h_{\text{new}}); \]
\[ h = \max(h_{\text{min}}, h); \]

elseif \((h \leq h_{\text{min}})\) then
  % The integration failed. Set an error flag, print an error message if required and exit the integration;
else
  % Update integration step-size and try again:
  \[ h_{\text{new}} = \min \left( \alpha_f h, \beta h \left( \frac{TOL}{Le_{\text{max}}} \right)^{1/p} \right); \]
end if
\[ h = \max(h_{\text{min}}, h_{\text{new}}); \]

end if

Table 2.2: Our algorithm for controlling the step-size sequence, where without loss of generality we have assumed \(h > 0\).

5 and 4 in 1980. Then Hairer and Wanner wrote a code, known as \textit{DOPRI54}, with seven stages, but, being FSAL, this method has effectively six stages [13]. The idea of FSAL for ERK methods extends to ERKN methods; see, for example, [12].

In this thesis we are using the following non-symplectic, variable-step-size ERKN integrators:

- Integrator \textit{ERKN689} is a nine stage, 6-8 FSAL pair [12];
- Integrator \textit{ERKN101217} is a seventeen stage, 10-12 non-FSAL pair [12].
2.2.1.2 Round-off error control for ERKN methods

In this thesis, we perform experiments with the local tolerance close to the machine precision \((2.2 \times 10^{-16})\). Therefore, we investigate the possibility of reducing the growth of round-off error in the explicit Runge-Kutta-Nyström methods using the technique known as compensated summation [55]. The idea of compensated summation is based on estimating the dominant contribution term of the round-off error. To explain the round-off error control technique, we consider the following solution formula

\[
y_n = y_{n-1} + h y'_{n-1} + h^2 \sum_{j=1}^{s} b_j K_j.
\]

Equation (2.2.4) contains three types of errors; the integration error already in \(y_n\) from the previous time-steps, the round-off error in the formation of \(h y'_{n-1}\) and \(h^2 \sum_{j=1}^{s} b_j K_j\), and the round-off error caused by adding terms on the right-hand side of (2.2.4). If the integration time-step is small then \(h y'_{n-1} + h^2 \sum_{j=1}^{s} b_j K_j\) is small compared to \(y_{n-1}\). Hence, the round-off error will be dominated by adding \(h y'_{n-1} + h^2 \sum_{j=1}^{s} b_j K_j\) to \(y_{n-1}\).

In each time-step we estimate the round-off error caused by adding \(h y'_{n-1} + h^2 \sum_{j=1}^{s} b_j K_j\) to \(y_{n-1}\) and then update the solution as follows. First, we calculate

\[
\tau = h y'_{n-1} + h^2 \sum_{j=1}^{s} b_j K_j - \delta,
\]

where \(\delta\) is the estimated round-off error on the previous time-step (at the start of the integration \(\delta = 0\)). Since \(h y'_{n-1} + h^2 \sum_{j=1}^{s} b_j K_j\) and \(\delta\) are small compared to \(y_{n-1}\), the error caused in the formation of \(\tau\) is negligible. The solution is then updated to

\[
Y_n = y_{n-1} + \tau,
\]

and the estimated round-off error for the time-step is calculated as

\[
\delta = Y_n - y_n - \tau.
\]
The solution is then updated as $y_n = Y_n$. The derivative formula in system (2.2.1) also uses the same concept to control the round-off error.

<table>
<thead>
<tr>
<th>$TOL$</th>
<th>$ERKN689$</th>
<th>$ERKN101217$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$E_{r,\text{max}}$</td>
<td>$E_{v,\text{max}}$</td>
</tr>
<tr>
<td>$10^{-08}$</td>
<td>4.370×10$^{-2}$</td>
<td>0.02%</td>
</tr>
<tr>
<td>$10^{-10}$</td>
<td>1.30×10$^{-4}$</td>
<td>-0.84%</td>
</tr>
<tr>
<td>$10^{-12}$</td>
<td>1.70×10$^{-6}$</td>
<td>71.8%</td>
</tr>
<tr>
<td>$10^{-14}$</td>
<td>9.74×10$^{-7}$</td>
<td>86.0%</td>
</tr>
<tr>
<td>$10^{-16}$</td>
<td>3.10×10$^{-6}$</td>
<td>70.8%</td>
</tr>
</tbody>
</table>

Table 2.3: The values of $E_{r,\text{max}}$ and $E_{v,\text{max}}$ for $ERKN689$ and $ERKN101217$ obtained with and without round-off error control applied to the Jovian problem over one million years for the local error tolerances $10^{-08}$, $10^{-10}$, $10^{-12}$, $10^{-14}$, $10^{-16}$.

To investigate the effectiveness of compensated summation with ERKN methods we solved the Jovian problem described in Section 1.4.2. The integration was performed over $10^6$ years using $TOL = 10^{-2i}$, for $i = 4, 5, \ldots, 8$. Table 2.3 shows the values of $E_{r,\text{max}}$ and $E_{v,\text{max}}$, calculated based on definition of the maximum global error for the explicit Runge-Kutta-Nyström integrators $ERKN689$ and $ERKN101217$. The column labelled With lists $E_{r,\text{max}}$ and $E_{v,\text{max}}$ when compensated summation was used in the integration. The column labelled Without lists the percentage change in $E_{r,\text{max}}$ and $E_{v,\text{max}}$ when compensated summation is not used.

With one exception for $ERKN689$, the values for $E_{r,\text{max}}$ and $E_{v,\text{max}}$ with round-off error control are always less than $E_{r,\text{max}}$ and $E_{v,\text{max}}$ without round-off error control. The exception is for $TOL = 10^{-10}$, where the values of $E_{r,\text{max}}$ and $E_{v,\text{max}}$ in the column Without are close to zero and insignificant compared with values for smaller tolerances. The maximum difference was observed for $TOL = 10^{-14}$. Here, the values for $E_{r,\text{max}}$ and $E_{v,\text{max}}$ obtained without round-off error control were approximately 86% and 84% more than those obtained with round-off error control. For $ERKN101217$, except for
$TOL = 10^{-08}$, we found that the round-off error control technique is not very effective, the errors in the position and velocity obtained with round-off error control are not always less than $E_{r,\text{max}}$ and $E_{v,\text{max}}$ without round-off error control. This could be because the average time-step for $\text{ERKN101217}$ over $10^6$ years is large. For example, with $TOL = 10^{-14}$, $\text{ERKN101217}$ takes a time-step of approximately 144 days on average over $10^6$ years, and hence, the assumption that $hy_n' + h^2 \sum_{j=1}^{s} b_j K_j$ is small relative to $y_{n-1}$ is invalid. Therefore, for $\text{ERKN101217}$, using $TOL = 10^{-2i}$, with $i = 5, 6, 7, 8$, it is not recommended to use the round-off error control technique.

### 2.2.2 Störmer methods

Störmer methods are an important class of methods for solving systems of second-order differential equations. Introduced by Störmer [59], the methods have long been utilised for accurate long-term simulations of the Solar System [30]. Störmer developed a simple method by adding the Taylor series for $y(t_n + h)$ and $y(t_n - h)$, and eliminating the derivative terms; see, for example, [36, p. 462]. The resulting method for the solution is given by

$$y_{n+1} = 2y_n - y_{n-1} + h^2 f_n,$$

where $f_n = y_n''$. Higher-order methods are obtained by using differences that involve values of $f$ from the end of the previous steps, for example,

$$y_{n+1} = 2y_n - y_{n-1} + h^2 f_n + \frac{h^2}{12} \left( \frac{59}{20} f_n - \frac{176}{20} f_{n-1} + \frac{194}{20} f_{n-2} - \frac{96}{20} f_{n-3} + \frac{19}{20} f_{n-4} \right).$$

The Störmer method of order $p$ for the position and velocity can be written as

$$y_{n+1} = 2y_n - y_{n-1} + h^2 \sum_{j=0}^{p-1} \alpha_j f_{n-j},$$

$$y_{n+1}^\prime = \frac{1}{h} (y_n - y_{n-1}) + h \sum_{j=0}^{p-1} \beta_j f_{n-j},$$

(2.2.6)
where the coefficients $\alpha_j$ and $\beta_j$ can be obtained using generating functions. The starting values $y_1, y_2, y_3, \ldots, y_{p-1}$ are normally computed using a one-step method.

The Störmer method can also be expressed in terms of a backward-difference interpolation polynomial passing through the points $(t_j, f_j)$, for $j = n - p + 1, \ldots, n$. This gives

$$y_{n+1} = 2y_n - y_{n-1} + h^2 \sum_{j=0}^{p-1} \gamma_j \nabla^j f_n,$$

$$y'_n = \frac{1}{h}(y_n - y_{n-1}) + h \sum_{j=0}^{p-1} \sigma_j \nabla^j f_n,$$

where, $\nabla^0 f_n = f_n$ and $\nabla^{j+1} f_n = \nabla^j f_n - \nabla^j f_{n-1}, \ j = 0, 1, \ldots$, and the coefficients are

$$\gamma_j = (-1)^j \int_0^1 (1 - s) \left( \left( \begin{array}{c} -s \\ j \end{array} \right) + \left( \begin{array}{c} s \\ j \end{array} \right) \right) ds,$$

$$\sigma_j = (-1)^j \left( \int_{-1}^0 (1 + s) \left( \begin{array}{c} -s \\ j \end{array} \right) ds + \int_0^1 \left( \begin{array}{c} -s \\ j \end{array} \right) ds \right).$$

The coefficients $\gamma_j$ and $\sigma_j$ can also be obtained from a recurrence relation. For example, $\gamma_j$ can be calculated using

$$\gamma_j = 1 - \frac{2}{3} d_2 \gamma_{j-1} - \ldots - \frac{2}{j+2} d_{j+1} \gamma_0,$$

where, $\gamma_0 = 1$ and $d_j = 1 + \frac{1}{2} + \ldots + \frac{1}{j}$.

Störmer methods can be implemented with a variable step-size; see, for example, [7].

Grazier [26] recommended an order-13, fixed-step-size Störmer method that uses backward differences in summed form, summing from the highest to the lowest differences. The test results in [28] for simulations of the Sun, Jupiter, Saturn, Uranus and Neptune in double precision showed that the error in the energy and phase error grows as $t^{1/2}$ and $t^{3/2}$, respectively, to within numerical uncertainty when the step-size is $(\frac{1}{1024})$-th (4.1 days) of Jupiter’s orbital period. This choice of step-size ensures that the local truncation
error of the Störmer method is well below machine precision. In this thesis, we use the fixed-step-size Störmer method of order 13 and refer to it as the $\tilde{S}$-$13$ integrator.

### 2.2.3 Extrapolation integrators

Especially in applied mathematics and engineering, one frequently has to deal with sequences and series. One serious drawback to the effective use of sequences or series is that they often converge slowly. Therefore, convergence acceleration methods, based on the very natural idea of extrapolation, have been used for many years. Extrapolation provides a strong means of accelerating the convergence of solutions that arise from discretization methods, and also have strong connections with, for example, projection methods, continued functions and Padé approximations.

In the early 20th century Richardson [52] introduced the discretization technique now known as Richardson extrapolation to improve the rate of convergence of a sequence, but was unable to obtain a suitable starting step for the central difference discretization. As a consequence, extrapolation methods were little used until the appearance of the pioneering code of Bulirsch and Stoer [5] in 1966. Later, successful extrapolation codes were developed by Deuflhard and his collaborators; we refer to [10] for a comprehensive overview of relatively recent extrapolation methods for ordinary differential equations.

#### 2.2.3.1 ODEX2 integrator

For the direct numerical solution of systems of second-order ordinary differential equations, Hairer, Nørsett and Wanner [35] developed an extrapolation code $ODEX2$ based upon the explicit midpoint rule with order selection and step-size control. The $ODEX2$ integrator is efficient for all tolerances, especially for high precision.
2.3 Step-size variation

Here, we investigate the step-size variation for the variable-step-size integrators ERKN689, ERKN101217, and ODEX2 applied to the Jovian problem. The eccentricities of the orbits of the Jovian planets are no more than 0.1 and there are no close-encounters between the planets. Therefore, the variable-step-size integrators should require small step-size variation. Table 2.4 shows the step-size variation for the above integrators applied to the Jovian problem over one million years for the local error tolerances in the range $10^{-16}$ to $10^{-08}$. The columns $h_{mn}$ and $h_{mx}$ list the percentage variation in the minimum and maximum step-sizes relative to the mean step-size. For example, $h_{mn}$ is calculated as

$$h_{mn} = 100 \left( \frac{h_{mn} - \bar{h}}{\bar{h}} \right),$$

where, $h_{mn}$ is the smallest step-size used and $\bar{h}$ the mean step-size. For these results, we considered the on-scale step-sizes only by ignoring the first few step-sizes in the transient region near $t = 0$ as well as the final step-size.

<table>
<thead>
<tr>
<th>TOL</th>
<th>ERKN689</th>
<th>ERKN101217</th>
<th>ODEX2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^{-08}$</td>
<td>$-17%$</td>
<td>$84%$</td>
<td>$-20%$</td>
</tr>
<tr>
<td>$10^{-10}$</td>
<td>$-17%$</td>
<td>$99%$</td>
<td>$-20%$</td>
</tr>
<tr>
<td>$10^{-12}$</td>
<td>$-18%$</td>
<td>$115%$</td>
<td>$-19%$</td>
</tr>
<tr>
<td>$10^{-14}$</td>
<td>$-18%$</td>
<td>$134%$</td>
<td>$-20%$</td>
</tr>
<tr>
<td>$10^{-16}$</td>
<td>$-18%$</td>
<td>$152%$</td>
<td>$-34%$</td>
</tr>
</tbody>
</table>

Table 2.4: The step-size variation for the variable-step-size integrators ERKN689, ERKN101217, and ODEX2 applied to the Jovian problem over one million years, with the local error tolerance TOL as specified in the first column.

The step-size variation depends upon both the integrator and the tolerance chosen, and ranges from approximately $-34\%$ to $152\%$. The largest variation between the maximum and minimum step-sizes occurs for ERKN689 with TOL = $10^{-16}$, where it is a factor of three, with $h$ ranging from $0.89\bar{h}$ to $2.52\bar{h}$. For the purpose of our work, we regard this variation as small. This small step-size variation enables us to add a fixed-step-size
integrator $\tilde{S}$-13 (Störmer of order 13) in this thesis. We also observe from Table 2.4 that
the variation in the step-size for $ERKN101217$ and $ODEX2$ is small for the tolerances we
used.

To see the effect of round-off error, we also performed integrations with $TOL = 10^{-14}$ in quadruple-precision arithmetic. The percentage variations of $h_{mn}$ and $h_{mx}$ were approximately $-18\%$ and $133\%$ for $ERKN689$, $-20\%$ and $21\%$ for $ERKN101217$, and $-30\%$ and $21\%$ for $ODEX2$. The step-size variations obtained in quadruple-precision arithmetic agrees reasonably well with Table 2.4 and we conclude the round-off error is not significant with $TOL = 10^{-14}$.

2.4 Numerical experiments for long-term simulation

First we consider the error growth in the position and velocity using the variable-step-size integrators $ODEX2$, $ERKN689$, and $ERKN101217$. We obtained reference solutions in quadruple-precision. For example, one reference solution is obtained using $ERKN101217$ with $TOL = 10^{-18}$. To justify this particular choice for the reference solution, we integrated the Jovian problem using the $ERKN101217$ integrator with $TOL = 10^{-20}$. The maximum difference between the positions and velocities of these two solutions is no more than $4.61 \times 10^{-13}$. We also integrated the Jovian problem in quadruple-precision with the tolerance $TOL = 10^{-18}$, but using the $ERKN689$ integrator and found that the maximum difference between the solution and that for $ERKN101217$ with $TOL = 10^{-18}$ was no more than $5.11 \times 10^{-13}$. This suggests that the $ERKN101217$ integrator with $TOL = 10^{-18}$ is sufficiently accurate to obtain the reference solution.

Figure 2.1 illustrates the unweighted $L_2$-norm of the estimated maximum global error in the position as a function of tolerance for the three variable-step-size integrators $ODEX2$, $ERKN689$, and $ERKN101217$ applied to the Jovian problem over one million years. We chose the range $10^{-16}$ to $10^{-08}$ for the local error tolerance, because $10^{-16}$ is
close to machine precision in double-precision arithmetic and tolerances greater than $10^{-08}$ lead to errors that are too large to be meaningful. For the estimation of the maximum global error, we evaluate the position at 10 evenly spaced points on every time-step (care has been taken to make efficient use of storage) using different integrators. For most integrations, the maximum global error occurs at the end point of the integration. The result for ODEX2 is a maximum global error that ranges from $8.0 \times 10^{-5}$ to $1.1 \times 10^{2}$.

We observe that the smallest and largest values of the maximum global error occur for $TOL = 10^{-16}$ and $TOL = 10^{-08}$, respectively. The graph for ODEX2 exhibits three phases: In the middle phase, with $TOL \in [10^{-15}, 10^{-12}]$ the round-off error does not affect the global error. Round-off error has an effect for $TOL \leq 10^{-16}$, which we further investigated by using smaller tolerances, i.e, $TOL = 10^{-16-k}$, for $k = .2, .4, .6, .8$, and 1. As $TOL$ decreases below $10^{-16}$, the maximum global error starts to increase again, which indicates the influence of the round-off error. The phase for $TOL > 10^{-11}$ shows a global error of approximately $10^{1}$ AU, which is the diameter of Jupiter’s orbit. The integrator
still finds the orbit but the phase error could be as much as 180°. We evaluated the end-point phase error using the formula described in Section 2.1 and found that it is approximately 172°.

Let us now consider the integrator \textit{ERKN101217} in Figure 2.1. Here, the maximum global error ranges from 8.7 × 10^{-6} to 4.7 × 10^{-1}. The smallest maximum global error is again obtained with \textit{TOL} = 10^{-16} and the largest with \textit{TOL} = 10^{-08}. We observe that from \textit{TOL} = 10^{-11} to 10^{-16} there is little change in the maximum global error. Therefore, if the smallest maximum global error is required then \textit{TOL} = 10^{-16} should be used, but otherwise, a small sacrifice in accuracy could potentially save a significant amount of CPU-time.

The integrator \textit{ERKN689} has the maximum global error ranging from 9.7 × 10^{-7} to 4.4 × 10^{-2}, with the smallest maximum global error obtained at \textit{TOL} = 10^{-14} and the largest at \textit{TOL} = 10^{-08}. Therefore, nothing is gained by decreasing the tolerance from 10^{-14} to 10^{-16}. The smallest maximum global error at \textit{TOL} = 10^{-14} is an indicator that the round-off error affects the global error when using tolerances between 10^{-14} and 10^{-16}. To measure the possible effect of round-off error, we performed experiments in quadruple-precision. We obtained the maximum global error in the position as a function of tolerance for the local error tolerances 10^{-16} to 10^{-10} using \textit{ERKN689} and \textit{ERKN101217}. We observed that both curves are straight and maintain a difference of about 1.5 orders of magnitude for each tolerance. In addition, the graph is not bending up for \textit{ERKN689} using the small tolerance of 10^{-16}. This confirms the effect of round-off error in the double-precision arithmetic.

We conclude from Figure 2.1 that, for local error tolerances ranging from \textit{TOL} = 10^{-16} to 10^{-08}, the integrator \textit{ERKN689} is the most accurate and \textit{ODEX2} is the least accurate integrator.

Let us now compare this performance with the $\bar{S}$-$13$ integrator. Figure 2.2 contains the graphs of the norm of the estimated global error in the position as a function of $t$ for the
Jovian problem using the integrators $\tilde{S}$-13, ODEX2, ERKN689, and ERKN101217 over $10^6$ years. The integration with the $\tilde{S}$-13 integrator was performed in double-precision using a step-size of four days.

We performed two sets of experiments. For the first set, we chose $TOL$ so that the maximum global error in the position over $10^6$ years was approximately $10^{-4}$. This required using $TOL = 10^{-16}, 10^{-10},$ and $10^{-11}$ for ODEX2, ERKN689, and ERKN101217, respectively. Figure 2.2 contains the graphs of the norm of the estimated global error as a function of $t$. For small $t$, ERKN689 and ERKN10121 are more accurate than the other two integrators, but there is a crossover approximately at $5 \times 10^4$ years. We see in Figure 2.2 that the three variable-step-size integrators have similar end-point global errors and $\tilde{S}$-13 has an end-point global error that is almost one order of magnitude smaller.

![Figure 2.2: The error growth in position for the integrators $\tilde{S}$-13, ODEX2, ERKN689, and ERKN101217 (the graphs are smoothed using a filter command with WS = 50; filtering is explained later in the section) applied to the Jovian problem over one million years. The selection of local error tolerances is subject to a prescribed accuracy (maximum global error $\approx 10^{-4}$), labeled by ERKN689-G and ERKN101217-G, and best observed (smallest maximum global error) accuracy, labeled by ERKN689-M, S-13-M, and ERKN101217-M.](image-url)
To gain insight into the error growth depicted in Figure 2.2, we used unweighted linear least squares to fit a power law $\alpha t^\beta$ to the error growth in position. We found that the integration error for the integrators ERKN689 and ERKN101217 grows approximately as $t^2$ (quadratic growth), while for ODEX2 and $\bar{S}$-13 it is approximately $t^{3/2}$; see also Table 3.7. The error growth for ODEX2 is slower than expected. Therefore, we repeated the integrations for ODEX2 by increasing the tolerance from $TOL = 10^{-16}$ to $10^{-15}$ and $10^{-14}$; then we observe approximately the quadratic error growth for $TOL = 10^{-14}$ and $10^{-15}$. We have no explanation for the slow growth for ODEX2 when $TOL = 10^{-16}$.

The second set of experiments for the integrators $\bar{S}$-13, ERKN689, and ERKN101217, labelled S-13-M, ERKN689-M and ERKN101217-M in Figure 2.2, respectively, are done with the tolerance chosen to give the smallest maximum global error. This meant using $TOL = 10^{-14}$ and $10^{-16}$ for ERKN689 and ERKN101217, respectively. For $\bar{S}$-13, we performed experiments with step-sizes of 4, 10, 15, 20, 25, and 30 days; see Table 2.5. We observe that $\bar{S}$-13 achieves best accuracy with a step-size of approximately 10 days. This step-size is larger than four days recommended by Grazier et al. [26], because the orbital eccentricity of the bodies is smaller. The performance of the ODEX2 integrator at the prescribed accuracy, as shown in Figure 2.1, is also at the best observed accuracy for the local error tolerance of $10^{-16}$. When performed at the best observed accuracy, there is no longer a crossover of the $\bar{S}$-13 integrator with the integrators ERKN689 and ERKN101217. At the end of $10^6$ years of integration, ERKN689 achieves the best accuracy and ERKN101217 achieves the next best accuracy.

<table>
<thead>
<tr>
<th>Days</th>
<th>4</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. global error in position</td>
<td>$1.96 \times 10^{-5}$</td>
<td>$1.08 \times 10^{-5}$</td>
<td>$1.89 \times 10^{-5}$</td>
<td>$3.95 \times 10^{-5}$</td>
<td>$6.23 \times 10^{-5}$</td>
<td>$1.05 \times 10^{-4}$</td>
</tr>
</tbody>
</table>

Table 2.5: The maximum global error as a function of the step-size for the fixed-step-size integrator $\bar{S}$-13, applied to the Jovian problem over one million years.

Some of the plots in these kinds of experiments have high-frequency oscillations. In
order to smooth that data, the filter command in Matlab was employed with a window size of 50. The appropriate choice of window size is important. We have experimented (using the experiments illustrated in Figure 2.2 with the exclusion of those labelled S-13M, ERKN689-M, and ERKN101217-M) for values of window sizes, 1, 10, 20, and 50 as shown in Figure 2.3. Figure 2.3(a) shows the result without filtering (WS=1). There are enough oscillations of sufficient amplitude that it is difficult to distinguish the graphs. If the window size is small, as shown in Figure 2.3(b) (WS = 10) then quite a few oscillations are still there and it is not clear which of the integrators is being crossed. A window size of 50 seems to be a sensible value for this set of experiments. As is shown in Figure 2.3(d), it is quite clear that $\tilde{S}$-13 crosses only the integrators ERKN689 and ERKN101217. Filtering can complicate the interpretation of results for the first WS points, but this effect can be removed by omitting the first WS points.

Figure 2.3: Experiments with the window size for the Matlab function filter. The window size is set to 1, 10, 20 and 50 in plots (a) to (d), respectively.
Let us now consider the accuracy of the integrators using the relative error in the energy and angular momentum. Figure 2.4 shows the error growth in the energy for the Jovian problem. The integration has been performed in double-precision over $10^6$ years using the same local error tolerances and integrators as for the results shown in Figure 2.2. For this set of experiments, we used the filter command in Matlab with a larger window size of $WS = 100$, because the oscillations were more pronounced than the set of experiments shown in Figure 2.2. These experiments are performed to obtain the relative error in energy on each time-step using appropriate combinations of integrators and interpolation schemes. We use a local interpolation scheme and sample the solution at 10 evenly spaced data points, from which we determine the local maxima for the relative error in energy.

Figure 2.4: The error growth in the energy for the four integrators $\tilde{S}-13$, ODEX2, ERKN689, and ERKN101217 (filter command used with $WS = 100$) applied to the Jovian problem over one million years. The local tolerance was chosen to give the smallest maximum global error (the ‘-M’ graphs), and a maximum global error $\approx 10^{-4}$ (the ‘-G’ graphs).

For the integrators ERKN689 and ERKN101217 (labeled by ERKN689-G and ERKN101217-G in Figure 2.4, respectively) we observe an error growth proportional to $t$ in energy and angular momentum (not shown here). This is similar to the results depicted
in Figure 2.2. For $ODEX2$, the error growth shows oscillations. The integrations were repeated for $ODEX2$ by increasing the tolerance from $TOL = 10^{-16}$ to $10^{-15}$ and $10^{-14}$, which causes the oscillations to disappear for $TOL = 10^{-15}$ and $10^{-14}$. This indicates that round-off error is the cause of the oscillations. Approximately linear error growth in energy and angular momentum was observed particularly for $ODEX2$ with $TOL = 10^{-14}$. As in Figure 2.2, the integrators $ODEX2$ and $\bar{S}-13$ with step-sizes of four days, cross the integrators $ERKN689$ and $ERKN101217$. However, this crossover for the relative error in energy occurs at a smaller $t$ than for the global error in position. We observe from Figure 2.4 that, for the relative error in energy, the integrator $ERKN689$ using $TOL = 10^{-14}$ (labeled by ERKN689-M) again achieves the smallest global error.

![Graph showing efficiency plots](image)

**Figure 2.5:** Efficiency plots showing the number $N_{fev}$ of function evaluations against the $L_2$-norm of the maximum global error in position, obtained for the variable-step-size integrators $ERKN689$, $ERKN101217$, and $ODEX2$, applied to the Jovian problem over one million years with $TOL$ ranging from $10^{-16}$ to $10^{-08}$.

Let us now consider the efficiency of the integrators, which is the amount of work to attain a prescribed accuracy. One way of measuring the amount of work is to count the number of function evaluations. Figure 2.5 shows graphs of the number of function evaluations
against the maximum global error in position, obtained for the variable-step-size integrators ERKN689, ERKN101217, and ODEX2, and applied to the Jovian problem over one million years with \( TOL \) ranging from \( 10^{-16} \) to \( 10^{-10} \). As described in Figure 2.1 the best accuracy for ERKN689 is achieved at \( TOL = 10^{-14} \), which needs approximately 1.64 and 3.37 times more function evaluations than ERKN101217 and ODEX2, respectively. If we consider tolerances such that all three integrators achieve approximately the same accuracy \( 10^{-4} \) then ERKN101217 is the most efficient, because it uses the least number of function evaluations. The integrator ERKN689 is approximately 2.35 and ODEX2 approximately 2.60 times more expensive than ERKN101217. Our conclusion changes slightly when the maximum global error permitted is increased from \( 10^{-4} \) to approximately \( 10^{-3} \) and \( 10^{-2} \). The integrator ERKN101217 again achieves the best accuracy compared to the integrators ODEX2 and ERKN689. For an accuracy of approximately \( 10^{-3} \), the integrator ODEX2 is approximately 1.60 and ERKN689 approximately 2.08 times more expensive than ERKN101217. For an accuracy of approximately \( 10^{-2} \), the integrator ODEX2 is approximately 1.60 and ERKN689 approximately 1.85 times more expensive than ERKN101217.

We also investigated the CPU-time taken by the same variable-step-size integrators applied to the Jovian problem over one million years with local error tolerances in the range from \( 10^{-16} \) to \( 10^{-08} \). For \( TOL = 10^{-08} \), we found that ODEX2 and ERKN101217 take almost the same CPU-time, but the maximum global error for ERKN101217 is approximately three orders of magnitude smaller than that for ODEX2. For the same tolerance, ERKN689 is almost two times more expensive than ERKN101217 and ODEX2, but has approximately one and four orders of magnitude better accuracy, respectively. For maximum global errors of approximately \( 10^{-4} \), \( 10^{-3} \), and \( 10^{-2} \), ERKN101217 takes the least CPU-time. Hence, the integrator ERKN101217 is the cheapest option.

For the given range of tolerances from \( 10^{-16} \) to \( 10^{-10} \), we found that ERKN689 achieves a maximum global error (at \( TOL = 10^{-14} \)), which is approximately one and two orders of magnitude better than the maximum global error achieved by ERKN101217 and
$ODEX_2$, respectively. At the same point, $ERKN689$ is almost 2.68 and four times more expensive than $ERKN101217$ and $ODEX_2$, respectively. These results clearly illustrate a trade-off between accuracy and efficiency.
The primary focus of this chapter is to measure the accuracy and efficiency of different types of interpolation schemes for a variety of gravitational problems. The continuous approximations obtained by Hermite interpolation schemes and interpolants for ODEX2 and ERKN integrators are discussed in this chapter. The interpolants are a special example of an interpolation scheme, which produce an approximation that is continuous across one step and across the complete interval of integration. The interpolants are quite expensive in comparison with the other interpolation schemes. Therefore, one of the objectives of this chapter is to investigate the possibilities of replacing the interpolants of
certain integrators by other interpolation schemes, perhaps at the cost of a little bit of accuracy.

The numerical experiments are performed for the different integrators together with one-step, two-step, and three-step Hermite interpolation schemes, as well as the interpolants. We integrate over a short time interval of length $2\pi$ and several time intervals as long as 100 million years. For the graphs, we used the Matlab filter command with a window size of $WS = 100$.

### 3.1 Continuous approximations

As discussed in Chapter 2, the well-known one-step methods for the numerical solution of a system of ordinary differential equations are normally formulated such that numerical approximations are produced on mesh points $t_0 < t_1 < t_2 < \ldots < t_f$, which are determined by the step-size selection strategy of these numerical methods. As soon as a numerical approximation of $y_{true}(t)$ is required where $t$ is not at one of the mesh points, the numerical approximation must be extended to a continuous approximation.

Continuous approximations are needed when detecting close-encounters between a massive body and a test particle. It is possible that a close-encounter occurs within the time-step and not at the end points. Such a close-encounter would not be detected if we only have numerical approximations at the end points. One way of detecting such close-encounters is to generate numerical solutions within the integration time-step by using a far smaller time-step when the test particle is near the massive body compared with when it is far from the massive body. In practice, this approach would be inefficient because all bodies, including those that are not undergoing close-encounters, would be integrated with the smaller time-step, resulting in a possibly tremendous increase in CPU-time. Thus, an alternative approach is needed to avoid this inefficiency. The numerical approximation at the mesh points may be extended to a continuous approximation, which provides the numerical solution at any point $t$, for $t$ within the time-step interval. Pioneering work on continuous approximations for Runge-Kutta methods have appeared in the literature [39],
Figure 3.1: Classification of the errors when using a numerical integrator together with an interpolation polynomial.

and also for explicit methods [54]. This work was extended by others; see, for example, [2, 14–16, 20, 35, 61, 62].

Figure 3.1 shows a local segment of the numerical solution $y_{\text{num}}$ over the time interval $[t_{n-1}, t_n]$. This solution is obtained by interpolation over data calculated from the local problem (2.1.3). The interpolation polynomial goes through the points at the beginning and the end of the time-step $[t_{n-1}, t_n]$. The local solution to (2.1.3) has a local error that is 0 at $t = t_{n-1}$. In contrast, the global error is not 0 for any $t \in [t_{n-1}, t_n]$. There are two contributions to the global error. First, the interpolation polynomial is not formed using true data. Rather, it is using an approximation at the end of the time-step. Second, even if it is true data then there is an interpolation error at the intermediate times $t \in [t_{n-1}, t_n]$. If we denote the estimated global error in an interpolated value on $[t_{n-1}, t_n]$ by $E_{r,int}(t)$, then the relative error with respect to the estimated end-point global error is

$$R_{\text{err}}(t) = \frac{\text{Estimated global error in an interpolated value on } [t_{n-1}, t_n]}{\text{Estimated global error in the numerical solution at } t_n}.$$  

$$= \max_{t \in [t_{n-1}, t_n]} \frac{E_{r,int}(t)}{E_{r,end}(t_n)}.$$  

(3.1.1)
If the integration error dominates the interpolation error then the relative error $R_{err}(t)$ tends to 1 as $t \rightarrow \infty$.

3.1.1 Hermite interpolation schemes

Hermite interpolation uses derivative and function values and is named after Charles Hermite (1822 - 1901). We used four schemes: one-step (cubic and quintic Hermite), two-step and three-step. The cubic Hermite interpolation polynomial is of degree 3, while the quintic, two-step and three-step Hermite interpolation polynomials are of degrees 5, 8 and 11, respectively. The interpolation schemes are derived using a Newton divided difference approach, which is described in Section 3.1.1.1. There is a second approach, which we call the direct approach, that is frequently used by other researchers, for example, see [32]. This approach is particularly suited for cubic and quintic Hermite interpolation schemes, and we describe it in Sections 3.1.1.2 and 3.1.1.3.

3.1.1.1 Newton divided difference approach

To determine the interpolating polynomial $P_m(t)$ for the $m$ points $(t_j, y_j)$, $j = 0, 1, ..., m - 1$, using the Newton divided difference (NDD) approach, we write $P_m(t)$ as

$$P_m(t) = a_0 + a_1(t - t_0) + a_2(t - t_0)(t - t_1) + ... + a_m(t - t_0)...(t - t_{m-1}),$$

where the $a$’s are are calculated from the divided differences. The $i^{th}$ divided difference can be calculated using

$$f[t_0, t_1, ..., t_i] = \frac{f[t_1, ..., t_i] - f[t_0, ..., t_{i-1}]}{(t_i - t_0)},$$

see Table 3.1. We now discuss how the NDD must be modified when derivative values are used. Let us consider the first-order differences in Table 3.1. For example, if $t_1 = t_0$ then we have

$$\lim_{t_1 \rightarrow t_0} f[t_0, t_1] = f'(t_0).$$
Similarly, for the second-order differences in Table 3.1, if, for example, $t_2 = t_0$ then we find
\[
\lim_{t_2 \to t_0} f[t_0, t_1, t_2] = \frac{f''(t_0)}{2}.
\]

<table>
<thead>
<tr>
<th>$j$</th>
<th>$t_j$</th>
<th>$f(t_j)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$t_0$</td>
<td>$y_0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f[t_0, t_1]$</td>
</tr>
<tr>
<td>1</td>
<td>$t_1$</td>
<td>$y_1$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f[t_0, t_1, t_2]$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f[t_1, t_2]$</td>
</tr>
<tr>
<td>2</td>
<td>$t_2$</td>
<td>$y_2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f[t_1, t_2, t_3]$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f[t_2, t_3]$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f[t_1, t_2, t_3, t_4]$</td>
</tr>
<tr>
<td>3</td>
<td>$t_3$</td>
<td>$y_3$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f[t_2, t_3, t_4]$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f[t_3, t_4]$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f[t_1, t_2, t_3, t_4, t_5]$</td>
</tr>
<tr>
<td>4</td>
<td>$t_4$</td>
<td>$y_4$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f[t_3, t_4, t_5]$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$f[t_2, t_3, t_4, t_5]$</td>
</tr>
<tr>
<td>5</td>
<td>$t_5$</td>
<td>$y_5$</td>
</tr>
</tbody>
</table>

Table 3.1: An illustrative Newton divided difference table.

Hence, for Hermite interpolation schemes we can use the NDD approach if the derivatives replace the corresponding divided differences.

### 3.1.1.2 Cubic Hermite interpolation

In this section and the next, we describe the direct approach for obtaining the cubic and quintic Hermite polynomials. The cubic Hermite interpolation polynomial $P_3(t)$ for the time-step from $t = t_{n-1}$ to $t = t_n$ interpolates the data $(t_{n-i}, y_{n-i})$ and $(t_{n-i}, y'_{n-i})$ at time $t_{n-i}$, for $i = 1$ and 0. In the direct approach, the cubic Hermite interpolation polynomial is written as
\[
P_3(t) = a_0 y_{n-1} + a_1 H y'_{n-1} + a_2 y_n + a_3 H y'_{n},
\]
where,

\[
\begin{align*}
    a_0 &= (\tau - 1)^2(2\tau + 1), \\
    a_1 &= (\tau - 1)^2\tau, \\
    a_2 &= \tau^2(3 - 2\tau), \\
    a_3 &= \tau^2(\tau - 1),
\end{align*}
\]

and \( H = t_n - t_{n-1} \) with \( \tau = (t - t_{n-1})/H \). Since the values of \( y \) and \( y' \) at both ends of each step are interpolated, the piecewise defined approximation \( y_{num}(t) \) formed from the cubic Hermite polynomial is continuous and has a continuous first derivative.

### 3.1.1.3 Quintic Hermite interpolation

The quintic Hermite interpolation polynomial \( P_5(t) \) for the time-step from \( t = t_{n-1} \) to \( t = t_n \) interpolates the data \((t_{n-i}, y_{n-i}), (t_{n-i}, y'_{n-i}),\) and \((t_{n-i}, y''_{n-i})\) at time \( t_{n-i} \), for \( i = 1 \) and 0. As for cubic Hermite interpolation, the quintic Hermite interpolation polynomial can be derived using a direct approach and written as

\[
P_5(t) = a_0y_{n-1} + a_1Hy'_{n-1} + a_2H^2y''_{n-1} + a_3y_n + a_4Hy'_n + a_5H^2y''_n,
\]

where,

\[
\begin{align*}
    a_0 &= (1 - \tau)^3(6\tau^2 + 3\tau + 1), \\
    a_1 &= (1 - \tau)^3\tau(3\tau + 1), \\
    a_2 &= (1 - \tau)^3\tau^2/2, \\
    a_3 &= \tau^3(6\tau^2 - 15\tau + 10), \\
    a_4 &= \tau^3(1 - \tau)(3\tau - 4), \\
    a_5 &= \tau^3(\tau - 1)^2/2,
\end{align*}
\]
and \( H = t_n - t_{n-1} \) with \( \tau = (t - t_{n-1})/H \), as before. Since the values of \( y, y', \) and \( y'' \) at both ends of each step are interpolated, the piecewise defined approximation \( y_{num}(t) \) formed from the quintic Hermite polynomial is continuous and has continuous first and second derivatives.

### 3.1.1.4 Two-step Hermite interpolation polynomial

The two-step Hermite interpolation polynomial \( P_8(t) \) for the two-time-steps from \( t = t_{n-2} \) to \( t = t_n \) interpolates the data \((t_{n-i}, y_{n-i}), (t_{n-i}, y'_{n-i}), \) and \((t_{n-i}, y''_{n-i})\) at time \( t_{n-i} \), with \( i = 2, 1, \) and 0. The two-step Hermite interpolation polynomial \( P_8(t) \) can then be written in Horner’s nested multiplication form as

\[
P_8(t) = D_{11} + (t - t_{n-2})(D_{22} + (t - t_{n-2})(D_{33} + (t - t_{n-2})(D_{44} + (t - t_{n-1})(D_{55} + (t - t_{n-1}) (D_{66} + (t - t_{n-1})(D_{77} + (t - t_{n})(D_{88} + (t - t_{n})D_{99})))))))))
\]

where the coefficients \( D_{ii}, i = 1, \ldots, 9, \) are obtained using a NDD table. Since the values of \( y, y', \) and \( y'' \) at both ends of each step are interpolated, the piecewise defined approximation \( y_{num}(t) \) formed from the two-step Hermite polynomial is continuous and has continuous first and second derivatives.

### 3.1.1.5 Three-step Hermite interpolation polynomial

Similarly, the three-step Hermite interpolation polynomial interpolates the data \((t_{n-i}, y_{n-i}), (t_{n-i}, y'_{n-i}), \) and \((t_{n-i}, y''_{n-i})\) at time \( t_{n-i} \), for \( i = 3, 2, 1, \) and 0. Hence, it is the degree-11 polynomial \( P_{11}(t) \) defined over a three-time-step from \( t = t_{n-3} \) to \( t = t_n \). Using Horner’s nested multiplication form, we can write \( P_{11}(t) \) as

\[
P_{11} = D_{11} + (t - t_{n-3})(D_{22} + (t - t_{n-3})(D_{33} + (t - t_{n-3})(D_{44} + (t - t_{n-2})(D_{55} + (t - t_{n-2}) (D_{66} + (t - t_{n-2})(D_{77} + (t - t_{n-1})(D_{88} + (t - t_{n-1})D_{99} + (t - t_{n-1}) (D_{1010} + (t - t_{n})D_{1212})))))))))))
\]
As for the two-step Hermite interpolation polynomial, the coefficients $D_{ii}, i = 1, \ldots, 12,$ of $P_{11}(t)$ are obtained using NDD. Since the values of $y, y', \text{ and } y''$ at both ends of each step are interpolated, the piecewise defined approximation $y_{\text{num}}(t)$ formed from the three-step Hermite polynomial is continuous and has continuous first and second derivatives.

We compared the maximum error in position and the CPU-time for $P_3$ and $P_5$ evaluated using NDD and the direct approach. The comparison was done for one period of Kepler problem for eccentricities of 0.05 to 0.9 (see Section 3.2.1 for more details on the experiment), and the Jovian problem.

For the two-body problem, no significant differences in the maximum error as CPU-time were observed between these two approaches. For the Jovian problem, the direct approach takes approximately half the CPU-time of the NDD approach. The coefficients of the polynomial for the NDD approach depend on the components of the solution vector. For the direct approach the coefficients are independent of the components, so they can be used as a vector to approximate polynomials and that will save CPU-time.

In the rest of the thesis, cubic and quintic Hermite interpolation schemes are implemented using the direct approach. For two-step and three-step Hermite interpolation schemes we implemented the NDD approach, because it is really difficult to find the coefficients for the direct approach.

### 3.1.2 Continuous approximation with embedded Runge-Kutta-Nyström methods

Dormand and Prince [15] and then Baker et al. [2] developed continuous approximation for embedded Runge-Kutta-Nyström pairs, in which a third RKN process of order $p^*$ is used to approximate the solutions, $y(t_{n-1+\alpha})$ and $y'(t_{n-1+\alpha}),$ where $t_{n-1+\alpha} = t_{n-1} + \alpha h_{n-1}$ with $\alpha$ typically in $(0, 1).$ These approximations, which are usually called interpolants
and are denoted by \( y^*_n \) and \( y'_{n-1} + \alpha \), can be expressed as

\[
y^*_n - 1 = y_n - 1 + \alpha h_n - 1 + \alpha^2 h_n^2 \sum_{i=1}^{s^*} b^*_i(\alpha) k_i^*,
\]

\[
y'_{n-1} + \alpha = y'_{n-1} + \alpha h_n - 1 + \sum_{i=1}^{s^*} b^*_i(\alpha) k_i^*
\]

where,

\[
k_i^* = f(t_n - 1 + c_i^* \alpha h_n - 1, y_n - 1 + c_i^* \alpha h_n - 1 y'_n - 1 + \alpha^2 h_n^2 \sum_{j=1}^{i-1} a_{ij}^* k_j^*), \quad i = 1, \ldots, s^*.
\]

If \( s^* = s \), then no extra function evaluations are required to compute \( y^*_n \) and \( y'_{n-1} + \alpha \).

As before, the prime in \( b^*_i \) is purely notational and does not involve the differentiation operator. The approximation to \( y_{num}(t) \) is continuous and has continuous first and second derivatives, because it interpolates \( y, y' \), and \( y'' \) at both ends of each step. For ERKN methods the derivative interpolants can be derived separately, and not as the derivative to the solution interpolants themselves.

For \( ERKN101217 \), we used three existing interpolants: a 23-stage interpolant with \( p^* = 10 \), a 26-stage interpolant with \( p^* = 11 \), and a 29-stage interpolant with \( p^* = 12 \). The coefficients for these interpolants are not tabulated in this thesis but are freely available on-line [2]. For \( ERKN689 \), we used an 8th-order interpolant with 12 stages. The coefficients for the interpolant for \( y \) and \( y' \) of \( ERKN689 \) were provided by Sharp (private communication) and are tabulated in Appendix A.

### 3.1.3 Continuous approximation with ODEX2

The \( ODEX2 \) integrator has as solution interpolant. We added a derivative interpolant by differentiating the solution interpolant. The solution interpolant provides an approximation to the \( i^{th} \) component of the solution at time \( t \). The polynomial \( P_i(\alpha) \) for the solution
interpolant can be written as

\[ P_\mu(\alpha) = y_{(i)} + \alpha y_{(M+i)} + \alpha(1 - \alpha)y_{(2M+i)} + \alpha^2(1 - \alpha)y_{(3M+i)} + \alpha^3(1 - \alpha)^2y_{(4M+i)} + \alpha^2(1 - \alpha)^3y_{(5M+i)} + \alpha - \alpha^2\sum_{j=0}^{k} \frac{(\alpha_1)^j}{j!}y_{(M(5+j)+i)}, \]

where \( \alpha_1 = (t - t_{n-1})/h - 0.5 \) and \( M \) is the number of ordinary differential equations. The degree of \( P_\mu(\alpha) \) is \( \mu + 4 \), with \(-1 \leq \mu \leq 2k\), where \( k \) is related to the integration order.

The polynomial used for the continuous approximation of the derivative provides an approximation to the \( i^{th} \) component of the derivative at time \( t \). The order of this derivative interpolant is one less than \( P_\mu(\alpha) \) and can be written as

\[ P'_\mu(\alpha) = \frac{1}{h}[y_{(M+i)} + (1 - 2\alpha)y_{(2M+i)} + (2\alpha - 3\alpha^2)y_{(3M+i)} + \alpha^2(1 - \alpha)(3 - 5\alpha)y_{(4M+i)} + \alpha(1 - \alpha)^2(2 - 5\alpha)y_{(5M+i)} + (\alpha - \alpha^2)^3(1 - 2\alpha) \sum_{j=0}^{k} \frac{(\alpha_1)^j}{j!}y_{(M(5+j)+i)} + 3(\alpha - \alpha^2)^2(1 - 2\alpha) \sum_{j=0}^{k} \frac{(\alpha_1)^j}{j!}y_{(M(3+j)+i)}]. \]

### 3.1.4 Computational cost of interpolation schemes

Let us now consider the CPU-time by looking at individual interpolation schemes. Our expectation, at least for the interpolation schemes, is that the CPU-time is proportional to the number of multiplications. If one interpolation scheme uses twice as many multiplications then the CPU-time is expected to be twice as large. There will not be many divisions, and the number of subtractions and additions is typically proportional to the number of multiplications. Normally, when timing a program, an overhead is introduced. Therefore, care has been taken not to include such overheads in the final results. We also checked reproducibility of the results and observed a maximum variation of not more than 2.5%.
As discussed earlier, there are two different approaches to form interpolation schemes. Here, the experiments are performed using a direct approach for cubic and quintic Hermite interpolation, and the Newton divided difference approach for 2-step and 3-step Hermite interpolation schemes. In most cases, interpolation schemes are split into two subroutines, one for finding the coefficients and one for evaluating the polynomials. For \textit{ERKN689} and \textit{ERKN101217}, with the interpolants we have additional stage derivatives (function evaluations). Overall, we have three different groups of interpolation schemes:

(a) For cubic and quintic Hermite interpolation schemes, we evaluate the coefficients of the polynomial, which are independent of the components, and the polynomial as one subroutine.

(b) For 2-step and 3-step Hermite interpolation schemes, we have two subroutines:

(i) The calculation of the coefficients by forming a Newton divided difference table;

(ii) The evaluation of the polynomial.

(c) For the interpolants, we have three subroutines:

(i) The evaluation of the coefficients $b^*$;

(ii) The evaluation of the additional stage derivatives;

(iii) The evaluation of the polynomial using (i) and (ii).

For \textit{ODEX2}, the pieces of information required to form the interpolant are considered part of the integration, and we only consider the evaluation of the polynomial; see Table 3.4.

Since the coefficients of the polynomials for cubic and quintic interpolations are independent of the components, the experiments for these interpolation schemes are performed as one unit. As can be seen from the formulae in Sections 3.1.1.2 and 3.1.1.3, the quintic Hermite interpolation scheme uses approximately 93% more multiplications than cubic Hermite interpolation when applied to the Jovian problem. When we did our experiment, we found that the quintic Hermite interpolation scheme uses approximately 96% more CPU-time than cubic Hermite interpolation, which is in good agreement with the expected value.
Table 3.2: The CPU-time in seconds for evaluating the stage derivatives (without the cost of additional function evaluations) for ERKN689 and ERKN101217 applied to the Jovian problem.

<table>
<thead>
<tr>
<th>Integrator</th>
<th>ERKN689</th>
<th>ERKN101217</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>9-stage</td>
<td>17-stage</td>
</tr>
<tr>
<td>CPU-Time</td>
<td>$6.99 \times 10^{-06}$</td>
<td>$1.48 \times 10^{-05}$</td>
</tr>
</tbody>
</table>

Table 3.2 shows the CPU-time for finding the stage derivatives of the pairs (without the cost of additional stage derivatives) used in ERKN689 and ERKN101217 when applied to the Jovian problem. With ERKN689 we use the property FSAL (first same as last), so that we need only 8 derivative evaluations per step. Similarly, the 12-stage interpolant has effectively 11 stage derivatives. Observe from Table 3.2 that the average CPU-time consumed per stage is approximately $8.74 \times 10^{-07}$ and $8.71 \times 10^{-07}$ for ERKN689 and ERKN101217, respectively. Therefore, the expected CPU-time for ERKN689 with a 12-stage interpolant is approximately $9.61 \times 10^{-06}$. For ERKN101217, the expected CPU-time for finding coefficients is approximately $2.00 \times 10^{-05}$, $2.26 \times 10^{-05}$, and $2.52 \times 10^{-05}$ with 23-stage, 26-stage, and 29-stage interpolants, respectively.

<table>
<thead>
<tr>
<th>Interpolation Polynomial</th>
<th>ERKN689</th>
<th>ERKN101217</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2-step</td>
<td>3-step</td>
</tr>
<tr>
<td>CPU-Time</td>
<td>$2.83 \times 10^{-6}$</td>
<td>$5.54 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

Table 3.3: The CPU-time in seconds for finding the coefficients of the interpolation schemes and evaluating all the stage derivatives of the interpolants applied to the Jovian problem.

Table 3.3 gives the CPU time needed to find the coefficients of the interpolation schemes and evaluate all the derivatives for the interpolants when solving the Jovian problem. We observe that all values in Table 3.3 have reasonably good agreement with the prescribed values for CPU-time. Note also that the 3-step Hermite interpolation scheme in Table
3.3 uses approximately 96% more multiplications than the 2-step Hermite interpolation scheme, which is reasonably well matched by our finding of 93%.

<table>
<thead>
<tr>
<th>Interpolation</th>
<th>2-step</th>
<th>3-step</th>
<th>ERKN68</th>
<th>ERKN101217</th>
<th>ODEX2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU-Time</td>
<td>$6.7 \times 10^{-7}$</td>
<td>$1.18 \times 10^{-6}$</td>
<td>$2.97 \times 10^{-7}$</td>
<td>$4.42 \times 10^{-7}$</td>
<td>$4.96 \times 10^{-7}$</td>
</tr>
</tbody>
</table>

Table 3.4: The CPU-time in seconds for evaluating the position and velocity polynomials using different interpolation polynomials applied to the Jovian problem.

Table 3.4 shows the CPU-time for evaluating the position and velocity components using the different interpolation polynomials. The 3-step interpolation scheme uses approximately 76% more CPU-time than the 2-step interpolation, which is again in agreement with the CPU-times observed in Table 3.3. Similarly, the difference in CPU-time between the 23-stage and 29-stage interpolants is twice the difference between the 23-stage and 26-stage interpolants which is in good agreement with the difference observed in Table 3.3.

<table>
<thead>
<tr>
<th>Interpolation</th>
<th>Cubic</th>
<th>Quintic</th>
<th>2-step</th>
<th>3-step</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU-Time</td>
<td>$3.832 \times 10^{-7}$</td>
<td>$4.960 \times 10^{-7}$</td>
<td>$1.111 \times 10^{-6}$</td>
<td>$1.574 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interpolation</th>
<th>ERKN689</th>
<th>ERKN101217</th>
<th>ODEX2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Polynomial</td>
<td>12-stage</td>
<td>23-stage</td>
<td>26-stage</td>
</tr>
<tr>
<td>CPU-Time</td>
<td>$8.400 \times 10^{-7}$</td>
<td>$1.638 \times 10^{-6}$</td>
<td>$1.916 \times 10^{-6}$</td>
</tr>
</tbody>
</table>

Table 3.5: The CPU-time in seconds for evaluating the acceleration for one asteroid using different interpolation polynomials.

Table 3.5 shows the cost of evaluating the acceleration for one asteroid, which consists of the costs of searching the work array, of evaluating the polynomial for the position of the planets, and of evaluating the acceleration term. We can use either bisection or linear search for searching the work array. As discussed in Chapter 4 in the description of the
main algorithm, we integrate the planets over a big interval and then go back to integrate the asteroids individually. We used linear searching because found it is more efficient than bisection, because you pretty much know what the sub-interval is. Therefore, the next time-step for the asteroid is typically in the same or next sub-interval, because the time-step for the asteroid is not going to be larger than for the planets.

3.2 Numerical experiments for long-term simulations

In this section, we examine the error growth in the position, velocity and conserved quantities for the Kepler and Jovian problems over short- and long-term integrations. The experiments for short-term integrations are performed using four different types of interpolation schemes applied to the Kepler problem over the interval of $2\pi$. The experiments for long-term integrations are performed for the Jovian problem using four different integrators and nine different types of interpolation schemes over one million years of integration.

3.2.1 Kepler problem with different eccentricities

The error in the position and velocity of the Kepler problem is given by the $L_2$-norm

$$E_r(t) = ||r_{\text{num}}(t) - r_{\text{true}}(t)||_2,$$

$$E_v(t) = ||v_{\text{num}}(t) - v_{\text{true}}(t)||_2,$$

where $r_{\text{num}}(t)$ and $r_{\text{true}}(t)$ are the vectors of the numerical and true solutions, and $v_{\text{num}}(t)$ and $v_{\text{true}}(t)$ are the vectors of the derivatives to the numerical and true solutions, respectively. These are the same definitions as given in Section 2.1, but with different notation, i.e., for the two-body problem $y_{\text{num}} = r_{\text{num}}$ and $y'_{\text{num}} = v_{\text{num}}$. The solution to the Kepler problem is periodic with period $2\pi$. We do not have to calculate the reference solution, so the Kepler problem is well suited for testing the accuracy of integration over a short
time interval. This assumes the step-size is chosen so that \( t = 2\pi \) is hit exactly.

The graphs in Figure 3.2 are for experiments performed with the cubic, quintic, two-step, and three-step Hermite interpolation schemes applied to the Kepler problem over the interval \([0, 2\pi]\) for eccentricities in the range \([0.05, 0.9]\); note that, in reality, planets and test particles do not have eccentricity 0, and we used 0.05 as an approximate upper bound for the eccentricities of the Jovian planets. The selection of these interpolation schemes is motivated by the fact that they can be used with all the integrators described in this thesis. The interpolants, on the other hand, are related to specific integrators; for example, the 12-stage interpolant can only be used with the \( ERKN689 \) integrator. For the experiments shown in Figure 3.2, the interval of integration is subdivided into 30 evenly spaced sub-intervals; experiments with different numbers of sub-intervals are recorded in Table 3.6. We then evaluate the position and velocity at 10 evenly spaced points on each sub-interval using different interpolation schemes. Note that we also tested with up to 100 sample points and observed a variation in the error of not more than 1%.
The information, such as, positions, velocities, and times, are saved in separate files. In a post-processing step, we then calculate the errors in the positions and velocities with respect to the analytical solution that we obtain at the stored values of time. The velocity polynomials for all these interpolation schemes are obtained by differentiating their corresponding position polynomials.

From Figure 3.2, we observe a clear pattern; as the eccentricity increases, the maximum error in the position also increases. The variation in the error is understandable, because the error depends upon the eccentricity. To illustrate this fact, recall from Chapter 1 that the analytical solution to the Kepler problem is given by

\[(y_1(t), y_2(t)) = [\cos(\eta) - e, \sqrt{1 - e^2} \sin(\eta)]^T, \quad (3.2.1)\]

where \(\eta\) is the eccentric anomaly satisfying \(t = \eta - e \sin(\eta)\) and the interpolation error, for example, for the \(y_1\)-component of this solution can be written as \(ah^{b+1}y_{(p+1)\xi}^{(p+1)!}(\xi)^p\). The first three derivatives of \(y_1\) are

\[
\begin{align*}
\frac{dy_1}{dt} &= -\frac{\sin(\eta)}{1 - e \cos(\eta)}, \\
\frac{d^2y_1}{dt^2} &= -\frac{e - \cos(\eta)}{(1 - e \cos(\eta))^3}, \\
\frac{d^3y_1}{dt^3} &= \frac{\sin(\eta)(1 + 2e \cos(\eta) - 3e^2)}{(1 - e \cos(\eta))^5}.
\end{align*}
\]

It is clear that these and all subsequent derivatives are expected to involve the factor \(1/(1 - e \cos(\eta))\). Since the minimum value of \(|1 - e \cos(\eta)|\) gets smaller and smaller as \(e\) increases, it is expected that the error increases as \(e\) increases; a similar argument holds for the \(y_2\)-component. Indeed, for all interpolation schemes the minimum error in the position occurs at eccentricity 0.05 and the maximum error at eccentricity 0.9 in Figure 3.2. We also observe in Figure 3.2 that, for small eccentricity like \(e = 0.05\), the difference in the errors between consecutive interpolation schemes is approximately two orders of magnitude. As \(e\) increases to 1, this difference decreases and all four errors in Figure 3.2 appear to converge. We also computed the error in the velocity and found that it is nearly
two orders of magnitude larger than the error in the position. These experiments were also performed in quadruple-precision, but there was hardly any difference between the estimated errors obtained in double- and quadruple-precision. For example, using a 3-step interpolation scheme with eccentricity 0.05 and 0.9, the differences between the estimated errors in the position obtained in double- and quadruple-precision are $4.40 \times 10^{-15}$ and $1.67 \times 10^{-14}$, respectively. We conclude that the interpolation schemes are not affected a great deal by the round-off error when using 30 evenly spaced sub-intervals.

As mentioned earlier, the same sets of experiments described in Figure 3.2, were also done with different numbers of sub-intervals. We experimented with 17, 79, 255, and 1080 evenly spaced sub-intervals over the interval $[0, 2\pi]$. The associated errors for $e = 0.05$ are shown in Table 3.6. This particular selection of the number of sub-intervals is due to the fact that we wish to maintain the best observed accuracy of the integrators $ODEX2$, $ERKN101217$, $ERKN689$, and $\tilde{S}$-13; see Figure 2.1 and note that we use a time-step of

<table>
<thead>
<tr>
<th>$N_{\text{sub}}$</th>
<th>Cubic</th>
<th>Quintic</th>
<th>2-step</th>
<th>3-step</th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>$0.71 \times 10^{-6}$</td>
<td>$0.19 \times 10^{-8}$</td>
<td>$0.87 \times 10^{-11}$</td>
<td>$0.21 \times 10^{-12}$</td>
</tr>
<tr>
<td>79</td>
<td>$0.15 \times 10^{-8}$</td>
<td>$0.19 \times 10^{-12}$</td>
<td>$0.56 \times 10^{-16}$</td>
<td>$0.55 \times 10^{-16}$</td>
</tr>
<tr>
<td>255</td>
<td>$0.14 \times 10^{-10}$</td>
<td>$0.18 \times 10^{-15}$</td>
<td>$0.55 \times 10^{-16}$</td>
<td>-</td>
</tr>
<tr>
<td>1080</td>
<td>$0.44 \times 10^{-13}$</td>
<td>$0.56 \times 10^{-16}$</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.6: The maximum global error in position for eccentricity 0.05 attained by different interpolation schemes applied to the Kepler problem over the interval $[0, 2\pi]$ with four choices of numbers of evenly spaced sub-intervals. The dash means the combination is not used.

4 days for $\tilde{S}$-13. For example, the $ODEX2$ integrator applied to the Jovian problem achieves best accuracy using tolerance $10^{-16}$ and an average time-step of approximately 260 days over one million years. Since Jupiter’s orbital period is approximately 4320 Earth days, a time-step of 260 days gives approximately 17 steps.

The results in Table 3.6 show reasonably good agreement with the expected values
calculated from the orders of the polynomial, discounting the possible increase in round-off error from using a higher-order interpolation scheme and a large number of sub-intervals. For example, using the cubic Hermite interpolation scheme, and going from 17 to 79 sub-intervals, the expected value is \((17/79)^4 \times (0.71 \times 10^{-06}) \approx 1.52 \times 10^{-09}\) which has very good agreement with the value \(0.15 \times 10^{-08}\) mentioned in Table 3.6.

From Table 3.6 we find that the accuracy for a given interpolation scheme improves if the number of sub-intervals increases. We also deduce from Table 3.6 that it makes no sense to use any of the four interpolation schemes with \textit{ODEX2} (at least for the planets) if the required maximum global error is \(10^{-15}\) and 17 sub-intervals are used. For 79 sub-intervals (used for \textit{ERKN101217}) only the 2-step and 3-step Hermite interpolation schemes achieve the required accuracy. Similarly, for \textit{ERKN689}, the quintic and 2-step interpolation schemes achieve the required accuracy, whereas for the \textit{S-13} integrator, only the quintic Hermite interpolation scheme does.

Except for large intervals of integration, the Kepler problem closely resembles the Jovian problem; because of the weak interaction between the planets, the Jovian problem can be considered as five two-body problems for short intervals. Hence, it is no surprise that these same combinations of integrators and interpolation schemes, which achieved the set accuracy \(10^{-15}\) in Table 3.6, also turn out to achieve the best accuracies when we perform \(N\)-body simulations; this is discussed in detail in Chapter 5.

To test the analysis that led to the definition of the error \(R_{err} (3.1.1)\), we integrate the Jovian problem over one million years using all the integrators and interpolation schemes described so far; for a summary of the different combinations of integrators and interpolation schemes, we refer to Figures 4.1 and 4.2 in Chapter 4. On every accepted integration time-step, we calculate the \(L_2\)-norm of the relative error in energy and angular momentum at 10 evenly spaced values of time. The maximum of these 10 error values is taken as the maximum error (care has been taken to make efficient use of storage) on the integration time-step. The \textit{S-13} integrator uses a step-size of four days, and we set the tolerances to \(10^{-16}, 10^{-10}, \) and \(10^{-11}\) for \textit{ODEX2}, \textit{ERKN689}, and \textit{ERKN101217}. 
respectively (the variation of tolerance is subject to achieving a maximum global error of $10^{-4}$, see Figure 2.1).

![Graph showing the ratio of the maximum relative error in energy at the interior points to the endpoint (of a step) relative error in energy for the $\tilde{S}$-13 integrator using cubic and quintic Hermite interpolations (the Matlab filter command with WS = 100 was used) applied to the Jovian problem over one million years.](image)

Figure 3.3: The ratio of the maximum relative error in energy at the interior points to the endpoint (of a step) relative error in energy for the $\tilde{S}$-13 integrator using cubic and quintic Hermite interpolations (the Matlab filter command with WS = 100 was used) applied to the Jovian problem over one million years.

Figure 3.3 depicts the ratio of the maximum relative error in energy at the interior points to the endpoint relative error in energy for the $\tilde{S}$-13 integrator using cubic and quintic Hermite interpolation schemes. We observe from the plot for the cubic Hermite interpolation polynomial that $R_{err}$ appears bounded from above by $5 \times 10^4$ for small $t$ and then gradually decreases to no more than $6 \times 10^2$ at $10^6$ years. The large value of $R_{err}$ may be due to the fact that the interpolation polynomial used for the velocity components is only of order 2. The plot for the quintic Hermite interpolation polynomial shows that the relative error in energy is approximately 1 for the entire interval of integration. This verifies the result of Grazier et al. [32] that, when the time-step is chosen so that the $\tilde{S}$-13 methods satisfy Brouwer’s law, one-step quintic Hermite interpolation is sufficiently accurate. A similar behaviour with cubic and quintic polynomials is observed for the
relative error in angular momentum.

Figure 3.4 represents the ratio of the maximum relative error in energy at the interior points to the endpoint relative error in energy for ERKN101217 using cubic, quintic, two-step and three-step interpolation polynomials, which are of degrees 3, 5, 8 and 11, respectively; and 3 interpolants of 23, 26, and 29 stages, having orders 10, 11, and 12, respectively. We observe for the low-order interpolation schemes, i.e., cubic, quintic, and two-step, that $R_{\text{err}}$ in energy does not decrease to 1 at the end of $10^6$ years. With all three interpolants the $R_{\text{err}}$ is approximately 1 for the entire interval of integration. We also observe that the 3-step Hermite interpolation polynomial achieves the same accuracy (with $R_{\text{err}}$ in energy approximately 1) at the end of the integration, although initially, it is dominated by the interpolation error. Therefore, until about $10^4$ years, it is better to use one of the interpolants if the smallest error is required. However, if the required
integration time is large enough, the integration error starts dominating and then it is better to switch to a 3-step interpolation polynomial because this is going to save on CPU-time and does not sacrifice accuracy.

We also performed experiments measuring $R_{\text{err}}$ for the energy and angular momentum for $ERKN689$ using cubic, quintic, and two-step Hermite interpolation schemes, and together with its 12-stage interpolant; and for $ODEX2$ using cubic, quintic, two-step and three-step interpolation schemes, and together with the interpolant. For $ERKN689$, the two-step interpolation scheme and the 12-stage interpolant achieved a ratio of approximately 1. For $ODEX2$, the only continuous approximation that achieved the same accuracy was the interpolant and its derivative that comes with $ODEX2$.

To observe the behaviour of conserved quantities like energy and angular momentum using different interpolation schemes with different integrators, we performed many experiments. All these experiments were done for the Jovian problem over up to $10^8$ years. One of the strong observations about all of these experiments that applies except when using the quintic Hermite interpolation scheme with the $\bar{S}$-$13$ integrator is the following: if a low-order interpolation scheme is used with a high-order integrator then the interpolation error at a particular time will dominate the integration error and the total error will not increase with $t$. This is because the interpolation error does not grow with time. For example, if a 12-th order integrator is used with cubic Hermite interpolation, then the total error is given by

$$\text{Total error} = Ct^{3/2}h^{12} + Dh^4$$

$$\cong Dh^4, \quad C, D = \text{Constant},$$

provided $t$ is not too large and $C$ is not significantly larger than $D$. On the other hand, if the interpolation polynomial and the integrator are of the same order, then the total error will increase with the passage of time.

We now want to study the influence of the step-size on the relative error in energy and
Figure 3.5: The maximum relative error in the energy for the cubic, quintic, 2-step, and 3-step Hermite interpolation schemes against different step-size sequences applied to the Jovian problem over one period of Jupiter.

angular momentum while performing long-term simulations using different interpolation schemes with different integrators. Therefore, we set up an experiment to anticipate the behaviour of this relative error; see Figure 3.5. This experiment provides insight about the interpolation error in energy and angular momentum. We integrate the Jovian problem using the ERKN101217 integrator over a time interval that equals one period of Jupiter. We obtain a reference solution by integrating in quadruple precision with $TOL = 10^{-18}$ to eliminate the possible effect of integration error. We run six experiments, where we record position, velocity and acceleration at the end of every 5, 20, 50, 100, 250, and 500 days, by forcing the integrator to hit these time-points. The period of Jupiter is not divisible by 5, 10, 50, 100, 250, and 500 and hence the last step of each integration was shorter than the previous steps. The selection of such step-size sequences is done in accordance with the average step-sizes taken by different integrators when integrating the Jovian problem. For example, over $10^6$ years, $S$-13 takes a time-step of 4 days, while on average, over the same time interval, ERKN689 with $TOL = 10^{-10}$, ERKN101217 with $TOL = 10^{-11}$ and
ODEX2 with \( TOL = 10^{-16} \) take approximately 65, 290 and 260 days, respectively.

For different step-size sequences, we use appropriate interpolation schemes; for example, as discussed earlier, the 3-step interpolation scheme need not be used for the \( S\)-13 integrator with a step-size of 4 days, since quintic Hermite interpolation is sufficient. Then on each step-size, we use interpolation and sample the solution at 10 equally-spaced data points, from which we determine the local maxima for the relative error in energy and angular momentum. For each interpolation scheme, we observe that there is considerable variation in the relative error in energy and angular momentum as a function of the step-size. For example, the biggest variation of approximately 9 orders of magnitude for the relative error in energy was observed with quintic Hermite interpolation scheme over the range of 5 to 500 days step-size sequences. This particular example has reasonably good agreement with the expected difference of approximately 10 orders of magnitude, because the expression for relative error in energy is dominated by the velocity term of order 5 with quintic interpolation. This means that, by using expression \( \alpha h^{p+1} \frac{\gamma^{(p+1)}(\xi)}{(p+1)!} \) for the interpolation error as discussed in Chapter 1, the expected difference of 5 and 500 days step-size sequences is \((500/5)^5 = 10^{10}\).

Figure 3.6 contains the graphs of the relative error in energy for \( ERKN101217 \) when used with cubic, quintic (one-step), two-step and three-step interpolation polynomials, of degrees 3, 5, 8 and 11, respectively; and the three interpolants of 23, 26, and 29 stages, having orders 10, 11, and 12, respectively. The \( ERKN101217 \) integrator for an integration of \( 10^6 \) years using \( TOL = 10^{-11} \) requires, on average, a step-size of approximately 290 days. We find that the total error for the cubic, quintic, and 2-step interpolation schemes does not increase with \( t \). It remains pretty much constant, as was expected, because the interpolation error dominates the integration error.

The interpolation errors at the end of the integration have reasonably good agreement with the results obtained in Figure 3.5. With all three interpolants, i.e., 23-stage, 26-stage and 29-stage, we have obtained an accuracy of approximately \( 10^{-11} \). With the 3-step interpolation scheme, which is cheaper than all three interpolants, the accuracy is
Figure 3.6: The relative error in the energy for ERKN101217 using cubic, quintic, 2-step, 3-step Hermite interpolations, and 23-stage, 26-stage, 29-stage interpolants (the Matlab filter command with WS = 100 was used) applied to the Jovian problem over one million years for a local error tolerance of $10^{-11}$.

about the same at the end of the integration, although initially, it is dominated by the interpolation error. Therefore, it is better to use an interpolant if best accuracy is required. However, if the time interval is large enough, about $10^5$ years or more, then the integration error starts to dominate and it is better to switch to a 3-step interpolation, because this will save CPU-time and does not affect the accuracy. We also found (not shown) that the 2-step interpolation which requires less CPU-time than 3-step interpolation will give the same accuracy after even longer integration times.

Figure 3.7 shows the relative error growth in energy for ERKN689 using cubic, quintic, and two-step Hermite interpolation schemes together with a 12-stage interpolant. For $TOL = 10^{-10}$ the average time-step is approximately 65 days. For two-step interpolation and the 12-stage interpolant, there is no real difference between these two graphs, unlike for ERKN101217 using 3-step interpolation and its interpolants as shown in Figure 3.6. The two errors increase side by side right from the beginning, indicating that there is
Figure 3.7: The relative error in the energy for ERKN689 using cubic, quintic, 2-step Hermite interpolations, and the 12-stage interpolant (the Matlab filter command with WS = 100 was used) applied to the Jovian problem over one million years for a local error tolerance of $10^{-10}$.

no sense in using the 12-stage interpolant, because it costs more CPU-time than 2-step interpolation.

We extend the above experiment for ERKN689 by performing an integration of 10 million years. After approximately 7 million years (not shown) it is beneficial to switch to quintic interpolation, because it takes less CPU-time and is easier to use for the Newton-Raphson method when detecting close-encounters at a later stage of the main program. Quintic Hermite interpolation is not under consideration for the ERKN101217 integrator, because the total error differs by three orders of magnitude in comparison with ERKN689.

Figure 3.8 illustrates the relative error growth in energy for ODEX2 using cubic, quintic, two-step and three-step interpolation schemes together with its interpolant. On average, ODEX2 uses a time-step of approximately 260 days if we set $TOL = 10^{-16}$; normally it is not recommended to take such a small tolerance, because it is very close
Figure 3.8: The relative error in the energy for ODEX2 using cubic, quintic, 2-step, 3-step Hermite interpolations, and its built-in interpolant (the Matlab filter command with WS = 100 was used) applied to the Jovian problem over one million years for a local error tolerance of $10^{-16}$.

to the machine precision and round-off error could affect the results greatly. We observe that the graph for the interpolant shows oscillations, which indicates the round-off error is significant. We also found that there are quite a few rejected time-steps. Combined with there being no rejected time-steps with either of the tolerances $10^{-14}$ and $10^{-15}$, this is further evidence that the results for $TOL = 10^{-16}$ are affected by round-off error. Nevertheless, we are gaining an accuracy of approximately $10^{-11}$ (determined by a linear least square fit using data obtained by continuous approximation). This accuracy for $TOL = 10^{-16}$ is approximately one order of magnitude better than the accuracy obtained with $TOL = 10^{-15}$, which is expected when reducing the tolerance by a factor of 10. This observation suggests that the round-off error is insignificant. The behaviour of the interpolant at the end of the integration shows an abrupt dip followed by rapid oscillations. We investigated this further by performing the experiments with the same tolerance for up to $10^8$ years. While the oscillations continue, both the relative errors in energy and angular momentum increase with the passage of time and the curve ends at a total error
around $10^{-9}$ and $10^{-10}$, respectively.

We performed the same experiments using the $\bar{S}$-13 integrator with step-size of four days, along with cubic and quintic Hermite interpolation schemes. We found for cubic interpolation that the total error in energy and angular momentum does not grow with $t$, but the maximum global error has reasonably good agreement with the results obtained using a step-size sequence of five days in the experiments (see Figure 3.5). For quintic Hermite interpolation the error growth in energy and angular momentum is below linear.

<table>
<thead>
<tr>
<th>Integrator</th>
<th>Interpolation</th>
<th>$E_r$</th>
<th>$E_v$</th>
<th>$H_{rel}$</th>
<th>$L_{rel}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERKN689</td>
<td>—</td>
<td>1.980</td>
<td>1.987</td>
<td>0.986</td>
<td>0.971</td>
</tr>
<tr>
<td>ERKN689</td>
<td>2-step</td>
<td>—</td>
<td>—</td>
<td>0.991</td>
<td>0.979</td>
</tr>
<tr>
<td>ERKN689</td>
<td>12-stage</td>
<td>—</td>
<td>—</td>
<td>0.985</td>
<td>0.970</td>
</tr>
<tr>
<td>ERKNI012</td>
<td>—</td>
<td>1.997</td>
<td>1.996</td>
<td>0.998</td>
<td>1.005</td>
</tr>
<tr>
<td>ERKNI01217</td>
<td>3-step</td>
<td>—</td>
<td>—</td>
<td>0.677</td>
<td>0.683</td>
</tr>
<tr>
<td>ERKNI01217</td>
<td>23-stage</td>
<td>—</td>
<td>—</td>
<td>0.998</td>
<td>1.005</td>
</tr>
<tr>
<td>ERKNI01217</td>
<td>26-stage</td>
<td>—</td>
<td>—</td>
<td>0.997</td>
<td>1.004</td>
</tr>
<tr>
<td>ERKNI01217</td>
<td>29-stage</td>
<td>—</td>
<td>—</td>
<td>0.997</td>
<td>1.005</td>
</tr>
<tr>
<td>ODEX2</td>
<td>—</td>
<td>1.514</td>
<td>1.457</td>
<td>0.327</td>
<td>0.474</td>
</tr>
<tr>
<td>ODEX2</td>
<td>3-step</td>
<td>—</td>
<td>—</td>
<td>0.589</td>
<td>0.632</td>
</tr>
<tr>
<td>ODEX2</td>
<td>interpolant</td>
<td>—</td>
<td>—</td>
<td>0.325</td>
<td>0.465</td>
</tr>
<tr>
<td>$\bar{S}$-13</td>
<td>—</td>
<td>1.377</td>
<td>1.542</td>
<td>0.689</td>
<td>0.718</td>
</tr>
<tr>
<td>$\bar{S}$-13</td>
<td>quintic</td>
<td>—</td>
<td>—</td>
<td>0.687</td>
<td>0.724</td>
</tr>
</tbody>
</table>

Table 3.7: The exponent $b$ of the power law for the global error in position and velocity, and relative error in the energy and angular momentum for the Jovian problem over one million years. We used local error tolerances $10^{-11}$, $10^{-10}$, and $10^{-16}$ with different interpolation schemes, for the integrators $\text{ERKN689}$, $\text{ERKNI01217}$, $\text{ODEX2}$, and $\bar{S}$-13, respectively. The step-size for $\bar{S}$-13 is four days.

Table 3.7 shows the exponent $b$ in a linear least squares fit for the power law $at^b$. In particular, when low-order interpolation schemes are used with high-order variable-step-size integrators, the exponent $b$ of the power law should be very close to zero. For example, for $\text{ERKN689}$ using cubic and quintic Hermite interpolation schemes, the exponent $b$ varies from 0.029 to 0.35 for the relative error in energy and angular momentum. We already noted that the $\text{ODEX2}$ integrator is affected by round-off error if $\text{TOL} = 10^{-16}$. To observe the extended behavior of the relative error in energy and angular momentum, we integrated $\text{ODEX2}$ using $\text{TOL} = 10^{-16}$ over $10^7$ years and observed the values for
the relative error in energy and angular momentum as 0.76 and 0.89 respectively. We also repeated experiments for \textit{ODEX2} with $TOL = 10^{-15}$ and $10^{-14}$ and observed that the value of the exponent increases. In particular, with $TOL = 10^{-14}$, the value of $b$ is approximately 1.92 for the global error in position; for the relative error in energy and angular momentum, it is approximately 0.99 and 1.01, respectively.
Algorithm and initial conditions

One goal of this thesis is to investigate the accuracy of different combinations of numerical integrators and interpolation schemes for $N$-body simulations. $N$-body simulations are performed using either a low-order symplectic method with a large step-size and a special technique, possibly non-symplectic, for handling close-encounters between test particles and massive bodies, or by using a high-order non-symplectic method, with the step-size chosen so the integration is done very accurately. There are two challenges to performing accurate simulations: keeping the accumulated error small enough, and ensuring the simulations can be done in a reasonable amount of time.
When the simulation is done in IEEE double-precision, the accumulative round-off error in a long interval of integration can cause a large error in the position of a massive body or a test particle, even though the step-size is chosen such that the truncation error is below machine precision. The round-off error can be in the first significant digit, unless techniques to control the round-off error are used; eventually, even with these techniques, all digits are lost if one integrates far enough.

A large number of test particles combined with a long interval may require a considerable amount of CPU-time to complete the N-body simulations on a single processor. The elapse time can be reduced substantially if more than one processor is available, because the absence of interaction between test particles readily permits doing the simulations in parallel. If there are \( k \) processors, where \( k \) divides \( N \), then \( N/k \) of the test particles, along with a copy of the massive bodies, are placed on each processor and treated as a separate simulation; effectively, the \( N \)-body simulation is done by means of \( k \) independent \( N/k \)-body simulations. The absence of interaction between test particles also means that the test particles can be integrated using a multirate scheme. The massive bodies are integrated forward in time from \( t \) to \( t + \Delta t \). The test particles are then integrated one at a time from \( t \) to \( t + \Delta t \) using individual step-size sequences; the positions of the massive bodies for the calculation of the forces on the test particles are found by interpolation.

The multirate algorithm of [32], uses different step-sizes as follows. The step-size \( H \) is used to integrate the orbital motion of the Sun and planets at all times, and the orbital motion of test particles when they are not undergoing a close-encounter. The step-size \( h_i \ll H \), for \( i = 1, \ldots, N_p \), is used to integrate the orbital motion of test particles during a close-encounter with the \( i \)-th massive body, where \( N_p \) is the number of planets and the Sun. The position of the Sun and planets during a close-encounter is calculated using Hermite interpolation. This interpolation scheme is also used to approximate the orbits of the test particles and to decide if a test particle has collided with a massive body.

In this chapter we present and illustrate a general multirate algorithm for \( N \)-body
simulations. We describe the core part of our algorithm in Section 4.1. We follow this in Section 4.2 with a description of enhancements to the core algorithm that reduce CPU-time. We then conclude with the initial conditions of the asteroids.

## 4.1 Core Algorithm

Table 4.1 contains an overview of the core part of our algorithm, which integrates the entire system of massive bodies and test particles from some pre-specified initial time $t_0$ to a pre-specified final time $t_f$. Lines 1 to 4 in Table 4.1 specify the input values to the algorithm. To run the code, one needs to supply the values for $t_0$ and $t_f$, the positions $y_0$ and velocities $y'_0$ for the massive bodies at time $t = t_0$, and the radii of the activity spheres $R_{act}^i, i = 1, ..., N_p$, for the massive bodies, where, as before, $N_p$ is the number of massive bodies. Recall from Chapter 1 that the radius of the activity sphere is larger than the physical radius of the individual body; see Figure 1.1 and Table 1.1. The initial values $y_0$ and $y'_0$ can be the final values from a previous simulation. One must also specify the positions $z_i$ and velocities $z'_i, i = 1, ..., N_a,$ of the test particles at time $t_0$, where $N_a = N - N_p$ is the number of test particles. The user should also select a numerical integrator. The algorithm options include the specification of the integration method for the massive bodies and test particles, the interpolation scheme for the position of the massive bodies when integrating the test particles, and the scheme used to detect close-encounters and ejections. Furthermore, the user must specify the initial step-size $H$ for the massive bodies; the initial step-size $h_i, i = 1, ..., N_a$ for each test particle; the length $\Delta t$ of each sub-interval obtained by subdividing the complete interval $[t_0, t_f]$ of integration (the reason for subdivision is explained below); the local error tolerance $TOL$ for the integration of the massive bodies and test particles; and other possible integration options. If the massive bodies are to be integrated using a fixed-step-size scheme then the initial value of $H$ will be used throughout the simulation.

The integration phase consists of a nested loop with three parts. The execution of
the outer loop, lines 6–33 in Table 4.1, is handled by the routine Massive (line 8), which attempts to integrate the massive bodies from \( t = t_m \) to \( t = t_m + \Delta t \). The interval \( \Delta t \) is a lot larger than the typical step-size used by the integration of the massive bodies. Hence, the integrator for the massive bodies will take more than one step to integrate from \( t_m \) to \( t_m + \Delta t \).

```
1 Input: \( t_0, t_f \)
2 Input: \( y_0, y'_0, R_{\text{act}}^i, i = 1, ..., N_p \)
3 Input: \( z_{i,0}, z'_{i,0}, i = 1, ..., N_a = N - N_p \)
4 Input: \( H, h_i, i = 1, ..., N_a, \Delta t, TOL, \text{options} \)
5 \( t_m = t_0 \)
6 while \( t_m < t_f \& N_a > 0 \) do
7 \( t_{\text{save}} := t_m \)
8 Massive \( (t_m, t_m + \Delta t, \text{MassiveFailed}) \)
9 if MassiveFailed then
10 exit
11 end
12 \( t_m = t_m + \Delta t, n_{\text{alive}} := 0 \)
13 for \( i = 1, ..., N_a \) do
14 \( t_p := t_{\text{save}}, \text{alive} := \text{true} \)
15 while \( t_p < t_m \& \text{alive} \) do
16 TestParticle \( (t_p, h_p, \text{TpFailed}) \)
17 if TpFailed then
18 alive = false
19 else
20 CloseEncounter \( (\text{alive}) \)
21 if alive then
22 Ejection \( (\text{alive}) \)
23 end
24 end
25 end
26 if alive then
27 \( n_{\text{alive}} = n_{\text{alive}} + 1 \)
28 Save \( z_i, z'_i, \) and interpolation information
29 end
30 \( N_a = n_{\text{alive}} \)
31 end
```

Table 4.1: The core algorithm for our \( N \)-body simulations.

If the integration from \( t_m \) to \( t_m + \Delta t \) of the massive bodies fails, which is an unlikely event, the flag MassiveFailed is set to true and the simulation terminated. If the inte-
igration succeeds, **Massive** returns with the positions and velocities of the massive bodies at \( t = t_m + \Delta t \) and any additional information needed to form a sufficiently accurate approximation to the positions, velocities, and possibly accelerations of the massive bodies over the interval \([t_m, t_m + \Delta t]\). The approximation is a piecewise-defined polynomial function formed from information in each integration time-step taken by **Massive** while going from \( t = t_m \) to \( t = t_m + \Delta t \).

The two level loop (lines 13–30) runs through the test particles that are still present in the simulation at time \( t_{\text{save}} \) (\( = t_m \)). We label such test particles with the boolean ‘alive’. For each test particle, the routine **TestParticle** (lines 15–25) attempts to integrate the test particle from \( t = t_{\text{save}} \) to \( t = t_m + \Delta t \) in single-step mode, that is, **TestParticle** performs an integration step from \( t = t_p \) to \( t = t_p + h_i \), where \( t_p = t_{\text{save}} \) at the start. If the integration step fails, the most common cause of which, in our experience, is requesting too much accuracy, the test particle is deemed not alive and its integration terminated.

If the integration step succeeds, **TestParticle** returns with the position \( z_i \) and velocity \( z_i' \) of the \( i \)-th test particle at the end of the step and any additional information needed to form a sufficiently accurate approximation to \( z_i \) on the interval \([t_p, t_p + h_i]\). The routine **CloseEncounter** is then invoked to check if the test particle would have made a close-encounter with any of the massive bodies on the integration step just taken. If no close-encounter has occurred, **Ejection** is invoked to check if the particle has been ejected from the Solar System. If it has not, the test particle is deemed alive at the end of the step. The integration of the \( i \)-th test particle continues until it is either terminated through a close-encounter or ejection, it cannot be integrated, or \( t = t_m + \Delta t \) is reached.

In the last case, the count ‘alive’ on the number of test particles is increased by one, and the position and velocity of the test particle at \( t = t_m + \Delta t \) is saved. Once the integration of all the test particles from \( t = t_{\text{save}} \) to \( t = t_m + \Delta t \) has been attempted and the results processed, \( N_a \) is set to the number of alive particles at \( t = t_m + \Delta t \).

Figure 4.1 shows a schematic that presents different combinations of integrators, interpolation schemes and non-linear equation solvers used in this thesis. The integrators
used for routine **Massive** are ERKN689, ERKN101217, ODEX2, and ŝ-13, which were discussed in Chapter 2. The interpolation schemes used for the continuous approximation of the massive bodies are the cubic, quintic, 2-step and 3-step Hermite interpolation schemes discussed in Chapter 3. The integrators used for routine **TestParticle** are

![Diagram](image_url)

**Figure 4.1:** A schematic that presents different combinations in the main algorithm of the integration schemes of the planets and asteroids with interpolation schemes.

ERKN689, ERKN101217, and ODEX2. The interpolation schemes used for the continuous approximation of the test particles are the same as used for the massive bodies. The non-linear equation solvers include the Newton-Raphson (NR) and Secant (SEC) meth-
ods. We also added the quadratic formula (QDF) for cubic Hermite interpolation and the quartic formula (QRF) for the quintic Hermite interpolation scheme. We note that there is a complicated interaction between different sources of errors introduced by the integration of the massive bodies, the interpolation scheme used for the approximation of the massive bodies, the integration of the test particles, the interpolation scheme used for the approximation of the test particles, and, to a lesser extent, by the non-linear equation solver.

![Diagram](image_url)

**Figure 4.2:** A schematic with interpolants that presents different combinations in the main algorithm.
Figure 4.2 is Figure 4.1 with the Hermite interpolation schemes replaced by the additional interpolants discussed in Chapter 3. The integrators used for the routines **Massive** and **TestParticle** are **ERKN689**, **ERKN101217**, and **ODEX2**. For the continuous approximation with **ERKN689**(CON-ERKN68), we have a 12-stage derivative, order-8 interpolant. For the continuous approximation of **ERKN101217**(CON-ERKN1012), we have three interpolants of 23-stage, 26-stage, and 29-stage derivatives, having orders 10, 11, and 12, respectively. For **ODEX2**(CON-ODEX2), we have one variable-order interpolant.

4.1.1 The routine Massive

The main role of the integrator for the massive bodies is to produce a sufficiently accurate continuous approximation to $u_n(t)$ and to $u'_n(t)$ on each integration step, where $u_n(t)$ denotes the local solution as defined earlier. Let $P_n(t)$ be the continuous approximation to $u(t)$ on the step from $t = t_{n-1}$ to $t = t_n$, and $y_n$ the numerical approximation to $u_n(t_n)$.

The piecewise-defined continuous approximations for the integration steps from $t = t_m$ to $t = t_m + \Delta t$ is formed by a family of such continuous approximations $P_n(t)$. We require $y_n$ and $P_n(t)$ to satisfy the following two conditions:

C1. The norm $||y_n - u_n(t_n)||$ is close to the smallest possible value for the arithmetic being used, and

C2. There exists $0 < \alpha \ll 1$, such that

$$\max_{t \in [t_{n-1}, t_n]} \left\{ \frac{||P_n(t) - u_n(t)||}{||y_n - u_n(t_n)||} \right\} \leq 1 + \alpha.$$ 

As discussed in Chapter 2, the truncation error in the position for non-symplectic integration methods grows as $t^2$. When non-symplectic methods are implemented in a standard way, the round-off error is usually systematic, and grows as $t^2$ as well. Brouwer [4] showed that if the step-size for any integration method is chosen so that the local truncation error is below machine precision and the round-off error is stochastic, then the errors in
conserved quantities and dynamical variables, such as position, grow as \( t^{1/2} \) and \( t^{3/2} \), respectively. These power laws are now known collectively as Brouwer’s Law and, since they represent the best power-law growth possible when accurate solutions are required, they suggest that Massive should be executed using an integration scheme that satisfies Brouwer’s Law.

The first implementation that satisfied Brouwer’s Law was developed by Grazier [26]; see also [28, 29]. Laskar et al. [44] presented a symplectic method of order \( O(h^8\varepsilon) + O(h^4\varepsilon^2) \), where \( h \) is the step-size and \( \varepsilon \) is a typical planetary mass expressed as a function of Solar masses. They performed a simulation of the eight planets, Pluto and the Moon, using a step-size of 1.83 days and found that the error in the energy at any given time is well approximated by a Gaussian distribution. Hairer et al. [34] investigated the propagation of the round-off error for implicit Gauss Runge-Kutta methods when fixed-point iteration is used to solve for the stages. They showed that the order-12 method can be implemented so that the error in the energy grows approximately as \( t^{1/2} \) when performing simulations of the Sun, Jupiter, Saturn, Uranus, Neptune, and Pluto with a step-size of 167 days.

Integration methods that do not satisfy Brouwer’s Law should not be excluded because the multiplicative constant in the power law \( t^2 \) could be a lot smaller than that for the power law \( t^{3/2} \). In this case, there will be a crossover value \( t_c \): for \( t \leq t_c \), the method that does not satisfy Brouwer’s Law will have a smaller error than the method that does. For \( t > t_c \), the conclusion is reversed.

As an example, we mention the following. As part of a large set of comparisons, Sharp [56] compared an ERKN integrator called RKNINT with the order-13 Störmer integrator of Grazier [26] on a ten-million-year simulation of the Sun, Jupiter, Saturn, Uranus and Neptune. Sharp confirmed the expected \( t^2 \) and \( t^{3/2} \) error growth for RKNINT and the Störmer integrator, respectively, to within experimental uncertainty. However, Sharp found that the norm of the global error for RKNINT at ten million years was an order of magnitude smaller than that for the Störmer integrator. Sharp also found that
RKNINT required considerably less CPU-time than the Störmer integrator. The value of \( t_c \) in the example above appears to be large, but it is not clear whether this is true in general. In Chapter 2 we investigated this in more detail. In particular, we considered the error growth for the case where an accurate approximation is required to the solution across a step, and not just at the end of each step. It is important to have a good estimate for the value of \( t_c \). If \( t_c \) is just a few thousand years, then the existence of \( t_c \) is only of academic interest. On the other hand, if \( t_c \) is tens of millions of years, then its existence is of practical importance.

### 4.1.2 The routine TestParticle

The integration method in **TestParticle** must produce an accurate continuous approximation to orbits on each step. This is the same requirement as for the integration method in **Massive**, but the orbits of the test particles are more difficult to integrate than those for the massive bodies, because they include possible close-encounters between two bodies and the test particles could be highly eccentric orbits.

The close-encounters necessitate a variable-step-size integration method, or something equivalent, such as integration with smaller step-size (but this will increase the CPU-time and possibly add round-off error). The variable-step-size integration method can also be employed when a test particle is not undergoing a close-encounter, but there is no compulsion to do so; another method could be used, leading to a hybrid integration scheme for test particles. This approach was adopted by Chambers [8], who used a variable-order, variable-step-size extrapolation method for close-encounters, and a low-order symplectic method otherwise. The switch between the two integration methods is controlled by a changeover function.

Opening up the integration to a hybrid scheme leads to a richness of possibilities but comes at a cost of complicating the algorithm. To illustrate these complications, we describe two contrasting schemes. We assume that an acceptable continuous local approximation is available for all the integration methods discussed.
The first hybrid scheme uses the fixed-step $\tilde{S}-13$ scheme of Grazier [26] and a variable-step-size ERKN method for the close-encounters. There are four critical aspects to this hybrid scheme:

(i) the test for the switch from the $\tilde{S}-13$ to the ERKN method;
(ii) the step-size to use for the first step of the ERKN method;
(iii) the test for the switch from the ERKN to $\tilde{S}-13$ method;
(iv) re-starting the $\tilde{S}-13$ method.

As a test particle approaches a massive body, the acceleration of the test particle will increase. This will increase the norm of its highest-order difference for the $\tilde{S}-13$ method. This norm can be monitored and if it is larger than a prescribed value $D$, the switch to the ERKN method made. The test can be made more robust by using the sum of the norms of the two highest differences. This will reduce the likelihood of fortuitous cancellation that might occur if just one difference is used. A suitable value for $D$ can be found by performing numerical experiments before the simulation is run. Given the CPU-time required for the simulation, and the fact that the same $D$ can be used in other simulations with the same equations of motion for the massive bodies, the time spent performing the experiments is worthwhile.

The above test is an example of a test that uses the numerical solution. Another type of test uses the model being integrated. One such test is to switch when the distance of the test particle from the centre of the massive body is less than a prescribed value. As with $D$, a suitable value for the distance can be found by numerical experiments before the simulation is performed.

Once the switch to the ERKN method is made, we want the step-size used by the method to be on scale as quickly as possible. There are at least two ways that this might be achieved. One way is to use the on-scale step-size from the previous encounter between a test particle and the same massive body. A second way is to allow the step-size selection scheme to find a suitable step-size.
The test for the switch from the ERKN back to the $\bar{S}$-13 method can be based on the distance of the test particle from the centre of the massive body. Alternatively, the step-size for the step just taken can be used: if it is greater than a prescribed value, found by numerical experiments, the switch to the $\bar{S}$-13 method is made.

4.1.3 CloseEncounter

The general approach to detecting a close-encounter between a massive body and a test particle on an integration step by TestParticle is first to form a continuous approximation that accurately approximates the distance between the massive body and the test particle. Then, use this approximation to decide if the distance between the massive body and the test particle on the step was less than or equal to the close-encounter radius for the massive body. If yes, a close-encounter occurred and the test particle is removed from the simulation.

4.1.4 Removal

Once the integration over the interval $\Delta t$ in Table 4.1 is completed (CloseEncounter and Ejection are invoked), the pieces of information, for example, positions, velocities, and possibly accelerations of the test particles can be updated using the following routine, where remove($i$) is true if the $i^{th}$ test particle is to be removed.

\[
k = 0
\]

\[
\text{for } i = 1 .. N_a
\]

\[
\text{if (.not. remove($i$)) then}
\]

\[
k = k + 1
\]

Copy information of the $i^{th}$ test particle into $k^{th}$ location of work-array

\[
\text{end}
\]

\[
\text{end}
\]

An alternative scheme is to remove a test particle as soon as a close-encounter or ejection
occurs. This is done by eliminating position, velocity, and possibly acceleration components, and then updating the number of bodies left.

It does not really matter which approach is used because the CPU-time for removal is insignificant compared with that for the rest of the code.

4.2 Other integration details

It seems natural to choose the final step-size $h_{\text{last}}$ on the integration from $t = t_{\text{save}}$ to $t = t_{\text{save}} + \Delta t$, such that the integration of the massive bodies hits the point $t = t_{\text{save}} + \Delta t$ exactly. However, then $h_{\text{last}}$ will almost always be smaller than required for the local error test on the step. This reduction in step-size will usually make the integration inefficient compared with an integration where the step-size is unrestricted. There are at least two ways in which this inefficiency might be reduced.

One is to relax the requirement that $t = t_{\text{save}} + \Delta t$ be hit exactly and let the integrator choose $h_{\text{last}}$ to satisfy the local error test. Except in rare circumstances, the integrator will then step past $t = t_{\text{save}} + \Delta t$, and integration will be performed for at least one value of $t > t_{\text{save}} + \Delta t$. This will require the value of $y(t)$ which will, thus, necessarily be obtained by extrapolating the interpolant for $y$. If $(t - t_{\text{save}} - \Delta t)/H$ is too large, the error in the extrapolation will be unacceptably large. This situation can be avoided by putting an upper bound on $(t - t_{\text{save}} - \Delta t)/H$, but this bound will be problem dependent.

A second possible way to reduce the inefficiency is to increase $\Delta t$ and still have the integrator hit $t = t_{\text{save}} + \Delta t$. Increasing $\Delta t$ would reduce the percentage effect of any reduction in $h_{\text{last}}$.

Even when using an integrator that satisfies Brouwer’s Law, we have found numerically that the round-off error in a long simulation can be in the second or third significant figure when using double-precision. This round-off error can be made insignificant by doing the simulations in quadruple-precision. However, on most computers, this comes
at a large increase in CPU-time for each equation, because quadruple precision is usually
done in software, making it two orders of magnitude slower than double precision.

There is the intermediate possibility of integrating the massive bodies in quadruple
precision and the test particles in double precision. We tested whether the CPU-time
required is acceptable, when using quadruple precision in this way, by simulating the
orbital motion of the Sun, Jupiter, Saturn, Uranus and Neptune over 100 million years
using the standard Newtonian model, where the massive bodies are treated as point
masses. Except for special values of $t$, every evaluation of the derivatives of the test
particles requires the evaluation of the continuous approximation of the massive bodies.
How much this evaluation increases the CPU-time for the simulation will depend on the
model used for the equations of motion, the interpolation scheme, and how this scheme
is implemented.

### 4.3 Initial conditions of asteroids

In order to start the simulations of the $N$-body problem we must select suitable initial
conditions for the Sun, planets and the asteroids. In this section we describe the process
of calculating the initial conditions of asteroids.

Figure 4.3 depicts Keplerian orbital elements for a body whose orbital plane inter-
sects a reference plane. This intersection is known as the line of nodes because it connects
the centre of mass with the ascending and descending nodes (an orbital node is one of
the two points where an orbit crosses the plane of reference to which it is inclined). For
Earth-orbiting satellites, the reference plane is normally Earth’s equatorial plane. If the
satellites are in the orbits about the Sun then the reference plane is usually the ecliptic.
This plane together with the vernal point defines a reference frame.

There are six orbital elements needed to define an orbit of an asteroid; they are
described in Table 4.2. In all of our tests, except for the HRC problem, the initial
conditions of the asteroids are calculated in such a way that the semi-major axis \( a \) is normally distributed about the midpoint of the semi-major axis for Jupiter and Saturn, with a standard deviation equal to the semi-major axis for Jupiter subtracted from the semi-major axis for Saturn and divided by six. The eccentricity \( e \) is obtained from a negative exponential distribution. The inclination is normally distributed about zero and the remaining three orbital elements are chosen from uniform distributions.

Figure 4.4 depicts the \( x - y \) projection of the orbits of Jupiter, Saturn, Uranus, and Neptune together with the initial positions of one set of 1000 asteroids we used in our testing.
<table>
<thead>
<tr>
<th>Elements</th>
<th>Definitions</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>The semi-major axis measures the size of the orbit; in our simulations ( a ) is expressed in astronomical units.</td>
</tr>
<tr>
<td>e</td>
<td>The eccentricity describes the shape of the orbit.</td>
</tr>
<tr>
<td>i</td>
<td>The inclination, describes the vertical tilt of the ellipse relative to the reference plane.</td>
</tr>
<tr>
<td>ω</td>
<td>The ascending node, describes the orientation of the ellipse.</td>
</tr>
<tr>
<td>Ω</td>
<td>The argument of periapsis, horizontally orients the ascending node of the ellipse.</td>
</tr>
<tr>
<td>ν</td>
<td>The true anomaly, specifies the current position of the orbiting body along the ellipse.</td>
</tr>
</tbody>
</table>

**Table 4.2: Description of the six orbital elements.**

![Diagram of Jovian planets and asteroids](image_url)

**Figure 4.4:** The orbits of Jovian planets and the initial positions for a set of 1000 asteroids, all projected on the x-y plane.
In this chapter we present numerical testing for the accuracy and efficiency of detecting close-encounters between the massive bodies and the massless bodies. As in the previous chapters, we compare various combinations of different integrators and interpolation schemes used by the main algorithm on a collection of test problems. We start this chapter with the selection of quality asteroids. We describe the error for a close-encounter and end with a discussion of the numerical test results.
5.1 Selection of quality asteroids

When we started performing simulations with the main algorithm, we were puzzled by the outcomes: a slight change in tolerance had an enormous effect on the results. For instance, performing simulation in quadruple-precision with a set of 10 randomly selected asteroids, using an extremely small tolerance of $10^{-24}$, we detected 8 close-encounters. When the simulation was repeated with the same combination of the integrator and interpolation scheme using an even smaller tolerance of $10^{-26}$, the number of detected close-encounters changed to 7. After much experimenting, we decided to investigate the orbital behaviour of all asteroids because we suspected an instability in the orbit of one or more asteroids.

Figure 5.1 (top) shows the norm of the difference at the end of the integration between the solutions calculated using double- and quadruple-precision of 1000 asteroids against their minimum distance from the Jovian planets. The integration was performed in double-precision using $ERKN101217$ with its 29-stage interpolant over one million days. The reference solution was obtained in quadruple-precision using the small tolerance $10^{-18}$.

At the end of the integration, a big difference was observed in the orbits of some of the asteroids. This confirms that our puzzling results were, indeed, due to instability of the orbits of individual asteroids. We also found that some of them are deflected away from their true orbits up to 100 (astronomical) units.

The decision to exclude the unstable asteroids may seem to limit the applicability of our work. Our justification for this exclusion is that the emphasis in $N$-body simulations of asteroids and other small bodies is on the long-term evolution of the bodies because the longer such bodies are present, the more likely they are to be of concern.

Figure 5.1 (bottom) shows the classification of these asteroids. They have been classified into four different categories: the bound $TOLL = 10^{-4}$ is our acceptable norm of the difference of the position at the end of the integration calculated using double and quadruple precision of the asteroids; see the description above. The other bound is a minimum distance of $R_{act} = 0.3$ from the Jovian planets, which ensures that the close-
Figure 5.1: (Top) The norm of the difference of the position at the end of the integration calculated using double and quadruple precision of a set of 1000 asteroids against the minimum distance to the Jovian planets using the ERKN101217 integrator with its 29-stage interpolant over one million days. The dashed lines bound four categories of asteroids as described in the bottom part of the figure. (Bottom) The classification of stable and unstable asteroids including those that do and do not make close-encounters (CE).
Figure 5.2: The growth of the global error for a set of 5 quality asteroids using the ERKN101217 integrator with its 29-stage interpolant over approximately one million days, with $TOL = 10^{-16}$. Shown are the error growth (The Matlab filter command with WS = 100 was used for both position and velocity) in position (top) and the error growth in velocity (bottom).
encounters can be expected. The four categories are then: “CE but UNSTABLE” for asteroids that make close-encounters but deflect too far away from their true orbits; “UNSTABLE and NO CE” for asteroids that deflect too far away from their true orbits and make no close-encounters within one million days; “CE and STABLE” for asteroids that make close-encounters and also remain acceptably close to their true orbits; and, finally, “STABLE but NO CE” for asteroids that remain close to their true orbits, but make no close-encounters within one million days. We observe from Figure 5.1 (top) that a high percentage of asteroids are stable (the maximum difference is no larger than $10^{-4}$). However, most of the stable asteroids do not make close-encounters; of course, when integrated further, they could change their category. We also find quite a few asteroids in the category “CE but UNSTABLE”, and some of the asteroids that make a close-encounter fall in the category “CE and STABLE”. Note that these asteroids were randomly chosen from a sensible distribution. If the asteroid is unstable then it will in a relatively short time be making a close encounter or it will be ejected from the Solar System. Therefore, if we are interested in long-term simulations it is better to restrict the selection to stable asteroids only. Moreover, in order to test the accuracy of the main algorithm, we should select asteroids that make close-encounters and are stable. Therefore, we only choose asteroids from the category “CE and STABLE” and we call these quality asteroids. We identified quality asteroids by running sets of 1000 asteroids at the time. Only 7 asteroids were quality asteroids in the first set of 1000 asteroids; see Figure 5.1 (top). We repeated the experiment 100 times with different sets of 1000 asteroids to obtain approximately 1100 quality asteroids from the set of 100,000 asteroids. These 1100 quality asteroids are used in the experiments described below.

Figure 5.2 shows the error growth in position and velocity for five of the quality asteroids that make their first close-encounter near the end of $10^6$ days of integration. We used the integrator $ERKN101217$ with its 29-stage interpolant over one million days and $TOL = 10^{-16}$; the interval of integration was subdivided into 1000 evenly-spaced sub-intervals. At the end of each sub-interval, we estimated the errors (see Chapter 3 for more details on the estimation of errors) in the position and velocity of the individual
asteroid. For this set of experiments, we again used the filter command with a window size of \( WS = 100 \). The quality asteroids are integrated until just before making their first close-encounter. We observe a variation of nearly four orders of magnitude in the global error for these five quality asteroids. The smallest global error achieved is around \( 10^{-10} \). Hence, we conclude that the accuracy at the time of a close-encounter can vary significantly among quality asteroids.

5.2 Close-encounter error

In this section we discuss two types of close-encounter errors:

(a) The error in a single close-encounter detected during a simulation, and
(b) The error in all close-encounters in a simulation.

To illustrate the error in a single close-encounter detected during a simulation, consider a planet at the centre of the activity sphere with radius \( R_{\text{act}} \), as sketched in Figure 5.3, and suppose we use three different combinations of integrators and interpolation schemes in the main algorithm. Their integration time-steps are denoted by a small circle, a triangle, and a cross, respectively. The three combinations will result in three slightly different orbits, but we used a single trajectory in Figure 5.3; the difference between orbits is insignificant compared to the radius of the activity sphere for our purposes here.

Consider the orbit generated by the combination denoted by the small circle. The end of two time-steps is well inside the activity sphere. The distances between the planet and the asteroid at the end of the time-steps inside the activity sphere are less than the radius of the activity sphere and the close-encounter is reported. Next, consider the orbit approximation corresponding to time-steps denoted by a cross. The end of one time-step is inside the activity sphere, which we also report as a close-encounter. Finally, consider the third orbit associated with the triangles. This orbit takes a large time-step and the end of the step does not lie inside the activity sphere. Hence, we would miss the close-encounter for this orbit.
Figure 5.3: An illustration of error in a single close-encounter detected during a simulation. The integration time-steps of three different combinations of the main algorithm are denoted by symbols: small-circle, triangle, and cross, respectively.

Even though two of the three orbit approximations would report a close-encounter, there is likely to be a difference in the time of the close-encounter. There should be a unique time of close-encounter to quantify which of these solutions has to be chosen for the time of close-encounter. In order to ensure fair comparisons between different combinations of the main algorithm, we need to define a unique time of close-encounter.

One possibility is to take the distance between the planet and the asteroid exactly equal to the radius of the activity sphere and record the corresponding time this equality first occurs as the time of close-encounter. Physically, there is also a unique global minimum distance between the planet and the asteroid for a single close-encounter. So, another possibility is to take this global minimum distance. For both possibilities, we use a non-linear equation solver (NLES) and an interpolation scheme to find an approximation of the time of close-encounter. As far as the second possibility is concerned, it has the disadvantage that the close-encounter is recorded as soon as the time-step lies inside the activity sphere, which may be before the point of minimal distance. Hence, the second definition may require extrapolation, or the need to integrate the asteroid further for one or more time-steps, in a post-process routine that stores all this information. This particular situation is illustrated in Figure 5.3, for the combination denoted
by small circle: two time-steps lie inside the activity sphere, one before and the other after the point of minimal distance. We consider only the first close-encounter between the planet and a particular asteroid; after the first CE, the asteroid is removed from the simulation. Therefore, we decide to use the simplest criterion and associate the time of close-encounter by determining the first point on the activity sphere.

5.2.1 Detecting close-encounter using NLES

A close-encounter is recorded as soon as the condition $||r_p - r_a||_2 \leq R_{act}$ is satisfied, where $r_p$ and $r_a$ are the position vectors of the planet and the asteroid, respectively. We calculate the distance between the planet and the asteroid at the left and right end-points of the first time-step and at the right end-points for each subsequent time-step. If the distance is less than the radius of the activity sphere then we report a close-encounter. Otherwise, we use interpolation and NLES to check for a close-encounter inside the time-step; this ensures that we also detect a close-encounter for the orbit associated with the integration time-steps denoted by triangles, shown in Figure 5.3.

There are two general possibilities to approximate the distance between the planet and the asteroid. One possibility is to approximate each of the components for $r_p$ and $r_a$ and then form the difference between the vectors $r_p$ and $r_a$. Another possibility is to approximate $\Delta = ||r_p - r_a||_2$ directly.

For example, let us use polynomials of degree 3, denoted by $P^p_3(t)$ and $P^a_3(t)$, as continuous approximations for the planet and the asteroid, respectively. Then $\Delta = ||r_p - r_a||_2$ can be approximated as

$$\Delta^2 = ||r_p - r_a||^2_2 \approx ||P^p_3(t) - P^a_3(t)||^2_2 = ||P_3(t)||^2_2.$$ 

We can easily approximate $\Delta^2$ by the cubic Hermite polynomial using the values for $\Delta^2_{n-1}$, $(\Delta^2)_{n-1}$ and $\Delta^2_n$, $(\Delta^2)_n$ at the left and right end-points of the time-step from $t_{n-1}$ to $t_n$. So, there are two different approximations. The first approximation involves a square root, whereas, the second approximation does not.
Once the close-encounter is detected, either at the end-point or inside the interval, we follow another criterion to observe the accuracy of each close-encounter, namely, $\Delta - R_{act} = 0$, as discussed in Section 5.2.1.1. Despite the fact that there is a potential difficulty caused by $|\Delta - R_{act}| = ||r_p - r_a|| - R_{act}$ not being differentiable when $\Delta = R_{act}$, we found the scheme worked well in practice.

If we choose the Newton-Raphson method for the non-linear equation solver we need to find the derivative of the distance with respect to time. For the Hermite interpolation polynomials, we obtain the derivative directly through differentiation of the $k$-th degree polynomial, with $k = 3, 5, 8, 11$. For the ERKN integrators, derivative interpolants are derived separately and not as derivatives of an interpolant.

Another important factor for the non-linear equation solver is the initial estimate of the solution, especially in the case of the Newton-Raphson method. If the initial estimate is sufficiently close to the root then the Newton-Raphson method converges quadratically. We performed experiments using several types of initial estimates, for example, the left end-point, the right end-point, and the mid-point of the interval, but it is difficult to determine one particular initial estimate that is suitable for all close-encounters in a simulation. For one particular combination in the main algorithm, we found, at one time, that the left end-point was an appropriate initial estimate, because the close-encounter was detected close to this left end-point of the interval. On the other hand, this initial estimate was not a good choice for the second close-encounter in the simulation, because the second close-encounter occurred close to the right end-point. Therefore, we decided to take the midpoint of the interval as an initial estimate.

5.2.1.1 Linear inverse interpolation scheme

For a close-encounter, there exists a well-defined point of intersection between the activity sphere and the trajectory of the asteroid. This is ensured by the fact that the minimum distance between the planet and the asteroid is less than the radius of the activity sphere.
at one of the end-points of a step or inside the interval, which we established by the continuous approximation and the non-linear equation solver. Therefore, to measure the numerical accuracy of the close-encounter, we take the magnitude of the difference between the radius of the activity sphere and the distance between the planet and the asteroid. Hence, we wish to find \( t \) such that the position vector \( r_a \) at time \( t \) satisfies

\[
|\Delta - R_{act}| = ||r_p - r_a||_2 - R_{act} = 0.
\]

We obtain a first estimate by taking the linear polynomial approximation rather than using the mid-point of the interval. We use linear inverse interpolation and this leads to an initial time estimate \( t_{num} \) for the close-encounters defined by

\[
t_{num} = t_{n-1} + \frac{R_{act} - \Delta_{n-1}}{\Delta_n - \Delta_{n-1}}(t_n - t_{n-1}).
\]

Here, \( t_{n-1} \) and \( t_n \) are the times at the left and right end-points of the interval, and \( \Delta_{n-1} \) and \( \Delta_n \) are the distances between the planet and the asteroid obtained at \( t_{n-1} \) and \( t_n \), respectively. We found that linear inverse interpolation typically leads to an initial estimate that is within \( 10^{-02} \) of the true solution.

The true time \( t_{ref} \) of a close-encounter is determined by using the reference solution obtained in quadruple-precision and a numerical accuracy \( TOL_{ce} \) defined as

\[
||r_p - r_a||_2 - R_{act} \leq TOL_{ce}.
\]  
(5.2.1)

We used \( TOL_{ce} = 10^{-15} \) for the reference solution. Once this numerical accuracy is achieved we report this time as the reference close-encounter time \( t_{ref} \). We also store information such as the positions and velocities of the planet and the asteroid, and refer to them as reference positions and reference velocities of the close-encounter. To test the accuracy of the reference solution, we initially integrated a set of 10 quality asteroids using the combination of the \( ERKN101217 \) integrator with the 29-stage interpolant over
a period of one million days and $TOL = 10^{-18}$. We then integrated the same set of asteroids under the same conditions but using a different combination, namely, the $ODEX2$ integrator with its interpolant. For these two different combinations, the average difference in $t_{ref}$ was no more than $0.21 \times 10^{-08}$, which suggests that the solution found using the combination of the $ERKN101217$ integrator and its 29-stage interpolant is sufficiently accurate.

For numerical approximations in double-precision the value of $TOL_{ce}$ has been relaxed to $10^{-10}$. As for quadruple-precision, we also store $t_{num}$ with corresponding positions and velocities to make comparisons between different combinations of the main algorithm.

5.2.2 Admissible close-encounter error

The difference between a true close-encounter for the reference solution and a numerically detected close-encounter for the simulation can be substantial. We illustrate this in

Figure 5.4. The trajectories of the reference solution (obtained in quadruple-precision) of the planet and the asteroid are denoted by solid lines. The orbits of the numerical solution (obtained in double-precision) are denoted by dotted lines. We assume that the
positions of both planet and asteroid contain errors from different sources, for example, due to integration errors or interpolation errors. Hence, the position of the planet may deviate from its true location in a different way than the position of the asteroid. This means that the detected close-encounter, marked as B in Figure 5.4, could not only lie in quite a different location, but also occurs at a substantially different time with respect to the true close-encounter, marked as A in Figure 5.4.

![Figure 5.5: An illustration of an admissible close-encounter error between double and quadruple-precision orbits, where the positions of the planet and asteroid are represented by blue and red dots, respectively and circles represent the activity spheres.](image)

We use the close-encounter time as the measure to define whether a close-encounter is admissible or not. An admissible close-encounter corresponds to a true close-encounter, but we may also detect what we call fake close-encounters that do not correspond to a true close-encounter. Figure 5.5 illustrates the concept of an admissible close-encounter error between double- and quadruple-precision orbits, where the positions of the planet and asteroid are represented by blue and red dots, respectively. There is no unique definition or criterion for deciding if a close-encounter is admissible. Since the time-step is related to the tolerance, one possible criterion is to fix a large value, for example, 50 days throughout a simulation; any close-encounter time within 50 days of an actual close-encounter time would then count as a close-encounter. The drawback of this criterion is that 50 days
over a period of one million days is a large value. On the other hand, if the fixed value is small, say 5 days, over the same period of integration then this criterion is not justified towards the end of the integration time, because the integration error will be too large at the end of the integration. Indeed, we expect that the close-encounters error in time will increase with the total integration time, because the error in the positions of the planets and asteroids grows with the integration time. This means that, for sufficiently large \( t \), all double-precision close-encounter positions and their corresponding close-encounter times will be unacceptable, that is, we would record the close-encounter, but the associated time of close-encounter would not correspond to the true time of close-encounter. This also implies a limit on the integration time.

Since the error in the position grows as \( t^2 \) and the average time-step \( h_{avg} \) is proportional to the \( p^{th} \) root of the tolerance, i.e., \((TOL)^{\frac{1}{p}}\), we implemented another possible criterion by using the following formula

\[
|t_{ref} - t_{num}| = C \max(1, t^2) h_{avg}^p.
\]

Here, \( p \) is the order of that particular integrator, and \( C \) is a constant. If this close-encounter error in time is within the limit of the admissible close-encounter error in time then we count it as a true close-encounter. Otherwise, the close-encounter is referred to as a fake close-encounter.

Figure 5.6 shows the close-encounter errors in time. The experiment has been performed with 1000 quality asteroids using the combination of the \( ERKN101217 \) integrator and its 29-stage interpolant for both planets and asteroids over one million days with a local error tolerance of \( TOL = 10^{-16} \). The reference solution was obtained in quadruple-precision using tolerance \( TOL = 10^{-18} \). The experiment has been performed in such a way that an asteroid is removed from the simulation once it makes a first close-encounter with a Jovian planet. We observe that the close-encounter error in time varies considerably (approximately 12 orders of magnitude difference) among the quality asteroids. As expected, the deviation in time grows with the total integrated time.
Figure 5.6: The close-encounter error $|t_{\text{ref}} - t_{\text{num}}|$ verses time for a set of 1000 quality asteroids using the \textit{ERKN101217} integrator with its 29-stage interpolant over one million days for a local error tolerance of $10^{-16}$.

Let us compare this experiment with the one illustrated in Figure 5.2, where the difference between the errors in the positions of asteroids 1 and 3 (denoted by AST1 and AST3, respectively) is approximately three orders of magnitude. The close-encounter errors in the positions of these same two asteroids are now $5.66 \times 10^{-07}$ and $3.52 \times 10^{-04}$, respectively, which is again a difference of almost three orders of magnitude. Similar behavior has been observed for the errors in the velocities. We conclude that the close-encounter error in time is primarily induced by the deviation in terms of positions (and velocities), which depend strongly upon the stability of the individual asteroids.

### 5.3 Close-encounters at a given numerical accuracy

While we use the close-encounter error in time as our main measure, we also monitor positions and velocities at the moment of close-encounter, to be able to differentiate further between the performances of the various combinations of integrators and interpolations in the main algorithm.
Let us first consider only the close-encounter error in time to distinguish between different combinations for the main algorithm. To this end, we perform the first set of experiments with 1000 quality asteroids, and integrate using a combination with local error tolerances such that we attain an accuracy of $10^{-04}$ for the maximum global error in position. The local error tolerances for the variable-step-size integrators $ERKN689$, $ERKN101217$, and $ODEX2$ were $TOL = 10^{-10}, 10^{-11},$ and $10^{-16}$, respectively. For these experiments, we used 58 plausible combinations of different integrators, interpolation schemes, and nonlinear equation solvers out of the approximately 600 possible combinations.

<table>
<thead>
<tr>
<th>Integrator Planets</th>
<th>Interpolation Planets</th>
<th>Integrator Asteroids</th>
<th>Interpolation Asteroids</th>
<th>Total CEs</th>
<th>True CEs</th>
<th>Missed CEs</th>
<th>Fake CEs</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ERKN689$ quintic</td>
<td>$ERKN689$ 2-step</td>
<td>$ERKN689$ 12-stage</td>
<td>$ERKN689$ quintic</td>
<td>1000</td>
<td>1000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$ERKN101217$ 2-step</td>
<td>$ERKN101217$ 2-step</td>
<td>$ERKN101217$ 23-stage</td>
<td>$ERKN101217$ 26-stage</td>
<td>999</td>
<td>997</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>$ERKN101217$ 3-step</td>
<td>$ERKN101217$ 3-step</td>
<td>$ERKN101217$ 23-stage</td>
<td>$ERKN101217$ 26-stage</td>
<td>1000</td>
<td>996</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>$ERKN101217$ 23-stage</td>
<td>$ERKN101217$ 23-stage</td>
<td>$ERKN101217$ 26-stage</td>
<td>$ERKN101217$ 29-stage</td>
<td>1000</td>
<td>1000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$ODEX2$ interpolant</td>
<td>$ODEX2$ interpolant</td>
<td>$ODEX2$ interpolant</td>
<td>$ODEX2$ interpolant</td>
<td>1000</td>
<td>1000</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.1: An analysis of detecting close-encounters using plausible combinations of the main algorithm for a maximum global error of $10^{-04}$. Only those combinations that detect at least 99% true close-encounters are listed.

Table 5.1 shows only those plausible combinations that detect at least 99% true close-encounters over one million days. The columns in Table 5.1 are as follows:

Column 1: the integrators used for the integration of the Jovian planets;

Column 2: the interpolation schemes used for the continuous approximation of the Jovian planets;
Column 3: the integrators used for the integration of asteroids;

Column 4: the interpolation schemes for the continuous approximation of asteroids;

Column 5: the total number of detected close-encounters;

Column 6: the total number of true close-encounters;

Column 7: the total number of missed close-encounters; and

Column 8: the total number of fake close-encounters.

Most combinations detect exactly 1000 close-encounters, and all detections are true close-encounters (CEs). The combination of the $ERKN101217$ integrator with the two-step Hermite interpolation scheme detects 999 close-encounters. However, of these 999 close-encounters, only 997 are true close encounters ($|t_{\text{ref}} - t_{\text{num}}|$ is within the prescribed range) and two are, in fact, fake close-encounters. Therefore, this combination has missed three close-encounters. Also, the combination of the $ERKN101217$ integrator with the three-step Hermite interpolation scheme seems to have detected 100% close-encounters, but only 996 are true close-encounters. Hence, this combination missed four close-encounters, and detected four different fake close-encounters.

In order to distinguish combinations that detect all 1000 true close-encounters, we perform further numerical experiments to decide which combination has the best accuracy.

Let us first focus on the value of the close-encounter error in time as a measure of accuracy for different combinations. Figure 5.7 shows histograms of the numbers of true close-encounters against the close-encounter errors in time using two combinations, namely, $ERKN689$ with quintic Hermite interpolation and $ERKN689$ with its 12-stage interpolant. Both combinations detect 100% true close-encounters but they exhibit considerable differences in close-encounter time errors. For $ERKN689$ with quintic Hermite interpolation, the maximum and the minimum close-encounter errors are 8.50 and $8.89 \times 10^{-02}$, respectively. For $ERKN689$ with its 12-stage interpolant the maximum and
Figure 5.7: Close-encounter errors in time for a given integration accuracy of $10^{-04}$. Shown are the combinations of ERKN689 with quintic Hermite interpolation (top) and ERKN689 with its 12-stage interpolant (bottom).
the minimum errors are 2.634 and $1.33 \times 10^{-07}$, respectively. The red curve in Figure 5.7 is the normal distribution curve and we found reasonably good agreement, especially with the combination of \textit{ERKN689} and quintic Hermite interpolation. There is also a reasonably good agreement in standard deviation for \textit{ERKN689} with its 12-stage interpolant, but not in terms of the mean; see Figure 5.7 (bottom).

<table>
<thead>
<tr>
<th>Integrator Planets</th>
<th>Interpolation Planets</th>
<th>Integrator Planets</th>
<th>Interpolation Planets</th>
<th>Maximum Time-diff</th>
<th>Average Time-diff</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{ERKN689}</td>
<td>quintic</td>
<td>\textit{ERKN689}</td>
<td>quintic</td>
<td>8.500</td>
<td>0.317$\times 10^{+1}$</td>
<td>3m48s</td>
</tr>
<tr>
<td>\textit{ERKN689}</td>
<td>2-step</td>
<td>\textit{ERKN689}</td>
<td>2-step</td>
<td>10.079</td>
<td>0.334$\times 10^{+1}$</td>
<td>+107%</td>
</tr>
<tr>
<td>\textit{ERKN689}</td>
<td>12-stage</td>
<td>\textit{ERKN689}</td>
<td>12-stage</td>
<td>2.634</td>
<td>0.104$\times 10^{+0}$</td>
<td>+272%</td>
</tr>
<tr>
<td>\textit{ERKN101217}</td>
<td>23-stage</td>
<td>\textit{ERKN101217}</td>
<td>23-stage</td>
<td>1.033</td>
<td>0.171$\times 10^{−2}$</td>
<td>+94%</td>
</tr>
<tr>
<td>\textit{ERKN101217}</td>
<td>26-stage</td>
<td>\textit{ERKN101217}</td>
<td>26-stage</td>
<td>1.063</td>
<td>0.181$\times 10^{−2}$</td>
<td>+130%</td>
</tr>
<tr>
<td>\textit{ERKN101217}</td>
<td>29-stage</td>
<td>\textit{ERKN101217}</td>
<td>29-stage</td>
<td>1.114</td>
<td>0.177$\times 10^{−2}$</td>
<td>+227%</td>
</tr>
<tr>
<td>\textit{ODEX2}</td>
<td>interpolant</td>
<td>\textit{ODEX2}</td>
<td>interpolant</td>
<td>3.560</td>
<td>0.825$\times 10^{−2}$</td>
<td>+178%</td>
</tr>
<tr>
<td>\textit{$\bar{S}$-13}</td>
<td>quintic</td>
<td>\textit{ERKN689}</td>
<td>12-stage</td>
<td>2.865</td>
<td>0.112$\times 10^{+0}$</td>
<td>+55%</td>
</tr>
<tr>
<td>\textit{$\bar{S}$-13}</td>
<td>quintic</td>
<td>\textit{ERKN101217}</td>
<td>23-stage</td>
<td>1.172</td>
<td>0.602$\times 10^{−2}$</td>
<td>+6%</td>
</tr>
<tr>
<td>\textit{$\bar{S}$-13}</td>
<td>quintic</td>
<td>\textit{ODEX2}</td>
<td>interpolant</td>
<td>2.300</td>
<td>0.730$\times 10^{−2}$</td>
<td>-27%</td>
</tr>
</tbody>
</table>

Table 5.2: The close-encounter errors in time using those combinations of the main algorithm that detect 100% true close-encounters. The local error tolerance is chosen so the maximum global error is approximately $10^{-04}$.

Another important factor with the combination of \textit{ERKN689} and its 12-stage interpolant is that approximately 650 close-encounters have absolute errors less than or equal to $10^{-02}$. This is a far greater number of close-encounters than that for the combination of \textit{ERKN689} with quintic interpolation. Therefore, we consider both the maximum and the average close-encounter errors as a measure of accuracy for the combinations in the main algorithm.

Table 5.2 shows close-encounter errors in time for those combinations that detect 100% true close-encounters. We considered the maximum and average close-encounter errors together with the CPU-time. If we use the \textit{ERKN689} integrator for the integration of the Jovian planets, then the combination of \textit{ERKN689} and its 12-stage interpolant
achieves the smallest average close-encounter error in time, but also consumes approximately 2.1 and 3.7 times more CPU-time than the combinations of ERKN689 with quintic and 2-step Hermite interpolation schemes, respectively. For the ERKN101217 integrator, the average close-encounter error in time is almost the same with all three interpolants. However, as far as the CPU-time is considered, the combination with its 23-stage interpolant is the cheapest option.

If we use the $\tilde{S}$-13 integrator for the planets, then the combination with ERKN101217 and its 23-stage interpolant gives the best accuracy. The cheapest option (least CPU-time) is the combination of $\tilde{S}$-13 and ODEX2 with its interpolant, which is approximately 31% cheaper, but loses approximately 21% accuracy to the combination of $\tilde{S}$-13 with ERKN101217. For all different combinations listed in Table 5.2, the smallest average close-encounter error in time has been achieved by the combination of ERKN101217 and its 23-stage interpolant.

<table>
<thead>
<tr>
<th>Integrator Planets</th>
<th>Interpolation Planets</th>
<th>Integrator Planets</th>
<th>Interpolation Planets</th>
<th>Maximum P-error</th>
<th>Average P-error</th>
<th>Maximum A-error</th>
<th>Average A-error</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERKN689 quintic</td>
<td>ERKN689 quintic</td>
<td>0.621</td>
<td>9.53×10^{-4}</td>
<td>0.046</td>
<td>1.77×10^{-3}</td>
<td>0.131</td>
<td>1.86×10^{-3}</td>
</tr>
<tr>
<td>ERKN689 2-step</td>
<td>ERKN689 2-step</td>
<td>0.621</td>
<td>9.56×10^{-4}</td>
<td>0.056</td>
<td>1.86×10^{-3}</td>
<td>0.003</td>
<td>1.64×10^{-3}</td>
</tr>
<tr>
<td>ERKN689 12-stage</td>
<td>ERKN689 12-stage</td>
<td>0.032</td>
<td>1.08×10^{-4}</td>
<td>0.015</td>
<td>6.47×10^{-4}</td>
<td>0.017</td>
<td>6.12×10^{-4}</td>
</tr>
<tr>
<td>ERKN101217 23-stage</td>
<td>ERKN101217 23-stage</td>
<td>0.003</td>
<td>9.47×10^{-5}</td>
<td>0.007</td>
<td>1.06×10^{-4}</td>
<td>0.012</td>
<td>1.12×10^{-4}</td>
</tr>
<tr>
<td>ERKN101217 26-stage</td>
<td>ERKN101217 26-stage</td>
<td>0.003</td>
<td>1.13×10^{-4}</td>
<td>0.007</td>
<td>1.12×10^{-4}</td>
<td>0.008</td>
<td>1.10×10^{-4}</td>
</tr>
<tr>
<td>ERKN101217 29-stage</td>
<td>ERKN101217 29-stage</td>
<td>0.003</td>
<td>1.10×10^{-4}</td>
<td>0.008</td>
<td>1.10×10^{-4}</td>
<td>0.008</td>
<td>1.10×10^{-4}</td>
</tr>
<tr>
<td>ODEX2 interpolant</td>
<td>ODEX2 interpolant</td>
<td>0.014</td>
<td>5.72×10^{-3}</td>
<td>0.024</td>
<td>4.91×10^{-2}</td>
<td>0.017</td>
<td>5.33×10^{-2}</td>
</tr>
<tr>
<td>$\tilde{S}$-13 quintic</td>
<td>$\tilde{S}$-13 quintic</td>
<td>0.032</td>
<td>1.10×10^{-4}</td>
<td>0.015</td>
<td>6.09×10^{-4}</td>
<td>0.006</td>
<td>5.33×10^{-4}</td>
</tr>
<tr>
<td>$\tilde{S}$-13 quintic</td>
<td>$\tilde{S}$-13 quintic</td>
<td>0.001</td>
<td>6.44×10^{-4}</td>
<td>0.006</td>
<td>5.33×10^{-4}</td>
<td>0.008</td>
<td>5.33×10^{-4}</td>
</tr>
<tr>
<td>$\tilde{S}$-13 quintic</td>
<td>$\tilde{S}$-13 quintic</td>
<td>0.030</td>
<td>7.86×10^{-4}</td>
<td>0.012</td>
<td>4.03×10^{-3}</td>
<td>0.008</td>
<td>4.03×10^{-3}</td>
</tr>
</tbody>
</table>

Table 5.3: The close-encounter errors in the positions and velocities of the planets and asteroids. Integrations are done for a maximum global error of 10^{-64}. Shown are close-encounter errors in positions (top) and velocities (bottom).
We investigate these results further in Table 5.3, where we also evaluate close-encounter errors in the positions and velocities of the planets and the asteroids. Table 5.3 shows maximum and average close-encounter errors in the positions and velocities of planets (P-error) and asteroids (A-error) using the same combinations as in Table 5.2. If the average close-encounter error in the position is used as a measure of accuracy, then the combination of \texttt{ERKN101217} with its 23-stage interpolant achieves the best accuracy. This combination also achieves smallest close-encounter errors in time. The same applies to close-encounter errors in the velocities of the planets and asteroids. Therefore, the overall conclusion is still that the best accuracy is achieved by the combination of \texttt{ERKN101217} and its interpolants; there is not much difference between the interpolants, but the combination with its 23-stage interpolant is recommended because it uses least CPU-time.

<table>
<thead>
<tr>
<th>Integrator</th>
<th>Planets</th>
<th>Interpolation</th>
<th>Planets</th>
<th>Integrator</th>
<th>Interpolation</th>
<th>Planets</th>
<th>Interpolation</th>
<th>Maximum Time-diff</th>
<th>Average Time-diff</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{ERKN689}</td>
<td>quintic</td>
<td>\texttt{ERKN689}</td>
<td>quintic</td>
<td>\texttt{ERKN689}</td>
<td>quintic</td>
<td>\texttt{ERKN689}</td>
<td>12.649</td>
<td>0.451$\times 10^{-1}$</td>
<td>2m53s</td>
<td></td>
</tr>
<tr>
<td>\texttt{ERKN689}</td>
<td>2-step</td>
<td>\texttt{ERKN689}</td>
<td>2-step</td>
<td>\texttt{ERKN689}</td>
<td>2-step</td>
<td>\texttt{ERKN689}</td>
<td>15.635</td>
<td>0.483$\times 10^{-1}$</td>
<td>+102%</td>
<td></td>
</tr>
<tr>
<td>\texttt{ERKN689}</td>
<td>12-stage</td>
<td>\texttt{ERKN689}</td>
<td>12-stage</td>
<td>\texttt{ERKN689}</td>
<td>12-stage</td>
<td>\texttt{ERKN689}</td>
<td>5.638</td>
<td>0.214$\times 10^{-1}$</td>
<td>+275%</td>
<td></td>
</tr>
<tr>
<td>\texttt{ERKN101217}</td>
<td>23-stage</td>
<td>\texttt{ERKN101217}</td>
<td>23-stage</td>
<td>\texttt{ERKN101217}</td>
<td>23-stage</td>
<td>\texttt{ERKN101217}</td>
<td>4.655</td>
<td>0.152$\times 10^{-1}$</td>
<td>+150%</td>
<td></td>
</tr>
<tr>
<td>\texttt{ERKN101217}</td>
<td>26-stage</td>
<td>\texttt{ERKN101217}</td>
<td>26-stage</td>
<td>\texttt{ERKN101217}</td>
<td>26-stage</td>
<td>\texttt{ERKN101217}</td>
<td>2.756</td>
<td>0.113$\times 10^{-1}$</td>
<td>+188%</td>
<td></td>
</tr>
<tr>
<td>\texttt{ERKN101217}</td>
<td>29-stage</td>
<td>\texttt{ERKN101217}</td>
<td>29-stage</td>
<td>\texttt{ERKN101217}</td>
<td>29-stage</td>
<td>\texttt{ERKN101217}</td>
<td>2.371</td>
<td>0.104$\times 10^{-1}$</td>
<td>+312%</td>
<td></td>
</tr>
<tr>
<td>\texttt{ODEX2}</td>
<td>interpolant</td>
<td>\texttt{ODEX2}</td>
<td>interpolant</td>
<td>\texttt{ODEX2}</td>
<td>interpolant</td>
<td>\texttt{ODEX2}</td>
<td>2.078</td>
<td>0.428$\times 10^{-2}$</td>
<td>+256%</td>
<td></td>
</tr>
<tr>
<td>\texttt{S-13}</td>
<td>quintic</td>
<td>\texttt{ERKN689}</td>
<td>12-stage</td>
<td>\texttt{ERKN689}</td>
<td>12-stage</td>
<td>\texttt{ERKN689}</td>
<td>5.638</td>
<td>0.218$\times 10^{-1}$</td>
<td>+55%</td>
<td></td>
</tr>
<tr>
<td>\texttt{S-13}</td>
<td>quintic</td>
<td>\texttt{ERKN101217}</td>
<td>23-stage</td>
<td>\texttt{ERKN101217}</td>
<td>23-stage</td>
<td>\texttt{ERKN101217}</td>
<td>3.064</td>
<td>0.137$\times 10^{-1}$</td>
<td>+34%</td>
<td></td>
</tr>
<tr>
<td>\texttt{S-13}</td>
<td>quintic</td>
<td>\texttt{ODEX2}</td>
<td>interpolant</td>
<td>\texttt{ODEX2}</td>
<td>interpolant</td>
<td>\texttt{ODEX2}</td>
<td>1.376</td>
<td>0.607$\times 10^{-2}$</td>
<td>-16%</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.4: Close-encounter errors in time using those combinations of the main algorithm that detect 100% true close-encounters. Integrations are done for a maximum global error of $10^{-03}$.

We performed a second set of experiments to investigate how our conclusions depend on the given integration accuracy. To this end, we again used 1000 quality asteroids, but choose the local error tolerance so that the maximum global error is approximately $10^{-03}$. The approximate values of local error tolerances for the variable-step-size integrators \texttt{ERKN689}, \texttt{ERKN101217}, and \texttt{ODEX2} are now $10^{-09}$, $10^{-10}$, and $10^{-15}$, respectively. The results for those combinations that detect 100% true close-encounters are shown in Table 5.4. If we compare these results with Table 5.3, we observe that the overall conclusion has changed. In Table 5.4, the best accuracy is achieved by the combination of the \texttt{ODEX2}
integrator with its interpolant. The second best accuracy is achieved by the combination of \( \tilde{S} \)-13 and \( ODEX2 \) with its interpolant. This second best combination is approximately 4.2 times cheaper but loses approximately 42% accuracy (average close-encounter error in time) compared with the best combination of \( ODEX2 \) and its interpolant.

<table>
<thead>
<tr>
<th>Integrator Planets</th>
<th>Interpolation Planets</th>
<th>Integrator Asteroids</th>
<th>Interpolation Asteroids</th>
<th>Maximum P-error</th>
<th>Average P-error</th>
<th>Maximum A-error</th>
<th>Average A-error</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERKN689</td>
<td>quintic</td>
<td>ERKN689</td>
<td>quintic</td>
<td>0.868</td>
<td>0.136×10(^{-3})</td>
<td>0.073</td>
<td>0.252×10(^{-3})</td>
</tr>
<tr>
<td>ERKN689</td>
<td>2-step</td>
<td>ERKN689</td>
<td>2-step</td>
<td>0.867</td>
<td>0.137×10(^{-3})</td>
<td>0.090</td>
<td>0.269×10(^{-3})</td>
</tr>
<tr>
<td>ERKN689</td>
<td>12-stage</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>0.148</td>
<td>0.222×10(^{-2})</td>
<td>0.043</td>
<td>0.135×10(^{-2})</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>23-stage</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>0.057</td>
<td>0.131×10(^{-3})</td>
<td>0.032</td>
<td>0.901×10(^{-4})</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>26-stage</td>
<td>ERKN101217</td>
<td>26-stage</td>
<td>0.046</td>
<td>0.120×10(^{-3})</td>
<td>0.016</td>
<td>0.634×10(^{-4})</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>29-stage</td>
<td>ERKN101217</td>
<td>29-stage</td>
<td>0.039</td>
<td>0.107×10(^{-3})</td>
<td>0.014</td>
<td>0.592×10(^{-4})</td>
</tr>
<tr>
<td>ODEX2</td>
<td>interpolant</td>
<td>ODEX2</td>
<td>interpolant</td>
<td>0.011</td>
<td>0.318×10(^{-4})</td>
<td>0.014</td>
<td>0.256×10(^{-4})</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>0.148</td>
<td>0.234×10(^{-2})</td>
<td>0.043</td>
<td>0.137×10(^{-2})</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>0.040</td>
<td>0.123×10(^{-3})</td>
<td>0.015</td>
<td>0.760×10(^{-4})</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ODEX2</td>
<td>interpolant</td>
<td>0.023</td>
<td>0.674×10(^{-4})</td>
<td>0.012</td>
<td>0.403×10(^{-4})</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Integrator Planets</th>
<th>Interpolation Planets</th>
<th>Integrator Asteroids</th>
<th>Interpolation Asteroids</th>
<th>Maximum P-error</th>
<th>Average P-error</th>
<th>Maximum A-error</th>
<th>Average A-error</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERKN689</td>
<td>quintic</td>
<td>ERKN689</td>
<td>quintic</td>
<td>0.836×10(^{-3})</td>
<td>0.154×10(^{-3})</td>
<td>0.691×10(^{-4})</td>
<td>0.152×10(^{-4})</td>
</tr>
<tr>
<td>ERKN689</td>
<td>2-step</td>
<td>ERKN689</td>
<td>2-step</td>
<td>0.827×10(^{-3})</td>
<td>0.156×10(^{-3})</td>
<td>0.814×10(^{-4})</td>
<td>0.162×10(^{-4})</td>
</tr>
<tr>
<td>ERKN689</td>
<td>12-stage</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>0.756×10(^{-4})</td>
<td>0.239×10(^{-4})</td>
<td>0.196×10(^{-4})</td>
<td>0.459×10(^{-5})</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>23-stage</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>0.681×10(^{-4})</td>
<td>0.213×10(^{-4})</td>
<td>0.196×10(^{-4})</td>
<td>0.556×10(^{-5})</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>26-stage</td>
<td>ERKN101217</td>
<td>26-stage</td>
<td>0.413×10(^{-5})</td>
<td>0.156×10(^{-6})</td>
<td>0.939×10(^{-7})</td>
<td>0.398×10(^{-7})</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>29-stage</td>
<td>ERKN101217</td>
<td>29-stage</td>
<td>0.455×10(^{-6})</td>
<td>0.144×10(^{-6})</td>
<td>0.808×10(^{-7})</td>
<td>0.369×10(^{-7})</td>
</tr>
<tr>
<td>ODEX2</td>
<td>interpolant</td>
<td>ODEX2</td>
<td>interpolant</td>
<td>0.304×10(^{-6})</td>
<td>0.606×10(^{-7})</td>
<td>0.860×10(^{-8})</td>
<td>0.162×10(^{-8})</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>0.736×10(^{-4})</td>
<td>0.246×10(^{-4})</td>
<td>0.232×10(^{-5})</td>
<td>0.457×10(^{-5})</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>0.476×10(^{-4})</td>
<td>0.187×10(^{-5})</td>
<td>0.871×10(^{-6})</td>
<td>0.513×10(^{-6})</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ODEX2</td>
<td>interpolant</td>
<td>0.205×10(^{-4})</td>
<td>0.791×10(^{-5})</td>
<td>0.540×10(^{-5})</td>
<td>0.241×10(^{-5})</td>
</tr>
</tbody>
</table>

Table 5.5: The close-encounter errors in the positions and velocities of the planets and asteroids. The integrations are done for a maximum global error of \( 10^{-03} \). Shown are close-encounter errors in positions (top) and velocities (bottom).

Table 5.5 shows the maximum and average close-encounter errors in the positions and velocities of the planets and asteroids; compare with Table 5.2. For a maximum global error of \( 10^{-3} \), we again reach the same conclusions for best accuracy not only in time, but also in positions and velocities: at this integration accuracy, the combination of \( ODEX2 \) with its interpolant achieves the smallest error in terms of time and it also achieves the smallest maximum global error for close-encounter errors in the positions and velocities of the planets and asteroids. Similarly, the combination of \( \tilde{S} \)-13 and \( ODEX2 \), which achieves the second best accuracy for the close-encounter error in time, also achieves the second-best accuracy for the close-encounter errors in positions and velocities of the planets and asteroids.
5.4 Best observed numerical accuracy

Finally, we conduct a set of experiments with 1000 quality asteroids such that the smallest maximum global error is attained by each of the integrators; see also Figure 2.1. The local error tolerances for the variable-step-size integrators ERKN689, ERKN101217, and ODEX2 are set to $TOL = 10^{-14}, 10^{-16},$ and $10^{-16}$, respectively. Table 5.6 shows the close-encounter errors in time for all plausible combinations that detect 100% true close-encounters. Unlike the other tables, the combinations of ERKN101217 with the 2-step and 3-step Hermite interpolation schemes have been included here, because these combinations detected 100% true close-encounters at the prescribed integration accuracy. However, they are not as accurate as the combinations of ERKN101217 with its interpolants.

<table>
<thead>
<tr>
<th>Integrator Planets</th>
<th>Interpolation Planets</th>
<th>Integrator Asteroids</th>
<th>Interpolation Asteroids</th>
<th>Maximum Time-diff</th>
<th>Average Time-diff</th>
<th>CPU Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERKN689</td>
<td>quintic</td>
<td>ERKN689</td>
<td>quintic</td>
<td>2.117</td>
<td>0.770×10^{-10}</td>
<td>13m21s</td>
</tr>
<tr>
<td>ERKN689</td>
<td>2-step</td>
<td>ERKN689</td>
<td>2-step</td>
<td>2.958</td>
<td>0.819×10^{-10}</td>
<td>+102%</td>
</tr>
<tr>
<td>ERKN689</td>
<td>12-stage</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>0.238</td>
<td>0.666×10^{-2}</td>
<td>+268%</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>2-step</td>
<td>ERKN101217</td>
<td>2-step</td>
<td>7.590</td>
<td>0.172×10^{-11}</td>
<td>+1%</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>3-step</td>
<td>ERKN101217</td>
<td>3-step</td>
<td>11.65</td>
<td>0.180×10^{-11}</td>
<td>+67%</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>23-stage</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>0.088</td>
<td>0.264×10^{-3}</td>
<td>+55%</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>26-stage</td>
<td>ERKN101217</td>
<td>26-stage</td>
<td>0.093</td>
<td>0.300×10^{-3}</td>
<td>+92%</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>29-stage</td>
<td>ERKN101217</td>
<td>29-stage</td>
<td>0.067</td>
<td>0.238×10^{-3}</td>
<td>+212%</td>
</tr>
<tr>
<td>ODEX2</td>
<td>interpolant</td>
<td>ODEX2</td>
<td>interpolant</td>
<td>3.560</td>
<td>0.825×10^{-2}</td>
<td>-21%</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>1.123</td>
<td>0.119×10^{-11}</td>
<td>+51%</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>1.157</td>
<td>0.594×10^{-2}</td>
<td>-15%</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ODEX2</td>
<td>interpolant</td>
<td>2.300</td>
<td>0.730×10^{-2}</td>
<td>-79%</td>
</tr>
<tr>
<td>ERKN689</td>
<td>12-stage</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>0.028</td>
<td>0.871×10^{-4}</td>
<td>+41%</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>23-stage</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>0.058</td>
<td>0.175×10^{-3}</td>
<td>+144%</td>
</tr>
</tbody>
</table>

Table 5.6: The close-encounter errors in time at best observed integration accuracy using those combinations of the main algorithm that detect 100% true close-encounters.

For the ERKN689 integrator, the combination with its 12-stage interpolant achieves the best accuracy. For the ERKN101217 integrator, the combination with its 29-stage interpolant achieve very similar accuracy to the combination with its 23-stage interpolant. However, we recommend using the combination with 23-stage interpolant, because it uses approximately half the CPU-time of the combination with the 29-stage interpolant. We also investigated two new combinations, namely, ERKN689 with its 12-stage interpolant
for the planets and $ERKN101217$ with its 23-stage interpolant for the asteroids; and $ERKN101217$ with its 23-stage interpolant for the planets and $ERKN689$ with its 12-stage interpolant for the asteroids. The motivation behind these mixed combinations is the fact that the orbits of planets have low eccentricity, but for asteroids they could have high eccentricity. So, fundamentally, there are two different types of particles. We find that the addition of new combinations is worthwhile, because the best accuracy is now achieved by the combination of $ERKN689$ for the planets and $ERKN101217$ for the asteroids, whereas the second best accuracy is achieved by the opposite combination of $ERKN101217$ for the planets and $ERKN689$ for the asteroids. We obtain similarly improved results using other mixed combinations, but these do not achieve the best accuracy, and we omit further details.

Let us now investigate the close-encounter errors in the positions and velocities of the planets and asteroids when integrating such that the smallest maximum global error is attained; the results are shown in Table 5.7. As in the previous sets of experiments, the combinations that achieved the best and second best accuracy for the close-encounters in time, maintain their respective categories for the close-encounter errors in the positions and velocities of the planets and asteroids. Hence, again the two combinations of $ERKN689$ with 12-stage for the planets and $ERKN101217$ with 23-stage for the asteroids, and $ERKN101217$ with 23-stage for the planets and $ERKN689$ with 12-stage interpolant for the asteroids are best and second best, respectively.

For the same sets of combinations as listed in Table 5.6, we performed experiments to measure other parameters, such as, accepted (Accptd) and rejected (Rejctd) time-steps, the number of function (Fcn-evl) and polynomial (Poly-evl) evaluations for both planets and asteroids, and the total number of iterations (Itratns) used to solve equation (5.2.1). These experiments have been performed with only 100 quality asteroids over one million days for the same local error tolerances as in the previous sets of experiments. For all the integrators, not a single planetary rejected time-step was recorded. In contrast, the integrators used to integrate the asteroids do reject time-steps, and biggest ratio between
<table>
<thead>
<tr>
<th>Integrator Planets</th>
<th>Interpolation Planets</th>
<th>Integrator Planets</th>
<th>Interpolation Planets</th>
<th>Maximum P-error</th>
<th>Average P-error</th>
<th>Maximum A-error</th>
<th>Average A-error</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERKN689</td>
<td>quintic</td>
<td>ERKN689</td>
<td>quintic</td>
<td>0.128</td>
<td>0.238×10⁻¹</td>
<td>0.011</td>
<td>0.429×10⁻²</td>
</tr>
<tr>
<td>ERKN689</td>
<td>2-step</td>
<td>ERKN689</td>
<td>2-step</td>
<td>0.137</td>
<td>0.242×10⁻¹</td>
<td>0.015</td>
<td>0.456×10⁻²</td>
</tr>
<tr>
<td>ERKN689</td>
<td>12-stage</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>0.004</td>
<td>0.683×10⁻⁴</td>
<td>0.002</td>
<td>0.413×10⁻⁴</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>2-step</td>
<td>ERKN101217</td>
<td>2-step</td>
<td>0.425</td>
<td>0.485×10⁻¹</td>
<td>0.044</td>
<td>0.957×10⁻²</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>3-step</td>
<td>ERKN101217</td>
<td>3-step</td>
<td>0.523</td>
<td>0.524×10⁻¹</td>
<td>0.064</td>
<td>0.102×10⁻¹</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>23-stage</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>0.001</td>
<td>0.219×10⁻⁵</td>
<td>0.001</td>
<td>0.160×10⁻⁵</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>26-stage</td>
<td>ERKN101217</td>
<td>26-stage</td>
<td>0.001</td>
<td>0.251×10⁻⁵</td>
<td>0.001</td>
<td>0.183×10⁻⁵</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>29-stage</td>
<td>ERKN101217</td>
<td>29-stage</td>
<td>0.001</td>
<td>0.214×10⁻⁵</td>
<td>0.0004</td>
<td>0.142×10⁻⁵</td>
</tr>
<tr>
<td>ODEX2</td>
<td>interpolant</td>
<td>ODEX2</td>
<td>interpolant</td>
<td>0.014</td>
<td>0.572×10⁻⁴</td>
<td>0.024</td>
<td>0.491×10⁻⁴</td>
</tr>
<tr>
<td>Š-13</td>
<td>quintic</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>0.019</td>
<td>0.125×10⁻³</td>
<td>0.007</td>
<td>0.711×10⁻⁴</td>
</tr>
<tr>
<td>Š-13</td>
<td>quintic</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>0.019</td>
<td>0.641×10⁻⁴</td>
<td>0.007</td>
<td>0.334×10⁻⁴</td>
</tr>
<tr>
<td>Š-13</td>
<td>quintic</td>
<td>ODEX2</td>
<td>interpolant</td>
<td>0.030</td>
<td>0.786×10⁻⁴</td>
<td>0.012</td>
<td>0.403×10⁻⁴</td>
</tr>
<tr>
<td>ERKN689</td>
<td>12-stage</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>0.0001</td>
<td>0.539×10⁻⁶</td>
<td>0.0002</td>
<td>0.525×10⁻⁶</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>23-stage</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>0.001</td>
<td>0.179×10⁻⁵</td>
<td>0.0003</td>
<td>0.105×10⁻⁵</td>
</tr>
</tbody>
</table>

Table 5.7: The close-encounter errors in the positions and velocities of the planets and asteroids integrated at best observed accuracy. Shown are close-encounter errors in positions (top) and velocities (bottom).
<table>
<thead>
<tr>
<th>Integrator</th>
<th>Interpolation</th>
<th>Integrator</th>
<th>Interpolation</th>
<th>Acctd</th>
<th>Acctd</th>
<th>Rejctd</th>
<th>Fcn-evl</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planets</td>
<td>Planets</td>
<td>Asteroids</td>
<td>Asteroids</td>
<td>P-stps</td>
<td>A-stps</td>
<td>A-steps</td>
<td>Planets</td>
</tr>
<tr>
<td>ERKN689</td>
<td>quintic</td>
<td>ERKN689</td>
<td>quintic</td>
<td>58329</td>
<td>1466469</td>
<td>1</td>
<td>524962</td>
</tr>
<tr>
<td>ERKN689</td>
<td>2-step</td>
<td>ERKN689</td>
<td>2-step</td>
<td>58329</td>
<td>1466498</td>
<td>1</td>
<td>524965</td>
</tr>
<tr>
<td>ERKN689</td>
<td>12-stage</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>58329</td>
<td>1466557</td>
<td>3</td>
<td>699949</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>2-step</td>
<td>ERKN101217</td>
<td>2-step</td>
<td>21820</td>
<td>681809</td>
<td>151</td>
<td>370953</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>3-step</td>
<td>ERKN101217</td>
<td>3-step</td>
<td>21820</td>
<td>681911</td>
<td>266</td>
<td>370965</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>23-stage</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>21820</td>
<td>680858</td>
<td>158</td>
<td>501861</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>26-stage</td>
<td>ERKN101217</td>
<td>26-stage</td>
<td>21820</td>
<td>681316</td>
<td>160</td>
<td>567321</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>29-stage</td>
<td>ERKN101217</td>
<td>29-stage</td>
<td>21820</td>
<td>682023</td>
<td>156</td>
<td>632781</td>
</tr>
<tr>
<td>ODEX2</td>
<td>interpolant</td>
<td>ODEX2</td>
<td>interpolant</td>
<td>6730</td>
<td>134302</td>
<td>9212</td>
<td>231595</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>250000</td>
<td>1466643</td>
<td>1</td>
<td>250000</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>250000</td>
<td>680237</td>
<td>151</td>
<td>250000</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ODEX2</td>
<td>interpolant</td>
<td>250000</td>
<td>134801</td>
<td>9085</td>
<td>250000</td>
</tr>
<tr>
<td>ERKN689</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>21820</td>
<td>58329</td>
<td>680929</td>
<td>171</td>
<td>699949</td>
</tr>
<tr>
<td>ERKN101217</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>21820</td>
<td>1466519</td>
<td>3</td>
<td>501861</td>
<td></td>
</tr>
<tr>
<td>ODEX2</td>
<td>interpolant</td>
<td>ODEX2</td>
<td>interpolant</td>
<td>6008133</td>
<td>11604891</td>
<td>5193752</td>
<td>344</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>17599825</td>
<td>77725448</td>
<td>58659980</td>
<td>207</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>15662520</td>
<td>43575507</td>
<td>27232029</td>
<td>221</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ODEX2</td>
<td>interpolant</td>
<td>17717036</td>
<td>45648803</td>
<td>27250351</td>
<td>229</td>
</tr>
<tr>
<td>ERKN689</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>19781419</td>
<td>39340562</td>
<td>27067130</td>
<td>228</td>
<td></td>
</tr>
<tr>
<td>ERKN101217</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>11593420</td>
<td>39340562</td>
<td>27067130</td>
<td>228</td>
<td></td>
</tr>
<tr>
<td>ERKN101217</td>
<td>ERKN101217</td>
<td>2-step</td>
<td>11595320</td>
<td>39340562</td>
<td>27067130</td>
<td>228</td>
<td></td>
</tr>
<tr>
<td>ERKN101217</td>
<td>ERKN101217</td>
<td>3-step</td>
<td>11597209</td>
<td>39340562</td>
<td>27067130</td>
<td>228</td>
<td></td>
</tr>
<tr>
<td>ERKN101217</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>15662520</td>
<td>43575507</td>
<td>27232029</td>
<td>221</td>
<td></td>
</tr>
<tr>
<td>ERKN101217</td>
<td>ERKN101217</td>
<td>26-stage</td>
<td>17717036</td>
<td>45648803</td>
<td>27250351</td>
<td>229</td>
<td></td>
</tr>
<tr>
<td>ERKN101217</td>
<td>ERKN101217</td>
<td>29-stage</td>
<td>19781419</td>
<td>47742178</td>
<td>27278636</td>
<td>228</td>
<td></td>
</tr>
<tr>
<td>ODEX2</td>
<td>interpolant</td>
<td>ODEX2</td>
<td>interpolant</td>
<td>6008133</td>
<td>11604891</td>
<td>5193752</td>
<td>344</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>17599825</td>
<td>77725448</td>
<td>58659980</td>
<td>207</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>15648318</td>
<td>43558673</td>
<td>27207218</td>
<td>226</td>
</tr>
<tr>
<td>S-13</td>
<td>quintic</td>
<td>ODEX2</td>
<td>interpolant</td>
<td>6010242</td>
<td>5342327</td>
<td>5207426</td>
<td>338</td>
</tr>
<tr>
<td>ERKN689</td>
<td>ERKN101217</td>
<td>23-stage</td>
<td>15664374</td>
<td>39340562</td>
<td>27067130</td>
<td>228</td>
<td></td>
</tr>
<tr>
<td>ERKN101217</td>
<td>ERKN689</td>
<td>12-stage</td>
<td>17599825</td>
<td>77723483</td>
<td>58655909</td>
<td>206</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.8: Study of additional accuracy measures using the same combinations and integration tolerances as in Table 5.6.
rejected and accepted time-steps are observed for the combinations that use the $ODEX2$ integrator. We note that this finding supports the fact that $ODEX2$ with $TOL = 10^{-16}$ is possibly affected by round-off error. We also found that the number of iterations needed to solve equation (5.2.1) to an accuracy of $10^{-10}$ is (on average) only two to four; this is independent of the interpolation scheme or interpolant used for any combination in Table 5.7.
The main objective of this thesis was to develop an efficient multirate algorithm for modelling close-encounters in the Solar System. We analysed and compared the efficiency and the error growth for different numerical integrators and interpolation schemes. Our numerical testing involved comparing combinations of different integrators and interpolation schemes on a collection of test problems. This collection of test problems consisted of Kepler’s two-body problem and the more realistic problems involving the Sun and four Gas-giants with and without additional test particles. Throughout the thesis, we examined the growth of the global error in the positions and velocities of the bodies, and the relative error in the energy and angular momentum of the system. In addition, we investigated the close-encounter errors in time, positions and velocities of the bodies.
The N-body simulations were performed over as much as $10^8$ years. In this thesis, we performed extensive investigation of N-body simulations with high-order integrators.

We performed experiments to observe the error growth in the positions and velocities using the variable-step-size integrators ODEX2, ERKN689, and ERKN101217, applied to the Jovian problem over one million years for local error tolerances in the range $10^{-16}$ to $10^{-08}$. Our work has several possible implications. Overall, we observed that for the local error tolerances in the range $TOL = 10^{-16}$ to $10^{-08}$, despite the popularity of the extrapolation integrator, ODEX2 is the least accurate. We found among the explicit Runge-Kutta-Nyström pairs we tested that the smallest possible global error in the integration was obtained using the 6-8 pair and not the 10-12 pair. This was an unexpected because extensive numerical testing of others has shown that increasing the order of a pair usually reduces the smallest possible global error. Our result implies that it would be worthwhile investigating explicit Runge-Kutta Nyström pairs of orders neighbouring the 6-8 pair.

The interpolation schemes play a vital role in our algorithm. We constructed three different types, namely, one-step (cubic and quintic Hermite), two-step and three-step Hermite interpolation schemes. The low-order interpolation schemes are unlikely to be used in practice. We have included them in our testing because we wanted to access the effectiveness of these interpolation schemes. Especially, it makes sense to use a low-order interpolation scheme with a high-order Störmer method when the step-size is four days (for the Jovian problem), because this choice means the interpolation error is below machine precision. We performed many experiments using different interpolation schemes with different integrators both over a short time interval of length $2\pi$ and several long time intervals of duration as long as 100 million years. For short-term simulations, we investigated the performance of these interpolation schemes applied to the Kepler problem over the interval $[0, 2\pi]$ for eccentricities in the range $[0.05, 0.9]$. We observed that the maximum error in position is monotonically increasing as a function of eccentricity. For a given number of sub-intervals we used in this thesis, the higher-order interpolation
schemes achieve better accuracy and for a given interpolation scheme the accuracy improves if the number of sub-intervals is increased. For long term simulations, experiments were performed for the Jovian problem integrated up to $10^8$ years. One of the strongest conclusions from these experiments is that, the order of the continuous approximation should, with one notable exception in this thesis, be compatible with the order of the integrator. The notable exception occurs for higher order Störmer methods when used with small step-sizes. The order of the continuous approximation can be significantly less than the order of the Störmer method. The reason why we get away with low-order continuous approximation is because the $\tilde{S}-13$ integrator uses artificially small step-size of four days to eliminate the truncation error at machine precision and we are just left with the round-off error. However, if we want the truncation error to be $10^{-16}$, then we might use the step-size of sixteen days rather than four days. When we use sixteen days, then the low-order continuous approximation (quintic Hermite interpolation) would fail.

We investigated the dependence of the error on the orbital behaviour of asteroids. To this end, approximately 1100 quality asteroids were selected from the set of 100,000 asteroids. Different sets of experiments with 1000 quality asteroids each were performed to attain the numerical accuracy of $10^{-03}$, $10^{-04}$, and the best observed numerical accuracy by the individual integrator as described in Figure 2.1. The objective of these sets of experiments was to see how our conclusions depend on the given numerical accuracy. All these experiments were performed with 58 plausible combinations of different integrators, interpolation schemes, and non-linear equation solvers. We considered the maximum and the average close-encounter error together with the cost of CPU-time. These experiments illustrated a trade-off between the accuracy and the efficiency. One implication of these trade-offs seems to be that if we want to sacrifice an accuracy, for example, of approximately 21%, we will have to use approximately 31% extra CPU-time, an amount that could be significant for long-term simulations.

We also investigated two new combinations, namely, \textit{ERKN689} with its 12-stage interpolant for the planets and \textit{ERKN101217} with the 23-stage interpolant for the as-
teroids; and vice versa, $ERKN101217$ with the 23-stage interpolant for the planets and $ERKN689$ with its 12-stage interpolant for the asteroids. The motivation behind the mixed combinations was the fact that the orbits of planets have low eccentricity and for asteroids they could have high eccentricity. So, fundamentally, there are two different types of particles. Therefore, two different combinations of integrators and appropriate interpolation schemes could be tested. We observed that the addition of new combinations is worthwhile because the best accuracy was obtained by the combination of $ERKN689$ for the planets and $ERKN101217$ for the asteroids whereas the second best accuracy was attained by the other combination of $ERKN101217$ for the planets and $ERKN689$ for the asteroids. One implication of these results is that it would be worthwhile investigating over mixed combinations, for example, the new ERKN pairs of Sharp [57] with suitable continuous approximations.

Further work can be done by performing accurate integration with the developed multirate algorithm. For example, it is of interest to test these algorithms on the Earth’s five mass extinction events [51]: the Ordovician-Silurian extinction, the late Devonian extinction, the Permian-Triassic extinction, the end Triassic extinction, and the Cretaceous-Tertiary extinction.
Appendix

Table A.1 contains the values for $\mu_j, j = 1, \ldots, 5$, which is the gravitational constant $G$ times the mass $m_j$ of the $j^{th}$ planetary body, namely, Sun, Jupiter, Saturn, Uranus, and Neptune. Table A.2 contains the initial positions and velocities for the Jovian problem; see also Section 1.4.2. Table A.3 contains the initial positions and velocities for the Helin-Roman-Crockett (HRC) problem; see also Section 1.4.3.

Table A.4 contains a set of interpolants for ERKN689. The solution interpolant is order 8 and the derivative interpolant which is the derivative of the solution interpolant is order seven. The interpolants use 12 stages. In Table A.4 we have

$bt(i,j) = \text{the coefficient of } t^i \text{ for the } j^{th} t \text{ for the solution interpolant.}$
bpt(i,j) = the coefficient of \( t^i \) for the \( j^{th} \) b for the derivative interpolant.

### A.1 Jovian problem

\[
\begin{array}{llll}
\mu_1 &=& 0.295912208285591102582 \times 10^{-3}, & \mu_2 = 0.282534210344592625472 \times 10^{-6}, \\
\mu_3 &=& 0.8459468504830659289285 \times 10^{-7}, & \mu_4 = 0.128881623813803488851 \times 10^{-7}, \\
\mu_5 &=& 0.153211248128427618918 \times 10^{-7}.
\end{array}
\]

**Table A.1:** The values \( \mu_j, \) \( j = 1, \ldots, 5, \) for the five bodies ordered from Sun, Jupiter, Saturn, Uranus and Neptune.

<table>
<thead>
<tr>
<th></th>
<th>( x )</th>
<th>( y )</th>
<th>( z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun</td>
<td>( 0.920949868632869 \times 10^{-3} )</td>
<td>( 0.230416603075620 \times 10^{-2} )</td>
<td>( 0.912721704852388 \times 10^{-3} )</td>
</tr>
<tr>
<td>Jupiter</td>
<td>( 0.3350285173564643 \times 10^1 )</td>
<td>( -0.3471457282981824 \times 10^1 )</td>
<td>( -0.157123696488948 \times 10^1 )</td>
</tr>
<tr>
<td>Saturn</td>
<td>( -0.8971584118413477 \times 10^1 )</td>
<td>( 0.2281986174163616 \times 10^1 )</td>
<td>( 0.1331251331416312 \times 10^1 )</td>
</tr>
<tr>
<td>Uranus</td>
<td>( -0.1002083045458687 \times 10^1 )</td>
<td>( 0.1732581263930256 \times 10^2 )</td>
<td>( 0.7605737768120762 \times 10^1 )</td>
</tr>
<tr>
<td>Neptune</td>
<td>( -0.291936978874257 \times 10^2 )</td>
<td>( -0.7716981025967714 \times 10^4 )</td>
<td>( -0.2426332656583918 \times 10^1 )</td>
</tr>
<tr>
<td>Sun</td>
<td>( -0.4665246664531984 \times 10^{-5} )</td>
<td>( -0.3149154564335707 \times 10^{-5} )</td>
<td>( -0.1269852543206254 \times 10^{-5} )</td>
</tr>
<tr>
<td>Jupiter</td>
<td>( 0.5580977920917778 \times 10^{-2} )</td>
<td>( 0.495911982658174 \times 10^{-2} )</td>
<td>( 0.1991007074196164 \times 10^{-2} )</td>
</tr>
<tr>
<td>Saturn</td>
<td>( -0.1862917203661242 \times 10^{-2} )</td>
<td>( -0.4987007735981776 \times 10^{-2} )</td>
<td>( -0.1981527265350456 \times 10^{-2} )</td>
</tr>
<tr>
<td>Uranus</td>
<td>( -0.3959919409682252 \times 10^{-2} )</td>
<td>( -0.379062939672767 \times 10^{-3} )</td>
<td>( -0.1101198433831003 \times 10^{-3} )</td>
</tr>
<tr>
<td>Neptune</td>
<td>( 0.8160828111535530 \times 10^{-3} )</td>
<td>( -0.2775247414144566 \times 10^{-2} )</td>
<td>( -0.1157385882979126 \times 10^{-2} )</td>
</tr>
</tbody>
</table>

**Table A.2:** Rows 1 to 5 are the initial positions and Rows 6 to 10 are the initial velocities for the Jovian problem.
### A.2 HRC problem

<table>
<thead>
<tr>
<th></th>
<th>( x )</th>
<th>( y )</th>
<th>( z )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun</td>
<td>(-0.1597551822288177 \times 10^{-5})</td>
<td>(-0.725409815779006 \times 10^{-5})</td>
<td>(-0.3038348598973975 \times 10^{-7})</td>
</tr>
<tr>
<td>Jupiter</td>
<td>(-0.3109433296611612 \times 10^{-2})</td>
<td>(-0.6477134819096109 \times 10^{-2})</td>
<td>(-0.4357172559451174 \times 10^{-4})</td>
</tr>
<tr>
<td>Saturn</td>
<td>(-0.4717678753258388 \times 10^{-2})</td>
<td>(-0.3413503592855709 \times 10^{-2})</td>
<td>(-0.246925287795303 \times 10^{-3})</td>
</tr>
<tr>
<td>Uranus</td>
<td>(-0.3227888778570112 \times 10^{-2})</td>
<td>(-0.2386568620156009 \times 10^{-2})</td>
<td>(-0.506197878986374 \times 10^{-4})</td>
</tr>
<tr>
<td>Neptune</td>
<td>(-0.315232794479188 \times 10^{-2})</td>
<td>(-0.1931132154404109 \times 10^{-3})</td>
<td>(-0.6952342277721326 \times 10^{-4})</td>
</tr>
<tr>
<td>Comet</td>
<td>(-0.1800219023380088 \times 10^{-2})</td>
<td>(-0.8521337694196810 \times 10^{-2})</td>
<td>(-0.1052106206437703 \times 10^{-3})</td>
</tr>
</tbody>
</table>

Table A.3: Rows 1 to 6 are the initial positions and Rows 7 to 12 are the initial velocities for the HRC problem.

### A.3 Coefficients - ERKN689

The coefficients of stages one to nine are given in Dormand et al. [12].

Abscissae of the appended stages:

\( c(10) = 3 / 7\) , \( c(11) = 4 / 9\) , \( c(12) = 8 / 11\)

Interior weights of the appended stages:

\( a(10,1) = 6848971 / 564950498\)

\( a(10,2) = 0 / 1\)

\( a(10,3) = 273509175 / 5165261696\)

\( a(10,4) = 257094825 / 9039207968\)

\( a(10,5) = -6705423 / 2582630848\)
\[ a(10,6) = \frac{11270925}{903920796} \]
\[ a(10,7) = -\frac{2471425}{516526196} \]
\[ a(10,8) = -\frac{307735148636978}{3512185612130070449} \]
\[ a(10,9) = 73396740280104 / 71677257390409601 \]
\[ a(11,1) = 5114730856 / 398655683181 \]
\[ a(11,2) = 0 / 1 \]
\[ a(11,3) = 3108959600 / 56950811883 \]
\[ a(11,4) = 4781297200 / 132885227727 \]
\[ a(11,5) = 15030448 / 2109289329 \]
\[ a(11,6) = -\frac{97641200}{132885227727} \]
\[ a(11,7) = 10174400 / 56950811883 \]
\[ a(11,8) = 111930646967360 / 216991078945200567 \]
\[ a(11,9) = -\frac{92773831000}{164014420971429} \]
\[ a(11,10) = -\frac{480200}{43046721} \]
\[ a(12,1) = 64357631527142016 / 324204668753372021439 \]
\[ a(12,2) = 0 / 1 \]
\[ a(12,3) = 78383303053694633600 / 756477560424534716691 \]
\[ a(12,4) = 444842790865778713600 / 5295342922971743016837 \]
\[ a(12,5) = 817203140293062400 / 28017687423130915433 \]
\[ a(12,6) = 3103192375787878400 / 252159186808178238897 \]
\[ a(12,7) = -\frac{5995500241469660800}{2269432681273604150073} \]
\[ a(12,8) = -\frac{9922047358867785728}{7432476838072446476547} \]
\[ a(12,9) = 2644979449523238144 / 1252790880491425218647 \]
\[ a(12,10) = 194944099008191232 / 4002526774732987919 \]
\[ a(12,11) = -1253763555768940352 / 4002526774732987919 \]

Coefficients for the solution interpolant.

\[ bt(1,1) = 0 / 1 \]
\[ bt(2,1) = 1 / 2 \]
\[ bt(3,1) = -\frac{531982571}{191676240} \]
\[ bt(4,1) = 35769225887 / 4600229760 \]
\[ bt(5,1) = -\frac{1575297773}{127784160} \]
\[ bt(6,1) = 8567464307 / 766704960 \]
\[ bt(7,1) = -\frac{6203117527}{1150057440} \]
\[ bt(8,1) = 26168659 / 24339840 \]
\[ bt(3,3) = 367775 / 114093 \]
\[ bt(4,3) = -\frac{69380225}{547646} \]
bt(5,3) = 503899925 / 21905856
bt(7,3) = 52204075 / 4694112
bt(3,4) = 3059900 / 798651
bt(5,4) = 48087975 / 1419824
bt(3,5) = 35178 / 12677
bt(5,5) = 12470109 / 405664
bt(3,6) = 247900 / 88739
bt(5,6) = 505000975 / 12778416
bt(3,7) = 576925 / 342279
bt(5,7) = 229116775 / 7301952
bt(3,8) = 206944 / 798651
bt(5,8) = 26960477 / 4791906
bt(3,9) = 576925 / 342279
bt(5,9) = 229116775 / 7301952
bt(3,10) = 92909096 / 1874385
bt(5,10) = 3637555817 / 1150057440
bt(3,11) = 1594323 / 36220
bt(5,11) = 43005633 / 1014160
bt(3,12) = 54918391 / 9996720
bt(5,12) = -12406241683 / 139954080
bt(3,1) = 1 / 1
bt(5,1) = 8567464307 / 14995080
bt(3,3) = -69380225 / 1369116
bt(5,3) = -732232975 / 5476464
bt(3,4) = 367775 / 38031
bt(5,4) = 485205633 / 1014160
bt(3,5) = -69380225 / 1369116
bt(5,5) = 485205633 / 1014160
bt(3,6) = 56581886779 / 1679448960
bt(5,6) = 547412349 / 26657920
bt(3,7) = 1134186781 / 3748770
bt(5,7) = 11077570933 / 93302720
bt(3,8) = 1134186781 / 3748770
bt(5,8) = 11077570933 / 93302720
bt(3,9) = 1134186781 / 3748770
bt(5,9) = 11077570933 / 93302720
bt(3,10) = 1134186781 / 3748770
bt(5,10) = 11077570933 / 93302720
bt(3,11) = 1134186781 / 3748770
bt(5,11) = 11077570933 / 93302720
bt(3,12) = 1134186781 / 3748770
bt(5,12) = 11077570933 / 93302720

Coefficients for the derivative interpolant.

bpt(1,1) = 1 / 1
bpt(3,1) = 35769225887 / 1150057440
bpt(5,1) = 8567464307 / 127784160
bpt(7,1) = 26168659 / 3042480
bpt(3,3) = -69380225 / 1369116
bpt(5,3) = -732232975 / 5476464
bpt(3,4) = 367775 / 38031
bpt(5,4) = 485205633 / 1014160
bpt(3,5) = -69380225 / 1369116
bpt(5,5) = 485205633 / 1014160
bpt(3,6) = 56581886779 / 1679448960
bpt(5,6) = 547412349 / 26657920
bpt(3,7) = 1134186781 / 3748770
bpt(5,7) = 11077570933 / 93302720
bpt(3,8) = 1134186781 / 3748770
bpt(5,8) = 11077570933 / 93302720
bpt(3,9) = 1134186781 / 3748770
bpt(5,9) = 11077570933 / 93302720
bpt(3,10) = 1134186781 / 3748770
bpt(5,10) = 11077570933 / 93302720
bpt(3,11) = 1134186781 / 3748770
bpt(5,11) = 11077570933 / 93302720
bpt(3,12) = 1134186781 / 3748770
bpt(5,12) = 11077570933 / 93302720
bpt(7,3) = $-1563925 / 86928$

bpt(3,4) = $-159297025 / 2395953$

bpt(5,4) = $-159297025 / 2395953$

bpt(7,4) = $-295075 / 7244$

bpt(2,5) = $105534 / 12677$

bpt(3,5) = $1371737 / 25354$

bpt(4,5) = $62350545 / 405664$

bpt(5,5) = $79734075 / 354956$

bpt(6,5) = $414103075 / 1419824$

bpt(7,5) = $-694089 / 14488$

bpt(2,6) = $743700 / 88739$

bpt(3,6) = $-50660075 / 798651$

bpt(4,6) = $2525004375 / 12778416$

bpt(5,6) = $-3734225 / 50708$

bpt(6,6) = $666953675 / 2738232$

bpt(7,6) = $-694089 / 14488$

bpt(2,7) = $206944 / 266217$

bpt(3,7) = $-184979075 / 4107348$

bpt(4,7) = $134802385 / 4791906$

bpt(5,7) = $-116846648 / 2395953$

bpt(6,7) = $41301607 / 1026837$

bpt(7,7) = $-68684 / 5433$

bpt(2,8) = $-20344 / 9055$

bpt(3,8) = $1712426 / 81495$

bpt(4,8) = $3274537 / 43464$

bpt(5,8) = $4710087 / 36220$

bpt(6,8) = $-3274537 / 43464$

bpt(7,8) = $31619 / 9055$

bpt(2,9) = $-92909096 / 624795$

bpt(3,9) = $184979075 / 4107348$

bpt(4,9) = $134802385 / 4791906$

bpt(5,9) = $-116846648 / 2395953$

bpt(6,9) = $41301607 / 1026837$

bpt(7,9) = $-68684 / 5433$

bpt(2,10) = $-92909096 / 624795$

bpt(3,10) = $2268373562 / 1874385$

bpt(4,10) = $-3637555817 / 999672$

bpt(5,10) = $12950247289 / 2499180$

bpt(6,10) = $-53151818167 / 14995080$

bpt(7,10) = $195784743 / 208265$

bpt(2,11) = $4782969 / 36220$

bpt(3,11) = $-1126123479 / 1014160$

bpt(4,11) = $689278977 / 202832$

bpt(5,11) = $-990074583 / 202832$

bpt(6,11) = $485205633 / 144880$

bpt(7,11) = $-64304361 / 72440$

bpt(2,12) = $-54918391 / 3322240$

bpt(3,12) = $56581886779 / 419862240$

bpt(4,12) = $-12406241683 / 27990816$

bpt(5,12) = $33232712799 / 46651360$

bpt(6,12) = $-33090987919 / 59980320$

bpt(7,12) = $547412349 / 3332240$

Table A.4: Coefficients used for the interpolants of ERKN689.

For the Table A.4 we also have

bt(1,i) = 0, i = 1,..., 12,   bt(2,i) = 0, i = 2,..., 12
bt(3,2) = 0, j = 3,..., 8,   bt(4,2) = 0, i = 2,..., 12
bt(1,i) = 0, i = 2,..., 12,   bpt(2,2) = 0, j = 2,..., 7
References


